eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Konputazio Zientziak eta Adimen Artifizialaren Saila
Departamento de Ciencias de la Computación e Inteligencia Artificial

Contributions to learning Bayesian network models from weakly supervised data: Application to Assisted Reproductive Technologies and Software Defect Classification

by

Jerónimo Hernández-González

Supervised by Iñaki Inza and Jose A. Lozano

Dissertation submitted to the Department of Computer Science and Artificial Intelligence of the University of the Basque Country (UPV/EHU) as partial fulfilment of the requirements for the PhD degree in Computer Science

Donostia - San Sebastián, October 2015

To who,
whatever the distance,
is by my side

Acknowledgments

First of all, I would like to thank my advisors Iñaki Inza and Jose Antonio Lozano. Their wisdom, willingness to discuss and patience have made me grow both personally and professionally. Their contribution to the development of this dissertation has been essential. Their wise guidance has enabled me to achieve a satisfactory outcome in this dissertation. Thank you very much.

I would like to given a special mention to my colleagues of the Intelligent Systems Group. I have enjoyed an excellent work environment, where people are always willing to help whenever someone needs it. Moreover, there are many other people at the Faculty of Computer Science with whom I have shared these years. My sincere thanks to all of them.

During these years, I have worked together with many other people. I would like to thank the dedication of the people at the Unit of Assisted Reproduction of the Hospital Donostia, especially Lorena Crisol and Arantza Guembe. In spite of my limited experience, I am pretty sure that an interdisciplinary project is easier with people who work hard as they do. I have also had the pleasure of working with Daniel Rodriguez (University of Alcala) and Rachel Harrison (Oxford Brookes University) in our study of software defect classification. Finally, these words would not be so intelligible without the supervision of John Kennedy, who has revised the English writing of all my publications.

I would like to express my gratitude towards the people and institutions that gave me the opportunity to live a very enriching experience during my stay at the LAMDA group of Nanjing University, China. During the development of this dissertation, I have received economical support from the University of the Basque Country, the Basque and Spanish governments, and the European Commission.

Last but not least, I am very thankful to my family. I have always had their encouragement to meet new challenges and, in spite of the distance, I have always felt their support. I am absolutely sure that I would not be writing these lines without their daily support and confidence. Marta, *Ma*, *Pa*, Lore, *Uska*, Javi, Aitor, Asier, this work is also yours.

Contents

Part III Applications

Part IV Conclusions and Future Work

1

Introduction

The current use of information technologies in many different domains of today's world is leading to a generation of vast amounts of data. The sources of data are growing exponentially. Everyday, new systems are being monitored with more and more sensors and devices that track the status of a machine, register the sales of a shop or control the condition of a patient in a hospital. The storage and management of this vast amount of data has already challenged the capability of information systems. Many times, data is collected without a clear perspective of its posterior utility. However, the need of maintaining a historical archive does not justify the collection of such an amount of data nor the investments that it requires. Thus, the analysis of the collected data with the objective of extracting useful information is, nowadays, an expanding issue. Data holders aim to obtain the knowledge that allows them to improve their everyday procedures. Companies aspire to the expected economical reward of these improvements. Other projects, such as the UN Global Pulse, attempt to acquire knowledge which eventually could enable the standard of living to improve.

Collected data involves limited value since, because it consists of records of past events, it does not provide any new knowledge or information. The entry in the Collins English Dictionary for "data" states that it is "a series of observations, measurements, or facts", whereas "information" is defined as the "knowledge acquired through experience or study". Data is not knowledge on its own; it needs to be analyzed and interpreted in order to extract some kind of useful information from it. Classical data analysis techniques have traditionally been used to extract information from data. However, due to the limited resources available for the experts, these techniques usually required a substantial amount of time to analyze a database. This fact used to restrict the number of cases/variables which could be considered. Dealing with the vast data collections that new information technologies gather requires other kinds of methodologies. Beyond the classical data analysis techniques, data mining and machine learning computational methodologies aim to extract information from data by taking advantage of the high processing power of computers.

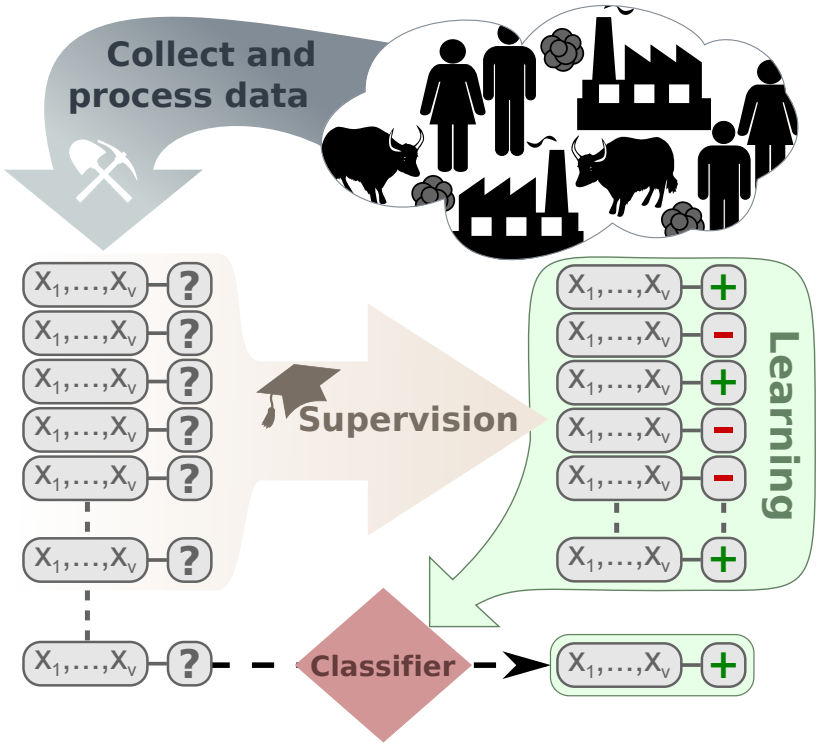


Fig. 1.1. General procedure of supervised classification problems.

These methods use preprocessed and structured data, which is analyzed by means of certain statistical techniques to figure out hidden relationships and extract interesting information.

One of these kinds of computational data analysis methodologies supervised classification techniques. In a classification problem there is a finite and known set of possible categories which the different cases of the problem belong to. In this context, learning means inferring from a set of examples the mapping of cases to categories which underlies the problem. The term *supervised* indicates that the provided set of cases is fully categorized. Thus, supervised classification techniques try to infer from the categorized data the function that maps cases and categories in order to build a classification model that anticipates the category of new cases.

However, the use of the new information technologies has not uniformly reduced the cost of collecting different types of data. Whereas collecting examples has become easier in many problems, obtaining their respective categories remains a hard/costly task. This has given rise to the question of whether it is possible to learn with missing data in the collected set of examples. Specifically, in this dissertation we focus on the difficulty to collect and provide the

complete categorization of the examples which are used to learn the classifier, a novel subfield of machine learning known as *weakly supervised classification* or *partially labeled learning*.

The attempt to solve new real-life problems throughout classification techniques has made evident the difficulty/impossibility of obtaining, on many occasions, a fully/reliably supervised dataset for training as required in the standard supervised classification framework. The popular semi-supervised learning framework [23] already considered a training dataset where only a subset of examples are labeled (categorized). Based on the fact that a fully reliable dataset is unavailable, different authors have proposed to take maximum advantage of the partial class information available in their particular problem/application. This has quickly drawn a wide spectrum of classification frameworks which provide weakly supervised examples, partially labeled due to different causes. Sometimes, it is a problem of accessibility, where the required information cannot be certainly assessed and only partial evidence is available. In this way, the learning from label proportions [81] or learning from candidate labeling sets [115] problems provide some global figures about the class labels of the examples separated in subsets. In others cases, non-exhaustive labeling processes, which are cheaper and faster than the thorough and exact traditional strategies, are carried out to obtain the training information of supervision [35, 100, 101]. The referred powerful information technologies have also been used to cheaply label the examples of a problem. This is the case of learning from crowds [141], where many labelers of arguable reliability subjectively categorize the collected examples, providing multiple non-expert labelings that can be combined to learn a reliable classifier. In this dissertation, our first contribution explores and describes the subfield of weakly supervised classification, clearly delimiting what *weak supervision* is and what it is not. Moreover, another two contributions explore the possibility of learning to classify in two problems of the field with datasets that lack a reliable labeling. Both problems are studied in-depth, and methodologies to learn from their characteristic kind of data are proposed. Finally, the proposed methodologies have been adapted to solve two real applications.

1.1 Contributions of the dissertation

The contributions of dissertation are organized in three parts: base theory, methodological developments and applications.

In the first part, the theoretical bases of this dissertation are presented. Specifically, in Chapter 2 supervised classification and its extension to the multi-dimensional framework (where several class variables are jointly predicted) are defined. Standard supervised classification is then used as a guide to explore the field of weakly supervised classification. Precisely, our first contribution consists of a three-axes taxonomy of weakly supervised classification problems that proposes a novel ordering of the field. Each of the three axes

represents a fundamental characteristic for depicting weakly supervised classification problems: all these classification frameworks are characterized by the *models of supervision* that they implement in the learning and prediction stages and by their *instance-label relationship*. Similarities and differences between different weakly supervised frameworks are identified, and unexplored areas that could lead to new challenging frameworks are characterized.

In Chapter 3, Bayesian network classifiers (BNCs) are presented, together with their base concepts and properties. Specifically, the four kinds of graph structures representative of the BNCs used in this dissertation (naive Bayes, tree-augmented naive Bayes, K-dependence Bayesian classifier and a structure specific for multi-dimensional classification) are presented. General strategies for learning this kind of probabilistic graphical models from —both complete and partial— data are also presented.

In Part II, our methodological contributions for learning from weakly supervised problems are presented. The tackled problems are learning from label proportions (LLP) and learning from crowds (CrL). Both are methodologies based on the (Structural) Expectation-Maximization strategy (Section 3.3.2) for learning Bayesian network classifiers (Section 3.4).

Specifically, our proposal for the LLP problem, where the only information of supervision provided consists of label proportions associated with subsets of instances, is presented in Chapter 4. The problem is analyzed, the usefulness of the partial class information provided in a range of scenarios of this problem is assessed, and the class uncertainty that it introduces is characterized. Our SEM-based proposal, which shows a competitive behavior regarding the state-of-the-art methods, can use two different exhaustive configurations in the case of low class uncertainty and, by means of a MCMC approximate procedure, scales well when the uncertainty of the problem grows. The statistical tests carried out looking for differences among the different versions of the method do not show significant differences between the exhaustive and the approximate probabilistic versions.

The second weakly supervised problem analyzed in Part II is the problem of learning from crowds (Chapter 5). For the sake of simplicity, the binary classification scenario is analyzed in the case of learning when the labels (categories) of the training examples are provided by a set of non-expert annotators. The performance of basic techniques is analyzed in different scenarios, establishing those where non-trivial techniques are expected to perform better. Our proposal for learning BNCs in multi-dimensional classification problems is based on the EM strategy. The reliability of the annotators is estimated and their opinions are weighted accordingly. This has been achieved by the adaptation of the counting procedure for estimating the maximum likelihood parameters of the BNCs. Different configurations of the model estimate and combine the reliability weights in different ways. Our proposal has been tested in a set of experiments learning multi-dimensional Bayesian network classifiers from synthetic and real datasets which are previously transformed to the multi-dimensional learning from crowds framework.

Two real applications related with each of the weakly supervised classification problems theoretically studied in Part II are explored in Part III. Each application is analyzed in a different chapter of this dissertation (Chapters 6 and 7), proposing an adapted methodology for dealing with the specific weakly supervised classification problems resulting from both real applications.

In Chapter 6, a study of the assisted reproductive technologies (ART) problem is analyzed in an integral way through the use of weakly supervised techniques. Our solution uses all the information collected by physicians during the whole ART procedure, also considering examples of uncertain fate for model learning. Four different approaches are used: standard supervised classification, positive-unlabeled learning, learning from label proportions and, a novel weakly supervised framework called learning from positive-unlabeled proportions (PUP). State-of-the-art techniques are used to solve the standard and positive-unlabeled learning problems. For the LLP problem, our methodological contribution presented in Chapter 4 is applied. The exposed SEM-based methodology has been extended to deal with the PUP problem, which only provides a minimum number of positive examples for each associated subset of instances. Some results of clinical relevance have been inferred. The learnt classifiers that predict the viability of a cycle show a good performance. The relevance of the cycle features for determining embryo implantation is appreciated, as well as the fact that the collected data does not fully describe the embryo implantation, but it does describe the embryo development. Obtained classifiers have been proved to rank the medium-quality embryos of our case study more consistently than the currently conducted embryo selection criteria. Thus, their probabilistic assessment could be used as an alternative embryo score.

In the field of software engineering, the second real application (Chapter 7) aims to classify software defects which are reported by users throughout issue tracking systems. The labeling, carried out by human sources, is commonly incomplete, noisy or erroneous. In our specific case, a set of defects reported for the Compendium software project and labeled by five annotators have been used to illustrate our proposals to the learning from crowds paradigm. To the extent of our knowledge, the crowd learning framework is a novel approximation in the software engineering literature. In order to deal with this real application, the methodology proposed in Chapter 5 has been adapted to this multi-class problem. Moreover, two complementary approaches that specifically deal with two characteristics of the problem (a binary decomposition strategy for the multi-class problem and a sampling boosting strategy for the imbalance nature of the problem) have been successfully adapted to the learning from crowds framework. In general, the experimental results show the enhanced performance achieved by our crowd learning solutions regarding standard supervised learning using the most-voted labels.

Finally, a fourth part is included where conclusions, future work and publications supporting this dissertation are presented.

Part I

Background

2

Supervised Classification

Supervised classification [125] is one of the most popular fields of machine learning. Its objective is to learn a classifier that reliably approximates a classification task which is generalized/inferred from a set of categorized examples of a problem of interest. The learnt classifier is posteriorly used in the prediction stage to anticipate the class label of new *unlabeled* examples. In this context, the term "supervised" indicates that, in the learning stage, the examples are *always* provided with their real class label (category).

However, a fully supervised dataset for training, as required in the standard framework, cannot always be provided. In some problems, only weakly supervised data can be obtained. Solutions proposed for learning from different kinds of partially labeled data have led to the foundation of a new subfield of machine learning called *weakly supervised classification* [83, 122, 185] (a.k.a. partially supervised learning). Weak supervision refers to the lack of a full supervision for the provided data, which conditions the process of learning a classifier. Nevertheless, a classifier can be efficiently inferred from this partial class information in most of the occasions. Similarly, the examples for prediction are traditionally provided completely unlabeled, although there exist situations where partial class information is also available during the prediction stage [35, 101]. The kind of partial class information available in the prediction stage has to be known previously to learn the classifier: the built classifier has to know how to take advantage of that extra information. In this way, the performance of the obtained classifiers can be easily enhanced [35, 101].

In this chapter, we introduce the formal definition of supervised classification together with the related notation that will be used throughout this dissertation. The definition of multi-dimensional supervised classification is also formalized. Next, the first contribution of this dissertation, which consists of a taxonomy of weakly supervised classification problems, is proposed. Apart from (a) the type of supervision in the data provided for learning and (b) the type of supervision in the data provided for prediction, the taxonomy considers another axis: (c) the instance-label relationship defined by the problem (e.g., in the multi-label framework [178] each instance is categorized

by a set of labels). By means of the proposed taxonomy, the (dis)similarities between different classification frameworks can be assessed and discussed. Finally, the weakly supervised classification problems explored throughout this dissertation are formally and individually defined.

2.1 Supervised classification

2.1.1 Base concepts

A random variable X is a function that assigns a value x to the outcomes of a random experiment, and a random vector $\boldsymbol{X} = (X_1, \ldots, X_n)$ involves n random variables, X_j. The n-tuple $\boldsymbol{x} = (x_1, \ldots, x_n)$ resulting from assessing a value from each random variable is an instance. In this dissertation, we only consider discrete random variables. Thus, each random variable X_j has an associated set of possible values $x_j \in \{1, \ldots, d_j\}$ of cardinality d_j.

The joint probability distribution of $\boldsymbol{X}$ is given by $p(\boldsymbol{X} = \boldsymbol{x})$ or just $p(\boldsymbol{x})$. We denote the marginal probability distribution of X as $p(x)$. The joint probability distribution of X_i and X_j is represented by $p(x_i, x_j)$, and $p(x_i|x_j)$ is the conditional probability distribution of X_i given $X_j = x_j$.

2.1.2 Standard supervised classification

Formally, a supervised classification problem is described by a set of n predictive variables $\boldsymbol{X} = (X_1, \ldots, X_n)$ and a class variable C. Each example of the problem is an instance $(\boldsymbol{x}, c) = (x_1, \ldots, x_n, c)$ of the random vector $(\boldsymbol{X}, C)$ with the value $x_j \in \mathcal{X}_j$ that takes each predictive variable X_j in a specific situation and the associate class value $c \in \mathcal{C}$. The instance space $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ is defined as the set of all possible instances $\boldsymbol{x}$, and the set of values that the class variable can take, a.k.a. class labels, form the label space $\mathcal{C}$.

A classifier $\hat{\Psi}$ is a function that maps the instance space into the label space:

$$\begin{aligned} \hat{\Psi}\colon \quad & \mathcal{X} \to \mathcal{C} \\ & (x_1, \ldots, x_n) \mapsto c \end{aligned}$$

In statistical classification, the existence of an unknown instance-label joint probability distribution $p(\boldsymbol{x}, c)$ is assumed. In this context, a classification rule is a classifier function that maps instances to class labels based on an underlying probability distribution. For example, the standard *winner-takes-all* rule returns the class label with the largest conditional probability given the instance $\boldsymbol{x}$:

$$\hat{c} = \hat{\Psi}(\boldsymbol{x}) = \operatorname*{argmax}_{c} p(c|\boldsymbol{x}) \tag{2.1}$$

Therefore, learning techniques infer the generative probability distribution $p(\boldsymbol{x}, c)$ or, directly, the conditional probability distribution $p(c|\boldsymbol{x})$ from a set of examples $D = \{(\boldsymbol{x}^1, c^1), \ldots, (\boldsymbol{x}^N, c^N)\}$ which are assumed to be independent and identically distributed (i.i.d.) samples of the original unknown probability distribution. A classifier is learnt such that it minimizes some misclassification cost function, the *0/1 loss function* (constant 1-value cost for misclassified examples) being the simplest and most popular one. An advantageous classifier will generalize and, given a new unlabeled example, will be able to accurately infer its class label. This is measured by performance evaluation metrics. One of these, the *classification error*, is defined as the probability that the classifier $\hat{\Psi}$ misclassifies an instance $\boldsymbol{x}$,

$$\epsilon(\hat{\Psi}) = p(\hat{\Psi}(\boldsymbol{X}) \neq C) = E_{(\boldsymbol{x},c)\sim p(\boldsymbol{X},C)} \mathbb{I}[\hat{\Psi}(\boldsymbol{x}) \neq c]$$

where $\mathbb{I}[condition]$ returns 1 if *condition* is true and 0 otherwise. Another commonly used performance metric, the *accuracy*, is simply $acc(\hat{\Psi}) = 1 - \epsilon(\hat{\Psi})$.

2.1.3 Multi-dimensional supervised classification

A multi-dimensional (MD) classification problem [6, 148] is described by a set of n predictive variables $\boldsymbol{X} = (X_1, \ldots, X_n)$ and m class variables $\boldsymbol{C} = (C_1, \ldots, C_m)$. Accordingly, each example of the problem is an instance $(\boldsymbol{x}, \boldsymbol{c}) = (x_1, \ldots, x_n, c_1, \ldots, c_m)$ of the random vector $(\boldsymbol{X}, \boldsymbol{C})$ with the value $x_j \in \mathcal{X}_j$ that takes each predictive variable X_j in a specific situation and the associate class values $c_k \in \mathcal{C}_k$. The instance space $\mathcal{X}$ is defined as in the standard unidimensional problem. However, in this framework the label space $\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_m$ denotes all the possible joint label assignments $\boldsymbol{c}$ to the m class variables (label configurations).

In the adaptation of the general definition of a classifier —a function that maps the instance space into the label space— to the MD framework, a multidimensional classifier $\hat{\Psi}$ returns a m-tuple of class labels:

$$\begin{aligned} \hat{\Psi}\colon \quad & \mathcal{X} \to \mathcal{C} \\ & (x_1, \ldots, x_n) \mapsto (c_1, \ldots, c_m) \end{aligned}$$

In statistical classification, the existence of an unknown instance-label joint probability distribution $p(\boldsymbol{x}, \boldsymbol{c})$ is assumed. In this context, a classification rule maps instances to label configurations based on an underlying probability distribution. The straightforward extension of the winner-takes-all rule, known as *joint classification rule* is:

$$\hat{\boldsymbol{c}} = \hat{\Psi}(\boldsymbol{x}) = \operatorname*{argmax}_{\boldsymbol{c}} p(\boldsymbol{c}|\boldsymbol{x})$$

In this framework, learning techniques infer the generative probability distribution $p(\boldsymbol{x}, \boldsymbol{c})$ or, directly, the conditional probability distribution $p(\boldsymbol{c}|\boldsymbol{x})$ from a set of examples $D = \{(\boldsymbol{x}^1, \boldsymbol{c}^1), \ldots, (\boldsymbol{x}^N, \boldsymbol{c}^N)\}$ which are assumed to be i.i.d. samples of the original unknown probability distribution. Again, the objective is to learn a classifier that generalizes and, given a new unlabeled example, is able to *accurately* infer its label configuration. However, the generalization of the classification error to MD domains,

$$\epsilon(\hat{\Psi}) = p(\hat{\Psi}(\boldsymbol{X}) \neq \boldsymbol{C}) = E_{(\boldsymbol{x},\boldsymbol{c})\sim p(\boldsymbol{X},\boldsymbol{C})}\mathbb{I}[\hat{\Psi}(\boldsymbol{x}) \neq \boldsymbol{c}]$$

and the corresponding global accuracy, $acc(\hat{\Psi}) = 1-\epsilon(\hat{\Psi})$, are very demanding. That is, learning a classifier minimizing/maximizing one of these scores is a hard task because failing in one of the m predicted class labels penalizes the whole (joint) prediction. A simple alternative is the per-class performance evaluation, where each class variable is evaluated separately,

$$\epsilon_k(\hat{\Psi}) = p(\hat{\Psi}_k(\boldsymbol{X}) \neq C_k) = E_{(\boldsymbol{x},\boldsymbol{c})\sim p(\boldsymbol{X},\boldsymbol{C})}\mathbb{I}[\hat{\Psi}_k(\boldsymbol{x}) \neq c_k]$$

where $\hat{\Psi}_k(\boldsymbol{x})$ is the label assigned by classifier $\hat{\Psi}$ to the class variable C_k. The output of this evaluation metric is a vector $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_m)$.

2.2 Weakly supervised classification

Abstracted from the generative probability distribution, with a non-statistical perspective, the existence of an unknown target function $\Psi : \mathcal{X} \rightarrow \mathcal{C}$ that (i) *individually categorizes each instance with a single label* is usually assumed in standard supervised classification [125]. Learning techniques infer (ii) *from a set of fully labeled examples* $D = \{(\boldsymbol{x}^1, c^1), \ldots, (\boldsymbol{x}^N, c^N)\}$ of the problem a mapping function or classifier $\hat{\Psi}$ that approximates the real function Ψ. The objective is to build a classifier $\hat{\Psi}$ that accurately predicts the class label c of (iii) *new unlabeled examples* $(\boldsymbol{x}, ?)$.

A quick look at recent literature is enough to realize that the increasing number and variety of non-standard supervised classification problems cannot be described by means of this standard definition. In the previous paragraph, three well-established components of the definition have been emphasized. At least one of the indicated components is not fulfilled by the non-standard classification frameworks collected for this work. First of all, not all the problems involve samples which can be described by means of an instance-label pair: e.g., the multi-label framework [178], where the examples are categorized with one or more class labels. Secondly, some frameworks cannot provide a fully labeled dataset for training: e.g., the semi-supervised framework [23], where not all the training examples are labeled. Thirdly, certain class information can be known for the examples at prediction time: e.g., someone could be interested in categorizing a group of examples and it is known that they belong to different categories [101]. Each of these ideas, the three axes on which

		Categorization	
		SL	ML
Example	SI	$\Psi : \mathcal{X} \to \mathcal{C}$	$\Psi : \mathcal{X} \to 2^{\mathcal{C}}$
	MI	$\Psi : 2^{\mathcal{X}} \to \mathcal{C}$	$\Psi : 2^{\mathcal{X}} \to 2^{\mathcal{C}}$

Table 2.1. Four possible definitions of the target function Ψ. An example is composed of a single (SI) or multiple (MI) instances. The categorization is composed of a single (SL) or multiple (ML) class labels.

the proposed taxonomy is based, will be discussed in-depth in the following subsections from the point of view of weakly supervised classification.

2.2.1 Instance-label relationship

In standard supervised classification, each instance represents an example of the problem and is categorized with a single class label (single-instance single-label, SISL). There exist other popular state-of-the-art frameworks that do not follow this standard instance-label (IL) relationship: in the multi-label classification framework [178], each single instance is categorized with multiple (one or more) class labels (SIML); in the multiple-instance learning problem [47], a set of instances (which represents an example) is categorized with a single class label (MISL); and the multi-instance multi-label framework [201] involves both examples of multiple instances and categorizations of multiple labels (MIML).

In general, it may be agreed that a classifier $\hat{\Psi}$ is built as an approximation of the real unknown target function Ψ. The definition of the domain and image of the target function Ψ determines the instance-label relationship of a problem. On the one hand, the domain of Ψ comprises all the possible examples of the problem, which compose the instance space $\mathcal{X}$. There are two possible configurations: each example is represented (a) by a single instance, as in the standard framework [125], where the domain of Ψ matches the instance space $\mathcal{X}$, or (b) by multiple instances [47], where the domain of Ψ is the power set $2^{\mathcal{X}}$. On the other hand, the image of the target function Ψ comprises all the possible categorizations, which compose the label space $\mathcal{C}$. There are also two possible configurations: a categorization is represented (a) by a single class label, as in the standard framework, with the image of Ψ matching the label space $\mathcal{C}$, or (b) by multiple class labels [178], where the image is the power set $2^{\mathcal{C}}$. Thus, both the examples and the categorizations can show a single or multiple configuration. This idea leads to a first subdivision of classification problems: a different instance-label relationship is observed for each of the four possible definitions of the unknown target function Ψ (Table 2.1).

Note that the instance-label relationship can be used for characterizing both weakly and standard supervised classification problems, i.e., it is not

an exclusive feature of weakly supervised classification problems. In the related literature, the interested reader can find classification problems with an alternative IL relationship which provide standard fully supervised data (e.g., all the illustrative frameworks mentioned so far in the current subsection [178, 47, 201]), weakly supervised classification problems with the standard IL relationship [35, 81] or problems that combine an alternative IL relationship with weak supervision [175, 195]. The inclusion of this characteristic as an axis of our taxonomy of weakly supervised classification problems allows us to leave this feature out of the discussion over weak supervision —the IL relationship has been confused several times with weak supervision [64]. From this section on, *example* and *categorization* are used as two general terms that take a particular meaning according to the instance-label relationship defined by the target function of each specific problem.

2.2.2 Supervision in the learning stage

According to the standard definition of supervised classification, a set of fully supervised examples has to be provided in the learning stage in order to learn a classifier. Loss functions, performance evaluation, feature subset selection or discretization techniques are a few examples of the different procedures that, during the learning stage, take advantage of this requirement.

However, collecting such a complete set of examples is not always possible. Many authors have dealt with classification problems in which the class information provided for the training examples is partial. The most popular framework with this characteristic is probably the semi-supervised learning [23] problem. Over the years, the determination of the researchers of the machine learning community, who have attempted to collect any kind of class information available in their specific problem in order to learn with as much supervision as possible, has led to the emergence of many different types of supervision. For instance, in the partial labels problem [35], each training example is provided with a set of candidate categorizations that includes the real one. In the case of the multi-label learning with weak labels problem [175], the set of labels provided with each example is actually a subset of the set of labels that compose the real full categorization of the SIML example. In this paper, we use the term *supervision model* to refer to the specific type of supervision used to categorize the examples of a weakly supervised classification problem. In Table 2.2, a representative set of supervision models collected in the literature is briefly described.

In these novel frameworks, the use of a weak supervision model and the consequent absence of a fully reliable labeling for the training dataset prevents us from using standard supervised classification techniques. Specific methodologies have been proposed for dealing with different kinds of weakly supervised data. Eventually, proposed techniques have been proved to successfully learn from the specific kind of data.

Model	References	Description
Full-supervision	[125, 178, 47, 201]	For each example, complete class information is provided.
Unsupervision	[125]	No class information is provided with the examples.
Semi-supervision	[23]	Part of the examples are provided fully supervised. The rest are unsupervised.
Positive-unlabeled	[19, 111, 60, 164]	Part of the examples are provided fully supervised, all of them with the same categorization. The rest are unsupervised.
Candidate labels	[35, 70, 90]	For each example, a set of class labels is provided. In this set, the class label(s) that compose the real categorization of the example are included.
Probabilistic labels	[94]	For each example, the probability of belonging to each class label is provided. This probability distribution is expected to assign high probability to the real label(s).
Incomplete	[14, 175, 196]	For each example, a subset of the labels that compose its real categorization is provided (SIML or MIML, Table 2.1).
Noisy labels	[12, 202]	For each example, complete class information is provided, although its correctness is not guaranteed.
Crowd	[141, 192]	For each example, many different non-expert annotators provide their (noisy) categorization.
Mutual label constraints	[100, 101, 162]	For each **group** of examples, an explicit relationship between their class labels is provided (e.g., all the examples have the same categorization).
Candidate labeling vectors	[115]	For each **group** of examples, a set of labeling vectors (including the real one) is provided. A labeling vector provides a class label for each examples of a group.
Label proportions	[128, 136, 81]	For each **group** of examples, the proportion of examples belonging to each class label is provided.

Table 2.2. Collection of supervision models.

2.2.3 Supervision in the prediction stage

Traditionally, class information has only been supplied for the data provided in the learning stage. According to the standard definition, a supervised classification problem provides completely uncategorized examples for prediction. However, some authors have already proposed using weakly supervised examples in the prediction stage. The full-class set problem [101], where a classifier is learnt from a traditional set of fully categorized examples, stands out because it implements a weak supervision model in the prediction stage. In the motivating application, a teacher who wants to automatically register the attendance at their lessons uses a set of individually identified (categorized) photographs of the students to learn a classifier. Then, for the faces detected in a photograph of the whole classroom, the classifier individually predicts the identity of each face knowing that no identity can be predicted twice. That is, the examples for prediction implement a *mutual label constraints* supervision model: *among the examples of the provided group, none has the same categorization*. Cour et al. [35] were also motivated by another application in which face-images have to be classified according to their identity. In this case, at the prediction stage each face is provided to the classifier with a set of possible

identities (allegedly, considerably smaller than the complete set of identities). That is, their motivating problem implements the *candidate labels* supervision model (see Table 2.2) in the prediction stage. With this information, the classifier selects the predicted identity among a reduced set of candidates, and not among the set of all possible identities ($\mathcal{C}$). An interesting particularity of this problem is that it implements the same supervision model in the learning stage; it infers the classifier also from a set of examples labeled with candidate labels too. In this way, a weakly supervised classification problem that shows the same supervision model in both stages is solved.

The weak supervision of the examples provided in the prediction stage may enhance the accuracy of the learnt classifiers. Partial class information can be straightforwardly used to skew the probabilities of predicting the different possible categorizations or, even, to discard some of them. Thus, contrary to the consequences derived from their use in the learning stage, the use of weak supervision models in the prediction stage is expected to enhance the performance of the obtained classifiers. There is disparity in the computational cost arising from the use of supervision models in this stage: Whereas the *candidate labels* model reduces the number of possible categorizations that the classifier considers (complexity reduction), the *mutual label constraints* model implies joint predictions that can make the prediction task more complex. However, in all the cases, the improvement in terms of classification performance is unquestionable for all the supervision models. In spite of the benefits, state-of-the-art weakly supervised frameworks rarely consider a weak supervision model in the prediction stage. Certainly, partial class information in prediction cannot be provided for all the problems. The class information is inherently available in the prediction stage only in a limited set of problems and carrying out a *costly* process for collecting weakly supervised examples in this stage makes no sense: Precisely, classifiers are built to anticipate this information. However, whenever partial class information is available for prediction, it is highly advisable to build classifiers that take advantage of it. Note that while the classifier is being built, it has to be taught to take advantage of the specific partial class information provided for the prediction examples. That is, the supervision models which a weakly supervised classification problem follows in both learning and prediction stages determine the process of building a solution.

2.3 A taxonomy of weakly supervised classification problems

In the last few years, a spectrum of classification problems and applications which depart from the standard definition of supervised classification have been addressed by the community. In this paper, we characterize weakly supervised classification problems according to the supervision models implemented in both learning and prediction stages. The consideration of the instance-label

				SUPERVISION MODEL	
Problem	Description	Application (e.g.)	IL rel.	Learning	Prediction
Standard problem [125]	Learning with full categorized examples	Handwritten digit recogn.	SISL	Full-supervision	Unsupervision
Semi-supervised [23]	Learning with categorized and uncategorized examples	Text classification	SISL	Semi-supervision	Unsupervision
Positive-unlabeled [19]	Learning with examples of a category and other uncategorized examples	Spam detection, Gene prediction	SISL	Positive-unlabeled	Unsupervision
Mislabeled data [12] Ambiguous labels [202]	Learning with maybe wrong-categorized examples	Subjective labeler	SISL	Noisy Labels	Unsupervision
Partial labels [35]	Learning and prediction with uncategorized examples that have a set of possible categorizations	Classifying photographs with captions	SISL	Candidate labels	Unsupervision / Candidate labels
Multiple labels [94]	Learning with uncategorized examples that, with some probability, belong to a certain categorization	Bioinformatics	SISL	Probabilistic labels	Unsupervision
Partial equivalence relations [100]	Learning with groups of examples of the same/different categorization	Computer vision	SISL	Mutual label constraints	Unsupervision
Full-class set [101]	Prediction for a group of examples, all of them with a different categorization	Automatic attendance recording	SISL	Full-supervision	Mutual label constraints
Label proportions [81] Aggregate outputs [128]	Learning with groups of examples only knowing how many of them belong to each categorization	Embryo Selection, Polls prediction	SISL	Label proportions	Unsupervision
Candidate labeling sets [115]	Learning with groups of examples and sets of possible categorizing vectors	Classifying photographs with captions	SISL	Candidate labeling vectors	Unsupervision
Learning from crowds [141, 192]	Learning with examples categorized with many candidate noisy categorizations	Image annotation	SISL	Crowd	Unsupervision
Multi-label [178]	Learning with examples that belong to several categorizations at the same time	Film genre prediction	SIML	Full-supervision	Unsupervision
Semi-supervised multi-label [26]	Learning with examples categorized with multiple labels or uncategorized	Text categorization	SIML	Semi-supervision	Unsupervision
ML with weak label [175] ML incomplete class [14]	Learning with examples categorized with a subset of the real multiple labels	Image annotation	SIML	Incomplete	Unsupervision
Set classification [132]	Prediction for a group of examples, all of them with the same categorization	Face recognition with multiple photos	SIML	Full-supervision	Mutual label constraints
MIL [47]	Learning with multiple-instance examples that are positive if at least one of their instances is	Molecule activation prediction	MISL	Full-supervision	Unsupervision
G-MIL [190]	Learning with examples represented by several instances with generalized function for positives	Key-and-lock prediction problem	MISL	Full-supervision	Unsupervision
MISSL [138]	Learning with categorized and uncategorized multiple-instances examples	Content-based image retrieval	MISL	Semi-supervision	Unsupervision
MIML [201]	Learning with examples represented with several instances that belong to several categorizations	Classifying texts, images or videos	MIML	Full-supervision	Unsupervision
SSMIML [195]	Learning with multiple-instance examples categorized with multiple labels or uncategorized	Video annotation	MIML	Semi-supervision	Unsupervision
MIML with weak labels [196]	Learning with multiple-instance examples categorized with a subset of the real multiple labels	Image annotation	MIML	Incomplete	Unsupervision

Table 2.3. Brief description of classification problems and characterization according to the three axes of the taxonomy.

relationship of the problems allows us to delimit the field of weakly supervised classification, highlighting the differences with other non-standard classification problems which have been commonly misconceived as weakly supervised problems. Using these three features to organize the spectrum of problems, a taxonomy of weakly supervised classification problems is straightforwardly obtained.

Table 2.3 shows a summary of the non-standard supervised classification problems collected for this study. The name, a short description and an example of a real application are used for depicting each problem, which is finally characterized according to the three axes of the taxonomy. Although the collection of problems is not exhaustive, we consider that it is large, varied and representative for the exposed objective of illustrating the taxonomy. Other problems could be easily incorporated to this taxonomy describing them according to the three axes.

Similarities between supervision models. In their attempt to describe the weakly supervised classification field, García-García and Williamson [64] used the name of *degrees of supervision*. This suggestive name seems to allude to a certain degree of supervision or amount of class information related to each supervision model. Thus, one might be tempted to meet the challenge of proposing a rank of supervision models according to their amount of class information. However, this intuition is not completely precise: the same supervision model provides a different amount of information in different scenarios. Consider, for example, the *candidate labels* supervision model. It can be agreed that an example with two candidate categorizations involves more class information than another with three candidates. Accordingly, a dataset with this supervision model that has, on average, two candidate categorizations per example involves more information than another dataset with three candidates on average. Therefore, although it is unquestionable that *full-supervision* and *unsupervision* are respectively the models with the most and the least class information, it would be unfair to propose such a rank of supervision models regarding the amount of class information.

However, although for the sake of simplicity no subdivision of supervision models has been explicitly displayed in the taxonomy (nor in the corresponding tables), they can be roughly separated in two groups: supervision models that provide class information for each example individually and those that provide class information jointly for a group of examples. On the one hand, let us consider the novel *probabilistic labels* and *candidate labels* supervision models (Table 2.2) for illustrating the similarities between supervision models in this first group. In the latter, a group of categorizations (including the correct one) is provided for each example, whereas in the former a probability distribution indicates the probability of each possible categorization being the real one. The *probabilistic labels* model can represent the *candidate labels* model (same probability, $1/t$, for the t candidate labels and null probability for the rest) but not the other way around. Let us now consider the *incom-*

plete supervision model, defined only for SIML or MIML relationships. The multiple label configuration implies that the image of the target function Ψ is the power set of the label space ($2^{\mathcal{C}}$, Table 2.1); i.e., the set of all the possible categorizations comprises all the subsets of $\mathcal{C}$. In this context, the *incomplete* supervision model provides, for each example, a subset of the labels $\hat{\boldsymbol{c}}$ which are included in the real categorization, $\hat{\boldsymbol{c}} \subseteq \boldsymbol{c}$. This information can be used to discard those categorizations which do not include the provided labels. Thus, any remaining subset of labels $\boldsymbol{c}'$ that does contain all the labels in $\hat{\boldsymbol{c}}$ ($\boldsymbol{c}'$ fulfills that $\hat{\boldsymbol{c}} \subseteq \boldsymbol{c}' \subseteq \mathcal{C}$) can be considered, in this context, a *candidate* categorization. Other members of this first group are the standard full-supervision, semi-supervision and positive-unlabeled models. On the other hand, the second group of supervision models comprises the *label proportions* and the *mutual label constraints* models (see Table 2.2), which can be seen as restrictive versions of the *candidate labeling vectors* model. The *label proportions* model can be described by those labeling vectors that assign class labels to the examples fulfilling the provided label proportions of the group. An analogous reconsideration can be used to fit the *mutual label constraints* into the description of the *candidate labeling vectors* model. For example, if a constraint indicates that all the examples of a group belong to the same class, the corresponding consistent labeling vectors are those that assign the same label to all the examples of the group.

Dissimilar problems. By means of this taxonomy, basic differences between problems that otherwise could be considered as similar are noticed. For example, the use of groups of instances, a feature shared by the learning from label proportions (LLP) [81] and the multiple-instance learning (MIL) [47] problems, has a different nature in both problems. On the one hand, the MIL problem involves a target function that defines a MISL instance-label relationship, that is, an example is described by means of a group of instances. On the other hand, each single instance represents an example in the LLP problem (i.e., it follows the standard SISL relationship) but the available class information is not specific enough to individually categorize each example: *"In a group of three instances, two are positive examples and one is negative"* is valid class information (*label proportions* supervision model) in this problem. A similar distinction can be made among frameworks that provide a group of labels for each example: in the multi-label (ML) framework [178] —which follows a SIML instance-label relationship— each categorization is composed by a group of labels, whereas the partial labels problem [35] —which follows the standard SISL instance-label relationship— implements the *candidates labels* supervision model, where the provided group of labels is a set of candidate categorizations that includes the real one. Such a case of ambiguity, where a problem of interest cannot be certainly defined according exclusively to the provided data, should be elucidated by supplementary expert knowledge. Thus, it is worth emphasizing the importance of a clear definition of the problem that, by means of our three-axes taxonomy, can be placed in

SISL	SIML	MISL	MIML
Standard problem [125] Semi-supervised [23] Positive-unlabeled data [19] Mislabeled data [12] Ambiguous labels [202] Partial labels [35] Multiple labels [94] Partial equivalence relations [100] Full-class set [101] Label proportions [81] Aggregate outputs [128] Candidate labeling sets [115] Learning from crowds [141]	Multi-label [178] Semi-supervised multi-label [26] Set classification [132] ML with weak label [175] ML with incomplete class [14]	MIL [47] G-MIL [190] MISSL [138]	MIML [201] SSMIML [195] MIML with weak label [196]

Table 2.4. Classification problems distributed according to their instance-label relationship.

the field of weakly supervised classification, revealing the (subtle) similarities and differences with other frameworks. Being aware of these (dis)similarities, future researchers may find the best way to deal with the weakly supervised data of their (novel) problems in order to provide a suitable technique that efficiently deals with it.

Unexplored frameworks. The gaps observed in our taxonomy may be seen as further challenges, unexplored frameworks for the research community which could involve real applications. For example, it is worth noting the reduced number of addressed problems that combine both a non-standard instance-label relationship and a scheme of supervision models alternative to the standard one (*full-supervision* in learning stage, *unsupervision* in prediction). As shown in Table 2.4, most of the current weakly supervised frameworks show the standard single-instance single-label (SISL) relationship. Although the ML, MIL and, more recently, MIML paradigms have received considerable attention, only the *semi-supervision* model [23, 26, 138, 195] and the *incomplete* supervision model [14, 175, 196] have been extensively applied to problems with non-standard instance-label relationships.

Table 2.5 shows the distribution of the collected problems according to the supervision models implemented in their learning (rows) and prediction (columns) stages. Note that those supervision models that do not make sense in each stage are not shown: e.g., the *unsupervision* model in the learning stage or the *full-supervision* model in the prediction stage. Neither are the *probabilistic labels*, *candidate labeling vectors*, *label proportions*, *incomplete* and *crowd* supervision models displayed in columns because these have never been used for prediction. In general, the trend towards using the standard scheme of supervision models or, at least, using the *unsupervision* model in the prediction stage remains strong. Few works [101, 35, 132] that take advantage of weakly supervised data in the prediction stage appear in the literature. Although it is evident that the availability of weakly supervised data in the prediction

SUPERVISION MODEL IN LEARNING STAGE	SUPERVISION MODEL IN PREDICTION STAGE: Unsupervision	Candidate labels	Mutual label constraints
Full-supervision	Standard problem [125] Multi-label [178] MIL [47] G-MIL [190] MIML [201]		Full-class set [101] Set classification [132]
Semi-supervision	Semi-supervised [23] Semi-supervised multi-label [26] MISSL [138] SSMIML [195]		
Positive-unlabeled	Positive-unlabeled data [19]		
Candidate labels	Ambiguously labeled data [201] Partial labels [35]	Partial labels [35]	
Probabilistic labels	Multiple labels [94]		
Incomplete	ML weak label [175] ML incomplete class [14] MIML weak label [196]		
Noisy labels	Mislabeled data [12] Ambiguously labeled data [202]		
Crowd	Learning from crowds [141]		
Mutual label constraints	Partial equivalence relations [100]		
Candidate labeling vectors	Candidate labeling sets [115]		
Label proportions	Label proportions [81] Aggregate outputs [128]		

Table 2.5. Classification problems distributed according to the supervision models implemented in the learning and prediction stages.

stage is not as usual as in the learning stage, the number of problems that use a weak supervision model in the prediction stage is still strikingly reduced. It is especially noteworthy due to the fact that this kind of information can be easily used to enhance the performance of the learnt classifiers. Only one problem that learns and predicts with weakly labeled data has been documented, the partial labels problem [35]. Its idea of collecting and providing the same kind of partial class information in both learning and prediction stages seems suitable for other frameworks described in the literature, especially for those problems that implement weak supervision models which have emerged as a cheap alternative for collecting labels. Indeed, this approach becomes meaningless if collecting the weakly supervised data in the prediction stage becomes a hard and costly process. That is, this kind of approach is suitable for problems which inherently/effortlessly provide partial class information at prediction time. Once the community becomes conscious of the presence of weak supervision in the prediction stage, other problems/applications which can benefit from it are expected to be identified and novel proposals with

alternative schemes of supervision models are presumed to fill gaps of the state-of-the-art (Table 2.5).

Possible extensions of the proposed taxonomy. Standard supervised classification has a clear definition of the information that a classifier must return: the predicted categorization of the examples. However, some authors have described problems where a different kind of information is expected. This is the case of Ning and Karypis [132] who, in spite of dealing with a problem with an underlying SIML relationship (Table 2.1), learns a classifier which predicts a single label that is shared by a provided group of examples. In other problems, the expected prediction consists of a subset, a ranking or a probability distribution over all the possible class labels according to their correspondence with the provided example [43, 73, 118]. Note that the kind of information that a classifier is expected to return for each example has to be known when the classifier is built and, from this point of view, it could constitute an extra axis of the presented taxonomy.

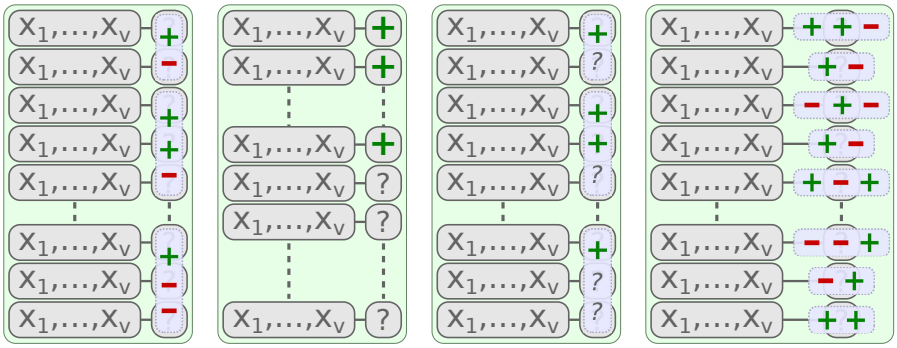
Beyond the four instance-label relationships addressed in this paper (Table 2.1), different frameworks have been proposed in the literature: problems where there exists no absolute membership to any categorization for the examples (label ranking [16, 184] or label distribution [66, 187]), or problems where the image of the target function cannot be represented just by a single class variable (e.g., multi-dimensional framework [6, 133]). These frameworks are known under the general names of structured output and/or multi-target learning [179, 186]. It would be interesting to investigate if the taxonomy axis that represents the instance-label relationship should take a wider definition to cover these non-standard problems.

2.4 Weakly supervised problems in this dissertation

Throughout this dissertation, four different weakly supervised classification frameworks are considered. Two of them are studied in-depth and explored in Part II. Each of the four frameworks are used in Part III for modeling a real application. In this section, all of them are formally described.

