Technical Report
EHU-KZAA-TR-2015/01

eman ta zabal zazu

Universidad del País Vasco Euskal Herriko Unibertsitatea

UNIVERSITY OF THE BASQUE COUNTRY
Department of Computer Science and Artificial Intelligence

On the optimal usage of labelled examples in semi-supervised multi-class classification problems

Jonathan Ortigosa-Hernández, Iñaki Inza, and Jose A. Lozano

April, 2015

Donostia-San Sebastián, Spain
www.ehu.es/ccia-kzaa
hdl.handle.net/10810/4562

On the optimal usage of labelled examples in semi-supervised multi-class classification problems

Technical Report EHU-KZAA-TR-2015/01

Jonathan Ortigosa-Hernández, Iñaki Inza, and Jose A. Lozano

Abstract

In recent years, the performance of semi-supervised learning has been theoretically investigated. However, most of this theoretical development has focussed on binary classification problems. In this paper, we take it a step further by extending the work of Castelli and Cover [1] [2] to the multi-class paradigm. Particularly, we consider the key problem in semi-supervised learning of classifying an unseen instance $\mathbf{x}$ into one of K different classes, using a training dataset sampled from a mixture density distribution and composed of l labelled records and u unlabelled examples. Even under the assumption of identifiability of the mixture and having infinite unlabelled examples, labelled records are needed to determine the K decision regions. Therefore, in this paper, we first investigate the minimum number of labelled examples needed to accomplish that task. Then, we propose an optimal multi-class learning algorithm which is a generalisation of the optimal procedure proposed in the literature for binary problems. Finally, we make use of this generalisation to study the probability of error when the binary class constraint is relaxed.

Keywords: Semi-supervised learning, probability of error, labelled and unlabelled samples, multi-class classification.

Contents

1 Introduction

Throughout recent years, the problem of learning from both labelled and unlabelled observations has been of practical relevance. In many applications, an enormous amount of unlabelled examples is available with little cost, whilst obtaining enough labelled examples to learn a classifier may be costly and time consuming. In such cases, semi-supervised learning (SSL) [9] appears to be a tool that is able to obtain accurate classifiers in such circumstances.

Within the state-of-the-art literature, SSL has been empirically and theoretically studied. Regarding the practical applications, it has been used to tackle (i) binary problems [5], (ii) problems with multiple class values [10], or even (iii) multi-dimensional problems [11], where several multi-class variables have to be predicted simultaneously. The probability of error of SSL has also been theoretically investigated. However, the scope of the studied problems does not cover the entire range of the practical applications. The majority of the theoretical works proposed in this area have mainly focussed on standard binary problems [1] [2] [3] [7]. To the best of our knowledge, only in [8], is multi-class framework explicitly tackled; yet, it has been studied with a slightly different perspective as how it is in this paper. Moreover, most of the works assume the datasets have a large enough number of labelled observations [7] [8], which is an unnatural situation in this scenario.

For those reasons, we think that it is interesting and demanding to generalise several of the theoretical findings of the state-of-the-art literature in SSL binary problems to the scenarios where there are more than two classes, concentrating on the cases where the number of labelled observations is minimal. However, as we show throughout the whole paper, the previous state-of-the-art studies do not straightforwardly work for multi-class problems, so there are several theoretical gaps that must be covered.

Therefore, in order to allow a potential enlargement of the scope of the theoretically studied SSL problems, in this paper, we first perform an exhaustive review of the previous theoretical findings. It is focussed on the frameworks utilised in each study, the feasibility of their conclusions to the multi-class frameworks and the remaining open questions found in them. So, guided by this, we contribute with a natural extension to the multi-class paradigm of the SSL binary framework already proposed by Castelli and Cover [1] [2]. This extension is performed by addressing the following issues:

- First, the proposal of an optimal theoretical SSL algorithm able to work in the multi-class framework: PC$_{\text{SSL}}$ (**P**ermutation of **C**omponents in **S**emi-**S**upervised **L**earning). It is a natural extension of the optimal procedure proposed in [1] and [2] for binary problems.

- Even under the assumption of having ∞ unlabelled records and identifiability of the decision regions, labelled samples are still needed to determine the labels of the K decision regions. However, what is the minimum number of labelled records needed to uniquely determine the decision regions? In the case of binary problems, just one labelled datum is needed [1]. In the multi-class scenario, however, the calculation of this value becomes more complex. For that reason, in this paper, we define and calculate l_K as the expected minimum number of labelled records to uniquely determine those K decision regions.

- A formula to calculate the probability of error, $P_e(l, \infty)$ (given l labelled instances and an infinite number of unlabelled records), for SSL problems where the binary constraint is relaxed and the pairwise intersections among the decision regions are empty. When the regions are non-mutually disjoint, upper and lower bounds are given for $P_e(l, \infty)$, generalising the statements of [1] [2] to the multi-class scenario. In both scenarios, $P_e(l, \infty)$ decreases to the Bayes error exponentially fast in l.

The rest of the paper is structured as follows: In Section 2, the notation, the properties and the proposed multi-class framework are introduced. Then, the state-of-the-art literature is reviewed in Section 3. Section 4 reviews the framework proposed in [1] and [2] for binary problems, highlighting the issues that must be solved before extending it. Our algorithm PC$_{\text{SSL}}$ is proposed in Section 5. In that section, the Voting learning procedure [8], the recently proposed multi-class approximated method, is also introduced. In Section 6, the problem of determining the minimum number of labelled records needed to determine the decision regions in the multi-class framework is tackled. Whilst Section 7 is devoted to the calculation of the probability of error in the SSL multi-class scenario, in Section 8, we carry out an empirical experimentation on the contributions of this paper. Then, the issue of extending the contributions of this paper to practical SSL is approached in Section 9. Section 10 provides a summary of the paper. Lastly, the source code. which ensures the replicability of the exposed studies, can be found in the appendix.

2 General notation and Framework

Firstly, we introduce the multi-class framework which will be used throughout the rest of the paper and which has been borrowed and extended from that proposed by Castelli and Cover in the key works [1] and [2] for binary problems.

2.1 Framework

As we want to study the optimal probability of error $P_e(l,u)$ of classifying the instance $(\mathbf{x}^{(0)}, c^{(0)})$ in the SSL multi-class scenario having l labelled instances and u unlabelled records, the following framework is proposed: Let $D = \mathcal{L} \cup \mathcal{U}$ be a training dataset of a common SSL problem with K classes which can be divided into two different subsets: $\mathcal{L}$, the set of l labelled examples $\{(\mathbf{x}^{(1)}, c^{(1)}), \ldots, (\mathbf{x}^{(l)}, c^{(l)})\} = \{(\mathbf{x}^{(n)}, c^{(n)})\}_{n=1}^{l}$, and $\mathcal{U}$, the set of u unlabelled examples $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(u)}\} = \{\mathbf{x}^{(m)}\}_{m=1}^{u}$. Due to the fact that the applications of SSL deal with very few labelled examples and a huge amount of unlabelled data ($l/u \sim 0$) [9], the theoretical studies usually make the reasonable assumption of having $l > 0$ labelled and ∞ unlabelled records. Moreover, with this assumption and by proposing an optimal learning algorithm, we can establish a fundamental limit in the performance of any existing SSL multi-class algorithm. Therefore, unless otherwise specified, we assume that $u = \infty$. Then, let the class labels $\{c^{(n)}\}_{n=1}^{l}$ be l i.i.d. random values where the prior probability of observing a sample of class c_i is $\eta_i = P(C = c_i) > 0, i = 1, \ldots, K$ and $\sum_i \eta_i = 1$. We also assume that each observation $\mathbf{x} \in \mathcal{L}$ is i.i.d. according to a mixture component $f(\mathbf{x}|C = c_i; \boldsymbol{\theta}_i) \in \mathcal{F}$, where $\mathcal{F}$ is a function set containing the mixture components of a mixture density. There, $\boldsymbol{\theta}_i$ stands for the set of the parameters of the mixture component i, being $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)$ the vector of the parameters of the whole mixture. For simplicity of notation, henceforth, we denote $f(\mathbf{x}|C = c_i; \boldsymbol{\theta}_i)$ by $f_i(\mathbf{x})$. Then, we define the mixture joint density, which generates the labelled samples, as

$$f(\mathbf{x}, c) = \sum_{i=1}^{K} \eta_i f_i(\mathbf{x}) \mathbb{1}(c = c_i), \tag{1}$$

where the function $\mathbb{1}(c = c_i)$ is 1 if $c = c_i$, and 0 otherwise, i.e. each mixture component models just one class value.

The infinite unlabelled samples appear to be i.i.d. random variables distributed according to the mixture density given by

$$f(\mathbf{x}) = \sum_{i=1}^{K} \eta_i f_i(\mathbf{x}) \tag{2}$$

and which corresponds to the marginal of $f(\mathbf{x}, c)$ on $\mathbf{x}$.

Let $(\mathbf{x}^{(0)}, c^{(0)})$ be a instance to be classified and distributed according to the joint density (1). As we want to infer $c^{(0)}$ from the observation $\mathbf{x}^{(0)}$, when $\forall i$, $f_i(\mathbf{x})$ and η_i are known, the optimal classifier is given by the Bayes decision rule (BDR)

$$\hat{c}^{(0)} = \arg\max_i \eta_i f_i(\mathbf{x}^{(0)}) \tag{3}$$

with a corresponding probability of error

$$e_B = 1 - \sum_{i=1}^{K} \eta_i \int\limits_{R_i} f_i(\mathbf{x}) d\mathbf{x}, \tag{4}$$

which is called the Bayes error and is the highest lower bound of the probability of error of any classification rule. There,

$$R_i = \{\mathbf{x} : \eta_i f_i(\mathbf{x}) - \max_{i' \neq i} \eta_{i'} f_{i'}(\mathbf{x}) > 0\} \tag{5}$$

is the region where $\eta_i f_i(\mathbf{x})$ is maximum and so the instances are assigned to the class c_i. However, this classifier cannot be used in practise as, in general, $f_i(\mathbf{x})$ and η_i are unknown.

2.2 Identifiability

Having an infinite amount of unlabelled examples ($u = \infty$) available is equivalent to knowing $f(\mathbf{x})$, i.e. the mixture density can almost surely be recovered from the unlabelled data [1]. So, in order to be able to take advantage of the information provided by the unlabelled data, in this framework, we assume that the mixture $f(\mathbf{x})$ is identifiable, i.e. the components of the mixture $f_1(\mathbf{x}), \ldots, f_K(\mathbf{x}) \in \mathcal{F}$ and the class priors $\eta_1, \ldots, \eta_K$ can be uniquely decomposed from the density function. This assumption is well-grounded since it holds for most of the well-known distributions. In the continuous case and having a finite K, $f(\mathbf{x})$ is identifiable iff $\mathcal{F}$ is said to be linearly independent [12]. i.e. for real constants $\alpha_i, i = 1, 2, \ldots, K$,

$$\sum_{i=1}^{K} \alpha_i f_i(\mathbf{x}) = 0 \Longrightarrow \forall i, \alpha_i = 0.$$

Particularly, it has also been shown that the mixtures of univariate Gaussian, Gamma, exponential, Cauchy and Poisson functions are identifiable iff there are no empty components ($\forall i, \exists \mathbf{x}$ s. t. $f_i(\mathbf{x}) \neq 0$), and there are not two components with the same parameters ($\forall i \neq j, \exists \mathbf{x}$ s. t. $f_i(\mathbf{x}) \neq f_j(\mathbf{x})$). In general, discrete distributions are not identifiable, except for the case of binomial and multinomial distributions. They are identifiable if $K < \infty$ [13] [14] [15].

2.3 Probability of error in the absence of labelled examples

Next, if the generative model is identifiable, we can recover all the single-component distributions and all the class priors of the generative model from just the unlabelled data. However, it is only identifiable up to a permutation π of its single-components. That is, in a scenario of absence of labelled data, each recovered component can be labelled with a conventional name j which does not necessarily coincide with the real label i of the component. This permutation can be defined as $\pi = (\pi(1), \ldots, \pi(K)) \in S_K$, where each element $\pi(j) = i$ denotes that the j-th decomposed component distribution, i.e. $f_{\pi(j)}(\mathbf{x})$, is associated to the i-th class value, and S_K represents the set of all possible component-label correspondences. Then, the mixture distribution (eq. (2)) can be expressed as follows:

$$f(\mathbf{x}) = \sum_{i=1}^{K} \eta_i f_i(\mathbf{x}) = \sum_{j=1}^{K} \eta_{\pi_c(j)} f_{\pi_c(j)}(\mathbf{x}),$$

where π_c is the unknown correct correspondence between real labels i and decomposed components j, which cannot be determined without labelled records. Generalising the conclusions of [9] for binary problems, it can be seen that, by means of the infinite unlabelled records, the set containing all the possible models which can generate the data is reduced to just only a set containing $K!$ possibilities (where the real generative model is included). Unfortunately, without labelled data, this reduction is pointless, as shown in the following theorem:

Theorem 1. ***(Probability of error with no labelled data)***[1] *The probability of error of classifying a new sample $(\mathbf{x}^{(0)}, c^{(0)})$ of a K-class problem with any classifier learnt with no labelled examples and any $u \geq 0$ number of unlabelled samples coincides with the probability of error of the random classifier, which it is equal to*

$$P_e(0,u) = e_0 = \frac{(K-1)}{K},\ \forall u \geq 0. \tag{6}$$

Proof.

