CLUSTER-BASED SEMI-SUPERVISED ENSEMBLE LEARNING

by

RODRIGO GABRIEL FERREIRA SOARES

A thesis submitted to
The University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Computer Science
College of Engineering and Physical Sciences
The University of Birmingham
January 2014

UNIVERSITY OF
BIRMINGHAM

University of Birmingham Research Archive

e-theses repository

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

To my grandparents.
And to my lovely Kelly.

Abstract

Semi-supervised classification consists of acquiring knowledge from both labelled and unlabelled data to classify test instances. The cluster assumption represents one of the potential relationships between true classes and data distribution that semi-supervised algorithms assume in order to use unlabelled data. Ensemble algorithms have been widely and successfully employed in both supervised and semi-supervised contexts. In this Thesis, we focus on the cluster assumption to study ensemble learning based on a new cluster regularisation technique for multi-class semi-supervised classification. Firstly, we introduce a multi-class cluster-based classifier, the Cluster-based Regularisation (ClusterReg) algorithm. ClusterReg employs a new regularisation mechanism based on posterior probabilities generated by a clustering algorithm in order to avoid generating decision boundaries that traverses high-density regions. Such a method possesses robustness to overlapping classes and to scarce labelled instances on uncertain and low-density regions, when data follows the cluster assumption. Secondly, we propose a robust multi-class boosting technique, Cluster-based Boosting (CBoost), which implements the proposed cluster regularisation for ensemble learning and uses ClusterReg as base learner. CBoost is able to overcome possible incorrect pseudo-labels and produces better generalisation than existing classifiers. And, finally, since there are often datasets with a large number of unlabelled instances, we propose the Efficient Cluster-based Boosting (ECB) for large multi-class datasets. ECB extends CBoost and has lower time and memory complexities than state-of-the-art algorithms. Such a method employs a sampling procedure to reduce the training set of base learners, an efficient clustering algorithm, and an approximation technique for nearest neighbours to avoid the computation of pairwise distance matrix. Hence, ECB enables semi-supervised classification for large-scale datasets.

Acknowledgements

Firstly, I would like to greatly thank my supervisor, Prof. Xin Yao, for his insightful comments, diligent guidance, generosity and kindness towards me. He has facilitated and encouraged this research tremendously. I'm extremely grateful to him for accepting me as his doctoral researcher. My career has immensely benefited from his encouragement, vast wisdom and discernment in conducting this research.

I would also like to give many thanks to my friend, Dr. Huanhuan Chen, for immensely helping to clarify various ideas in this research. He has always been hugely helpful to me. He has encouraged this work and provided many valuable advices to enhance this research and my professional life.

I'm very grateful to my thesis group members, Dr. Ata Kabán and Dr. John Bullinaria, for closely observing this work and offering insightful comments and advices.

I would like to thank the support from The CAPES Foundation, Ministry of Education of Brazil. My thesis research was also supported by the EU iSense grant (No. 270428).

Special thanks to Jakramate Bootkrajang who has been my Ph.D. student mate, office mate and house mate for three years. He has also helped with various discussions of ideas.

Many special thanks to Dr. Leandro Minku who has welcomed me in Birmingham. He has been one of the most helpful people that I have had the fortune to meet. Aside from these, I thank all my research student mates who have helped in all these years.

My very special thanks to my family, who has given me unconditional love, support and encouragement throughout my entire life. And my deepest gratitude to my lovely

future wife, Kelly. I hardly have words to express how amazing she has been in order to make this journey possible.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

This Chapter introduces the semi-supervised learning (SSL), the research questions and the contributions of this Thesis. Section 1.1 discusses the SSL problem. In Section 1.2.2, we describe the underlying assumptions in SSL algorithms. Section 1.4 discusses the usefulness of ensemble techniques in SSL. In Section 1.5, we highlight the research questions that we address in this Thesis. Section 1.6 summarizes the significant contributions achieved in this work. And, in Section 1.8, we conclude this Chapter with the outline of this Thesis.

1.1 Semi-supervised learning

Traditional machine learning techniques use only labelled instances (that is, the pairs of features and labels) to perform training. However, labelled data is usually expensive and time consuming to obtain. For example, a learning task might require expensive sensors and human experts to gather and label all the data.

On the other hand, it might be convenient to collect plenty of unlabelled data, which are typically cheap and abundant. Therefore, it is natural to employ such unlabelled data to improve predictive performance. SSL aims to use large amounts of unlabelled instances

along with labelled data to build better learning machines. As SSL requires less human effort and delivers potentially higher accuracy, it became popular in the machine learning community, in both theory and practice [Zhu, 2008].

In the SSL context, we have three main problems: semi-supervised clustering, semi-supervised regression and semi-supervised classification. The semi-supervised clustering refers to the problem of clustering data with some labelled points in the form of constraints: there should be some points that must belong to the same cluster and others that must not. The goal is to deliver a better partition than using the unlabelled data alone. Semi-supervised regression aims to learn a function. And semi-supervised classification (SSC) consists of predicting labels of test data. The algorithms of both regression and classification attempt to improve the generalisation accuracy in comparison with using only labelled data.

In this Thesis, we will focus on SSC, where algorithms learn from both labelled and unlabelled instances in order to assign a label (or posterior class probabilities in multi-class classification) to test instances.

1.2 Semi-supervised classification

Semi-supervised classification consists of using the available training instances to train a classifier that can be used to predict the classes of test data. These test instances are assumed to follow the same probability distribution as the available training data. The predictive accuracy of a trained classifier is evaluated by its generalisation from the training instances to test data.

Formally, we can define SSC as the choice from the given set of functions $f \in \mathcal{F} \rightarrow \mathbf{X} \times \mathbf{Y}$ based on a training set $\mathbf{D}$ of random independent identically distributed (i.i.d.) observations drawn from an unknown probability distribution $p(\mathbf{x}, \mathbf{y})$, such that the obtained function $\mathbf{f}(\mathbf{x})$ best predicts the true class for test instances $(\mathbf{x}, \mathbf{y})$, which are assumed to

follow the same probability distribution $p(\mathbf{x}, \mathbf{y})$ as the training set.

The training set $\mathbf{D} = \mathbf{L} \cup \mathbf{U}$ is composed of L labelled instances

$$\mathbf{L} = (\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_L, \mathbf{y}_L) \in \mathbf{X} \times \mathbf{Y},$$

and U unlabelled instances

$$\mathbf{U} = \mathbf{x}_1, \ldots, \mathbf{x}_U \in \mathbf{X}.$$

The total number of training instances is $N = L + U$ and often $U \gg L$. For multi-class classification, the labels $\mathbf{y}_n$ are class probabilities, where $0 \leq y_{ni} \leq 1$ and $\sum_{i=1}^{C} y_{ni} = 1$, where C is the number of classes.

Typically, SSC algorithms learn the optimal function $\mathbf{f}^*(\mathbf{x})$ by minimising a loss function $\mathcal{L}$,

$$\mathbf{f}^*(\mathbf{x}) = \text{argmin}_{\mathbf{f} \in \mathcal{F}} \left(\mathcal{L} \left(\mathbf{f}(\mathbf{x}), \mathbf{y} \right) \right),$$

which measures the loss or discrepancy between input and desired output associated with the learning machine, and then choosing the function from the given set of candidates with the lowest loss.

The aim of SSC is to improve a classifier in comparison to using the labelled data $\mathbf{L}$ alone. However, the effectiveness of SSC relies on the prerequisite that the data distribution, which unlabelled instances will help to elucidate, is related to the true class distribution. That is, the data distribution $p(\mathbf{x})$ should be useful for the inference of classes, $p(\mathbf{y}|\mathbf{x})$. If such a SSL fundamental assumption holds, a semi-supervised classifier will outperform a supervised method trained with only labelled data [Chapelle et al., 2006]. And, in recent years, various studies have shown that these two fundamental characteristics are true for a number of datasets [Chapelle et al., 2006, Chen and Wang, 2011, Valizadegan et al., 2008]. In order to link the knowledge acquired from the unlabelled data distribution to the class distribution, SSC algorithms employ one or more assumptions.

In Section 1.2.2, we discuss such assumptions.

As we will discuss later in this Thesis, the learning from unlabelled data is achieved by matching the assumptions in the classifier with the actual class structure. If this matching is not found, the use of unlabelled data may degrade the original supervised classification accuracy. So, although we have cheap and easily obtained unlabelled data, some decisions (for example, regarding design of good models, kernels, similarity functions, etc.) become more critical in SSC context than in supervised learning. This is the trade-off for the lack of label information in training data.

1.2.1 Transductive and inductive learning

SSC can be either transductive or inductive [Chapelle et al., 2006]. A classifier is transductive if test instances are employed as unlabelled training data. Such a method is not able to generalise predictions to unseen data and its generalisation is evaluated with its predictions on unlabelled data. The aim of transductive learners is to use the distribution of the available test data while predicting labels to such instances. Typically, classifiers that build graphs to represent the data are transductive (for example, the Spectral Graph Transducer (SGT) algorithm [Joachims, 2003]). Such methods use both test and training instances in the construction of graphs in order to predict labels for test points.

In contrast, inductive learners can generalise their predictions to unseen data. The training set of an inductive learner is composed of both labelled and unlabelled data and accuracy is measured on unseen data. As in supervised classification, inductive methods in SSC learn a decision boundary that will be used to generalise labels for unseen test instances. The classifier introduced in Valizadegan et al. [2008] is an example of an inductive learner.

1.2.2 Semi-supervised learning assumptions

Besides the fundamental assumption that states that the data distribution should be relevant for the classification problem, SSC algorithms also possess other assumptions in order to relate the distribution of instances, which the unlabelled data will help to clarify, to the inference of the class distribution.

In order to link the knowledge acquired from the unlabelled data to the true class distribution, SSC algorithms employ one or more of the following three assumptions [Chapelle et al., 2006].

- The *smoothness assumption* assumes that if two instances are close to each other, they are likely to yield similar outputs. Such an assumption is necessary for the generalisation from a finite set of training instances to a set of test points [Chapelle et al., 2006]. That is, if two instances x_1 and x_2 are similar, the output of the learner $f(x_1)$ and $f(x_2)$ should be also similar. This assumption is fundamental for any machine learning algorithm in order to generalise from one instance to others.

- The *cluster assumption* states that if two instances x_1 and x_2 are similar and lie on the same cluster (high-density region), they are likely to be of the same class, yielding similar outputs $f(x_1)$ and $f(x_2)$. That is, it is assumed that two different classes are unlikely to lie in the same high-density region. Equivalently, the cluster assumption can be formulated as the *low-density separation* assumption, which states that the decision boundary should lie in a low-density region [Chapelle et al., 2006]. Although both definitions are equivalent, they might lead to different algorithms.

- The *manifold assumption* assumes that the true structure of the data lies in a low-dimensional manifold embedded in the high dimensional data space. And, by using such manifold instead of the original structure, the classifier would have higher generalisation accuracy. That is, the learning of a transformation $\varphi(x_1)$ of instance

x_1 into a lower dimension manifold delivers more accurate generalisation than the use of the original instance x_1.

1.3 Discussion on supervised and semi-supervised classification

In order to facilitate the understanding on how semi-supervised classifiers can improve generalisation performance, we can analyse the Figure 1.1 (figure extracted from Zhu [2009]). Assuming each class is a coherent group (for example, we have two gaussian clusters), we can notice that the decision boundary can shift if we use the unlabelled data in the training of a classifier. The dashed line denotes the decision boundary generated with only labelled instances (intuitively, a line at the mean distance between the two labelled points would be the best generaliser for this dataset). While the solid line represents the shifted decision boundary, produced with both labelled and unlabelled data. In this case, when we consider unlabelled data, besides the influence of labelled instances, the data distribution (including labelled and unlabelled points) also affects the resulting decision boundary. This fact might produce stronger generalisation since the learning algorithm is using more information: the unlabelled data.

Another example of semi-supervised classification can be depicted from a generative model, such as Gaussian Mixture Models (GMM). Figure 1.2 (figure extracted from Zhu [2009]) shows the impact of unlabelled data on decision boundaries. The gaussians describing each class are adjusted with unlabelled data, due to the parameter optimisation procedure (usually the Expectation-Maximisation algorithm) maximising different quantities: the probabilities $p(\mathbf{L}, \mathbf{y}|\boldsymbol{\theta})$ and $p(\mathbf{L}, \mathbf{y}, \mathbf{U}|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the parameter vector of GMM.