2.4.1 Learning from label proportions

Learning from label proportions (LLP) is a weakly supervised classification framework that, sharing the description and objectives of the standard supervised classification (Section 2.1.2), only differs in the supervision model implemented in the learning stage. In this case, the N training examples are individually unlabeled. The dataset (see a graphical description in Figure 2.1(a)) is divided in b *bags* $D = \boldsymbol{B}_1 \cup \boldsymbol{B}_2 \cup \cdots \cup \boldsymbol{B}_b$, where $\boldsymbol{B}_i \cap \boldsymbol{B}_j = \emptyset, \forall i \neq j$. A bag $\boldsymbol{B}_i = \{\boldsymbol{x}^{i1}, \boldsymbol{x}^{i2}, \ldots, \boldsymbol{x}^{iN_i}\}$ groups N_i instances ($\sum_{i=1}^{b} N_i = N$) and provides the only information of supervision in this paradigm: the N_{ic} values



(a) LLP (b) PU (c) PUP (d) CrL

Fig. 2.1. Graphical description of the partially labeled training data that characterizes the different weakly supervised frameworks explored in this dissertation.

or *counts* ($\sum_{c\in\mathcal{C}} N_{ic} = N_i$) which indicate the number of instances in $\boldsymbol{B}_i$ that belong to class label c. Similarly, bag class information can be provided in terms of *proportions* [136], $p_{ic} = N_{ic}/N_i \in [0, 1]$, with $\sum_{c\in\mathcal{C}} p_{ic} = 1$.

2.4.2 Learning from positive and unlabeled examples

Positive-unlabeled (PU) learning is a weakly supervised classification framework only defined for binary classification problems —i.e., the label space comprises two labels ($|\mathcal{C}| = 2$) which are commonly referred to as positive and negative labels ($\mathcal{C} = \{-, +\}$). Regarding the standard supervised classification (Section 2.1.2), this framework only differs in the supervision model in the learning stage. Just a subset of the examples of the training dataset D is labeled; the rest of the instances are unlabeled. Specifically, all the labeled instances have the same positive (main) class label. Thus, $D = D_P \cup D_U$ where $D_P = \{(\boldsymbol{x}^1, +), (\boldsymbol{x}^2, +), \ldots, (\boldsymbol{x}^{N_P}, +)\}$ is the subset of N_P positive instances and $D_U = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^{N_U}\}$ is the subset of N_U unlabeled instances (with $N = N_P + N_U$). Figure 2.1(a) shows a graphical description of the representative PU dataset D.

2.4.3 Learning from positive and unlabeled proportions

The learning from positive and unlabeled proportions (PUP) framework is a novel weakly supervised problem with a supervision model in the learning stage halfway between the label proportions and positive-unlabeled supervision models. As the PU framework, it works over binary classification problems, $\mathcal{C} = \{-, +\}$. Similar to LLP, the examples of the training dataset D (Figure 2.1(c)) are provided grouped in b bags $D = \boldsymbol{B}_1 \cup \boldsymbol{B}_2 \cup \cdots \cup \boldsymbol{B}_b$, where $\boldsymbol{B}_i \cap \boldsymbol{B}_j = \emptyset, \forall i \neq j$. Each bag $\boldsymbol{B}_i = \{\boldsymbol{x}^{i1}, \boldsymbol{x}^{i2}, \ldots, \boldsymbol{x}^{iN_i}\}$ groups N_i examples

and the associated N_{i+} value ($N_{i+} \leq N_i$) indicates the *minimum* number of positive examples in $\boldsymbol{B}_i$. There are other $(N_i - N_{i+})$ examples in $\boldsymbol{B}_i$ which are unlabeled: nothing is known about their class label. As a standard supervised classification problem (Section 2.1.2), the objective is to learn a classifier that infers the label of new unseen examples.

2.4.4 Learning from crowds

The learning from crowds (CrL) framework implements a weak supervision model in training which, for each example, provides the labels annotated by a set of subjective non-expert annotators. The information of supervision of each instance $\boldsymbol{x}^i$ can be codified by a t-tuple $\boldsymbol{l}^i$, where $l_a^i \in \mathcal{C}$ indicates the class label assessed by annotator A_a for $\boldsymbol{x}^i$. Therefore, the dataset (see a graphical description in Figure 2.1(d)) is composed of N examples $D = \{(\boldsymbol{x}^1, \boldsymbol{l}^1), (\boldsymbol{x}^2, \boldsymbol{l}^2), \ldots, (\boldsymbol{x}^N, \boldsymbol{l}^N)\}$. As only the way in which the information of supervision is collected and provided has changed, the objective and other assumptions of standard supervised classification (Section 2.1.2) remain the same.

Learning from crowds in MD domains. In this dissertation, the learning from crowds paradigm is also explored in multi-dimensional domains. In this case, the information of supervision of each instance $\boldsymbol{x}^i$ can be codified by a $(t \times m)$-matrix $\boldsymbol{L}^i$, where L_{ak}^i indicates the label for the class variable C_k assessed by annotator A_a for that instance $\boldsymbol{x}^i$. Therefore, the dataset of a problem of *multi-dimensional learning from crowds* (MDCrL) is composed of N examples as follows, $D = \{(\boldsymbol{x}^1, \boldsymbol{L}^1), (\boldsymbol{x}^2, \boldsymbol{L}^2), \ldots, (\boldsymbol{x}^N, \boldsymbol{L}^N)\}$. As the crowds only affect the way in which the information of supervision is collected and provided, the objective and other assumptions of standard MD classification (Section 2.1.3) remain the same.

3

Bayesian Networks

Bayesian network models [97, 131] are based on the sound and well-studied theory of probabilistic graphical models [21, 105]. Apart from the induction of classification functions, they have been used for probabilistic inference [107], optimization [104] and in the field of bioinformatics [102].

Bayesian network classifiers [5] are a specific case of Bayesian network models that have been regularly used as probabilistic classifiers. Depending on the level of dependencies between the variables codified by the model, they range from very simple classifiers, such as naive Bayes [76], to complete structures that link every pair of variables. The outstanding interpretability of Bayesian network models has motivated our choice as probabilistic classifiers for this thesis. Influences and dependencies among variables can be induced from the explicit probability relationships. The use of probabilistic classifiers allows us to obtain the conditional probability distribution over the class labels given an instance. This kind of information is essential in our methodological developments (Part II) for learning from weakly supervised data, and in our analyses of real applications (Part III) for supplying detailed class information to the final users.

Firstly, different concepts used in the definition of probabilistic graphical models are presented. Then, probabilistic graphical models and Bayesian networks are formally defined. State-of-the-art techniques for learning these models with complete and missing data are exposed. Finally, Bayesian network classifiers and, specifically, the different types of models considered in this dissertation are described.

3.1 Basic concepts

Probabilistic graphical models rely on both probability and graph theory. Some concepts of these areas need to be clarified before proceeding to introduce the probabilistic graphical models and the Bayesian networks.

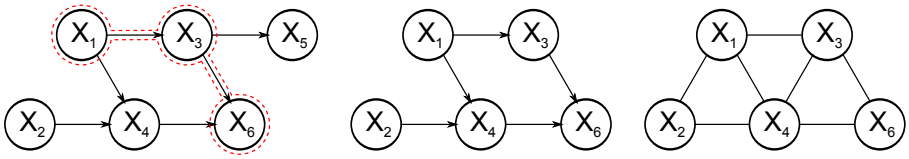
A graph is a pair $\mathcal{G} = (\boldsymbol{X}, \boldsymbol{R})$ where $\boldsymbol{X} = \{X_1, \ldots, X_n\}$ is a non-empty finite set of *nodes* and $\boldsymbol{R}$ is a set of edges. An edge R_{ij}, which represents a link among the pair of nodes (X_i, X_j), is undirected if edge R_{ji} also belongs to $\boldsymbol{R}$. Otherwise ($R_{ji} \notin \boldsymbol{R}$), R_{ij} is a directed edge or *arc*.

A *directed* graph is defined as a graph $\mathcal{G} = (\boldsymbol{X}, \boldsymbol{R})$ where all the edges $R_{ij} \in \boldsymbol{R}$ are arcs. In this context, we say that X_i is a *parent* of X_j —and X_j is a *child* of X_i— if $R_{ij} \in \boldsymbol{R}$ is an arc in $\mathcal{G}$. Moreover, a *path* is a sequence of T arcs in $\boldsymbol{R}$ such that the destination node of any arc is the origin of the following arc (given $R_{ij}^{(t)}$ and $R_{kh}^{(t+1)}$, $j = k$ and $t \in \{1, \ldots, (T-1)\}$). Alternatively, a path can be defined as a sequence of $(T+1)$ nodes such that the t-th node is a parent of the $(t+1)$-th node in $\mathcal{G}$, with $t \in \{1, \ldots, T\}$. Given a directed graph $\mathcal{G}$, X_i is an ancestral node of X_j if there is a path from X_i to X_j in $\mathcal{G}$. Based on this concept, the *ancestral set of nodes* containing $\boldsymbol{Y}$ is the set of nodes formed by $\boldsymbol{Y} \subset \boldsymbol{X}$ and all the ancestral nodes of the elements of $\boldsymbol{Y}$ in $\mathcal{G} = (\boldsymbol{X}, \boldsymbol{R})$.

A directed graph $\mathcal{G}$ is acyclic (DAG) if no circular path —a path that starts and ends in the same node— is found. An *ancestral ordering* of a DAG $\mathcal{G} = (\boldsymbol{X}, \boldsymbol{R})$ is a total ordering of the nodes in $\boldsymbol{X}$ such that, for all $R_{ij} \in \boldsymbol{R}$, the order satisfies that X_i appears before X_j. The *moral graph* associated to a DAG $\mathcal{G}$ is the undirected graph obtained by adding an arc between all the pairs of nodes with a child in common in $\mathcal{G}$ and then transforming all the arcs into undirected edges.

The concept of d-separation is fundamental for understanding the semantic of the probabilistic graphical models based on DAGs. The intuitive definition of this concept presented by Lauritzen [106] relies on the concept of u-separation for undirected graphs.

Definition 1. *Given an undirected graph $\mathcal{G} = (\boldsymbol{X}, \boldsymbol{R})$ and three disjoint subsets of nodes ($\boldsymbol{U}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$) of $\boldsymbol{X}$, $\boldsymbol{U}$ **u-separates** $\boldsymbol{Y}$ and $\boldsymbol{Z}$ in $\mathcal{G}$ if every path in $\mathcal{G}$ between a node belonging to $\boldsymbol{Y}$ and another belonging to $\boldsymbol{Z}$ contains at least one node belonging to $\boldsymbol{U}$.*



(a) Original structure (b) Ancestral set of $\boldsymbol{Q}$ (c) Moral graph of (b)

Fig. 3.1. Graphical example of introduced concepts. X_1 is a *parent* of X_3. In red, a *path* between X_1 and X_3. If $\boldsymbol{U} = \{X_3, X_4\}$, $\boldsymbol{Y} = \{X_1\}$ and $\boldsymbol{Z} = \{X_6\}$ (with $\boldsymbol{Q} = \boldsymbol{U} \cup \boldsymbol{Y} \cup \boldsymbol{Z}$), (b) shows the ancestral set of nodes of $\boldsymbol{Q}$ and (c) its moral graph. $\boldsymbol{U}$ *d-separates* $\boldsymbol{Y}$ and $\boldsymbol{Z}$ as any path between $\boldsymbol{Y}$ and $\boldsymbol{Z}$ goes through $\boldsymbol{U}$.

Definition 2. *Given a DAG $\mathcal{G} = (\boldsymbol{X}, \boldsymbol{R})$ and three disjoint subsets of nodes ($\boldsymbol{U}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$) of $\boldsymbol{X}$, $\boldsymbol{U}$ **d-separates** $\boldsymbol{Y}$ and $\boldsymbol{Z}$ in $\mathcal{G}$ if $\boldsymbol{U}$ u-separates $\boldsymbol{Y}$ and $\boldsymbol{Z}$ in the moral graph of the smallest ancestral set of nodes which contains $\boldsymbol{Y}$, $\boldsymbol{Z}$ and $\boldsymbol{U}$.*

3.2 Probabilistic graphical models

A probabilistic graphical model (PGM) is a mathematical tool that allows us to model a joint probability distribution over a set of random variables $\boldsymbol{X} = \{X_1, \ldots, X_n\}$. It is represented by a pair $(\mathcal{G}, \boldsymbol{\theta})$ composed of a graph structure $\mathcal{G}$ codifying the dependence relationships between the random variables and a set of parameters $\boldsymbol{\theta}$. Although different types of graphs have been used to represent the structure of probabilistic graphical models, we are interested in probabilistic graphical models based on directed acyclic graphs (DAG-based PGMs).

In probability theory, given three disjoint sets of random variables $\boldsymbol{U}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$, $\boldsymbol{Y}$ is conditionally independent of $\boldsymbol{Z}$ given $\boldsymbol{U}$ if and only if $p(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{u}) = p(\boldsymbol{y}|\boldsymbol{u})$, for any possible configuration $\boldsymbol{u}$, $\boldsymbol{y}$ and $\boldsymbol{z}$. In DAG-based PGMs, conditional independence can be expressed in terms of the previously introduced d-separation criterion:

Definition 3. *Given a PGM $(\mathcal{G}, \boldsymbol{\theta})$ with a DAG $\mathcal{G} = (\boldsymbol{X}, \boldsymbol{R})$ and three disjoint subsets of variables ($\boldsymbol{U}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$) of $\boldsymbol{X}$, $\boldsymbol{Y}$ is conditionally independent of $\boldsymbol{Z}$ given $\boldsymbol{U}$ if $\boldsymbol{U}$ d-separates $\boldsymbol{Y}$ and $\boldsymbol{Z}$ in $\mathcal{G}$.*

Assuming that the set of variables $\boldsymbol{X}$ is ordered according to some ancestral ordering of the DAG $\mathcal{G}$, the set of parents of a variable X_j ($\mathbf{PA}_j$) d-separates X_j from any previous variable in the ancestral ordering, X_i ($i < j$). That is, X_j is conditionally independent of any X_i ($i < j$) given its parents, $\mathbf{PA}_j$. By means of this property, the joint probability distribution of $\boldsymbol{X}$, which is usually expressed given the chain rule as:

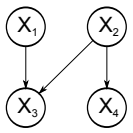
$$p(\boldsymbol{x}) = p(x_1, \ldots, x_n) = \prod_{j=1}^{n} p(x_j|x_1, \ldots, x_{j-1}) \tag{3.1}$$

when codified by a DAG-based PGM, can be factorized as:

$$p(\boldsymbol{x}) = \prod_{j=1}^{n} p(x_j|\boldsymbol{pa}_j) \tag{3.2}$$

The second component of a PGM, besides the graph structure, is a set of parameters $\boldsymbol{\theta}$. Although, for the sake of simplicity, Equation 3.2 will be used throughout this dissertation, its complete expression should be:

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \prod_{j=1}^{n} p(x_j|\boldsymbol{pa}_j, \boldsymbol{\theta})$$



X_j	d_j	$\mathbf{PA}_j$	D_j
X_1	3	$\emptyset$	-
X_2	2	$\emptyset$	-
X_3	2	(X_1, X_2)	6
X_4	2	X_2	2

X_j	Model parameters
X_1	$\theta_{1\emptyset 1} = p(x_{11}), \theta_{1\emptyset 2} = p(x_{12}), \theta_{1\emptyset 3} = p(x_{13})$
X_2	$\theta_{2\emptyset 1} = p(x_{21}), \theta_{2\emptyset 2} = p(x_{22})$
X_3	$\theta_{311} = p(x_{31}\|x_{11}, x_{21}), \theta_{321} = p(x_{31}\|x_{11}, x_{22}), \theta_{331} = p(x_{31}\|x_{12}, x_{21}),$
	$\theta_{341} = p(x_{31}\|x_{12}, x_{22}), \theta_{351} = p(x_{31}\|x_{13}, x_{21}), \theta_{361} = p(x_{31}\|x_{13}, x_{22}),$
	$\theta_{312} = p(x_{32}\|x_{11}, x_{21}), \theta_{322} = p(x_{32}\|x_{11}, x_{22}), \theta_{332} = p(x_{32}\|x_{12}, x_{21}),$
	$\theta_{342} = p(x_{32}\|x_{12}, x_{22}), \theta_{352} = p(x_{32}\|x_{13}, x_{21}), \theta_{362} = p(x_{32}\|x_{13}, x_{22})$
X_4	$\theta_{411} = p(x_{41}\|x_{21}), \theta_{421} = p(x_{41}\|x_{22}), \theta_{412} = p(x_{42}\|x_{21}), \theta_{422} = p(x_{42}\|x_{22})$

Fig. 3.2. Example of a Bayesian network model: structure, table of parents and number of possible values, and model parameters. It models a joint probability distribution which factorizes as $p(x_1, x_2, x_3, x_4) = p(x_1) \cdot p(x_2) \cdot p(x_3|x_1, x_2) \cdot p(x_4|x_2)$.

3.3 Bayesian network models

Bayesian network models are DAG-based PGMs where all the random variables are discrete. Thus, a Bayesian network model consists of a DAG $\mathcal{G}$ and a set of parameters $\boldsymbol{\theta}$. The consideration of DAGs allows the joint probability distribution $p(\boldsymbol{x})$ to factorize according to Eq. 3.2, which usually involves a considerably reduced set of parameters, $\boldsymbol{\theta}$, with respect to that of the general factorization (Eq. 3.1).

The set of parameters $\boldsymbol{\theta}$ defines all the probability distributions $p(x_{jl}|\boldsymbol{pa}_{jk})$. Each parameter $\theta_{jkl} = p(x_{jl}|\boldsymbol{pa}_{jk})$ denotes the probability that variable X_j takes its l-th possible value given that the parents $\mathbf{PA}_j$ of X_j take their k-th value. Each variable X_j has its own set of d_j possible values. Accordingly, the set of possible values of a random vector is the product of the set of possible values of each random variable in it. In this way, the parents $\mathbf{PA}_j$ of variable X_j take $D_j = \prod_{i/X_i \in \mathbf{PA}_j} d_i$ different values.

3.3.1 Learning Bayesian network models from data

Both the graph of conditional (in)dependencies and the model parameters of a Bayesian network model can be estimated from a set of examples of a domain of interest or can be provided by means of domain-expert knowledge. The latter option, although it may be affordable in the case of simple models, becomes impracticable as the size of the model (nodes, arcs and corresponding parameters) grows: the induction of the structure gets more and more difficult and the probabilities to consider get more complicated to understand and estimate by means of expert knowledge. The former option, inferring a BN from a set of previous examples of the problem, is usually a more feasible and, nowadays, more popular approach. Many different techniques have been proposed

in the literature to learn Bayesian networks. Several thorough surveys, where different proposals are collected and discussed, have been published [79, 131].

When only a dataset is provided, a method that learns Bayesian network models usually implements two stages: the structural learning, where the structure $\mathcal{G}$ of conditional (in)dependencies is inferred, and the parametric learning, where all the model parameters needed to codify the joint probability distribution $\boldsymbol{\theta}$ are estimated.

Parametric learning estimates a particular set of parameters $\boldsymbol{\theta}$ from the provided data based on some criterion. Among the different techniques proposed in the literature for parametric learning, the best known approaches are the maximum likelihood (ML) and the maximum a posteriori (MAP) estimations.

ML, the estimation technique considered throughout this dissertation, selects the set of parameters $\hat{\boldsymbol{\theta}}$ (for a fixed graph structure $\mathcal{G}$) that maximizes the probability of observing the dataset D:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(D|\mathcal{G}, \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{N} p(\boldsymbol{x}^i|\mathcal{G}, \boldsymbol{\theta})$$

The parameters that maximize the likelihood function $p(D|\mathcal{G}, \boldsymbol{\theta})$ can be obtained by means of frequency counts [79]:

$$\hat{\theta}_{jkl} = \frac{N_{jkl}}{N_{jk}}$$

In practice, a smoothing technique is usually implemented to prevent assigning zero or one probabilities. In this dissertation, the Laplace estimator, a classic additive smoothing technique, has been used. Therefore, the parameters $\hat{\boldsymbol{\theta}}$ are calculated as:

$$\hat{\theta}_{jkl} = \frac{N_{jkl} + 1}{N_{jk} + d_j}$$

When using a Bayesian approach for learning, an a priori distribution over all the possible sets of parameters is used —particularly, the Dirichlet distribution— and the estimation of the parameters is naturally carried out by the MAP estimation: the set of parameters $\hat{\boldsymbol{\theta}}$ that has the highest a posteriori probability given the dataset D and a graph structure $\mathcal{G}$:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathcal{G}, D) \propto \arg\max_{\boldsymbol{\theta}} p(D|\mathcal{G}, \boldsymbol{\theta}) \cdot p(\mathcal{G}, \boldsymbol{\theta})$$

The set of parameters $\hat{\boldsymbol{\theta}}$ that maximizes this expression can be calculated, given the set of hyper-parameters $\boldsymbol{\alpha} = (\alpha_{jk1}, \ldots, \alpha_{jkd_j})$, which are the parameters of the a priori Dirichlet distribution that represents the prior knowledge about the $\hat{\boldsymbol{\theta}}$, by means of the following formula:

$$\hat{\theta}_{jkl} = \frac{N_{jkl} + \alpha_{jkl}}{N_{jk} + \alpha_{jk}}$$

where N_{jkl} is the number of examples in D where the variable X_j takes its l-th value and $\mathbf{PA}_j$ takes its k-th configuration, $N_{jk} = \sum_{l=1}^{d_j} N_{jkl}$ and $\alpha_{jk} = \sum_{l=1}^{d_j} \alpha_{jkl}$.

Note that the smoothed Laplace estimator can, in this way, be seen as a MAP (or Bayesian) estimator with all the α_{jkl} hyper-parameters equal to 1.

Structural learning infers from a set of examples D the graph of conditional (in)dependencies of a Bayesian network model. Proposed techniques can be roughly divided in two groups: algorithms that detect conditional (in)dependencies and algorithms with a score+search approach.

On the one hand, (in)dependence detection based algorithms try to represent by means of a DAG a large percentage (all of them, if possible) of the detected conditional (in)dependence relationships. Some methods are provided a dataset D where independence tests are carried out to establish conditional independencies. Others use a provided joint probability distribution where the conditional independencies can be checked or, directly, a list of candidate conditional independencies. In this first group, the most popular method is probably the PC algorithm [172].

On the other hand, score+search algorithms tackle the problem of structure induction as an optimization problem. The search of the best structure is an NP-hard problem [28]. Therefore, heuristic search methods are used to obtain fitted structures in a reasonable time. Greedy search [32], genetic algorithms [103], estimation of distribution algorithms [8], Markov Chain Monte Carlo [129] and nature-inspired optimization [38] are a few examples of search techniques which have been already used to infer the graph structure $\mathcal{G}$.

A scoring function for graph structures usually guides the search process. Different types of scoring function have been proposed. One of them is the marginal likelihood (averaged probability of a dataset D given a structure $\mathcal{G}$ over all the possible values of the parameters). This score, a.k.a. K2 score, can be calculated in a closed form under certain assumptions [32]. The log-likelihood (for a given structure $\mathcal{G}$, the maximum likelihood estimation of the parameters $\boldsymbol{\theta}$ can be obtained and, subsequently, the log-likelihood of D given $\mathcal{G}$ and $\boldsymbol{\theta}$) has also been considered. However, due to the fact that it increases as the $\mathcal{G}$ structure gets more complex, a penalization term is usually added. This penalization term usually consists of the product of the network dimension (e.g., number of parameters) and a penalization function. In this context, the Bayesian information criterion (BIC) is the most popular score [154]. Scoring functions based on information theory have also been proposed. The minimum description length criterion (MDL) [145] states that the best structure to describe a dataset is that which minimizes the encoding length of the data and the model. The resulting equation is similar to that of the BIC metric, although it is obtained in a completely different way. Other

scores, such as entropy [84] or mutual information [31], have also been used to guide the structural search of Bayesian network models.

3.3.2 Learning Bayesian networks in the presence of missing data

As previously explained, in the case of availability of a complete dataset, the network structure of a Bayesian network model can be inferred using heuristic methods and all the model parameters can be calculated with maximum likelihood estimates by means of frequency counts. However, the presence of missing data increases the complexity of the learning process.

In this dissertation, we are especially interested in methodologies that learn both the parameters and structure of a Bayesian network from weakly supervised data —i.e., from data that present missing values in the class variable. In the general case, different techniques have been proposed in the literature to learn BN models from incomplete data, most of them based on local structural search. For instance, Ramoni and Sebastiani [139] proposed a hill climbing method to build a DAG and applied Bound and Collapse [140] to learn the parameters (the sufficient statistics are calculated by means of a convex optimization process that uses upper and lower bounds of the statistics as initialization). The proposal of Riggelsen [143] also uses a hill-climbing strategy, and its main novelty is an imputation procedure based on the Markov blanket of the variable with missing values to predict its value. In a different approach, the iterative method proposed by Singh [166] creates, at each step, t datasets by predicting the value of the missing data with the BN obtained in the previous iteration. For each complete dataset, a BN is learnt, and then the t BNs are merged. Other methods use different techniques to learn BN models from incomplete data: evolutionary methods [129, 194], boostrap [50], MCMC [129, 144], etc. There also exists extensive literature about learning BNs for clustering considering the existence of a hidden variable which would reflect the cluster/class membership (i.e., the class labels of the training examples are missing) [135].

However, many of the aforementioned methods for learning in the presence of missing values [50, 166, 194] make use of the Structural Expectation-Maximization strategy [57] (or of the basic Expectation-Maximization strategy [45]). In this dissertation, this widely used and theoretically-founded strategy has been implemented to learn Bayesian network models from weakly supervised data.

The Expectation-Maximization (EM) strategy, proposed by Dempster et al. [45], is an iterative procedure that can be used to obtain the maximum likelihood parameters in the presence of missing data. It can also be used to obtain the *maximum a posteriori* estimate or to fill up missing data. Each iteration consists of two steps, expectation (E) and maximization (M). The E-step estimates the missing data as the conditional expectation of the likelihood given the current fit for the model parameters. In the M-step, the model

parameters are re-estimated such that the likelihood is maximized given the data completed in the E-step. Under fairly general conditions, the iterative increment of the likelihood has been proved to converge to a stationary value. Most of the times, a stationary value means local maximum since convergence to a global maximum is not guaranteed, although in some rare cases it can be trapped in a non-optimal point such as a saddle point [121].

Following the description of the EM strategy given by McLachlan and Krishnan [121] for computing the maximum-likelihood estimate, if $\boldsymbol{y}$ is considered the observed data, $\boldsymbol{x}$ the completed data and $\boldsymbol{\theta}$ the real vector of model parameters (unknown), the two steps of the $(t+1)$-th iteration are:

E-step: Using the current estimate $\hat{\boldsymbol{\theta}}^{(t)}$ of the parameters, calculate the conditional expectation of the complete-data log likelihood, $\log \mathcal{L}(\boldsymbol{\theta}|\boldsymbol{x})$, given the observed data vector $\boldsymbol{y}$:

$$Q(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}^{(t)}) = E_{\hat{\boldsymbol{\theta}}^{(t)}}\{\log \mathcal{L}(\boldsymbol{\theta}|\boldsymbol{x}) \mid \boldsymbol{y}\}$$

M-step: Choose $\hat{\boldsymbol{\theta}}^{(t+1)}$ such that, for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$:

$$Q(\hat{\boldsymbol{\theta}}^{(t+1)}; \hat{\boldsymbol{\theta}}^{(t)}) \geq Q(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}^{(t)})$$

The Structural Expectation-Maximization (SEM) strategy, proposed by Friedman [57], adds a structural learning stage to the classical parametric EM strategy for inferring the graph structure from data when it is not provided. It incorporates an outer loop to the parametric-convergence loop of the classical strategy, and iteratively improves an initially-proposed structure. In this dissertation, this structural improvement uses a one-step local search algorithm, searching for the structure that maximizes the complete-data minimal description length (MDL) score. If the local MDL-maximal structure is the current one, the algorithm stops. Otherwise, it tries to find the best set of model parameters for the new maximal structure as the first stage of a new iteration of the structural outer loop.

The neighborhood for the local search is composed of all the structures that can be obtained by removing, adding or reversing an arc. If a Bayesian network model of constrained structure is used, a valid operator consists of removing an arc from the original structure and adding another different arc which fulfills the restrictions of the considered type of model. That is, it should only consider those changes that produce a network structure of the same type as the original structure.

3.4 Bayesian network classifiers

In this dissertation, we use Bayesian network models as probabilistic classifiers. Although the number of different types of Bayesian network models that

have been used in the literature for classification purposes is noteworthy [5], we are only interested in a type of classifiers whose structure is specially designed for the classification task.

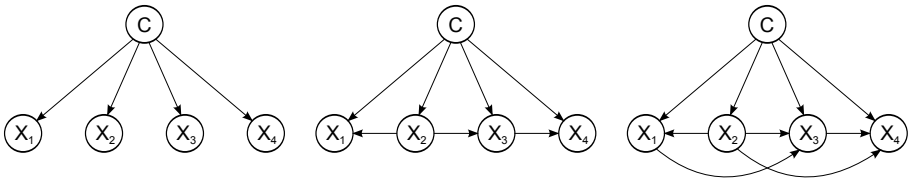
As presented in Chapter 2, a supervised classification problem is described by a set of n predictive variables $\boldsymbol{X} = (X_1, \ldots, X_n)$ and a special variable, the class variable C. A Bayesian network classifier is also represented by a pair $(\mathcal{G}, \boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is the set of parameters of the conditional probability functions of each variable given its parents, and $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{R})$ codifies the DAG among the $(n+1)$ random variables, $\boldsymbol{V} = (X_1, \ldots, X_n, C)$. A classification task aims to infer the value c of the class variable given a case of the problem $\boldsymbol{x}$ and, to this end, it could be assumed that the class variable depends on the predictive variables (Eq. 2.1). However, the number of parameters of such a Bayesian network would be exponential to the number of predictive variables, n, requiring a costly learning process. If the Bayes theorem and the factorization of Eq. 3.2 are applied:

$$p(c|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|c) \cdot p(c)}{p(\boldsymbol{x})} = \frac{1}{p(\boldsymbol{x})} \cdot p(c) \cdot p(\boldsymbol{x}|c) \propto p(c) \cdot \prod_{j=1}^{n} p(x_j|\boldsymbol{pa}_j, c)$$

this expression where all the predictive variables are conditioned to the class variable is obtained. In a classification task, the example $\boldsymbol{x}$ is given and, therefore, $p(\boldsymbol{x})$ is constant. As the objective is to find the class label c that maximizes the conditional probability given the example, $\operatorname{argmax}_c p(c|\boldsymbol{x})$, the term $p(\boldsymbol{x})$ can be removed: The class label c that maximizes the resulting expression, which is directly proportional ($1/p(\boldsymbol{x}) > 0$) to $p(c|\boldsymbol{x})$, is the same as that which maximizes the conditional probability. Based on this reasoning, Bayesian network classifiers where the class variable is parent of all the predictive variables (and not the other way around) have been shown to be appropriate for classification. The general classification rule can be defined as:

$$\hat{c} = \operatorname*{argmax}_c p(c) \cdot \prod_{j=1}^{n} p(x_j|\boldsymbol{pa}_j, c)$$

where $\boldsymbol{pa}_j$ is the value of the set of predictive variables which are parents of X_j in the DAG, apart from the class variable. Depending on the restrictions imposed on the sets of parents $\mathbf{PA}_j$, models of different complexity are obtained. Specifically, naive Bayes (NB), tree augmented naive Bayes (TAN) and K-dependence Bayesian network (KDB) classifiers are considered in this dissertation (see Figure 3.3). Based on the assumption of conditional independence between the predictive variables given the class variable, the naive Bayes presents the simplest network structure. In spite of its simplicity, it has achieved very competitive results in many domains [76]. TAN and KDB are the next step forward in terms of network structure complexity: they allow models to capture some conditional dependencies between predictive variables.



(a) Naive Bayes (b) TAN (c) KDB ($K = 2$)

Fig. 3.3. Examples of the structure of naive Bayes, tree-augmented naive Bayes and K-dependence Bayesian network classifiers.

3.4.1 Naive Bayes

The naive Bayes (NB) classifier [76] (a.k.a. simple Bayesian classifier, independent Bayes or idiot Bayes) assumes conditional independence between the predictive variables given the class variable. This assumption is reflected in its representative simple and fixed graph structure (Figure 3.3(a)) and allows the classifier to be defined as:

$$\hat{c} = \operatorname*{argmax}_c p(c) \cdot \prod_{j=1}^{n} p(x_j|c)$$

Learning a naive Bayes classifier can be performed efficiently. The parametric learning only involves the learning of the probabilities $p(c)$ and the conditional probabilities $p(x_j|c)$. Moreover, structural learning is not necessary since naive Bayes classifiers have a fixed structure. In this way, given a complete dataset, the time complexity of the learning process of naive Bayes classifiers is $O(N \cdot n)$.

3.4.2 Tree augmented naive Bayes

A tree augmented naive Bayes (TAN) is a Bayesian network classifier based on naive Bayes that considers a weaker assumption of conditional independence than that assumed in naive Bayes models. The structure keeps the class variable as a parent of all the predictive variables, although additional edges between predictive variables are allowed. That is, dependencies between predictive variables can be captured. These dependencies between predictive variables are incorporated encoded as a tree (see Figure 3.3(b)). In this way, each predictive variable has, at most, two parents in the network structure: the class variable and one (if any) predictive variable. In this case, the classification rule is defined as:

$$\hat{c} = \operatorname*{argmax}_c p(c) \cdot \prod_{j=1}^{n} p(x_j|x_i, c)$$

where X_i is the predictive variable parent of X_j in the tree structure, if any.

Friedman et al. [58] proposed an algorithm to learn TAN structures from complete datasets in polynomial time, $O(N \cdot n^2)$. It is an adaptation of the Chow-Liu algorithm [31] that uses the *conditional* mutual information between predictive variables given the class rather than the mutual information. The algorithm creates a complete undirected graph with the edges weighted according to the conditional mutual information between the linked variables given the class. The Kruskal algorithm [98] is then used to obtain the maximum weighted spanning tree. Next, using a randomly chosen predictive variable as the root, the edges of the obtained (undirected) tree are directed in accordance. Finally, the class variable is included in the structure and arcs from it to all the predictive variables are included in the DAG. Accordingly, as new arcs are included in the DAG, the number of parameters which have to be estimated during the parametric learning step increases: the enlarged set of conditional probabilities $p(x_j|x_i, c)$, apart from $p(c)$.

3.4.3 *K*-dependence Bayesian classifiers

Regarding TAN, a K-dependence Bayesian classifier (KDB) is a step forward in the level of dependencies considered between predictive variables. The KDB structure is also based on that of the naive Bayes classifier, but allowing each predictive variable to have, at most, K predictive variables as parents (besides the class variable). Following this definition, the TAN classifier can be considered a 1-dependence Bayesian classifier (1DB), and the naive Bayes classifier, a 0DB; therefore, KDB can be seen as a generalization of both classifiers. The classifier can be defined as:

$$\hat{c} = \operatorname*{argmax}_c p(c) \cdot \prod_{j=1}^{n} p(x_j|\boldsymbol{pa}_j, c)$$

where $\mathbf{PA}_j$ is, at most, a set of K predictive variable parents of X_j ($0 \leq |\mathbf{PA}_j| \leq K$).

Due to the fact that learning unrestricted Bayesian network models is NP-hard, Sahami [152] proposed a greedy algorithm that learns KDB structures from complete datasets in time $O(N \cdot n^2 \cdot |\mathcal{C}| \cdot d_{max}^2)$, where d_{max} is the maximum number of values that a predictive variable may take. The algorithm first calculates the mutual information of each predictive variable with respect to the class and the conditional mutual information of all the pairs of predictive variables given the class. Iteratively, it selects the variable with the largest mutual information with respect to the class which has not yet been chosen. It selects the K previously chosen variables with largest conditional mutual information with the current variable given the class. If the set of already considered variables is smaller than K, it uses all the variables of this set. Then, $K + 1$ arcs (at most) are included in the DAG pointing to the current variable: one from each chosen variable and another from the class variable.

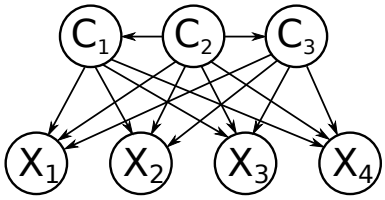


Fig. 3.4. MBC structure considered in Chapter 5.

This process is repeated until all the variables are connected in the DAG. Regarding the parametric learning, the set of parameters to be estimated for this kind of classifiers includes $p(c)$ and the enlarged set of conditional probabilities $p(x_j|\boldsymbol{pa}_j, c)$.

3.4.4 Multi-dimensional Bayesian network classifiers

Multi-dimensional Bayesian network classifiers [6] (MBCs) generalize the standard Bayesian network classifiers to classification problems with more than one class variable (see definition in Section 2.1.3). As any Bayesian network, it is represented by a pair $(\mathcal{G}, \boldsymbol{\theta})$. However, the graph structure of a MBC, $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{R})$, codifies the arcs $\boldsymbol{R}$ (conditional dependencies) between nodes $\boldsymbol{V} = (X_1, \ldots, X_n, C_1, \ldots, C_m)$ (random variables) which, in this case, comprise n predictive variables $\boldsymbol{X} = (X_1, \ldots, X_n)$ and m class variables $\boldsymbol{C} = (C_1, \ldots, C_m)$. The set of arcs $\boldsymbol{R}$ is usually partitioned into three subsets: $\boldsymbol{R}^C$, which collects the arcs exclusively between the class variables; $\boldsymbol{R}^X$, which collects the arcs between the predictive variables; and $\boldsymbol{R}^{XC}$, which collects the arcs from the class variables to the predictive variables. The MBC definition incorporates the restriction that arcs from the predictive to the class variables are not allowed in $\boldsymbol{R}^{XC}$.

Different types of MBCs have been proposed in the literature [6, 39, 148, 182]: multi-dimensional naive Bayes (MDnB), tree-augmented (MDTAN), polytree-augmented (MDPoly) or J/K dependences Bayesian classifier (MDJ/K). In this dissertation, we have used a basic MBC structure (Fig. 3.4): a tree between the class variables and each predictive variable having all the class variables as parents. The classification rule of this kind of MBC is defined as:

$$\hat{\boldsymbol{c}} = \operatorname*{argmax}_{\boldsymbol{c}} p(\boldsymbol{c}) \cdot \prod_{j=1}^{n} p(x_j|\boldsymbol{c})$$

where $\boldsymbol{c}$ is a joint class label assignment to all the class variables and $p(\boldsymbol{c}) = p(c_r) \cdot \prod_{k \neq r} p(c_k|c_l)$ is decomposed according to the tree structure (being C_r the root class variable and C_l the class variable parent of C_k).

The MBCs selected for this dissertation provide a fixed structure which does not need to be learnt —it consists of m fixed NB structures and a tree

among class variables which, in practice, has been fixed. As the number of parameters resulting from the referred structure strongly depends on the number of class variables (m), the corresponding parametric learning process would become unfeasible in a MD problem with many classes. Without this drawback, several methods to learn from data completely labeled MBC structures which involve a restrained number of parameters have been proposed in the related literature [6, 148]. However, this kind of MBCs (Fig. 3.4) has only been considered in a controlled experimental setting (Section 5.5) where we have used a reduced number of class variables. Avoiding the structural learning step is, for the referred study, an interesting property which facilitates the interpretability of the obtained results. Thus, the learning process is reduced to parametric learning of a set of parameters which is composed of $p(c_r)$, $p(c_k|c_l)$ and $p(x_j|\boldsymbol{c})$.

Part II

Contributions to learning from weakly supervised data

4

Learning from label proportions

In the first part of this thesis we have described the field of weakly supervised classification. A problem of this field, where the information of supervision cannot be completely collected, is the learning from label proportions framework. In this problem, formally introduced in Section 2.4.1, the provided training dataset is composed of unlabeled instances and is divided into disjoint groups. General class information is given within the groups: the proportion of instances of the group that belong to each class. Among the many different real applications described in the literature for this framework, the assisted reproductive technologies —posteriorly in this dissertation we propose a solution for dealing with this real application— and the election votes problems —personal votes are collected in polling stations, all the votes are aggregated in each polling station and only the global figures are published— stand out. To deal with the exposed problem, we propose a method based on the Structural EM strategy (Section 3.3.2) that learns Bayesian network classifiers (Section 3.4). Four versions of our proposal are evaluated on synthetic data and compared with state-of-the-art approaches on real datasets from public repositories. The results obtained show a competitive behavior for the proposed algorithm.

4.1 Introduction

In classical supervised classification, the objective is to build a predictive model from a dataset of labeled instances such that, given a new unlabeled example, the model will assign it to one of the already-known class labels. In the most common situation, each instance in the dataset consists of the description of the example and its associated class label [77]. Moreover, other problems where obtaining labeled examples is difficult (semi-supervision) have received considerable attention in the literature [23]. However, in recent years new problems in which the available class-membership information of the provided examples (a.k.a. information of supervision) does not consist of the typ-

ical class value for each (labeled) instance have been proposed. Thus, standard learning strategies, which have been developed for learning from supervised or semi-supervised domains, can not be straightforwardly applied. Therefore, new specific strategies (or adaptations of classical strategies) that learn from the new kinds of non-fully labeled datasets are necessary. The specific techniques, in order to be efficient, are expected to extract as much knowledge as possible from the available information of supervision.

In this dissertation, we deal with problems in which the relation between an instance and its associated class label is lost. This may be due to the black-box nature of the problem, privacy preserving, non-monitoring process, etc. In this framework, the unlabeled instances are grouped and only global class information is available for the instances of each group: the label proportions. A particular application of this general framework is the problem of embryo selection in *assisted reproductive technologies* (ART) [126]. In the most critical step of an ART cycle, gynecologists have to select the embryos to be transferred to the uterus of the woman among a set of embryos that have been cultured for several days (in Spain, by law, 3 embryos at most can be transferred in an ART cycle). During the culture period, some relevant features are observed and collected for each individual embryo. Then, after the transference, doctors can observe, using preclinical imaging techniques, the number of those transferred embryos that are implanted (and induce a pregnancy), but it is not possible to know *which* individual embryo is implanted. Thus, in a dataset for this problem, each instance represents a transferred embryo and each group includes the embryos transferred in the ART cycle that it represents. The class label, which should indicate whether or not the specific transferred embryo became implanted or not, is individually unknown for the instances of the dataset. However, some kind of information of supervision is available for each group of instances: the number of positive instances (implanted embryos) in the corresponding ART cycle.

Another real case that involves the same kind of data is that of election votes, where some parties stand for institutions and, in each polling station, each party gets a known number of votes. The global election results are known, but which party each citizen voted for is unknown. By knowing the population census and some socioeconomic data of the voters, it could be possible to estimate the probability of a citizen voting for a party [99]. More real instances of the problem include the analysis of single particle mass spectrometry data [128], e-commerce [136], spam and image filtering [136], fraud detection [150], etc.

The presented problem relates to the multiple-instance learning problem since, in both cases, the training dataset is divided into disjoint groups of instances. Multiple-instance learning (MIL) [47] is a supervised classification problem where an example is represented by a group of instances and there is a global label per group (or example). In MIL, the objective is to learn from and classify groups of instances. However, the problem we are dealing with

considers class label assignments to the individual instances, despite being unknown in training time.

There exist in the literature several methods to deal with the learning from label proportions (LLP) problem. The first time that a method was proposed to learn from this kind of data was in [99], where Kück and Freitas present a MCMC strategy. But it was Musicant et al. [128] who gave the first definition of the LLP problem, which they called aggregated outputs. They use the counts of labels (instead of proportions) as general class information per group. In their paper, basic adaptations of KNN, ANN, SVM and Decision Trees are proposed.

Simultaneously, Quadrianto et al. [136] gave an alternative definition based on label proportions. Their method, called MeanMap, models the conditional class probability using conditional exponential models. Although their method is primarily defined to deal with problems where the label proportions of the test set are known, it incorporates a functionality that estimates these proportions when they are not given. Following a similar definition of the problem but without requiring label proportions of the test set, Rueping [150] proposes an algorithm to learn SVMs for this problem.

Other authors implement a different strategy to learn from LLP datasets. Their contributions consist of a procedure that firstly reduces the uncertainty of the data provided, estimating the class label of each unlabeled instance. This generates a complete dataset which can be used to train a classifier using any classical method for supervised data. In this way, Chen et al. [27] proposed a method based on kernel K-means for solving this problem of label assignment. Later, Stolpe and Morik [174] presented a similar method which solves this problem using an evolutionary strategy that looks for the predictive variable weights that lead to the clustering (K-means) that best fits the label proportions.

The main contributions of this chapter are as follows:

- The development of an algorithm based on the Structural Expectation-Maximization (SEM) strategy [57] to learn Bayesian network classifiers for the LLP problem.
- The development of several variants of the method, two of which have been specifically designed to deal with (complex) LLP scenarios with high degree of uncertainty in the class label of the individual instances.
- The use of joint label assignments, i.e., only the label assignments which fulfill the label proportions of the groups are considered.
- The proposal of a new framework for testing LLP methods, which covers the whole spectrum of LLP scenarios in terms of complexity for a given dataset.

The Bayesian network classifiers that our SEM method learns show a good performance behavior through different LLP scenarios of increasing class uncertainty. Moreover, it obtains competitive results with respect to state-of-the-art methods.

The rest of the chapter is organized as follows. In the next section, the class uncertainty in the LLP problem is explored. Then, four versions of a new algorithm based on the Structural EM strategy which learns Bayesian network classifiers in the LLP framework are proposed. Later, the experiments are presented in four subsections: an experimental demonstration of the usefulness of the extra class information provided in the LLP problems using the semi-supervised learning approach as a baseline-performance reference, an evaluation of the approximate reasoning of our method by means of local probabilistic label assignments, an analysis on synthetic data that evaluates the efficacy of our proposals in different experimental conditions, and a comparison with state-of-the-art approaches. Finally, some conclusions and future work are presented.

4.2 The problem of learning from label proportions

In a problem of learning from label proportions (LLP), the examples are grouped in *bags* —or disjoint sets of examples— where each instance has been separated from its label. For some reason, the individual pairing relation (instance, label) is lost and, therefore, each bag provides two separate equal-sized unpaired groups: the group of instances and the group of labels. The group of labels can be presented as the proportion of instances that belong to each class label. Note that these label proportions do not indicate a *belief* (probability) in the number of instances that belong to each class but the real *exact* number. Formally defined in Section 2.4.1, the weak supervision model in the learning stage, which is responsible of the presence of groups of instances in its characteristic training dataset, is the only difference in this problem regarding the standard supervised classification.

4.2.1 Uncertainty associated to the label proportions

The difficulty of pairing each instance with its class label could be thought as a basic definition of uncertainty associated to the label proportions. Thus, assuming that each bag has its own label proportions and, therefore, involves its particular uncertainty, it is possible to distinguish between two kinds of bags. On the one hand, if all the instances in bag $\boldsymbol{B}_i$ belong to the same class ($\exists c \in \mathcal{C} : N_{ic} = N_i$), there is class certainty and the individual instances may be considered labeled. This kind of bag is called *full bag*. Following the example of the embryo selection in ART, a bag is full if the corresponding ART cycle finished with either all the embryos implanted or no embryo implanted. However, bags usually have instances that belong to different classes ($\forall c \in \mathcal{C} : N_{ic} < N_i$). In this case, the class label of an individual instance is unknown (class uncertainty), although the instances in $\boldsymbol{B}_i$ are known to belong to one of the class labels specified in the label proportions of $\boldsymbol{B}_i$. This kind of bags are known as *non-full bags*. Following the previous example, this case is observed

when some of the transferred embryos (but not all of them) became implanted. It is important to note that the uncertainty in a non-full bag $\boldsymbol{B}_i$ is higher when the labels are well-distributed (balanced), i.e., the difference among counts N_{ic}, for all $c \in \mathcal{C}$, is minimized.

The combined uncertainty of the bags that configure a LLP dataset of interest determines the complexity of learning in that scenario. In this way, the least complex LLP scenario has a dataset exclusively composed of full bags, which is the configuration of minimal uncertainty. As mentioned before, the instances of a full bag may be considered labeled instances, so the least complex LLP scenario is as complex as the same problem in a classical supervised scenario. On the contrary, the dataset of the most complex LLP scenario is only composed of non-full bags, all of them with the label proportions matching the proportions of the class labels in the whole dataset. Note that this is not in contradiction with the previous paragraph, where the bag with balanced label proportions is considered the most uncertain bag. Consider a LLP problem where the proportions of the class labels in the whole dataset are not balanced. The presence of bags with balanced label proportions (the most uncertain bag) in the scenario necessarily implies the presence of other bags with the label proportions even more unbalanced than the label proportions of the whole dataset (a less uncertain bag). Therefore, as the low uncertainty unbalanced bags compensate the high uncertainty of the balanced bags, this bag configuration may not represent the most complex scenario. In addition to the distribution of labels, the uncertainty is also determined by the bag size (N_i), as the proportion of instances with uncertain labels in the whole dataset indirectly depends on this value.