- **When $f(\mathbf{x})$ is unknown:** When there are not enough unlabelled records ($u < \infty$) to determine the mixture density (and its components), the class $c^{(0)}$ of the unseen distance must be determined by the uniformly random classifier. In such a case, let the event A be defined as the probability of correctly choosing the right class for $c^{(0)}$ over a choice of K different classes; $P(A) = 1/K$. Then, the probability of committing an error is

$$P_e(0,u) = 1 - P(A) = \frac{(K-1)}{K}, \forall u < \infty.$$

- **When $f(\mathbf{x})$ is known:** With ∞ unlabelled examples, the mixture is identifiable up to a permutation π of the components. Let the observation $\mathbf{x}^{(0)}$ be drawn from the j-th decomposed component, $f_{\pi(j)}(\cdot)$, i the unknown true label such that $\pi_c(j) = i$, and let us define the following two events:

$$B_1 \triangleq \{\eta_{\pi_c(j)} f_{\pi_c(j)}(\mathbf{x}^{(0)}) - \max_{j' \neq j} \eta_{\pi_c(j')} f_{\pi_c(j')}(\mathbf{x}^{(0)}) > 0\}$$

$$B_2 \triangleq \{\eta_{\pi_c(j)} f_{\pi_c(j)}(\mathbf{x}^{(0)}) - \max_{j' \neq j} \eta_{\pi_c(j')} f_{\pi_c(j')}(\mathbf{x}^{(0)}) < 0\}$$

B_1 is the event which represents achieving a correct answer in the application of the Bayes decision rule over $\mathbf{x}^{(0)}$ and B_2 represents the opposite- By the definition, $P(B_1) = (1 - e_B)$ and $P(B_2) = e_B$. Then, the probability of error is

$$\begin{aligned} P_e(0,\infty) &= P(\hat{c}^{(0)} \neq c^{(0)}) = 1 - P(\hat{c}^{(0)} = c^{(0)}) = \\ &= 1 - P(\hat{c}^{(0)} = c^{(0)}|B_1)P(B_1) - P(\hat{c}^{(0)} = c^{(0)}|B_2)P(B_2) = \\ &= 1 - P(\pi_a)(1 - e_B) - P(\pi_a)e_B. \end{aligned}$$

where π_a and π_b are two permutations such that $\pi_a(j) = i$ and $\pi_b(j') = i$.As no labelled data are provided to determine the correspondence π, it has to be randomly chosen. Then, the probability of choosing those

[1] This holds true independently of the value of the class priors $\eta_1, \ldots, \eta_K$.

permutations is $P(\pi_a) = P(\pi_b) = (K-1)!/K! = 1/K$. After substituting these probabilities in the previous formula and after some algebra, we obtain the same expression:

$$P_e(0,\infty) = \frac{(K-1)}{K}.$$

□

Then, in SSL, the use of several labelled examples is crucial. Only for $l > 0$, the unlabelled records influence the reduction of the probability of error.

3 Literature review

The probability of error of SSL has been investigated in the literature. Throughout recent years, several key results have been presented on this topic. Although these papers theoretically approach SSL by means of different frameworks and under different assumptions, their findings are equivalent in most of the cases. In the following paragraphs, we taxonomise these theoretical proposals into three different subsets assumed in the SLL community (a summary can be found in Table 1):

1. Papers which deal with correct models[2], i.e. the semi-supervisely learned models match the generative models,
2. works in which incorrect models are assumed, i.e. the models do not match the generative distribution, and
3. papers dealing with imperfect models, i.e. those models which, despite not matching the generative models perfectly, have a presumedly small error.

Although incorrect models and imperfect models have been clearly defined in the literature, they are almost equivalent. Neither of them match the generative distribution of the data which causes performance degradation of the learned classifiers. The subtle difference relies on the perspective of the authors towards them. While the authors who deal with incorrect models only perceive the degradation of the performance, the authors dealing with imperfect models study the impact of the difference between the generative and learnt models on the resulting error, or they even try to improve the safeness of SSL techniques. In the following paragraphs, we review several contributions to these three different approaches.

3.1 Correct models

3.1.1 Ratsaby and Venkatesh [3]

The authors try to shed some light on the question "How many unlabelled examples is each labelled example worth?" under the Probably Approximately Correct (PAC) learning framework. Their goal is to determine how the error rate depends on the sample sizes l and u, and on the dimensionality n. In order to achieve this goal, several assumptions are made: (1) Learning the correct model. (2) Two-class multivariate Gaussian mixture problem with equal unit variance matrices ($\boldsymbol{\theta}_i = \{\boldsymbol{\mu}_i, \boldsymbol{I}\}$). (3) Equiprobable class priors, i.e. $\eta_i = 1/2$. (4) $\mathbf{x}$ is n-dimensional.

[2] Note that, by the assumptions of an infinite number of unlabelled examples and identifiability, our paper relies on this category.

Model	Ref.	Framework	Addressed Question	Assumed Model	MC	Conclusion
CORRECT	[3]	PAC learning	How many unlabelled examples is each labelled example worth?	Multi-variate Gaussian mixture model with unit covariance matrices and equiprobable classes.	No	If the parametric model assumptions are satisfied, labelled examples are exponentially more valuable than unlabelled examples in reducing the error.
	[1], [2]	Decision theory	How the optimal probability of error varies in l and u?	Identifiable mixture densities.	No	
	[4]	Parameter estimation	What is the contribution of unlabelled data in the parameter estimation?	Generic parametric models $p(x, c\|\theta) = p(x\|\theta)p(c\|x, \theta)$.	Yes	Unlabelled data always helps due to the fact that the Fisher information is increased.
INCORRECT	[5]	Bayesian networks	Does unlabelled data always help?	Naive Bayes, and Tree-augmented naive Bayes.	No	Unlabelled data only helps when the learned matches the generative model.
	[6]	Parameter estimation	What is the relationship between the model misspecification and performance degradation?	Identifiable mixture densities (Gaussian mixtures for the examples).	No	As the number of unlabelled record increases, the probability of degradation is positive with incorrect model.
IMPERFECT	[7]	Density estimation	What is the value of labelled and unlabelled data when the assumed densities do not follow the parametric model?	Two equiprobable n-dimensional spherical Gaussian mixtures. Let e be the difference between them.	No	The error is reduced exponentially with the number of labelled records until e. After that, the error is only reduced polynomially fast.
	[8]	Density estimation	What is the convergence rate of the error?	Identifiable mixture densities.	Yes	Similar results to [1] and [2] for the multi-class scenario.

Table 1: Summary of the theoretical SSL state-of-the-art literature (MC = multi-class).

With the aim of reaching a rough measure on the value of one labelled example in terms of unlabelled samples, the authors calculate how many unlabelled records must be added to a supervised algorithm to remove just one labelled record while keeping a similar error. The result is the following:

$$\frac{u_{SSL}}{(l_{SUP} - l_{SSL})} = \frac{zn^n}{\epsilon^3 \delta \log n},$$

where u_{SSL} is the number of unlabelled examples in a semi-supervised problem, l_{SUP} is the number of labelled instances in a purely labelled problem, l_{SSL} is the number of labelled examples in the semi-supervised problem, z is a constant, ϵ is the upper bound of the error with at least $(1 - \delta)$ confidence, and n is the dimensionality. They conclude that unlabelled data are extremely helpful due to the fact that they reduce the demands on the number of labelled examples. Then, each labelled datum is more valuable and the probability of error decreases exponentially fast in l_{SSL}, not polynomially fast in l_{SUP}, as happens in supervised learning.

3.1.2 Castelli and Cover [1] [2]

The same conclusion as in the previous work is reached but from the perspective of the decision theory framework and by weakening several assumptions. The detailed explanation of their findings can be found in Section 4.

3.1.3 Zhang and Oles [4]

The authors address the problem of the value of unlabelled data, i.e. how unlabelled records help in reducing the probability of error, by analysing their efficacy in the estimation of the parameters of the model. They argue that unlabelled examples have a positive impact on the efficacy of the estimations of the real model parameters $\boldsymbol{\theta}$, in the cases of combining both (1) parametric generative models defined as $p(x, c|\boldsymbol{\theta}) = p(x|\boldsymbol{\theta})p(c|x, \boldsymbol{\theta})$, and (2) SSL techniques where a classifier is trained with labelled and unlabelled data in an iterative manner. Then, the authors claim that under the correct model assumption, adding unlabelled data always helps because Fisher information is increased:

$$I_{(l+u)}(\boldsymbol{\theta}) = I_l(\boldsymbol{\theta}) + I_u(\boldsymbol{\theta}),$$

where $I_{(l+u)}(\boldsymbol{\theta})$ is the Fisher information of $\boldsymbol{\theta}$ using both labelled and unlabelled subsets, $I_l(\boldsymbol{\theta})$ using the labelled subset, and $I_u(\boldsymbol{\theta})$ using the unlabelled data.

3.1.4 Cohen et al. [5]

The authors address the question of whether unlabelled data always helps. By means of Bayesian network classifiers (naive Bayes and tree augmented naive Bayes models), they focus on the convergence of the semi-supervised maximum likelihood estimator of the model, $\boldsymbol{\theta}^*$,. They argue that the limiting value of the MLE, as the number of labelled and unlabelled records increases, is a linear combination of the supervised and unsupervised expected log-likelihood functions:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \Big[\beta E\big[\log p(c, \mathbf{x}|\boldsymbol{\theta})\big] + (1-\beta)E\big[\log p(\mathbf{x}|\boldsymbol{\theta})\big]\Big],$$

where β is the probability of sampling labelled data, i.e. the ratio of the amount of labelled and unlabelled observations. They conclude that unlabelled examples always improve the performance when the correct model assumption is met, and may degrade it when the opposite happens.

3.2 Incorrect models

3.2.1 Yang and Priebe [6]

Under the assumption of learning the correct model, SSL techniques seem to work appropriately. However, when this requirement is not met, performance degradation may occur in the classifiers as unlabelled examples are introduced. Therefore, in order to study the degradation, the authors define $\boldsymbol{\theta}_l^*$ as the limiting value of the supervised MLE (as the number of labelled data increases) of the real model parameters $\boldsymbol{\theta}$, and $\boldsymbol{\theta}_u^*$ as the limit of the unsupervised MLE (as the number of unlabelled records increases) of $\boldsymbol{\theta}$, assuming that the generative model is a finite Gaussian mixture model ($\boldsymbol{\theta}_i = \{\mu_i, \sigma\}$) and the estimators exist under mild regularity conditions. First, the authors corroborate the achievements of [1] when the correct model assumption is met by proving that both limits tend to the same parameter value. However, when the learnt model is misspecified, the supervised and the semi-supervised MLE parameters may converge to different values, i.e. $\boldsymbol{\theta}_l^* \neq \boldsymbol{\theta}_u^*$. They also state that for any fixed finite l or $l \to \infty$, as $l/u \to 0$, the limit of the maxima of the semi-supervised likelihood parameters is the unsupervised MLE limit $\boldsymbol{\theta}_u^*$, and degradation may appear: If $P_e(f(\mathbf{x}, c|\boldsymbol{\theta}_l^*)) < P_e(f(\mathbf{x}, c|\boldsymbol{\theta}_u^*))$, then for a given misspecified model, $\exists l$, s.t.

$$\lim_{u\to\infty} P_e(l, u) = P\{P_e(f(\mathbf{x}, c|\boldsymbol{\theta}_l^*)) < P_e(f(\mathbf{x}, c|\boldsymbol{\theta}_u^*))\} > 0.$$

That is, for incorrect models, SSL yields degradation with positive probability as $u \to \infty$.

3.3 Imperfect models

3.3.1 Sinha and Belkin [7]

The authors focus on the situation when the correct model assumption is only satisfied to a certain degree of precision, either because the assumed model is correct but the dataset is imperfect or because the assumed model does not follow the generative model. For the purpose of this paper, we aim for the latter case. There, ϵ is defined as a perturbation size, i.e. a rough measure that indicates to what extent the true model differs from the assumed model.

It is proved that, under the assumption of having two equiprobable spherical Gaussian mixture components as generative models, as labelled examples are added to a training set with infinite unlabelled records, the probability of error is reduced exponentially in the number of labelled examples (as argued in previous works) but only until $e_B + \epsilon$. After that, the perturbation ϵ is only reduced polynomially fast in l. Moreover, they also state that, for a positive perturbation size, there is a number of unlabelled examples that beyond which any extra additions do not decrease the probability of error.

3.3.2 Chen and Li [8]

Although the authors extend the findings of [7] by assuming an imperfect model, it is better to remark their efforts to theoretically deal with the multi-class framework. Under the correct model assumption, they show that labelled examples reduce the probability of error exponentially fast, as happens in binary problems. They also proposed an approximated algorithm (called Voting) that can utilise the unlabelled data efficiently, i.e. achieving a fast convergence rate. Although, to the best of our knowledge, it is the only theoretical algorithm proposed for the multi-class framework, it does not achieve the optimal probability of error as we demonstrate throughout the paper.

4 Semi-supervised learning in binary problems

In this section, we review the key works [1] and [2] highlighting why new strategies must be adopted in the multi-class scenario since they cannot be straightforwardly generalised.

4.1 Obtaining the binary classifier

Under the assumptions of (i) learning the correct model, (ii) having the unlabelled samples distributed according to the identifiable mixture density $f(\mathbf{x}) = \eta_1 f_1(\mathbf{x}) + \eta_2 f_2(\mathbf{x})$, and (iii) having $f_1(\cdot)$, $f_2(\cdot)$, η_1, and η_2 $(= 1 - \eta_1)$ unknown, the authors define a procedure (see Algorithm 1) to obtain the optimal binary classifier in SSL.

Algorithm 1 Optimal theoretical procedure for SSL binary problems [1] [2]

```
1: LEARNING TASK:
     • Stage 1 Use unlabelled set $\mathcal{U}$ to obtain $f(\mathbf{x})$ and, by identifiability, a permutation of its components
       ($f_{\pi(1)}(\cdot)$, $f_{\pi(2)}(\cdot)$, $\eta_{\pi(1)}$, and $\eta_{\pi(2)}$).
     • Stage 2 By means of the likelihood ratio test and the labelled set $\mathcal{L}$, determine the correspondence
       between the real classes and the current mixture components:
       $\hat{\pi}(1) = 1$ and $\hat{\pi}(2) = 2$, or $\hat{\pi}(2) = 1$ and $\hat{\pi}(1) = 2$.
2: CLASSIFICATION TASK:
     • Stage 3 Assign the sample $\mathbf{x}^{(0)}$ to the class induced by the BDR using the learned model.
```

The procedure can be divided into two major parts: (a) the *learning task*, where a model is learnt using the training dataset D, and (b) the *classification task*, where the unseen instances are classified according to the previously learnt model. The learning task is split into two stages. First, the components of the mixture are identified by means of the unlabelled subset $\mathcal{U}$ (Stage 1), and then, the labelled subset $\mathcal{L}$ is used in the likelihood ratio test to assign a class to each component (Stage 2). Finally, the classification task is composed of just one stage, namely Stage 3, in which the BDR is used to determine the class of the unseen instance given the assignment of the two previously made mixture components.