Figure 1.3 (figure extracted from Zhu [2009]) shows the sensitiveness of SSC algorithms to SSL fundamental assumption: the data distribution should possess a relationship with

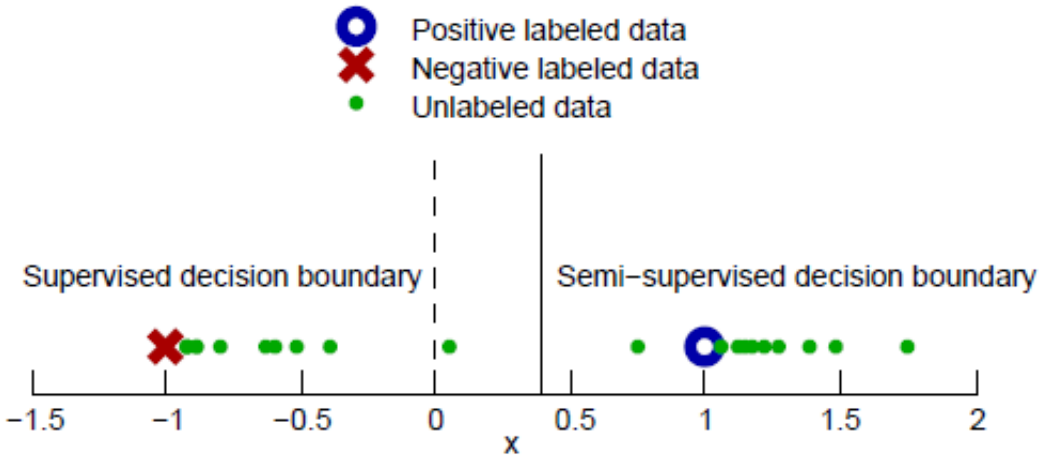


Figure 1.1: How SSC algorithms generate decision boundaries.

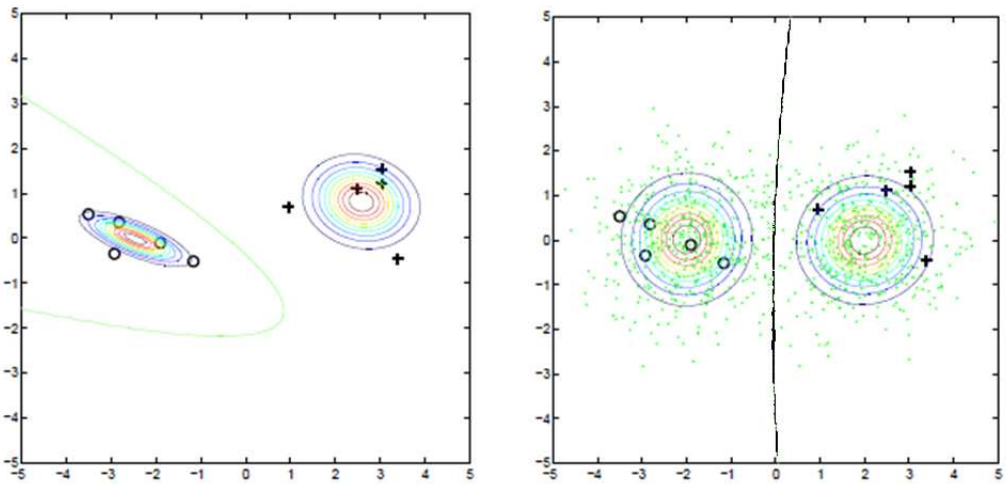


Figure 1.2: On the left-hand side, GMM considers only labelled data. On the right-hand side, both labelled and unlabelled data influences the decision boundary.

the true class distribution. Such a figure depicts the difference between the classical Support Vector Machine (SVM) [Vapnik, 1998] and one of its semi-supervised versions, Semi-Supervised Support Vector Machines (S3VM) [Bennett and Demiriz, 1998]. SVM attempts to find the largest gap between two groups of data. In such an example, the true decision boundary is the dotted line. On the left-hand side of the figure, the supervised decision boundary (depicted by the solid line) is generated exactly between the two

labelled instances. When considering unlabelled data, S3VM shifts the decision boundary (denoted by the dashed line), however they are both on a local minimum. On the right-hand side, the true decision boundary is not related with the data distribution. As expected, S3VM generated an incorrect decision boundary, as it finds the opposite diagonal of the figure. This is due to the fact that the fundamental SSL assumption does not hold in such a dataset. In this sense, in the attempt to find the largest gap considering the unlabelled instances, S3VM is not able to not learn the true decision boundary that lies outside of such a gap. Therefore, for the dataset in Figure 1.3, SSC algorithms might not improve generalisation when compared to supervised methods.

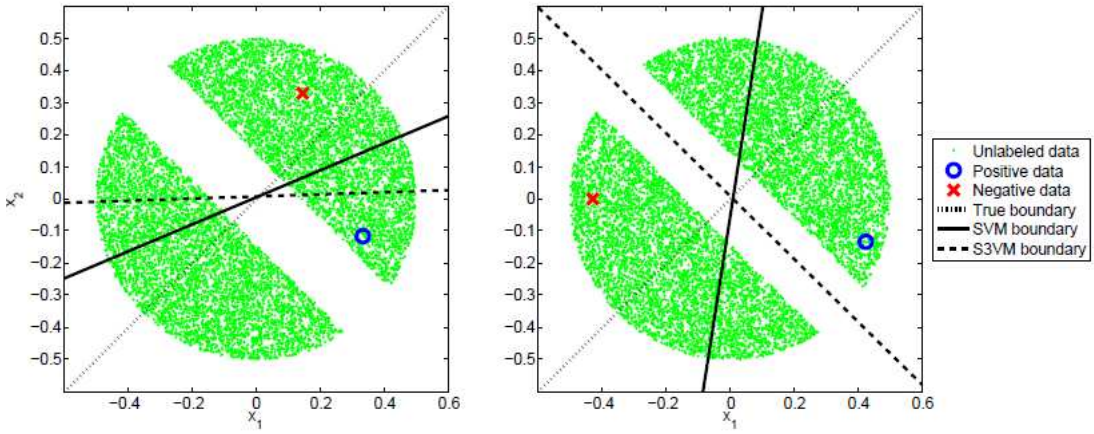


Figure 1.3: On the left-hand side, both labelled instances lie in the same high-density region. On the right-hand side, each labelled instance is a different cluster. In this dataset, true decision boundary does not correspond to the gap between high-density regions. The data distribution is not useful for the inference of the class distribution. In this case, SSC algorithms might not improve generalisation when compared to supervised methods.

As we can notice, a certain degree of relationship between SSL assumptions and data structure is essential to the performance of SSC algorithms. This is the compromise for the lack of label information. We showed examples where classifiers improve generalisation with unlabelled data (Figures 1.1 and 1.2) and a situation where the use of unlabelled instances is harmful to the performance of SSC algorithms (Figure 1.3).

It is important to highlight that the example on the left-hand side of Figure 1.3 represents a pathological case where we have only two labelled instances that are not representative of the classes. Although labelled points are usually scarce, they are important to identify regions of potential classes. Labelled instances should be i.i.d. samples from the underlying distribution of the dataset. Uninformative labelled points cause classes to be neglected. The absence or unfair distribution of labelled points for one class lead to predictions with bias either towards or against that particular class. If such instances are not representative of the class distribution, the decision boundary generated by a SSC algorithm might be degraded.

1.4 Cluster-based classification and ensemble learning

SSC is challenging due to the unknown relationship between the true decision boundary and the data distribution. In supervised learning, such a relationship might be irrelevant, as a decision boundary may be generated regardless the data distribution. In contrast, even if there is such a relationship in a given SSC dataset, we might not know which of the aforementioned SSL assumptions should be implemented in a classifier. However, when the implemented assumption is correct, the generalisation accuracy of a SSC algorithm is generally higher than supervised methods.

For real-world datasets, we might not have prior knowledge of the true class structure and its corresponding SSL assumption. Intuitively, if the dataset has a manifold structure, it is expected that algorithms that use the manifold assumption to deliver a better performance when compared to other SSC methods. This is also the case for datasets where the cluster assumption holds [Chapelle et al., 2006].

Various existing classifiers implement all three assumptions: smoothness, manifold and cluster assumptions. Such algorithms may yield better generalisation performance than

specialised methods (for example, classifiers that implement only the cluster assumption) with the incorrect assumption [Chen and Wang, 2011]. However, for example, in cases where there is a clear cluster structure for classes, the quality of the decision boundary generated by these algorithms might be limited by the search for a manifold in such datasets. Moreover, such methods also depend on the compromise between assumptions, which might be a challenging task. Therefore, in such situations, classifiers specialised in finding cluster structures, cluster-based classification, may yield significantly superior generalisation performance.

Ensemble techniques are widely employed in supervised classification due to the ability of reducing individual error made by base classifiers [Liu and Yao, 1999, Tang et al., 2009, Wang et al., 2009, Yao and Liu, 2004]. The combination of classifiers is also helpful in semi-supervised classification [Zhou, 2009]. The reasons for such a fact are: the performance of ensembles can be further improved, even though individual learners cannot be improved using unlabelled data; when there are very few labelled instances, unlabelled data should be exploited for constructing a strong ensemble; and unlabelled data can also increase the diversity of base learners [Zhou, 2009].

Therefore, combining a group of suitable classifiers, as an ensemble of classifiers, can improve the generalisation performance when compared with a single classifier in both supervised [Nguyen et al., 2006] and semi-supervised classification [Zhou, 2009, Valizadegan et al., 2008, Chen and Wang, 2011]. In this Thesis, we focus on investigating ensemble learning for cluster-based classification.

1.5 Motivations

In this section, we summarise the research questions that we address in this Thesis. Firstly, we highlight the limitations of existing cluster-based algorithms. Secondly, we highlight an important issue that arises when ensemble techniques are employed in SSC.

And, finally, we describe the limitations of applying existing classifiers to large datasets in SSC.

1.5.1 SSC with cluster regularisation

In cluster-based approaches, most algorithms attempt to find a low-density region (largest margin separator) to separate classes, avoiding generating a decision boundary inside clusters (traversing high-density regions). Transductive Support Vector Machine (TSVM) [Joachims, 2002] is a typical example of a margin-based classifier.

Most of the existing cluster-based SSC approaches might not produce a suitable decision boundary when classes are overlapping, due to the search for a largest margin separator with very few labelled instances. In such a case, the decision boundary should be generated in an overlapping region, however margin-based algorithms might not find such regions, especially when there is limited labelled data to constrain the decision boundary in the optimum region.

Such algorithms are sensitive to the position of the few labelled instances within a high-density region. If there are labelled points in an overlapping region (on the border of a cluster), margin-based methods might produce an incorrect decision boundary, since labelled instances have a greater impact on the learning than the unlabelled points. That is, decision boundaries generated by most of the cluster-based algorithms are mainly determined by the position of limited labelled instances. Therefore, such classifiers are not robust to few labelled data.

In contrast, some clustering algorithms can often easily achieve better performance with overlapping classes when compared to margin-based methods, as demonstrated in Chapter 3. Intuitively, in some cases where there is no clear gap between clusters, discovering high-density regions might be less challenging than finding low-density gaps between these regions. And clustering algorithms are specifically designed to search for

high-density regions. Therefore, in some cases, clustering algorithms can deliver more accurate estimates over the data distribution than methods that seek the largest margin. We discuss such issues in detail in Chapter 3.

In this sense, we can apply soft cluster structure, in the form of posterior probabilities, to regulate the impact of each labelled and unlabelled instance on the training of a classifier according to their position within a high-density region. Hence, such a classifier would be able to identify labelled instances in the borders of clusters and minimise their importance for the learning process, while increasing the influence of instances in centres of clusters. Therefore, the decision boundary produced by this classifier would also be a function of such a cluster structure. This learning technique would be more robust to the position of the few labels in the clusters and, hence, would improve its generalisation performance.

1.5.2 Ensemble learning in SSC

As in other SSC techniques, the performance of ensemble learning is strongly affected by the use of unlabelled data. Particularly for ensemble learning, an important question arises: at what level of an ensemble one should consider using unlabelled instances. To our knowledge, such an issue was not addressed in literature. Therefore, we present a study on the usefulness of employing unlabelled data at both ensemble and base learner levels, comparing to using such data at ensemble level only. In this sense, this Thesis will investigate solutions for such an issue.