4.3 Learning Bayesian network classifiers for the LLP problem

Two basic solutions to the LLP problem would be to set the expected counts for each instance of the dataset with the label proportions of the whole dataset or with the label proportions of the corresponding bag. However, both are suboptimal solutions, quickly overcome by applying some of the classical techniques that deal with datasets that have missing data, such as the SEM strategy. As explained in Section 3.3.2, the Structural EM strategy provides a suitable framework to learn, in the presence of missing data [57], the probabilistic classifiers used in this dissertation, the Bayesian network classifiers (Section 3.4). It extends the classical two-steps strategy (Expectation and Maximization steps for parametric learning) to alternate the phases of network structure estimation and parametric learning. Although this general strategy can be used to learn in the presence of missing data, it is not designed to deal with the additional information provided by the label proportions, which could be used to improve the learning process. We propose a method based on the Structural EM strategy that exploits the extra information of the bag

Algorithm 1 Pseudo-code of the Structural EM strategy.

```
1: procedure StructuralEM(D, maxItP, maxItS, ε)
2:     D̂ ← initializeData(D)
3:     G^(0) ← structuralLearning(D̂)
4:     i = 0
5:     repeat
6:         θ^(0) ← parametricLearning(D̂, G^(i))
7:         j = 0
8:         repeat
9:             D̂ ← completeData(D, θ^(j), G^(i))
10:            θ^(j+1) ← parametricLearning(D̂, G^(i))
11:            j = j + 1
12:        until (diff(θ^(j), θ^(j−1)) < ε) Or (j = maxItP)
13:        D̂ ← completeData(D, θ^(j), G^(i))
14:        G^(i+1) ← findMaxNeighborStructure(D̂, G^(i))
15:        i = i + 1
16:    until (G^(i) = G^(i−1)) Or (i = maxItS)
17:    return (G^(i), θ^(j))
18: end procedure
```

label proportions during the learning process in order to obtain more accurate Bayesian network classifiers. The pseudo-code of Algorithm 1 illustrates the mechanics of the Structural EM strategy implemented by our LLP method.

In order to build an initial model, the method first learns the whole network structure from a complete dataset that previously has been obtained in the data-initialization step (line 2 in Algorithm 1). This first structure is learnt by means of the specific methods presented in Section 3.4 for each type of classifier; the method of Friedman et al. [58] for TAN classifiers, and the method of Sahami [152] for KDB classifiers (line 3 in Algorithm 1). Later, this original structure will be iteratively improved (line 14 in Algorithm 1) using a one-step local search method. The neighborhood for the local search is composed of all the structures that can be obtained by removing one edge from the original structure and adding another different edge which keeps the conditional independence assumptions of the given type of classifier. That is, it only considers those changes that produce a network structure of the same type as the original structure. For example, the neighborhood of a TAN classifier groups all the tree augmented naive Bayes structures that can be obtained by removing one conditional dependence between two predictive variables and adding another dependency between a different pair of predictive variables. The best step is calculated according to the last version of the complete dataset (line 13 in Algorithm 1).

The initial model is completed with the parametric learning step (line 6 in Algorithm 1), which performs a classical maximum-likelihood learning of the model parameters given the current structure. Using the same procedure,

the model parameters are re-estimated (line 10 in Algorithm 1) for each new completion of the dataset (line 9 in Algorithm 1).

There are two procedures in the Structural EM strategy that are susceptible to using the information provided by the label proportions and both involve filling up the missing data. On the one hand, the data-initialization procedure where the original dataset is completed according to some heuristic criteria that fulfills the label proportions (line 2 in Algorithm 1). On the other hand, the data-completion procedure (lines 9 and 13 in Algorithm 1), which takes into account the prediction given by the current fit of the model in order to complete the dataset fulfilling the label proportions.

The most important application of the information given by the label proportions is its use in the reduction of the number of possible assignments of labels to the instances of non-full bags (unlabeled instances). Given a group of N_i unlabeled instances with a class variable that takes its value from a set $\mathcal{C}$, the number of possible assignments in this scenario without any further label information is $|\mathcal{C}|^{N_i}$, because all the possible assignments are considered (each instance with each possible class label in $\mathcal{C}$). But, if this group of instances forms a bag $\boldsymbol{B}_i$ with its corresponding label counts $\{N_{ic}\}_{c\in\mathcal{C}}$ (or label proportions, note that $p_{ic} = N_{ic}/N_i, \forall c \in \mathcal{C}$), which indicate that there are N_{ic} instances that belong to class c in $\boldsymbol{B}_i$, the number of possible assignments will be reduced to:

$$s_i = \binom{N_i}{N_{i1}\dots N_{ic}} = \frac{N_i!}{\prod_{c\in\mathcal{C}} N_{ic}!} \tag{4.1}$$

because not all the label assignments are possible. In this situation, the assignment of a label to an instance in $\boldsymbol{B}_i$ affects other assignments in the same bag, i.e., the assignment of $\boldsymbol{x}^j \in \boldsymbol{B}_i$ to class c affects the probability that any of the other instances in $\boldsymbol{B}_i$ belongs to each possible class label. For this reason, the individual assignments of labels can not be considered independently in a LLP scenario; joint assignments of labels to all the instances of a bag should be considered. A joint assignment is represented as a *completion* of labels, $\boldsymbol{e} = (e_1, e_2, \dots, e_{N_i})$, where each e_j takes its value from $\mathcal{C}$ and represents the class value that is assigned to the j-th instance of $\boldsymbol{B}_i$. According to the previous reasoning, only those completions that fulfill the label proportions of the corresponding bag are allowed:

$$p_{ic} = \frac{1}{N_i}\sum_{j=1}^{N_i} \mathbb{I}[e_j = c], \forall c \in \mathcal{C}$$

where $\mathbb{I}[condition]$ returns 1 if *condition* is true and 0 otherwise. The completions that fulfill this condition are known as *consistent completions*. Each bag $\boldsymbol{B}_i$ has its own set $\boldsymbol{Z}_i$ of consistent completions of size s_i.

In the same way, a *probabilistic completion* $\boldsymbol{E}$ assigns instance $\boldsymbol{x}^j$ to class label c with probability E_{jc}, where $\sum_{c\in\mathcal{C}} E_{jc} = 1$ and $E_{jc} \geq 0, \forall \boldsymbol{x}^j \in \boldsymbol{B}_i$ and $\forall c \in \mathcal{C}$. Specifically, a *probabilistic consistent completion* of a bag $\boldsymbol{B}_i$ is

a probabilistic joint assignment that fulfills, on average, the label proportions of $\boldsymbol{B}_i$:

$$p_{ic} = \frac{1}{N_i} \sum_{j=1}^{N_i} E_{jc}, \forall c \in \mathcal{C}$$

Since it is impossible to consider individual label assignments to the instances in a non-full bag of a LLP problem, the probability of assigning a label c to an instance $\boldsymbol{x}^j$ cannot be independently calculated. It should be calculated by means of the probability of the joint assignment of labels to all the instances of the bag. As instances are considered to be independently generated, the joint probability is calculated as the product of the conditional probabilities of each class label given the corresponding instance, using the current model. In this way, for a non-full bag $\boldsymbol{B}_i$, the joint probability of a consistent completion, $\boldsymbol{e}$, is:

$$p(\boldsymbol{e}|\boldsymbol{B}_i) = \prod_{j=1}^{N_i} p(C = e_j | X_1 = x_1^j, \ldots, X_n = x_n^j) \tag{4.2}$$

As previously mentioned, the information of the label proportions has been incorporated to our method in the data-initialization (line 2 in Algorithm 1) and data-completion (lines 9 and 13 in Algorithm 1) procedures. Both procedures build consistent completions to complete the missing data of the original dataset: the class label of instances in non-full bags. In the data-completion procedure, where there exists a current fit of the model that can be used to fill up the dataset in a more reliable way, we propose three different approaches to build the consistent completions using this model. Note that the techniques explained are applied individually to each bag of the dataset.

PEM is a probabilistic version that, in order to fill up the class variable of the instances in a non-full bag $\boldsymbol{B}_i$, calculates the probability of these instances belonging to each possible class. These calculations use the concept of joint probability (Equation 4.2) and consider all the consistent completions in order to build the probabilistic consistent completion ($\boldsymbol{E}$) that fills up the missing data. Given an instance $\boldsymbol{x}^j$ in $\boldsymbol{B}_i$, the probability of belonging to class c (E_{jc}) is calculated by adding up the joint probability of the consistent completions that assign instance $\boldsymbol{x}^j$ to the class label c and, then, normalizing it with respect to all possible class labels:

$$E_{jc} = \frac{\sum_{\boldsymbol{e} \in \boldsymbol{Z}_i | e_j = c} p(\boldsymbol{e}|\boldsymbol{B}_i)}{\sum_{\boldsymbol{e} \in \boldsymbol{Z}_i} p(\boldsymbol{e}|\boldsymbol{B}_i)} \tag{4.3}$$

In the initialization of this version, the dataset is completed with the probabilistic completion $\boldsymbol{E}$ that assigns the instances in $\boldsymbol{B}_i$ to each class label with probability equal to the label proportions of the bag:

$$E_{jc} = p_{ic}, \forall \boldsymbol{x}^j \in \boldsymbol{B}_i \wedge \forall c \in \mathcal{C}$$

NPEM is a non-probabilistic version. In the initialization procedure, the instances of each bag $\boldsymbol{B}_i$ are assigned to the class label that indicates a randomly-chosen consistent completion $\boldsymbol{e}$ of $\boldsymbol{B}_i$. In the data-completion procedure, this version considers all the consistent completions and selects the one with the highest joint probability. Then, according to the selected consistent completion, each instance in the bag is assigned to the corresponding label.

The two versions presented, PEM and NPEM, need to go through all the consistent completions in order to calculate the (probabilistic) consistent completion that fills up the original dataset, which determines the complexity of the methods. In this way, in the E-step of both versions PEM and NPEM, estimating the consistent completion of a non-full bag of N_i instances has a time complexity:

$$N_i \cdot ((|\mathcal{C}| - 1) \cdot T(\mathbb{M}) + (|\mathcal{C}| - 1)) + N_i \cdot s_i$$

where s_i is the number of consistent completions (Eq. 4.1) and $T(\mathbb{M})$ is the time complexity required by the current fit of the model $\mathbb{M}$ to calculate the conditional probability $p_{\mathbb{M}}(c|\boldsymbol{x})$ (it is calculated for $|\mathcal{C}| - 1$ labels as the last one can be calculated by summing up the others). This is computationally expensive and becomes unfeasible when the uncertainty of the LLP scenario grows, which in LLP implies a large number of consistent completions per bag. We propose a third version to deal with high complexity scenarios, i.e., LLP scenarios where bags are large and/or their label proportions are balanced.

MCEM also carries out a probabilistic completion. While it is initialized as the PEM version, it uses a Markov Chain Monte Carlo (MCMC) procedure to obtain an approximate probabilistic completion in the data-completion stage [121]. MCMC [67, 13] is an iterative procedure that uses Markov chains to approximate a probability distribution f of interest and the Monte Carlo strategy to approximate expectations from samples $\boldsymbol{X}^{(t)}$ drawn from the Markov chain, $E(f(\boldsymbol{X}); s) \approx \frac{1}{s}\sum_{t=1}^{s} f(\boldsymbol{X}^{(t)})$. If the Markov chain fulfills irreducibility and aperiodicity conditions, as samples are drawn from a finite number of possible states, the sequence converges to a stationary distribution that simulates the probability distribution of interest.

Since the samples drawn previous to the chain convergence stay dependent on the initial state, MCMC implements a *burn-in* stage in which these samples are not considered to calculate the expectations [13]. It is usually the end-user who determines the number of discarded samples according to their estimation of the *time* needed for the chain to converge. In this way, the MCMC procedure requires two parameters: the number of samples for the burn-in stage (bi), and the *number of samples* that are actually used to approximate the expectation (s).

Specifically, our approach implements a rejection MCMC procedure [67]. Rejection means that, during the sampling process, a new sample can be rejected if its probability is lower than the probability of the previous sample.

Algorithm 2 Pseudo-code of the MCMC process implemented in MCEM.

procedure MCMC(bi, s, $\{N_{ic}\}_{c\in\mathcal{C}}$)
  for $t = 1 \rightarrow (bi + s)$ **do**
    $\hat{\boldsymbol{e}} \sim nextConsistentCompletion(\boldsymbol{e}^{(t)}, \{N_{ic}\}_{c\in\mathcal{C}})$
    $u \sim U(0, 1)$
    if $u \leq \alpha(\boldsymbol{e}^{(t)}, \hat{\boldsymbol{e}})$ **then**
      $\boldsymbol{e}^{(t+1)} \leftarrow \hat{\boldsymbol{e}}$
    else
      $\boldsymbol{e}^{(t+1)} \leftarrow \boldsymbol{e}^{(t)}$
    end if
  end for
  return $E_{jc} = \frac{1}{s}\sum_{t=bi+1}^{bi+s} \mathbb{I}[e_j^{(t)} = c], (j \in \{1, \ldots, N_i\}) \wedge (c \in \mathcal{C})$
end procedure

When this happens, the place of the rejected sample is filled in by a copy of the previous sample. Then, given the current state $\boldsymbol{X}^{(t)}$, the probability that a state $\hat{\boldsymbol{X}}$ becomes the new sample $\boldsymbol{X}^{(t+1)}$ in the Markov chain, $p(\hat{\boldsymbol{X}}|\boldsymbol{X}^{(t)})$, is:

$$p(\hat{\boldsymbol{X}}|\boldsymbol{X}^{(t)}) = \begin{cases} 0 & \hat{\boldsymbol{X}} \notin S(\boldsymbol{X}^{(t)}) \\ \dfrac{\alpha(\boldsymbol{X}^{(t)}, \hat{\boldsymbol{X}})}{|S(\boldsymbol{X}^{(t)})|} & \hat{\boldsymbol{X}} \in S(\boldsymbol{X}^{(t)}) \end{cases}$$

where $S(\boldsymbol{X}^{(t)})$ is the set of all the possible next states of $\boldsymbol{X}^{(t)}$ in the Markov chain, and $\alpha(\cdot, \cdot)$ represents the probability of non-rejection. This is defined as:

$$\alpha(\boldsymbol{X}, \boldsymbol{Y}) = \min(1, p_{\mathbb{M}}(\boldsymbol{Y})/p_{\mathbb{M}}(\boldsymbol{X}))$$

where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are possible states of the Markov chain, and $p_{\mathbb{M}}(\cdot)$ is the probability of a state given the current fit of the model $\mathbb{M}$.

In this method, the MCMC process performs as shown in Algorithm 2 for each non-full bag $\boldsymbol{B}_i$. The parameters bi and s are the number of samples drawn for burn-in and to calculate the expectation, respectively. Each Markov chain state (or sample, $\hat{\boldsymbol{X}}$) represents a consistent completion $\boldsymbol{e}$ of $\boldsymbol{B}_i$. In this Markov chain, the next state $\boldsymbol{e}^{(t+1)}$ (abusing the notation, we use $\boldsymbol{e}^{(t)}$ to denote the consistent completion $\boldsymbol{e} \in \boldsymbol{Z}_i$ sampled at step t, $\boldsymbol{X}^{(t)} = \boldsymbol{e}$) could be any state with a consistent completion that has swapped two assignments of different labels from $\boldsymbol{e}^{(t)}$. The number of possible next states for any state $\boldsymbol{e}^{(t)}$ is:

$$|S(\boldsymbol{e}^{(t)})| = \sum_{c,c'\in\mathcal{C}|c\neq c'} N_{ic} \cdot N_{ic'}$$

It can be easily demonstrated that the implemented Markov chain fulfills the aperiodicity and irreducibility conditions, and that in the limit, the stationary distribution is our probability distribution $p_{\mathbb{M}}(\cdot)$ of interest.

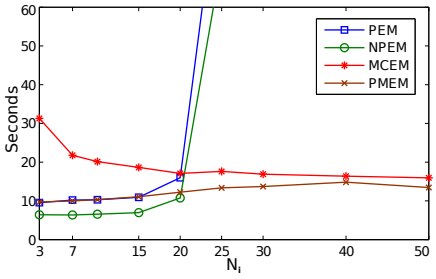


Fig. 4.1. Average computational time needed by the four versions of the proposed algorithm to learn TAN classifiers over 30 datasets (sampled from TAN models, with 31 binary variables —including the class variable— and 1000 examples). The bag size (N_i) is varied to simulate different experimental conditions. All the tests have been performed with an Intel Core i5 (2,3 GHz) with 4GB of main memory.

In practice, the next state is composed in such way that two randomly chosen instances assigned to different classes in $\boldsymbol{e}^{(t)}$ just swap their class labels ($nextConsistentCompletion(\cdot,\cdot)$ in Algorithm 2). Moreover, in our approach the probability $p_{\mathbb{M}}(\cdot)$ is the joint probability of the consistent completion of the given state using the current fit $\mathbb{M}$ of the Bayesian network model (Equation 4.2). The MCMC process approximates the probability of each instance in $\boldsymbol{B}_i$ belonging to each class and uses these probabilities to compose the probabilistic completion, $\boldsymbol{E}$, which is then used to complete the data. The probability of assigning an instance $\boldsymbol{x}^j$ to a class c is approximated as the proportion of samples of the chain in which the referred instance belongs to class c:

$$E_{jc} = \frac{1}{s} \sum_{t=bi+1}^{bi+s} \mathbb{I}[e_j^{(t)} = c], \forall \boldsymbol{x}^j \in \boldsymbol{B}_i \wedge \forall c \in \mathcal{C} \tag{4.4}$$

Therefore, the time complexity of this version MCEM is independent of the uncertainty that involves the specific LLP scenario. In this case, the time required to compute the approximate probabilistic completion of a non-full bag is constant in the number of samples (bi for burn-in and s for calculating expectations):

$$N_i \cdot ((|\mathcal{C}| - 1) \cdot T(\mathbb{M}) + (|\mathcal{C}| - 1)) + N_i \cdot (bi + s)$$

The behavior that motivated us to build this third version of the method is shown in Fig. 4.1. It represents the computational time spent by the method to learn tree-augmented naive Bayes (TAN) classifiers using the different data-completion approaches. The computational time that MCEM needs to learn a classifier is almost constant when the dataset is aggregated with different bag sizes (N_i). However, the other two previously presented methods become

unfeasible as N_i grows. A similar figure is obtained if other Bayesian network classifiers are learnt, but in different time-scales according to their learning complexities (TAN [58] or 2DB [152]). These complementary figures can be seen in the webpage associated with this study[1].

The application of the MCMC procedure to all the bags carried out by this third version of the method can produce a senseless situation. As shown in Fig. 4.1, in the least complex LLP scenarios our MCEM method spends twice or even three times the computational time required by the exhaustive versions (PEM and NPEM) to perform the *exact* calculations that MCEM tries to *approximate*. This matter is addressed in a fourth version of the method:

PMEM is a hybrid of both probabilistic versions previously presented. The ideas behind PEM, which calculates the exact probability of each class label for the instances of a bag taking into account the label proportions, and MCEM, which approximates these probabilities through a MCMC procedure, are combined in this last version. By default, PMEM builds up a probabilistic completion calculating the probability of an instance belonging to each class label as PEM does (Eq. 4.3). If the number of consistent completions of a specific bag is unfeasible to this exhaustive approach (see Fig. 4.1), an approximate probabilistic completion is built up by means of the MCMC procedure (MCEM, Eq. 4.4). The decision threshold, which could be an interesting topic for future research, is based on the MCMC-specific parameters. Thus, if the number of consistent completions of a particular bag is larger than the summation of bi (number of samples for burn-in) and s (number of samples for calculating the expectation), the approximate MCEM version is applied. Otherwise, it makes use of PEM.

The immediate benefit of this approach can be observed in Fig. 4.1. In general, it is the best proposal in terms of computational time: in scenarios of low uncertainty, it behaves similarly to PEM and in more complex scenarios it overcomes MCEM since the probabilistic completion of some particularly unbalanced bags (the label proportions indicate that a specific class label is under- or over-represented in that bag) is addressed with the exact procedure of PEM. According to this description, the time complexity of this last version for computing the probabilistic consistent completion of a non-full bag is:

$$N_i \cdot ((|\mathcal{C}| - 1) \cdot T(\mathbb{M}) + (|\mathcal{C}| - 1)) + N_i \cdot \min\{s_i, (bi + s)\}$$

4.4 Experiments

We have designed a set of experiments with four main objectives: (1) to test the benefit of using the information provided by the label proportions during the learning process, (2) to analyze the precision of the approximations

[1] `http://www.sc.ehu.es/ccwbayes/members/jeronimo/LLP/`

performed by our MCMC procedure, (3) to evaluate our proposals when dealing with different experimental conditions and (4) to compare it with other state-of-the-art methods in LLP.

The most common real applications of LLP involve private data, which reduces the public availability of real LLP datasets. For this reason, it is common in the related literature to use synthetic [99] or classical supervised datasets [128, 136], and transform them into LLP domains in order to validate the proposed algorithms. In these experiments, we use both kinds of data.

The proposed methods have been implemented using a publicly available Java language data mining library[2] developed by our research group. In the webpage associated with this study we release an easy-to-use executable code of our proposal.

Simulating LLP datasets. A classical labeled dataset can be transformed into a LLP dataset by building (or aggregating) bags with labeled instances. *Aggregation* is the process in which, somehow, the instances of a dataset are grouped in bags and, for each bag, the class labels are separated from their instances and used to calculate the label proportions.

In our preliminary study [80], in order to control the complexity of an aggregated LLP dataset, we proposed a method for aggregation that allows to control the uncertainty in the bags of an aggregated dataset. Based on the method proposed by Musicant et al. [128], we use the *mean label entropy* (MLH) to measure the uncertainty of a LLP dataset. MLH calculates the entropy of the label distribution (given by the label proportions) that the bags of the dataset have on average.

The method aggregates bags in a dataset such that the MLH value of the dataset reaches a desired value. First, the instances of the original labeled dataset are ordered according to their class label, where the instances with the same label appear consecutively. The minimum MLH value is obtained when bags are aggregated with contiguous instances over this ordered dataset. Based on this, by swapping two instances that have different labels and are located in different bags, the MLH value is modified. Then, a simple way to configure a dataset with a specific level of entropy is to swap instances until the desired MLH level is reached. In order to make this more comprehensible, MLH is mapped into the interval $[0, 1]$ dividing it by the maximum MLH value of the domain. A LLP dataset is considered to reach its maximum MLH value when all its bags are non-full bags and all of them fulfill the label proportions of the whole dataset.

Evaluation of learning methods in the LLP scenario. The evaluation of a learning method in a LLP scenario presents new challenges. The instances of a LLP dataset are provided grouped in bags and, some of these (non-full bags) are indivisible for the validation process since the class labels of individual instances are unknown (i.e., different labels in a bag make it impossible to

[2] ICLab: `http://sourceforge.net/projects/iclab/`

know, a priori, the specific label of each instance). Therefore, the adaptation of classical validation techniques (cross-validation, training/test, bootstrap...) to this framework is not immediate since some dataset divisions are invalid. In particular, the popular cross-validation (CV) technique requires performing a division of the dataset into validation folds such that the number of instances in all the folds is the same. Making such a division of the original LLP dataset in CV-folds by respecting the integrity of the bags and, at the same time, trying to keep the same number of instances in each fold is not straightforward. In fact, this problem can be seen as a generalization for more than two subsets of the classical combinatorial *optimization* problem called "number partitioning".

This issue is even more complicated when the validation folds are also required to be stratified. In this case, the optimization problem is constrained by an additional condition: the general label proportions of the folds have to fulfill the global label proportions of the dataset.

In order to skip this optimization process and to be able to compare with other methods in the related literature, in these experiments we reproduce the strategy for the evaluation of the LLP methods that other authors have carried out previously [128, 136]. Given the original labeled dataset, first it is divided into folds for validation using classical validation techniques for labeled datasets. Next, the instances in those folds used for training are separated from their labels as long as bags are aggregated. Thus, the instances for testing remain the original labeled instances, which makes the validation easier. However, as the aggregation step depends on a previous validation division, the validation process is performed with different bag configurations at each iteration. Moreover, information that is unavailable in a real LLP dataset (individual instance labels) is used to stratify the training/testing datasets in the validation process.

Default setting of the method parameters. Our method requires some parameters to work. We have configured the current implementation with a set of default values for these parameters. If need, they can be easily changed.

The base strategy of our proposal, the Structural EM approach, uses three parameters: a threshold that indicates parametric convergence —the loop stops when the relative difference between the MLE value of two models learnt in consecutive iterations is below this threshold—, and that is set by default to 0.1%; and the maximum numbers of iterations for both structural and parametric convergence, which are both fixed to 200 iterations by default. The MCEM version also requires another two parameters to be set: by default, 1000 samples of burn-in and 10000 samples to approximate the label probability expectation.

4.4.1 The usefulness of the information provided by the label proportions: the semi-supervised learning approach as baseline-performance reference

A typical LLP dataset has full and non-full bags, that is, a subset of instances with known class labels (those in full bags, where all the instances belong to the same known class) and a complementary subset of instances with unknown class labels (those in non-full bags). From this point of view, LLP resembles the description of a semi-supervised dataset [23]. In fact, a LLP dataset can be easily transformed into a semi-supervised dataset by removing the bag configuration of the LLP domain. Moreover, the objective of both LLP and semi-supervised approaches is to predict the class of new unlabeled single instances. In this way, semi-supervised learning can be considered the most similar learning approach with respect to LLP if the extra information that the LLP approach considers is not taken into account. In the following set of experiments, our aim is not to compare the LLP and semi-supervised frameworks, but to check the benefits of the use of this extra information in the learning process, using the semi-supervised approach as a baseline-performance reference.

For these experiments we have used synthetic datasets, which are sampled from tree-augmented naive Bayes (TAN) models. The sampled models have 30 binary predictive variables and a class variable with 2 or 3 class labels. The model parameters are randomly generated by sampling a Dirichlet distribution with all the hyper-parameters equal to 1. The datasets have been sampled with 100 or 1000 examples. For each combination of these characteristics, 30 datasets have been sampled, resulting in a final number of 120 synthetic datasets (2 class cardinalities $\times$ 2 sample sizes $\times$ 30 datasets).

In order to generate different semi-supervised learning scenarios, we transform LLP datasets specifically generated with increasing complexity. We establish the required uncertainty for a LLP scenario with a particular configuration of two parameters: the bag size (N_i) and the MLH entropy. As explained in Section 2.1, the uncertainty of a LLP dataset is related to the number of full/non-full bags (the larger the uncertainty, the lower the number of full bags). In these experiments, the semi-supervised datasets are obtained as follows: given a LLP dataset (generated as explained before), the instances in a full bag are transformed into labeled instances of the resulting semi-supervised dataset; those in non-full bags are the unlabeled instances in the semi-supervised domain. Thus, although we do not establish manually the proportion of unlabeled instances in semi-supervised experiments, by controlling both parameters (N_i and MLH) we can infer the proportion of unlabeled instances in the resulting semi-supervised dataset. In this case, we use bag size values $N_i = \{3, 15, 30, 50\}$ and entropy values MLH $= \{0.0, 0.25, 0.5, 0.75\}$.

Specifically for these experimental setting, we have implemented a method for semi-supervised learning which follows the basic Structural EM strategy as explained in Section 3.2. It performs a probabilistic completion and, therefore,

in the E-step the missing data (i.e., class value of unlabeled instances) is completed using the probability that the particular instance belongs to each possible class label (according to the current fit of the model). Similar to PMEM (the other method tested), the semi-supervised method performs an uninformed initialization, where the unlabeled instances are assigned to a randomly-chosen class.

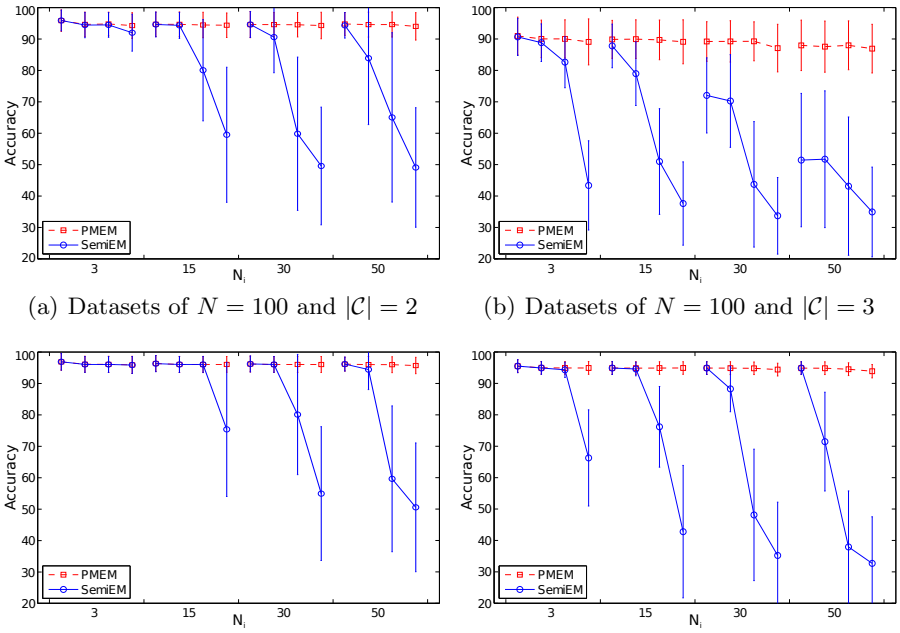
Our PMEM version (which was selected due to its ability to cope efficiently with all the aggregated LLP scenarios) and the exposed semi-supervised method are evaluated learning three different types of Bayesian network classifiers (NB, TAN and 2DB) in different experimental conditions. Both methods have been validated using a 10×5 fold cross validation (CV). As explained before, the synthetic dataset is aggregated into a LLP dataset using only the training instances at each iteration of the CV process. Subsequently, this LLP training set is transformed into a semi-supervised dataset by removing its bag configuration. This procedure guarantees that both the division in folds for the CV process and the relative proportion of labeled/unlabeled instances are the same for the evaluation of both methods.

Note that, as any EM-based algorithm, tuning the methods could partially improve their performance in these experiments. However, for the sake of simplicity, we have used a basic configuration for both the LLP and the semi-supervised methods in order to design a fair experimental setting and not to confuse the purpose of these experiments: to emphasize the potential of the label proportions.

Results. A summary of the experiments is shown in Fig. 4.2. Each subfigure represents a set of experiments using different dataset characteristics (2 or 3 class labels, and 100 or 1000 instances). In order to analyze the results, two general considerations have to be taken into account. In general, since it is more difficult to guess the correct label of an instance if the number of candidate labels is larger, the uncertainty involving partial-unlabeled datasets with 2 possible class labels is considered to be lower than the uncertainty of datasets with 3 possible class labels. On the other hand, more accurate models can be learnt when more data is available.

When the dataset has a binary class, a semi-supervised method is able to learn models as accurate as those learnt by a LLP method when there is low uncertainty in terms of entropy (MLH) and bag size (N_i). When the uncertainty grows, the performance of the semi-supervised method decreases; in this scenario, a larger dataset can not compensate the information provided by the label proportions. Thus, the difference between the performance of the two methods remains noticeable (Figures 4.2(a) and 4.2(c)).

While the behavior is similar for datasets with 3 class labels, the results are even more sensitive to the degree of uncertainty. Variations of the parameters that define the uncertainty of a LLP scenario (bag size and, mainly, entropy) cause large differences between the performance of the LLP and the semi-supervised approaches (Figures 4.2(b) and 4.2(d)). In this case, a larger



(a) Datasets of $N = 100$ and $|\mathcal{C}| = 2$

(b) Datasets of $N = 100$ and $|\mathcal{C}| = 3$

(c) Datasets of $N = 1000$ and $|\mathcal{C}| = 2$

(d) Datasets of $N = 1000$ and $|\mathcal{C}| = 3$

Fig. 4.2. Mean accuracy and associated standard deviation of our PMEM and the semi-supervised method to learn TAN classifiers in a 10×5 fold CV over 30 datasets (sampled from TAN classifiers with 30 binary predictive variables). Each subfigure represents experiments with different dataset size (N) and number of class values ($|\mathcal{C}|$). In the x-axis, 16 different LLP scenarios used to aggregate the synthetic datasets (the combination of 4 values of bag size $N_i = \{3, 15, 30, 50\}$ and 4 values of entropy MLH $= \{0.0, 0.25, 0.5, 0.75\}$) are represented.

dataset can compensate the information provided by the label proportions only when the uncertainty is low, i.e., the values of one or both parameters, bag size (N_i) and entropy (MLH), are low.

Taking as a reference the semi-supervised learning framework, where no information of supervision is available for the unlabeled instances, the additional information of supervision provided by the label proportions has been shown to be a solid contribution to learn more accurate classifiers. In the least complex scenarios, with a large number of full bags, the difference of performance between both frameworks can be overcome with more examples. However, the contribution of the label proportions is really significant when dealing with a complex LLP scenario. In this case, a larger dataset can not compensate the extra information of LLP.

4.4.2 An evaluation of the MCMC procedure by means of probabilistic label assignments

In the previous experiments we have shown the importance of incorporating the label proportions to the learning process. According to Figure 4.2, the behavior of the hybrid PMEM method remains similar along the complexity spectrum of LLP scenarios; from the least complex domains, where PMEM uses the exhaustive procedure (PEM), to more complex scenarios, where it applies the approximate MCMC procedure (MCEM). Based on these results, we could think that the MCEM method is able to approximate precisely the exact reasoning of PEM. In this section, we have carried out a set of experiments in order to test this hypothesis. For this comparison, we have study the probabilistic labels assigned by both methods in order to show how MCEM approximates the behavior of PEM not only in terms of accuracy.

The same synthetic datasets generated for the previous set of experiments have been used: 120 TAN-generated synthetic datasets (2 class cardinalities $\times$ 2 sample sizes $\times$ 30 datasets). Our exact PEM method, and the approximate MCEM have been evaluated learning three different types of Bayesian network classifiers (NB, TAN and 2DB) in different experimental conditions. Both methods have been validated using a 10×5 fold cross validation (CV) in different LLP scenarios, aggregated with bag size values $N_i = \{3, 7, 15\}$ and entropy values MLH $= \{0.0, 0.25, 0.5, 0.75\}$. As PEM is the version which performs exact calculations, it has been taken as a reference in the experiments. Note that the presence of this exhaustive method restricts the bag size values that can be used in the comparison (see Fig. 4.1).

In order to evaluate the comparison from different points of view, several relevant measures have been collected:

- Difference in terms of Root Mean Square Error (RMSE) between the final probabilistic labels assigned by PEM and MCEM to the unlabeled instances of the *training* dataset.
- Difference in terms of RMSE between the final probabilistic labels assigned by PEM to the unlabeled instances of the *training* dataset and their real labels.
- Difference in terms of RMSE between the probabilistic labels assigned to the *test* instances by the classifiers learnt with PEM and MCEM.
- Difference in terms of RMSE between the probabilistic labels assigned to the *test* instances by the classifiers learnt with PEM and their real labels.
- Accuracy of the classifiers learnt with PEM over the *test* instances.
- Accuracy of the classifiers learnt with MCEM over the *test* instances.

The referred RMSE difference has been calculated as follows:

$$RMSE(\boldsymbol{E}^{(u)}, \boldsymbol{E}^{(v)}) = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\frac{1}{|\mathcal{C}|} \sum_{c=1}^{|\mathcal{C}|} (E_{ic}^{(u)} - E_{ic}^{(v)})^2}$$

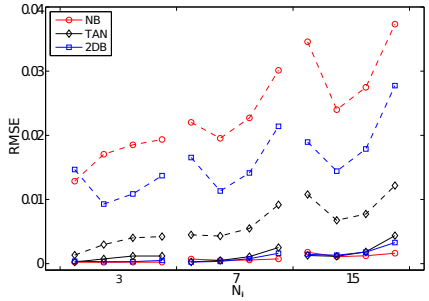
where $\boldsymbol{E}^{(u)}$ and $\boldsymbol{E}^{(v)}$ are two probabilistic completions that assign a class label c to the i-th instance with probability $E_{ic}^{(u)}$ and $E_{ic}^{(v)}$, respectively. When the comparison involves real labels, they are adapted to the probabilistic framework as follows: if c is the real label of the i-th instance, $E_{ic}^{(o)} = 1$; the rest of class values are assigned zero probability ($\forall c' \neq c \in \mathcal{C}, E_{ic'}^{(o)} = 0$).

Figure 4.3 shows the results of the presented comparison. Four subfigures describe the training process and its corresponding test step in two scenarios which only differ in the size of the dataset used ($N = \{100, 1000\}$). All the subfigures in Fig. 4.3 show mean values over 30 datasets $\times$ 10 repetitions. Related standard deviations are not included for the sake of clarity.

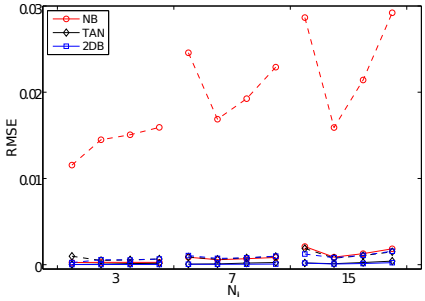
According to the continuous lines of Fig. 4.3(a) and Fig. 4.3(b), which show the RMSE difference of the probabilistic labels between PEM and MCEM, the approximate assignments of MCEM fit precisely the exact probabilistic labels of PEM. As expected, the difference slightly increases in larger complex scenarios. Moreover, when the dataset size (N) increases, the difference is reduced to 0 in almost any scenario. With larger datasets, the classifiers also tend to fit better the real labels (dashed lines), with the only exception of NB classifiers. That is, both methods are learning from data completed with similar probabilistic label assignments, which in fact are both close to the real labels (remember that in these experiments only generative TAN models are used).

As the RMSE differences of the top figures have been calculated with the last assignment of probabilistic labels to the instances of the training dataset previous to learn the definitive classifiers, it could be expected that if dashed lines are not placed in the lower part of the top figures (i.e., the probabilistic label assignments of PEM and MCEM do not fit the real labels), the predictions performed by these classifiers will not be accurate. For example, the referred behavior is observed with 2DB classifiers: when the dashed lines rise (in the top figures), the accuracy rates of the corresponding classifiers decrease in the bottom figures, and the differences between the probabilistic labels predicted by the classifiers (learnt with PEM and MCEM; continuous lines), and between those of PEM and the real labels (dashed lines) increase. Obviously, differences between the probabilistic label assignments of PEM and MCEM are also reflected in the different accuracy rates obtained by the corresponding classifiers (see the non-concentric disconnected markers in the bottom-row figures). This behavior is partially compensated with a larger dataset, which allows the methods to learn more precise probabilistic labels.

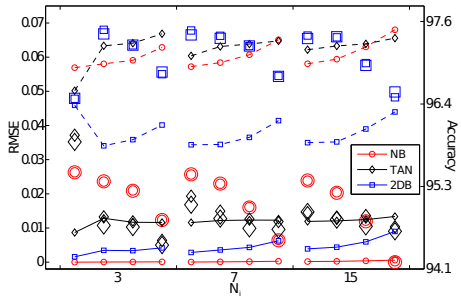
From the previous reasoning, we could formulate the following statement: the more imprecise the probabilistic labels in training, the less accurate the learnt classifiers. As explained, it holds in that direction; however, the opposite (more precise labels induce more accurate classifiers) is not correct. This is reflected in the behavior of the TAN classifiers, which overfit the training data. The difference between the probabilistic labels assigned by our methods and the real labels is relatively low in training (dashed lines in top figures) and



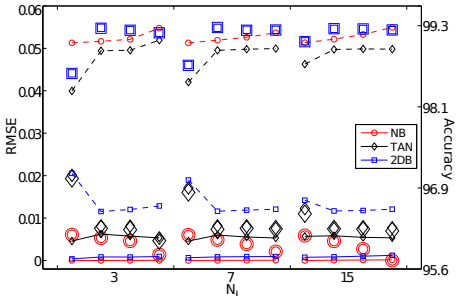
(a) Probabilistic label comparisons in training instances ($N = 100$).



(b) Probabilistic label comparisons in training instances ($N = 1000$).



(c) Probabilistic label comparisons and accuracy in test instances ($N = 100$).



(d) Probabilistic label comparisons and accuracy in test instances ($N = 1000$).

Fig. 4.3. Results of the comparison between PEM and MCEM evaluated in a 10×5 fold CV over 30 datasets (sampled from TAN classifiers with 31 binary variables, including class variable). The learnt NB, TAN and 2DB classifiers are displayed with different markers. In the x-axis, 12 different LLP scenarios are used to aggregate the synthetic datasets (the combination of bag size values $N_i = \{3, 7, 15\}$ and entropy values MLH $= \{0.0, 0.25, 0.5, 0.75\}$).
Figures in each column represent experiments with different dataset sizes ($N = \{100, 1000\}$).
Figures in the top row show the results of the comparison in *training* instances: the continuous lines represent the RMSE difference between the final probabilistic labels assigned by PEM and MCEM; the dashed lines represent the RMSE difference between PEM and the real labels.
Figures in the bottom row show the results of the comparison in *test* instances: the continuous lines represent the RMSE difference between the probabilistic labels assigned by classifiers learnt with PEM and MCEM; the dashed lines represent the RMSE difference between PEM and the real labels. The disconnected markers show the mean accuracy of the classifiers learnt by PEM (medium size) and MCEM (large size).

becomes notably larger in test (dashed lines in bottom figures). This implies a low accuracy of the TAN classifiers, as observed.

As the underlying generative models in these experiments are based on TAN models, the naive Bayes classifiers (which are not able to capture relationships between predictive variables) show the least accurate behavior. In this case, although the approximate MCMC procedure fits the exact procedure almost perfectly, neither MCMC nor the exact approach can fit the real labels.

4.4.3 Experiments with synthetic data

The main objective of this set of experiments with synthetic data is the study of the behavior of the four versions of the proposed method learning three types of Bayesian network classifiers from datasets aggregated in different LLP scenarios. Based on the previous experimental setting, we have performed an extended comparison with additional configurations for each experimental characteristic in order to validate our proposals over a broader set of LLP scenarios.

In order to sample the synthetic datasets, in addition to TAN models, naive Bayes, 2-dependence Bayesian network and Bayesian networks with unrestricted structure models have been used. Moreover, we have generated datasets with an intermediate number of examples ($N = 500$). Therefore, in these experiments the total number of synthetic datasets is 720 (4 generative models $\times$ 2 class cardinalities $\times$ 3 sample sizes $\times$ 30 datasets per configuration).

Regarding the aggregation of LLP scenarios with different uncertainty levels, the two involved parameters, bag size (N_i) and MLH entropy, take values $N_i = \{3, 7, 15, 20, 30, 40, 50\}$ and MLH $= \{0.0, 0.25, 0.5, 0.75\}$. As shown in Fig. 4.1, only the versions of our method based on MCMC can deal with bag sizes larger than 20, so PEM and NPEM are not evaluated for $N_i = \{30, 40, 50\}$.

Finally, we introduce another feature to the experimental configuration: three types of Bayesian network classifiers (NB, TAN and 2DB) are learnt by our algorithms for each synthetic dataset and LLP scenario.

As a summary, we have evaluated the 4 versions of the method using 10×5 fold CV when learning 3 Bayesian network classifiers from 720 synthetic dataset, each of them aggregated in 28 different LLP scenarios.

Results. In order to analyze the results of the large set of experiments performed in this section, we use different techniques. First of all, the statistical framework proposed by Demšar [46] and García and Herrera [63] is applied. This framework indicates a way to perform a statistical validation looking for significant differences in the performance of the methods under comparison. In our case, where several methods have to be compared, this framework first performs a Friedman test, which analyzes whether the methods follow the same

probability distribution (null-hypothesis). If the Friedman test null-hypothesis is rejected, several post-hoc tests are applied to compare the methods by pairs.

In this section, we have analyzed the obtained results from two different points of view: a comparison of our four methods (where the classifiers are considered another characteristic of the experimental setting); and a comparison where the learnt classifiers are considered as a part of the learning method (4 learning methods $\times$ 3 types of classifiers: 12 SEM+BNC methods). Two blocks of statistical tests have been applied, each of them from one of these two points of view.

Firstly, the presented statistical framework has been applied to compare the performance of the four versions of our proposal from a general point of view. In this way, we take into account the experiments performed by all the versions, discarding the experiments with bag size $N_i = \{30, 40, 50\}$, which were only tested in our MCMC-based versions. As exposed before, there are 24 groups of 30 datasets generated with the same characteristics. In this comparison, each dataset is aggregated in 16 different LLP scenarios and NB, TAN and 2DB classifiers are learnt from the aggregated LLP datasets. Then, taking into account all these factors, for each version of the method, the results of 34560 experiments have been obtained and used in this test.

A Friedman test [46] comparing the results of the different versions of our method (PEM, NPEM, MCEM and PMEM) indicates that they do not follow the same probabilistic distribution. Given the average ranks of Table 4.1, the referred test determines that there are statistically significant differences between the results of the four versions of our proposal when the type I error is fixed to $\alpha = 0.05$.

As the Friedman test rejects the null hypothesis, post-hoc paired tests have been performed to discover differences between pairs of methods using the Holm procedure [46]. In this case, the post-hoc paired tests can not find statistical differences between the PEM and the PMEM versions of our proposal at $\alpha = 0.05$, which could be expected as PMEM probably has used the exact approach —the same as PEM— frequently (only bag sizes lower than or equal to 20 have been tested). The rest of the tests reject the corresponding null hypothesis, indicating statistically significant differences between the pairs of methods compared. As a conclusion of this first set of statistical tests, based on the obtained results, the version with the largest rank, NPEM, can be considered the worst proposal, whereas a single best proposal can not be established between PEM and PMEM.

The second set of statistical tests has been applied to find significant differences between different types of Bayesian network classifiers learnt with

Method	NPEM	MCEM	PMEM	PEM
Av. Rank	2.791	2.427	2.397	2.385

Table 4.1. Average Ranks of the versions of our proposal

each version of our method (SEM+BNC combinations). In this way, 12 methods are considered in the comparison: each combination of a version of our method (PEM, NPEM, MCEM and PMEM) and a type of classifier (NB, TAN and 2DB). Therefore, the number of experimental conditions (11520) is three times smaller than in the previous comparison, since the type of classifier was considered before as a factor of the experimental conditions, being now part of the methods under comparison.

By means of the Friedman test, which rejects the null hypothesis, the results of the 12 tested methods are proved to involve statistically significant differences. Moreover, Figure 4.4 shows a graphical representation of the significant differences found between pairs of methods by the subsequent Holm procedure. Each method is positioned in the scale according to its mean rank. Methods connected by a bold horizontal line are not significantly different at the fixed $\alpha = 0.05$ threshold. According to this figure, the null hypothesis of the paired tests, which considers that the results of the two methods under comparison follow the same distribution, is not rejected in the following cases:

- When the versions PEM, MCEM and PMEM are tested learning the same type of Bayesian network classifier, the Holm procedure can not find significant differences between them. The main conclusion is that both the approximate version MCEM, and consequently the combined version PMEM, are able to obtain similar results with respect to PEM, regardless of the type of Bayesian network classifier.
- NPEM learning NB classifiers, and PEM and PMEM learning TAN classifiers, do not show statistical differences. This shows the inability of NPEM to learn Bayesian network classifiers as accurate as the other versions of our method. Although this depreciate behavior is observed with all the BN classifier, in this case the NB classifiers (which are globally ranked in an intermediate position) learnt by the NPEM show similar accuracy rates than other versions learning TAN classifiers (the classifier with lowest average results as shown in Tab. 4.4).