According to [1], this procedure is optimal, i.e. it achieves the highest lower bound of the probability of error of any semi-supervised classifier, since all the three stages are optimal. By means of identifiability and the correct model assumption, the recovered components are a permutation of the components of the unknown real model. The likelihood ratio test is optimal for two simple hypotheses and the BDR (eq. (3)) is the optimal classification rule. Unfortunately, this optimal procedure cannot be directly transferred to the multi-class scenario. Although, both Stage 1 and Stage 3 can be straightforwardly used to deal with $K \geq 2$ classes, the optimality of Stage 2 can only be guaranteed for $K = 2$. In the multi-class framework, new procedures must be proposed for Stage 2 as we need to deal with more than two simple hypotheses. In Section 5, we tackle this problem.

4.2 Minimum number of labelled examples

Labelled records are needed to correctly determine the correspondence between the classes and the decomposed mixture components (Stage 2 of Algorithm 1). But, how many labelled examples are needed to carry out such a task?

It is shown in [1] that, for the case of $K = 2$, just one labelled example is enough. Once the two components have been identified (as $f_{\pi(1)}(\cdot)$ and $f_{\pi(2)}(\cdot)$) in Stage 1, with just one labelled datum $(\mathbf{x}, c_i)$, the correspondence π can be, correctly or incorrectly, uniquely determined. By means of the likelihood ratio test (Stage 2), the component that is maximum ($f_{\pi(j)}(\cdot)$) in the region R_j where the instance lies is labelled with the label of the instance ($\pi(j) = i$). Then, the other component is labelled by a process of elimination.

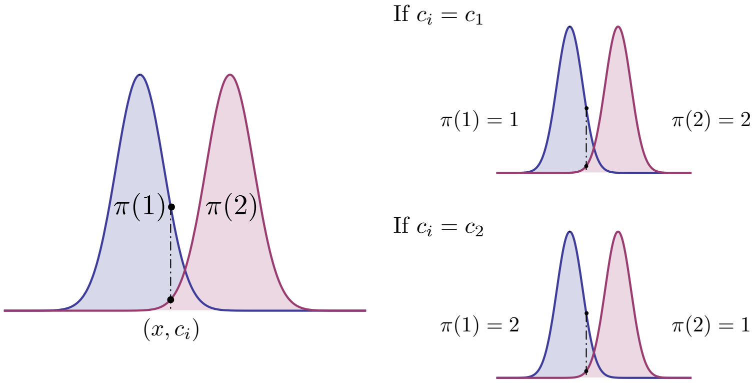


Figure 1: Labelling a mixture of two Gaussian distributions.

Setting up an example, in Figure 1 the labelled record lies in the first region, where $f_{\pi(1)}(\cdot)$ is maximum. If the instance has the label 1, then the first recovered component would be labelled as class 1, and the second, by process of elimination, as class 2. On the contrary, the first recovered component would be class 2, and the second, class 1. But is just one labelled example enough to assign a label to each component in the multi-class paradigm? The answer is no. With more than two classes and a labelled datum, we can only identify just one component, the one where the datum seems to belong to. For the rest of the components, there is not enough information in the subset $\mathcal{L}$. So, how much labelled data is needed to uniquely determine the correspondence π of a K class problem? We deal with this issue in Section 6.

4.3 Probability of error

Under the proposed binary framework, the probability of error, $P_e(l, u)$, is calculated in [1]. First of all, the authors prove for the case of binary problems that ($P_e(0, u) = 1/2, \forall u \geq 0$ (Theorem 1)). Then, they stated that with infinite labelled examples the Bayes error is reached ($P_e(\infty, u) = e_B, \forall u \geq 0$), and that the probability of error of having just one labelled example and no unlabelled data is $P_e(1, 0) \leq 2\eta_1\eta_2 \leq 1/2$.

After all these specific scenarios, the authors make use of Algorithm 1 to study the value of $P_e(l, \infty)$. First, they analyse the case of $P_e(1, \infty)$, where only one labelled datum plus infinite unlabelled records are available. Under the correct model assumption and by identifiability, Stage 1 cannot lead to a classification error. Therefore, in this case, a classification error only occurs when either Stage 2 or Stage 3 yields an incorrect answer, i.e. either (i) the classes of the mixture components are reversed or either (ii) the BDR misclassifies the instance. When both Stage 2 and Stage 3 result in wrong answers (the classes of the mixtures are reversed and the BDR misclassifies the instance), both mistakes cancel each other out in the 2-class scenario. Let the event $A \triangleq \{\text{error in Stage 2}\}$, then $P(A) = e_B$ and the probability of error is calculated as:

$$\begin{aligned} P_e(1,\infty) &= P(\hat{c}^{(0)} \neq c^{(0)}) = P(\hat{c}^{(0)} \neq c^{(0)}|A)P(A) + P(\hat{c}^{(0)} \neq c^{(0)}|\bar{A})P(\bar{A}) = \\ &= (1-e_B)e_B + e_B(1-e_B) = 2e_B(1-e_B). \end{aligned}$$

In general, for the case of l labelled examples, the authors follow a similar reasoning to the calculation of $P_e(1,\infty)$, i.e. determining when an error is committed in just one of the two previously exposed stages (Stage 2 and Stage 3). However, in this general case, $P(A)$ does not coincide with e_B, and therefore it must be calculated. Under these premises, they reach the conclusion that the probability of error is:

$$P_e(l,\infty) - e_B = exp\big\{-lZ + o(l)\big\},$$

where $Z = -\log\left\{2\sqrt{\eta_1\eta_2}\int\sqrt{f_1(\mathbf{x})f_2(\mathbf{x})d\mathbf{x}}\right\}$ is the Bhattacharyya distance between the densities $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ multiplied by a term equal to $\log\left(2\sqrt{\eta_1\eta_2}\right)$. In [2], they extend their work to the case of $u < \infty$, reaching to

$$P_e(l,u) - e_B = O\Big(\frac{1}{u}\Big) + exp\big\{-lZ + o(l)\big\}.$$

The authors conclude that, in the case of binary problems, it turns out that unlabelled samples are only polynomially valuable, whilst labelled samples are exponentially valuable in reducing the error. So then, what is the probability of error, $P_e(l,\infty)$, when the binary class constraint is relaxed? Does this conclusion still hold in those cases? In Section 7, our main objective is to address these.

5 Semi-supervised multi-class learning strategies

Guided by the aforementioned concerns, we now tackle the first of them; proposing a strategy for Stage 2 which is able to determine a correspondence π by using a labelled set $\mathcal{L}$ with $K \geq 2$ classes, i.e. *to assign each mixture component to a specific class*. In the following subsections, we first introduce Voting [8] as the only method for Stage 2 where the binary constraint is relaxed which has already been proposed in the literature. However, since it does not make optimal usage of the data, we propose PC$_{\text{SSL}}$, an optimal multi-class learning strategy for Stage 2, which is a natural extension and generalisation of the one proposed by [1]. So, in the studied scenario, the whole multi-class procedure remains as can be seen in Algorithm 2.

Algorithm 2 Theoretical procedure for SSL multi-class problems

```
LEARNING TASK:
    • Stage 1 Use unlabelled set $\mathcal{U}$ to obtain $f(\mathbf{x})$ and, by identifiability, a permutation of its components
      ($f_{\pi(j)}(\cdot)$, and $\eta_{\pi(j)}, j = 1, ..., K$).
    • Stage 2 Use the labelled set $\mathcal{L}$ to determine the correspondence between the classes and the mixture components, i.e. the permutation of the components $\hat{\pi}$, by means of the semi-supervised
      multi-class learning procedure: (i) Voting [8], or (ii) our proposal PC$_{\text{SSL}}$
CLASSIFICATION TASK:
    • Stage 3 Assign the sample $\mathbf{x}^{(0)}$ to the class induced by the BDR using the learned model.
```

5.1 Voting

In [8], the authors propose Voting as a simple method to determine the permutation π by extending the majority vote method for binary problems [17] to the multi-class framework.

There, it is assumed that the regions R_j (see equation 5) are known by identifiability, and that the observations of $\mathcal{L}$, i.e. $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(l)}\}$, can be split into K different subsets named $\mathcal{L}_i$, for $i = \{1, ..., K\}$, such that, each $\mathcal{L}_i$ stands for the set containing all the observations in $\mathcal{L}$ of the class value c_i. Under these premises, the learning method is as follows: First, it counts the labels in each region and then, assigns the permutation which maximises the total number of counts. Formally, it can be defined as follows:

$$\hat{\pi}_v = \arg\max_{\pi} V(\pi; \mathcal{L}) = \arg\max_{\pi} \sum_{i=1}^{K} |\mathcal{L}_i \cap R_{\pi^{-1}(i)}|. \tag{7}$$

where $|\mathcal{L}_i \cap R_{\pi^{-1}(i)}|$ is the number of examples of class c_i found in the region $R_{\pi^{-1}(i)}$. Although Voting is asymptotically optimal (as $l \to \infty$), it does not make optimal usage of the dataset when l is relatively small, the natural domain for SSL.

5.2 PC$_{\text{SSL}}$

Due to the aforementioned drawbacks of the Voting procedure, we propose a new theoretical SSL strategy which makes optimal usage of the labelled data. It is named PC$_{\text{SSL}}$ (**P**ermutation of **C**omponents in **S**emi-**S**upervised **L**earning), and it uses the principle of maximum likelihood to determine the label permutation π of the previously decomposed components. It not only coincides with the method for Stage 2 of Cover and Castelli for $K = 2$ ([1] and [2]), but also it is a natural extension of that method to the multi-class framework. Formally, the learning strategy is as follows:

$$\hat{\pi}_p = \arg\max_{\pi} L(\pi; \mathcal{L}) = \arg\max_{\pi} \prod_{i=1}^{K} \prod_{\mathbf{x} \in \mathcal{L}_i} \eta_i f_{\pi^{-1}(i)}(\mathbf{x}). \tag{8}$$

Briefly, PC$_{\text{SSL}}$ works as follows: it returns the correspondence π between the classes and the identified components with the highest likelihood function $L(\pi; \mathcal{L})$. The following theorem proves the optimality of our proposal:

Theorem 2. ***(Optimality of PC$_{\text{SSL}}$)*** *PC$_{\text{SSL}}$ is an optimum learning procedure for Stage 2 of Algorithm 2.*

Proof. Let $\pi^* = \arg\max_{\pi} P(\pi|\mathcal{L})$ be the BDR for classifying a labelled subset into one of the $K!$ different possible permutations. Since it is the optimal classifier, PC$_{\text{SSL}}$ can be proved to be optimum if both classifiers are equivalent, i.e. $\pi^* = \hat{\pi}_p, \forall \mathcal{L}$. To prove this statement, we reduce the BDR to PC$_{\text{SSL}}$ by rewriting the optimal rule as

$$\pi^* = \arg\max_{\pi} \Pr\{\pi, \mathcal{L}\} = \arg\max_{\pi} f(\mathcal{L}|\pi) P(\pi),$$

where the notation $\Pr\{\cdot\}$ is used to omit measure-theoretical details for the sake of clarity. Regarding $f(\mathcal{L}|\pi)$, as *max* measures of disjoint events are independent and fixing π, we reach the conclusion that it is equal to the likelihood:

$$f(\mathcal{L}|\pi) \quad = \prod_{i=1}^{K} \eta_i f(\mathcal{L}_i|\pi) = \prod_{i=1}^{K} \prod_{\mathbf{x} \in \mathcal{L}_i} \eta_i f(\mathbf{x}|\pi^{-1}(i)) = \prod_{i=1}^{K} \prod_{\mathbf{x} \in \mathcal{L}_i} \eta_i f_{\pi^{-1}(i)}(\mathbf{x}) = L(\pi; \mathcal{L})$$

Concerning the model priors $P(\pi)$, it is reasonable to assume them to be uniformly distributed (as in the key work of Cover and Castelli [1]), which holds when the components are indexed in a random uniform manner. Therefore,

$$\pi^* = \arg\max_{\pi} f(\mathcal{L}|\pi)P(\pi) = \arg\max_{\pi} L(\pi;\mathcal{L}) = \hat{\pi}_p$$

□

5.3 Computational complexities of both procedures

While the solution of eq. (7) (Voting) and eq. (8) (PC$_{\text{SSL}}$) for a given $\mathcal{L}$ can be straightforwardly obtained by an exhaustive search over the $K!$ factorial permutations, this process can be simplified, in terms of computational complexity, by rewriting both equations as linear assignment problems and, then, using the Hungarian [18] to find the solution. Therefore, we consider $[n^V_{ij} = |\mathcal{L}_i \cap R_j|]$ as the cost matrix for a Voting strategy where each element n^V_{ij} represents the number of labelled examples with class value i appearing in R_j and $N^P = [n^P_{ij} = \sum_{\mathbf{x}\in\mathcal{L}_i} \log f_{\pi^{-1}(i)}(\mathbf{x})]$ as the square matrix representing the cost matrix of PC$_{\text{SSL}}$, where each element n^P_{ij} is the log-likelihood of labelled examples with class value i regarding the component $f_{\pi^{-1}(i)}$. Then, by applying the Hungarian algorithm over these cost matrices, the optimal assignment[3] π of labels, given a cost matrix, is achieved in polynomial time ($O(K^3)$). This transformation also solves the original ambiguity of Voting; in [8], further details are not provided about how Voting deals with the ties.