In SSC literature, most ensemble methods optimise a semi-supervised loss function at ensemble level and use supervised base classifiers [Valizadegan et al., 2008, Zheng et al., 2009, Chen and Wang, 2011]. That is, the unlabelled data is only considered for the ensemble algorithm and the base classifiers receive such data as pseudo-labelled instances.[1] However, if the ensemble algorithm (for example, boosting framework [Friedman, 2001])

[1] In this work, pseudo-labels are posterior class probabilities artificially assigned to unlabelled instances by some method.

does not predict an instance correctly, relying on pseudo-labels for unlabelled instances might reinforce errors in the optimisation process, since supervised base classifiers will learn exactly the class that is assigned to a given instance.

In this context, the use of semi-supervised base learners might alleviate such an issue by handling the pseudo-labelled instances as actual unlabelled data. That is, the learning of pseudo-labelled data, as presented to the semi-supervised base learner, is dependent on the pseudo-label distribution, instead of learning each instance and its possibly erroneous pseudo-label individually. Therefore, the optimisation might not propagate a previous error caused by an incorrect pseudo-label. Whereas a supervised base classifier would learn exactly the labels that are presented to it. Thus, such a supervised base classifier would generate an incorrect decision boundary and would propagate the errors to the remainder of the ensemble training.

Semi-supervised base classifiers based on cluster assumption (as detailed in Chapter 3) address an incorrect pseudo-label according to its situation in the dataset. If an instance is in a high-density region, the pseudo-labels of its neighbours should be shared with such point. That is, the distribution of pseudo-labels would be employed to assess whether two instances should belong to the same class. Therefore, semi-supervised base learners might alleviate the problem of incorrect decision boundary (especially when it traverses high-density regions) by considering the distribution of unlabelled data, instead of using only the pseudo-label assigned to an instance. In this sense, we would have more reliable use of pseudo-labels than learning the exact pseudo-label of each instance independently.

1.5.3 Efficient algorithm for large datasets

Due to the large number of available unlabelled data, semi-supervised training sets often have tens of thousands of instances. Therefore, the learning algorithms must be able to handle such large-scale datasets. Recently, various ensemble algorithms have been

introduced with improved generalisation performance when compared to single classifiers. However, the existing ensemble algorithms are not able to handle large-scale SSC datasets.

The typical large number of unlabelled instances has a great impact on the efficiency of existing semi-supervised classifiers. Among methods that implements the cluster assumption (cluster-based algorithms), TSVM [Joachims, 2002] is a popular choice. However, this method requires time $\mathcal{O}(N^3)$ where N is the number of instances. Classifiers based on the manifold assumption (manifold-based algorithms) are also time consuming with computational complexity of $\mathcal{O}(N^3)$ or $\mathcal{O}(VN^2)$ where V is the number of neighbours [Szummer and Jaakkola, 2002, Zhu and Ghahramani, 2002]. For such binary classifiers, this issue is aggravated in multi-class classification, where decomposition approaches, such as *one-vs-all*, are employed, which require additional computational time.

As mentioned before, ensemble learning has been successfully employed in both supervised [Nguyen et al., 2006] and semi-supervised [Valizadegan et al., 2008, Chen and Wang, 2011] classification to improve generalisation when compared to single classifiers. However, the use of existing ensemble techniques in large-scale SSC datasets is limited due to time and memory requirements. For example, RegBoost [Chen and Wang, 2011] is a binary ensemble classifier that, if implemented with SVM,[1] requires time of $\mathcal{O}(VN \log N + TS^3 + TVU)$, where T is the number of base learners and S is the sample size. And, due to the computation of nearest neighbours, RegBoost requires memory of $\mathcal{O}(N^2)$, which also might prevent its application to large datasets.

A few multi-class ensemble approaches have been proposed [Valizadegan et al., 2008], however, despite having implemented the cluster assumption, these algorithms do not exploit the soft partition information, considering clusters as disjoint sets (hard clustering). And, likewise RegBoost, the classifier proposed by Valizadegan et al. [2008] also requires memory of $\mathcal{O}(N^2)$, which might degrade both time and memory efficiencies and might

[1] SVM is the base classifier recommended in Chen and Wang [2011].

limit its use in large datasets.

1.6 Contributions

We introduce relevant research questions in SSC: the use of soft partitions as regularisation for multi-class classification; the impact of unlabelled data in ensemble design; and ensemble techniques that enable SSC for large-scale datasets. We describe the contributions of this Thesis as follows.

1. **Algorithms for multi-class semi-supervised classification**.

   Most existing SSC methods in literature are binary classifiers, therefore such algorithms depend on suboptimal decomposition techniques, such as *one-vs-all* and *one-vs-one*, to perform multi-class classification. And they are prone to issues with imbalanced classes and different output scales of binary classifiers [Valizadegan et al., 2008]. Thus, in this Thesis, we focus on developing cluster-based classifiers that can perform multi-class classification effectively and efficiently.

2. **Semi-supervised classification with cluster regularisation**.

   In order to design an ensemble technique to solve the issues mentioned in Section 1.5, we design a new multi-class semi-supervised single classifier that overcomes the limitations of existing methods highlighted in Section 1.5.1.

   As mentioned before, most cluster-based methods attempt to find the largest margin between high-density regions (clusters). When overlapping high-density regions are present, with sparse labelled instances on their borders, these classifiers may not produce good predictions, although these inherent clusters might be easily identified.

   In this Thesis, we propose a cluster-based multi-class algorithm, ClusterReg (Cluster-based Regularisation). Unlike other cluster-based algorithms, it does not depend on gaps between potential clusters, but captures the partition information, as posterior

probabilities, from a clustering algorithm in order to improve the decision boundary.

By employing soft partitions in our method, the generalisation becomes more robust to the position of the few labels in the clusters. Unlike other cluster-based methods, our algorithm can easily establish a decision boundary between clusters without being misguided by neither the overlapping classes nor the few labels, when the dataset possesses a cluster structure for classes. ClusterReg achieves such robustness by incorporating soft clustering into a new regularisation technique.

ClusterReg regards the structure arising from the clustering algorithm as a soft partition. That is, each instance is assigned a probability of belonging to a given cluster, unlike hard partition where clusters are strictly disjoint. By using soft partitions (also known as soft clustering), we can address uncertain instances (likely lying on low density region, that is, on the border of clusters) differently from the more confident ones (likely lying on denser regions of clusters). Soft clustering helps the algorithm to address uncertain instances (with low probabilities for all clusters) as instances lying in gaps, therefore helping the classifier to generate the decision boundary in low-density regions.

One contribution of this work is to introduce a classifier that employs any clustering algorithm into SSC to regularise its decision boundary. The proposed algorithm (i) is robust in the presence of fewer labelled points, and is robust to the position of labelled data in clusters by considering the strength of clustering algorithms in a natural way and (ii) is able to improve the performance of a given classifier when the classes or clusters overlap, compared to other cluster-based algorithms.

ClusterReg can use any clustering algorithm with a proper processing of its output. It can employ any classifier that is able to minimise the proposed loss function. Therefore, ClusterReg can be seen as a framework for SSC methods.

3. **A fully semi-supervised ensemble for multi-class classification**.

   In Section 1.5.2, we raised the question of which level of an ensemble should use the available unlabelled instances. Such unlabelled data can be considered at the ensemble level, the base classifier level, or both. To our knowledge, such issue has not been addressed until now, as most algorithms use unlabelled data only at the ensemble level and employ supervised base classifiers. In this Thesis, we present a study on the usefulness of employing unlabelled data at both ensemble and base learner levels. We compare such an approach to the use of unlabelled instances at the ensemble level only. We propose the Cluster-based Boosting algorithm (CBoost) for multi-class classification. Such a method extends ClusterReg and, unlike most semi-supervised ensembles in the literature, is composed of semi-supervised base classifiers.

   Unlike other ensemble classifiers where unlabelled data is presented to the base classifiers as pseudo-labelled instances, both ensemble algorithm and base classifiers optimise the loss function introduced in Chapter 3, which uses both clustering algorithm and unlabelled data in a regularisation mechanism.

   CBoost is able to learn from the clustering neighbourhood structure of pseudo-labels assigned by the ensemble, which leads to better generalisation when compared to learning the exact pseudo-labels individually. CBoost can overcome incorrect pseudo-label assignments used in the training of a new base classifier. CBoost is robust to the position of labelled data within a cluster and is able to handle the potential presence of overlapping classes. Experiments in Chapter 4 confirmed that the proposed method is significantly superior to state-of-the-art ensemble methods and can improve the generalisation of single classifiers.

4. **Efficient boosting for semi-supervised classification**.

As discussed in Section 1.5.3, due to the large number of available unlabelled data, a semi-supervised training set can often have tens of thousands of instances. Therefore, the learning algorithms must be able to handle such large-scale datasets. Recently, various ensemble algorithms have been introduced with improved generalisation performance when compared to single classifiers. However, the existing ensemble algorithms are not able to handle typical large scale datasets.

In order to perform classification in large datasets, we propose the Efficient Cluster-based Boosting (ECB) algorithm. ECB uses a regularisation technique, based on posterior cluster probabilities, to avoid generating a decision boundary in high-density regions. In order to reduce the computational complexity of ECB, (i) base learners are trained with a subset of the unlabelled data along with all available labelled instances at each iteration; (ii) we also employ an approximation technique to increase the efficiency of time and memory in the computation of nearest neighbours; (iii) and use an efficient clustering algorithm. We provide a theoretical discussion on the reasons why ECB might be able to achieve good performance with small amounts of sampled data and a relatively small number of base learners. Our experiments confirmed that ECB scales to large datasets whilst delivering comparable generalisation to state-of-the-art methods.

1.7 Publications resulting from the thesis

The research outcome from Chapter 3 was reported in the following publication.

[Soares et al., 2012] R. G. F. Soares, H. Chen, and X. Yao. Semisupervised classification with cluster regularization. *IEEE Transactions on Neural Networks and Learning Systems*, 23(11): 1779–1792, November 2012.

The study in Chapter 4 was reported in

R. G. F. Soares, H. Chen, and X. Yao. A fully semi-supervised ensemble approach to multi-class classification. Submitted to *IEEE Transactions on Neural Networks and Learning Systems*.

And the investigation in Chapter 5 was reported in

R. G. F. Soares, H. Chen, and X. Yao. Efficient boosting for semi-supervised classification. Submitted to *The Journal of Machine Learning Research*.

1.8 Outline of the Thesis

This chapter presented the concepts that will be used throughout this Thesis: Semi-supervised classification and its assumptions, cluster-based classification and ensemble learning. We also summarised the contributions of this Thesis. The remainder of this Thesis is as follows.

Chapter 2 discusses several relevant algorithms in the literature of SSC. We categorise such methods according to implemented assumptions and internal mechanisms. And we discuss the limitations of existing classifiers, which are related to the research questions introduced in Section 1.5.

In Chapter 3, we address the motivations discussed in Section 1.5.1 and introduce a new algorithm with regularisation based on soft partitions. Such a method performs predictions according to the cluster structure along with scarce labelled data. Such a chapter also presents an instantiation of the proposed algorithm and provides an experimental analysis on the impact of such a regularisation technique on both transductive and inductive settings. Such an analysis confirms the improvement in generalisation ability over state-of-the-art methods.

Chapter 4 presents an analysis on the research question raised in Section 1.5.2. It introduces a study on the usefulness of employing unlabelled data at both ensemble and base learner levels, in comparison with using such data at the ensemble level only. We propose an ensemble technique that extends the classifier introduced in Chapter 3. And,

unlike other semi-supervised ensembles in the literature, it is composed of semi-supervised base classifiers. We also present experiments that confirm that the proposed method is significantly superior to state-of-the-art ensembles and can improve the generalisation of single classifiers.

In Chapter 5, we propose a multi-class boosting algorithm designed for large-scale datasets, such a method uses a regularisation technique, based on posterior cluster probabilities. We theoretically discuss how ECB is able to achieve good performance with small amounts of sampled data and a relatively small number of base learners. Our experiments confirmed that ECB scales to large datasets whilst delivering comparable generalisation to state-of-the-art methods.

Finally, Chapter 6 summarises this Thesis and describes potential subjects for future research.

CHAPTER 2

Semi-supervised Algorithms

In this Chapter, we report relevant SSC algorithms according to underlying assumptions and internal mechanisms. Section 2.1 describes generative algorithms, sections 2.2 and 2.3 present the self-training and Co-training frameworks. In Section 2.4, we describe classifiers with the manifold assumption. Section 2.5 discusses cluster-based algorithms and highlights potential drawbacks in their mechanisms. In Section 2.6, we report ensemble techniques that possess multiple assumptions. And, in Section 2.7, we discuss the limitations of the algorithms described in this Chapter and we relate such shortcomings to our motivations, as shown in Section 1.5.