As previously explained, the uncertainty that involves the considered LLP scenario determines the complexity of the learning process in a LLP domain

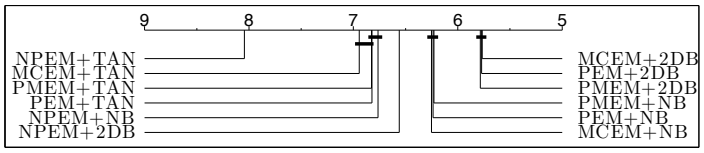
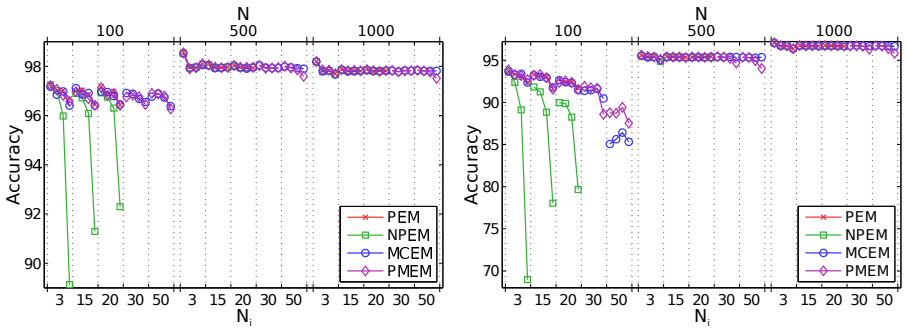


Fig. 4.4. Result of the Holm procedure, applied to the 12 SEM+BNC combinations (NB, TAN and 2DB classifiers learnt with the four methods), finds statistically significant differences at $\alpha = 0.05$. Each method is positioned in the scale according to its ranking and the bold horizontal lines link methods that are not significantly different.



(a) Experiments with datasets of $|\mathcal{C}| = 2$. (b) Experiments with datasets of $|\mathcal{C}| = 3$.

Fig. 4.5. Comparison of the capability of the four versions of our proposal to cope with the uncertainty of different LLP scenarios. In these figures, the average accuracies of TAN classifiers learned in different LLP scenarios are shown. Datasets of $N = \{100, 500, 1000\}$ instances, 30 predictive variables and a class variable of 2 or 3 class labels were generated from 2DB models (30 datasets for configuration). Each line links the results of a version of our method in different LLP scenarios that have the same bag size (N_i) and only differ in the entropy value, MLH = $\{0.0, 0.25, 0.5, 0.75\}$.

and, in this study, it is measured with two parameters: bag size (N_i) and MLH entropy. Thus, the uncertainty level grows when the values of these parameters increase. The capability of the four versions of our method to learn accurate classifiers in different LLP scenarios (which are simulated varying both parameters) is shown in Figure 4.5. In order to perform a more reliable comparison, the type of generative model (2DB) of the synthetic data and the type of learnt classifier (TAN) are fixed. We can conclude that NPEM is the most sensitive version to the increase of uncertainty, particularly when the available data is scarce ($N = 100$). MCEM, the strictly approximated approach, shows a behavior similar to that of PEM, the exact probabilistic approach. PMEM, the hybrid that combines the exact and approximate approaches, also shows a behavior consistent with the methods on which it is based. MCEM and PMEM, the only feasible versions when there exist bags of size $N_i = \{30, 40, 50\}$, show in these complex LLP scenarios a moderate reduction of accuracy. The differences between them are almost indiscernible in the results of the tests performed over datasets with two class values, which makes us think that PMEM must have had few opportunities to apply the exact reasoning approach. However, in the experiments with datasets of 3 class values the improvement is clear, which gives an idea of the additional complexity that involves a non-binary class variable and the difficulties of our approximate approach to learn in these scenarios. In general, although they can not be compared with the exact PEM version (the reference version in less complex scenarios), the behavior of the approximate versions in LLP sce-

narios with large bags draws a decreasing curve that follows the dynamic of the performance reduction of other less complex scenarios.

Although the behavior of the different versions of the method is similar for 2 or 3 class values (Figures 4.5(a) and 4.5(b)), they display different scales of accuracy. The different scale is due to the added complexity of the domain with three class labels with respect to the domain with only two class labels.

Moreover, the influence of the increasing size of the datasets can be appreciated in both subfigures of Fig. 4.5. The larger the datasets, the lower the effect of the uncertainty in the accuracy of the learnt classifiers. This is also reflected by means of the associate standard deviations, which are considerably reduced when the dataset size is increased. That could mean that the learnt models are more accurate, i.e., the SEM method frequently reaches similar optimal states. In fact, with large datasets, the differences between the versions of the methods become almost imperceptible.

From this set of experiments with synthetic data, we can conclude that the version that implements strictly a probabilistic approximated completion, MCEM, provides consistent estimations with respect to the real approach. In this way, statistical tests show that the results of MCEM are not significantly different from those of PEM. However, the most important conclusion is that the hybrid PMEM method can be reliably used in any LLP scenario, since it has been shown to properly integrate the exact and approximate approaches to take the best of both.

4.4.4 Comparison with state-of-the-art methods

In this last set of experiments, we present a comparison with a representative subset of LLP state-of-the-art methods. The papers in the current literature can be grouped depending on their definition of the LLP problem: those that follow a LLP definition based on proportions (Kück and Freitas [99], Quadrianto et al. [136], Rueping [150], Stolpe and Morik [174]); and those that follow a LLP definition based on aggregated outputs or counts (Musicant et al. [128], Chen et al. [27]). We have compared our method with the LLP techniques proposed in the most referenced paper (in related literature) of each group: Quadrianto et al. [136] and Musicant et al. [128]. These are compared with the four versions of our proposal learning three different types of Bayesian network classifiers (NB, TAN and 2DB). As exposed before, when N_i is larger than 20, MCEM and PMEM are the only affordable techniques.

Due to the non-public availability of some executable-code of these methods, we have designed separated comparisons for each paper. In the comparison with Quadrianto et al. [136], we carry out an honest reproduction of their experiments. With respect to the Musicant et al. [128] proposals, we have reproduced their best method following their description. In the same way, we have decided to use the same datasets which are, in both cases, real datasets from two public repositories (UCI [54] and LibSVM [22]). Equal-frequency with 3 intervals is used to discretize continuous variables.

dataset	MeanMap		PMEM		
	unweighted	weighted	NB	TAN	2DB
WDBC	23.29 ± 2.68	14.22 ± 1.79	12.13 ± 4.49	$\mathbf{10.18 \pm 2.90}$	11.95 ± 3.88
Australian	34.44 ± 4.03	29.58 ± 3.71	22.99 ± 1.96	$\mathbf{16.96 \pm 1.49}$	16.68 ± 1.05
SVMguide3	24.28 ± 2.20	$\mathbf{18.50 \pm 1.73}$	34.05 ± 3.74	31.13 ± 2.60	28.13 ± 1.59
Splice	33.43 ± 1.65	21.12 ± 2.59	22.12 ± 0.54	30.82 ± 1.03	40.05 ± 1.15
Protein	57.46 ± 0.02	57.46 ± 0.02	$\mathbf{47.84 \pm 1.09}$	48.20 ± 0.82	48.68 ± 1.12
SensIT	28.25 ± 2.60	$\mathbf{23.51 \pm 0.78}$	32.72 ± 0.64	33.77 ± 1.36	34.60 ± 2.14
DNA	20.01 ± 1.26	16.80 ± 1.19	$\mathbf{14.55 \pm 0.74}$	19.28 ± 1.32	17.13 ± 1.03

Table 4.2. Comparison of the PMEM version of our method, learning different types of Bayesian network classifiers, with respect to the unweighted and weighted MeanMap (Quadrianto et al. [136]). Several UCI/LibSVM datasets are aggregated using overdetermined systems for binary or 3-class label datasets. 10×10 fold CV is used for validation. The results are shown in terms of mean classification error and its associated standard deviation.

Comparison with Quadrianto et al. [136] proposal. The first comparison is performed with respect to the proposals of Quadrianto et al. [136]. The authors use real datasets from two public repositories (UCI [54] and LibSVM [22]) that are aggregated in different ways in order to create different LLP scenarios. The authors do not use any general strategy to transform labeled datasets into LLP datasets, but they specify the label proportions in a fixed number of bags with different bag sizes for each experiment. In terms of the number of bags, the authors carry out experiments where the number of bags is equal to the number of classes ($b = |\mathcal{C}|$), and experiments where the number of bags is larger (called *overdetermined systems* in [136]).

For the first set of experiments ($b = |\mathcal{C}|$), in the case of class-binary datasets, the authors aggregate a LLP dataset with two bags: A full bag (only instances of class 1) and a non-full bag with its label proportions matching the proportions of the class labels in the original dataset. This definition is equivalent to that of the *positive unlabeled learning* framework [19]. In the case of datasets with three classes, the authors propose similar extreme conditions. We consider that it makes no sense to use a LLP method to learn from this kind of data when the literature provides specific methods to cope with this extreme scenario. For this reason, our comparison does not cover this set of experiments.

In overdetermined system experiments, where the number of bags is larger than the number of class labels, 8 bags are aggregated following a table of label proportions exposed in [136] for class-binary datasets and, in a similar way, 6 bags are aggregated for 3-class label datasets. For these datasets, the authors evaluate two versions of their proposal, called weighted and unweighted MeanMap.

We have reproduced the experimental conditions of the overdetermined systems and, using the same datasets, our hybrid PMEM method has been evaluated using 10×10 fold cross validation (the validation approach used in the framework of Quadrianto et al. [136]). Given the fact that bags in these experiments are huge (in the smallest dataset, WDBC, more than 85% of

the non-full bags aggregated with overdetermined systems have more than 30 instances), the exhaustive PEM and NPEM versions can not be applied. Moreover, in such complex LLP scenarios, although PMEM is not expected to improve significantly the results of MCEM, it has been chosen for these experiments in order to take advantage of the least complex bags, if possible. For the same reason, the rest of versions of our method can not be applied. The MCMC-specific parameters were fixed to $10,000$ iterations of burn-in and $10,000$ samples. Some datasets could not be used because, in their original form [54, 22], they have more than 2 classes and the authors used them as binary datasets without specifying how the original data was transformed.

The results are shown in Table 4.2 in terms of mean classification error and its associated standard deviation. Results for both the weighted and unweighted versions of the Quadrianto et al. [136] method and our PMEM version learning different types of Bayesian network classifiers are shown. The classifiers learnt by PMEM show competitive results with respect to Quadrianto et al. [136] proposals. In four of seven domains, our method achieves a lower error rate than their method. In only two domains, weighted MeanMap clearly outperforms PMEM. In the case of the Splice dataset, it is not possible to establish the method with best results without a detail specific comparison. Despite the fact that weighted MeanMap [136] shows a slightly lower error rate than PMEM learning NB classifiers in this domain, it also shows a larger standard deviation that makes the error difference non-reliable. Therefore, in more than a half of the datasets used in this reproduction of the Quadrianto et al. experimental setting [136], PMEM actually outperforms their best version.

Comparison with Musicant et al. [128] proposals The second comparison is performed with respect to the proposals of Musicant et al. [128]. In this paper, the authors present their techniques (basic adaptations to the LLP approach of K-Nearest Neighbor, Artificial Neural Network, Support Vector Machine and Decision Tree classifiers) and evaluate them in different LLP scenarios. The evaluation was performed over three UCI [54] datasets: Ionosphere, Dermatology and Breast Cancer Wisconsin. Looking at the results of their experimental section, the method that learns Decision Trees (DT) [128] shows the best general performance. Bearing this in mind, we have decided to compare only with the DT proposal in order to reduce the size of this comparison,.

Musicant et al. [128] indicate that their methods could be tuned in order to improve the results. Similarly, the objective of these experiments is to show how different LLP scenarios affect the learning process of each method. This can only be fairly achieved by fixing the base classifier parameters and, in order to generate different experimental conditions, modifying only the parameters that determine the uncertainty of a LLP scenario (bag size and entropy). In this way, we have reproduced the method that learns DT classifiers according to the provided description [128] and, for the parameters of our

	0.0	0.25	0.5	0.75	0.0	0.25	0.5	0.75	
3	96.3 ± 0.51	96.5 ± 0.46	96.3 ± 0.34	96.3 ± 0.40	81.2 ± 3.82	81.8 ± 3.46	78.7 ± 4.24	81.1 ± 3.77	PEM
	96.1 ± 0.56	96.3 ± 0.36	96.3 ± 0.22	96.5 ± 0.39	84.5 ± 2.39	85.3 ± 1.87	84.6 ± 3.25	82.7 ± 2.77	
	96.2 ± 0.41	96.1 ± 0.42	95.9 ± 0.41	96.0 ± 0.41	88.9 ± 2.28	89.6 ± 2.40	**90.0 ± 2.80**	**89.9 ± 2.38**	
	96.6 ± 0.36	96.4 ± 0.30	**96.5 ± 0.38**	**96.6 ± 0.46**	85.2 ± 2.40	83.0 ± 3.06	78.6 ± 3.95	77.5 ± 4.90	NPEM
	96.3 ± 0.32	96.2 ± 0.41	96.4 ± 0.40	96.3 ± 0.46	84.8 ± 2.95	84.5 ± 2.60	84.5 ± 1.87	83.7 ± 2.86	
	95.9 ± 0.44	95.8 ± 0.32	96.2 ± 0.36	96.1 ± 0.42	89.6 ± 1.44	**90.4 ± 0.98**	89.3 ± 1.89	89.7 ± 2.46	
	96.5 ± 0.41	**96.6 ± 0.36**	**96.5 ± 0.38**	96.3 ± 0.87	82.5 ± 3.75	83.6 ± 2.45	81.2 ± 5.48	77.5 ± 6.26	MCEM
	96.5 ± 0.28	96.5 ± 0.40	96.4 ± 0.35	96.2 ± 0.58	84.1 ± 3.12	83.6 ± 2.50	82.8 ± 3.07	83.1 ± 2.34	
	96.3 ± 0.34	95.7 ± 0.57	95.7 ± 0.40	95.8 ± 0.48	**90.2 ± 1.13**	90.0 ± 1.63	88.5 ± 2.10	88.9 ± 1.46	
	96.6 ± 0.39	**96.6 ± 0.32**	96.3 ± 0.48	96.3 ± 0.40	84.1 ± 2.14	83.3 ± 2.69	79.0 ± 2.22	75.9 ± 5.42	PMEM
	96.0 ± 0.58	96.0 ± 0.41	96.2 ± 0.31	96.4 ± 0.48	84.5 ± 2.25	85.6 ± 1.42	83.6 ± 2.93	81.8 ± 2.53	
	96.0 ± 0.22	96.1 ± 0.45	95.9 ± 0.52	95.8 ± 0.39	89.1 ± 2.47	90.3 ± 1.23	89.3 ± 2.43	87.4 ± 2.37	
	92.7 ± 0.91	85.1 ± 1.26	71.4 ± 1.02	67.4 ± 0.65	85.3 ± 1.57	83.5 ± 1.07	72.7 ± 2.04	68.1 ± 3.55	DT
15	96.4 ± 0.65	**96.5 ± 0.35**	96.5 ± 0.25	**96.6 ± 0.37**	82.1 ± 4.37	78.8 ± 4.42	81.7 ± 2.89	74.8 ± 4.96	PEM
	96.3 ± 0.34	96.1 ± 0.91	96.2 ± 0.28	96.3 ± 0.38	85.1 ± 2.69	87.2 ± 2.26	83.1 ± 2.72	84.3 ± 2.68	
	96.3 ± 0.32	95.8 ± 0.59	95.8 ± 0.46	96.0 ± 0.47	89.3 ± 2.40	**89.9 ± 1.28**	89.6 ± 1.67	87.4 ± 2.60	
	96.4 ± 0.46	**96.5 ± 0.32**	96.2 ± 0.50	**96.6 ± 0.29**	80.8 ± 3.55	83.0 ± 3.60	78.9 ± 6.65	76.3 ± 4.97	NPEM
	96.4 ± 0.45	95.9 ± 0.57	96.2 ± 0.37	96.5 ± 0.48	84.4 ± 2.54	84.8 ± 1.97	83.5 ± 2.33	83.1 ± 4.67	
	96.2 ± 0.44	95.8 ± 0.55	95.8 ± 0.52	95.9 ± 0.44	**89.7 ± 1.87**	89.0 ± 2.03	87.1 ± 2.94	85.6 ± 3.50	
	96.5 ± 0.45	95.9 ± 0.86	**96.7 ± 0.36**	96.3 ± 0.37	81.5 ± 3.85	81.4 ± 2.22	76.7 ± 5.17	75.6 ± 3.42	MCEM
	96.4 ± 0.47	96.4 ± 0.42	96.5 ± 0.29	96.2 ± 0.42	85.3 ± 1.76	83.0 ± 3.72	83.4 ± 2.33	83.5 ± 2.76	
	95.9 ± 0.35	96.1 ± 0.40	95.9 ± 0.48	96.0 ± 0.42	89.4 ± 1.87	89.0 ± 1.99	88.6 ± 1.76	**87.8 ± 2.64**	
	96.1 ± 0.43	96.3 ± 0.73	96.6 ± 0.37	96.3 ± 0.32	83.9 ± 4.01	81.5 ± 5.31	81.1 ± 3.46	76.5 ± 7.06	PMEM
	96.3 ± 0.52	96.4 ± 0.37	96.3 ± 0.44	96.1 ± 0.52	84.4 ± 2.08	83.7 ± 1.86	83.2 ± 1.76	83.3 ± 2.08	
	95.7 ± 0.78	96.1 ± 0.29	95.9 ± 0.33	96.0 ± 0.42	88.6 ± 3.35	89.4 ± 1.73	**90.1 ± 2.04**	86.5 ± 2.82	
	90.4 ± 0.65	80.9 ± 1.20	71.3 ± 1.13	67.8 ± 0.75	81.2 ± 2.14	74.2 ± 1.63	64.7 ± 1.66	57.9 ± 1.91	DT
50	96.4 ± 0.32	96.4 ± 0.31	**96.3 ± 0.43**	96.1 ± 0.77	82.7 ± 1.65	82.8 ± 2.15	76.6 ± 5.62	75.2 ± 2.94	MCEM
	96.1 ± 0.58	96.0 ± 0.41	96.1 ± 0.39	96.0 ± 0.53	84.9 ± 1.85	84.5 ± 1.93	84.8 ± 2.79	82.4 ± 2.52	
	96.1 ± 0.47	96.1 ± 0.47	96.0 ± 0.58	95.3 ± 0.52	88.1 ± 1.48	**90.0 ± 0.52**	87.8 ± 2.41	85.5 ± 2.99	
	96.6 ± 0.33	**96.5 ± 0.41**	**96.3 ± 0.58**	**96.3 ± 0.44**	84.2 ± 4.68	81.8 ± 2.43	78.5 ± 4.29	73.8 ± 3.41	PMEM
	96.1 ± 0.47	96.1 ± 0.49	95.9 ± 0.55	95.5 ± 0.44	85.2 ± 1.33	83.9 ± 2.18	84.1 ± 2.79	80.9 ± 2.93	
	95.9 ± 0.28	96.1 ± 0.34	95.3 ± 0.47	93.4 ± 0.95	**88.8 ± 1.21**	89.0 ± 1.66	**89.6 ± 2.37**	**86.5 ± 2.21**	
	86.6 ± 1.13	80.8 ± 1.70	71.1 ± 1.11	67.5 ± 0.45	82.1 ± 1.43	72.8 ± 2.51	61.9 ± 3.53	56.1 ± 1.87	DT
	BCW				Ionosphere				

Table 4.3. Comparison of our proposal with Musicant et al. [128] best method, Decision Trees. For each version of our method and each LLP scenario, three rows are shown, representing a different type of Bayesian network classifier: NB, TAN and 2DB (from top to bottom). Results are shown in terms of accuracy and associated standard deviation, evaluated over 3 UCI [54] datasets for increasing bag size (N_i, vertical axis) and entropy (MLH, horizontal axis) using a 10×5 fold CV.

proposal (Structural EM and MCMC-specific parameters), the default values previously exposed have been used.

With the implementation of their method, we can use our aggregation procedure to simulate different LLP scenarios. As explained before, although the aggregation method of Musicant [128] is similar, our method covers the spectrum of LLP scenarios of different complexity in a more complete manner. In this way, we use bag size values $N_i = \{3, 7, 15, 30, 50\}$ and MLH entropy values $\text{MLH} = \{0.0, 0.25, 0.5, 0.75\}$. The four versions of our proposal and the Musicant et al. [128] method that learns DT classifiers have been evaluated over the three UCI [54] datasets that are used in the exposed paper. For each experiment, the results of our methods learning NB, TAN and 2DB classifiers are presented in terms of accuracy and associated standard deviation in Table 4.3.

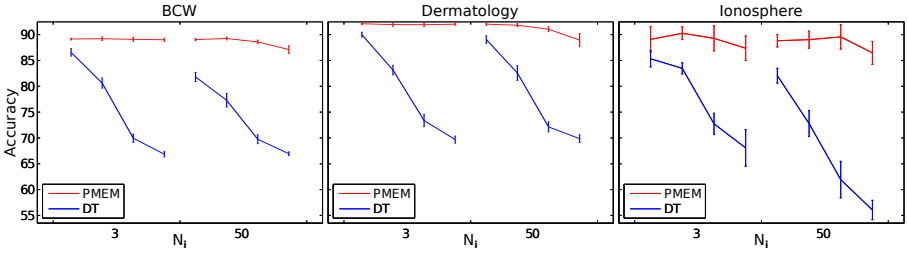


Fig. 4.6. Comparison over three datasets of the degradation of our PMEM version learning 2-dependence Bayesian network classifiers and the proposal of Musicant et al. [128] that learns decision trees (DT). In the x-axis, 8 LLP scenarios are represented, with $N_i = \{3, 50\}$ and MLH $= \{0.0, 0.25, 0.5, 0.75\}$. Mean accuracies and associated standard deviation over a 10×5 fold CV are shown.

Due to space constraints, only a representative subset of the experiments is shown in Table 4.3 (the complete comparison can be observed in the webpage associated with this study). As it can be seen in the table, our methods are able to learn Bayesian network classifiers which are, in the three domains and in all the LLP scenarios, more accurate than the DT method of Musicant et al. [128]. The classifiers learnt using the DT method of Musicant et al. [128] undergo a higher degradation (in terms of accuracy) than the classifiers learnt by any of the four versions of our proposal as the uncertainty of the LLP scenario increases.

With the objective of highlighting this behavior, Figure 4.6 reflects the loss in terms of accuracy of the classifiers learnt by the DT method [128] and that of the 2DB classifiers learnt with our PMEM method in extreme LLP scenarios (the lowest –$N_i = 3$ and MLH $= 0.0$– and the highest –$N_i = 50$ and MLH $= 0.75$– uncertain scenarios are displayed). On one hand, in the Dermatology and Breast Cancer Wisconsin domains the DT classifiers show an accuracy degradation of more than 25 points in percentage terms. Similarly, in the third domain (Ionosphere), DT classifiers lose more than 30 percentage points. On the other hand, 2DB classifiers learnt with PMEM show a degradation of 3-4 points in the three domains. Thus, the results show that the loss of accuracy between these extreme scenarios is much smaller when our method is used, which indicates that our proposal deals with the uncertainty of LLP problem in a better way than the DT of Musicant et al. [128]; or from the opposite point of view, it can be considered that PMEM efficiently takes advantage of the extra information given by the label proportions.

4.5 Conclusions and future work

In this chapter we have proposed four competitive versions of a Structural EM method to learn Bayesian network classifiers for a classification problem where

the only information of supervision provided consists of label proportions associated with subsets of instances (LLP).

We have shown that the label proportions associated to the bags (groups of instances) provide relevant class information that can be used to learn more accurate classifiers. Specifically, our proposal shows a competitive behavior with respect to state-of-the-art techniques, as has been shown in the comparison with the most representative and influential LLP methods. Among the four versions of our method, a probabilistic version that performs exact calculations, PEM, shows the best results, but it is not scalable when the uncertainty of the problem grows. We have overcome this issue with the proposal of another probabilistic version, MCEM, that performs an approximation to the exact version, showing a good behaviour in situations that are unaffordable for PEM. The associated statistical tests do not show significant differences between both versions.

Finally, we propose the PMEM version, which combines the exact and approximate procedures in a method that only uses approximate reasoning (MCEM) when the exact approach (PEM) is unfeasible. It uses the MCMC-specific parameters to establish a threshold (burn-in, bi, plus number of samples for calculating estimations, s) for the maximum number of explored consistent completions. For future work it would be interesting to study the possibility of calculating automatically and for each bag individually the parameters of the MCMC procedure (burn-in and number of samples). This would imply a non-constant threshold in the maximum number of explored consistent completions for each bag in PMEM, so it would also be necessary to study the implications of this decision. More questions could be raised about this issue: would it be worth carrying out the exact approach when the number of consistent completions (s_i) is slightly larger than the current threshold ($bi+s$)? That is, if s_i exceeds the number of samples generated in the MCMC procedure ($bi + s$) by *a small number* of consistent completions, which approach should be applied? What should be considered "a small number" in this context?

For future work, it would also be interesting to follow the idea expressed by Kück and Freitas [99] regarding the "relaxation" of the notion of label proportions. That is, considering a problem with groups of instances where each group is provided with the *probabilities* that a randomly chosen instance of the bag belongs to each possible class label. Learning from this new framework would involve new challenges, although we consider that we could apply the knowledge acquired in this study as the new framework can be seen as a generalization of the LLP problem.

5

Learning from crows in multi-dimensional domains

In the introduction of this dissertation (Section 2.2), different reported factors that make the process of fully labeling a dataset impossible have been exposed. In the previous chapter, we analyzed a weakly supervised classification problem motivated by a real application that undergoes an untraceable process after which the label of each example cannot be certainly assessed. In this chapter we discuss a weakly supervised classification problem that involves a source of uncertainty of a completely different nature: the questionable reliability of the annotator who labels the examples.

Formally defined in Section 2.4.4, learning from crowds is a classification problem where the provided training instances are labeled by multiple (usually conflicting) annotators. Several platforms, such as the Amazon Mechanical Turk, provide an environment where a proposed labeling task is solved by many different volunteers. These platforms can be used to cheaply obtain the labels for a set of examples. Nevertheless, as the reliability of the volunteers is usually *questionable*, not one but many labelings are collected. In different scenarios of this problem, straightforward strategies show an astonishing performance. In this chapter, we characterize the crowd scenarios where these basic strategies show a good behavior. In parallel, we identify those scenarios where non-trivial methods for combining the multiple labels are expected to overcome straightforward strategies. In this context, we also extend the learning from crowds paradigm to the multi-dimensional classification domain (Section 2.4.4). By measuring the quality of the annotators, the presented EM-based method overcomes the lack of a fully reliable labeling for learning multi-dimensional Bayesian network classifiers (Section 3.4.4): As the expertise is identified and the contribution of the relevant annotators promoted, the model parameters are optimized. The good performance of our proposal is demonstrated through different sets of experiments.

5.1 Introduction

Supervised classification is a field of machine learning that develops techniques which try to replicate automatically the categorizing behavior of a classification problem of interest. Provided a set of examples together with certain category information (class labels), the inherent relationship between instances and class labels is deduced. The reliability of the provided training labeling is a strong assumption based on which most of the techniques that take part in the machine learning process have been developed (evaluation techniques, performance scores, learning methods, etc.).

Relaxing the reliability assumption, the partially supervised learning framework [155] deals with datasets in which the training examples are not completely/certainly labeled, mainly due to the difficulty and high cost of the expert labeling process. In the case of *learning from crowds* (CrL) [141], each instance of the provided dataset is labeled by several annotators of unknown trustfulness. As no *gold-standard* (i.e., a reliable class-membership information for the instances) is provided, both the learning and the evaluation have to be performed using the subjective labels of a crowd of mainly non-expert annotators.

Dealing with the low reliability that characterizes the data collected from a crowd is the main challenge of CrL. Many sources of noise that would harm the truthfulness of the provided labels have been modeled in different proposals of the related literature. However, extremely basic strategies to combine the labels provided by several annotators report excellent results in different crowd scenarios. In this study, we analyze the characteristics of these scenarios, and identify the rest of crowd scenarios where an alternative (more elaborate) methodology would outperform the basic approaches.

The interest in the CrL methodology has mainly been motivated by the easy and cheap access to data that the emerging information technologies facilitate. It is reasonable to apply this cheap labeling procedure to problems where the data collection process is expensive, such as the *multi-dimensional* (MD) framework [6, 148] where several class variables have to be labeled for each example. The difficulty of obtaining labeled (training) data in a MD problem relates to the set of class variables that have to be labeled for each example. Learning with completely labeled training data, the MD problem —which has also been studied as a specific instantiation of the more general *multi-target learning* framework [199]— has received ample attention, with different proposals as the multi-dimensional Bayesian network classifiers (MBC) [6].

Combining both CrL and MD frameworks, we present a novel methodology for learning MBCs using data labeled by a crowd of annotators. It recognizes the annotator reliability as a relevant source of noise, a standard consideration in the related literature [192, 141, 41, 42, 191, 30]. In this way, our method —based on the Expectation-Maximization (EM) strategy [45]— is able to model the expertise of the annotators of the crowd. The estimated truthfulness of

the annotators is used to calibrate their contribution to the learning process, promoting the labels provided by outstanding annotators.

The rest of the chapter is organized as follows. First of all, a review of partially supervised problems allows us to locate in the related literature the problem of interest, the *multi-dimensional learning from crowds* (MDCrL). Next, we characterize the crowd scenarios where the weak performance of basic strategies justifies the use of more complex methodologies. Then, our method for learning (the parameters of) MBCs from a crowd is presented, followed by a set of experiments that test it in complex crowd scenarios. Finally, some conclusions and future work are presented.

5.2 Related problems

Following the non-complete supervision idea of the semi-supervised learning paradigm (where only a subset of the training examples are labeled), over the past decades different *partially supervised learning* problems have been explored. Learning accurate classifiers remains the main objective of these classification problems, although the lack of a gold-standard prevents the use of standard learning methodologies, requiring the development of specific techniques. The more recently fashioned CrL problem is closely related with other partially supervised problems that have received significant attention for many decades.

For example, the *learning from multiple experts* framework [37, 167, 181] combines the labels of a fixed set of known experts. Its development has been closely related to the interest in real applications where the difficulty of labeling instances is high (e.g., medical diagnosis, CAD systems, etc.). Dawid and Skene [37] proposed an EM method that estimates, for each expert, the probability of confusing two class labels using an estimated gold-standard. Later, Smyth et al. [167] proposed using degrees of certainty associated with each provided label in order to focus the learning process on highly reliable instances. The main contribution of Wiebe et al. [193] was a procedure that searches for possible correlations between the labels of different experts. The self-consistency of the labels provided by an expert allows Valizadegan et al. [181] to identify labelers that introduce random noise, the least damaging kind of noise. The opposite strategy consists of learning a classifier from the annotations of each expert and, finally, combining all the models [112]. This strategy works only in those scenarios where all the annotators label a large set of the instances. In a CrL problem, where each (non-expert) annotator chooses the instances to label, a classifier learnt from the annotations of a *lazy* annotator could perform poorly.

Another characteristic that defines a CrL problem is the low reliability of the provided labels. As it cannot be assumed that the annotators are domain experts, the subjective labeling may involve incompleteness, impreciseness and/or incorrectness. The *learning from noisy data* problem [12, 202]

faces the same challenges in the case of a single labeler. Without the possibility of comparing with other points of view (annotators), different strategies have to be used to identify the wrong-labeled instances. Shanmugam and Breipohl [160], who proposed two effective strategies to correct the wrong labels, demonstrated that the degree of noise can be compensated with larger datasets. This idea was posteriorly reasserted by Lugosi [114] in his key study of sources of noise, where three kinds of error are identified: random error, external error (depends on the right label), and consistent error (depends on the instance). Lugosi proved that the first two kinds of error are asymptotic optimal (i.e., the error can be compensated with a larger dataset), whereas the latter can only be overcome with labels of different annotators (i.e., a multi-expert/crowds framework).

In the *learning from partial labels* problem [35], being unable to certainly label an instance, a set of labels is provided with the guarantee that the real label is in the provided set. As the different labels of the set cannot be attributed to independent sources (multiple annotators), the selection among the candidate labels is based on the consistency with other similar instances. Jin and Ghahramani [94] proposed an extended problem where a probability distribution over the set of candidates measures the confidence on each label. A CrL problem which has lost the information about the annotator that provided each label follows this formulation. Jin and Ghahramani used an EM method to assess the most-probable label for each instance while the classifier is learnt.

Combining ideas of two well studied problems of the state-of-the-art, the multiple labeling and the noisy/uncertain labels, the *learning from crowds* (CrL) problem has led to a new successful and useful learning paradigm [161, 168]. Snow et al. [168] measured the value of the contribution of non-expert annotators: the knowledge of a domain expert might be matched with the combined knowledge of four non-expert annotators. In another key study, Sheng et al. [161] compared the benefits of relabeling an instance making use of a different annotator (i.e., increasing the reliability of the labeling) with respect to obtaining new labeled instances (i.e., exploring the instance space). Restricted by a limited number of annotators, the authors defend that repeated labeling loses effectiveness as the reliability of the annotators drops.

In order to cope with the unreliability of the annotations, a vast majority of the state-of-the-art CrL techniques [192, 141, 41, 42, 191, 30] try to infer the expertise of the annotators, which later is used to calibrate their contribution in the learning process. Additionally, Dekel and Shamir [41] proposed a method to detect annotators which provide contrary information (evil teachers). The same authors [42] remove the labels of very unreliable annotators, which would not affect the performance of the learnt model in the case of a large dataset. Trying to cover different sources of noise, Whitehill et al. [192] proposed modeling the instance difficulty, and Welinder et al. [191] studied the annotator reliability from three points of view: competence, expertise and bias of their annotations. For the problem of evaluation, Cholleti et al. [30] proposed using iteratively each annotator as gold-standard for learning

different classifiers and finally combining them. However, probably the most general and challenging proposal was presented by Raykar et al. [141], who developed an EM method (adjustable to binary/multi-class/ordinal/regression problems) that iteratively estimates a set of reliability weights from an estimated gold-standard to induce a predictive model.

In their proposal, Raykar et al. [141] model the ability of each annotator to predict each *class label* individually, an idea that applies to the multi-label and multi-dimensional domains, where different categories are assessed for each instance. As far as we know, the few studies that consider multi-label problems labeled by many annotators do not assess the reliability of the annotators. An usual strategy [159, 175, 14] reduces the framework to deal with a *multi-label problem with weak labels*: for each instance, annotators only provide the labels which they are sure about (i.e., a subset of its real set of labels). Eventually, the most similar problem to our MDCrL framework was proposed by Younes et al. [197]: a multi-label problem annotated by multiple noisy annotators. Given a subset with the assigned (or dismissed) labels for each instance, the method has to look for a complete assignment of labels which fulfills the provided incomplete labeling. With a more unorganized provision of the data, the *folksonomy* (or *social tagging*) problem [176] tries to learn from instances labeled by many (unidentified) annotators who tag with an unrestricted set of terms (or labels). The unlimited number of labels per instance and the unavailability of a close set of possible class labels are probably the main challenges of a problem with many real applications (e.g., recommender systems of the collaborative Web).

5.3 Exploring different crowd scenarios

The classic supervised classification finds in the labels provided by a trustworthy domain expert the reliability required to learn accurate classifiers. The CrL paradigm seeks this reliability in the group knowledge of a crowd of t annotators. Before a new method is proposed to deal with this kind of data, it is interesting to pay attention to the particularities of the CrL problem. In a considerable variety of crowd scenarios, simple methodologies for combining the group knowledge show a good behavior. In this section, we shed some light on the advantages of two well-known basic approaches and describe their favorable scenarios (in general, well-informed scenarios where the reliability of some annotators is high and/or the number of annotators is large). Once these favorable scenarios are characterized, the new method could focus on the remaining scenarios and try to extract all the useful information from the crowd knowledge.

Assuming that all the annotators are not equally reliable, this work has been developed over the main hypothesis that the annotators with expert knowledge can be identified among the crowd and used to improve the accuracy of the learnt classifiers. For the sake of simplicity, the different reasonings

of this section have been carried out for the basic scenario where the annotators label a *single binary class variable*. Accordingly, the labels provided by annotator A_a have a specific reliability rate r_a, which means that with probability $(1 - r_a)$ the provided label is wrong. We will consider that an annotator A_e is a domain *expert* if their real reliability rate r_e is larger than the reliability rate r_a of any other regular annotator (or *novice*) A_a. However, we assume that there is *one (and only one) domain expert* in the crowd of annotators, unless otherwise indicated. The presence of an only expert among the crowd simplifies the identification and measurement of the relevance that their contribution reaches during the learning process, with respect to the rest of (novice) annotators.

5.3.1 The most-voted label strategy

The straightforward strategy in problems with multiple annotators is known as *majority voting* (MV). For each example $\boldsymbol{x}$, taking into account the labels provided by the different annotators, the most frequent class label is considered the actual label:

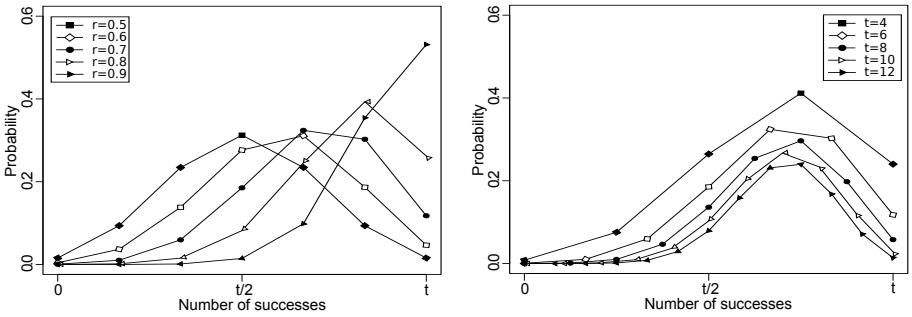
$$MV(\boldsymbol{l}) = \underset{c \in \mathcal{C}}{\operatorname{argmax}} \sum_{a=1}^{t} \mathbb{I}[l_a = c]$$

where $\mathbb{I}[condition]$ returns 1 if *condition* is true and 0 otherwise, and $\boldsymbol{l}$ is a t-tuple that saves the label l_a provided by each annotator A_a. This procedure outputs a new completely supervised dataset D where a specific (the most-voted) class label is assigned to each instance, $(\boldsymbol{x}, MV(\boldsymbol{l}))$. In order to learn a predictive model from this completed dataset, any standard supervised classification technique can be used.

For the sake of clarity, let us assume that *all* the t annotators have the same reliability rate, r_a. In this way, the probability of having Y annotators providing the real label for a single binary class variable follows a Binomial distribution with t repetitions and success probability r_a, $Y \sim B(t, r_a)$. Thus, the probability of the real label being the most-voted one is:

$$p_S^{MV} = 1 - F(t/2; t, r_a) = \begin{cases} \displaystyle\sum_{i=\lceil t/2 \rceil}^{t} \binom{t}{i} \cdot r_a^i \cdot (1 - r_a)^{t-i} & , t \text{ is odd} \\ \dfrac{1}{2}\binom{t}{t/2} \cdot r_a^{t/2} \cdot (1 - r_a)^{t/2} + & \\ + \displaystyle\sum_{i=t/2+1}^{t} \binom{t}{i} \cdot r_a^i \cdot (1 - r_a)^{t-i} & , t \text{ is even} \end{cases} \tag{5.1}$$

Understanding p_S^{MV} as the reliability of the labels obtained with MV, it can be shown that these are more (or equally) reliable than the labels provided by



(a) Fixed $t = 6$, increasing r (b) Fixed $r = 0.7$, increasing t

Fig. 5.1. Evolution of the probability mass function of the Binomial distribution $B(t, r)$ as the value of *one* of its parameters —t is the number of trials and r the probability of success— increases. Fixing the value of a parameter in both figures, each lines represents the function for a specific value of the other parameter.

single annotators when $r_a > 0.5$ and $t > 2$. In fact, as observed in Figure 5.1, the probability of guessing the real label increases as one or both parameters of the Binomial distribution $B(t, r_a)$ rise. Both Figure 5.1(a) and 5.1(b) show the same trend: as the value of the specific parameter increases, the area under the mass function curve shifts to the second half of the figure (where the number of annotators who are right is $Y > t/2$). That is, the scenarios where the majority of annotators provides the right label are more probable. In practice, this approach is a very interesting solution in the case of rich information of supervision (large t and/or r_a) because of the high reliability of the MV labels ($p_S^{MV} \to 1$).

5.3.2 Having found the expert: do we need a crowd? The expert selection strategy

The vast majority of methods proposed in the CrL literature [192, 141, 41, 42, 191, 30] look for domain experts among the crowd of annotators in order to give more relevance to their annotations during the learning process. In different ways, all of them look for (different concepts of) expertise and propose successful approaches to integrate it in the learning process. But, if we have found a domain expert among the crowd, can the rest of the annotators contribute or should we ignore them? In order to answer this key question, let us assume that we are able to identify an expert among a crowd of novice labelers.

Contrary to what intuition says, identifying expertise does not suppose hard work if there is at least some basic knowledge among the crowd. This ability is observed in basic measures to estimate the reliability of the annota-

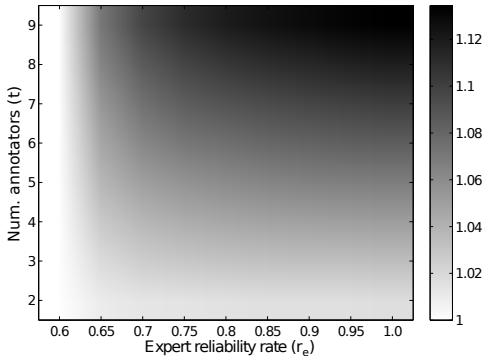


Fig. 5.2. Expected relevance rate of a single expert annotator with respect to a novice ($E[w^e]/E[w^a]$, Eq. 5.3) as the reliability of the expert (r_e) and the total number of annotators (t) are increased. The reliability rate of the novices is fixed ($r_a = 0.6$).

tors: e.g., the mean consensus of the rest of annotators on the labels provided by an annotator A_a in a dataset of N instances [168],

$$w^a = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{t-1}\sum_{a'\neq a}\mathbb{I}[l^i_{a'} = l^i_a] \tag{5.2}$$

Assuming that the real reliability rates of the annotators (r_a, $a \in \{1,\ldots,t\}$) are available, the expected value of this estimated reliability weight (Eq. 5.2) in the case of a binary class variable is,

$$E[w^a] = \frac{1}{t-1}\left(r_a\sum_{a\neq a'} r_{a'} + (1-r_a)\sum_{a\neq a'}(1-r_{a'})\right) \tag{5.3}$$

Let us assume a crowd scenario where there is a domain expert A_e and the rest of annotators are novices with the same reliability rate (r_a), such that $r_e > r_a$. It can be shown that Eq. 5.2 is a measure which is able to detect the expertise of the outstanding annotator(s) if the reliability rate of the rest of annotators (novices) is $r_a > 0.5$. Using Equation 5.3, the rate of the expected weight of an expert annotator A_e with respect to the expected weight of a novice A_e of the crowd can be easily calculated (i.e., $E[w^e]/E[w^a]$). Fig. 5.2 shows this *expected relevance* measure as, given a fixed crowd reliability rate ($r_a = 0.6$), the number of annotators (t) and expert reliability rate (r_e) increase. The dark area in the upper-right corner of the figure indicates that the expert relevance would increase with both r_e and t. While a larger t adds strength to the process with extra viewpoints of additional annotators, a larger r_e involves a higher probability of coincidences with other annotators. Although not shown in Figure 5.2, Eq. 5.2 requires a minimum crowd reliability ($r_a > 0.5$) in order to identify the expert correctly. Otherwise, the

real expert could be considered as a saboteur ($r_a < 0.5$) or could just remain undetected ($r_a = 0.5$).

Under the realistic assumption that the real expert annotator can be found, a simple solution is to consider his/her labels the correct ones and build a new completely labeled dataset based on his/her predictions. As both this *expert selection* (ES) approach and the MV technique select a single and specific label assignment for instances annotated by a crowd, their reliability can be compared. Let us consider the probability of obtaining the right label for both approaches in a crowd scenario where there is a single domain expert (with reliability rate, r_e) and the rest are novices (with the same reliability rate, r_a). In ES, once the expert has been found, the reliability rate r_e can be considered as the probability of being right; for the MV approach, Eq. 5.1 can be extended with individual reliability rates ($r_e \neq r_a$) to provide the probability of the most-voted label being the real one. Using all this information, it is possible to calculate the probability of both techniques providing the same label; i.e., the probability of the label provided by the expert also being the most-voted one among all the annotators. Similarly, both approaches disagree in those cases where the crowd contradicts the opinion of the expert: i.e., the most-voted label does not coincide with the label provided by the expert. But contradicting the expert opinion is not always a wrong idea as the expert fails with probability $(1 - r_e)$. In this way, two situations of disagreement between both techniques can be identified: (a) the crowd agrees the wrong label against the correct opinion of the expert, which happens with probability,

$$p_{defeat} = r_e \cdot p(Y \leq \frac{t-1}{2} - 1) = r_e \cdot \left(\sum_{i=0}^{\lfloor (t-1)/2 - 1 \rfloor} \binom{t-1}{i} \cdot r_a^i \cdot (1 - r_a)^{t-(1+i)} \right)$$

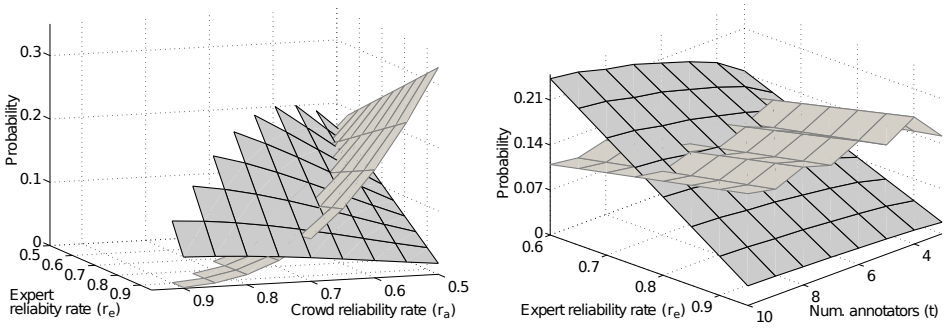
and (b) the crowd prevents a wrong labeling of the expert, with probability,

$$\begin{aligned} p_{save} &= (1 - r_e) \cdot \left(1 - p(Y \leq \left\lceil \frac{t-1}{2} \right\rceil) \right) \\ &= (1 - r_e) \cdot \left(1 - \sum_{i=0}^{\lceil (t-1)/2 \rceil} \binom{t-1}{i} \cdot r_a^i \cdot (1 - r_a)^{t-(1+i)} \right) \end{aligned}$$

If the number of annotators t is even, as ties in MV are solved randomly, both probabilities p_{defeat} and p_{save} have to be updated with the corresponding tie-solving probability,

$$\begin{aligned} p_{defeat} &= p_{defeat} + \frac{1}{2} r_e \cdot \binom{t-1}{t/2} \cdot r_a^{t/2-1} \cdot (1 - r_a)^{t/2} \\ p_{save} &= p_{save} + \frac{1}{2}(1 - r_e) \cdot \binom{t-1}{t/2} \cdot (1 - r_a)^{t/2-1} \cdot r_a^{t/2} \end{aligned}$$

To conclude, the application of a MV strategy (instead of ES) is justified in those scenarios where the probability of preventing the wrong opinion of



(a) Fixed $t = 6$, increasing r_a and r_e

(b) Fixed $r_a = 0.6$, increasing r_e and t

Fig. 5.3. Probability of the crowd correcting a wrong label of the expert (p_{save}, dark surface) against the probability of the crowd damaging a correct annotation of the expert (p_{defeat}, light surface). Both figures depend on the number of annotators in the crowd (t), the reliability rates of the crowd annotators (r_a) and the expert (r_e).

the expert is larger than the probability of imposing a crowd-predicted wrong label despite the correct opinion of the expert ($p_{save} > p_{defeat}$). Figure 5.3 shows graphically the areas of domination of each approach, and the boundary between them. According to the figure, the knowledge of a single expert is enough when its reliability is significantly larger than the crowd reliability ($r_e >> r_a$). Similarly, as the number of annotators (t) increases, the reliability of MV rises and the expert should be considered alone only if his/her reliability is outstanding.

5.3.3 Scenarios for improvement: beyond basic crowd learning strategies

While MV is the best strategy when there are a large number of both high-reliable annotators and instances, ES overcomes MV when there are few annotators and one of them is a domain expert with an outstanding reliability. The crowd scenarios that are not definitely solved with basic techniques are those with data scarcity: few instances and labelers; absence of outstanding experts. This general description characterizes the space of crowd scenarios where a new method could overcome the basic alternatives studied.