6 Minimum number of labelled examples

SSL is usually applied in domains where labelled data are very expensive and/or difficult to obtain, but crucial (eq. (6)). For that reason, we think it is necessary to tackle the second issue of Section 4: the minimal number of labelled data needed in Stage 2 of Algorithm 2 to unambiguously determine a permutation. Under the proposed framework, this issue can be translated into the calculation of the minimum number of labelled data needed to uniquely determine one possible permutation π without leaving any possibility to chance. Note that, when there is ambiguity, e_B cannot be reached.

As stated, in binary problems, just one labelled example is enough. It can also be easily seen that, in general, $(K-1)$ labelled instances with different label values are needed to do so and the remaining component is determined by a process of elimination. However, we cannot ensure having $(K-1)$ different labels in a particular labelled set $\mathcal{L}$ due to the randomness of the data [19]. Hence, for multi-class problems, expectations must be taken. We need to calculate l_K, i.e. *the expected minimum number of instances needed to have a labelled set with $(K-1)$ different class values among them.*

6.1 The expected minimum number of labelled examples

First, we determine l_K under the assumption of having all class priors equiprobable. This calculation is given by:

Theorem 3. ***(Minimum number of labelled examples)*** *Let the family of mixtures $\mathcal{F}$ be linearly independent. Let $K \leq \infty$ be the number of classes and the number of mixture components. Let the class priors be equiprobable, i.e. $\forall i, \eta_i = \eta = 1/K$. Then, the expected minimum number of labelled instances needed to uniquely determine the class labels of the K components of a mixture is :*

[3] Do not confuse the optimal solution for a linear assignment problem with the optimal probability of error. Under this setting, PC$_{\text{SSL}}$ remains as an optimal algorithm and Voting as a sub-optimal algorithm.

$$l_K = \begin{cases} 1 & \textit{if}\, K = 2 \\ \sum_{j=2}^{K-1} (-1)^j \binom{K}{j} (j-1) \Big(\frac{K-j}{K}\Big)^{(K-2)} \Big(K - 1 + \frac{(K-j)}{j}\Big) & \textit{if}\, K > 2 \end{cases} \tag{9}$$

Proof. The proof for $K = 2$ can be found in [1]. For higher values of K, let L_K be a random variable representing the minimum number of instances needed to obtain examples of $(K-1)$ different classes. Assuming equiprobability, $P(L_K = l)$, for $l \geq (K-1)$, is a fraction whose numerator is the number of favourable cases and whose denominator is the number of all possible cases:

$$P(L_K = l) = \frac{P_K S_2(l-1, K-2)}{PR_{K,l}},$$

where P_K is the number of permutations of K elements, $S_2(\cdot,\cdot)$ is the Stirling number of second kind, and $PR_{K,l}$ is the number of l-permutations of K elements with repetition (formulae in [20]). Then, we calculate the expectation of the random variable L_K in l as $l_K = E[L_K] = \sum_{l=K-1}^{\infty} lP(L_K = l)$. After some algebra, we reach equation (9), for $K > 2$. □

Problem	K	l_k
16K ImageNet	$15,589$	$143,911$
22K ImageNet	$21,841$	$208,992$
21K WebData	$21,171$	$201,921$
97K WebData	$96,812$	1.07×10^6

Table 2: l_K for highly multi-class problems [21].

Figure 2a shows the growth of l_K for $K = \{2, \ldots, 80\}$ when the priors are equiprobable, i.e. $\eta_i = 1/K, \forall i$. It can be seen that it grows linearly in the number of classes K. This growth is due to the fact that this assumption among the priors is a hard constraint for the minimum labelled examples required. However, we want to remark the main benefit of calculating l_K for equiprobable priors; it is the lower expected bound of labelled examples needed for any possible configuration of K different class priors. For that reason, in practise, the study of l_K gains great importance for high values of K, such as in the recently proposed highly multi-class scenario [21], where $K > 1,000$. In [21], the authors deal with the problem of image classification in a supervised manner. However, since huge amounts of unlabelled images can be easily gathered, it is a matter of time to make use of unlabelled data in highly multi-class problems, as in [5] [22]. For such problems, l_K can be of vital importance for being a lower bound of the required labelled data. As an illustration, Table 2 presents, for each dataset used in [21], its correspondent l_K.

6.2 Relations between the class priors and l_K

Now, we relax the assumption of equiprobability. In the first place, we start calculating l_K for ternary problems:

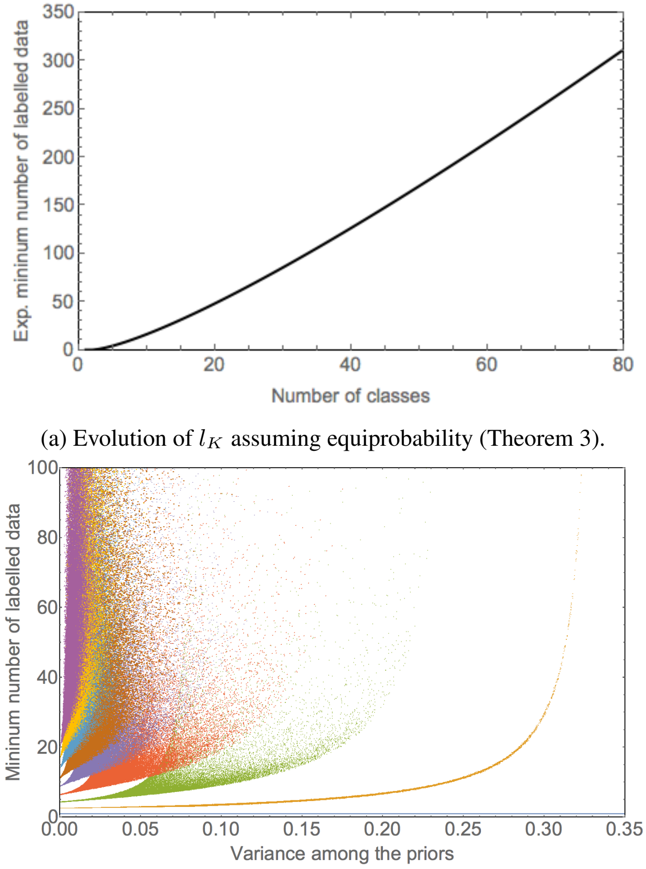
Theorem 4. ***(Minimum labelled examples for ternary problems)*** *Let $f(\mathbf{x})$ be identifiable and let $K = 3$ be the number of classes with priors $\eta_i > 0$, $\sum_{i=1}^{3} \eta_i = 1$. Then, the expected minimum number of labelled instances needed to uniquely determine the class labels of the components of the mixture is*

$$l_3 = 2 + \sum_{i=1}^{3} \frac{\eta_i^2}{1-\eta_i}. \tag{10}$$

Proof. Let L_3 be a random variable representing the minimum number of instances needed to obtain two different labels. Assuming each class c_i has a prior η_i and $\sum_{i=1}^{3} \eta_i = 1$, $P(L_3 = l)$ has the following form:

$$P(L_3 = l) = \eta_1^{(l-1)}(1-\eta_1) + \eta_2^{(l-1)}(1-\eta_2) + \eta_3^{(l-1)}(1-\eta_3).$$

Which stands for the probability of having $(l-1)$ labelled examples of one class, and just one example of one of the two other classes. Then, we calculate the expectation of L_3 in l. After some algebra, we reach the equation (10). □



(a) Evolution of l_K assuming equiprobability (Theorem 3).

(b) The growth of $l_2, \ldots, l_{10}$ (lower populations represents lower values of K) as $Var(\boldsymbol{\eta})$ increases (Monte Carlo method).

Figure 2: Minimal number of labelled data, l_K.

When we want to determine l_K for $K \geq 4$ with non-equiprobable class priors, it becomes intractable. Therefore, in order to avoid such a combinatorial explosion and to obtain an idea of the evolution of l_K with respect to the priors, we perform an empirical study to determine the growth of l_K when the class priors are non-equiprobable for the cases $K = \{2, \ldots, 10\}$; We generate a population of size $50,000$ independent samplings by a Dirichlet distribution with all its K hyper parameters set to 1. Since the Dirichlet distribution is the conjugate of the multinomial distribution [23], each sample of the population represents a vector of class prior probabilities $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_K)$ uniformly distributed along the domain of $\boldsymbol{\eta}$. Then, for each sample $\boldsymbol{\eta}$, we calculate l_K by means of a Monte Carlo method by averaging the minimum number of instances needed to have $(K-1)$ different classes over $10,000$ independent samplings. In order to be able to show the results in a two-dimensional figure, we also calculate the variance of each sample $\boldsymbol{\eta}$, i.e. $Var(\boldsymbol{\eta}) = \frac{1}{K}\sum_{i=1}^{K}(\eta_i - \bar{\eta})^2$, where $\bar{\eta} = \frac{1}{K}\sum_{i=1}^{K}\eta_i$. Note that the variance is highly correlated to the degree of imbalance among the priors. Then, Figure 2b shows the result of this simulation; how l_K grows as the variance is increased. There, it can be seen that the lowest value of l_K for every K always fits with the equiprobability ($Var(\boldsymbol{\eta}) = 0$), and from that point all the l_K values exponentially grow as the variance is increased. Only for $K = 2$, it remains constant. Here, it can be clearly noticed that the multi-class framework is much harder, at least in the number of labelled examples needed, than the binary scenario.

7 Probability of error in the multi-class framework

In this section, we deal with the last highlighted concern of Section 4; determining the probability of error $P_e(l, \infty)$ in the multi-class scenario under the correct model assumption.

In binary problems, the probability of error is calculated by exploiting the inherent characteristic of Algorithm 1: a classification error happens when either Stage 2 or 3 of Algorithm 1 yields an incorrect answer; but when both stages result in wrong answers, the mistakes cancel each other out [1], [2]. Unfortunately, this characteristic does not apply for the multi-class scenario. Here, the casuistry gets more complex; a classification error also occurs when either Stage 2 or Stage 3 of Algorithm 2 yields an incorrect answer. However, when both stages result in wrong answers, the mistakes do not necessarily cancel each other out. What's more, most of the mistakes in both stages lead to a final misclassification. Driven by these thoughts, we calculate the probability of error when a determined labelled subset $\mathcal{L}$ is given to derive the obtained results to the case when just the number of labelled records l is given. The following lemma formulates the probability of error for any learning procedure for Stage 2, including Voting and PC$_{\text{SSL}}$, when the BDR is applied over a returned permutation:

Lemma 1. *Let $\mathcal{L}$ be a labelled subset distributed according to a generative model $f(\mathbf{x}, c)$. Let the marginal of $f(\mathbf{x}, c)$ on $\mathbf{x}$ be an identifiable mixture density $f(\mathbf{x}) = \sum_{i=1}^{K}\eta_i f_i(\mathbf{x})$ which represents the distribution of the infinite unlabelled records. Let π be the correspondence returned by the learning procedure $\Pi(\cdot)$, (Stage 2 of Algorithm 2), and let $R_{\pi^{-1}(i)}$ be defined as in eq. (5). Then, the probability of committing an error in classifying an unseen instance with the BDR after assuming the correspondence π is*

$$P(e|\pi) = 1 - \sum_{i=1}^{K}\eta_i \int_{R_{\pi^{-1}(i)}} f_i(\mathbf{x})d\mathbf{x}. \tag{11}$$

Proof. According to Algorithm 2, a correct prediction occurs depending on whether the optimal classification rule, which assumes a learnt correspondence π, over the unseen instance $\mathbf{x}$, hits the real class value (c_i). If the region where $f_i(\mathbf{x})$ is maximum is R_j (it holds that $i = \pi_c(j)$), there are only two cases in which the real class is correctly predicted:

$$\begin{array}{llll} \text{Case 1: if } \mathbf{x} \in R_j & \wedge & \pi(j) & = i \\ \text{Case 2: if } \mathbf{x} \in R_{j' \neq j} & \wedge & \pi(j') & = i \end{array}$$

As can be seen, in both cases, the region to where $\mathbf{x}$ belongs can be named as $\pi^{-1}(i)$, which is equal to j in Case 1, and to j' in Case 2. Therefore, the probability of correctly classifying $\mathbf{x}$ is just the probability of $\mathbf{x}$ being in the region labelled as i, i.e. $R_{\pi^{-1}(i)}$ and this formula can be easily generalised to all the possible values of i in the range $\{1, \ldots, K\}$ as:

$$P(\bar{e}|\pi) = \sum_{i=1}^{K} \eta_i \int_{R_{\pi^{-1}(i)}} f_i(\mathbf{x}) d\mathbf{x},$$

After that we can reach formula (11) taking into account that the probability of error is the opposite to the probability of a correct classification, $P(\bar{e}|\pi) = 1 - P(e|\pi)$. □

Although the previous lemma formulates the probability of error of Stage 3 in Algorithm 2 independently of the learning procedure used for Stage 2, we are interesting in calculating the probability of error of the whole procedure for a given number of labelled data l:

Theorem 5. ***(Probability of error)*** *The probability of error of classifying an unseen instance in the multi-class scenario, given l labelled records and infinite unlabelled records, is*[4]*:*

$$P_e(l, \infty) = \sum_{\pi \in S_K} P(\Pi_l = \pi) P(e|\pi), \tag{12}$$

where $P(\Pi_l = \pi)$ denotes the probability of choosing, using the learning procedure $\Pi(\cdot)$, the permutation π given l labelled data (Stage 2 of Algorithm 2) and $P(e|\pi)$ the probability of misclassification with the BDR after assuming the correspondence π (Stage 3 of Algorithm 2). Finally, S_K represents the set of all possible permutations of size K representing all the correspondences between labels and components.