2.1 Generative methods

Generative models are one of the earliest semi-supervised methods [Zanda and Brown, 2009]. Such algorithms attempt to estimate the conditional density $P(\mathbf{x}|\mathbf{y})$ by making explicit assumptions on the form of the data: the conditional distributions $P(\mathbf{x}|\mathbf{y})$ and class priors $P(\mathbf{y})$, where $\mathbf{x}$ and $\mathbf{y}$ are the input and target variables, respectively. For example, Gaussian Mixture Models (GMM) assume that $P(\mathbf{x}|\mathbf{y})$ is a Gaussian and use the Expectation-Maximisation (EM) algorithm to find the parameters that describe each

class [Dempster et al., 1977, Zhu, 2008]. GMM implements the cluster assumption, since a given cluster belongs to only one class [Chapelle et al., 2006].

Since generative methods learn how the data is distributed, $P(\mathbf{x})$, instead of the association between attributes and classes, such algorithms can naturally incorporate unlabelled data. However, these classifiers do not possess the knowledge about the true data distribution beforehand, such as the class of functions to describe the data. Thus, generative methods have to use strong assumptions on the data distribution. If the mixture model assumption is correct, the use of unlabelled data is guaranteed to improve the supervised classification accuracy. Otherwise, considering unlabelled instances may decrease classification accuracy [Castelli and Cover, 1995, 1996, Ratsaby et al., 1995].

Another approach is the cluster-and-label technique [Demiriz et al., 1999, Dara et al., 2002], which employs a clustering algorithm for finding structures in the data. It aims to match the generated partition with the real structure of the data. With the partition at hand, such an algorithm labels each cluster according to the labelled instances that belong to the given cluster. This method fails in the cases where the clustering does not represent the underlying structure of the data.

The EM algorithm was also employed in the prediction of fault-prone modules in the software engineering context [Seliya et al., 2004, Seliya and Khoshgoftaar, 2007b]. The authors could successfully employ a generative technique to find the data structure and classify software modules with limited fault-prone data (imbalanced datasets). And Seliya and Khoshgoftaar [2007a] formulated the software quality estimation problem as a semi-supervised clustering task.

2.2 Self-training

Self-training is commonly used in semi-supervised tasks [Zhu et al., 2009]. In such an algorithm, a given supervised learner is trained with only labelled instances. Afterwards,

such a learner predicts labels for a set of unlabelled instances. Then, the most confident instances along with the predicted label (pseudo-labels) are added to the labelled data to form a new training set, which will be used to retrain the given classifier. This procedure is repeated until a stopping criterion is met. Self-training is a wrapper method, in which any supervised learning algorithm can be used. Mixture models using the EM algorithm can be seen as self-training procedures.

If an error occurs in the classification, such a self-training process might reinforce the error, leading to low classification accuracy (its performance might be lower than the accuracy delivered by training with labelled data alone).

Moreover, since self-training selects only the most confident instances to be included in the new training set, only the easily classified instances are considered in the learning process. In this sense, the algorithm might only learn the instances that have already been learnt, without any improvement on the decision boundary when compared to using only labelled data.

2.3 Co-training

In the Co-training algorithm [Blum and Mitchell, 1998], the features in the training set are divided into two different sets (views). Such a division can be naturally achieved if the data intrinsically have two possible feature sets, or by applying some artificial method to separate attributes, such as random selection. The algorithm also assumes that both feature subsets are, at some extent, meaningful for the training of a classifier and such subsets are conditionally independent with respect to the class. Co-training consists of two classifiers, where each learner is assigned to a view. Initially, each method is trained with the labelled data from its respective view. Then, each classifier labels the unlabelled data of its own view (pseudo-labelling) and adds the most confident instances along with their predicted labels to the training set of its counterpart. Afterwards, both learners are

retrained with the newly labelled instances. In fact, both classifiers teach each other, and tend to agree on labelled and unlabelled instances.

The assumption on the quality of the views is essential to the generalisation of both classifiers. And the assumption of conditional independence between the views is necessary for mapping the most confident instances of one classifier into training points of its counterpart [Zhu, 2008].

Nigam [2001] demonstrated that co-training delivers good performance when the assumption of conditional independence holds. However, co-training possesses strong assumptions on the division of attributes. In order to relax this assumption, Zhou and Li [2005] proposed the Tri-training algorithm, which uses three learners. In order to train one classifier, such a framework employs the agreement between the remainder two learners to label unlabelled instances that will be used in the training set of the given learner.

Tri-training and Co-training may overfit with the use of the most confident instances, leading to the degradation of classification accuracy. This fact arises when the division of the feature set is not straightforward. That is, both methods depend on the quality of the subsets of attributes.

More generally, multiview learning [de Sa, 1993] represents the paradigm in which co-training and tri-training are included. This paradigm is based on the agreement among various learners that use distinct views. In multiview learning, several different classifiers (with different learning mechanisms) are trained with only labelled data and are required to agree on predictions of unlabelled instances. Brefeld et al. [2006], Sindhwani et al. [2006] successfully applied multiview learning to semi-supervised regression.

2.4 Manifold-based methods

Most manifold-based methods assume that there should be a low dimensional manifold structure embedded in the data space. Typically, these algorithms build graphs to rep-

resent all instances [Zhu, 2008]. The nodes represent the instances (both labelled and unlabelled) and the edges denote similarity between points. In order to perform predictions, these methods usually assume label smoothness in the obtained graph.

Most of algorithms that implement the manifold assumption are graph-based methods. Their loss functions estimate two objectives: the predicted labels should be similar to the neighbouring labels and predicted labels should deliver smoothness across the graph. Some of these methods only differ from each other by the choice of the cost function and the regulariser. For example, Blum and Chawla [2001] address semi-supervised learning as a graph mincut problem.

Spectral Graph Transducer (SGT) [Joachims, 2003] can be seen as a semi-supervised version of the K nearest-neighbours classifier. This binary classifier uses unlabelled instances to build a graph. The nature of its manifold assumption is in the fact that predictions are based on the neighbourhood of an instance within the graph.

The Semi-supervised Multilayer Perceptron [Malkin et al., 2009] is an extension of the standard Multilayer Perceptron (MLP) to binary SSC. In order to handle unlabelled data, such an algorithm includes four terms in its loss function. The first term produces the supervised learning; the second term is a graph regulariser (leads to smooth solutions over the graph); the third is an entropy regulariser; and the last is the term for weight decay. It uses stochastic gradient descent to optimise such a loss function. Malkin et al. [2009] demonstrated significant improvements when compared to supervised MLP. [Constantinopoulos and Likas, 2008] proposed the use of Probabilistic Radial Basis Function networks (PRBF) based on the EM algorithm for multi-class SSC. Such a method was employed in order to implement an incremental active learning procedure.

Most of the graph-based approaches only focus on the optimisation functions. The graph construction, an important part of the learning procedure, is not often included in such frameworks. In consequence, the issue of graph construction has not been extensively

studied yet [Zhu, 2008].

Manifold-based classifiers usually cannot generalise to unseen (test) data, that is, they are often inherently transductive, which is the case for the methods proposed in Belkin and Niyogi [2003], Joachims [2003], Zhu et al. [2003], Zhu and Lafferty [2005]. This can prevent the application of graph-based methods in problems requiring fully inductive classifiers. However, Belkin et al. [2006] proposed the Laplacian Support Vector Machine (LapSVM) classifier, an alternative extension to graph-based algorithms that is able to predict labels of newly available test instances, which avoids the retraining of the algorithm. And Melacci and Belkin [2011] reduced the computational complexity of LapSVM by solving its primal formulation.

These methods, except PRBF, are proposed for binary classification, which could be a shortcoming. They depend on the decomposition of multi-class datasets into a set of different binary tasks, leading to problems of imbalanced classification and different output scales of binary classifiers [Valizadegan et al., 2008].

2.5 Cluster-based methods

Among the methods based on cluster assumption, we can highlight the Transductive Support Vector Machine (TSVM) [Joachims, 2002]. It is an extension of SVM (Support Vector Machine). TSVM uses unlabelled data to find the decision boundary with the largest margin between classes. Unlike SVM, TSVM tries to maximise the margin with a linear boundary by considering both labelled and unlabelled instances, which might deliver higher generalisation accuracy [Vapnik, 1998]. Unlabelled instances help to avoid generating the decision boundary in dense regions [Zhu, 2008]. However, if dense regions are overlapping, TSVM might not find the correct decision boundary between such regions (clusters). And, in this case, this algorithm might be sensitive to the position of the limited labelled points in such clusters.

Chakraborty [2011] introduced the Bayesian semi-supervised Support Vector Machine (Semi-BSVM) for binary classification. Semi-BSVM aims to find the largest margin using both labelled and unlabelled data. Its loss function was designed with a penalty term to represent unlabelled data. Semi-BSVM was successfully compared to supervised classification methods when the unlabelled data was informative, especially in cases where the amount of unlabelled instances was greatly larger than labelled data. However, similarly as TSVM, Semi-BSVM is a binary classifier that is sensitive to overlapping high-density regions with labelled data in low-density regions.

Moreover, for multi-class classification problems, these methods have a similar drawback to other binary SSC algorithms: they depend on a suboptimal decomposition of the dataset into a number of independent binary classification problems.

Wang et al. [2012] proposed a multi-class classifier that, as TSVM, seeks the largest margin separator. It employs a loss function that uses the concept of label membership to weight the pertinence of a given instance to each class. In order to have more reliable labels for unlabelled points, such a function also considers each instance as a weighted average of its neighbours. However, this method does not distinguish between instances in low and high density regions, that is, an uncertain instance (with respect to its membership) lying on the border of a cluster has the same influence in the training as any other instance, whereas the intuition behind the cluster assumption suggests that the sharing of labels should be more reliable in high density regions.

As mentioned before, these cluster-based methods try to find potential gaps between high-density regions (clusters). When overlapping high-density regions are present, with sparse labelled instances on their borders, these classifiers may not produce good predictions, although these inherent clusters might be easily identified by clustering algorithms.

2.6 Ensemble methods

Combining several suitable classifiers, as an ensemble of learners, can enhance the generalisation performance of the group when compared to a single classifier [Nguyen et al., 2006]. Some of these methods use an ensemble of supervised algorithms, while others use an ensemble of semi-supervised methods as base learners.

Semi-supervised MarginBoost (SSMB) [d'Alché Buc et al., 2002] is a generalization of MarginBoost [Grandvalet et al., 2001]. It relies on the assignment of pseudo-labels, based on the current ensemble predictions, to unlabelled data. Such unlabelled instances are sampled for the training of a new semi-supervised base classifier at each iteration. SSMB minimises a loss function that includes a monotonically decreasing cost function, the margin for labelled instances and a pseudo-margin.[1] Such a method requires semi-supervised base classifiers.

ASSEMBLE [Bennett et al., 2002] is a semi-supervised margin-based boosting algorithm. Such an algorithm performs a greedy optimisation by maximising pseudo-margin using a boosting method and relies on pseudo-labels to perform the training of base classifiers. ASSEMBLE uses only instances with the highest confidence into the training set of base classifiers, which may only increase the margin without improving the current decision boundary. The trained base learners are likely to share the decision boundary with other classifiers at early iterations. Then, in case of early incorrect pseudo-labels, errors will affect future base learners and might degrade generalisation ability of the ensemble.

SemiBoost [Mallapragada et al., 2009] combines the similarity information among the instances and the classifier predictions to obtain more reliable pseudo-labels. It is a graph-based ensemble approach and its loss function has the smoothness, manifold and cluster

[1] Pseudo-margin is the confidence of a classifier on the predicted labels for the unlabelled instance. In order to increase the pseudo-margin between two instances, a classifier can increase its confidence for one instance and decrease it for the other point. It can also be interpreted as the decision boundary between unlabelled points.

assumptions. Such a method uses supervised base learners. Such a method can be used to improve classification accuracy of any supervised single algorithm using unlabelled instances.

SemiBoost is a binary classification algorithm. Valizadegan et al. [2008] proposed an extension of SemiBoost, Multi-Class Semi-Supervised Boosting (MCSSB), in order to perform multi-class classification.

Song et al. [2011] proposed another boosting-based algorithm for multi-class semi-supervised classification, where base classifiers are only required to have accuracy of at least $1/C$ (C is the number of classes). It employs a margin-based loss function that is prone to reinforce misclassification. This fact is due to the error function that attempts to maximise the margin between the ensemble output and the pseudo-labels of unlabelled data. However, this method delivered superior performance when compared to ASSEMBLE and supervised AdaBoost.