A completely different point of view consists of using all the labels provided by all the annotators in the learning process. It can be shown that, estimating specific reliability weights per annotator (in a similar way to Eq. 5.2), the degree of noise introduced in the learning process is reduced as the estimated individual weights match the real reliability rates ($w^a = r_a, a \in \{1, \ldots, t\}$) of the annotators. In the extreme scenario where only an expert annotator is considered (fully) reliable ($w^e = 1$ and $\sum_{a \neq e} w^a = 0$), although the noise is

minimized, the diversity of opinions is lost (i.e., the expert selection strategy). In this way, the new proposal will take advantage of the learning process to assign a fair influence to each annotator, with the objective of promoting the labels provided by the most reliable annotators over the labels of novices.

5.4 A method for learning MBCs from a crowd of annotators

We have applied the CrL paradigm to the multi-dimensional classification framework. As formally defined in Section 2.1.3, in a multi-dimensional classification problem there are multiple class variables and each example is assigned to a class label for each class variable. The MD framework has been formally extended to the CrL paradigm (MDCrL) in Section 2.4.4: for each example and class variable, annotators provide their labels according to their subjective opinion. Thus, the vector of reliability weights discussed in the previous section for the unidimensional scenario is transformed into the multi-dimensional domains by means of a matrix of reliability weights (w_k^a, with $a \in \{1, \ldots, t\}$ and $k \in \{1, \ldots, m\}$) which describes the expertise of each annotator A_a in each class variable C_k. Note that multi-label (ML) learning can be considered as a particular case of the multi-dimensional framework [6, 148, 199]. A ML problem which has a class variable with $|\mathcal{C}|$ possible labels can be represented as a MD problem with $|\mathcal{C}|$ class variables, where the presence/absence of each class label in the ML problem is modeled by a *binary* class variable in the equivalent MD problem. Throughout this transformation, all the reasoning presented for the MD framework also applies for ML problems.

In this section, we present a general method that learns multi-dimensional Bayesian network classifiers (Section 3.4.4) from the data collected from a multi-dimensional classification problem which has been annotated by a crowd. For this study, a specific type of MBCs has been selected (Figure 3.4). It provides a fixed structure which does not need to be learnt —it consists of m fixed NB structures and a tree among class variables which, in practice, has been fixed. As the number of parameters of these models sharply increases with the number of class variables, their learning process becomes unfeasible in problems with many classes. However, the proposed procedure works in the same way for other kinds of MBCs and, in this way, several methods proposed in the related literature which are able to learn MBCs with less complex structures from data completely labeled [6, 148] could be adapted. In our preliminary work [81], the *wrapper* algorithm proposed by Bielza et al. [6] was adapted to the MDCrL framework and embedded in a Structural EM [57] method; this strategy adds an external structural learning loop to the parametric EM procedure. In this work, we skip the structural learning step through the use of these MBCs of fixed structure, which allows us to focus all the efforts on the main objective: demonstrating the enhanced performance

Algorithm 3 Pseudo-code of our EM method

```
1: procedure EM(D, maxIt, ε)
2:     j = 0
3:     ŵ ← initializeWeights(D)
4:     θ^(j) ← parametricLearning(D,ŵ)
5:     repeat
6:         j = j + 1
7:         ŵ ← calculateWeights(D,θ^(j−1))
8:         θ^(j) ← parametricLearning(D,ŵ)
9:     until (diff(θ^(j), θ^(j−1)) < ε) Or (j = maxIt)
10:    return θ^(j)
11: end procedure
```

of the learnt models as the expert knowledge of outstanding annotators is detected and properly promoted.

Therefore, our method is based on the standard EM strategy (Section 3.3.2), which allows us to combine the learning of the model parameters and the estimation of the reliability weights of the annotators. It takes advantage of all the information of supervision available to learn classifiers, especially in complex crowd scenarios as described in the previous section (they can be easily extended to the multi-dimensional framework). To do this, at each iteration our proposal estimates from the data a set of reliability weights, $\boldsymbol{w}$, which specifies the reliability of the annotators. In parallel, model parameters are estimated from this kind of data (repeatedly annotated examples and annotator reliability weights) by means of a modification of the standard frequency-counting procedure.

The proposed method (its pseudocode is shown in Algorithm 3) first calculates the reliability weights of the annotators according to Equation 5.2, for each class label independently. Next, it iterates the classic EM procedure. First, the model parameters are estimated from the multiple and weighted annotations. During this step, the per-class reliability weights of the annotators have to be combined. In this work, two combination schemes have been considered: the *addition* and the *product* of per-class weights. Secondly, the reliability weights of each annotator in each class variable are re-estimated using the current fit of the model. To do this, two different approaches have been considered: (1) the *accuracy* of the labels of the annotator with respect to the predictions of the model, and (2) the *probability* according to the model of the labels provided by the annotator. Both steps are iterated until convergence.

Thus, the method can use four different configurations depending on the approach selected to perform the weight combination and the weight estimation steps. In the following subsections, a detail description of the method is provided together with the description of the alternative approaches for each step.

5.4.1 Combining per-class reliability weights for parameter estimation in learning from crowds

The information of supervision available in a MDCrL problem consists of multiple annotations for each instance and class variable. We have adapted the standard parameter estimation procedure to collect frequency counts from this kind of data, using the reliability weights w_k^a ($a \in \{1, \ldots, t\}$ and $k \in \{1, \ldots, m\}$) for performing an informed aggregation of the different contributions.

Applying the chain rule to our MBCs (Section 3.4.4), the joint probability of an example $(\boldsymbol{x}, \boldsymbol{c})$ factorizes as,

$$p(\boldsymbol{x}, \boldsymbol{c}) = p(c_r) \cdot \prod_{k \neq r} p(c_k | c_l) \prod_{j=1}^{n} p(x_j | \boldsymbol{c})$$

As usual, each conditional probability can be calculated from the dataset by means of frequency counts,

$$p(u_1 | u_2, \ldots, u_{|\boldsymbol{u}|}) = N(u_1, \ldots, u_{|\boldsymbol{u}|}) / N(u_2, \ldots, u_{|\boldsymbol{u}|})$$

where $(u_1, \ldots, u_{|\boldsymbol{u}|}) \subset (\boldsymbol{x}, \boldsymbol{c})$ is the instantiation of a set of variables $\boldsymbol{U} = \{U_1, \ldots, U_i\} \subseteq \boldsymbol{V} = (\boldsymbol{X}, \boldsymbol{C}) = (X_1, \ldots, X_n, C_1, \ldots, C_m)$ and $N(\boldsymbol{u})$ represents the corresponding counts. The Laplace estimator, a classic additive smoothing technique that prevents assigning zero or one probabilities, is implemented. By means of an adaptation of the counting procedure, $N(\cdot)$, the parametric learning process integrates the multiple and weighted labels,

$$N(\boldsymbol{u}) = \sum_{a=1}^{t} w_a^{\downarrow \boldsymbol{u}} \sum_{\boldsymbol{y} \in \mathcal{X}(D, A_a)} \mathbb{I}[y_{[U_1]} = u_1, \ldots, y_{[U_{|\boldsymbol{u}|}]} = u_{|\boldsymbol{u}|}]$$

where $[U_j]$ indicates the index of the variable $U_j \in \boldsymbol{U}$ in the original set of variables $\boldsymbol{V}$, and $\mathcal{X}(D, A_a)$ represents the set of instances of the original dataset D labeled by annotator A_a. Finally, $w_a^{\downarrow \boldsymbol{u}}$ is the *combined reliability weight* assigned to annotator A_a when just the variables $\boldsymbol{U} \subseteq \boldsymbol{V}$ are considered. In this stage, a strategy has to be chosen in order to combine the per-class reliability weights of all the annotators (w_k^a), constrained to $\sum_{a=1}^{t} w_a^{\downarrow \boldsymbol{u}} = 1$. In this work, two simple techniques that reach different orders of expert relevance are discussed: product or addition based combinations.

On the one hand, the combined reliability weight $w_a^{\downarrow \boldsymbol{u}}$ of annotator A_a can be calculated as the *product* of the per-class weights, taking into account only those class variables in $\boldsymbol{U}$,

$$w_a^{\downarrow \boldsymbol{u}} = \frac{\prod_{C_k \in \boldsymbol{U} \cap \boldsymbol{C}} w_k^a}{\sum_{a'=1}^{t} \prod_{C_k \in \boldsymbol{U} \cap \boldsymbol{C}} w_k^{a'}}$$

where $\boldsymbol{C}$ is the subset of class variables in $\boldsymbol{V}$ and $(\boldsymbol{U} \cap \boldsymbol{C})$ selects the class variables contained in $\boldsymbol{U}$. On the other hand, $w_a^{\downarrow \boldsymbol{u}}$ can be calculated as the *addition* of the per-class weights,

$$w_a^{\downarrow \boldsymbol{u}} = \frac{\sum_{C_k \in \boldsymbol{U} \cap \boldsymbol{C}} w_k^a}{\sum_{a'=1}^{t} \sum_{C_k \in \boldsymbol{U} \cap \boldsymbol{C}} w_k^{a'}}$$

In this study, we assume that the annotators may be experts in a subset of the class variables, and novices in the rest of them. Due to the combination of the per-class weights, the larger expert weight w_k^e of the annotator A_e in class C_k can be neutralized if his/her expertise is limited to this class variable or, alternatively, over-promoted if A_e is expert in a large subset of class variables. Let us assume a simplistic scenario where our method only assigns two types of weights (large —expert— w^l, and small —novice— w^s) and identifies an expert A_e that stands out in cc class variables at the same time. In the case of a count that involves all the class variables ($\boldsymbol{U} \cap \boldsymbol{C} = \boldsymbol{C}$), the *relevance rate* that reaches A_e with respect to other completely novice annotator A_a using the product-based combination approach is

$$\frac{w_e^{\downarrow \boldsymbol{u}}}{w_a^{\downarrow \boldsymbol{u}}} = \frac{\prod_{C_k \in \boldsymbol{U} \cap \boldsymbol{C}} w_k^e}{\prod_{C_k \in \boldsymbol{U} \cap \boldsymbol{C}} w_k^a} = \frac{(w^l)^{cc} \cdot (w^s)^{(m-cc)}}{(w^s)^m} = \left(\frac{w^l}{w^s}\right)^{cc} = h^{cc}$$

where $h = \frac{w^l}{w^s} > 1$ is the relevance that an expert can reach in a single class variable (i.e., how much the expert labels will be over-promoted). Similarly, when the addition-based approach is used,

$$\begin{aligned} \frac{w_e^{\downarrow \boldsymbol{u}}}{w_a^{\downarrow \boldsymbol{u}}} &= \frac{\sum_{C_k \in \boldsymbol{U} \cap \boldsymbol{C}} w_k^e}{\sum_{C_k \in \boldsymbol{U} \cap \boldsymbol{C}} w_k^a} = \frac{cc \cdot w^l + (m - cc) \cdot w^s}{m \cdot w^s} \\ &= \frac{cc \cdot h \cdot w^s + (m - cc) \cdot w^s}{m \cdot w^s} = \frac{cc}{m} \cdot h + \frac{m - cc}{m} \end{aligned}$$

This second approach assigns to the expert a final relevance equivalent to the mean per-class relevance (where the minimum value is 1 —irrelevant—, and h is the maximum value). However, in the product-based combination strategy, the relevance in each class variable is multiplied, which could finally make an expert annotator extremely relevant (depending on h and, mainly, on cc).

As explained before, our method normalizes the weights of the annotators to sum up to 1 ($\sum_{a=1}^{t} w_a^{\downarrow \boldsymbol{u}} = 1$). This calculation, which depends on the number of annotators, determines the expert relevance too. As the weight of an expert $w_e^{\downarrow \boldsymbol{u}}$ rarely doubles the weight of a novice ($w_a^{\downarrow \boldsymbol{u}} < w_e^{\downarrow \boldsymbol{u}} < 2w_a^{\downarrow \boldsymbol{u}}$), even after product-based per-class combination, the influence of the expert is reduced to tilt the decision towards their provided labels in the case of a tie. As observed in Figure 5.1(b), the probability of a tie decreases as the number of annotators increases, which reduces the expected influence of the expert.

5.4.2 Estimation of the reliability weights of the annotators

Without any other information, the degree of knowledge of the annotators is calibrated according to the labels that they provide. To do this, the provided labels have to be somehow evaluated. The best solution would be a comparison with a gold-standard. However, the lack of this gold-standard makes us consider alternative procedures, such as the direct comparison with other annotators carried out in the initialization of the method (Eq. 5.2). Once a first fit of the model $\mathbb{M}$ has been learnt, it can be used to estimate a gold-standard. Different gold-standards can be estimated according to the way in which the model is used. In this work, we explore two approaches which estimate per-class reliability weights that reach different degrees of expert relevance. In a first *accuracy-based* approach, the model is used to obtain the joint classification of each instance: $\hat{\boldsymbol{c}} = \text{argmax}_{\boldsymbol{c}\in\mathcal{C}}\, p_{\mathbb{M}}(\boldsymbol{x}, \boldsymbol{c})$. Then, each weight w_k^a is calculated as the mean accuracy of the annotator A_a in the class variable C_k, using the configurations predicted by the model, $\hat{\boldsymbol{c}}^i$, as a gold-standard:

$$w_k^a = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}[L_{ak}^i = \hat{c}_k^i]$$

The performance of this evaluation approach is limited by the use of the MBCs. As any probabilistic classifier, the classification error of a MBC model $\mathbb{M}$ (at least as large as the Bayes error) is:

$$Error(\mathbb{M}) = \sum_{\boldsymbol{x}\in\mathcal{X}} p_{\mathbb{G}}(\boldsymbol{x}) \cdot [1 - p_{\mathbb{G}}(\hat{\boldsymbol{c}}|\boldsymbol{x})]$$

where $\mathbb{G}$ is the generative model and $\hat{\boldsymbol{c}}$ is the joint class labeling provided by the classifier model $\mathbb{M}$ for sample $\boldsymbol{x}$. Moreover, in a multi-dimensional domain, the classification error of the model taking into account just a class variable C_k can be calculated as:

$$Error_k(\mathbb{M}) = \sum_{\boldsymbol{x}\in\mathcal{X}} p_{\mathbb{G}}(\boldsymbol{x}) \cdot [1 - p_{\mathbb{G}}(\hat{c_k}|\boldsymbol{x})]$$

where $p_{\mathbb{G}}(c_k|\boldsymbol{x}) = \sum_{\boldsymbol{c}'\in\mathcal{C}} p_{\mathbb{G}}(\boldsymbol{c}'|\boldsymbol{x}) \cdot \mathbb{I}[c'_k = c_k]$ is the marginal probability of class variable C_k. As we just consider labelers which provide wrong labels randomly (random noise), this measure gives an upper bound on the reliability weights calculated by means of this accuracy-based technique. For an annotator with a reliability rate r_a, the expected weight is,

$$E[w_k^a] = (1 - Error_k(\mathbb{M})) \cdot r_a + Error_k(\mathbb{M}) \cdot (1 - r_a)$$

In our second approach, the weight w_k^a is estimated using the *probability* of the labels provided by annotator A_a for class variable C_k. The probability of a class label given an instance $\boldsymbol{x}^i$ is calculated as its marginal probability according to the model $\mathbb{M}$,

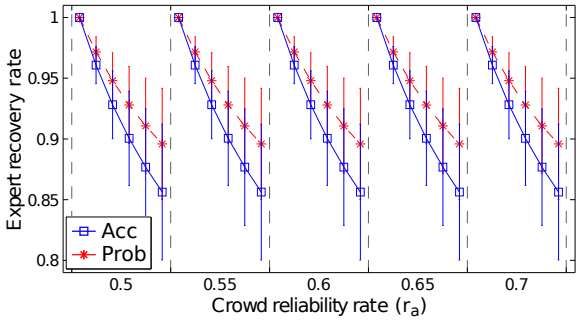


Fig. 5.4. Comparison of the ability of both weight estimation techniques to recover the *real* expert relevance: $h_{est}/h_{real} = (E[w_k^e] \cdot r_a)/(E[w_k^a] \cdot r_e)$. The mean value and associated standard deviation of this recovery rate calculated for 10 randomly generated MBCs is shown. The vertical divisions separate points with different crowd reliability values ($r_a = \{0.5, 0.55, \ldots, 0.7\}$). In each division, a line links points with increasing expert reliability ($r_e = \{r_a, r_a + 0.05, \ldots, r_a + 0.25\}$); i.e., different *real* expert relevance.

$$w_k^a = \frac{1}{N}\sum_{i=1}^{N} p_{\mathbb{M}}(L_{ak}^i|\boldsymbol{x}^i) = \frac{1}{N}\sum_{i=1}^{N}\sum_{\boldsymbol{c}\in\mathcal{C}} p_{\mathbb{M}}(\boldsymbol{c}|\boldsymbol{x}^i)\cdot\mathbb{I}[c_k = L_{ak}^i]$$

The expected value of a reliability weight w_k^a calculated with this probability-based technique is, in the case of binary class variables,

$$E[w_k^a] = \sum_{\boldsymbol{x}\in\mathcal{X}} p_{\mathbb{G}}(\boldsymbol{x})\cdot \sum_{c_k\in\{0,1\}} p_{\mathbb{M}}(c_k|\boldsymbol{x})\cdot(p_{\mathbb{G}}(c_k|\boldsymbol{x})\cdot r_a + (1-p_{\mathbb{G}}(c_k|\boldsymbol{x}))\cdot(1-r_a))$$

As this second weight estimation approach takes into account both right and wrong (according to the model classifier $\mathbb{M}$) labels to calculate the weights w_k^a, the expert annotators do not reach high relevance rates. We have used small MBCs (10 binary variables, 3 of them classes, with parameters randomly generated from a Dirichlet distribution with all the $\alpha_i = 1$) in order to illustrate the expected ability of both approaches to recover the real relevance rate of the experts. Thus, the ratio of the real expert relevance ($h_{real} = r_e/r_a$) and the expert relevance ($h_{est} = E[w_k^a]/E[w_k^a]$) that is expected to be discovered for any binary class variable C_k is shown in Figure 5.4. By increasing the crowd and expert reliability rates ($r_a = \{0.5, 0.55, \ldots, 0.7\}$ and $r_e = \{r_a, r_a + 0.05, \ldots, r_a + 0.25\}$ relative to the specific r_a value), experts of different real relevance are induced in order to show the diverging behavior of both weight estimation approaches. According to the figure, no approach is expected to fully recover the real expert relevance, which is more evident in the case of outstanding experts. Even so, the accuracy-based approach recovers the real relevance of the expert annotators in a larger proportion.

5.5 Experiments

In this section, different sets of experiments are performed: the capabilities of the proposed method are explored (expert detection and promotion), the different configurations of the method are compared and, finally, our proposal and the presented basic CrL techniques are compared in different crowd scenarios with synthetic and real (ML) data.

All the synthetic datasets used in this section have been sampled from one of our MBC models with 10 binary variables: 7 predictive and 3 class variables. The tree structure among the class variables, the only non-fixed part of the MBC graph, is built by randomly visiting all the class variables and randomly selecting a previously visited class variable as parent of the current one. Model parameters are randomly generated by sampling a Dirichlet distribution with all the hyper-parameters equal to 1.

Regarding the simulation of a dataset labeled by a crowd, an original MD dataset (real or synthetic, but completely labeled) and a $(t \times m)$-matrix that codifies the probability of each annotator to provide the right label for each class variable (reliability rates) are used. Given an instance of the original dataset, the class label vector is replicated t times to simulate the labels provided by t annotators. Then, for each annotator and each class variable, the real label is maintained or swapped (only binary class variables are used) according to the corresponding reliability rate.

In these experiments, our EM method stops when the relative difference between the parameters of two models learnt in consecutive iterations is below 0.1%, or when the maximum number of 200 iterations is reached.

5.5.1 Recovering and using the expert knowledge

Working under the realistic assumption that the expert knowledge can be discovered, one of the characteristics of our method is its ability to take advantage of this expert knowledge during the learning process. In Section 5.3 we showed that the detection of experts is not a challenging procedure in the majority of cases. However, the discussed procedure —a basic approach that assess the annotator reliability based on the consensus among the different annotations (Eq. 5.2)— is not able to deal with the unfavorable information (the crowd reliability is $r_a \leq 0.5$). Alternatively, our method employs an estimated gold-standard to assess the reliability of the annotators. In this first experimental setting, we test whether the use of this gold-standard allows our method to successfully deal with those scenarios where the basic approach cannot identify the expert(s).

In this section, we study both the ability of the different configurations of our proposal to identify the real experts and the degree of relevance that is assigned to them (h_{est}). As we actually know the identity of the real experts (in these experiments there are 3, one per class variable), their relevance can be calculated as $h_{est} = w_k^e / \sum_{a \neq e} w_k^a$. We consider that the method discovers the

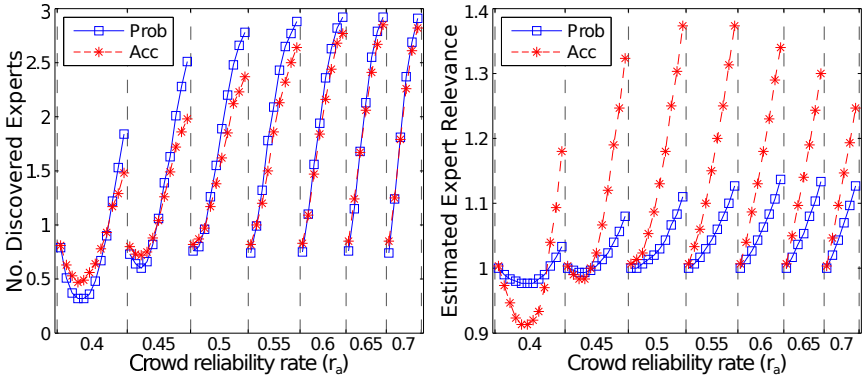


Fig. 5.5. Mean number of discovered real experts (left figure) and relevance assigned to them (right figure). Experiments in synthetic data generated with one expert per class variable and $cc = 2$, using the product-based weight combination procedure. Results for both weight acquisition approaches (based on accuracy or probability) are shown. Each vertical division groups experiments performed with the same crowd reliability value ($r_a = \{0.4, 0.45, \ldots, 0.7\}$). In each division, a line links related experiments with increasing expert reliability ($r_e = \{r_a, r_a + 0.05, \ldots, 0.95\}$), which allows us to generate experts of different relevance.

real experts if their relevance is larger than one ($h_{est} > 1$) in the corresponding class variable C_k. For this comparison, a fixed number of annotators ($t = 4$) has been used and, for each class variable, one of them has expert knowledge. In order to simulate different kind of experts, we generate scenarios where a single annotator is expert in several class variables at the same time ($cc = \{1, 2, 3\}$). Additionally, 7 crowd reliability rates $r_a = \{0.4, 0.45, \ldots, 0.7\}$ and different expert reliability rates $r_e = \{r_a, r_a + 0.05, \ldots, 0.95\}$ relative to the crowd reliability rate r_a used in the specific experiment have been induced. In total, 189 different crowd scenarios were generated, each of them replicated $1,000$ times (10 generative MBCs $\times$ 10 MD datasets $\times$ 10 crowd annotations) and evaluated in a 10×5-fold cross validation procedure.

Figure 5.5 collects the results of a representative subset of the experiments (only those which use the product-based weight combination technique and datasets with $cc = 2$). As expected, the method easily discovers 2 of the experts and tends to discover the 3 experts as the reliability of both experts and novices rises. Even in those cases where the crowd reliability rate is not favorable ($r_a \leq 0.5$), it is also able to detect the outstanding experts. When novices label randomly ($r_a = 0.5$), the real experts are mostly detected. When the novices are saboteurs ($r_a < 0.5$) and provide contrary information, the method firstly follows the majority (wrong) opinion of the saboteurs, considering the real expert a harmful annotator (which explains the initial drop in the number of discovered experts). However, as the real expert relevance rises (large $h_{real} = r_e / r_a$), the method is able to identify a larger number of real

experts —i.e., the method, to some extent, figures out that the majority opinion is wrong and promotes the expert knowledge. Figure 5.5 also reveals the different behavior of the method depending on the implemented weight acquisition procedure. As discussed in Section 5.4.2, the probability based weight acquisition procedure assigns a lower relevance to the expert annotators in comparison with the relevance obtained by means of the accuracy-based approach. Surprisingly, the probability-based approach finds a larger number of experts than the accuracy-based approach.

Throughout this discussion, we have left aside the influence of the number of classes of expertise (cc) of the expert(s). The contribution of a domain expert that provides reliable labels for several class variables (a large cc) strongly affects the learning process in complex crowd scenarios, where the proposed methodology tends to select a global expert in all the classes. Let us imagine a crowd scenario where an annotator A_e is a domain expert in $cc = 2$ classes (C_1 and C_2) and another annotator $A_{e'}$ is expert just in C_3. According to this idea, the system will assign a high reliability degree to A_e in all the classes. Due to the per-class weight combination procedure, the contributions of A_e to the counts that include the class variable C_3 (in which A_e is just a novice) are assigned larger weights than those of the real expert in C_3, $A_{e'}$. Note that in the MBCs used in these experiments (10 binary variables, 3 of them classes), more than 98% of the counts require the application of a per-class weight combination procedure. Consequently, the model will consider that the most reliable joint labeling is that provided by A_e, being unable to detect that $A_{e'}$ is the real expert of class variable C_3. Thus, as the performance in the classes of expertise of A_e (C_1 and C_2) strengthens, the local performance in C_3 is affected negatively and, consequently, so is the global performance.

Figure 5.6 shows the influence of increasing the number of classes of expertise ($cc = \{1, 2, 3\}$) —remember that a single expert per class is generated— over the final results (in terms of *micro/macro F1* measures [6]). As discussed in Section 5.4.1, the referred influence varies with the way in which each per-class weight combination approach deals with the reliability weights. Due to the enhanced expert relevance that characterizes the product-based strategy, better performing classifiers are learnt as the learning process relies more on the expert opinion. As expected, this outstanding behavior of the product-based combination strategy strengthens as the number of classes of expertise (cc) rises.

5.5.2 Looking for the best configuration of the proposed method

In Section 5.3, we characterized the crowd scenarios where each basic CrL strategy is the best option to combine the crowd information. We characterized the crowds scenarios with data scarcity as the space where our (non-basic) method could contribute to improving the results of these basic approaches. Consequently, in this kind of scenarios we have studied the behavior of the four different configurations of our proposal: two per-class weight combination

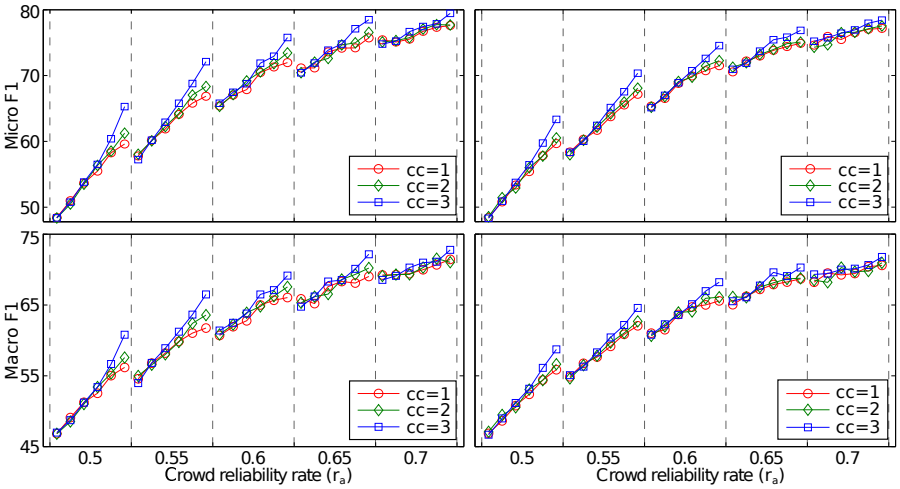


Fig. 5.6. Having a single expert per class variable, the figures display the behavior in terms of *micro/macro F1* measures of both per-class weight combination strategies (product-based —left column— and addition-based —right column—) as the number of classes in which the same expert stands out increases ($cc = \{1, 2, 3\}$). Each vertical division groups experiments performed with the same crowd reliability value ($r_a = \{0.5, \ldots, 0.7\}$). In each division, a line links related experiments with increasing expert reliability ($r_e = \{r_a, r_a + 0.05, \ldots, r_a + 0.25\}$).

procedures (addition-based —*Add*— and product-based —*Prod*—, presented in Section 5.4.1) per two reliability weight estimation approaches (accuracy-based —*Acc*— and probability-based —*Prob*—, presented in Section 5.4.2). Additionally, we have considered the assignment of equal reliability weights to all the annotators as a baseline strategy (*Eq*). The comparison with this equal-weights strategy allows us to objectively grade the contribution of the individual non-equal reliability weights that our method estimates. For the sake of fairness, *Eq* has been applied together with both weight combination procedures.

This comparison covers 360 different crowd scenarios: 4 different numbers of annotators $t = \{3, 4, 5, 6\}$, one expert per class with 3 different numbers of common classes per expert $cc = \{1, 2, 3\}$, 5 crowd reliability rates $r_a = \{0.5, 0.55, 0.6, 0.65, 0.7\}$ and 6 expert reliability rates $r_e = \{r_a, r_a + 0.05, \ldots, r_a + 0.25\}$ relative to the crowd reliability rate r_a used in the specific experiment. Each crowd scenario has been replicated 1,000 times (10 generative MBCs $\times$ 10 MD datasets $\times$ 10 crowd annotations), all of them evaluated in a 10×5-fold cross validation procedure.

In order to analyze the obtained results, we have used the statistical framework proposed by Demšar [46] and García and Herrera [63]. This framework indicates a way to perform a statistical validation looking for significant differences in the performance of the methods under comparison. In our case,

Method	*Eq+Add*	*Eq+Prod*	*Prob+Add*	*Prob+Prod*	*Acc+Prod*	*Acc+Add*
Av. Rank	4.32	4.28	3.74	3.23	2.84	2.59

Table 5.1. Average Ranks of the 4 configurations of the proposed method and another 2 with equal weights according to *mean accuracy* results.

where several methods (and configurations) have to be compared, this framework first performs a Friedman test, which analyzes whether the methods follow the same probability distribution (null-hypothesis). If the Friedman test null-hypothesis is rejected, several post-hoc tests are applied to compare the methods by pairs.

According to the Friedman test [46], statistically significant differences do exist between the *mean accuracy* results (mean value of the accuracy in each class variable) of the different methods and configurations under comparison (see average ranks in Table 5.1) when the type I error is fixed to $\alpha = 0.05$. As the Friedman test rejects the null hypothesis, post-hoc paired tests have been performed to discover differences between pairs of configurations using the Holm procedure [46] (see the associated critical difference diagram in Fig. 5.7). The post-hoc paired tests can not find statistically significant differences ($\alpha = 0.05$) between both configurations that use the accuracy-based weight estimation approach (*Acc+Prod* and *Acc+Add*). Moreover, no statistical difference can be found between both *Eq* configurations and the *Prob+Add* configuration (probability-based weight acquisition approach with addition-based combination). This behavior can be explained as the difficulty of the *Add* combination procedure to maintain or improve the low expert relevance that the *Prob* approach produces. On the other hand, the *Prod* combination procedure keeps and even takes advantage of that low *Prob*-obtained expert relevance.

Similar tests have been performed looking for statistical differences when *micro F1* and *macro F1* measures are used. All of them find the same differences between the configurations using the accuracy-based weight estimation technique (which show the best performance) and the rest of the configurations. Only the statistical tests based on the *micro F1* measure find statistically significant differences between *Acc+Add* and the best configuration, *Acc+Prod*.

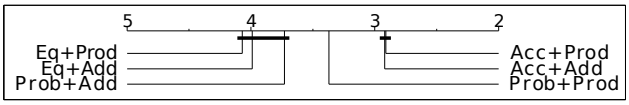


Fig. 5.7. Result of the Holm procedure marking statistically significant differences found at $\alpha = 0.05$. 4 configurations of our proposal and another 2 with equal weights are located in the scale according to their ranking (based on *mean accuracy* results). Bold horizontal lines link methods that are not significantly different.

5.5.3 Comparison with basic CrL techniques in synthetic data

Once shown that the use of non-equal individual reliability weights improves the performance of the learnt models, we have carried out a set of experiments to compare our proposal with both basic CrL techniques discussed in Section 5.3: majority voting and expert selection. In the multi-dimensional framework, the exposed techniques have been applied to each class variable separately. Based on the conclusions of the previous experimental tests, we decided to use the *Acc+Prod* configuration to set up our method.

As explained in Section 5.3, an alternative CrL proposal would be a notable contribution if it was able to overcome the simple techniques in crowd scenarios of data scarcity. In this experimental setting, complex MDCrL scenarios are simulated using 8 different numbers of annotators $t = \{3, 4, \ldots, 10\}$, one expert per class with 3 different numbers of common classes per expert $cc = \{1, 2, 3\}$, 5 crowd reliability rates $r_a = \{0.5, 0.55, 0.6, 0.65, 0.7\}$ and 6 expert reliability rates $r_e = \{r_a, r_a + 0.05, \ldots, r_a + 0.25\}$ relative to the specific r_a. The obtained results are the mean value over $1,000$ replications of the same crowd scenario (10 generative MBCs $\times 10$ MD datasets $\times 10$ crowd datasets), all of them evaluated in a 10×5-fold cross validation procedure.

Figure 5.8 shows the results of the comparison of our method using the *Acc+Prod* configuration with respect to MV and ES by means of *micro/macro F1* and *mean accuracy* measures in a representative subset of the experimental scenarios ($t = \{4, 6, 8\}$ and $cc = 2$). As expected, the three measures show that our method outperforms both baseline techniques in crowd scenarios with small expert relevance and a reduced number of annotators. As the expert relevance rises, the expert selection strategy reaches the performance of our method and, with even higher degrees of relevance, it finally outperforms our proposal (although Fig. 5.8 shows a maximum difference of $r_e = r_a + 0.25$, beyond this point the trend is expected to remain). On the other hand, an increase in the number of annotators (t) benefits MV, which gets closer to the behavior of our proposal as t increases (remember that MV does not require any expert detection procedure). Specifically, it can be observed in Fig. 5.8 that the number of annotators (t) required by MV to show a similar behavior to our method increases as the reliability rate of the novices (r_a) decreases.

The same statistical framework [46, 63] used in the previous subsection was applied to analyze the results of this comparison. All the statistical tests (based on *mean accuracy*, *micro F1* and *macro F1* measures, using $\alpha = 0.05$) show the same differences: The Friedman test [46] finds statistically significant differences between the results of the three methods, and the post-hoc paired tests using the Holm procedure [46] discover differences between all the pairs of methods.

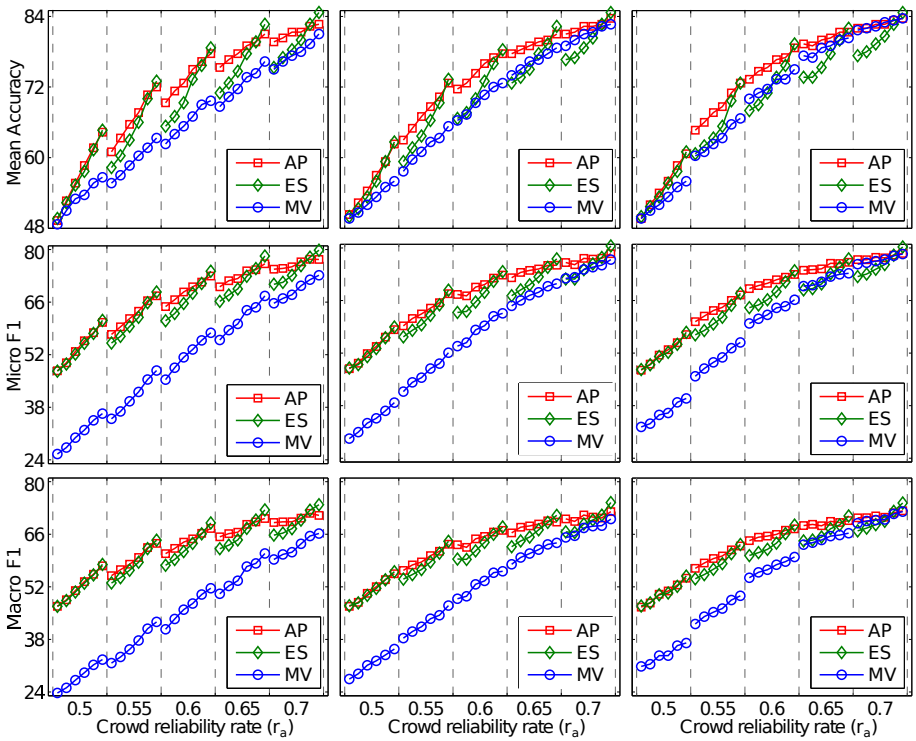


Fig. 5.8. Comparison of our method using the *Acc+Prod* configuration (AP) with Majority Voting and Expert Selection in terms of *mean accuracy* and *micro/macro F1*. Each column represents experiments with a different number of annotators, $t = \{4, 6, 8\}$. Each vertical division groups experiments performed with the same crowd reliability value ($r_a = \{0.5, \ldots, 0.7\}$). In each division, a line links related experiments with increasing expert reliability ($r_e = \{r_a, r_a + 0.05, \ldots, r_a + 0.25\}$).

5.5.4 Comparison in real multi-label data

With the objective of testing our proposal in real data, a comparison in crowd scenarios simulated from three real multi-label benchmarks[1] (*emotions*, *flags* and *scene*) has been performed. Based on the previous experimental setting, we use the real ML data transformed to the MD framework and the crowd annotations simulated as explained at the beginning of this section.

In this case, the experimental setting explores crowd scenarios with 3 different numbers of annotators $t = \{4, 6, 8\}$, 5 crowd reliability rates $r_a = \{0.5, 0.55, \ldots, 0.7\}$ and 6 expert reliability rates $r_e = \{r_a, r_a + 0.05, \ldots, r_a + 0.25\}$ relative to the specific r_a. Given a single expert per class, 4 different numbers of common classes of expertise per expert have been inferred

[1] http://mulan.sourceforge.net/datasets.html

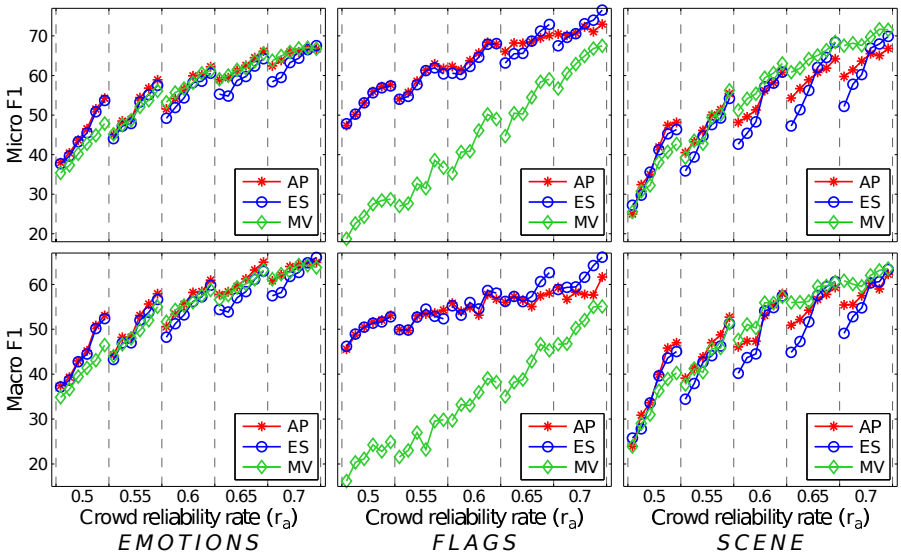


Fig. 5.9. Comparison in real ML data of the *Acc+Prod* configuration of our method (AP), Majority Voting and Expert Selection. Each column shows the *micro/macro F1* results of experiments with a different ML dataset (*emotions*, *flags* and *scene*). Using 4 annotators and $cc = 4$, each vertical division groups experiments performed with the same crowd reliability value ($r_a = \{0.5, \ldots, 0.7\}$). In each division, a line links related experiments with increasing expert reliability ($r_e = \{r_a, r_a + 0.05, \ldots, r_a + 0.25\}$).

$cc = \{1, 2, 4, 6\}$ (the datasets have 6, 7 and 6 binary class variables, respectively). This makes a total number of 360 different crowd scenarios tested in each real ML dataset, all of them evaluated in a 10×5-fold cross validation procedure. The results show the average value over 10 replications (different crowd datasets). Continuous variables have been discretized using equal-frequency with 3 intervals.

In a comparison of the *Acc+Prod* configuration of our method, MV and ES, Figure 5.9 shows the results in terms of *micro/macro F1* measures for a fixed number of annotators ($t = 4$) and number of classes of expertise ($cc = 4$). The performance of the three methods varies depending on the different domains and their characteristics (Tab. 5.2). Whereas the results of the MV strategy

Dataset	No. variables	No. class variables (m)	No. instances (N)
Emotions	78	6	593
Flags	26	7	194
Scene	300	6	2407

Table 5.2. Description of the three real ML datasets.

are far from those of our method and ES in any scenario of the *flags* domain, it shows a competitive performance in the *emotions* domain and overcomes our proposal in the most-informed scenarios of the *scene* domain. This reflects the difficulties of MV to deal with crowd scenarios of data scarcity such as the *flags* domain. However, our method shows a good behavior in the three domains and it is only clearly overcome in the most-informed scenarios of the *scene* domain.

The behavior of the methods in the rest of tested scenarios is similar to the one observed in Figure 5.9, showing the expected variations derived from the increase in the number of annotators. The results of the whole set of experiments were analyzed using the statistical framework [46, 63] previously presented. Using *micro F1* measures and $\alpha = 0.05$, the Friedman test finds statistically significant differences between the results of the three methods. The post-hoc paired tests performed with the Holm procedure [46] cannot find differences between ES and MV. Using *macro F1* measures, the Friedman test once more finds statistically significant differences between the results of the three methods, and the post-hoc paired tests using the Holm procedure discover differences between all the pairs of methods.

5.5.5 A more realistic scenario: Using multi-label classifiers as annotators

In order to reduce our direct influence in the generation of the crowd annotations, we have carried out a last set of experiments where each annotator is a classifier. The presence of classifier models among the annotators of a crowd has been detected and discussed in real CrL problems [41, 30, 200]. Based on this idea, we follow the experimental setting of [200] and use the predictions of the classifiers to simulate the annotations of different annotators. Without access to real MDCrL data, we consider that the combination of this idea with the real ML datasets (*emotions*, *flags* and *scene*) provides a realistic framework to test our proposal.

In order to simulate the annotators of this last experimental setting, a set of 8 multi-label classifiers implemented in the *MULAN* software[2] has been used: the lazy classifiers MLkNN and BRkNN, and the meta-classifiers Binary Relevance, Calibrated Label Ranking and Classifier Chain using naive Bayes and J48 as base classifiers.

The number of annotators in these experiments was fixed to $t = 4$ and, consequently, the classifiers employed in the specific replication of the experiment were randomly selected among the 8 previously presented. Each selected technique is used to learn a ML classifier from the original ML dataset. Next, the joint labeling that the different learnt ML classifiers predict for the instances of the original dataset are obtained. Finally, these predicted labels are used to simulate a MD dataset annotated by a crowd of 4 annotators. Each

[2] http://mulan.sourceforge.net/

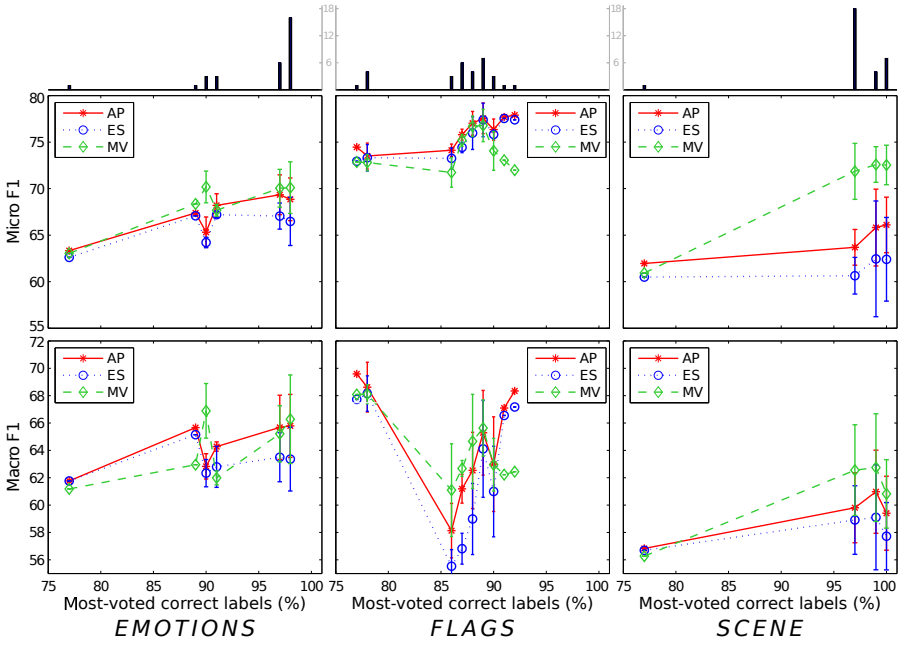


Fig. 5.10. Comparison, using ($t = 4$) ML classifiers as annotators, of the *Acc+Prod* configuration of our method (AP), Majority Voting and Expert Selection. Each column shows the *micro/macro F1* results of experiments with a different ML dataset (*emotions*, *flags* and *scene*). In each figure, the 30 experiment replications are located in the horizontal axis according to their percentage of labels which were correctly annotated by at least half of the annotators (as a measure of the complexity of crowd scenarios). When several replications coincide in the same percentage value (shown in the upper bar diagram), the mean value with the associated standard deviation is shown.

experiment has been evaluated in a 10×5-fold cross validation procedure, and the whole process has been replicated 30 times in order to reduce the dependence on the random selection of classifiers.

Figure 5.10 shows the performance in terms of *micro/macro F1* measures of our proposal and the basic CrL techniques (ES and MV) in crowd scenarios generated from the three ML domains. As the complexity of these experiments has not been artificially induced, the type of figure displayed in previous experiments cannot be used. Thus, each figure represents the 30 replications of the experiment according to their complexity measured in an alternative way: the proportion of labels correctly annotated by at least half of the annotators. This measure might be also understood as the proportion of correct labels which are also the most voted ones, that is, as its value tends to 100% the probability of MV overcoming the results of the other methods rises.

This alternative complexity measure also reflects the larger complexity of the *flags* domain. The experiments performed with the *emotions* and, mainly, the *scene* datasets are located over 90% in the horizontal axis according to this measure, which represents a large correct annotation of the instances. However, in the case of the experiments with the *flags* dataset, few experiments exceed that value. Particularly in the complex domains (*flags* and *emotions*), our method shows a competitive behavior and is able to widely overcome the results of the other techniques. However, in the *scene* domain, a dataset favorable for the MV strategy, our method only overcomes the results of that basic strategy in a scenario where the complexity measure does not reach 80%. In this comparison, ES is not able to reach the behavior of the other two methods.

5.6 Conclusions and future work

We have presented a general framework for learning multi-dimensional Bayesian network classifiers from data annotated by a crowd of annotators. Focused on improving the learning process in complex crowd scenarios, different ways to incorporate the information of the reliability of the annotators have been explored.

In order to reveal the strengths of the proposed method, this chapter presents a complete study of basic CrL approaches, characterizing the crowd scenarios where each discussed approach shows a better performance. Throughout the chapter, a set of useful guidelines which can be used to select the most convenient strategy to cope with a specific crowd scenario is presented. Finally, by means of a set of experiments performed with ML (real or synthetic) datasets transformed to the MDCrL framework, our proposal has been shown to overcome the simple approaches in crowd scenarios with data scarcity.

In real crowd scenarios, it cannot be assumed that every annotator labels all the instances and class variables —in this way, Sun et al. [175] proposed an extra valid state for the annotations (class-member, non-member, *unknown*). As future work, releasing this assumption would imply taking into account annotators who label few instances. This would require the redesign of the techniques that have been proposed to calculate the reliability weights. Moreover, in this study it is assumed that all the annotators provide wrong labels randomly. On the contrary, considering non-random noisy annotators (e.g., someone that tends to label incorrectly only the examples of a specific area of the instance space) could require a specific methodology.