Proof. First, we define $\mathbb{L} = \{\mathcal{L} \mid |\mathcal{L}| = l\}$ as the set containing all the possible labelled subsets with cardinality l and formulate $P_e(l, \infty)$ as

$$P_e(l, \infty) = \int_{\mathbb{L}} P(e|\mathcal{L}) P(\mathcal{L}) d\mathcal{L}.$$

Unfortunately, the number of labelled sets with cardinality l is infinite (except for the case of $l = 0$). Therefore, we need to rewrite this equation by partitioning $\mathbb{L}$ into several disjoint sets $\mathbb{L}_\pi$, i.e. $\mathbb{L} = \bigcup_{\pi \in S_K} \mathbb{L}_\pi \wedge \forall a, b, \mathbb{L}_{\pi_a} \cap \mathbb{L}_{\pi_b} = \emptyset$. Each $\mathbb{L}_\pi$ stands for $\{\mathcal{L} \mid |\mathcal{L}| = l \wedge \Pi(\mathcal{L}) = \pi\}$. Then, by the distribution property, the probability can be rewritten as

$$P_e(l, \infty) = \sum_{\pi \in S_K} \int_{\mathbb{L}_\pi} P(e|\mathcal{L}) P(\mathcal{L}) d\mathcal{L},$$

In this case, the probability $P(e|\mathcal{L}), \forall \mathcal{L} \in \mathbb{L}_\pi$ will be equal to the $P(e|\pi)$ (eq. (11)) since the returned permutation is π. Due to the fact that $P(e|\pi)$ is constant, it can be extracted from the integrand as a common factor. Finally, as $P(\Pi_l = \pi)$ is, by definition $\int_{\mathbb{L}_\pi} P(\mathcal{L}) d\mathcal{L}$, we rewrite the formula as that presented in the theorem. □

[4] Note that eq. (12) fits with the one proposed for binary problems in [1] (pp. 107, eq. (7)): $P(\Pi_l = \pi_c) P(e|\pi_c) + P(\Pi_l = \bar{\pi}_c) P(e|\bar{\pi}_c)$.

However, we are interested in calculating, under certain assumptions, the convergence rate of the probability of error in the multi-class framework. When possible, we also calculate a formula depending on l and K. For that reason, in the following sections we calculate it for two scenarios; (i) when there are no pairwise intersections among the components and (ii) when the components intersect among themselves.

7.1 Mutually disjoint components

When the pairwise intersection among the components is empty, there is no chance of committing an error using the BDR in the supervised scenario, i.e. $e_B = 0$. However, in the SSL framework, a classification error may occur when the correspondence π is determined (Stage 2).

In order to calculate the error, we take advantage of the main property of this scenario; with just one labelled datum of the class c_i in the labelled subset, we can unequivocally determine $\pi_c(j)$. Therefore, the calculation of the error turns into the calculation of the probability that a certain number $z \leq \min(K, l)$ of labels appear in l labelled records. Under the assumption of having all priors equiprobable, the probability of error is calculated based on this reasoning as follows:

Theorem 6. ***(Probability of error with zero Bayes error)*** *Let the mixture density $f(\mathbf{x})$ be an identifiable mixture. Let K be the number of classes and the number of mixture components. Let the class priors be equiprobable, i.e. $\forall i, \eta_i = \eta = \frac{1}{K}$. Then, the probability of error $P_e(l, \infty)$, given $l > 0$ labelled records and infinite unlabelled records when the components are mutually disjoint is given by:*

$$P_e(l, \infty) = \sum_{z=1}^{\min(K-2,l)} \frac{P_{K,z} S_2(l, z)}{PR_{K,l}} \Big(1 - \frac{z+1}{K}\Big), \tag{13}$$

where $P_{K,z}$ and $PR_{K,l}$ are the number of z-permutations without repetition and l-permutations with repetition of K, respectively. $S_2(l, z)$ is the Stirling number of 2nd kind [20].

Proof. To determine the probability of error, we just need to calculate the probability of finding $z \leq \min(K, l)$ different labels in l. When $z \geq (K-1)$ (see definition of l_K) we reach the real model, so, the probability of error is e_B, which is zero. Therefore, we can define the probability as follows:

$$P_e(l, \infty) = \sum_{z=1}^{\min(K-2,l)} P(\Psi_l = z) P(e|z), \tag{14}$$

where Ψ_l is a random variable representing the number of different labels, z in l and $P(e|z)$ is the probability of committing a classification error knowing the real correspondence of z labels with their components.

Regarding $P(\Psi_l = z)$, as all the priors are equiprobable; it is a fraction whose numerator is the number of selecting just z classes of K multiplied by the way of ordering them, and whose denominator is the number of all possible cases:

$$P(\Psi_l = z) = \frac{P_{K,z} S_2(l, z)}{PR_{K,l}}.$$

Then, $P(e|z)$ can be decomposed into the sum of the probability of misclassifying when we find a particular label among the labelled subset and the probability of misclassifying it when we do not have that label to identify a mixture: $P(e|z) = \Pr\{i \in z\} \times \Pr\{e|i\} + \Pr\{i \notin z\} \times \Pr\{e|i\}$, where $\{i \in z\}$ corresponds to the event that the unseen instance has the same label as one of the labels that appears in the labelled subset,

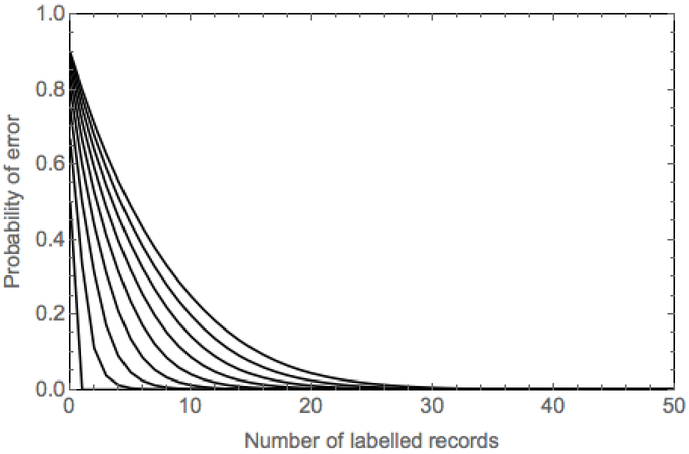


Figure 3: $P_e(l, \infty)$ for $K = \{2, \ldots, 10\}$ (lower lines are lower values of K) for mutually disjoint components ($e_B = 0$).

$\{i \notin z\}$ is the opposite, and Pr$\{e|i\}$ is the probability of misclassifying an instance of class i. By solving the previous formula, we obtain that $P(e|z)$ is equal to

$$\frac{z}{K} \times 0 + \frac{K-z}{K} \times \frac{(K-z)! - (K-z-1)!}{(K-z)!} = 1 - \frac{z+1}{K}. \tag{15}$$

Finally, by substituting the calculations of $P(\Psi_l = z)$ and $P(e|z)$ in equation (14), we reach formula (13). □

Corollary 1. *When the components are mutually disjoint, $P_e(l, \infty)$ converges to 0 exponentially fast in the sense of that*

$$o\left(\left(\frac{K-1}{K}\right)^l\right). \tag{16}$$

Proof. It is trivial to calculate the convergence order from the previous theorem. □

Figure 3 illustrates the variation of the error under the assumptions of models with equiprobable priors and $e_B = 0$ for K from 2 to 10. There, it can be seen that the probability of error converges exponentially fast to zero in l. Also, note that, in this scenario, both Voting and PC$_{\text{SSL}}$ are equivalent. Both of them obtain the optimal probability of error (eq. (13)).

7.2 Mutually non-disjoint components

When the $e_B > 0$, we provide an upper bound (Theorem 7) of $P_e(l, \infty)$ in order to determine the convergence rate in l of the optimal probability of error (Corollary 2). Here, we assume having all priors equiprobable, $\forall i, \eta_i = \eta = 1/K$.

Theorem 7. ***(Upper bound of the error)*** *When the components are not mutually disjoint and all the priors are equiprobable, the optimal probability of error of a model composed by K different identifiable mixture components is upper bounded by*

$$P_e(l,\infty) - e_B \leq 2\exp\Big\{\frac{-l\lambda^2}{2K}\Big\}, \tag{17}$$

where $\lambda \in (0,1]$ depends on the degree of intersection among the components of the mixture distribution and is defined by

$$\lambda = \frac{1}{K}\min_j \Big\{ \int_{R_j} f_i(\mathbf{x})d\mathbf{x} - \max_{z\neq i} \int_{R_j} f_z(\mathbf{x})d\mathbf{x}\Big\}. \tag{18}$$

Proof. Let the Voting learning procedure [8] be defined as the function $\Sigma : \mathbb{L} \to S_K$, where $\mathcal{L} \in \mathbb{L}$ is the labelled set and $\hat{\pi}_v \in S_K$ is the permutation of components returned by Voting. In [8], the *excess of risk* of the Voting procedure, $(\varepsilon(\Sigma))$, is defined as

$$E_{\mathbb{L}}\Big[\sum_{j=1}^{K}\int_{R_j}\big(\eta_{\pi_c(j)}f_{\pi_c(j)}(\mathbf{x}) - \eta_{\hat{\pi}_v(j)}f_{\hat{\pi}_v(j)}(\mathbf{x})\big)d\mathbf{x}\Big], \tag{19}$$

where $E_{\mathbb{L}}[\cdot]$ is the expectation with respect to the labelled sample and, $\pi_c(j)$ and $\hat{\pi}_v(j)$ correspond to the true class and the Voting bet of the j-th component, respectively. In that paper, they prove that $\varepsilon(\Sigma) \leq 2\exp\{-l\lambda^2/2K\}$.

Then, we just need to prove that $P_e(l,\infty) - e_B \leq \varepsilon(\Sigma_l)$. By means of the linearity property of both the integral and the expectation over equation (19), we reach

$$E_{\mathbb{L}}\Big[\sum_{j=1}^{K}\int_{R_j}\eta_{\pi_c(j)}f_{\pi_c(j)}(\mathbf{x})d\mathbf{x}\Big] - E_{\mathbb{L}}\Big[\sum_{j=1}^{K}\int_{R_j}\eta_{\hat{\pi}_v(j)}f_{\hat{\pi}_v(j)}(\mathbf{x})d\mathbf{x}\Big].$$

There, the first term is equal to $(1-e_B)$ since $\pi_c(j) = i$ (equation (11)). Then, the second is equal to $\int_{\mathbb{L}}(1-P(e|\mathcal{L}))P(\mathcal{L})d\mathcal{L}$ (where Voting is implicitly contained), which, following the same reasoning as in the proof of theorem 5, it is equal to

$$P_e^V(l,\infty) = \sum_{\pi\in S_K} P(\Sigma_l = \pi)P(e|\pi).$$

Note that, here, $P_e^V(l,\infty)$ is used instead of $P_e(l,\infty)$ to denote that the Voting classifier is used, not the optimal one. Then, by Theorem 2 ($P_e^V(l,\infty) \geq P_e(l,\infty), \forall l$), we can reach the conclusion of equation (17); $P_e(l,\infty)$ is upper bounded by $\varepsilon(\Sigma)$, which, in turn, is upper bounded by $2\exp\{-l\lambda^2/2K\}$. □

Corollary 2. *The probability of error, $P_e(l,\infty)$, decreases to e_B, at least, exponentially fast in the number of labelled data.*

Proof. It is trivial to calculate the convergence order from the previous theorem. □

The previous calculi prove that the optimal probability of error converges exponentially fast in l multiplied by a constant $\lambda \in (0,1]$. The latter only depends on the intrinsic characteristics of the components of the mixture; whilst models with mutually disjoint components show a value of $\lambda = 1$, models with a high level of overlapping show values of $\lambda \sim 0$. Note that, for values of λ close to 0, the decrease of the probability of error will be slower since, in problems with a high intersection of components, the process of discriminating the classes is intricate. In those cases, more labelled data will be required. However, is this upper bound good enough? Can the probability of error decrease faster? To answer these questions, we provide a lower bound of the optimal probability of error assuming the particular scenario of having a model composed by Gaussian mixture components with the same variance, i.e. $\boldsymbol{\theta} = \{\mu_1, \dots, \mu_K, \sigma\}$ and each $\boldsymbol{\theta}_i = \{\mu_i, \sigma\}$.

Lemma 2. *Assuming all the priors to be equiprobable and that the model is a Gaussian identifiable mixture of K components with the same variance (σ), let δ_M be the largest distance between the means of two components and let $\Phi(\cdot)$ be the CDF of a standard Gaussian distribution. Then, it holds that*

$$\min_{\pi} P(\Pi_l = \pi) \geq \Big(1 - \Phi\big(\frac{\delta_M}{2\sigma}\sqrt{l}\big)\Big)^{(K!-1)}. \tag{20}$$

Proof. First, $P(\Pi_l = \pi)$ can be rewritten as:

$$\sum_{(l_1,\dots,l_K)\in\mathcal{G}_K^l} P((l_1, \dots, l_K)) \times P(\Pi((l_1, \dots, l_K)) = \pi), \tag{21}$$

which is the decomposition of the probability in terms of the number of labelled examples of each class c_i in l, i.e. $l_i, \forall\, 1 \leq i \leq K$. There, $P((l_1, \dots, l_K))$ is the probability of having the distribution of labelled samples $(l_1, \dots, l_K)$ in l, $P(\Pi((l_1, \dots, l_k)) = \pi)$ is the probability of obtaining the permutation π with a determined distribution of labelled samples. $\mathcal{G}_K^l$ is the set containing all possible distributions of labels defined as the set containing all the integer partitions of l in exactly K addenda, but including zeros and taking into account the order of addenda. Formally, it is defined as

$$\mathcal{G}_K^l \triangleq \{(\gamma_1, \gamma_2, \dots, \gamma_K) | \sum_{z=1}^{K} \gamma_z = l \wedge \gamma_z \in \{0, 1, \dots, l\}, \forall z\}.$$

Then, we define $P(\Pi((l_1, \dots, l_k)) = \pi)$ in terms of the components of the mixture as

$$P\left(\frac{\prod_{i=1}^{K}\prod_{a=1}^{l_i} \eta f_{\pi^{-1}(i)}(x_a)}{\arg\max_{\tau\neq\pi}\{\prod_{i=1}^{K}\prod_{a=1}^{l_i} \eta f_{\tau^{-1}(i)}(x_a)\}} \geq 1\right),$$

where $f_j(x)$ is the density function of the component j. For simplicity of notation, from now on, we rename the numerator as f_π^l and the denominator as $\arg\max_{\tau\neq\pi}\{f_\tau^l\}$.