Zheng et al. [2009] extended the information regularisation framework to semi-supervised boosting. The authors proposed sequential gradient descent algorithms to optimise a semi-supervised loss function. This loss function incorporates all three SSL assumptions. The work of Song et al. [2011] also proposed a multi-class boosting classifier based on AdaBoost algorithm. SemiBoost, MCSSB and Song et al. [2011] employed supervised base classifiers trained with pseudo-labels based on the current ensemble predictions.

In Yu et al. [2012], the authors investigated the use of ensemble in high-dimensional SSC. Such an algorithm divides features into several subspaces, builds a graph for each subspace, trains a linear algorithm (base classifier) on each graph and combines these classifiers as an ensemble. The computational complexity of such a method is $\mathcal{O}(N^2D + NDD_{sub} + DD_{sub}^3)$, where N is the number of instances, D is the original dimensionality and D_{sub} is the subspace dimensionality.

SSMB, ASSEMBLE and the algorithms in Zheng et al. [2009] and Yu et al. [2012]

are designed for binary classification, therefore they depend on suboptimal decomposition methods to perform multi-class classification. Since such methods, along with MCSSB and Song et al. [2011], attempt to find the largest margin between classes, they might be sensitive to overlapping classes and to the position of labelled data in high-density regions. SSMB and ASSEMBLE do not handle semi-supervised assumptions explicitly [Chen and Wang, 2011].

For the optimisation process, SemiBoost and MCSSB derive their boosting algorithms by approximating loss functions with several bounds, so that the optimum of those bounds is used as their solutions. It is well known that the optimum of a loss function may be different from that its bounds. Thus, the tightness of these bounds may critically determine the performance of SemiBoost and MCSSB, even though their loss functions are convex.

Hady and Schwenker [2008] introduced a learning method in which a set of diverse classifiers are used in a co-training procedure. Such a semi-supervised framework can use any ensemble algorithm (for example, Bagging, AdaBoost) to build diverse ensembles. Such a co-training approach does not require multiple views, which can be useful in the cases where the division of the feature set is not straightforward. The authors demonstrated that error diversity among base classifiers leads to an effective co-training without requiring neither redundant and independent views nor different learning algorithms.

Hady et al. [2010] proposed a tree-structured ensemble where a multi-class problem is decomposed into a set of binary sub-problems. Each sub-problem (a binary classification) is represented as an internal node in a tree structure. The leaf nodes represent the classes. In each internal node, the algorithm performs a co-training procedure [Blum and Mitchell, 1998] using RBFN as base classifiers. The authors demonstrated that the combination of tree-structured ensemble and co-training is especially useful for classification with a large number of classes and a small amount of labelled data. However, this approach

uses only a base classifier to solve the binary sub-problems in the tree nodes. Such an algorithm does not exploit the generalisation ability of an ensemble for a single sub-problem. The tree-structured ensemble, similarly to co-training, may classify unlabelled instances incorrectly and such instances are used to train other classifiers, thus errors may be reinforced. Moreover, the co-training scheme employed in such a framework uses only the most confident unlabelled instances to teach the classifiers, which can lead to no improvement of boundary decision.

Saffari et al. [2009] used an objective function that includes a cluster regulariser and a margin regularisation term, which can lead to sensitiveness to overlapping classes and to the position of labelled data in high-density regions. The cluster regulariser uses assignments of a cluster algorithm to regularize the output of the ensemble according to the presence of labelled instances in a given cluster. However, such a term does not consider the cluster pertinence degrees of each instance, that is, it only uses the proportion of classes in a given cluster. Unlabelled instances are associated to the majority class of a cluster. In this sense, if a certain class is dominant in a cluster, it will be likely that all unlabelled instances are classified as such a majority class in that cluster. Even if there is a clear distinction between two classes in a given cluster, the proposed error function will force the classifier to consider all instances as belonging to the dominant class.

Saffari et al. [2010] proposed a multi-view margin-based algorithm that, in order to train a given base learner, uses priors generated by the other base classifiers. Such an algorithm introduced a loss function that can handle noisy priors. However, such a margin-based algorithm is sensitive to overlapping classes and to scarce labelled instances lying on the borders of clusters. Moreover, such a method does not consider the pseudo-label (priors) distribution in the neighbourhood structure of an unlabelled instance. If an incorrect prior occurs, the algorithm may reinforce such an error in the training of other base learners due to the use of supervised base classifiers that learn the exact priors that

are delivered by the rest of the ensemble.

RegBoost [Chen and Wang, 2011] employs three semi-supervised assumptions in its boosting algorithm. It uses a kernel density estimation approach to implement the cluster assumption, which penalises the classifier if it does not assign the same label to a pair of neighbour instances in a high-density region. However, if overlapping high-density regions are present RegBoost might not establish a good separation between these regions. Moreover, this algorithm is designed only for binary classification. As mentioned before, a decomposition technique, such as *one-vs-all* [Valizadegan et al., 2008], can be employed to extend the algorithm to multi-class problems. However, as expected, our experiments in Chapters 3, 4 and 5 demonstrated that RegBoost delivers inferior results when applied to multi-class real-world datasets.

2.7 Discussion

In this section, we discuss the limitations of the algorithms described in this Chapter and we relate such drawbacks to our motivations described in Section 1.5.

2.7.1 Limitations in cluster-based methods

As highlighted is this Chapter, most of the existing cluster-based techniques attempt to find potential gaps between high-density regions that might represent classes. In this sense, the generalisation performance of such approaches is limited when classes are overlapping and there is limited labelled data in a given cluster.

Datasets with overlapping classes are challenging for existing algorithms, since clusters do not have clear gaps between them. In this situation, decision boundaries generated by these methods are mainly determined by the position of the limited labelled instances within each cluster, which may cause the classifier to be sensitive to the position of those few labelled data in the given cluster. Such limitations corroborate our motivation

described in Section 1.5.1.

Additionally, most of semi-supervised algorithms are binary classifiers. Such methods depend on the decomposition techniques, such as *one-vs-all* or *one-vs-one*, in order to transform a multi-class datasets into a set of binary problems. This fact might lead to problems of imbalanced classification and different output scales of binary classifiers [Valizadegan et al., 2008].

Therefore, in Chapter 3, we present a multi-class classifier that is robust to overlapping classes and to the position of labelled data in clusters.

2.7.2 Unlabelled data in ensemble design

The ensemble techniques described in the previous section implement more than one SSL assumption. Intuitively, we expect that such ensemble-based algorithms that use more than two assumptions may yield higher average performance throughout datasets with unknown structures [Chen and Wang, 2011]. However, when only one assumption is present and/or the other assumptions are misleading, a specialised algorithm might be more effective, as demonstrated in Chapter 3.

The aforementioned ensembles may generate classifiers that can be confident on the prediction of unlabelled points, even though these unlabelled points are, in fact, misclassified. Moreover, except for MCSSB and Saffari et al. [2009], these ensembles are not specifically designed for multi-class problems and depend on suboptimal decomposition techniques, which might limit their performance as highlighted in the previous Section.

Except for SSMB[1], these algorithms rely on supervised base classifiers. In this sense, the base method will learn the exact pseudo-label assigned to each instance. In case of an incorrect pseudo-label occurs, this error might be reinforced by the base classifier and, therefore, would degrade the ensemble generalisation performance.

[1]SSMB uses Mixture models as semi-supervised base learners, which does not consider the structure of unlabelled by only enlarging the pseudo-margin, as in ASSEMBLE. This technique might reinforce errors during training.

Such limitations in SSC ensembles validate our motivation described in Section 1.5.2. Thus, Chapter 4 introduces an analysis of ensemble techniques with unlabelled data in different ensemble levels.

2.7.3 Time and memory requirements

The computational complexity of various popular SSC methods prevents their application to large datasets [Mann and McCallum, 2007]. Manifold-based algorithms require high computational effort due to the construction of a graph to represent the data. Such a graph has labelled and unlabelled points as vertices and labels are assigned to unlabelled vertices based on their neighbours.

Most existing cluster-based algorithms are also computationally intensive. TSVM [Joachims, 2002] attempts to find the largest margin between classes by searching for different label assignments for unlabelled data and calculating margins between dense regions of similarly labelled instances. Such a search is expensive and TSVM requires time of $\mathcal{O}(N^3)$, where N is the number of instances.

The aforementioned algorithms are binary classifiers. Thus, applying such time consuming algorithms to multi-class classification requires multiple and expensive runs caused by decomposition procedures [Saffari et al., 2009]. Such a drawback has a great impact on large-scale datasets.

Ensemble algorithms, in particular boosting techniques, were successfully employed in SSC Chen and Wang [2011], Saffari et al. [2009], Valizadegan et al. [2008]. For example, RegBoost requires time of $\mathcal{O}(VCN \log N + CTS^3 + TCVU)$, where V is the number of neighbours, C is the number of classes, T is the number of iterations, U is the number of unlabelled instances, for multi-class classification and, due to search for nearest neighbours, demands memory of $\mathcal{O}(N^2)$, which might be prohibitive for large datasets. Such a memory complexity, $\mathcal{O}(N^2)$, is common in algorithms that require nearest neighbours

computation or the storage of similarity measures.

Methods in Chen and Wang [2011], Yu et al. [2012] are binary classification algorithms and depend on the reduction of multi-class classification in multiple two-class problems. Such an issue is exacerbated by the training of several base classifiers.

In multi-class context, the computational complexity of MCSSB is $\mathcal{O}(TCU^2 + TS^3)$, where T is the number of boosting iterations, C is the number of classes and S is the number of sampled instances. MCSSB stores a similarity matrix and, likewise RegBoost, requires memory of $\mathcal{O}(N^2)$.

Applying the state-of-the-art algorithms described here to large-scale datasets is a challenging task due to their high computational complexity. Delalleau et al. [2006] proposed a sampling technique to reduce computational complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(S^2N)$, where S is the number of sampled instances. However, such a technique is designed for transductive graph-based algorithms and the experimental results in Chapelle et al. [2006] showed that the difference between such an algorithm and uniform random sampling is marginal. Other techniques for increasing efficiency can reduce the time complexity to $\mathcal{O}(S^3)$, where $S < N$, but also reduce performance [Zhu and Lafferty, 2005, Mann and McCallum, 2007].

Such time and memory requirements might limit the application of existing classifiers to large datasets. These limitations support our motivations described in Section 1.5.3. Therefore, in Chapter 5, we solve these issues by introducing a large-scale cluster-based algorithm for multi-class SSC.

2.8 Conclusions

In this Chapter, we presented the background for our study on semi-supervised ensembles. We categorised the different SSC algorithms according to their assumptions and mechanisms. We highlighted the advantages and drawbacks of each algorithm. We identified

the limitations that are common to cluster-based and ensemble methods. Such recurrent drawbacks are discussed and we present our motivations for this research.

In our discussions, we highlighted the reasons why cluster-based classifiers are sensitive to overlapping classes and to the position of the few labelled instances in a given cluster. These algorithms are designed for these ensembles are not specifically designed for multi-class problems and depend on suboptimal decomposition techniques, which might limit their performance. In ensemble learning, we pointed out that existing ensembles, during training, can be confident of the prediction of unlabelled points, even though these unlabelled points are, in fact, misclassified. This fact leads new base classifiers to reinforce such errors and degrade the ensemble performance. We also discussed that time and memory requirements limit the application of existing classifiers to large datasets.

These weaknesses of existing algorithms presented in this Chapter will be tackled in the next Chapters of this Thesis. In our experiments, we select algorithms described in this Section, so that we assess the improvement in both performance and efficiency of the proposed methods.

CHAPTER 3

Multi-class Semi-supervised Classification with Cluster-based Regularisation

In this Chapter, we address the research question raised in the Section 1.5.1 with the introduction of a new semi-supervised classifier. Such a method will be used as a base learner for semi-supervised ensemble techniques proposed in the next Chapters.

We introduce the Cluster-based Regularisation (ClusterReg) algorithm for multi-class SSC. ClusterReg uses soft partitions generated by clustering algorithms in a new regularisation technique. Such a classifier performs predictions according to the cluster structure along with limited labelled data. Our experiments confirmed that ClusterReg delivers good generalisation for real-world datasets. When data follows the cluster assumption, its performance is superior when compared to existing algorithms. Even when clusters have overlaps and misleading labelled instances, ClusterReg outperforms state-of-the-art methods.

The remainder of this Chapter is organised as follows. Next section presents the motivations for proposing a new semi-supervised classifier. Section 3.2 introduces the proposed algorithm in details. In Sections 3.3 and 3.4, we present instantiations of ClusterReg with

RBFN and MLP, respectively. Section 3.5 presents the experimental analysis. Finally, Section 3.6 discusses our contributions and Section 3.7 presents the conclusions.