Let us imagine a crowd scenario where the number of annotators labeling each instance is very different; or a different scenario where a brilliant domain expert labels only a few instances. It could be interesting to implement a complete framework which is able to choose in run-time the best approach

for each instance: majority voting if many labels have been provided, expert selection only for the instances labeled by highly reliable experts, etc.

Part III

Applications

6

Assisted Reproductive Technologies

In medicine, the problem of the assisted reproductive technologies (ARTs) — the difficulty of inducing a pregnancy without increasing the probabilities of the intrinsically risky multi-pregnancy— has received considerable attention. It is generally accepted in the related literature that there is room for advances in this field. Particularly, many medical decisions have to be taken during the whole procedure and, consequently, current research lines aim to increase the knowledge on the problem to support the decisions of the physicians. With this objective, different artificial intelligence and machine learning techniques have been applied to the ART problem.

In collaboration with the Unit of Assisted Reproduction of the Donostia Hospital, we have proposed an integral solution for the ART problem by means of a case study. The main objective is to gain evidence about the relevance of the collected data and its potential use for improving the rate of pregnancies. Based on an embryo-uterine design, four different approaches which provide valuable information for partially solving the ART problem have been proposed by our multi-disciplinary team. Three of these approaches are configured as weakly supervised classification problems, according to the description provided in Section 2.2. Contrary to the common practice in previous solutions to the problem, where embryos of unknown fate are usually discarded, our weakly supervised techniques consider every example (even embryos/cycles whose fate cannot be certainly established). The four approaches have been successfully tested in our case study. In the performed experiments, obtained classifiers outperform the implemented embryo selection criteria, proposed by the Spanish Association for Reproduction Biology Studies. Specifically, medium-quality embryos are extensively reordered. The study reveals the limits of the collected morphological features. Furthermore, it reflects the need for a thorough study of the implantation process and the identification of new features that describe it precisely.

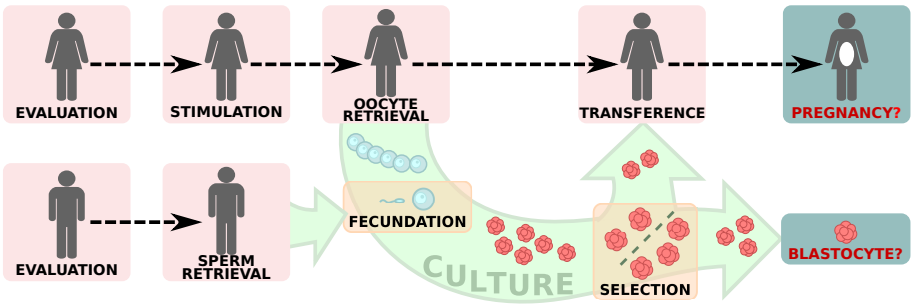


Fig. 6.1. Diagram of an ART cycle. Red boxes indicate an action with the patients. Yellow boxes indicate an action with oocytes/embryos. Both blue boxes represent two observable outcomes: (a) Has the woman got pregnant? (b) Has a non-transferred embryo reached blastocyte stage?

6.1 Introduction

Assisted reproductive technologies (ARTs) are a set of invasive medical techniques that attempt to induce a pregnancy. Each trial of a reproduction treatment applying a suitable ART is known as a cycle. When a woman undergoes an ART cycle, she follows a treatment of ovarian stimulation for several weeks in order to induce the development of multiple follicles with a large number of oocytes. By means of a punction, oocytes are retrieved using an ultrasound-guided transvaginal follicle aspiration. The mature oocytes are subsequently fertilized and the resulting embryos cultured for several days. The next step, the selection of the most promising embryo(s) to transfer to the uterus of the woman is a critical decision in the ART procedure. After transference, the occurrence of embryo implantation —a natural process that cannot be supervised by the specialist— determines the success of an ART cycle: Implantation of at least one of the transferred embryos leads the cycle to a pregnancy. See Figure 6.1 for a graphical description of a standard ART cycle.

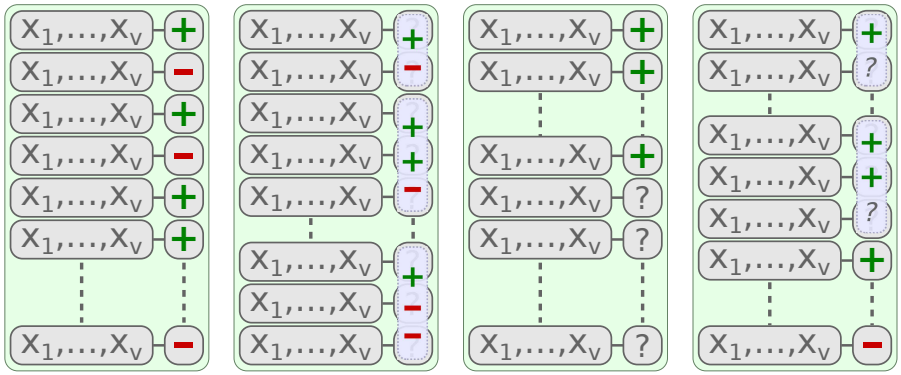
An accurate identification of those cycles that will end up in a pregnancy would surely be reflected in a significant improvement of the performance of the ARTs. For decades, there has been a persisting discussion on the features that determine the success of a cycle from those usually collected in the different stages of the ART procedure. Considered features can be separated in variables that describe the cycle —female and male evaluation, stimulation, etc. (red boxes in Fig. 6.1)— and those that describe each single oocyte/embryo (green stream and yellow boxes in Fig. 6.1). In their exhaustive reviews, Achache and Revel [1] and Ebner et al. [49] collected and discussed an extensive set of variables that have been considered for assessing the quality of both cycles and oocytes/embryos. The use of data analysis techniques in many research works, where the contribution of different features is evaluated, has resulted in the presentation of an unbounded number of embryo scores and selection criteria [36, 49, 53, 153, 173, 183, 203]. More

recently, taking advantage of the development of computational techniques, different machine learning (ML) paradigms, such as supervised classification, have been applied to the analysis of the ART problem by multidisciplinary research teams [33, 40, 71, 126, 134, 137, 146]. In supervised classification, a classification model that reproduces the inherent categorizing behavior of a problem of interest —which is learnt from a set of previous examples— is built. Each example describes a real case of the problem and is provided annotated with its real category (class label). A classifier will anticipate (accurately) the class label of new uncategorized examples. Applied to the ART problem, a straightforward approach is to use the data collected from previous cycles to learn *a classifier that predicts whether a new cycle will end up in a pregnancy*. The training examples (cycles) are certainly labeled from five weeks after embryo-transfer, when the existence of an evolutive pregnancy can be assessed by the use of ultrasound techniques. Many authors have used this approach for evaluating the relevance of a reduced set of *allegedly* determining variables (e.g., woman's age, time of sterility or number of previous cycles) [126, 146, 153]. On the contrary, in our study all the features collected by the physicians (listed in Table 6.2) have been initially considered as predictive variables: female and male characteristics, treatment, transference and a set of features that summarizes the characteristics of the oocytes/embryos obtained in the cycle. Although our learning techniques automatically calibrate the contribution of each predictive variable (feature), the use of a larger set of variables usually introduces irrelevant/redundant variables, which can be harmful to the performance of the classifier [96]. Thus, feature subset selection (FSS) techniques [72, 151] are applied for automatically identifying the relevant predictive variables and discarding those that are uninformative. With our strategy, physicians do not need to manually choose the relevant features: The model inputs all the collected variables and automatically establishes their contribution. Moreover, learnt classifiers can be used to test which cycle configuration (e.g., stimulation treatment) maximizes the probability of success.

As previously mentioned, the objective is to improve the pregnancy rate of the ARTs. Traditionally, the number of embryos to transfer is assumed to be positively correlated with the probability of pregnancy [52, 119]. Thus, the *multi-transference* cycles (more than one embryo was jointly transferred

		Implanted			
		0	1	2	3
Transfer	1	32	8	-	-
	2	140	45	29	-
	3	45	20	9	2

Table 6.1. Summary table with the number of ART cycles according to their number of transferred/implanted embryos in our case study.



(a) Full supervision, Pregnancy prediction (b) Label proportions, Implantation predict. (c) Positive-unlabeled, Pregnanble prediction (d) PU proportions, Implantable prediction

Fig. 6.2. Graphical description of the (partial) labeling of the training data that characterizes the four (weakly) supervised approaches used in this paper.

in that ART cycle) exceed 85% of the cycles in our case study (Table 6.1). Multi-transference may give room to the joint implantation of several embryos, eventually leading to a *multiple* pregnancy (in 22% of the multi-transference cycles of our case study; Table 6.1), which is widely considered risky for both the woman and the developing fetus(es) [52, 68, 109, 119, 142]. Therefore, multiple transference raises both the probability of pregnancy and the risk of multiple pregnancy [52, 119]. In order to reduce the occurrence of multiple pregnancy, legal restrictions limiting the maximum number of transferred embryos were established (in our case study, Spanish law limits it to 3). As the clinical ART procedure usually produces excess embryos, clinicians have to select those to transfer. In part opportunity, in part necessity, embryos to transfer have to be carefully selected as the transference of poor-quality embryos is a major contribution to ART failure [1, 49]. In the related literature, all these considerations have originated an interesting discussion on the possibility of selecting and transferring a set of promising embryos (or just one) that will lead to a *single* pregnancy [52, 68, 109, 119]. In the general scenario, the decision is two-fold: the number of embryos to transfer [109, 119] and their individual selection among the obtained embryos. The aim to provide an answer to the second question has motivated a different classification task which *predicts whether an embryo, in the case of being transferred, will get implanted*. Due to its utility for supporting the physicians' decision on the selection of embryos to transfer, this "implantation" prediction approach has become the most popular application of ML techniques in the field of the ARTs [33, 40, 71, 134, 137]. In our case study, the oocyte/embryonic morphological features collected by physicians (those required to apply the embryo

selection criteria proposed in [3]; Table 6.3) have been used as predictive variables, whereas the class variable *Implantation* indicates whether the embryo (example) got implanted. As non-transferred embryos never had the opportunity to get implanted, only transferred embryos have been used for training the classifiers in this approach. Contrary to the previous approach (pregnancy prediction), a classical fully labeled dataset —Fig. 6.2(a)— cannot be collected for this approach: Current medical techniques only allow clinicians to know the *number* of implanted embryos, not their *identity*. Although it is a common practice in the related literature to discard the embryos of unknown fate [40, 86, 137, 153, 183, 203], our specifically designed ML techniques allow us to also use them in model learning. The conclusions drawn from the results of any data analysis (including ML techniques) are affected by the sample size. Hence the effort to incorporate any available example to our analysis. It has already been shown that the *weakly supervised* ML techniques are able to efficiently learn from this kind of partially labeled data. Specifically, the embryo implantation approach is modeled by means of the learning from label proportions (LLP) [81] paradigm: the training dataset is presented divided in groups (bags) of unlabeled examples and, for each group, the number of positive/negative examples is provided —Fig. 6.2(b).

However, both previously discussed approaches present a problem: The conditions that benefit the implantation of an embryo and the consequent pregnancy have not been fully understood [1, 49]. In general, the implantation failure is assumed to be a consequence of embryonic or uterine factors. Furthermore, although uterine cavity or genetic abnormalities, thin endometrium, immunological factors or suboptimal ovarian stimulation [1, 34, 117, 165] have been argued to explain the occasional failure of good prognostic embryos in promising cycles, more research is required for a conclusive answer. When this *misunderstood* implantation failure —denoted by its acronym MIF henceforth in the chapter— repeatedly occurs in consecutive treatments, a recurrent implantation failure is diagnosed [34, 117, 165]. MIF occurrence can be seen as the limited ability of currently monitored cycle/embryonic features to describe the implantation process. Therefore, the use of this data for learning classifiers would limit their capability to predict a pregnancy or an implantation. Partially inspired by the embryo-uterine (EU) model [171], two alternative approaches that consider the impact of the MIF phenomenon from the point of view of both cycles and embryos have been designed. To be successful, an ART procedure requires both the transference of good quality embryos and a cycle which is able to lead to a pregnancy [1, 171]. These two states, which are independent/previous to implantation (or a hypothetical MIF occurrence) but achieved by any successful ART procedure, are represented by our alternative approaches.

Let us consider the first alternative approach, which learns classifiers that anticipate the *capability of an ART cycle to lead to a pregnancy*. Note that it does not predict pregnancies but it identifies good prognostic cycles that, in the best scenario (i.e., it is carried out with promising embryos without

	Variable	Possible values	Brief description
	Indication	endometriosis, failed intrauterine insemination, tubal factor, male, mixed, other, unknown	Indication of the cycle
	Infert.time	*numeric*	Time since infertility was detected
Female	Age	$(0, 30]$, $(30, 35]$, $(35, \text{inf})$	Age
	BMI	$(0, 20]$, $(20, 25]$, $(25, \text{inf})$	Body mass index
	Prev.Pregnancy	No, Yes	Has she ever got pregnant?
	Prev.Abortion	No, Yes	Has she ever aborted?
	Prev.Cycles	0,1,2+	Number of previously undergone ART cycles
	FSH	$[x \leq 10]$, $[10 < x]$	Quantity of follicle-stimulating hormone
	AMH	$[0, 0.5]$, $(0.5, 1]$, $(1, \text{inf})$	Quantity of anti-mullerian hormone
	antralFol	$[x \leq 4]$, $[4 < x]$	Number of antral follicles
	E2	$[x \leq 3000]$, $[3000 < x]$	Quantity of estradiol
	P4	$[x \leq 1.5]$, $[1.5 < x]$	Quantity of progesterone
	lEnd	*numeric*	Endometrial thickness
Male	Quality.Semen	A, N, O, OA, OAT	Quality of the semen
	REM	$[0, 0.5]$, $(0.5, 1]$, $(1, \text{inf})$	Total pregressive sperm recovery
Stimulation	Protocol	PC, PL	Stimulation protocol
	Stimulation	FSH+Lhrec, FSHrec, FSHrec+hMG, FSHur, FSHur+hMG, hMG	Stimulation treatment
	dEst	*numeric*	Number of days of stimulation
	unitFSH	*numeric*	Units of FSH
	unitLH	$[0]$, $(0, 1500]$, $(1500, \text{inf})$	Units of LH
Summary embryos	No.Oocytes	*numeric*	Number of retrieved oocytes
	No.MII	*numeric*	Number of mature oocytes (MII state)
	No.Embryos	*numeric*	Number of embryos
	FertilityRate	$[0, 0.5]$, $(0.5, 1]$	*No.Embryos / No.MII*
	No.Transf.Emb	1, 2, 3	Number of transferred embryos
	SelectiveTransf	No, Yes	Were transferred embryos selected? (*No.Embryos > No.Transf.Emb*)
Outc.	Pregnancy	No, Yes	Did she get pregnant?
	No.Sacs	0, 1, 2, 3	Number of gestational sacs

Table 6.2. Features collected for each ART cycle.

MIF occurrence), would end up in a pregnancy. For the sake of simplicity, we have decided to name this approach as "pregnanble" prediction, denoting the characterization of a state that possibly (not certainly) leads to a pregnancy. In this approach, the training examples represent cycles and are described by all the cycle features (Table 6.2) —with the exception of those describing the process of transference: *No.Transf.Emb* and *SelectiveTransf*. The training dataset is divided in clearly separable two subsets. A subset of positive examples (cycles), which is composed of all those cycles that ended up in a pregnancy: As our target variable assesses the accomplishment of a pre-requirement of the pregnancy, any cycle that ends up in a pregnancy was obviously capable of leading to a pregnancy. Similarly, a cycle which failed due to the transference of poor quality embryos (or to the referred MIF oc-

	Variable	Possible values	Brief description
Oocyte	Vac	No, Yes	Presence of vacuoles
	SER	No, Yes	Presence of smooth endoplasmic reticulum clusters
	PVS	Normal, Augmented	Description of the perivitelline space
	PB	Normal, Abnormal	Description of the first polar body
	Technique	IVF, ICSI	Fertilization technique
D+1	PB.1	1, 2, 3+	Number of polar bodies
	Z	Z1, Z2, Z3, Z4	Scott's pronuclear grade [156]
D+2	nCel.2	{4}, {2;5}, {other}	Number of cells
	frag.2	$[0, 10]$, $(10, 25]$, $(25, 35]$, $(35, 100]$	Percentage of cell fragmentation
	symmet.2	No, Yes	Are the blastomeres symmetric?
	PZ.2	Normal, Abnormal	Presence of abnormalities in the pellucid zone
	vac.2	No, Yes	Presence of vacuoles
	multiNuc.2	No, Yes	Presence of multi-nucleation in a cell (no.nuclei ≥ 2)
	Quality.2	A, B, C, D	ASEBIR quality grade [3]
	Transfer	No, Yes	Embryo selected for transference
Outc.	Implanted	No, Yes	Did it get implanted?
	Blastocyte	No, Yes	Did it reach the blastocyte stage?

Table 6.3. Features collected for each oocyte/embryo. *Implanted* and *Blastocyte* variables cannot be always annotated by clinicians.

currence) cannot be considered as a negative example of this approach, whose aim is the identification of promising cycle configuration. That is, the failure of a cycle does not always imply a bad cycle configuration. And as current medical techniques cannot determine which specific cause is responsible for the failure of each cycle, the respective examples (second subset) are annotated as unlabeled. This is modeled by means of another weakly supervised classification paradigm, the positive unlabeled (PU) learning [18], where a binary classifier is learnt from a dataset that only contains positive-labeled and a majority of unlabeled examples —Fig. 6.2(c). PU techniques allow us to learn from this kind of data, taking into account that there could be both positive and negative examples in the unlabeled subset.

Applied to embryo assessment, this idea of anticipating the achievement of a state previous to implantation characterizes *embryos that, in favorable conditions, are able to get implanted*; that is, embryos of correct development. Two days after oocyte fecundation (*D+2*), when embryo selection and transference are carried out in our case study, an embryo has only achieved a basic development [49, 53, 71]. Our physicians carried on the culture of non-transferred embryos and vitrified and preserved those that reached blastocyte state. Blastocyte formation is considered an indicator of correct development and desirable embryo quality [53, 71]. Thus, our fourth approach, the "implantable" prediction, tries to identify embryos that reach this state (good prognostic). The features of Table 6.3 are used for describing the examples (embryos) and the approach is modeled by means of a weakly supervised classification paradigm that uses as training data both transferred and non-

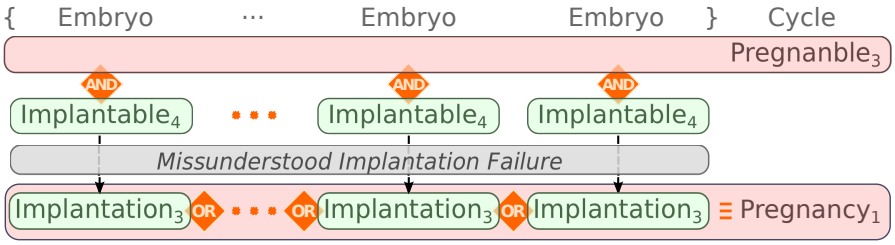


Fig. 6.3. Relationship between the four approaches considered for the ART problem. A $pregnancy_1$ occurs if at least one embryo $implantation_2$ occurs. Although some cases unexpectedly fail —the implantation process is not fully understood yet— an embryo $implantation_2$ requires a good prognostic embryo transferred in a good prognostic cycle. The $pregnanble_3$ and $implantable_4$ approaches identify good prognostic cycles and embryos, respectively.

transferred embryos. The latter, cultured in the laboratory until blastocyte state, are individually labeled. However, fairly labeling the former subset leads us to a weak supervision scenario. Implanted embryos did certainly develop to blastocyte. As in the previous approach, a similar reasoning is applied to failed embryos: A transfer-embryo that did not get implanted cannot be labeled (as negative) because there are other valid causes for a failure apart from a halt in its development. Additionally, as the implanted embryos are not individually identified, the real label of each transferred embryo is only known for those cycles where *all* the transferred embryos got implanted. As far as we know, this is a new weakly supervised classification problem that has not been addressed in the related literature. The training dataset —Fig. 6.2(d)— is divided in a subset of labeled and another subset of unlabeled examples (non-transferred and transferred embryos, respectively). The unlabeled examples are grouped in bags (embryos transferred in the same cycle) and, for each bag, the number of positive and the number of unlabeled examples are provided. Since it can be seen as a combination of the weak supervision scenarios of the PU and LP paradigms, henceforth we will refer to this framework as the *positive unlabeled proportions* (PUP) problem.

The aim of the classifiers learnt with both alternative approaches is not to predict the ART procedure success but to allow specialists to identify, in the first case, valid cycle configurations and promising embryos in the second case. Combined, they can be used to arrange a promising cycle that is predicted to fulfill two basic requirements of a hypothetical pregnancy. All the relationships among the four approaches are graphically described in Figure 6.3.

In this chapter, an integral analysis of an ART case study is carried out using the exposed four approaches and their respective (weakly) supervised classification frameworks. Next, specific solutions for learning Bayesian network classifiers in each approach are explained. Then, a large set of experiments designed to explore the predictive ability of the classifiers learnt in each

approach is presented, their results are discussed and a set of conclusions is drawn. Several interesting future research lines are noted.

6.2 Material and methods

6.2.1 Data

The database has been collected by the Unit of Assisted Reproduction of the Hospital Donostia (Spain) during 18 months (January 2013 - July 2014). The population consisted of 375 consecutive patients participating in the IVF-ICSI program of Hospital Donostia. A total of 1835 embryos were analyzed. Among them, all the selected embryos were transferred on day *D+2* (48 hours after fertilization). The embryo selection criteria of the Association for Reproduction Biology Studies [3] (ASEBIR, according to its initials in Spanish), extensively used by assisted reproduction units in Spain, was followed.

The database is composed of a table of *cycles* and a separate table that collects the respective *embryos*. The examples of the tables are related by a one-to-n relationship: Each cycle with the set of oocytes extracted in that procedure. In the table of cycles, the set of collected features includes characteristics of the female and the male, stimulation treatment, statistics of the collected embryos and outcome variables (See Table 6.2 for a complete list). On the other hand, the features for describing embryos (Table 6.3) are oocyte/embryonic quality grades, oocyte/embryonic morphological characteristics and outcome variables. A complete description of the database can be found in the webpage associated with this study[1].

The database has been processed and specifically adapted to each of the four frameworks. Table 6.4 shows the description of the dataset that is provided for each approach. Note that implantation and implantable approaches are evaluated twice: using embryonic or cycle/embryonic features as predictive variables.

[1] `http://www.sc.ehu.es/ccwbayes/members/jeronimo/art`

Dataset	Predictors	Examples (-/+/?)	Bags (full)
1 Pregnancy	26	330 : 217/113/0	–
2 Implantation	14/40	696 : 447/72/177	330 (256)
3 Pregnanble	24	375 : 0/158/217	–
4 Implantable	14/38	1835 : 741/367/727	375 (41)

Table 6.4. Description of the dataset obtained from the data collected in our case study for each approach.

6.2.2 Protocol

The IVF management mostly consisted of GnRH antagonist protocol. Briefly, the suppression of pituitary FSH and LH secretion was performed with 0.25 mg cetrorelix (Cetrotide; Asta Medica, Frankfurt, Germany) administered daily when two or more follicles reached 13-14 mm in diameter. In some cases, down-regulation with a GnRH analogue, triptorelin acetate (Synarel, laboratorios Seid) on a long protocol was performed. Ovarian stimulation was performed with recombinant FSH (Gonal F, Merck Serono), highly purified urinary FSH (Angelini) or highly purified urinary menopausal gonadotropins (Menopur, Ferring) depending on the characteristics of each patient. The doses of hMG and FSH have been adjusted according to the ovarian response. Ovulation was triggered with 250 mg Ovitrelle (Serono) and transvaginal ultrasound-guided oocyte retrieval was scheduled 36 hours after the hCG injection. Oocytes were inseminated 4 hours after retrieval. Concerning the choice of insemination technique, ICSI was performed in cases with less than 1.5 million motile sperm recovered after capacitation, low rates of fertilization ($< 30\%$) in a previous IVF cycle and/or previous intrauterine insemination failures. In the remaining cases, conventional IVF was used. Finally, embryo transfer was performed on day 2 and the luteal phase was supplemented with micronized progesterone (Utrogestan, Laboratorios Seid or Progeffik, laboratorios Effik), vaginally 200 mg/12 hours. Pregnancy test was carried out 14 days after embryo transference.

6.2.3 Methodologies for learning Bayesian network classifiers in the four approaches

The four approaches designed in this analysis require specific methodologies for learning Bayesian network classifiers (Section 3.4). In the context of the ARTs, several authors have already used BNCs as probabilistic classifiers. Sometimes, authors are able to provide a graph structure designed by expert knowledge [33]. In our work, similarly to Morales et al. [126], both the structure and the parameters of the model are learnt from the data. Specifically, three kinds of Bayesian network classifiers have been considered: naive Bayes (NB) [76], tree augmented naive Bayes (TAN) [58] and K-dependence Bayesian network (KDB) [152]. Based on the assumption of conditional independence between the predictive variables given the class variable, the naive Bayes presents the simplest network structure (see Figure 3.3). TAN and KDB are the next step forward in terms of network structure complexity and allow the models to capture some conditional dependencies between predictive variables.

As the four approaches are represented by different (weakly) supervised classification problems, specific techniques have been applied to solve each of them. In the case of the cycle pregnancy prediction approach, where the training examples are fully labeled, standard supervised classification techniques

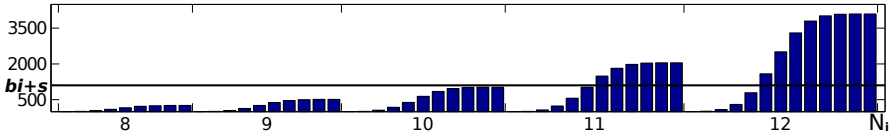


Fig. 6.4. Graphical summary of the number of consistent completions of a bag as its size (N_i) increases and the *minimum* number of positive examples decreases ($N_{i+} = \{N_i, N_i - 1, \ldots, 0\}$). The horizontal black line marks the threshold (in this study, $bi + s = 1100$) for applying a MCMC approximation to estimate the labeling.

(Section 3.4) can be used for learning the three considered types of BNCs. The methodology based on the Structural Expectation-Maximization (SEM) [45] strategy proposed in Chapter 4 for learning BNCs from label proportions has been used for dealing with the implantation prediction approach. Specifically, the referred three types of BNCs have been learnt using the PMEM version. In order to solve the pregnanble prediction approach, the methodology proposed by Calvo et al. [18] for learning BNCs from the respective positive-unlabeled dataset has been used. This is a wrapper method that combines the learning of BNCs with the adjustment of the parameter p, which indicates the proportion of positive examples in the unlabeled dataset.

Learning Bayesian network classifiers from positive-unlabeled proportions. The fourth approach, the implantable prediction, is represented by a different weakly supervised classification framework that, to our knowledge, is novel in the related literature: the learning from positive and unlabeled proportions (formally described in Section 2.4.3). In this novel weak supervised model, the examples in the dataset D_{PUP} are provided grouped in b bags where each bag provides a value $N_{i+} \leq N_i$ which indicates the *minimum* number of instances in $\boldsymbol{B}_i$ which have a positive class label. Fortunately, in this specific application an additional set of fully labeled examples D_L is provided. Therefore, as shown in Fig. 6.2(d), the final training dataset D is the union of the referred D_{PUP} and D_L sets.

Inspired by our solution to the label proportions framework (Chapter 4), we propose a SEM-based method for learning BNCs for this novel paradigm. At each iteration of the EM procedure and for each bag, the probability of all the *consistent* joint assignments of class labels to the instances of the bag (any N_i-tuple of labels that contains at least N_{i+} positive class labels, $2^{N_i} - \sum_{h=0}^{N_{i+}-1} \binom{N_i}{h}$ in total) is obtained using the current fit of the model. Then, the dataset is completed by giving to each instance $\boldsymbol{x}^j$ and class label c a weight equal to the average probability of all the possible completions in which instance $\boldsymbol{x}^j$ has been labeled with class label c. Specifically, the developed technique is equivalent to the PMEM version of the method proposed in Chapter 4, using a MCMC-based procedure for the most uncertain scenarios. However, in this study no MCMC approximation has been necessary for

estimating a probabilistic consistent completion. Remember that the MCMC approximation is calculated when the number of consistent completions of a bag is larger than the addition of the MCMC parameters, bi and s (Figure 6.4). Since the maximum bag size in the case study is $N_i = 3$, the method can always calculate the exact probabilistic completion.

6.3 Results

A complete experimental setting has been designed to test the four methodologies presented for solving the different approaches. It tries to shed light on the validity of each approach and the predictive ability of the learnt classifiers. The exposed three types of BNCs (Fig. 3.3) have been learnt for each experimental configuration. All the continuous variables have been discretized using equal-frequency with 3 intervals. With respect to FSS, a multivariate and univariate strategies have been used. The former applies correlation-based feature subset selection [74] (with both backward and forward search strategies) to obtain a subset of non-redundant predictive variables highly correlated with the class variable. The latter carries out chi-square statistical tests between the class variable and each predictive variable, and uses the resulting p-values to build an order of relevant predictive variables. Different experiments have been carried out using the subset of the t most relevant variables ($t \in \{n, \ldots, max(n_p, 2)\}$, where n_p is the number of predictive variables with a p-value > 0.05). The methods implemented for implantation and implantable prediction approaches, which are based on the iterative EM strategy, stop when the relative mean difference between the parameters of two models learnt in consecutive iterations is below 0.1%, or when 200 iterations are completed. All the experiments have been validated using a (stratified, when possible) 10×5 fold cross validation (CV). For the sake of clarity, only the results of the best experimental configuration for each BN classifier are shown. The complete tables of results are publicly available in the webpage associated with this study.

Remember that the objective of these experiments is to show the predictive ability of the data collected in the laboratories, which is supporting the decision making process of the physicians in their daily practice. Learnt classifiers have not been specifically tuned in any of the experiments or approaches. Alternatives to enhance the performance of the learnt classifiers (e.g., probabilistic classifier with the boundary not in 0.5 or loss functions that penalize asymmetrically false positives and negatives) have not been considered. An in-depth learning methodology to maximize a specific quality metric, subject to the specialist's preferences, is beyond the scope of this work.

6.3.1 Pregnancy prediction

The first set of experiments tests the performance of the methodology presented for solving the cycle pregnancy prediction approach. Table 6.5 shows

BNC	Accuracy	Recall	Precision	F1	PPR
NB	0.64 ± 0.06	0.45 ± 0.09	0.48 ± 0.10	0.46 ± 0.07	0.33 ± 0.07
TAN	0.65 ± 0.05	0.30 ± 0.07	0.49 ± 0.11	0.37 ± 0.08	0.21 ± 0.05
2DB	0.65 ± 0.05	0.35 ± 0.09	0.49 ± 0.12	0.40 ± 0.08	0.25 ± 0.07

Table 6.5. Results of the *pregnancy* approach in terms of accuracy, recall, precision, F1 and predicted positive rate (PPR) metrics. Each row shows experiments that learn different BNCs: NB, TAN and 2DB.

the results of the experiments using four different performing measures: accuracy, recall, precision and F1 [169]. Evaluating the classifiers only by means of accuracy could be unfair as, according to the results, the simplest classifier (which always predicts the majority class) would be the most accurate classifier. Recall and precision metrics, which provide information on the ability to predict positive examples, have been used. In order to fairly analyze these results, the last column shows the percentage of instances predicted as positive (PPR). It should be highlighted that TAN classifiers reach precision values of 0.5, meaning that half of the predicted pregnancies are real pregnancies. The recall of NB classifiers is also close to 0.5, which indicates that almost half of the real pregnancies were correctly identified. In terms of F1, a metric that combines both recall and precision, NB models obtain the best results.

6.3.2 Implantation prediction

In this approach, the embryos in *full* bags (those that represent cycles where all or no embryo became implanted) are actually labeled. These have been used for calculating the ranking of relevant variables and for evaluation. A leave-one-*full-bag*-out procedure has been used, which takes at each iteration a full bag as the validation set. The CFS feature selection has been carried

BNC	Accuracy	Recall	Precision	F1	PPR
NB	0.86 ± 0.00	0.03 ± 0.00	0.40 ± 0.00	0.05 ± 0.00	0.01 ± 0.00
TAN	0.83 ± 0.00	0.04 ± 0.00	0.14 ± 0.00	0.06 ± 0.00	0.04 ± 0.00
2DB	0.85 ± 0.00	0.08 ± 0.00	0.30 ± 0.00	0.13 ± 0.00	0.04 ± 0.00
NB	0.76 ± 0.00	0.49 ± 0.00	0.29 ± 0.00	0.36 ± 0.00	0.23 ± 0.00
TAN	0.80 ± 0.00	0.27 ± 0.01	0.27 ± 0.01	0.27 ± 0.01	0.14 ± 0.00
2DB	0.78 ± 0.00	0.28 ± 0.00	0.24 ± 0.00	0.26 ± 0.00	0.16 ± 0.00

Table 6.6. Results of the *implantation* approach in terms of accuracy, recall, precision, F1 and predicted positive rate (PPR) metrics. Each row shows experiments that learn different BNCs: NB, TAN and 2DB. The results of experiments performed with two different datasets are shown: in upper rows, only the embryonic features are used as predictive variables, and in lower rows, the cycle features are also included.

BNC	psRecall	psF1	Lee	PPR
NB	0.88 ± 0.06	0.61 ± 0.04	1.00 ± 0.14	0.78 ± 0.07
TAN	0.89 ± 0.06	0.61 ± 0.05	1.00 ± 0.15	0.79 ± 0.07
2DB	0.86 ± 0.06	0.60 ± 0.05	0.96 ± 0.14	0.78 ± 0.06

Table 6.7. Results of the *pregnanble* approach in terms of psRecall, psF1 [18], Lee [108] and predicted positive rate (PPR) metrics. Each row shows experiments that learn different BNCs: NB, TAN and 2DB.

out using a transformed dataset completed according to the label proportions of the bags.

The same four measures used in the previous set of experiments (accuracy, recall, precision and F1) characterize the results of this set of experiments in Table 6.6. The horizontal division indicates the use of two different datasets: (1) a dataset with just embryonic features as predictive variables and (2) a dataset with embryonic and cycle features as predictive variables. Both datasets are even more unbalanced than the dataset of the previous approach (see Table 6.4) and the classifiers learnt only with embryonic predictors show no predictive ability: PPR values are often 0 and they reach, at most, a poor 0.04. When the training dataset includes predictive variables of the cycle the proportion of predicted positives rises notably, also improving the predictive ability. NB classifiers stand out with the best results in terms of recall (values close to 0.5), precision (values close to 0.3) and F1 (values over 0.3).

6.3.3 Pregnanble prediction

In this set of experiments in a positive-unlabeled framework, the wrapper procedure that tunes parameter p (the real unknown proportion of positive examples in the unlabeled subset) was set up with 3 repetitions and 5 candidate values. A 5×5 fold CV is used for validation in this inner tuning loop, respecting the proportion of labeled examples in the division into CV folds of the dataset. Regarding the FSS setting, an adaptation to the PU framework of the standard CFS [20] has been implemented. In order to perform the chi-square statistical tests, the unlabeled instances were considered as negatives.

In this approach, unlabeled data *has* to be used for evaluation and, therefore, the standard evaluation metrics are not valid. Table 6.7 shows the results of the performed experiments in terms of three different performing metrics: *psRecall* (proportion of examples in the labeled subset predicted as positive), *psF1* [18] (an F1-based metric for PU) and Lee [108] metrics. In this case, there are few differences among the three kinds of classifiers. The classifiers identify as positives $70 - 80\%$ of the validation cycles, which is almost twice the rate of labeled (positive) instances. Thus, the psRecall values are generally larger than 0.8, which would indicate that the classifiers are able to identify a majority of positive examples. Classification performance is evaluated in

BNC	Accuracy	Recall	Precision	F1	PPR
NB	0.69 ± 0.03	0.53 ± 0.05	0.53 ± 0.05	0.53 ± 0.04	0.33 ± 0.04
TAN	0.69 ± 0.04	0.51 ± 0.08	0.54 ± 0.05	0.52 ± 0.05	0.31 ± 0.05
2DB	0.69 ± 0.03	0.53 ± 0.06	0.54 ± 0.06	0.53 ± 0.04	0.33 ± 0.04
NB	0.70 ± 0.03	0.57 ± 0.05	0.54 ± 0.05	0.56 ± 0.04	0.35 ± 0.03
TAN	0.68 ± 0.03	0.49 ± 0.06	0.52 ± 0.05	0.50 ± 0.04	0.31 ± 0.03
2DB	0.66 ± 0.03	0.49 ± 0.06	0.49 ± 0.06	0.48 ± 0.04	0.33 ± 0.05

Table 6.8. Results of the *implantable* approach in terms of accuracy, recall, precision, F1 and predicted positive rate (PPR) metrics. Each row shows experiments that learn different BNCs: NB, TAN and 2DB. The results of experiments performed with two different datasets are shown: in upper rows, only the embryonic features are used as predictive variables, and in lower rows, the cycle features are also included.

other approaches by means of recall-precision. In the PU paradigm, the precision metric cannot easily be estimated/approximated. Given that the F1 metric is the harmonic mean of precision and recall, the psF1 values (significantly lower than psRecall) could indicate a reduced performance in terms of precision (large false positive rate).

6.3.4 Implantable prediction

In this approach, both transferred and non-transferred embryos have been used as training examples. The non-transferred examples, which are certainly labeled, have been used for calculating the ranking of relevant variables and for evaluation. The CFS-based feature selection has been carried out using a dataset completion that fulfills the proportions of positive examples, N_{i+}.

Once again, accuracy, recall, precision and F1 metrics depict the results of these experiments in Table 6.8. As explained in Section 3.2, two datasets (the first one using only the embryonic features as predictive variables and the second one including variables of the associated ART cycle) are used. In this case, the classifiers learnt with both datasets show a similar behavior. The proportion of predicted positives (PPR) is generally larger than 0.3, and precision, recall and F1 metrics usually exceed 0.5. NB classifiers are the classifiers that show the best results, although TAN and 2DB classifiers are competitive when the dataset with only embryonic predictive variables is used.

6.4 Discussion

The main objective of the assisted reproduction units is the improvement of the pregnancy rate of the ARTs. We propose an integral solution for the ART problem based on ML techniques which, taking into account all the information collected by physicians, learn classifiers which could be used to

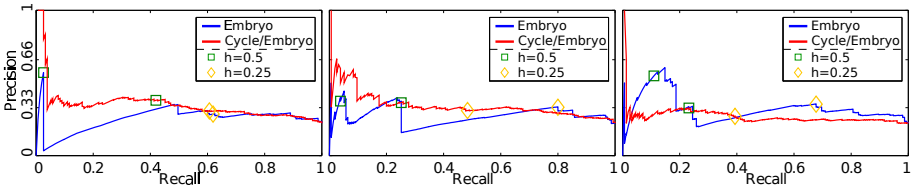


Fig. 6.5. Precision-recall curve of the best classifiers (NB, TAN and 2DB from left to right) learnt from the datasets with embryonic predictive variables (blue line) and cycle-embryonic predictive variables (red line). Positive examples in the non-full bags are assumed to be those with higher probability. The highlighted points represent the classifiers using 0.25 and 0.5 as 0/1 class decision threshold, h.

improve the implantation (and pregnancy) rates. Our integral proposal deals with up to four different subproblems, both previous and posterior to embryo implantation, to understand the predictive capability of the collected data. Classification techniques take advantage of all the available data. All the collected features are considered although only a subset of relevant (regarding the class) and non-redundant (among them) variables are finally used for model learning. Thus, no personal preference determines the allegedly relevant variables to be considered. Similarly, even the embryos/cycles of unknown fate are considered: Weakly supervised techniques take advantage of the unlabeled examples simultaneously guessing their category and learning the model.

The straightforward strategy to solve the ART problem using ML techniques consists of learning classifiers that predict whether an ART cycle will end up in a pregnancy. According to Table 6.5, the BNCs learnt in this approach have obtained significant results in our case study. One out of three cycles is predicted as positive, that is, pregnancies are predicted in a realistic proportion (see Table 6.4). Moreover, almost half of the real pregnancies are correctly identified by learnt models ($recall \approx 0.45$) and one out of two cycles is correctly predicted as pregnancy ($precision \approx 0.5$). To sum up, two out of three cycles are accurately classified. Although moderate, these results are promising since, as previously explained, tuning the learning process — customized to the specialist's preferences— can potentially produce classifiers of enhanced performance.

This first approach does not consider embryos individually, but globally as another feature of the cycle that has to be configured. This approach could be used, for instance, to optimize the *number* of embryos to transfer in a given cycle (*No.Transf.Emb* variable in Table 6.2). However, this first approach cannot handle information about the individual embryos, which is essential to identify promising embryos and select those to transfer. For this task, our second approach is a weakly supervised framework (label proportions [81]) that, using past embryos of both known and unknown fate, learns classifiers to predict the implantation of an embryo. As hypothesized, the learnt classifiers show a reduced performance, revealing the difficulty of predicting an embryo im-

plantation based on the data collected in this case study. The results in Table 6.6 vary sharply depending on the set of predictive variables used to describe the examples (embryos). On the one hand, the results of the classifiers learnt from the dataset described just by embryonic variables are poor: almost no implantation is predicted. On the other hand, when the second dataset that combines the embryonic and cycle features as predictive variables is used, learnt classifiers show a significant improvement in terms of all the metrics. Classifiers learnt from the second dataset achieve a realistic proportion of examples predicted as positive (predicted implantations). In terms of recall, half of the real implanted embryos are predicted as positive. And almost one out of three positively examples predicted as positive are real implanted embryos. This could have been understood as evidence of the low power of the collected (morphological) embryonic variables to predict an implantation. Only with the inclusion of variables describing the respective cycle does the performance of the learnt classifiers improve. Despite this dramatic difference in the results of Table 6.6, the classifiers learnt from both datasets show a similar ability to assign a variety of different posterior probabilities to the embryos used for evaluation. It can be seen in Figure 6.5 that, if the threshold of the classifiers learnt only with embryonic variables were tuned to optimize a metric (in this case, recall or precision), their performance would match up that of the classifiers learnt with both cycle and embryonic variables. These results remove all doubts about the embryonic variables: Their contribution is determinant in the implantation prediction. However, the area under the curve in Figure 6.5 is significantly larger (specifically in the case of NB and TAN models) when the cycle variables are also considered. This demonstrates the asserted contribution of the cycle information to the identification of promising embryos for implantation [1, 33, 146, 171].

Both alternative approaches, pregnanble and implantable prediction, borrow the idea of identifying promising cycles and embryos from the EU model [171], respectively, independently and previous to the implantation process [71]. As explained before, failed cases cannot always be annotated as negative examples according to this point of view. Incidentally, this allows us to avoid considering the possibility of MIF occurrence. A wrong representation of the problems could limit the predictive ability of classifiers of the pregnancy and implantation approaches. The design of the classification tasks represented by both alternative approaches fits the collected data in a more natural way than those of the two first approaches. Therefore, these should produce classifiers of enhanced performance regarding those of their equivalent pregnancy and implantation approaches.

On the one hand, the lack of negative examples in the pregnanble prediction approach makes its comparison with the equivalent pregnancy prediction approach difficult. The psF1 results are hardly comparable with the real F1 values obtained in the pregnancy prediction approach. The psF1 metric, which can be reliably used for model selection given that it is proportional to the real F1 metric [18], does not approximate the real value. psRecall, a solid approx-

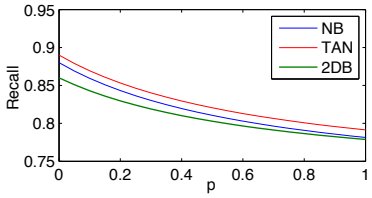


Fig. 6.6. Estimation of the real recall value for results of Table 6.7 as p value (assumed rate of positive examples in the unlabeled subset) increases. The formulae $rec = (PPR_+ \cdot p_+ + p \cdot PPR_? \cdot p_?)/(PPR_+ + p \cdot PPR_?)$ is used, where PPR_+ and $PPR_?$ are the rates of predicted positives in the labeled and unlabeled subsets, respectively. p_+ and $p_?$ are the relative size of each subset regarding the whole dataset.

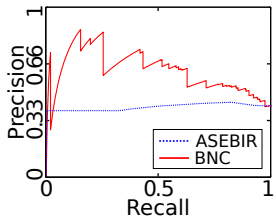
imation of the real recall metric, can be more easily interpreted. According to Figure 6.6, the results of this approach are significantly better than those of its equivalent pregnancy prediction approach in terms of recall: Even in the worst case ($p \to 1$), the classifiers of the pregnanble prediction approach show larger recall values ($0.78 >> 0.49$). That is, a large proportion of the real good prognostic cycles are identified as positives. Additionally, the significantly lower psF1 values in comparison with those of psRecall could reveal the reduced precision of the learnt classifiers, since the F1 metric is the harmonic mean of precision and recall. That is, a considerable number of false positives (real poor prognostic cycles predicted as positives) could be contributing to increasing the rate of predicted positives (PPR). In practice, learnt classifiers might seem quite uninformative since too many cycles are predicted as good prognoses. However, if these classifiers are optimized to accurately identify the negative case (i.e., poor prognostic cycles), a negative prediction of our classifiers could be reliably understood as a recommendation to revise the configuration of the cycle (e.g., stimulation treatment or number of embryos to transfer). In order to do so efficiently, high precision values for the negative class label are required, which unfortunately cannot be precisely estimated in a positive-unlabeled framework.

On the other hand and focusing again on the embryos, the implantable prediction approach undeniably improves the results of its equivalent implantation prediction scheme. The behavior of the classifiers is better in terms of all the metrics (recall values are 0.1 points higher on average, prediction values are doubled and F1 values are 0.2 points higher). Contrary to their previously commented influence on the results of the implantation approach, the results of the classifiers that also consider the cycle features as predictive variables barely improves in 0.03 points the results of the classifiers learnt only with embryonic variables. This dissimilar behavior corroborates the definition of the approach: the promising development of an embryo can be assessed previously and independently from the implantation procedure. Thus, the features describing the embryo are the only relevant variables for this task. However, the obtained results prove that the predictive capability of the collected em-

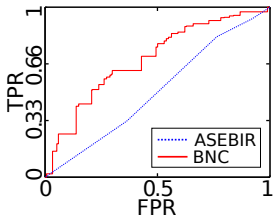
bryonic features is not impressive. The features collected in our case study for the oocytes/embryos, those recommended by the ASEBIR [3], are a set of morphological variables. Several authors claim that the predictive capability of scores exclusively based on morphological factors is limited [1, 49, 71]. The search, study and collection of other non-morphological features such as pre-implantation genetic diagnosis, embryo metabolomic and proteomic analysis, embryo morphokinetics analysis or endometrial receptivity tests have been proposed in the related literature [11, 10, 65, 95, 123, 134, 149, 158, 163]. This research line will surely allow authors to progress in the answer of this open question; its solution is expected to bring a significant leap forward in the ability to predict an embryo implantation and, consequently, in the performance of the ARTs. The comparison of the results of both implantation and implantable approaches supports the thought that the currently collected data explains, to a larger extent, the promising embryo development, rather than the embryo implantation. That is, other factors influence the implantation process, the uterus receptivity (good prognostic cycles, represented by the pregnanble approach) or MIF occurrence. In any case, a better understanding of the mechanisms regulating the embryo implantation is needed.

Based on the obtained results, it can be asserted that a recommender system for the ART problem based on the presented approaches would provide valuable information that could imply an improvement in the selection

(a) List of all the embryos horizontally ordered according to their probability of successful development and vertically depending on their ASEBIR grade.



(b) PR curve drawn using embryos of known fate.



(c) ROC-curve drawn using embryos of known fate.