Then, by defining the permutation π_D as the furthest permutation to π_c, i.e. $\pi_D = \arg\max_\pi \sum_{i=1}^{K} |\mu_{\pi^{-1}(i)} - \mu_{\pi_c^{-1}(i)}|$, it holds that

$$\min_{\pi} P\Big(\frac{f_\pi^l}{\arg\max\limits_{\tau\neq\pi}\{f_\tau^l\}} \geq 1\Big) = P\Big(\frac{f_{\pi_D}^l}{\arg\max\limits_{\tau\neq\pi_D}\{f_\tau^l\}} \geq 1\Big). \tag{22}$$

As there is no independency between the permutations, we cannot express the second term of the formula (22) as a product of π_D being greater than or equal to any other permutation τ. For that reason, we use the chain rule to decompose the formula in such a way that the first term of the chain has, $\forall l > 0$, the lowest probability:

$$P\Big(\frac{f^l_{\pi_D}}{\arg\max_{\tau\neq\pi_D}\{f^l_\tau\}} \geq 1\Big) = P\big(f^l_{\pi_D} \geq f^l_{\pi_c}\big) \times \prod_{j=2}^{K!} P\Big(f^l_{\pi_D} \geq f^l_{\pi_j} | f^l_{\pi_D} \geq f^l_{\pi_c}, \bigcap_{m=2}^{j-1} f^l_{\pi_D} \geq f^l_{\pi_m}\Big). \tag{23}$$

Since we assume a mixture of Gaussian components ($\boldsymbol{\theta} = \{\mu_1, \dots, \mu_K, \sigma\}$), doing some calculations over the first term of the chain in formula (23) we reach the conclusion that it is equal to

$$\begin{aligned} P\big(f^l_{\pi_D} \geq f^l_{\pi_c}\big) \quad & = 1 - \Phi\left(\frac{1}{2\sigma}\sqrt{\sum_{i=1}^{K} l_i(\mu_{\pi_D^{-1}(i)} - \mu_{\pi_c^{-1}(i)})^2}\right) \\ \text{and bounded to} \quad & \\ & \geq 1 - \Phi\left(\frac{1}{2\sigma}\sqrt{\sum_{i=1}^{K} l_i\delta_M^2}\right) = 1 - \Phi\Big(\frac{\delta_M}{2\sigma}\sqrt{l}\Big)\ . \end{aligned} \tag{24}$$

where $\delta_M = \max\{(\mu_{\pi_D^{-1}(i)} - \mu_{\pi_c^{-1}(i)})^2\}$ is the largest distance between two means. As equation (24) is, by definition of π_D, the lowest term in the product of equation (23), we can lower bound it by substituting the product by equation (24) to the $(K!-1)$ (number of terms in the chain rule) power. Then, substituting this value in equation (21), we reach formula (20): $\min_\pi P(\Pi_l = \pi)$ is greater than or equal to

$$\Big(1 - \Phi\big(\frac{\delta_M}{2\sigma}\sqrt{l}\big)\Big)^{(K!-1)} \overbrace{\sum_{\mathcal{G}^l_K} P((l_1, \dots, l_k)}^{1}\,. \tag{25}$$

□

Theorem 8. ***(Lower bound of the error)*** *Assuming that the model is a Gaussian identifiable mixture of K components with the same variance (σ) and equiprobable priors. Let δ_M be the largest distance between the means of two components and $Q(\cdot)$ a polynomial of degree 3. Then, the probability of error for non-disjoint components is lower bounded by*

$$P_e(l, \infty) - e_B \geq \frac{K!(1 - e_B) - (K-1)!}{\big(1 + \exp\{Q(\frac{\delta_M}{2\sigma}\sqrt{l})\}\big)^{(K!-1)}}. \tag{26}$$

Proof. First, we decompose formula (10) of the main manuscript as follows:

$$P_e(l, \infty) = \quad P(\Pi_l = \pi_c)P(e|\pi_c) + \sum_{\pi \in S_K \backslash \pi_c} P(\Pi_l = \pi)P(e|\pi). \tag{27}$$

It can be easily seen that, when l increases, whilst $P(\Pi_l = \pi_c)$ grows to 1, the remaining $P(\Pi_l = \pi), \forall \pi \neq \pi_c$ decrease to 0.

Since $P(e|\pi_c)$ is, by definition, equal to e_B and $P(\Pi_l = \pi_c) = 1 - \sum_{\pi \in S_K \backslash \pi_c} P(\Pi_l = \pi)$, by just substituting equation (20) of the previous lemma in (27) and subtracting e_B in both terms, we can lower bound the error ($P_e(l, \infty) - e_B$) by

$$\Big(1 - \Phi\big(\frac{\delta_M}{2\sigma}\sqrt{l}\big)\Big)^{(K!-1)} (1 - e_B) \sum_{\pi \in S_K \backslash \pi_c} P(e|\pi). \tag{28}$$

There, we start by calculating $\sum_{\pi\in S_K\setminus\pi_c} P(e|\pi)$. Note that:

$$\sum_{\pi\in S_K\setminus\pi_c} P(e|\pi) = \sum_{\pi\in S_K} P(e|\pi) - e_B \tag{29}$$

$$\sum_{\pi\in S_K} P(e|\pi) = K! - \sum_{\pi\in S_K} P(\overline{e}|\pi) \tag{30}$$

By substituting formula (9) of Lemma 1 (in the manuscript) in equation (30), we obtain

$$\sum_{\pi\in S_K} P(e|\pi) = K! - \sum_{\pi\in S_K}\sum_{i=1}^{K}\frac{1}{K}\int\limits_{R_{\pi^{-1}(i)}} f_i(\mathbf{x})d\mathbf{x} \tag{31}$$

By distributive property of addition and the fact that $(K-1)!$ permutations of S_K share the same element in the same position, formula (31) can be rewritten as

$$\sum_{\pi\in S_K} P(e|\pi) = K! - (K-1)!\,\overbrace{\frac{1}{K}\sum_{i=1}^{K}\sum_{j=1}^{K}\int\limits_{R_j} f_i(\mathbf{x})d\mathbf{x}}^{1},$$

where $j = \pi^{-1}(i)$. Then, we can substitute the result in formula (29) obtaining

$$\sum_{\pi\in S_K\setminus\pi_c} P(e|\pi) = K! - (K-1)! - e_B. \tag{32}$$

Secondly, as there is no closed form expression for the normal cumulative density function, we approximate $\Phi(x)$ by an inverse exponential as proposed by Page in [24]:

$$\Phi(x) \sim \Phi^{\text{Page}}(x) = 1 - (1 + \exp\{Q(x)\})^{-1}, \tag{33}$$

where $Q(x) = 1.5976x + 0.070565992x^3$ and the absolute error $\epsilon = |\Phi(x) - \Phi^{\text{Page}}(x)| \leq 1.4\times 10^{-4}, \forall x \geq 0$. Then, by substituting the formulas (33) and (32) in (28) and after some algebra, we reach formula (26). □

It can be easily noticed that, in the case of assuming a Gaussian mixture, both bounds are quite close, leaving not much room for improvement in the upper bound; they converge exponentially fast in l to e_B. Note also that, in this scenario, $\delta_M \in (0,\infty)$ plays the role of the constant λ in the general solution; values of δ_M close to zero represent problems with a high intersection of components and a slower decrease of the probability of error. This can give us an idea that the optimal probability of error without any assumptions on the model will also converge exponentially fast, not faster. In the general case, it cannot decrease faster than this specific scenario.

8 Experimental studies

In the previous sections, we have proposed an optimal learning procedure, PC$_{\text{SSL}}$, for the multi-class problem in the SSL scenario. By Theorem 2, when the correct model assumption is met, any learning procedure, including the previously proposed Voting strategy [8], will achieve an upper or equal probability of error than PC$_{\text{SSL}}$. However, under the same assumption, , or do both algorithms share a similar probability of error? Moreover, another interesting question arises in this framework; how does PC$_{\text{SSL}}$ behave when the correct model assumption is not met?

To answer these questions a *generative model* with the following characteristics is assumed: since it must be simply enough to be able to fully interpret the results and complex enough to be able to represent real world problems, we assume a generative model composed by K univariate Gaussian identifiable mixture components with unit variances and whose means are separated by a fixed factor $\delta \in \mathbb{R}$, i.e. $\boldsymbol{\theta}_i = \{\mu_i, \sigma_i\}$, where $\mu_i = \delta(i-1)$ and $\sigma_i = 1$. This factor determines the degree of overlapping among the classes. Also, we assume equiprobable class priors[5]. Regarding the *learning procedures*, we make use of both PC$_{\text{SSL}}$ and Voting, as they are, to the best of our knowledge, the only two theoretical procedures proposed in the literature for this problem. The behaviour of both learning procedures is simulated by a Monte Carlo method; the probability of error of each procedure and each value of l is estimated by averaging the resulting probability of error over $10,000$ independent trials (labelled datasets)[6].

8.1 Does PC$_{\text{SSL}}$ significantly outperform Voting?

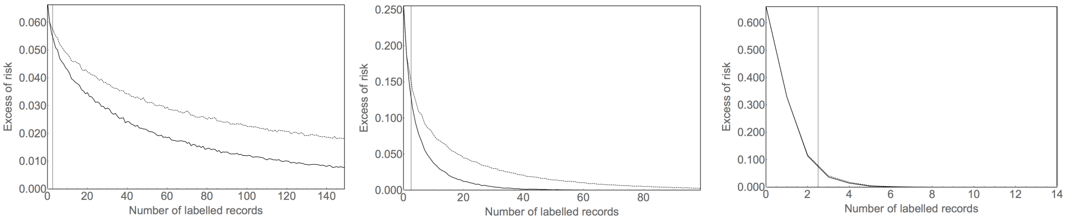
Then, the *first experiment* is carried out to address the first question, i.e. determining whether a significant difference between PC$_{\text{SSL}}$ and Voting exists. To do so, we have studied the behaviour of both procedures assuming the previous generative model for values[7] of $K = \{3, 4, 5, 6\}$. First, we exhaustively study the ternary case to, afterwards, check if the achieved conclusions can be extrapolated to higher values of K. Specifically, the probability of error of both PC$_{\text{SSL}}$ ($P_e^P(l,\infty) = P_e(l,\infty)$) and Voting ($P_e^V(l,\infty)$) learning algorithms has been calculated for $l = \{0, \ldots, l_{max}\}$ for three different levels of intersection among the components of the ternary problem, $\delta = \{0.25, 1, 5\}$. These three different δ values can be assumed to correspond to complex, medium and easy problems.

Figure 4 shows the behaviour of the probability of error for PC$_{\text{SSL}}$ and Voting as l increases in the studied ternary problem. Specifically, the different levels of intersection $\delta = \{0.25, 1, 5\}$ are represented by Figure 4a, Figure 4b, and Figure 4c, respectively. All the figures share the same shape. The x-axis represents the number of labelled data l and the y-axis represents the excess of risk of both procedures [2], i.e. $P_e^V(l,\infty) - e_B$ or $P_e^P(l,\infty) - e_B$. Note that, the x-axis is differently scaled for each problem due to the fact that complex problems require more labelled data (eq. (17)). Moreover, since e_B varies for different values of δ, the y-axis is also not equally scaled for the three scenarios. The corresponding Bayes error values for each $\delta = \{0.25, 1, 5\}$ are $e_B = \{0.6004, 0.4114, 0.0083\}$, respectively. Then, the upper decreasing curve (grey colour) is the excess of risk of Voting and the lower decreasing curve (black colour) corresponds to PC$_{\text{SSL}}$. The vertical line is l_K. As can be seen, the experiment coincides with the theoretical advances proposed in the paper: (i) the probability of error of both learning algorithms decreases exponentially fast in l (Theorem 7) and PC$_{\text{SSL}}$ always dominates Voting. It always achieves a lower (or equal) probability of error (Theorem 2). (ii) When $e_B \sim 0$, i.e. higher values of δ, both algorithms behave similarly in terms of probability of error (Theorem 6). (iii) In the opposite case, i.e. when δ is small, the room for improvement is quite narrow (e.g. $e_0 - e_B \sim 0.06$, for $\delta = 0.25$) and the complexity of the problem is really high. There, the probability of error of any algorithm will show a slower decrease in l (Theorem 7 and 8) and more labelled

[5] Note that if we consider unequal standard deviation, multivariate features, other geometry or non-normal probability densities, it may not be possible to perform all the calculations, e.g. the Bayes error.

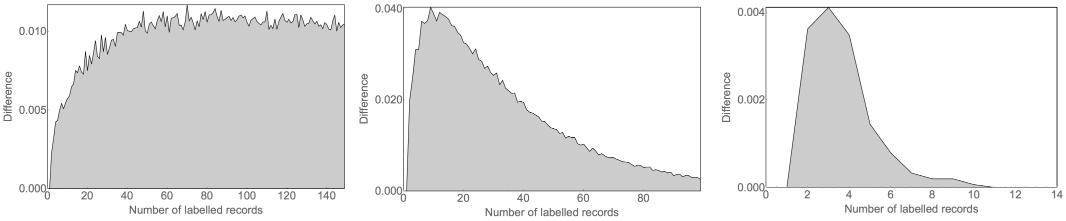
[6] For the sake of honesty, the same datasets are sampled for each procedure and each set of parameters. Moreover, the cases where the correspondence cannot unambiguously be determined are equally resolved for both procedures.

[7] Both learning algorithms are equivalent for binary problems.