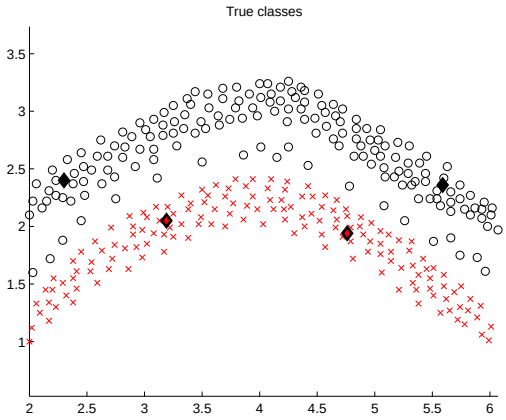
3.1 Introduction

Most cluster-based approaches attempt to find a low-density region to separate classes, avoiding generating the decision boundary inside clusters (traversing high-density regions). Transductive Support Vector Machine (TSVM) [Joachims, 2002] is a typical example. Existing cluster-based SSC methods do not work well when classes are overlapping. That is, the algorithm should be able to identify a gap with a relatively large number of instances.

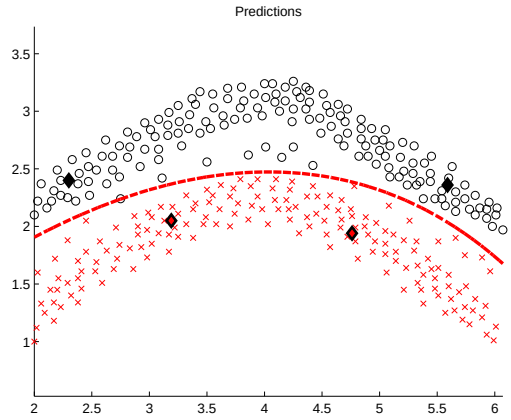
However, some clustering algorithms can often easily achieve better performance with overlapping classes when compared to the mentioned margin-based methods,[1] as demonstrated by simple synthetic examples in Figures 3.1–3.4.

The first dataset (two half-moons), in Figure 3.1a, has two labelled points (denoted as dark diamonds) and each "moon-shaped" cluster corresponds to one class. Both TSVM (Figure 3.1b) and ClusterReg (Figure 3.1c) are able to deliver a good decision boundary. The second dataset (Figure 3.2a) is a different version of the first with one inverted class, which produces a more challenging dataset. As TSVM is sensitive to the position of the single labelled points in each cluster (Figure 3.2b), it could not find a proper decision boundary. While ClusterReg, taking advantage of Self-Tuning Spectral Clustering (STSC) [Manor and Perona, 2004], was able to regularise the algorithm to fit the moon-shaped clusters, delivering a smooth decision boundary between classes (Figure 3.2c). The third dataset (Figure 3.3a) has three labelled instances and two classes. One class is sparsely

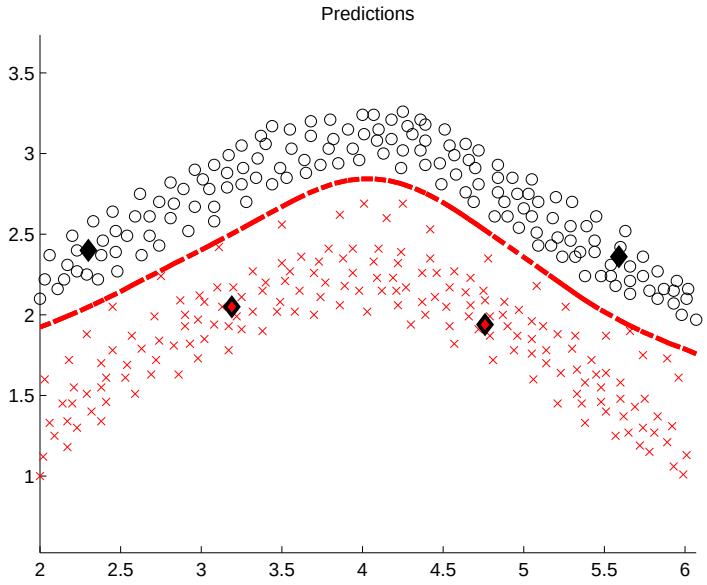
[1] Intuitively, as seen in Figure 3.4a, in some cases where there is no clear gap between clusters, discovering high-density regions is an easier task than finding low-density gaps between these regions. And clustering algorithm are specifically designed to search for high-density regions. Therefore, in some cases, clustering algorithms can deliver more accurate estimates over the data distribution than methods that seek the largest margin.



(a) True classes.



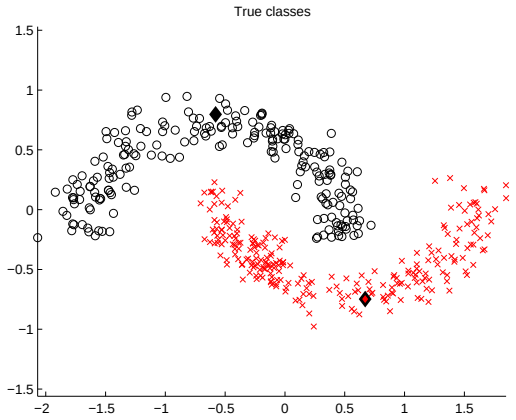
(b) Predictions of TSVM.



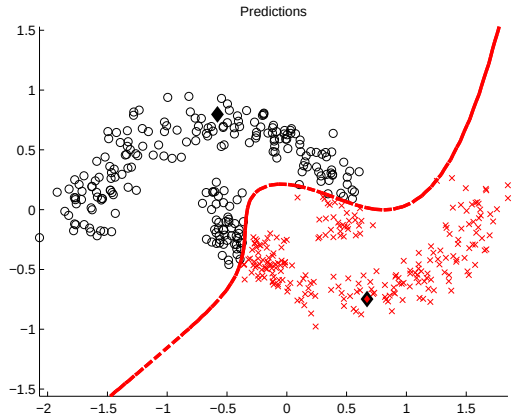
(c) Predictions of ClusterReg.

Figure 3.1: Synthetic two half-moons dataset. Each half-moon corresponds to one class.

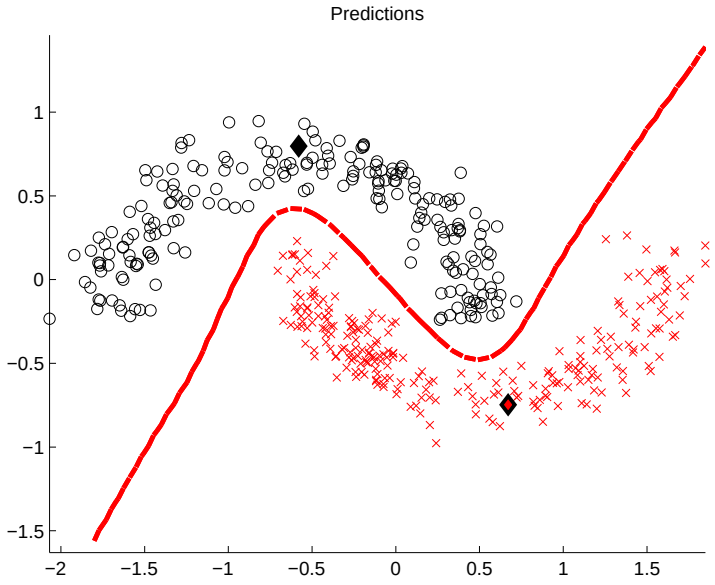
distributed while the second corresponds to a denser cluster inside the other class. The labelled points are arbitrarily placed to misguide classifiers. That is, the instances in the bottom of the sparse class are prone to be classified as belonging to the dense class. As expected, in Figure 3.3b, TSVM is not able to correctly predict the labels of the instances in the bottom of the sparse class, since there is no labelled instance in that



(a) True classes.



(b) Predictions of TSVM.

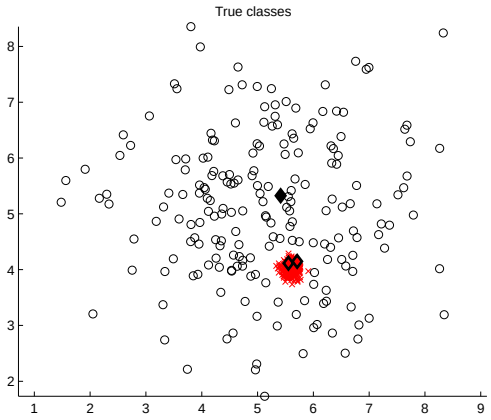


(c) Predictions of ClusterReg.

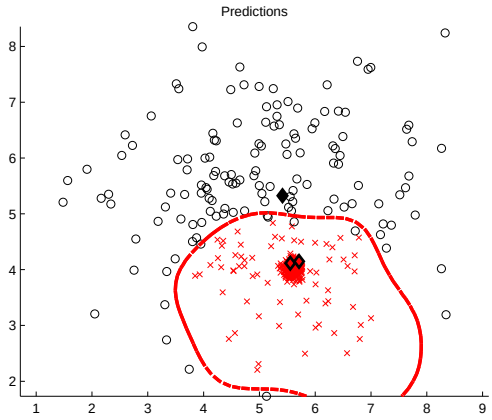
Figure 3.2: Inverted two half-moons.

region. However, ClusterReg incorporates the partition information from STSC to avoid traversing the dense and sparse cluster, which improves its robustness to the position of labelled instances, as shown in Figure 3.3c.

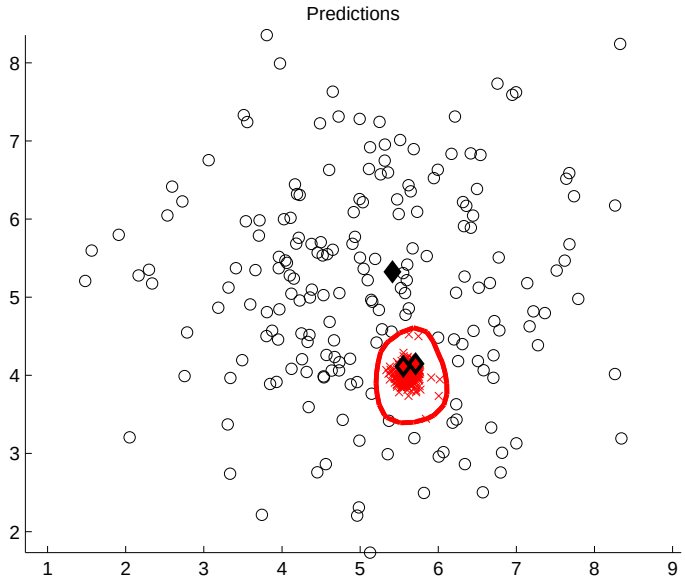
In Figure 3.4a, a two-dimensional dataset with six Gaussians corresponds to the true data distribution with six classes. The classes were designed to be overlapped and to



(a) True classes.

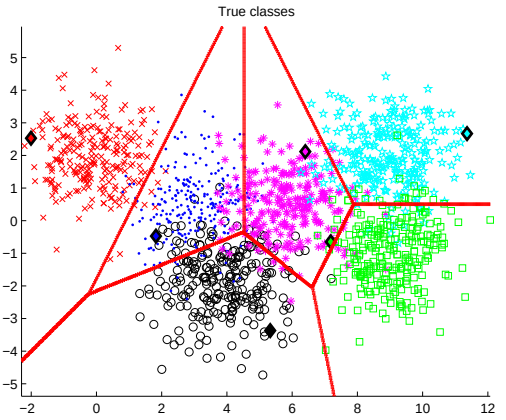


(b) Predictions of TSVM.

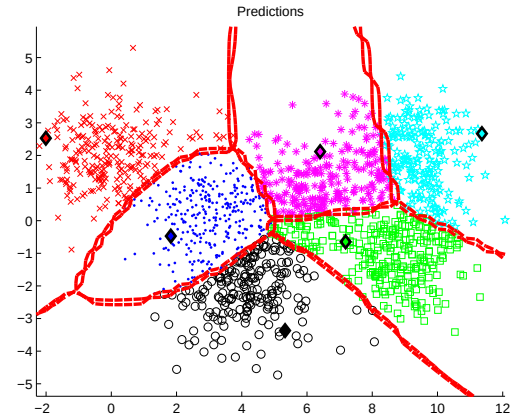


(c) Predictions of ClusterReg.