Fig. 6.7. Graphical comparison of a BNC learnt with the implantable approach and the ASEBIR quality grade [3].

of promising embryos and/or the configuration of good prognostic cycles. As shown in Figure 6.7(a), ordering embryos according to their probability of good development (implantable approach) does not completely match up with the ASEBIR ranking. ASEBIR proposes an ordinal four-categories ranking (A, B, C and D), where A and D respectively indicate the best and worst embryos. In detail, our classifiers generally agree with ASEBIR's criteria on identifying embryos of highest and lowest quality (A and D embryos). However, numerous quality C embryos are considered by our classification model more promising than a substantial set of quality B embryos. Both the precision-recall (PR) and the receiver operator characteristic (ROC) curves in Figure 6.7 graphically show the enhanced performance of our model with respect to the ASEBIR ranking. The observed disagreement in medium-quality embryos was not surprising since, as has been previously reported [53, 71], the most difficult task is not the identification of the highly promising embryos, but the classification of those of medium-quality. The reordering suggested by our ML models is supported by the daily practice of our group of physicians, who already consider an analogous variation of the ASEBIR criteria based on their direct observation of the evaluated embryos. Although the classifiers learnt with this implantable approach do not predict the implantation, the transference of embryos with promising development is associated with higher implantation rates [49, 53, 71]. Therefore, this fourth approach can provide a novel selection criteria in order to choose the embryos to transfer.

6.5 Conclusions and future work

A case study of the ART problem is analyzed in an integral way throughout the use of ML techniques. The reinterpretation of the objectives of the ART problem has led us to design four different (novel) approaches which are solved by means of weakly supervision classification techniques, considering also examples of uncertain fate for model learning. Specific solutions that learn classification models (BNCs) taking the most of the available weakly supervised data have been developed for each of these four approaches.

Our solution uses all the information collected by physicians during the whole ART procedure and evaluates its predictive capability in the four presented classification tasks. The learnt classifiers that predict the viability of a cycle show a promising performance. According to the results, the data collected for our case study cannot fully describe an embryo implantation, although the inclusion of the cycle features enhances the performance of the classifiers that predict embryo implantation. Collected data fits better the description of embryo development. Obtained classifiers have been proved to rank the medium-quality embryos of our case study more consistently than ASEBIR grade. The probabilistic assessment of the classifiers obtained for the alternative approaches can be consistently used for cycle configuration and embryo quality grading.

7

Defect classification

In software engineering, software defect classification can be useful for several tasks such as prioritizing software bugs or defect prediction. However, performing such a classification is difficult and generally involves extensive manual effort. These difficulties are usually reflected in the incomplete, noisy and erroneous labeling of the reported defects. Machine learning techniques have already been applied to automatically classify software defects. However, learning from a dataset labeled by a single subjective annotator, which does not represent the ground truth, can perform poorly. Defect classification, and all the problems where a set of partially reliable annotations can be collected, is a suitable real application to be solved by means of crowd learning techniques.

In this chapter, we apply an adaptation of the learning from crowds methodology presented in Chapter 5 to deal with the defect classification problem, an approximation that, to the extent of our knowledge, is novel in the related literature. To illustrate our proposal, a real application of IBM's orthogonal defect classification problem working on the issue tracking system of the Compendium software tool has been analyzed. The inferred framework can be described as a nine-label multi-class imbalance classification problem labeled by a set of five annotators. Apart from our learning from crowds methodology (Chapter 5) adapted to multi-class frameworks, other two approaches are considered, each of them focus on a different characteristic of the problem. On the one hand, a binary decomposition strategy specifically tries to alleviate the additional complexity of multi-class framework. On the other hand, a sampling strategy aims to deal with the class imbalance nature of our problem. The three proposed techniques, all of them adapted to the learning from crowds paradigm, are tested in a set of experiments where the majority voting strategy is used as a baseline. In general, the results of the designed experimental work show the enhanced performance of our crowd learning solutions regarding classical supervised learning using the most-voted labels.

7.1 Introduction

Defect classification is an important task during maintenance used for defect prioritisation, faster and cheaper defect resolution and analysis of module and component quality. However, it is a time-consuming task which has been traditionally performed manually by members of the developer team.

Recently, artificial intelligence based techniques, such as supervised classification, have been used to solve the problem of defect classification [88]. Standard supervised classification techniques infer the categorizing behavior of a problem of interest from a set of previous examples. Each example, which describes a specific case of the problem by means of a set of features, is provided together with its real category (class label). Given a set of *certainly labeled* examples, standard learning techniques produce classifiers that anticipate the category of new unlabeled examples (defects).

However, defect classification is quite a subjective task and achieving such a reliably labeled dataset is difficult. It is common that two different annotators disagree in categorizing the same report. Disagreement may happen, for example, due to the different personal expertise of the annotators on the specific fields or lack of a global perspective. Learning a classifier from a set of examples labeled by a single annotator can lead to reproducing the possibly inaccurate categorizations of that annotator. According to Lugosi's study of sources of error [114], taking into account the opinion (labeling) of several annotators for each example is a straightforward way to deal with the problem of learning from a single unreliable annotator. This is the basic idea behind the learning from crowds paradigm [141, 82], which combines the annotations of many different (non-expert) annotators to learn a classifier for the problem of interest.

In this study, we formulate this research question: *can we learn to classify defects using a set of (non-expert) opinions?* We address the research question proposing the use of machine learning techniques to infer a classification model based on the *novel* learning from crowds paradigm. To do so, an illustrative real application of the defect classification problem is analyzed. A dataset with defects/requirements reported in the Bugzilla tracking system of the Compendium project[1] was used. A total of 5 annotators, with experience in computer science, labeled the collected examples according to their subjective point of view. Each annotator associated each example to a category (impact) of the Orthogonal Defect Classification [29], a 13-category taxonomy that allows developers to separate defects depending on their impact on the costumer. The resulting dataset can be described as that of a *multi-class imbalance classification problem annotated by a crowd*. We propose a diverse set of approaches, all of them based on the learning from crowds paradigm, for specifically dealing with different aspects of the problem:

- A standard crowd learning approach.

[1] `http://compendium.open.ac.uk/bugzilla/`

- The adaptation of a *decomposition strategy*, which explicitly deals with the multi-class nature of our problem.
- An adaptation of a *sampling strategy*, which explicitly considers the unbalanced nature of this multi-class problem.

The rest of the chapter is organized as follows. In the next section, a throughout description of the defect classification problem is given and our illustrative real application is characterized. Next, the different approaches proposed to solve the problem are presented. Then, the experimental work is presented and the results are shown and discussed. Section 6 enumerates the threats to validity and, finally, some conclusions and future work are presented.

7.2 Background

A software defect or bug is a non fulfillment of intended usage requirements [93]. For each defect, a report is usually generated throughout an issue tracking system. A defect report is a clear description of the issue which can be used to replicate and fix the problem.

7.2.1 Defect Classification

The classification of software defects aims to capture the semantics of the reports of each type of defect. Software defect classification is valuable for several tasks such as prioritising software defects, improvement of defect prediction, assignment of defects to developers (team management), defect resolution, identifying the quality of modules or components, etc.

There is a large amount of literature related to defect classification since the seminal work by Endres [51] in 1975. Multiple models, variations and customisation of the initial taxonomies have been proposed by Demillo and Mathur [44], Grady [69], the IEEE Standard Classification for Software Anomalies [92], HP's Defect Origins, Types, and Modes [55] and the Orthogonal Defect Classification (ODC) [29]. Wagner summarized in a short position paper the work carried out on defect classification approaches and proposed a set of challenges [188]. Recently, Hall et al. [75] have analysed the literature and proposed a comprehensive taxonomy.

Orthogonal Defect Classification. Among all these taxonomies, IBM's ODC is the most popular classification scheme, in spite of having been criticized due to a variety of alleged drawbacks such as being neither fully orthogonal nor consistent in the terminology [157], difficulties to apply in practice [55], and complicated to customise to specific contexts [130, 48]. In a controlled experiment with students, Falessi and Cantone [56] also reported that there is affinity between some ODC defect types, that it is inefficient and that previous

training is needed to apply it. Nevertheless, IBM and other organisations have applied ODC to improve software development processes [17, 170, 9, 120].

ODC consists of a well defined four step process that must be followed: (i) *classify*, as data collection step; (ii) *validate*, to provide feedback based on the review of the classified defects; (iii) *assess* of ODC attributes and defect trend analysis; and (iv) *act* to implement the actions. According to IBM [7], ODC complements a defect report with a well-defined set of data. When a defect is reported following the ODC process, three attributes have to be added: (i) *ODC activity*, such as design review, unit test, etc.; (ii) *ODC trigger*, which is the environment or condition that led to the failure; and (iii) *ODC Defect Impact*, which relates the impact of the software defect on customer satisfaction. ODC Impact can be used besides severity to focus quality improvement effort on reducing the defects that most significantly impact customer satisfaction (as opposed to reducing the total number of defects). Once a defect is closed, further information is added: (i) *target* or entity fixed; (ii) *defect type*; (iii) *qualifier*; (iv) *source*; and (v) *age* or history of the entity fixed.

Typically, developers manually classify defects into the ODC categories based on the reported descriptions. However, performing such a classification is difficult and generally involves extensive manual effort using, for example, root-cause defect analysis (RCA) [110, 15] Recently, the application of machine learning techniques to software engineering problems is increasingly being explored. For example, reported data has been used to automatically distinguish between defects and requirements as, although these typically follow a similar software development life-cycle during maintenance, it is important to differentiate them [85]. Antoniol et al. [2] reported on the classification of reports during enhancement work or other kinds of activities. The authors achieved between 77% and 82% of correct classification using decision trees, naive Bayes and logistic regression.

Machine learning techniques have been also applied to the domain of mobile applications. Thus, Maalej and Nabil [116] classify app reviews into four categories: bug reports, feature requests, user experiences and ratings. Natural language processing techniques are applied to the manually annotated reviews. Their methodology achieve large precision and recall values.

Closer to our work, Huang et al. [89] describe AutoODC, an automatic defect classification approach based on ODC that categorizes reports stored in a Bugzilla system. The authors propose a SVM classifier that makes predictions taking advantage of certain defect descriptions manually added by developers. Also, Thung et al. [177] classify ODC defects into three *super*-categories (control and data flow, structural, and non-functional) which cover all the ODC defect types.

Crowdsourcing vs. Learning from crowds. In software engineering, *crowdsourcing* [87] usually refers to outsource the software development to an undefined network of developers through Web platforms. However, the idea of using external annotators has been applied to select or prioritize requirements. For example, Morales-Ramirez et al. [127] describe CrowdIntent, a platform

for the identification of intentions in online discussions. In their experiment, 20 participants grouped in pairs annotated sentences from two online discussion sites. The authors used the Kappa statistic to compute the agreement among participants.

In machine learning, learning from crowds [141, 83] is a weakly supervised classification problem where the examples (defects) provided for training the classification model are unreliably categorized (i.e., no golden-truth is available). The provided examples are labeled by a set of annotators of unknown *trustfulness*. Such labeling shows disagreements among annotators. In this work, we propose to use the learning from crowds paradigm to improve the classification of defects according to their *impact*. Different annotators (stakeholders) with different opinions or knowledge about the impact of a particular type of defects are used to categorize the training examples.

7.2.2 The Compendium Dataset

The dataset is composed of reports collected from the reporting system of the Compendium project, a software tool for mapping information, ideas and arguments. Issue tracking systems (ITS) or bug reporting systems are typically used by software projects for reporting and tracking defects as well as proposing new functionalities. Nowadays, ITS are also used for other project management and infrastructure decisions and code reviews. ITS organise the information through *tickets* and track the life-cycle of each ticket since it was first recorded until it is closed. Each ticket maintains data such as an identifier, summary, description, opening/closing/modification dates, who reported the defect, priority, severity, environment, current status, etc. There are multiple open source ITS platforms. Examples include Bugzilla, Launchpad, GitHub and RedMine. The ITS used by the Compendium project is implemented in

Annotator	*Category Impact*								
	Installability	*Integ/Security*	*Migration*	*Reliability*	*Performance*	*Documentation*	*Requirements*	*Usability*	*Capability*
A_1	92	6	91	119	14	29	192	392	27
A_2	82	14	20	4	15	36	236	267	288
A_3	86	10	89	117	13	19	139	473	16
A_4	87	12	21	86	14	28	239	279	196
A_5	87	9	25	97	14	25	242	353	110

Table 7.1. No. of labels (categories) of each type assigned by each annotator to the 962 Bugzilla entries

Bugzilla and collects support issues, feature requests and bug reports from the Compendium community.

The collected dataset comprises 962 examples, all the entries available in August 2014 (with the exception of some obvious spam). For each defect, only the informative fields have been considered: severity, summary and description. Severity is a 3-value variable (Bug, Support or Feature) and the other two are text fields.

Five annotators with experience in computer science were asked to annotate the examples according to the 13-category ODC standard [29]. As also observed by Huang et al. [89], only 9 out of the original 13 categories were used by the annotators to annotate the examples of the dataset. Therefore, in this study only the following 9 categories have been considered: Installability, Integrity/Security, Migration, Reliability, Performance, Documentation, Requirements, Usability and Capability. Table 7.1 shows the number of examples that each annotator assigned to each class label. Although the number of examples assigned by the different annotators is almost the same for some class labels (Installability, Performance and, to a lesser extent, Integrity/Security and Documentation), there exists high variability in the majority of class labels. Moreover, a similar number of annotations does not imply consensus. Table 7.2 shows the assignment of examples to labels based on the consensus among annotators: each cell shows the number of examples assigned to a class label by a certain number (row) of annotators. The last row shows the number of examples in which the consensus label is supported by a majority of annotators (three or more). This row provides an insight into the lack of homogeneity in the distribution of class labels. Therefore, the inferred framework may be formulated as an multi-class imbalance problem labeled by a crowd of 5 annotators. Specifically, Integrity/Security, Migration, Performance and Documentation will be considered as minority classes.

7.3 Methods

In this analysis, supervised classification techniques are applied to improve the software defect classification problem. Three approaches have been designed, each of them learning Bayesian network classifiers (BNC, Section 3.4). Specifically, three kinds of BNCs have been considered: naive Bayes (NB) [76], tree augmented naive Bayes (TAN) [58] and K-dependence Bayesian network (KDB) [152]. Based on the assumption of conditional independence between the predictive variables given the class variable, the naive Bayes presents the simplest network structure (see Figure 3.3). TAN and KDB are the next step forward in terms of network structure complexity and allow models to capture some conditional dependencies between predictive variables. For our purpose, BNCs are particularly suitable: their interpretability is outstanding (influences and dependencies among variables can be deduced from the explicit proba-

Number of annots.	*Category Impact*								
	Installability	*Integ/Security*	*Migration*	*Reliability*	*Performance*	*Documentation*	*Requirements*	*Usability*	*Capability*
2	6	0	27	26	2	4	61	40	10
3	6	3	11	32	1	4	66	69	48
4	20	2	11	39	3	12	100	129	10
5	59	5	2	1	9	9	37	96	2
[3,5]	85	10	24	72	13	25	203	294	60

Table 7.2. Number of examples in which a (sub)set of annotators agree in the assigned class label. Each row involves a different number of annotators that support that labeling. The last row shows the number of examples in which the majority of annotators —three or more— agree in the assigned label.

bility relationships) and the developed techniques take advantage of their probabilistic classifications/assessments.

7.3.1 Multi-class learning from crowds

As formally defined in Section 2.4.4, in the learning from crowds paradigm, the real class label of the examples is unknown and only the subjective opinions of a set of non-expert annotators is available. The information of supervision of each instance $\boldsymbol{x}^i$ can be codified by a t-tuple $\boldsymbol{l}^i$, where $l_a^i \in \mathcal{C}$ indicates the class label assessed by annotator A_a and, thus, the dataset is composed of N examples $D = \{(\boldsymbol{x}^1, \boldsymbol{l}^1), (\boldsymbol{x}^2, \boldsymbol{l}^2), \ldots, (\boldsymbol{x}^N, \boldsymbol{l}^N)\}$.

Our Expectation-Maximization (EM) based method previously proposed for the multi-dimensional learning from crowds framework (Section 5.4) has been adapted to this unidimensional but multi-class problem. The EM strategy [45] allows us to combine the estimation of a set of weights that model the reliability of each annotator and the learning of the model using the labels provided by a crowd of annotators. In our method, the E-step estimates the reliability weights of the annotators and, in the M-step, the model parameters are re-estimated such that the likelihood is maximized given the data and the weights estimated in the E-step. Iteratively, both steps are repeated. When TAN or KDB classifiers are learnt, the Structural EM (Section 3.3.2), which adds an outer loop to the traditional EM procedure for combining model parameter estimation and structural learning, is used.

For this study, two types of reliability weights, which codify the integrity of the labeling provided by each annotator, have been considered. On the one hand, a reliability weight per class label and annotator is used. These

per-label weights (w_c^a, for all $a \in \{1,\ldots,t\}$) codify the reliability of each annotator A_a when they provide examples of a specific class label c. On the other hand, the *confusion-matrix* weights ($W_{cc'}^a$, for all $a \in \{1,\ldots,t\}$ and $c, c' \in \{1,\ldots,|\mathcal{C}|\}$) codify, for each annotator, both the reliability of an annotator when they predict a class label and the probability of the annotator providing label c when c' is the real label. The classical counting procedure for model parameter estimation has been adapted to incorporate the crowd information (multiple weighted labelings). In the next subsection, a detailed description of the adapted procedure is presented. Then, the estimation of the annotator reliability weights carried out in the E-step is explained.

7.3.1.1 Parameter estimation in multi-class learning from crowds

The standard parameter estimation procedure has been adapted to collect frequency counts from multiple noisy annotations per example, using the annotator reliability weights in order to carry out an informed aggregation of the different contributions. Similar to the method proposed in Section 5.4, the parameter estimation procedure to collect frequency counts integrating the multiple and weighted labels can be expressed as follows:

$$N(\boldsymbol{u}) = \sum_{(\boldsymbol{x},\boldsymbol{l})\in D} \sum_{c=1}^{|\mathcal{C}|} \mathbb{I}[u_1 = v_{J_1}, \ldots, u_k = v_{J_k}] \cdot F_c^{\boldsymbol{l}}$$

where $\mathbb{I}[condition]$ is a function that returns 1 if *condition* is true and 0 otherwise, $\boldsymbol{u} = (u_1,\ldots,u_k)$ is an instantiation of the random vector $\boldsymbol{U} = (V_{J_1},\ldots,V_{J_k})$, a sub-vector of the original $\boldsymbol{V} = (\boldsymbol{X}, C)$ with $\{J_1,\ldots,J_k\} \subseteq \{1,\ldots,n+1\}$. Finally, $F_c^{\boldsymbol{l}}$ is the *labeling reliability factor* corresponding to the instantiation $\boldsymbol{u}$ according to the opinion of the annotators and their reliability weights (constrained to $\sum_{c=1}^{|\mathcal{C}|} F_c^{\boldsymbol{l}} = 1$). Depending on the type of annotator reliability weights, the labeling reliability factor $F_c^{\boldsymbol{l}}$ is calculated in a different way. On the one hand, using the *per-label* weights (w_c^a), it is calculated as,

$$F_c^{\boldsymbol{l}} = \frac{\sum_{a=1}^{t} \mathbb{I}[l_a = c] \cdot w_c^a}{\sum_{c'=1}^{|\mathcal{C}|} \sum_{a=1}^{t} \mathbb{I}[l_a = c'] \cdot w_{c'}^a}$$

On the other hand, $F_c^{\boldsymbol{l}}$ is calculated using the *confusion-matrix* reliability weights ($W_{cc'}^a$) as follows,

$$F_c^{\boldsymbol{l}} = \frac{\sum_{a=1}^{t} \mathbb{I}[c \in \boldsymbol{l}] \cdot W_{l_a c}^a}{\sum_{c'=1}^{|\mathcal{C}|} \sum_{a=1}^{t} \mathbb{I}[c' \in \boldsymbol{l}] \cdot W_{l_a c'}^a}$$

7.3.1.2 Estimation of the reliability weights of the annotators

As shown in Section 5.3, a simple estimation of the reliability weights of the annotators, which only uses the available multiple labelings, is obtained by

means of the consensus criterion. In the case of *per-label* weights, the consensus weight of an annotator A_a in class label c is,

$$w_c^a = \frac{1}{\Phi} \sum_{i=1}^{N} \mathbb{I}[l_a^i = c] \frac{1}{t-1} \sum_{a' \neq a} \mathbb{I}[l_{a'}^i = c] \tag{7.1}$$

with a normalization factor Φ such that $\sum_{c=1}^{|\mathcal{C}|} w_c^a = 1$. In the case of *confusion-matrix* weights, the consensus weight of an annotator A_a in class label c is,

$$W_{cc'}^a = \frac{1}{\Phi} \sum_{i=1}^{N} \mathbb{I}[l_a^i = c] \frac{1}{t-1} \sum_{a' \neq a} \mathbb{I}[l_{a'}^i = c'] \tag{7.2}$$

Once a model fit $\mathbb{M}$ is available, both strategies to estimate the reliability weights of the annotators (Section 5.4) are considered: (1) an accuracy-based strategy (*Acc*), where the class label predicted by the model for each instance, $\hat{c} = \text{argmax}_{c \in \mathcal{C}}\, p_{\mathbb{M}}(\boldsymbol{x}, c)$, is used as golden truth, and (2) a probability-based strategy (*Prob*), which uses the *probability* given by the model $\mathbb{M}$ to the labels assigned by each annotator to calculate their reliability weights. In the case of using *per-label* weights (w_c^a), both estimation techniques can be formulated as,

$$\begin{aligned} w_c^a &= \frac{1}{\Phi} \sum_{i=1}^{N} \mathbb{I}[l_a^i = c] \cdot \mathbb{I}[\hat{c}^i = c] \\ w_c^a &= \frac{1}{\Phi} \sum_{i=1}^{N} \mathbb{I}[l_a^i = c] \cdot p_{\mathbb{M}}(c|\boldsymbol{x}^i) \end{aligned} \tag{7.3}$$

with normalization factor $\Phi = \sum_{i=1}^{N} \mathbb{I}[l_a^i = c]$. And, in the case of using the *confusion-matrix* reliability weights ($W_{cc'}^a$), both estimation procedures are,

$$\begin{aligned} W_{cc'}^a &= \frac{1}{\Phi} \sum_{i=1}^{N} \mathbb{I}[l_a^i = c] \cdot \mathbb{I}[\hat{c}^i = c'] \\ W_{cc'}^a &= \frac{1}{\Phi} \sum_{i=1}^{N} \mathbb{I}[l_a^i = c] \cdot p_{\mathbb{M}}(c'|\boldsymbol{x}^i) \end{aligned} \tag{7.4}$$

where Φ is in both cases a normalization constant such that $\sum_{c'=1}^{|\mathcal{C}|} W_{cc'}^a = 1$.

A procedure that updates the annotator reliability weights relying exclusively in the discrimination ability of the learnt model could be detrimental. If our EM procedure iteratively converges to a harmful classifier that only predicts a subset of labels, the estimated reliability weights can considerably differ from the real reliability values. In order to avoid this undesirable deviation, our method allows us to use the consensus weights (Eq. 7.1 or Eq. 7.2,

as appropriate) throughout the iterations of the EM process as a *correction* term. Thus, in this case the annotator reliability weights are estimated using the average value among the consensus weights and the model-estimated weights: i.e., (Eq. 7.1 + Eq. 7.3) / 2 or (Eq. 7.2 + Eq. 7.4) / 2.

7.3.2 A decomposition strategy for dealing with the multi-class problem: weighted voting one-vs-one

Our classification framework is multi-class ($|\mathcal{C}| = 9$), a kind of problem that is intrinsically more complex than one of binary classification [62, 113]: the expected classification error increases with the number of possible class labels ($|\mathcal{C}|$). Multi-class decomposition groups a set of strategies which build multiple binary classifiers that partially solve the multi-class problem. These strategies produce a multi-class prediction combining the prediction of the binary classifiers. Although the decomposition strategies are specially convenient for certain types of classifiers that do not have a straightforward extension to the multi-class paradigm, it has been claimed [61, 62] that the use of decomposition techniques enhances the performance of the base classifiers independently of whether the base classifier can deal with multiple class labels or not.

Many different strategies have been proposed in the ML community for binary decomposition. Among all the proposed strategies, one-vs-one (a classifier is learnt for each pair of class labels) and one-vs-all (a classifier is learnt for each class label considering as negative examples all the examples belonging to the rest of the labels) are the most frequently used. With the objective of emphasizing the importance of the multi-class nature of this framework, we have analyzed the inferred defect classification problem using decomposition techniques. Avoiding the discussion about the best decomposition strategy that could obscure the analysis of the results, one-vs-one (OvO) decomposition with weighted voting has been implemented. This simple strategy provides a methodology recognized in the related literature as yielding good classification performance [91, 61]. Weighted voting OvO is based on the use of base classifiers that are able to provide *soft* predictions, that is, a numeric value indicating the strength of the confidence of the classifier on the predicted class label instead of the class label alone. Thus, this strategy learns $|\mathcal{C}| \cdot (|\mathcal{C}| - 1)/2$ classifiers, each one learnt from the examples belonging to a different pair of class labels. Given a new example, the label with the largest confidence combining the soft predictions of the base binary classifiers is predicted. In this study, naive Bayes (Fig. 3.3) has been chosen as the probabilistic base classifier. Given its well-known performance [76] and simple learning process (fixed structure and few parameters), it has been previously used for this purpose [4]. Moreover, as a BNC, it computes conditional probabilities $p_{\mathbb{M}}(c|\boldsymbol{x})$ that can be used as soft predictions when applying weighted voting OvO. The combination of the predictions of the probabilistic binary NB classifiers allows us to provide a multi-class probability distribution,

$$p_{\mathbb{M}}(c|\boldsymbol{x}) = \frac{1}{|\mathcal{C}|-1} \sum_{c' \neq c \in \mathcal{C}} p_{\mathbb{M}}^{cc'}(c|\boldsymbol{x})$$

where $p_{\mathbb{M}}^{cc'}(\cdot|\boldsymbol{x})$, equal to $p_{\mathbb{M}}^{c'c}(\cdot|\boldsymbol{x})$, is a probability distribution over the class labels $\{c, c'\}$ given the example $\boldsymbol{x}$ according to the base classifier specifically learnt for these two labels.

This strategy has been directly adapted to the learning from crowds paradigm. Training examples of each base classifier are those that have been assigned by any annotator to either of the class labels considered in the respective classifier. As $\sum_{c=1}^{|\mathcal{C}|} F_c^{\boldsymbol{l}} = 1$ and only 2 out of 9 class labels are considered to learn a base classifier, a selected example involves a specific weight (≤ 1) determined by the reliability of the annotators who assigned the example to the class labels considered in that base classifier.

7.3.3 Dealing with multiple unbalanced class labels: SMOTEBoost

As previously discussed, Table 7.2 makes it clear that our problem is a multi-class *imbalance* framework. A class imbalance problem [78] appears in supervised classification when a dataset exhibits an unequal distribution between the different class labels. This problem gives rise to many issues in different stages of the learning process: how to train with few representatives of the minority classes, fair evaluation of classifiers, etc. All these issues compromise the performance of the learning techniques [78]. The different proposals presented in the related literature can be roughly divided into two groups: sampling and cost-sensitive methods. Among them, SMOTE is a popular sampling technique based on the generation of synthetic examples that has achieved good performance in comparison with competing methods [24, 78].

In order to deal with our *multi-class imbalance* classification problem, we have chosen the SMOTEBoost technique [25]. This is a well-established method in the related literature [189] which is based on the boosting algorithm AdaBoost.M2 [56] and uses the SMOTE procedure [24] for specifically dealing with the unbalanced nature of the problem. AdaBoost's sampling technique consecutively learns f (base) classifiers inducing, at each step, the learning process to concentrate on the most difficult areas of the instance space to enhance the discriminating ability of the final classifier. SMOTE modifies the sampling distribution of AdaBoost to compensate the class imbalance problem of the training dataset.

Similar to the previous approach, the probabilistic naive Bayes has been chosen as base classifier, which can be learnt using weighted examples. This property allows us to replace the resampling procedure of AdaBoost by a learning process which considers examples weighted according to the sampling distribution [24]. As classifiers are learnt with examples of all the class labels, the conditional probability distribution over $\mathcal{C}$ given an example $\boldsymbol{x}$ is calculated as follows,

$$p_{\mathbb{M}}(c|\boldsymbol{x}) = \frac{1}{\Phi} \sum_{t=1}^{f} \log\left(\frac{1}{\beta^{(t)}}\right) \cdot p_{\mathbb{M}}^{(t)}(c|\boldsymbol{x})$$

where Φ is a normalization factor, $p_{\mathbb{M}}^{(t)}(\cdot|\boldsymbol{x})$ is the conditional probability distribution according to the t-th base classifier and the $\beta^{(t)}$ value balances its contribution.

In the adaptation to the *learning from crowds* paradigm, the AdaBoost pseudo-loss has been calculated using as golden-truth the most voted labels (weighted according to the annotator reliability weights). In the same way, we initialize and update our sampling probability distribution. In order to identify the k nearest neighbor examples with a certain class label in SMOTE, only the examples whose most voted labels are the considered class label are taken into account. Also in SMOTE, once a subset of original examples is chosen for generating a synthetic example, the value of each variable is chosen as the most common value among those assigned to the corresponding variable in the examples of the subset. The same procedure is used to obtain each annotator's label.

7.4 Experimental work

7.4.1 Experimental settings

The three approaches for dealing with the presented ODC problem have been tested in a set of experiments. Apart from the labels provided by the five annotators, the original database involves two text fields (*summary* and *description*) and a categorical variable that defines the type of defect. In a pre-processing stage, natural language processing techniques have been used to extract a relevant set of variables from both text fields in order to transform the available database into a dataset which can be handled by standard machine learning techniques. Stop-words and punctuation marks have been removed and upper-case characters changed to lower-case. Only words appearing at least twice are considered. For each word a numeric variable is created which, for each defect, takes as value the *Term Frequency-Inverse Document Frequency* (TF-IDF) ratio. The most relevant 91 predictive variables (words) according to the most-voted labels were selected. Finally, each numeric variable has been transformed into a binary variable using a step function which takes positive value if the original numeric value is larger than zero and, otherwise, negative.

In order to delimit the experiments, many of the parameters of the different techniques used by the three approaches have been fixed to default/recommended values. Our EM method uses two parameters: a threshold indicating parametric convergence (set to 0.1%) and the maximum number of iterations (fixed to 200). In the specific case of SMOTEBoost, $f = 100$

classifiers (iterations) have been generated. It generates 10 replicas for each example of the minority classes ($c = \{2, 3, 5, 6\}$). Five nearest neighbors are considered for each minority example using the Hamming distance for categorical variables.

Classical evaluation techniques can be uninformative in unbalanced scenarios. Note that in our Compendium problem the examples belonging to two class labels (*Requirements* and *Usability*) represent more than 50% of the instances of the dataset (according to the most voted class labels, Table 7.2). A classifier exclusively focused in predicting these two labels could reach, for instance, a 0.5 accuracy value, a noteworthy result in a 9-label multi-class classification problem. According to a per-class evaluation procedure, this illustrative classifier would perform well in just two labels becoming useless for the other seven labels. In order to present the results for the posterior discussion, metrics commonly used for assessing the performance in multi-class imbalance problems have been considered: A-mean [124] (the mean of the recall values), maximum and minimum recall values, and the mean of the F1 measures [78]. Accuracy values are also presented in the tables of results as a measure that helps us to understand how the global performance is sacrificed in order to enhance the performance in the minority classes.

Additionally, the learning form crowds paradigm makes the evaluation of the models even more difficult. The lack of a ground-truth (certain labeling) makes the use of standard evaluation techniques impossible. Given the relatively recent emergence of the learning from crowds paradigm, the evaluation of classifiers learnt in these scenarios is still a field to be explored. Urkullu et al. [180] recently presented the only work that, to the extent of our knowledge, has addressed this issue. In their work, they propose different evaluation strategies based on alternative crowd scenarios for model selection. Their experimental results allow them to assert that the *mean* value of the performance metric calculated using the labels of one annotator at a time as ground-truth is, in comparison with the other considered proposals, a strategy especially suitable for crowd scenarios where few annotators label a large proportion of instances (in our case, 5 annotators labeled all the examples). We obtained all the experimental results with a 10×5-fold cross validation procedure [147].

7.4.2 Results

The results of the presented experiments are shown in Table 7.3 in terms of A-mean. The results for five classifiers are displayed in the rows: three types of Bayesian network classifiers (NB, TAN and 2DB) learnt using standard multi-class learning (first approach, Section 7.3.1), weighted voting OvO (second approach, Section 7.3.2), and SMOTEBoost (third approach, Section 7.3.3).

Three features of our crowd learning techniques are adjustable: the type of annotator reliability weights (*per-label* and *confusion-matrix*), the weight estimation procedure (*Prob* and *Acc*) and the use, or not, of *consensus* weight

correction. The eight different configurations (all the possible combinations of the three referred features) are displayed in the columns of Table 7.3. In order to assess the relevance of the improvement achieved with the use of these non-trivial crowd learning techniques, the Majority Voting (MV) strategy is used as a baseline. This simple strategy completes the dataset labeling each example with the label most voted among the crowd of annotators and, in this way, learns as in a standard supervised classification framework. The best configuration for each approach (row) is highlighted.

Note that the performance of a classifier in all the class labels contributes equally in the A-mean metric, which promotes the influence of the minority classes. Complementarily, Table 7.4 shows the results of the experiments in terms of minimum/maximum recall values. Both values aim to provide lower and upper boundaries for the performance in the different class labels that contribute in the computation of the A-mean metric. Table 7.5 shows the results of the experiments in terms of F1-mean and Table 7.6 in terms of accuracy. Although the accuracy metric is not suitable for class imbalance problems, it can be used to appreciate variations in the global performance.

7.5 Discussion

The crowd learning techniques developed for solving our Compendium defect classification task are based in our proposal for multi-dimensional frameworks (Section 5.4), which has been adapted to the unidimensional multi-class framework. Both the *Prob* and *Acc* procedures have been adapted to this framework and tested in the experimental work. Although there are slight differences, the *Prob* procedure outperforms *Acc* in the majority of experiments. The differences are more evident in experiments that use *per-label* reliability weights. In this case, the configuration that involves *per-label* weights estimated with the *Acc* procedure without consensus correction works particularly badly. It shows low A-mean and accuracy values and only outperforms other configurations in terms of maximum recall. That is, the classifiers learnt with this configuration concentrate their predictions on one or very few (probably *majority*) class label(s) and ignore the rest. This is usually an undesirable property in class

		Per-label				*Confusion-matrix*			
Classifier	MV	*Prob*	*Prob+Cons*	*Acc*	*Acc+Cons*	*Prob*	*Prob+Cons*	*Acc*	*Acc+Cons*
1: NB	0.335	0.324	0.342	0.328	0.339	**0.362**	0.353	0.351	0.356
1: TAN	0.316	0.302	0.306	0.296	0.304	0.318	0.316	**0.320**	0.314
1: 2DB	0.295	0.324	**0.348**	0.320	0.341	0.285	0.291	0.288	0.286
2: OvO	0.337	0.350	0.354	0.297	0.344	0.361	0.355	**0.362**	0.356
3: SB	**0.420**	0.391	0.391	0.390	0.387	0.400	0.393	0.386	0.397

Table 7.3. Results in terms of A-mean of the different approaches (rows) for the different annotator reliability weight estimation and combination procedures. Majority Voting (MV) is used as a baseline strategy.

		Per-label				*Confusion-matrix*			
Classifier	MV	*Prob*	*Prob+Cons*	*Acc*	*Acc+Cons*	*Prob*	*Prob+Cons*	*Acc*	*Acc+Cons*
1: NB	0.06/0.81	0.01/0.87	0.04/0.87	0.00/0.87	0.04/0.87	0.10/0.85	0.09/0.86	0.09/0.85	0.09/0.86
1: TAN	0.03/0.80	0.00/0.85	0.02/0.85	0.01/0.87	0.02/0.85	0.08/0.83	0.03/0.83	0.07/0.81	0.04/0.82
1: 2DB	0.01/0.79	0.01/0.86	0.04/0.86	0.00/0.88	0.03/0.87	0.02/0.78	0.03/0.80	0.01/0.78	0.01/0.80
2: OvO	0.06/0.82	0.09/0.85	0.08/0.86	0.00/0.89	0.03/0.87	0.08/0.86	0.08/0.86	0.11/0.85	0.09/0.86
3: SB	0.14/0.73	0.04/0.79	0.09/0.72	0.02/0.77	0.06/0.77	0.10/0.78	0.11/0.80	0.10/0.75	0.11/0.79

Table 7.4. Results in terms of *minimum* and *maximum* recall of the different approaches (rows) for the different annotator reliability weight estimation and combination procedures. Majority Voting (MV) is used as a baseline strategy.

		Per-label				*Confusion-matrix*			
Classifier	MV	*Prob*	*Prob+Cons*	*Acc*	*Acc+Cons*	*Prob*	*Prob+Cons*	*Acc*	*Acc+Cons*
1: NB	0.333	0.314	0.332	0.315	0.327	**0.362**	0.352	0.350	0.356
1: TAN	0.311	0.284	0.294	0.275	0.290	0.316	0.313	**0.322**	0.312
1: 2DB	0.283	0.310	**0.337**	0.301	0.329	0.276	0.286	0.283	0.280
2: OvO	0.335	0.350	0.346	0.252	0.331	0.359	0.353	**0.361**	0.356
3: SB	**0.408**	0.331	0.323	0.328	0.322	0.349	0.340	0.321	0.349

Table 7.5. Results in terms of mean F1 of the different approaches (rows) for the different annotator reliability weight estimation and combination procedures. Majority Voting (MV) is used as a baseline strategy.

		Per-label				*Confusion-matrix*			
Classifier	MV	*Prob*	*Prob+Cons*	*Acc*	*Acc+Cons*	*Prob*	*Prob+Cons*	*Acc*	*Acc+Cons*
1: NB	0.500	0.496	**0.502**	0.492	0.499	0.495	0.493	0.493	0.494
1: TAN	0.498	0.496	**0.500**	0.493	0.497	0.484	0.482	0.483	0.483
1: 2DB	0.490	0.496	**0.505**	0.492	0.500	0.479	0.482	0.477	0.480
2: OvO	0.500	**0.505**	**0.505**	0.412	0.502	0.498	0.496	0.494	0.493
3: SB	**0.484**	0.439	0.426	0.429	0.429	0.441	0.426	0.408	0.433

Table 7.6. Results in terms of accuracy of the different approaches (rows) for the different annotator reliability weight estimation and combination procedures. Majority Voting (MV) is used as a baseline strategy.

imbalance problems, where an adequate performance in all the class labels is desired.

Another interesting trend observed in these experiments is the different performance of the classifiers learnt with consensus weight correction. When *per-label* reliability weights are used, this correction works well and allows the classifiers to achieve enhanced performance regarding the same classifier learnt with the same configuration and no correction. The enhanced performance is systematically observed in terms of all the metrics (A-mean, F1 and accuracy). However, when *confusion-matrix* weights are used, the consensus correction is revealed to be completely unnecessary: The performance is rarely improved and, in some experiments, it even harms the results.

In general, the crowd learning techniques proposed in this work are appropriate for solving the problem of automating defect classification. The simplest solution, a standard learning technique that uses the most-voted labels (MV) marks a baseline, whose robust behavior has already been explained in Section

5.3. In these experiments, the MV strategy achieves reasonable performance and, in the case of SMOTEBoost, crowd learning techniques cannot outperform it. In the rest of cases, our techniques consistently outperform the simple MV strategy.

With respect to the different approaches, we have successfully integrated the different methodologies with the learning from crowds paradigm. In the first case, standard crowd learning techniques induce BNCs which deal by themselves with this multi-class problem. The best results are usually associated to the experiments that learn naive Bayes classifiers. Learnt TAN and 2DB classifiers usually do not reach the overall performance of naive Bayes. On the one hand, TAN classifiers are specially susceptible to the use of *per-label* weights. On the other hand, the use of *confusion-matrix* weights seems to be harmful to the performance of 2DB classifiers. In general, the performance of naive Bayes classifiers stands out when *confusion-matrix* weights are used. In this case, the differences with the other types of Bayesian network classifiers are even more evident.

The second approach, the weighted voting OvO decomposition technique learning binary naive Bayes classifiers, is generally the best strategy in terms of accuracy. However, as mentioned earlier, this metric is not reliable in multi-class imbalance situations. Moreover, the differences with respect to the naive Bayes classifiers learnt in the first approach are inconsistent and, many times, negligible (specifically in the experiments that use matrix reliability weights). Thus, its enhanced accuracy performance could not compensate for the greater complexity of the decomposition procedure.

The last approach aims to deal with the unbalanced nature of our multi-class problem. The SMOTEBoost technique achieves the best performance in terms of the A-mean, a metric which is effective for assessing the performance in unbalanced scenarios. That is, this approach is the best option if the ability to detect/classify examples of the minority classes is valued. As hypothesized, SMOTEBoost partially sacrifices the performance in highly-populated class labels but its performance in minority class labels is enhanced. Consequently, the performance improvement in terms of the A-mean metric has a negative impact on the accuracy values. Similarly, the minimum and maximum recall values show also the corresponding expected trend.

7.6 Threats to Validity

Concerning external validity, an obvious threat arises because we only used one project, Compendium. Software systems usually have specific features such as application domain, development environment, number of people reporting, etc. Therefore, different systems are likely to differ in the distribution of types of defects. From this point of view, the proposed machine learning techniques need to be adjusted to the specific environment of each problem.

Concerning construct validity, the quality in the ITS makes it hard to easily classify defect data manually. We do not address other problems faced in the defect repositories such as defect duplicates. Some preprocessing decisions (such as removal of outliers or the text-to-feature preprocess configuration) could have been different. Standard procedures and default values have been used to limit the discussion to the usefulness of the class information provided by the multiple annotators. Moreover, the original database is publicly available[2] to make the experimentation replicable.

Internal validity is concerned with whether the automated classifications has arisen as a result of chance or not. In case of 9 balanced class labels, the probability of randomly assigning the right label to an example is $1/9 = 0.111$. Assuming a random assignation of labels according to the distribution of labels observed in our class imbalance problem, the probability of being right is approximately 0.235. Therefore, according to the A-mean values obtained by the implemented machine learning techniques, it can be concluded that the automated classifications are not a product of chance.

7.7 Conclusions and future work

In this study, the learning from crowds paradigm has been considered for solving the defect classification problem. Three different approaches have been proposed to deal with a real application of the ODC problem, characterized as a multi-class imbalance framework.

The first approach implements a standard crowd learning framework, specific for multi-class problems, which learns Bayesian network classifiers. Secondly, a binary decomposition approach is carried out as an alternative strategy to solve the multi-class problem. Finally, a third approach makes use of SMOTEBoost to specifically deal with the unbalanced nature of this multi-class problem. In all the cases two different kinds of reliability weights have been considered for studying the accuracy of the different annotators. Crowd learning techniques generally outperform the standard classification methodologies which use a training dataset completed with the most-voted labels. Although further research is required, the obtained results encourage the use of the *learning from crowds* paradigm to deal with the defect classification problem.

In spite of the difficulties in evaluating a multi-class imbalance problem labeled by a crowd of annotators, the results have shown that the SMOTEBoost based approach obtains the best results, confirming that the unbalanced nature of our multi-class problem is a challenging critical issue. Moreover, it can be seen that the best performance in case of use of *per-label* weights is observed when they are corrected by means of the consensus weights. However, the consensus weight correction does not enhance the results of those

[2] `http://www.sc.ehu.es/ccwbayes/members/jeronimo/odc/`

experiments using the *confusion-matrix* weights. This may be due to a larger stability of the *confusion-matrix* weights to represent the reliability of the annotators. In terms of accuracy, a global metric which does not reflect the performance in minority classes, both standard crowd learning techniques and weighted voting OvO defeats SMOTEBoost. As expected, SMOTEBoost sacrifices the global accuracy performance to enhance the local performance in every class label.

As future work, a more appropriate adaptation of SMOTEBoost to the learning from crowds paradigm could be proposed. In general, a formal description and study of (multi-class) imbalance classification problems labeled by a crowd of annotators may be of interest. Specifically, we would like to study the effect of a set of unbalanced annotations provided by skewed annotators on the learning process of a classification framework which is already a class imbalance problem [198].

Part IV

Conclusions and Future Work

8

Conclusions and Future Work

All the contributions of this dissertation are covered by the expanding field of weakly supervised classification. Although almost from the very beginning of the supervised classification discipline different problems which provide partially labeled examples (e.g., the semi-supervised learning [23]) have been a matter of study, in the last years there has been an explosion in the number of works and proposals dealing with different types of weakly supervised classification frameworks. The field is wide and the solved problems vary. Our first contribution consists of a taxonomy of weakly supervised classification problems which aims to order the field and establish the basis for the discussion about the similarities and differences among different weakly supervised frameworks.

Then, our methodological contributions (Part II) explore two different problems in the field: the learning from label proportions and the learning from crowds problems. In the respective chapters, we presented our proposals to learn Bayesian network classifiers from the specific weakly labeled data of each problem. But the quick development of the field cannot be explained without its close relationship with real-world applications. In our case, the developed methodologies have been tested in two real applications (Part III): the assisted reproduction technologies and the software defect classification problems. Both applications have been studied and specific techniques have been proposed to deal with them based on the methodological contributions of Part II. Promising results have been obtained in both applications.

This final chapter is organized as follows. First of all, Section 8.1 individually draws a more detailed set of conclusions for each of the contributions of this dissertation. Next, the list of publications obtained during the development of this thesis is presented in Section 8.2. Finally, Section 8.3 identifies possible research lines for future work that remain open after this dissertation.

8.1 Conclusions

This dissertation has been devoted to the analysis of the capability of learning to classify when the available data is weakly supervised. Along the development of this thesis, we realized that the lack of a clear description of the field of weakly supervised classification has led to the misconception of several problems. Consequently, in Section 2.2 we provided a global description of the field. Many non-standard classification problems have been considered and compared with each other. This has allowed us to identify three fundamental characteristics for depicting the problems of the field: all the weakly supervised classification frameworks are characterized by the models of supervision that they implement in the learning and prediction stages, and by their instance-label relationship. Each of these characteristics are considered as an axis of the proposed taxonomy of weakly supervised classification problems. By means of this novel organization, the similarities and differences between different weakly supervised frameworks can be assessed. A general division of weak supervision models is formulated: the supervision models that provide class information for each example individually and those that provide class information jointly for groups of examples. Additionally, unexplored areas that could lead to new challenging frameworks are identified in the gaps of the tables inferred from the proposed taxonomy.

The rest of research works included in this dissertation, corresponding to Chapters 4 to 7, are divided into two groups: methodological developments and solutions to real applications of different weakly supervised frameworks.

8.1.1 Methodological contributions

In Chapter 4, we presented our first methodological study of a weakly supervised classification problem, the learning from label proportions. The supervision model implemented in the learning stage of this problem provides class information for groups of examples: the proportions of examples in each group that belong to each class label are provided. In line with previous studies [128, 136], we have shown that this supervision model provides relevant class information that can be used to learn more accurate classifiers. We have proposed four competitive versions of a Structural EM method to learn Bayesian network classifiers from label proportions.

Our proposal shows a competitive behavior with respect to state-of-the-art techniques, as has been shown in a comparison with the most representative and influential LLP methods [128, 136]. Among the four versions of our method, two versions which go through all the possible label assignments perform exact calculations by means of probabilistic (the PEM version) and non-probabilistic (NPEM) procedures. This probabilistic exact version has shown the best results. However, both exact versions are not scalable when the uncertainty of the problem and the number of possible joint label assignments grow. Another probabilistic version, MCEM, overcomes this issue

performing an approximation to the exact version by means of a MCMC procedure. It shows a good behavior in scenarios that are unaffordable for PEM, and the statistical tests associated to the experimental comparison do not show significant differences between both versions. Both probabilistic versions are combined in the fourth version, PMEM, such that approximate reasoning (MCEM) is only used when the exact approach (PEM) is unfeasible. It uses the MCMC-specific parameters (burn-in, bi, plus number of samples for calculating estimations, s) to establish a threshold for the maximum number of consistent completions that can be explored throughout the exact procedure. This last version performs as well as PEM (no statistical significant differences are found) and shows the best performance in terms of time-consumption.