(a) Complex problem ($\delta = 0.25$). (b) Medium problem ($\delta = 1$). (c) Easy problem ($\delta = 5$)

Figure 4: Probability of error of Voting [8] (upper) and PC_{SSL} (lower curve) for $K = 3$ ($l_K = 2.5$).



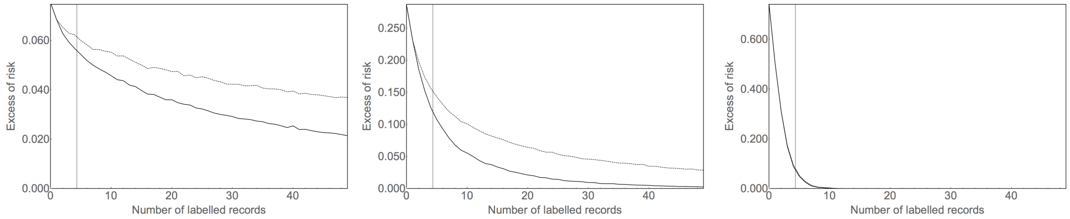
(a) Complex problem ($\delta = 0.25$). (b) Medium problem ($\delta = 1$). (c) Easy problem ($\delta = 5$)

Figure 5: Absolute differences between Voting and PC_{SSL}, i.e. $P_e^V(l, \infty) - P_e^P(l, \infty)$, for $K = 3$.

data will be required to achieve the best classifier. (iv) Finally, the results show that e_B is never achieved with less than l_K labelled examples (Theorem 3).

In order to properly quantify the magnitude of the absolute difference between the two theoretical SSL procedures, we also introduce Figure 5. There, the differences between the probabilities of error of both Voting and PC_{SSL} are shown for each l. Figures 5a, 5b, and 5c represent the previously defined complex, medium and easy problems, respectively. The y-axis in each figure is scaled between 0 and the highest difference found in the simulation. In general, the differences between the procedure show a similar shape throughout the problems. First, the difference between the probability of error of both theoretical procedures is 0 for both $l = 0$ (Theorem 1) and $l = 1$. After that, it grows until a determined value of l. Finally, beyond that point, the difference starts to decrease to 0, the point where Voting reaches e_B. Additionally, the results also reveal that, although the absolute differences vary for determined values of δ, the relative differences (w.r.t. the available room for improvement, i.e. $e_0 - e_B$) are greater for lower values of δ. Then, we can conclude that PC_{SSL} achieves a much better relative performance than Voting for low values of δ.

For higher values of K, i.e. $K = \{4, 5, 6\}$, we set $l_{max} = 50$. The presentation of the results follows the same style than the used for $K = 3$; whilst Figure 6, Figure 8 and Figure 10 shows the excess of risk of both Voting and PC_{SSL} for the values of $K = \{4, 5, 6\}$, respectively, Figure 7, Figure 9 and Figure 11 present the absolute differences between the probability of error of both learning procedures for the respective values of $K = \{4, 5, 6\}$. In each figure, the subfigure (a) stands for complex problems ($\delta = 0.25$), subfigure (b) for problems showing medium complexity ($\delta = 1$) and subfigure (c) shows the performance for easy problems showing a small degree of intersection among the components ($\delta = 5$). Regarding the obtained results, they are quite similar to ternary problems, i.e. the previous study can be straightforwardly extrapolated for higher values of K. The main difference is that, for higher multi-class problems, the behaviour of $K = 3$ is horizontally stretched when $l > 1$ due to the fact that both procedures need a higher value of l to reach e_B.

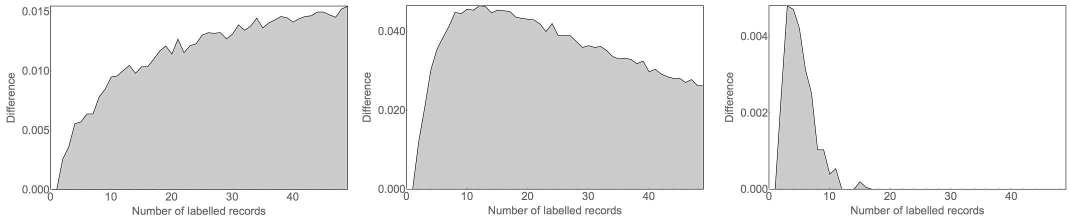


(a) Complex ($\delta = 0.25$).

(b) Medium ($\delta = 1$).

(c) Easy ($\delta = 5$)

Figure 6: Probability of error of Voting [8] (upper) and PC_{SSL} (lower curve) for $K = 4$ ($l_K = 4.33$).

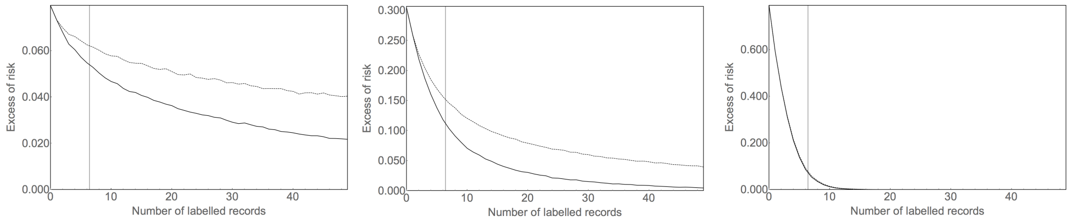


(a) Complex ($\delta = 0.25$).

(b) Medium ($\delta = 1$).

(c) Easy ($\delta = 5$)

Figure 7: Absolute differences between Voting [8] and PC_{SSL}, i.e. $P_e^V(l, \infty) - P_e^P(l, \infty)$, for $K = 4$.

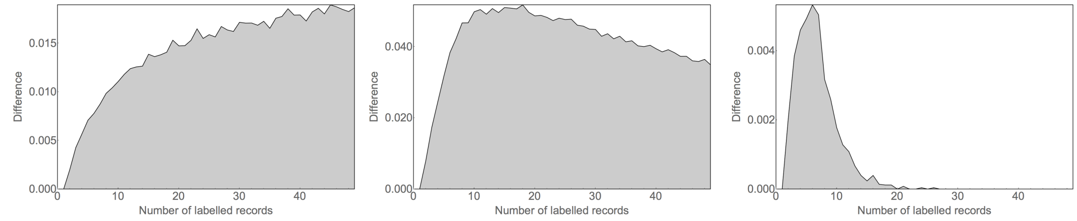


(a) Complex ($\delta = 0.25$).

(b) Medium ($\delta = 1$).

(c) Easy ($\delta = 5$)

Figure 8: Probability of error of Voting [8] (upper) and PC_{SSL} (lower curve) for $K = 5$ ($l_K = 6.42$).

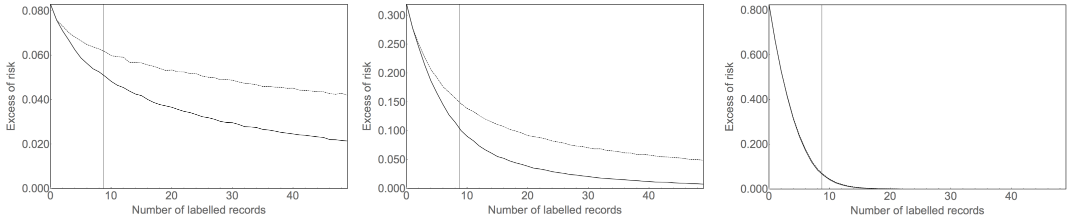


(a) Complex ($\delta = 0.25$).

(b) Medium ($\delta = 1$).

(c) Easy ($\delta = 5$)

Figure 9: Absolute differences between Voting [8] and PC$_{\text{SSL}}$, i.e. $P_e^V(l, \infty) - P_e^P(l, \infty)$, for $K = 5$.

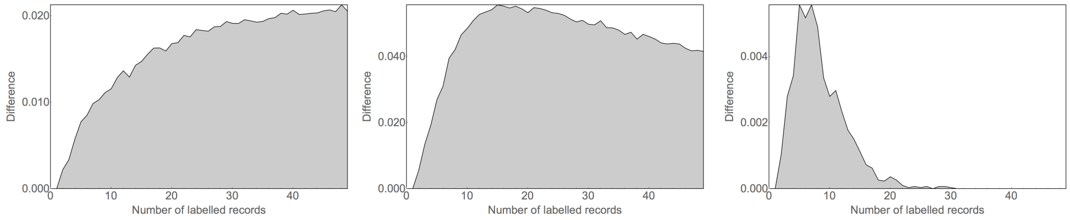


(a) Complex ($\delta = 0.25$).

(b) Medium ($\delta = 1$).

(c) Easy ($\delta = 5$)

Figure 10: Probability of error of Voting [8] (upper) and PC$_{\text{SSL}}$ (lower curve) for $K = 6$ ($l_K = 8.7$).



(a) Complex ($\delta = 0.25$).

(b) Medium ($\delta = 1$).

(c) Easy ($\delta = 5$)

Figure 11: Absolute differences between Voting [8] and PC$_{\text{SSL}}$, i.e. $P_e^V(l, \infty) - P_e^P(l, \infty)$, for $K = 6$.

8.2 How does PC$_{\text{SSL}}$ behave when the correct model assumption is not met?

The *second experiment* is devoted to studying the behaviour of PC$_{\text{SSL}}$ when the correct model assumption is not fulfilled, that is, when there is not enough unlabelled data to make a good estimation of the mixture density. To do so, we simulate that an incorrect model is obtained by a simple mechanism: the learnt model is also a K univariate Gaussian identifiable mixture components with unit variances and whose means are separated by a fixed factor δ. The class priors of this problem are also equiprobable. However, it is shifted a υ factor to the left. In this setting, υ varies in an arithmetic progression with a fixed difference of 0.25 from 0 to 5. Figure 12 sums up the behaviour of PC$_{\text{SSL}}$ for $K = 3$ and $\delta = 2.5$ (similar behaviours are

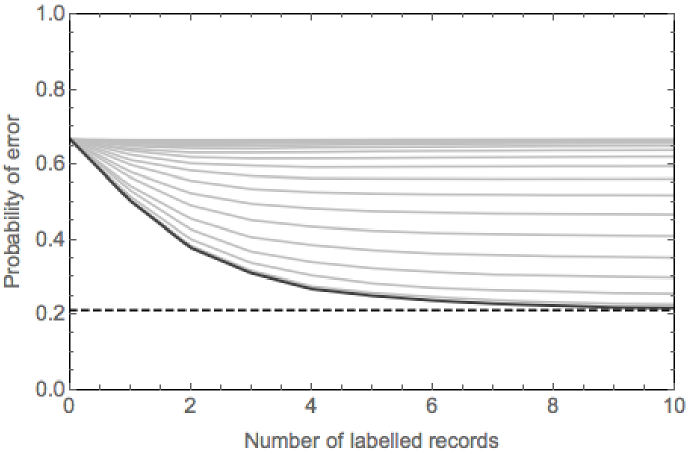


Figure 12: Evolution of the probability of error of $\mathrm{PC}_{\mathrm{SSL}}$ when the correct model assumption is not met.

found for different configurations). There, the black curve represents the correct model ($\upsilon = 0$), the grey curves correspond to different values of $\upsilon > 0$ (lower lines represent lower values of υ), and the horizontal dashed line is e_B. As can be seen, when this assumption does not hold, e_B can no be reached. When there is a slight difference among the models, reasonable performance can still be achieved, and the labelled data exponentially reduces the probability of error up to a difference ϵ between the asymptotic value of $\mathrm{PC}_{\mathrm{SSL}}$ and e_B [7]. This asymptotic value coincides with the unsupervised MLE, discussed in [6]. However, when υ grows, the reduction displays a more linear behaviour and the difference ϵ becomes higher. At the extreme, here $\upsilon = 5$, the probability of error, practically remains constant in l. This means that, in extreme cases of model misspecification, $P_e(l, u), \forall l, u \sim (K-1)/K$, i.e. the use of the labelled data does not reduce the probability of error [5] [6].

9 Remarks on the problem-solving in semi-supervised learning and open challenges

Although our main aim is to theoretically study the probability of error in the multi-class scenario, we want not only to discuss the potential impact of the theories presented in the designing of new practical learning algorithms, but also some challenges appearing in both theoretical and practical SSL scenarios. Thus, imagine we want to face a real-world problem using the theoretical advances presented in this paper. There, we basically face three different key questions:

9.1 How much unlabelled data should we gather?

Throughout the paper, we highlight the importance of the labelled records; if the correct assumption is met, they always help in making the labelled examples to reduce the probability of error faster than in supervised learning [5]. When $u = \infty$, the generative model can almost surely be recovered, therefore, the correct model assumption is met. However, in practical SSL, u is always finite. Moreover, there are also generative models which are not identifiable or, even, they do not follow mixture densities. For these practical cases, more general assumptions are used in order to support that the correct model can be learned. The assumptions are the following: smoothness assumption, cluster assumption and manifold assumption[8] [9]. However, these assumptions are hard to check in practise. They can only be tested by a trial-and-error procedure. If these assumptions are not met, the use of unlabelled data cannot guarantee any significant advantages over learning a purely supervised learning problem [25], so SSL techniques are not a good choice to solve the problem. In the opposite case, unlabelled data may help the performance of the classifier. There, we recommend the use of as much unlabelled data available so that a solid estimation of the generative model can be obtained by the learning algorithm and the correct model assumption can be met. Provided we have a good estimation of the model we can overtake, or even mitigate, the problems shown in the second experiment (incorrect model ass.). We emphasise that the problem of meeting the correct model assumption is still a challenging crucial issue. Another possible challenge for future work regarding the unlabelled data could be solved by [7], but assuming the correct model: Is there any number beyond which any extra additions of unlabelled data do not decrease the probability of error? In other words, which real number, in practise, corresponds to the infinite number of unlabelled records, broadly used in theoretical works.