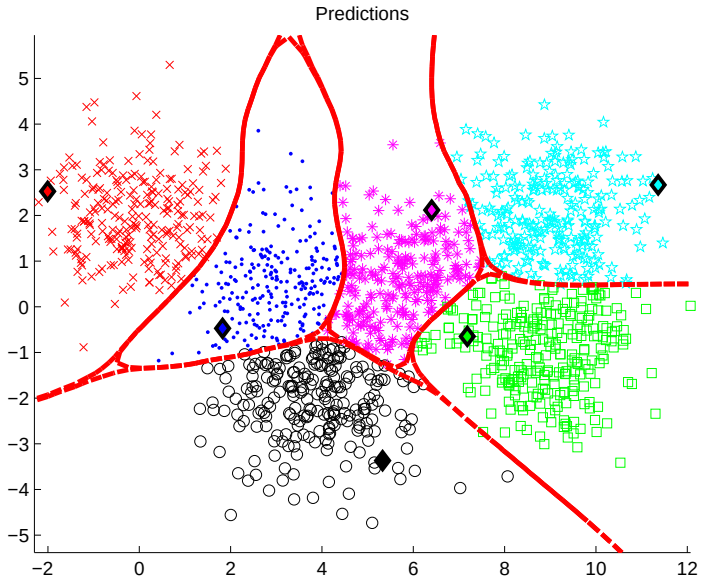
Figure 3.3: Dataset with one sparse and one dense classes corresponding to clusters. The denser cluster is placed in the sparser cluster. The labelled instances are arbitrarily chosen to mislead the classifiers. They would tend to classify the instances on the bottom of the sparse class as belonging to the tighter class. TSVM is sensitive to the position of the instances in the clusters, therefore it might not find the correct decision boundary. ClusterReg, as STSC can deal with clusters of arbitrary shapes, can take into account such partition and properly generate a decision boundary.



(a) True classes.



(b) Predictions of TSVM.



(c) Predictions of ClusterReg.

Figure 3.4: Dataset with 6 overlapping classes drawn from unit-variance isotropic Gaussians ($\mathcal{N}(\mu, \mathbf{I})$) and translated. Due to overlapping clusters, TSVM cannot find the appropriate decision boundary. ClusterReg, by considering the partition of a clustering algorithm, is able to find a better decision boundary.

possess a cluster structure. The labelled instances, denoted by black diamonds, were chosen to lie roughly on the borders of the classes.

Such data is challenging for existing algorithms, since these clusters do not have a clear

gap between them. The decision boundaries of these algorithms are mainly determined by the distribution of these limited labelled instances in the clusters. This fact may cause high sensitiveness to such labelled data, especially with scarce labels.

As an example of a multi-class case, shown in Figure 3.4b, TSVM could not find the appropriate gap and generated a decision boundary that traverses the clusters. However, clustering algorithms (such as Gaussian Mixture Models (GMM) or k-means [Xu and Wunsch, 2005], as verified in this case) may identify these six clusters.[1] When we apply the cluster structure to our method it becomes more robust to the position of the few labels in the clusters. Therefore, clustering algorithms can be properly employed in SSC to improve generalisation.

In this Chapter, we propose to incorporate clustering algorithms in ClusterReg to overcome the issues mentioned above. In this algorithm, we also consider the probability of each instance belonging to each cluster to regularise the proposed loss function in next Section. As shown in Figure 3.4c, unlike other cluster-based methods, our method can easily establish a decision boundary between clusters without being misguided by neither the overlapping classes nor the position of scarce labels. ClusterReg achieves such robustness by incorporating clustering algorithm into its mechanism. This simple case confirmed the benefits of our algorithm for overlapping classes.

ClusterReg regards the structure arising from the clustering algorithm as a soft partition. That is, each instance is assigned a posterior probability of belonging to a given cluster, unlike hard partition where clusters are strictly disjoint. By using soft partitions (also known as soft clustering), we can address uncertain instances (likely to lie on low density region, that is, in the border of clusters) differently from the more confident ones (likely to lie on denser regions of clusters). Soft clustering helps the algorithm regard

[1] The example shown in Figure 3.4 is suitable for k-means and GMM, because the clusters are spherical. There are situations where ClusterReg can use other clustering methods, such as spectral-based clustering algorithms, to estimate the cluster structure with arbitrary shape [Manor and Perona, 2004], which is the case for the datasets in Figures 3.1a, 3.2a and 3.3a.

uncertain instances (with low probabilities for all clusters) as instances lying on gaps, therefore helping the classifier to generate the decision boundary in such low-density region (according to the clustering algorithm).

The contribution of this Chapter is to propose an algorithm that employs any clustering algorithm[1] into SSC to regularise its training. The proposed algorithm (i) is robust to the presence of fewer labelled points, and is robust to the position of labelled data in clusters by considering the strength of clustering algorithms in a natural manner and (ii) is able to improve the performance of a given classifier when the classes or clusters overlap, compared to other cluster-based algorithms.

ClusterReg can use any clustering algorithm with a proper processing of its output. It can employ any classifier that is able to minimise the proposed loss function. Therefore, ClusterReg can be seen as a framework for SSC methods.

3.2 Cluster-based Regularisation algorithm

In this Section, we present the proposed multi-class semi-supervised algorithm, ClusterReg. First, we introduce the new loss function with a regularisation term based on posterior probabilities of cluster membership. Then, we present the multi-class version of such a loss function with cross-entropy for multi-class classification. And, finally, we introduce an initialisation procedure for classifiers in SSC.

3.2.1 General architecture and notations

Our proposed algorithm uses posterior probabilities of cluster membership to regularise the learning of unlabelled data. In order to learn an unlabelled instance n, ClusterReg uses a weighted average of the network outputs and potential true labels in the neighbourhood of that instance. Such an average is employed as an estimated label for the point n. In

[1] In this Chapter, we use clustering algorithm, K-means [Xu and Wunsch, 2005], STSC, GMM and Fuzzy GK Clustering [Gustafson and Kessel, 1978], to evaluate ClusterReg.

this sense, the desired output for instance n is assumed to be similar to the outputs and labels of its neighbours. That is, if the network assigns different labels to two similar instances (neighbours according to the output of a clustering algorithm), the training will be regularised. More similar neighbours have a greater impact on the estimated label. The contribution of each neighbour for the estimated label of n is weighted by a penalty that measures how similar two instances are. Such a penalty is calculated according to the posterior probabilities assigned by a clustering algorithm to each instance. Learning an unlabelled instance involves assessing its neighbourhood, which improves the reliability of the estimated label that is assigned to that point. Apart from the impact on the estimated label of instance n, the posterior probabilities generated by a clustering algorithm also weights the importance of the unlabelled instance n in comparison to the other instances.

The regularisation mechanism in ClusterReg avoids the generation of a decision boundary that traverses similar instances according to a clustering algorithm. It also reduces the impact of uncertain instances (potentially lying on low-density regions) on the training, therefore it produces a robust generalisation.

ClusterReg performs SSC by using neural networks to minimise a loss function especially designed for multi-class SSC. This function includes both supervised and semi-supervised losses. In order to learn the labelled instances, the supervised loss denotes the discrepancy between the posterior class probabilities $\mathbf{f}(\mathbf{x}_n) = \mathbf{f}_n = \{f_{ni}\}_{i=1}^{C}$, where $0 \leq f_{ni} \leq 1$, $i = 1, \ldots, C$ and $\sum_{i=1}^{C} f_{ni} = 1$, produced by the network and the desired class probabilities $\mathbf{y}_n = \{y_{ni}\}_{i=1}^{C}$, where $0 \leq y_{ni} \leq 1$, $i = 1, \ldots, C$ and $\sum_{i=1}^{C} y_{ni} = 1$.

The semi-supervised loss is represented by a regularisation term that implements both smoothness and cluster assumptions. The estimated label (also known as pseudo-label) u_{ni} for instance and class i is a weighted average of the outputs $\hat{y}_{ni} = f_{ni}$ of the neural network for its neighbours. In the case where a neighbour is a labelled instance $\hat{y}_{ni} = y_{ni}$, as shown in Equation 3.2. The weight assigned to each pair of instances, n and

k, is a penalty value $\gamma(\mathbf{q}_n, \mathbf{q}_k)$ that is related to the similarity between the vectors of probabilities of cluster membership, $\mathbf{q}_n$ and $\mathbf{q}_k$, of both instances. That is, if the classifier assigns different classes for similar instances, the regularisation will increase. Besides the influence on the estimated label u_{ni}, the cluster output $\mathbf{q}_n$ also weights the importance of instance n according to how certain the clustering algorithm is about the cluster assigned to that point. The impact of n is weighted by the highest probability in the vector $\mathbf{q}_n$, which is denoted by $\max(\mathbf{q}_n)$. We denote the matrix of posterior probabilities of cluster membership (soft partition) as $\mathbf{Q} = [q_{ij}]_{N \times K}$ with K clusters and N instances where the row vector $\mathbf{q}_n$ contains the probabilities of instance n belonging to each one of the K clusters.

With the minimisation of the proposed loss function, a neural network can learn the labelled instances and use the unlabelled points in a regularisation mechanism to avoid producing a decision boundary on high-density (sub-optimal) regions. The general architecture of ClusterReg is presented in Figure 3.5. And its steps are as follows.

1. Perform clustering to obtain matrix $\mathbf{Q}$ of posterior probabilities.

2. Pairwise penalty is calculated according to the posterior probabilities generated by a clustering algorithm.

3. The initialisation procedure assigns the initial pseudo-labels to the unlabelled instances according to the true labels available in each cluster.

4. The neighbourhood of a given instance is defined as those instances with the highest penalty values relative to that instance.

5. With the initial pseudo-labels, penalty values and nearest neighbours at hand, the classifier is trained for a small number of iterations with fixed pseudo-labels generated by the initialisation procedure. Then, the training resumes with updated pseudo-labels at each iteration.

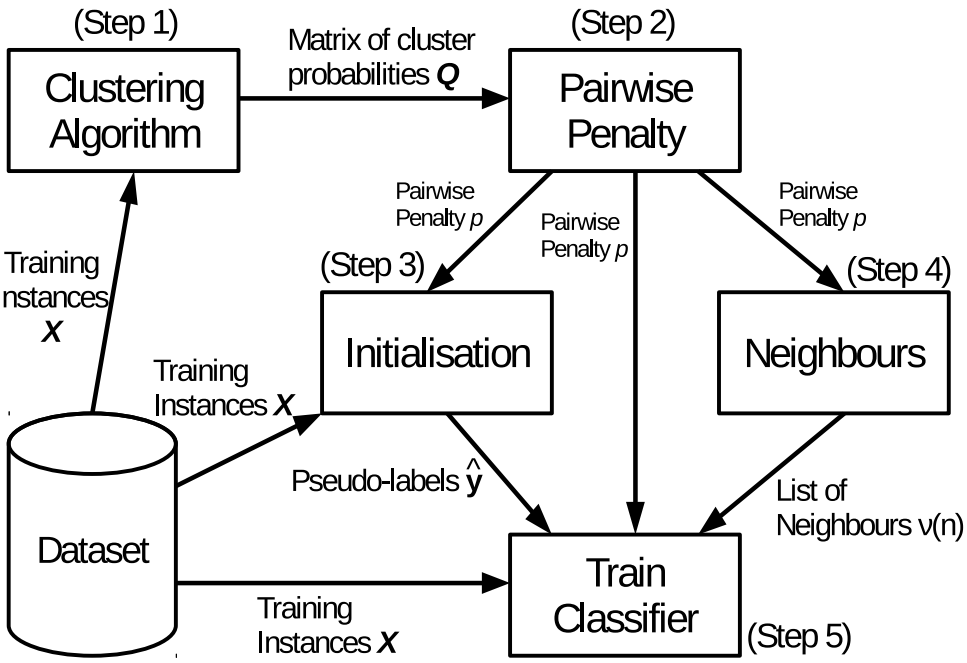


Figure 3.5: ClusterReg's architecture.

We detail these steps in the following sections.

3.2.2 A new semi-supervised loss function with cluster-based regularisation

In SSC, the training set $\mathbf{X} = \mathbf{L} \cup \mathbf{U}$ is composed of L labelled instances $\mathbf{L} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{L}$, where $0 \leq y_{ni} \leq 1$, $i = 1, \ldots, C$ and $\sum_{i=1}^{C} y_{ni} = 1$, and U unlabelled instances $\mathbf{U} = \{\mathbf{x}_n\}_{n=L+1}^{N}$ and $N = L + U$, often $U \gg L$. The aim of SSC is to improve a classifier in comparison to using the labelled data $\mathbf{L}$ alone.