In the second methodological study, we have gone into the learning from crowds problem (Chapter 5). In this weakly supervised classification problem, the supervision model implemented in the learning stage provides, for each example, a set of labels annotated by different unreliable annotators. A complete study of basic CrL strategies has been carried out, characterizing the crowd scenarios where each strategy shows a better performance. The straightforward majority voting strategy, which consists of completing the dataset with the most-voted labels to learn in a standard supervised classification framework, shows its strength in informed scenarios (a considerable set of competent annotators extensively label the examples of the dataset). From this study, a set of useful guidelines to select the most convenient strategy to cope with a specific crowd scenario can be inferred. We delimit the scenarios where the use of non-trivial methodologies are justified to enhance the obtained classifiers: scenarios with data scarcity where few unreliable annotators annotate the training examples.

We propose a general framework for learning multi-dimensional Bayesian network classifiers from data annotated by a crowd. Focusing on improving the learning process in crowd scenarios of data scarcity, different ways to incorporate the information about the reliability of the annotators have been explored. By means of a set of experiments performed with multi-label (synthetic and real) datasets transformed to the MDCrL framework, our proposal has been shown to overcome the simple approaches in crowd scenarios with data scarcity.

8.1.2 Applications

Initially conceived as a real application of the learning from label proportions problem, a case study of the problem of assisted reproductive technologies has been analyzed in an integral way through the use of machine learning techniques in Chapter 6. In collaboration with the Unit of Assisted Reproduction of the Donostia Hospital, the reinterpretation of the objectives of the ART problem has led to the design of four different approaches which provide a partial solution to the problem. Three out of the four approaches are described as (novel) weakly supervised classification problems and specific techniques have

been used for learning from the available data in each of the frameworks. Apart from a standard supervised classification problem, the other three subproblems are represented by the positive-unlabeled, label proportions and a novel weakly supervised classification problem. This novel framework, the learning from positive unlabeled proportions, provides the proportion of positive (the rest are assumed to be unlabeled) examples for each group in the training dataset. A SEM-based method has been developed to learn from data labeled by means of this novel weak supervision model.

Machine learning techniques had already been applied to the ARTs, although using a limited number of predictors and discarding embryos of uncertain fate. Our solution uses all the data collected by physicians during the ART treatment, considering also examples of uncertain fate for model learning, and evaluates its relevance for the four conceived classification tasks. This is one of the main differences with respect to the previous literature.

The learnt classifiers that predict the viability of a cycle show a promising performance. According to the results of the experiments, the collected data for characterizing embryos in our case study does not fully describe an embryo implantation. More research is required in order to find new relevant factors that determine the implantation of an embryo. In this direction, the experimental results of Section 6.3 show the relevance of the features describing the cycle for determining embryo implantation (and ART success). In fact, it can be appreciated that the data currently collected for characterizing embryos describes more consistently the embryo development. Classifiers obtained with this approach have been proved to rank the medium-quality embryos of our case study more reliably than ASEBIR [3] grade. These are promising results due to the difficulty in establishing a reliable quality embryo grade, especially for medium-quality embryos, as previously reported in the related literature. They support the use of the obtained classifiers and their probabilistic embryo assessments as an alternative embryo score.

In Chapter 7, a case study of the software defect classification problem has been analyzed by means of the learning from crowds paradigm. The real application, a set of defects reported in the issue tracking system of the Compendium software project, is described as a multi-class imbalance problem labeled by a set of five annotators. Apart from the methodology proposed in Chapter 5, adapted to multi-class problems and successfully used to learn BNCs (first approach), we have proposed two other approaches to deal with our case study. On the one hand, weighted voting one-vs-one, a binary decomposition strategy, has been implemented in the learning from crowds framework. It aims to specifically deal with the multi-class nature of the defect classification problem. On the other hand, our third approach makes use of SMOTEBoost to specifically deal with the unbalanced nature of this multi-class problem. In both cases, we have successfully integrated these classical techniques of standard supervised classification into the framework of learning from crowds. As expected, SMOTEBoost overcomes the other two approaches in terms of A-mean (a performance metric specifically suitable for

class imbalance problems). In terms of accuracy (a global metric which does not reflect the performance in minority classes), both standard crowd learning methodology and weighted voting OvO defeat SMOTEBoost. It can be observed that SMOTEBoost concurs with a drop in global accuracy performance of the classifier in order to enhance the local performance in every class label. In general, proposed crowd learning techniques generally outperform the standard supervised classification methodologies which learn from a training dataset completed with the most-voted labels. Therefore, the overall results encourage the use of the *learning from crowds* paradigm to deal with defect classification.

Regarding the configuration of our methodology to assess the reliability of the annotators, there is no predominant configuration. It can be seen that the best performance while using *per-label* weights is observed when they are corrected by means of the consensus weights. However, the consensus weight correction does not enhance the results of those experiments using the *confusion-matrix* weights. This may be due to a larger stability of the *confusion-matrix* weights to represent the reliability of the annotators.

8.2 Publications of the thesis

The research work conducted during this thesis has given rise to the following publications and submissions:

8.2.1 List of publications in referred journals

- **J. Hernández-González**, I. Inza and J. A. Lozano (2015) Multidimensional learning from crowds: usefulness and application of expertise detection. *International Journal of Intelligent Systems* 30(3); pp. 326–354.
- **J. Hernández-González**, I. Inza and J. A. Lozano (2013) Learning Bayesian network classifiers from label proportions. *Pattern Recognition* 46(12); pp. 3425–3440.

8.2.2 List of submitted papers

- **J. Hernández-González**, D. Rodriguez, I. Inza, R. Harrison and J. A. Lozano (2015) Classifying software defects: a learning from crowds approach. *IEEE Transactions on Software Engineering*; Submitted.
- **J. Hernández-González**, I. Inza, L. Crisol-Ortíz, M. A. Guembe, M. J. Iñarra and J. A. Lozano (2015) Novel weakly supervised classification techniques for human assisted reproduction: a case study. *Statistics in Medicine*; Submitted.
- **J. Hernández-González**, I. Inza and J. A. Lozano (2015) Weak supervision and other non-standard classification problems: a taxonomy. *Pattern Recognition Letters*; Submitted.

8.2.3 List of conference communications

- **J. Hernández-González**, I. Inza and J. A. Lozano (2015) A novel weakly supervised problem: Learning from positive-unlabeled proportions. In: *Proceedings the 16th Conference of the Spanish Association for Artificial Intelligence (CAEPIA)*, Albacete, Spain; *To appear*.
- **J. Hernández-González**, I. Inza and J. A. Lozano (2013) Learning from crowds in multi-dimensional classification domains. In: *Proceedings the 15th Conference of the Spanish Association for Artificial Intelligence (CAEPIA)*, Madrid, Spain; pp. 352–362.
- **J. Hernández-González** and I. Inza (2011) Learning naive Bayes models for multiple-instance learning with label proportions. In: *Proceedings the 14th Conference of the Spanish Association for Artificial Intelligence (CAEPIA)*, La Laguna, Spain; pp. 134–144.

8.2.4 Collaborations

- M. J. Iñarra, L. Crisol-Ortíz, M. A. Guembe, **J. Hernández-González**, I. Inza, J. A. Lozano and A. Lekuona (2014) Elección embrionaria para mejorar la tasa de éxito en reproducción asistida por medio de técnicas de aprendizaje automático. In: *XXX Congreso de la Sociedad Española de Fertilidad y V Congreso de Enfermería de la Reproducción*, Barcelona, Spain.

8.3 Future work

The research works carried out during the development of this thesis have certainly left open multiple questions that would need further research. In this section, we identify some of these open questions: ideas that would allow us to progress in the comprehension of the explored weakly supervised problems and suggestions for developing/adjusting the proposed methodologies.

In this dissertation, we have focused on weakly supervised classification. However, there are many other non-standard classification problems that could be considered in an extended taxonomy of classification problems. First of all, there are problems which learn classifiers that do not provide full-categorization predictions but a different kind of information, such as a ranking or a probability distribution over all the possible class labels [73, 118]. Note that this characteristic fulfills the condition that we used to consider the inclusion of features as axes of our taxonomy: *when a solution is built*, the kind of information that a classifier is expected to return has to be known. Beyond the four instance-label relationships considered in the proposed taxonomy (Table 2.1), many other frameworks have been proposed in the literature: problems where there exists no absolute membership to any categorization for

the examples (label ranking [16, 184] or label distribution [66, 187]), or problems where the image of the target function cannot be represented by just a single class variable (e.g., multi-dimensional framework [6, 133]). It would be interesting to investigate if the taxonomy axis that represents the instance-label relationship should cover these non-standard problems, which are usually named as structured output and/or multi-target learning [179, 186].

One of the main challenges in this field is the proposal of a whole framework for learning and validating classifiers using exclusively weakly supervised data. Many techniques (from evaluation to feature subset selection strategies) are based on the availability of fully labeled examples. Therefore, the referred standard techniques cannot be straightforwardly used for dealing with weakly supervised problems. So far, the use of simulated datasets have allowed researchers (including us) to develop and test novel methodologies that learn from different weakly supervised problems. However, the use of real weakly supervised datasets challenges the traditional learning and evaluation frameworks as the ground truth is not available. We have faced these difficulties when dealing with the real applications (Part III). With the current expansion of the field, a complete framework for learning and evaluating classifiers in absence of the ground truth would be extremely relevant.

As exposed in Section 2.2, it seems difficult to propose a ranking or global description for the whole spectrum of the weak supervision models due to the wide diversity of models found in the related literature. However, two groups of supervision models have been recognized. It would be very interesting to explore the possibilities of carrying out a formal mathematical study of the ability of learning in each of the groups of supervision models, perhaps establishing boundaries in the expected error of the learnt classifiers depending on the degree of class uncertainty in the provided data and sample size.

In Chapter 4, we have dealt with the learning from label proportions problem. Regarding the proposed methodology, specifically the final PMEM version, it would be interesting to study the possibility of automatically calculating for each bag individually the parameters of the MCMC procedure (burn-in and number of samples). This would establish a non-constant threshold in the maximum number of consistent completions considered with the exact procedure. The implications of this decision should also be studied.

From a theoretical point of view, it would also be interesting to consider a weakly supervised problem similar to LLP with a relaxed notion of proportions. That is, considering a supervision model with groups of examples where each group provides a probability distribution over the class labels for the examples of the bag. Similarly, another interesting extension for this kind of group-based supervision models is the provision of class information for *non-disjoint* groups of examples. Although, a priori, this particularity should help to reduce the level of class uncertainty, an in-depth research is required to establish the consequences of this relaxation.

In the context of learning from crowds, it is not realistic to assume that all the annotators label all the examples. Relaxing this assumption would imply

taking into account annotators who label few instances, making it more difficult to assess the reliability of annotators. This would require the redesign of the techniques that have been proposed to calculate the reliability weights. Similarly, in this dissertation it has been assumed that all the annotators provide wrong labels randomly. On the contrary, considering non-random noisy annotators (e.g., someone that tends to label incorrectly only the examples of a specific area of the instance space) requires the development of specific methodologies to deal with this type of annotations.

Exploring the possible extensions of our methodology, let us imagine a crowd scenario where the number of annotators that label each instance is very different; or a different scenario where a brilliant domain expert labels only sporadically. It could be interesting to implement a complete method which is able to choose in run-time the most appropriate strategy for each example: majority voting if many labels have been provided, expert selection only for the examples labeled by highly reliable experts, etc.

With respect to our proposals for the problem of human assisted reproduction, it would be interesting to carry out a clinical study in order to verify the enhanced performance of our classifiers with respect to the currently implemented embryo selection criteria. In order to do so, a previous study has to be carried out for finding the method configuration that optimizes the performance of the obtained classifiers. From a medical point of view, more research is required to find novel features influencing embryo implantation. In this way, the arrangement of potentially successful cycles and embryos will hopefully be improved. Moreover, the study carried out in Chapter 6, inspired by the embryo-uterine approach, analyzed both datasets (cycle and embryos) separately. As there exists a one-to-n relationship among the cycles and embryos, it would be interesting to try to model this problem by means of data analysis techniques for relational databases [59].

The software defect classification problem has opened several interesting questions. We have adapted to the learning from crowds paradigm two standard techniques that deal with specific issues of supervised classification problems: one-vs-one multi-class decomposition and SMOTEBoost. The issues that are solved by means of these techniques are observed in both standard and weakly supervised classification problems. Therefore, as the learning from crowds paradigm spreads out, it is expected that this and other types of standard supervised classification techniques are adapted to crowd learning. In this context, the adaptation of SMOTEBoost, the only proposed approach overcome by the MV strategy according to the experimental results, should be redesigned. From a theoretical point of view, the (multi-class) imbalance classification problem labeled by a crowd should be formally described. This description could lead to the study of the interaction among the class imbalance problem and the provision of unbalanced annotations. That is, to describe the learning process of a class imbalance classification problem which has been labeled by a set of skewed annotators [198].

References

1. Hanna Achache and Ariel Revel. Endometrial receptivity markers, the journey to successful embryo implantation. *Human Reproduction Update*, 12(6):731–746, 2006.
2. Giuliano Antoniol, Kamel Ayari, Massimiliano Di Penta, Foutse Khomh, and Yann-Gaël Guéhéneuc. Is it a bug or an enhancement?: A text-based approach to classify change requests. In *Proceedings of the 2008 Conference of the Center for Advanced Studies on Collaborative Research: Meeting of Minds*, pages 23:304–23:318, New York, NY, USA, 2008. ACM.
3. Manuel Ardoy and Gloria Calderón. *Clinical Embryology Papers: ASEBIR criteria for the morphological evaluation of human oocytes, early embryos and blastocysts*. Asociación para el Estudio de la Biología de la Reproducción (ASEBIR), Madrid, Spain, 2nd edition, 2008.
4. Andoni Arruti, Iñigo Mendialdua, Basilio Sierra, Elena Lazkano, and Ekaitz Jauregi. New one versus$^{\text{all}}_{\text{one}}$ method: Nov@. *Expert Systems with Applications*, 41(14):6251–6260, 2014.
5. Concha Bielza and Pedro Larrañaga. Discrete Bayesian network classifiers: a survey. *ACM Computing Surveys (CSUR)*, 47(1):5, 2014.
6. Concha Bielza, Guangdi Li, and Pedro Larrañaga. Multi-dimensional classification with Bayesian networks. *International Journal of Approximate Reasoning*, 52(6):705–727, 2011.
7. Anna M. Bizoń-Adamczyk. Part 1. classify and validate defects. In *Improve project quality with Rational Team Concert 3.0 and ODC*. IBM Corporation, May 2011. `http://www.ibm.com/developerworks/rational/library/improve-quality-Rational-team-concert-odc/improve-quality-Rational-team-concert-odc-pdf.pdf`.
8. Rosa Blanco, Iñaki Inza, and Pedro Larrañaga. Learning Bayesian networks in the space of structures by estimation of distribution algorithms. *International Journal of Intelligent Systems*, 18(2):205–220, 2003.
9. Norm Bridge and Corinne Miller. Orthogonal defect classification using defect data to improve software development. *Software Quality*, 3:1997–8, 1998.
10. Daniel R. Brison, Katherine Hollywood, Ruth Arnesen, and Royston Goodacre. Predicting human embryo viability: the road to non-invasive analysis of the secretome using metabolic footprinting. *Reproductive BioMedicine Online*, 15(3):296–302, 2007.

11. D.R. Brison, F.D. Houghton, D. Falconer, S.A. Roberts, J. Hawkhead, P.G. Humpherson, B.A. Lieberman, and H.J. Leese. Identification of viable embryos in IVF by non-invasive measurement of amino acid turnover. *Human Reproduction*, 19(10):2319–2324, 2004.
12. Carla E. Brodley and Mark A. Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167, 1999.
13. S. Brooks. Markov chain monte carlo method and its application. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(1):69–100, 1998.
14. Serhat Selcuk Bucak, Rong Jin, and Anil K. Jain. Multi-label learning with incomplete class assignments. In *Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2801–2808, 2011.
15. Luigi Buglione and Alain Abran. Introducing root-cause analysis and orthogonal defect classification at lower CMMI maturity levels. In *Proceedings of the International Conferences on Software Process and Product Measurement (Mensura'06)*, 2006.
16. Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22th International Conference on Machine Learning (ICML)*, pages 89–96, 2005.
17. M. Butcher, H. Munro, and T. Kratshmer. Improving software testing via ODC: Three case studies. *IBM Systems Journal*, 41(1):31–44, 2002.
18. Borja Calvo, Iñaki Inza, Pedro Larrañaga, and Jose A. Lozano. Wrapper positive bayesian network classifiers. *Knowledge and Information Systems*, 33(3):631–654, 2012.
19. Borja Calvo, Pedro Larrañaga, and Jose A. Lozano. Learning Bayesian classifiers from positive and unlabeled examples. *Pattern Recognition Letters*, 28(16):2375–2384, 2007.
20. Borja Calvo, Pedro Larrañaga, and Jose A. Lozano. Feature subset selection from positive and unlabelled examples. *Pattern Recognition Letters*, 30(11):1027–1036, 2009.
21. Enrique Castillo, José Manuel Gutiérrez, and Ali S. Hadi. *Expert systems and probabilistic network models*. Springer Science & Business Media, 1997.
22. Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines, 2011.
23. Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-supervised Learning*. The MIT Press, 2006.
24. Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, pages 321–357, 2002.
25. Nitesh V. Chawla, Aleksandar Lazarevic, Lawrence O. Hall, and Kevin W. Bowyer. Smoteboost: Improving prediction of the minority class in boosting. In *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 107–119. Springer, 2003.
26. Gang Chen, Yangqiu Song, Fei Wang, and Changshui Zhang. Semi-supervised multi-label learning by solving a sylvester equation. In *Proceedings of the 8th SIAM International Conference on Data Mining (SDM)*, pages 410–419, 2008.
27. Shuo Chen, Bin Liu, Mingjie Qian, and Changshui Zhang. Kernel k-means based framework for aggregate outputs classification. In *IEEE International Conference on Data Mining Workshops (ICDM Workshops 2009)*, pages 356–361, 2009.

28. David M. Chickering. Learning bayesian networks is np-complete. *Learning from Data: Artificial Intelligence and Statistics V*, 1996.
29. R. Chillarege, I.S. Bhandari, J.K. Chaar, M.J. Halliday, D.S. Moebus, B.K. Ray, and M.-Y. Wong. Orthogonal defect classification-a concept for in-process measurements. *IEEE Transactions on Software Engineering*, 18(11):943–956, Nov 1992.
30. Sharath R. Cholleti, Sally A. Goldman, Avrim Blum, David G. Politte, and Steven Don. Veritas: Combining expert opinions without labeled data. In *Proceedings of the 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 45–52, 2008.
31. C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.
32. Gregory F. Cooper and Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347, 1992.
33. Giorgio Corani, Cristina Magli, Alessandro Giusti, Luca Gianaroli, and Luca M. Gambardella. A bayesian network model for predicting pregnancy after in vitro fertilization. *Computers in Biology and Medicine*, 43(11):1783–1792, 2013.
34. C. Coughlan, W. Ledger, Q. Wang, Fenghua Liu, Aygul Demirol, Timur Gurgan, R. Cutting, K. Ong, H. Sallam, and T.C. Li. Recurrent implantation failure: definition and management. *Reproductive BioMedicine Online*, 28(1):14–38, 2015.
35. Timothée Cour, Benjamin Sapp, and Ben Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 12:1501–1536, 2011.
36. J.M. Cummins, T.M. Breen, K.L. Harrison, J.M. Shaw, L.M. Wilson, and J.F. Hennessey. A formula for scoring human embryo growth rates in in vitro fertilization: its value in predicting pregnancy and in comparison with visual estimates of embryo quality. *Journal of In Vitro Fertilization and Embryo Transfer*, 3(5):284–295, 1986.
37. A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28, 1979.
38. Luis M. De Campos, Juan M. Fernandez-Luna, José A. Gámez, and José M. Puerta. Ant colony optimization for learning Bayesian networks. *International Journal of Approximate Reasoning*, 31(3):291–311, 2002.
39. Peter R. de Waal and Linda C. van der Gaag. Inference and learning in multi-dimensional Bayesian network classifiers. In *Proceedings of the European Conferences on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 4724, pages 501–511. Springer, 2007.
40. Ana Debón, Inmaculada Molina, Suitberto Cabrera, and Antonio Pellicer. Mathematical methodology to obtain and compare different embryo scores. *Mathematical and Computer Modelling*, 57(5-6):1380–1394, 2013.
41. Ofer Dekel and Ohad Shamir. Good learners for evil teachers. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 233–240, 2009.
42. Ofer Dekel and Ohad Shamir. Vox populi: Collecting high-quality labels from a crowd. In *Proceedings of the 22nd Conference on Learning Theory (COLT)*, 2009.

43. Juan José del Coz, Jorge Díez, and Antonio Bahamonde. Learning nondeterministic classifiers. *J. Mach. Learn. Res.*, 10:2273–2293, 2009.
44. Richard A. Demillo and Aditya P. Mathur. A grammar based fault classification scheme and its application to the classification of the errors of TEX. Technical report, Software Engineering Research Center and Department of Computer Sciences, Purdue University, W. Lafayette, IN 47907, November 1995.
45. Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.
46. Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, Dec 2006.
47. Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1–2):31–71, 1997.
48. J. Duraes and H. Madeira. Definition of software fault emulation operators: a field data study. In *Proceedings of the International Conference on Dependable Systems and Networks*, 2003.
49. Thomas Ebner, Marianne Moser, Michael Sommergruber, and Gernot Tews. Selection based on morphological assessment of oocytes and embryos at different stages of preimplantation development: a review. *Human Reproduction Update*, 9(3):251–262, 2003.
50. Gal Elidan. Bagged structure learning of bayesian network. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, pages 251–259, 2011.
51. Albert Endres. An analysis of errors and their causes in system programs. In *Proceedings of the International Conference on Reliable Software*, pages 327–336, New York, NY, USA, 1975. ACM.
52. Lawrence Engmann, Noreen Maconochie, Seang Lin Tan, and Jinan Bekir. Trends in the incidence of births and multiple births and the factors that determine the probability of multiple birth after IVF treatment. *Human Reproduction*, 16(12):2598–2605, 2001.
53. Jeffrey D. Fisch, Herman Rodriguez, Richard Ross, Gail Overby, and Geoffrey Sher. The graduated embryo score (GES) predicts blastocyst formation and pregnancy rate from cleavage-stage embryos. *Human Reproduction*, 16(9):1970–1975, 2001.
54. A. Frank and A. Asuncion. UCI machine learning repository, 2010.
55. B. Freimut, C. Denger, and M. Ketterer. An industrial case study of implementing and validating defect classification for process improvement and quality management. In *Proceeding of the 11th IEEE International Symposium on Software Metrics (METRICS'05)*, pages 10–19, Sept 2005.
56. Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, volume 96, pages 148–156, 1996.
57. Nir Friedman. Learning belief networks in the presence of missing values and hidden variables. In *Proceedings of the 14th International Conference on Machine Learning (ICML 1997)*, pages 125–133, 1997.
58. Nir Friedman, Dan Geiger, and Moisés Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2–3):131–163, 1997.

59. Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. Learning probabilistic relational models. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 99, pages 1300–1309, 1999.
60. Gabriel P. C. Fung, Jeffrey X. Yu, Hongjun Lu, and Philip S. Yu. Text classification without negative examples revisit. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):6–20, 2006.
61. Mikel Galar, Alberto Fernández, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition*, 44(8):1761–1776, 2011.
62. Mikel Galar, Alberto Fernández, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. Dynamic classifier selection for one-vs-one strategy: Avoiding non-competent classifiers. *Pattern Recognition*, 46(12):3412–3424, 2013.
63. Salvador García and Francisco Herrera. An extension on "Statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, Dec 2008.
64. Dario García-García and Robert C. Williamson. Degrees of supervision. In *Proceedings of the 25th Annual Conference on Neural Information Processing Systems Workshops (NIPS)*, pages 897–904, 2011.
65. David K. Gardner, Michelle Lane, John Stevens, and William B. Schoolcraft. Noninvasive assessment of human embryo nutrient consumption as a measure of developmental potential. *Fertility and Sterility*, 76(6):1175–1180, 2001.
66. Xin Geng, Chao Yin, and Zhi-Hua Zhou. Facial age estimation by learning from label distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10):2401–2412, 2013.
67. Walter R. Gilks, Sylvia Richardson, and David J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, 1996.
68. Norbert Gleicher and David Barad. The relative myth of elective single embryo transfer. *Human Reproduction*, 21(6):1337–1344, 2006.
69. Robert B. Grady. *Practical Software Metrics for Project Management and Process Improvement*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.
70. Yves Grandvalet and Yoshua Bengio. Learning from partial labels with minimum entropy. Technical report, Centre interuniversitaire de recherche en analyse des organisations (CIRANO), 2004.
71. F. Guerif, A. Le Gouge, B. Giraudeau, J. Poindron, R. Bidault, O. Gasnier, and D. Royere. Limited value of morphological assessment at days 1 and 2 to predict blastocyst development potential: a prospective study based on 4042 embryos. *Human reproduction*, 22(7):1973–1981, 2007.
72. Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
73. Ole Martin Halck. Using hard classifiers to estimate conditional class probabilities. In *Proceedings of the 13th European Conference on Machine Learning (ECML)*, pages 124–134, 2002.
74. Mark A. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, Department of Computer Science, The University of Waikato, 1999.
75. T. Hall, D. Bowes, S. Counsell, L. Moonen, and A. Yamashita. Software fault characteristics: A synthesis of the literature. `http://bura.brunel.ac.uk/handle/2438/11013`, 2015.

76. David J. Hand and Keming Yu. Idiot's bayes—not so stupid after all? *International Statistical Review*, 69(3):385–398, 2001.
77. Trevor. Hastie, Robert. Tibshirani, and J Jerome H Friedman. *The elements of statistical learning*, volume 1. Springer New York, 2001.
78. Haibo He, Edwardo Garcia, et al. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.
79. David Heckerman. A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, Learning in Graphical Models, 1995.
80. Jerónimo Hernández-González and Iñaki Inza. Learning naive Bayes models for multiple-instance learning with label proportions. In *Proceedings of the 14th Conference of the Spanish Association for Artificial Intelligence (CAEPIA)*, pages 134–144, 2011.
81. Jerónimo Hernández-González, Iñaki Inza, and Jose A. Lozano. Learning Bayesian network classifiers from label proportions. *Pattern Recognition*, 46(12):3425–3440, 2013.
82. Jerónimo Hernández-González, Iñaki Inza, and Jose A. Lozano. Multidimensional learning from crowds: Usefulness and application of expertise detection. *International Journal of Intelligent Systems*, 30(3):326–354, 2015.
83. Jerónimo Hernández-González, Iñaki Inza, and Jose A. Lozano. Weak supervision and other non-standard classification problems: a taxonomy. *Pattern Recognition Letters*, 2015. Submitted.
84. Edward H. Herskovits and Gregory F. Cooper. Kukató: An entropy-driven system for construction of probabilistic expert systems from databases. In *Proceedings of the 6th Conference on Uncertainty in Artificial Intelligence*, pages 54–62. Elsevier Science Inc., 1990.
85. Kim Herzig, Sascha Just, and Andreas Zeller. It's not a bug, it's a feature: How misclassification impacts bug prediction. In *Proceedings of the 2013 International Conference on Software Engineering*, pages 392–401, 2013.
86. Jan Holte, Lars Berglund, K. Milton, C. Garello, Gianluca Gennarelli, Alberto Revelli, and Torbjörn Bergh. Construction of an evidence-based integrated morphology cleavage embryo score for implantation potential of embryos scored and transferred on day 2 after oocyte retrieval. *Human Reproduction*, 22(2):548–557, 2007.
87. Jeff Howe. The rise of crowdsourcing. *Wired Magazine*, 15(6):1–4, 2006.
88. LiGuo Huang, Vincent Ng, Isaac Persing, Mingrui Chen, Zeheng Li, Ruili Geng, and Jeff Tian. Autoodc: Automated generation of orthogonal defect classifications. *Automated Software Engineering*, 22(1):3–46, 2015.
89. LiGuo Huang, Vincent Ng, Isaac Persing, Ruili Geng, Xu Bai, , and Jeff Tian. Autoodc: Automated generation of orthogonal defect classification. In *Proceedings of the 26th IEEE/ACM International Conference on Automated Software Engineering*, pages 412–415, 2011.
90. Eyke Hüllermeier and Jürgen Beringer. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):419–439, 2006.
91. Eyke Hüllermeier and Stijn Vanderlooy. Combining predictions in pairwise classification: An optimal adaptive voting strategy and its relation to weighted voting. *Pattern Recognition*, 43(1):128–142, 2010.
92. IEEE. IEEE standard classification for software anomalies, Std. 1044-1993, 1993.
93. ISO/IEC. ISO/IEC 9126. software engineering – product quality, 2001.

94. Rong Jin and Zoubin Ghahramani. Learning with multiple labels. In *Proceedings of Advances in Neural Information Processing Systems 15 (NIPS)*, pages 897–904, 2002.
95. Mandy G. Katz-Jaffe, David K. Gardner, and William B. Schoolcraft. Proteomic analysis of individual human embryos to identify novel biomarkers of development and viability. *Fertility and Sterility*, 85(1):101–107, 2006.
96. Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.
97. Kevin B. Korb and Ann E. Nicholson. *Bayesian Artificial Intelligence*. CRC Press, Inc., 2003.
98. Joseph B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.
99. Hendrik Kück and Nando de Freitas. Learning about individuals from group statistics. In *Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence (UAI)*, pages 332–339, 2005.
100. Sanjiv Kumar and Henry A. Rowley. Classification of weakly-labeled data with partial equivalence relations. In *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.
101. Ludmila I. Kuncheva. Full-class set classification using the Hungarian algorithm. *International Journal of Machine Learning and Cybernetics*, 1(1–4):53–61, 2010.
102. Pedro Larrañaga, Borja Calvo, Roberto Santana, Concha Bielza, Josu Galdiano, Iñaki Inza, Jose A. Lozano, Rubén Armañanzas, Guzmán Santafé, Aritz Pérez, and V. Robles. Machine learning in bioinformatics. *Briefings in Bioinformatics*, 7(1):86–112, 2006.
103. Pedro Larrañaga, Cindy M. H. Kuijpers, Roberto H. Murga, and Yosu Yurramendi. Learning Bayesian network structures by searching for the best ordering with genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 26(4):487–493, 1996.
104. Pedro Larrañaga and Jose A. Lozano. *Estimation of distribution algorithms: A new tool for evolutionary computation*, volume 2. Springer Science & Business Media, 2002.
105. Steffen L. Lauritzen. *Graphical models*. Oxford University Press, 1996.
106. Steffen L. Lauritzen, A. Philip Dawid, Birgitte N. Larsen, and H. G. Leimer. Independence properties of directed markov fields. *Networks*, 20(5):491–505, 1990.
107. Steffen L. Lauritzen and David J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, pages 157–224, 1988.
108. Wee Sun Lee and Bing Liu. Learning with positive and unlabeled examples using weighted logistic regression. In *Proceedings of the 20th International Conference on Machine Learning*, pages 448–455, 2003.
109. Florence Lesourd, Olivier Parant, Muriel Clouet-Delannoy, and Jean Parinaud. Clinical and biological parameters influencing implantation: score to determine number of embryos to transfer. *Reproductive BioMedicine Online*, 12(4):453–459, 2006.

110. Marek Leszak, Dewayne E. Perry, and Dieter Stoll. Classification and evaluation of defects in a project retrospective. *Journal of Systems and Software*, 61(3):173–187, 2002.
111. Bing Liu, Wee S. Lee, Philip S. Yu, and Xiaoli Li. Partially supervised classification of text documents. In *Proceedings of the 19th International Conference on Machine Learning (ICML)*, pages 387–394, 2002.
112. Pedro L. López-Cruz, Pedro Larrañaga, Javier DeFelipe, and Concha Bielza. Bayesian network modeling of the consensus between experts: An application to neuron classification. *International Journal of Approximate Reasoning*, 55(1):3–22, 2014.
113. Ana Carolina Lorena, André C.P.L.F. De Carvalho, and João M.P. Gama. A review on the combination of binary classifiers in multiclass problems. *Artificial Intelligence Review*, 30(1-4):19–37, 2008.
114. Gábor Lugosi. Learning with an unreliable teacher. *Pattern Recognition*, 25(1):79–87, 1992.
115. Jie Luo and Francesco Orabona. Learning from candidate labeling sets. In *Proceedings of Advances in Neural Information Processing Systems 23 (NIPS)*, pages 1504–1512, 2010.
116. Walid Maalej and Hadeer Nabil. Bug report, feature request, or simply praise? on automatically classifying app reviews. In *Proceedings of the 23rd IEEE International Requirements Engineering Conference*, page to appear, 2015.
117. E.J. Margalioth, A. Ben-Chetrit, M. Gal, and T. Eldar-Geva. Investigation and treatment of repeated implantation failure following IVF-ET. *Human Reproduction*, 21(12):3036–3043, 2006.
118. Dragos D. Margineantu. Class probability estimation and cost-sensitive classification decisions. In *Proceedings of the 13th European Conference on Machine Learning (ECML)*, pages 270–281, 2002.
119. Peter M. Martin and H. Gilbert Welch. Probabilities for singleton and multiple pregnancies after in vitro fertilization. *Fertility and sterility*, 70(3):478–481, 1998.
120. R.G. Mays, C.L. Jones, G.J. Holloway, and D.P. Studinski. Experiences with defect prevention. *IBM Systems Journal*, 29(1):4–32, 1990.
121. Geoffrey J. McLachlan and Thriyambakam Krishnan. *The EM Algorithm and Extensions (Wiley Series in Probability and Statistics)*. Wiley-Interscience, 1997.
122. Ben Medlock and Ted Briscoe. Weakly supervised learning for hedge classification in scientific literature. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 992–999, 2007.
123. Carmen Mendoza, Estrella Ruiz-Requena, Esperanza Ortega, Nieves Cremades, Francisco Martinez, Rafael Bernabeu, Ermanno Greco, and Jan Tesarik. Follicular fluid markers of oocyte developmental potential. *Human Reproduction*, 17(4):1017–1022, 2002.
124. Aditya Menon, Harikrishna Narasimhan, Shivani Agarwal, and Sanjay Chawla. On the statistical consistency of algorithms for binary classification under class imbalance. In *Proceedings of the 30th International Conference on Machine Learning*, pages 603–611, 2013.
125. Tom Mitchell. *Machine Learning*. McGraw Hill, 1997.
126. Dinora A. Morales, Endika Bengoetxea, and Pedro Larrañaga. Selection of human embryos for transfer by Bayesian classifiers. *Computers in Biology and Medicine*, 38(11–12):1177–1186, 2008.

127. Itzel Morales-Ramirez, Dimitra Papadimitriou, and Anna Perini. Crowdintent: Annotation of intentions hidden in online discussions. In *Proceedings of the 2nd IEEE/ACM International Workshop on CrowdSourcing in Software Engineering*, 2015.
128. David R. Musicant, Janara M. Christensen, and Jamie F. Olson. Supervised learning by training on aggregate outputs. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM)*, pages 252–261, 2007.
129. James W. Myers, Kathryn B. Laskey, and Tod S. Levitt. Learning bayesian networks from incomplete data with stochastic search algorithms. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI '99)*, pages 476–485, 1999.
130. Taiga Nakamura, Lorin Hochstein, and Victor R. Basili. Identifying domain-specific defect classes using inspections and change history. In *Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering (ISESE'06)*, pages 346–355, New York, NY, USA, 2006. ACM.
131. Richard E. Neapolitan. *Learning Bayesian networks*, volume 38. Prentice Hall Upper Saddle River, 2004.
132. Xia Ning and George Karypis. The set classification problem and solution methods. In *Proceedings of the 9th SIAM International Conference on Data Mining (SDM)*, pages 847–858, 2009.
133. Jonathan Ortigosa-Hernández, Juan D. Rodríguez, Leandro Alzate, Manuel Lucania, Iñaki Inza, and Jose A. Lozano. Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers. *Neurocomputing*, 92:98–115, 2012.
134. Giacomo Patrizi, Claudio Manna, C. Moscatelli, and Luciano Nieddu. Pattern recognition methods in human-assisted reproduction. *International Transactions in Operational Research*, 11(4):365–379, 2004.
135. Jose Manuel Peña, Jose A. Lozano, and Pedro Larrañaga. An improved Bayesian structural EM algorithm for learning Bayesian networks for clustering. *Pattern Recognition Letters*, 21(8):779–786, 2000.
136. Novi Quadrianto, Alex J. Smola, Tibério S. Caetano, and Quoc V. Le. Estimating labels from label proportions. *Journal of Machine Learning Research*, 10:2349–2374, 2009.
137. Catherine Racowsky, Lucila Ohno-Machado, Jihoon Kim, and John D. Biggers. Is there an advantage in scoring early embryos on more than one day? *Human Reproduction*, 24(9):2104–2113, 2009.
138. Rouhollah Rahmani and Sally A. Goldman. MISSL: Multiple-instance semi-supervised learning. In *Proceedings of the 23rd international conference on Machine learning (ICML)*, pages 705–712, 2006.
139. Marco Ramoni and Paola Sebastiani. Learning bayesian networks from incomplete databases. In *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI '97)*, pages 401–408, 1997.
140. Marco Ramoni and Paola Sebastiani. Parameter estimation in bayesian networks from incomplete databases. *Intelligent Data Analysis*, 2(1–4):139–160, 1998.
141. Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, 2010.

142. ESHRE Campus Course Report. Prevention of twin pregnancies after IVF/ICSI by single embryo transfer. *Human Reproduction*, 16(4):790–800, 2001.
143. Carsten Riggelsen. Learning bayesian networks from incomplete data: An efficient method for generating approximate predictive distributions. In *Proceedings of the 6th SIAM International Conference on Data Mining (SDM)*, pages 130–140, 2006.
144. Carsten Riggelsen and A. Feelders. Learning bayesian network models from incomplete data using importance sampling. In *Proceedings of Artificial Intelligence and Statistics*, pages 301–308, 2005.
145. Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.
146. Stephen A. Roberts. Models for assisted conception data with embryo-specific covariates. *Statistics in Medicine*, 26(1):156–170, 2007.
147. Juan Diego Rodríguez, Aritz Perez, and Jose A. Lozano. Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):569–575, 2010.
148. Juan Diego Rodríguez, Aritz Pérez-Martínez, David Arteta, Diego Tejedor, and Jose A. Lozano. Using multidimensional Bayesian network classifiers to assist the treatment of multiple sclerosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 42(6):1705–1715, 2012.
149. W.E. Roudebush, J.D. Wininger, A.E. Jones, G. Wright, A.A. Toledo, H.I. Kort, J.B. Massey, and D.B. Shapiro. Embryonic platelet-activating factor: an indicator of embryo viability. *Human Reproduction*, 17(5):1306–1310, 2002.
150. Stefan Rüping. SVM classifier estimation from group probabilities. In *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, pages 911–918, 2010.
151. Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.
152. Mehran Sahami. Learning limited dependence Bayesian classifiers. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD 1996)*, pages 335–338, 1996.
153. R.R. Saith, Ashwin Srinivasan, Donald Michie, and Ian L. Sargent. Relationships between the developmental potential of human in-vitro fertilization embryos and features describing the embryo, oocyte and follicle. *Human Reproduction Update*, 4(2):121–134, 1998.
154. Gideon Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.
155. Friedhelm Schwenker and Edmondo Trentin. Partially supervised learning for pattern recognition. *Pattern Recognition Letters*, 37:1–3, 2014.
156. Lynette Scott, Ruben Alvero, Mark Leondires, and Bradley Miller. The morphology of human pronuclear embryos is positively related to blastocyst development and implantation. *Human Reproduction*, 15(11):2394–2403, 2000.
157. Carolyn B. Seaman, Forrest Shull, Myrna Regardie, Denis Elbert, Raimund L. Feldmann, Yuepu Guo, and Sally Godfrey. Defect categorization: Making use of a decade of widely varying historical data. In *Proceedings of the 2nd ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 149–157, New York, NY, USA, 2008. ACM.

158. Emre Seli, Carlijn G. Vergouw, Hiroshi Morita, Lucy Botros, Pieter Roos, Cornelius B. Lambalk, Naoki Yamashita, Osamu Kato, and Denny Sakkas. Noninvasive metabolomic profiling as an adjunct to morphology for noninvasive embryo assessment in women undergoing single embryo transfer. *Fertility and Sterility*, 94(2):535–542, 2010.
159. Sundararajan Sellamanickam, Charu Tiwari, and Sathiya Keerthi Selvaraj. Regularized structured output learning with partial labels. In *Proceedings of the 12th SIAM International Conference on Data Mining (SDM)*, pages 1059–1070, 2012.
160. K. Sam Shanmugam and Arthur M. Breipohl. An error correcting procedure for learning with an imperfect teacher. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1(3):223–229, 1971.
161. Victor S. Sheng, Foster J. Provost, and Panagiotis G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 614–622, 2008.
162. Noam Shental, Aharon Bar-Hillel, Tomer Hertz, and Daphna Weinshall. Computing gaussian mixture models with EM using equivalence constraints. In *Proceedings of the 17th Annual Conference on Neural Information Processing Systems Workshops (NIPS)*, 2003.
163. Geoffrey Sher, Levent Keskintepe, Mory Nouriani, Roumen Roussev, and Joel Batzofin. Expression of sHLA-G in supernatants of individually cultured 46-h embryos: a potentially valuable indicator of 'embryo competency' and IVF outcome. *Reproductive BioMedicine Online*, 9(1):74–78, 2004.
164. Pan Shirui, Zhang Yang, Li Xue, and Wang Yong. Nearest neighbor algorithm for positive and unlabeled learning with uncertainty. *Journal of Frontiers of Computer Science and Technology*, 4(9):769–779, 2010.
165. Alex Simon and Neri Laufer. Assessment and treatment of repeated implantation failure (RIF). *Journal of Assisted Reproduction and Genetics*, 29(11):1227–1239, 2012.
166. Moninder Singh. Learning bayesian networks from incomplete data. In *Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference (AAAI 97)*, pages 534–539, 1997.
167. Padhraic Smyth, Usama M. Fayyad, Michael C. Burl, Pietro Perona, and Pierre Baldi. Inferring ground truth from subjective labelling of venus images. In *Proceedings of Advances in Neural Information Processing Systems 7 (NIPS)*, pages 1085–1092, 1994.
168. Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 254–263, 2008.
169. Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.
170. M. Soylemez and A. Tarhan. Using process enactment data analysis to support orthogonal defect classification for software process improvement. In *Proceedings of the Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement (IWSM-MENSURA)*, pages 120–125, Oct 2013.

171. Andrew L. Speirs, Alexander Lopata, Michael J. Gronow, Geoffrey N. Kellow, and Walter I.H. Johnston. Analysis of the benefits and risks of multiple embryo transfer. *Fertility and sterility*, 39(4):468–471, 1983.
172. Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1):62–72, 1991.
173. C.V. Steer, C.L. Mills, S.L. Tan, S. Campbell, and R.G. Edwards. The cumulative embryo score: a predictive embryo scoring technique to select the optimal number of embryos to transfer in an in-vitro fertilization and embryo transfer programme. *Human Reproduction*, 7(1):117–119, 1992.
174. Marco Stolpe and Katharina Morik. Learning from label proportions by optimizing cluster model selection. In *Proceedings of the European conference on Machine learning and knowledge discovery in databases (ECML/PKDD 2011)*, volume 3, pages 349–364, 2011.
175. Yu-Yin Sun, Yin Zhang, and Zhi-Hua Zhou. Multi-label learning with weak label. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 2010.
176. Marliese Thomas, Dana M. Caudle, and Cecilia M. Schmitz. To tag or not to tag? *Library Hi Tech*, 27(3):411–434, 2009.
177. F. Thung, D. Lo, and Lingxiao Jiang. Automatic defect categorization. In *Proceedings of the 19th Working Conference on Reverse Engineering (WCRE)*, pages 205–214, 2012.
178. Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *Int. J. Data Warehous. Min.*, 3(3):1–13, 2007.
179. Grigorios Tsoumakas, Min-Ling Zhang, and Zhi-Hua Zhou. Tutorial on learning from multi-label data. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*, 2009.
180. Ari Urkullu, Aritz Perez, and Borja Calvo. On the evaluation and selection of learning algorithms with crowdsourced data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015. Submitted.
181. Hamed Valizadegan, Quang Nguyen, and Milos Hauskrecht. Learning classification models from multiple experts. *Journal of Biomedical Informatics*, 46(6):1125–1135, 2013.
182. Linda C. van der Gaag and Peter R. de Waal. Multi-dimensional Bayesian network classifiers. In *Proceedings of the 3rd European Workshop in Probabilistic Graphical Models*, pages 107–114. Prague, 2006.
183. Eric Van Royen, Katelijne Mangelschots, Diane De Neubourg, Marion Valkenburg, Muriel Van de Meerssche, Greet Ryckaert, Willy Eestermans, and Jan Gerris. Characterization of a top quality embryo, a step towards single-embryo transfer. *Human Reproduction*, 14(9):2345–2349, 1999.
184. Shankar Vembu and Thomas Gärtner. Label ranking algorithms: A survey. In *Preference learning*, pages 45–64. Springer–Verlag, 2010.
185. Alexander Vezhnevets, Vittorio Ferrari, and Joachim M. Buhmann. Weakly supervised structured output learning for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 845–852, 2012.
186. Willem Waegeman, Krzysztof Dembczynski, and Eyke Hullermeier. Tutorial on multi-target prediction. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2013.

187. Willem Waegeman, Jan Verwaeren, Bram Slabbinck, and Bernard De Baets. Supervised learning algorithms for multi-class classification problems with partial class memberships. *Fuzzy Set Syst.*, 184(1):106–125, 2011.
188. Stefan Wagner. Defect classification and defect types revisited. In *Proceedings of the 2008 Workshop on Defects in Large Software Systems*, pages 39–40, New York, NY, USA, 2008. ACM.
189. Shuo Wang and Xin Yao. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(4):1119–1130, 2012.
190. Nils Weidmann, Eibe Frank, and Bernhard Pfahringer. A two-level learning method for generalized multi-instance problems. In *Proceedings of the 14th European Conference on Machine Learning (ECML)*, pages 468–479, 2003.
191. Peter Welinder, Steve Branson, Serge Belongie, and Pietro Perona. The multidimensional wisdom of crowds. In *Proceedings of Advances in Neural Information Processing Systems 23 (NIPS)*, pages 2424–2432, 2010.
192. Jacob Whitehill, Paul Ruvolo, Tingfan Wu, Jacob Bergsma, and Javier R. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Proceedings of Advances in Neural Information Processing Systems 22 (NIPS)*, pages 2035–2043, 2009.
193. Janyce Wiebe, Rebecca F. Bruce, and Thomas P. O'Hara. Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 246–253, 1999.
194. Man Leung Wong and Yuan Yuan Guo. Learning bayesian networks from incomplete databases using a novel evolutionary algorithm. *Decision Support Systems*, 45(2):368–383, 2008.
195. Xin-Shun Xu, Yuan Jiang, Xiangyang Xue, and Zhi-Hua Zhou. Semi-supervised multi-instance multi-label learning for video annotation task. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 737–740, 2012.
196. Shu-Jun Yang, Yuan Jiang, and Zhi-Hua Zhou. Multi-instance multi-label learning with weak label. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1862–1868, 2013.
197. Zoulficar Younes, Fahed Abdallah, and Thierry Denoeux. Evidential multi-label classification approach to learning from data with imprecise labels. In *Proceedings of the 13th International Conference on Information Processing and Management of Uncertainty (IPMU)*, pages 119–128, 2010.
198. Jing Zhang, Xindong Wu, and Victor S. Sheng. Imbalanced multiple noisy labeling. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):489–503, Feb 2015.
199. Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2014.
200. Ping Zhang, Weidan Cao, and Zoran Obradovic. Learning by aggregating experts and filtering novices: a solution to crowdsourcing problems in bioinformatics. *BMC Bioinformatics*, 14(Suppl-12):S5, 2013.
201. Zhi-Hua Zhou, Min-Ling Zhang, Sheng-Jun Huang, and Yu-Feng Li. Multi-instance multi-label learning. *Artif. Intell.*, 176(1):2291–2320, 2012.

202. Xingquan Zhu, Xindong Wu, and Qijun Chen. Eliminating class noise in large datasets. In *Proceedings of the 20th International Conference Machine Learning (ICML)*, pages 920–927, 2003.
203. S. Ziebe, K. Petersen, S. Lindenberg, A.G. Andersen, A. Gabrielsen, and A. Nyboe Andersen. Embryo morphology or cleavage stage: how to select the best embryos for transfer after in-vitro fertilization. *Human Reproduction*, 12(7):1545–1549, 1997.