9.2 How much labelled data is required?

In order to avoid making assumptions about the generative model, when the labelled data are neither expensive nor difficult to obtain, we strongly believe that supervised learning techniques are more appropriate. In the opposite case, if the correct model assumption is met, we have proved that, in general, e_B is never achieved for labelled sets with a cardinality lower than l_K as expressed in Theorem 3. This holds true independently of the degree of imbalance and the degree of intersection among the components. Therefore, a higher number of labelled records must be collected. However, when the degree of imbalance grows, more labelled are required for the same purpose (l_K for non-equiprobable priors). Analogously, for problems with a high intersection, the decrease of the probability of error is slower and, although l_K expects that e_B can be reached, a much higher number of labelled data is probably required to reach it. So, we can conclude that this question is still a challenging issue. For this reason, we think that it is interesting to, in the future, propose sample complexities for l, not only on the unbalanced degree of the priors, but also on the complexity and dimensionality of the feature space. Sample complexities seem to be crucial in SSL, where labelled data are scarce.

9.3 Which SSL learning procedure can be used?

In cases where the family of the generative model is known and the number of unlabelled examples is enough to obtain a good estimation, Algorithm 2 can be directly applied to the problem. There, both PC_{SSL} and Voting [8] can be used as learning procedures. However, PC_{SSL} seems to be a more appropriate choice, not only due to its theoretical properties, but also for matching the time complexity of Voting. On the contrary, when the family of the generative model is unknown, we cannot use the generative densities. In those cases, we can use the theoretical advances of this paper to design a practical algorithm for Stage 2. For problems with linear decision boundaries, such as the Gaussian classification of the experiments, a simple procedure for determining the components can be proposed based on the nearest-centroid classifier [26]. However, instead of classifying the data, the labelled samples can be used to determine the label of the centroids. Formally, by sphering each centroid and classifying it according to the class values of the labelled

[8] **Smoothness assumption:** Points which are close to each other are likely to share a label. **Cluster assumption:** The data tend to form discrete clusters, and points in the same cluster are more likely to share a label. **Manifold assumption:** The data lie approximately on a manifold of much lower dimension than the input space.

data in that sphered space. In a manner analogous to the theoretical methodology proposed in this paper, in this very case, we can also follow a Voting methodology by counting the majority class of labelled data in the sphered space or the PC_{SSL}, determining the minimum distance in the possible permutations. This can be another interesting potential future work; proposing practical learning procedures for both linear and non-linear decision boundaries. Finally, and within this framework, it could be also interesting to investigate a competitive, or even optimal, procedure to correctly specify the classifier, using labelled data, when the correct model assumption is not met, similarly to [7] and [8].

10 Summary

In this paper, we perform a study on the SSL multi-class framework, since most of the works deal with just binary problems [1] [2]. For that reason, we take it a step further by extending the work of Castelli and Cover [1] [2] to the multi-class paradigm. Particularly, we consider the key problem in SSL of classifying an unseen instance $\mathbf{x}^{(0)}$ into one of K different classes, using a training dataset composed of l labelled records and $u = \infty$ unlabelled examples. However, the previous studies do not straightforwardly work for multi-class problems, so, in this paper, we make three main contributions: (i) PC$_{\mathrm{SSL}}$, an optimal theoretical multi-class learning algorithm for SSL problems, is proposed. (ii) We investigate the expected minimum number, l_K, of labelled data needed to determine the K decision regions. (iii) We study the optimal probability of error when the binary constraint is relaxed, concluding that labelled data exponentially reduces the probability of error. A discussion on the impact of our proposals in solving real-world problems finalises the paper.

Acknowledgment

This work is partially supported by the Basque Government (IT609-13), the Spanish Ministry of Science and Innovation (TIN2013-41272P and MEC-FPU AP2008-00766), the Carlos III Health Institute (COMBIOMED), Diputación Foral de Gipuzkoa (2011-CIEN-000060-01), the European Commission (NICaiA PIRSESGA-2009-247619), and Diputación Foral de Bizkaia (6-12-TK-2011-014). We also thank Aritz Pérez, Ekhiñe Irurozki and Leticia Hernando for their comments.

Appendix - Source Code

The Mathematica package [27] containing the formulae presented in the manuscript is available to download from **`https://github.com/jonathanSS/SSLMultiClass`**. Besides, the package also contains some experiments used to study the behaviour of the probability of error of both Voting [8] and PC$_{\text{SSL}}$ learning algorithms.

l_k-related functions

1) The function `LKEQ[K]` calculates the l_K value for a problem with `K` equiprobable clases.

——`LKEQ[K]` ——

Input parameters:

`K` Number of classes.

Output:

A real number, l_k.

---

2) The function `LKEQPlot[maxK]` prints a figure of the l_K values for problems of $\{1..\texttt{maxK}\}$ equiprobable classes. The Figure 2a of the manuscript has been created with this function.

——`LKEQPlot[maxK]` ——

Input parameters:

`maxK` Number of classes.

Output:

Plot in the standard output.

---

3) `LKPriors[priors, nRep]` use a Monte Carlo method with `nRep` repetitions to approximate l_k for a problem with `Length[priors]` non-equiprobable priors given by the variable `priors`.

——`LKPriors[priors, nRep]` ——

Input parameters:

`priors` List containing K class priors.

`nRep` Number of repetitions.

Output:

A real number, l_k.

---

4) `LKSampling[K, samplesize, nRep]` applies `LKPriors[priors, nRep]` with `nRep` repetitions over a population of class priors of size `samplesize` which are generated from a Dirichlet distribution with all the alpha hyper-parameters equal to 1.

——`LKSampling[K, samplesize, nRep]`

Input parameters:

`K`	Number of classes.
`samplesize`	Population size.
`nRep`	Number of repetitions for `LKPriors[...]`.

Output:

A list of l_k, one per each sample of the population.

5) The eq. (8) of Theorem 4 corresponds to `L3[n1,n2,n3]`. It calculates l_3 for a ternary problem with priors `n1`, `n2` and `n3`.

——`L3[n1,n2,n3]`

Input parameters:

`n1`	Class prior of the class c_1
`n2`	Class prior of the class c_2
`n3`	Class prior of the class c_3

Output:

A real number, l_3.

6) `L3Plot[]` plots `L3[n1,n2,n3]` assuming that `n1`$= \eta_1$ and `n2, n3`$= (1 - \eta_1)/(K - 1)$.

——`L3Plot[]`

Input parameters:

−−

Output:

Plot in the standard output.

Functions related to the probability of error

7) The function `ZeroEB[K,maxL]` calculates the probability of error for $l = 0..$`maxL` for a `K`-class problem when there is no intersection among the components.

——`ZeroEB[K,maxL]`

Input parameters:

`K`	Number of classes.
`maxL`	Maximum number of labelled examples.

Output:

Summary of results in the standard output.

8) `MCPCSSL[K,distance,sigma,maxL,nRep]` uses a Monte Carlo method with `nRep` repetitions to approximate $P_e(l, \infty)$ for $l = \{0..\texttt{maxL}\}$ assuming a mixture of `K` Gaussian. `distance` represents the distance between the adjacent means and `sigma` is the variance.

——`MCPCSSL[K,distance,sigma,maxL,nRep]` ————————————

Input parameters:

`K`	Number of classes.
`distance`	Distance between the means of adjacent components.
`sigma`	Variance of the Gaussian components.
`maxL`	Maximum number of labelled examples.
`nRep`	Number of repetitions for the Monte Carlo method.

Output:

Summary of results in the standard output.

---

9) The function `MCPCSSLBiased[K,distance,sigma,maxL,bias, nRep]` uses a Monte Carlo method with `nRep` repetitions to approximate $P_e(l, \infty)$ for $l = \{0..\texttt{maxL}\}$ assuming a a generative model composed of a mixture of `K` Gaussian. `distance` represents the distance between the adjacent means and `sigma` is the variance. In this simulation the learnt model is also a mixture of Gaussian components, but they are shifted a factor `bias` to the right, i.e. $\hat{\mu}_i - \mu_i = \texttt{bias}$. This function corresponds to the second experiment of the manuscript.

——`MCPCSSLBiased[K,distance,sigma,maxL,bias, nRep]` ————————

Input parameters:

`K`	Number of classes.
`distance`	Distance between the means of adjacent components.
`sigma`	Variance of the Gaussian components.
`maxL`	Maximum number of labelled examples.
`bias`	Bias between the learnt and the generative models.
`nRep`	Number of repetitions for the Monte Carlo method.

Output:

Summary of results in the standard output.

---

10) `MCVOTING[K,distance,sigma,maxL,bias, nRep]` uses a Monte Carlo method with `nRep` repetitions to approximate the probability of error of Voting for $l = \{0..\texttt{maxL}\}$ assuming a mixture of `K` Gaussian. `distance` represents the distance between the adjacent means and `sigma` is the variance.

——`MCVOTING[K,distance,sigma,maxL,bias, nRep]`——

Input parameters:

`K`	Number of classes.
`distance`	Distance between the means of adjacent components.
`sigma`	Variance of the Gaussian components.
`maxL`	Maximum number of labelled examples.
`nRep`	Number of repetitions for the Monte Carlo method.

Output:

Summary of results in the standard output.

---

11) `MCComparison[K,distance,sigma,maxL,bias, nRep]` uses a Monte Carlo method with `nRep` repetitions to approximate the probability of error of both $\mathrm{PC}_{\mathrm{SSL}}$ and Voting for $l = \{0..\texttt{maxL}\}$ assuming a mixture of `K` Gaussian. `distance` represents the distance between the adjacent means and `sigma` is the variance.

——`MCComparison[K,distance,sigma,maxL,bias, nRep]`——

Input parameters:

`K`	Number of classes.
`distance`	Distance between the means of adjacent components.
`sigma`	Variance of the Gaussian components.
`maxL`	Maximum number of labelled examples.
`nRep`	Number of repetitions for the Monte Carlo method.

Output:

Summary of results in the standard output.

References

[1] V. Castelli and T. Cover, “On the exponential value of labeled samples,” *Pattern Recognition Letters*, vol. 16, pp. 105–111, 1995.

[2] ——, “The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter,” *IEEE Transactions on Information Theory*, vol. 46, no. 6, pp. 2102–2117, 1996.

[3] J. Ratsaby and S. Venkatesh, “Learning from a mixture of labeled and unlabeled examples with parametric side information,” in *Proceedings of the eighth annual conference on Computational learning theory (COLT ’95)*, 1995, pp. 412–417.

[4] T. Zhang and F. Oles, “A probability analysis on the value of unlabeled data for classification problems,” in *Proceedings of the International Conference on Machine Learning (ICML’2000)*, 2000, pp. 1191–1198.

[5] I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, and T. S. Huang, “Semisupervised learning of classifiers: Theory, algorithms and their application to human-computer interaction,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 12, pp. 1553–1567, 2004.

[6] T. Yang and C. Priebe, “The effect of model misspecification on semi-supervised classification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 2093–2103, 2011.

[7] K. Sinha and M. Belkin, “The value of labeled and unlabeled examples when the model is imperfect,” in *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. The MIT Press, 2008.

[8] H. Chen and L. Li, “Semisupervised multicategory classification with imperfect model,” *IEEE Transactions on Neural Networks*, vol. 20, no. 10, pp. 1594–1603, 2009.

[9] O. Chapelle, B. Scholkopf, and A. Zien, *Semi-supervised learning*. The MIT Press, 2006.

[10] R. Xu, G. Anagnostopoulos, and D. Whunsch II, “Multi-class cancer classification by semi-supervised ellipsoid artmap with gene expression data,” in *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, 2004.

[11] J. Ortigosa-Hernández, J. Rodríguez, L. Alzate, M. Lucania, I. Inza, and J. A. Lozano, “Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers,” *Neurocomputing*, vol. 92, pp. 98–115, 2012.

[12] G. M. Tallis and P. Chesson, “Identifiability of mixtures,” *Journal of the Australian Mathematical Society (Series A)*, no. 32, pp. 339–348, 1982.

[13] B. Everitt and D. Hand, *Finite Mixture Distributions*. Chapman and Hall, 1981.

[14] D. Titterington, A. Smith, and U. Makov, *Statistical Analysis of Finite Mixture Distributions*. John Wiley & Sons, 1985.

[15] B. Grun and F. Leisch, “Identifiability of finite mixtures of multinomial logit models with varying and fixed effects,” Department of Statistics, University of Munich, Tech. Rep. 024, 2008.

[16] I. Cohen, “Semisupervised learning of classifiers with application to human-computer interaction,” Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2003.

[17] P. Rigollet, “Generalization error bounds in semi-supervised classification under the cluster assumption,” *Journal of Machine Learning Research*, vol. 8, pp. 1369–1392, 2007.

[18] H. Kuhn, “The hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, no. 2, pp. 83–95, 1955.

[19] P. Fox-Roberts and R. E., “Unbiased generative semi-supervised learning,” *Journal of Machine Learning Research*, vol. 15, pp. 367–443, 2014.

[20] R. P. Stanley, *Enumerative combinatorics*. Wadsworth Publ. Co., 1986.

[21] M. Gupta, S. Bengio, and J. Weston, “Training highly multiclass classifiers,” *Journal of Machine Learning Research*, vol. 15, pp. 1461–1492, 2014.

[22] H. Wang, H. Huang, and C. Ding, “Image annotation using multi-label correlated greens function,” in *Proceedings of the IEEE 12th International Conference on Computer Vision*, 2009, pp. 2029–2034.

[23] J. M. Bernardo, M. H. DeGroot, D. Lindley, and A. F. M. Smith, *Bayesian Statistics*. Valencia University Press, 1980.

[24] E. Page, “Approximation to the cummulative normal function and its inverse for use on a pocket calculator,” *Applied Statistics*, vol. 26, pp. 75–76, 1977.

[25] S. Ben-David, T. Lu, and D. Pal, “Does unlabeled data provably help? Worst-case analysis of the sample complexity of semi-supervised learning,” in *Proceedings of the 21st Annual Conference on Learning Theory (COLT 2008)*, 2008.

[26] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference and prediction*. Springer, 2009.

[27] Wolfram Research Inc., “Mathematica,” 2014.