In this work, we propose a new classifier in order to include clustering information in SSC. We use the output of clustering algorithms to regularise the loss function of the proposed algorithm. The first term of such a loss function is fully supervised, employing only the labelled instances to measure the difference between the classifier output and the true labels. The second term represents the semi-supervised regularisation procedure.

A loss function measures how predictions and desired output (true labels) differ. In

this algorithm, we assign labels to unlabelled data according to penalty values and neighbourhood defined through the use of clustering algorithm. Since there are no true labels for unlabelled data, Equation 3.1 denotes the estimated desired output for an unlabelled instance.

$$u_{ni} = \frac{\sum_{k \in \nu(n)} \gamma(\mathbf{q}_k, \mathbf{q}_n) \hat{y}_{ki}}{\sum_{k \in \nu(n)} \gamma(\mathbf{q}_k, \mathbf{q}_n)}, \tag{3.1}$$

where

$$\hat{y}_{ki} = \begin{cases} y_{ki}, & \text{if } k \text{ is labelled} \\ f_{ki}, & \text{if } k \text{ is unlabelled,} \end{cases} \tag{3.2}$$

where $\mathbf{f}_n$ denotes the output vector with posterior class probabilities of the classifier for instance n and C is the number of classes. The estimate u_{ni} is the probability of class i given instance n and $0 \leq u_{ni} \leq 1$, $i = 1, \ldots, C$ and $\sum_{i=1}^{C} u_{ni} = 1$. Such estimates are updated at each iteration of the training algorithm. The function $\nu(n)$ represents the set of nearest neighbours of n. The penalty $\gamma(\mathbf{q}_k, \mathbf{q}_n)$ is calculated according to the partition provided by the cluster algorithm. Basically, if instances n and k are similar (according to the structure of clustering method), a higher penalty will be assigned to that pair. $\hat{y}_{ki}$ can be either the true label y_{ki} if k is a labelled instance or the output f_{kj} if k is unlabelled. When k is unlabelled, $\hat{y}_{kj}$ is also known as pseudo-label of k. Then, u_{ni} (in fact, a posterior class probability) becomes a weighted average of current pseudo-labels of the neighbourhood of n.

The output of a clustering algorithm is a soft partition $\mathbf{Q}$. For example, $\mathbf{q}_n = (0.3, 0.1, 0.6)$ denotes that n has 30% of chance of belonging to the first cluster and so on. Consequently, the vector sums to one. And n belongs to the third cluster as it holds the highest probability. The proposed loss function is in Equation 3.3.

$$\mathcal{L}(\mathbf{f}_n, \mathbf{y}_n) = -\sum_{i=1}^{C} \left\{ \frac{I_{nL}}{L} \mathcal{C}[f_{ni}, y_{ni}] + \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} \mathcal{C}[f_{ni}, u_{ni}] \right\}, \tag{3.3}$$

where $I_{nL} = 1$ if $n \in \mathbf{L}$ and 0 otherwise, and $I_{nU} = 1$ if $n \in \mathbf{U}$ and 0 otherwise. f_{ni} denotes the output of the classifier for class i and instance n. $\mathcal{C}[y_{ni}, f_{ni}]$ can be any monotonically decreasing cost function, for example, mean squared error or cross entropy. $\gamma(\mathbf{q}_n, \mathbf{q}_k)$ is the penalty assigned to instance n and k. The parameter λ denotes the trade-off between the supervised loss and semi-supervised regularisation. C is the number of classes. And $\max(\mathbf{q}_n)$ returns the maximum value in the vector $\mathbf{q}_n$ to indicate the most probable cluster that instance n belongs to.

3.2.3 Cluster-based regularisation

The penalty function, presented in Equation 3.4, measures the similarity between vectors $\mathbf{q}_n$ and $\mathbf{q}_k$. By doing so, we consider the similarity as a direct outcome from the clustering algorithm. Such penalty function uses a similarity measures s that includes the correlation coefficient c (in Equation 3.6) and a similarity measure d based on Euclidean distance (in Equation 3.7). The penalty function maps the similarity function s into penalty values in the regularisation term. The function $s(\mathbf{q}_n, \mathbf{q}_k)$ in Equation 3.5 is normalised in $[0, 1]$.

$$\gamma(\mathbf{q}_n, \mathbf{q}_k) = \sin\left(\frac{\pi}{2}\left(s(\mathbf{q}_n, \mathbf{q}_k)\right)^{\kappa}\right), \tag{3.4}$$

The parameter κ controls the steepness of the mapping from similarity to penalisation. This value regulates the degree in which a decision boundary traverses a cluster. If we increase κ, we relax the cluster assumption by allowing the classifier to divide a high-density region. On the other hand, decreasing this parameter forces the classifier to avoid producing the decision boundary inside clusters. This form of penalty is flexible to allow different levels of penalisation for highly similar instances, while assigning low penalty to instances with low similarity, according to the parameter κ [Chen and Wang, 2011].[1]

[1] κ is chosen as a positive value in $[1, 12]$. Lower values lead to no difference of penalty for dissimilar instances and higher values do not penalise even the most similar instances.

There are various approaches to measure the similarity between vectors. In this Chapter, we focus on the correlation coefficient and the Euclidean distance (transformed into similarity) between the probability vectors $\mathbf{q}_n$ and $\mathbf{q}_k$. Using Euclidean distance alone may not capture all the information between two vectors. Suppose we have the probability vectors $\mathbf{u} = (0.8, 0, 0.2)$, $\mathbf{v} = (0.5, 0.2, 0.3)$ and $\mathbf{w} = (0.8, 0.2, 0)$, we can notice that the instances they represent belong to the same cluster, which is the one with the highest probability. The Euclidean distance between $\mathbf{u}$ and $\mathbf{v}$ is $||\mathbf{u} - \mathbf{v}|| = 0.37$ and $||\mathbf{u} - \mathbf{w}|| = 0.28$. However, $\mathbf{v}$ has higher chance of belonging to the third cluster than the second, which is also the case for $\mathbf{u}$. Whereas for $\mathbf{w}$, the second highest probability is for the second cluster. In this sense, $\mathbf{v}$ should be the point more similar to $\mathbf{u}$, instead of $\mathbf{w}$. Therefore, although all the corresponding instances belong to the same cluster, the correlation between their cluster probability distribution should be considered. Then, we use Pearson's correlation coefficient along with the Euclidean distance to calculate the penalisation for a pair of points.

Formally, Equation 3.5 denotes the proposed similarity function, which is normalised in $[0, 1]$.

$$s(\mathbf{q}_n, \mathbf{q}_k) = c(\mathbf{q}_n, \mathbf{q}_k) * d(\mathbf{q}_n, \mathbf{q}_k). \tag{3.5}$$

Equation 3.6 shows the similarity concerning the correlation between two probability vectors and $c(\mathbf{q}_n, \mathbf{q}_k)$ is in $[-1, 1]$.

$$c(\mathbf{q}_n, \mathbf{q}_k) = \frac{\sum_{i=1}^{K}(q_{ni} - \bar{q}_n)(q_{ki} - \bar{q}_k)}{\sqrt{\sum_{i=1}^{K}(q_{ni} - \bar{\mathbf{q}}_n)^2}\sqrt{\sum_{i=1}^{K}(q_{ki} - \bar{\mathbf{q}}_k)^2}}, \tag{3.6}$$

where $\bar{q}_n$ is the mean of the vector $\mathbf{q}_n$. For the second similarity measure, we compute all the pairwise Euclidean distances between the probability vectors and normalise them in $[0, 1]$. Then, we transform the Euclidean distance into similarity as shown in Equation 3.7. Therefore, similar instances should be close to each other and highly correlated.

$$d(\mathbf{q}_k, \mathbf{q}_n) = 1 - \frac{||\mathbf{q}_k - \mathbf{q}_n|| - d_{min}}{d_{max} - d_{min}}, \tag{3.7}$$

where d_{max} and d_{min} are the maximum and minimum Euclidean distances over all pairwise distances, respectively.

As we intend to use the structure arising from the clustering algorithm to calculate the similarity in the regularisation term, we also employ this information to find the nearest neighbours $\nu(n)$. Then, the nearest neighbours of n are the V instances with the highest $\gamma(\mathbf{q}_k, \mathbf{q}_n)$.

Following the smoothness assumption, the regularisation term in Equation 3.3 penalises the classifier if it assigns different labels to similar instances. Such an assumption is implemented by the product $\gamma(\mathbf{q}_k, \mathbf{q}_n)\mathcal{C}[f_{ni}, u_{ni}]$. That is, if the classifier produces different outputs for two similar instances, the loss and penalty will be high, causing a large regularisation to the training. On the other hand, if the penalty is low (the instances are not similar according to the clustering algorithm), the assignment of distinct labels to the couple of instances will be irrelevant.

Regarding the cluster assumption, we use the density information in $\mathbf{Q}$ to regularise the classifier, following the posterior probabilities generated by a clustering algorithm. In order to improve the cluster assumption, we also add the maximum value in the probability vector, $\max(\mathbf{q}_n)$, as a factor in the second term of the loss function. It weights the importance of instance n as an estimate of the density in its region. The higher this value the higher the density is. Thus, we penalise the training if the classifier assign two different labels to the instance to be learned n and its neighbour k; and the penalty will be even higher if n is in a high density region, according to the clustering algorithm. Therefore, the classifier will avoid delivering a decision boundary that traverses clusters.

3.2.4 Multi-class cluster-based loss function

In this work, we instantiate ClusterReg with neural networks. In multi-class classification, the output nodes of a neural network represent classes. In this sense, each output node denotes the predicted probability (or confidence) for its respective class. Therefore, the output vector $\mathbf{f}_n$ represents a probability distribution and the cross-entropy function can properly measure the difference between such a predicted distribution and the desired distribution [Plunkett and Elman, 1997].

Cross-entropy and softmax activation function are a natural pairing, therefore both functions should be used in multi-class classification [Dunne Campbell, 1997, Bishop, 2006]. In this sense, we can instantiate the proposed loss function to multi-class classification using the cross-entropy cost function (Equation 3.8).

$$\mathcal{L}(\mathbf{f}_n, \mathbf{y}_n) = -\sum_{i=1}^{C} \left\{ \frac{I_{nL}}{L} y_{ni} \log\left(f_{ni}\right) + \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} u_{ni} \log\left(f_{ni}\right) \right\}, \tag{3.8}$$

And we employ softmax activation function (Equation 3.9) for the output nodes.

$$f_{ni} = \text{softmax}(z_{ni}) = \frac{\exp(z_{ni})}{\sum_{j=1}^{C} \exp(z_{nj})}, \tag{3.9}$$

where z_{ni} is the net input for output node i, that is, the linear combination of weights and inputs of the node i for instance n.

$$z_{ni} = \sum_{j=1}^{M} \phi_{nj} * w_{ij} = \boldsymbol{\phi}_n * \mathbf{w}_i,$$

where $\mathbf{w}_i = \{w_{ij}\}_{j=1}^{M}$ is a column vector of weights, $\boldsymbol{\phi}_n = \{\phi_{nj}\}_{j=1}^{M}$ is a row vector with the output of the hidden nodes of the neural network[1] and M is the number of hidden

[1]The matrix $\boldsymbol{\phi} = \{\boldsymbol{\phi}_n\}_{n=1}^{N}$ is also known as the design matrix of RBFN.

nodes. And the output $0 \leq f_{ni} \leq 1$, $i = 1, \ldots, C$ and $\sum_{i=1}^{C} f_{ni} = 1$.

3.2.5 Initialisation procedure

For many SSC algorithms, if the classifier assigns the same class to every unlabelled instance, the training error will be in a local optimum [Mann and McCallum, 2007]. This fact is due to the loss function comparing a predicted output with similar outputs of its neighbours. It is important to highlight that the loss associated to this local optimum is greater than that of the desired solution. However, in some cases, the classifier might not be able to obtain a lower loss and find a more useful local minima. In order to overcome such a meaningless local optimum, we use an initialisation procedure that employs the distribution of labelled points in clusters to assign initial pseudo-labels to unlabelled data.

In this sense, the proposed initialisation procedure of ClusterReg has a great impact in the outcome of the training. This procedure ensures that, at the first iterations, the classifier has more reliable estimates over the labels of unlabelled instances. These estimates are weighted pairwise penalty values within each cluster. Without such a technique, the generated decision boundary would be highly ineffective, degrading the generalisation of the method.

At the start, ClusterReg does not have the estimated labels (pseudo-labels) of the neighbours of a given instance n to perform regularisation. The output of cluster algorithm is employed to set the pseudo-labels $\hat{\mathbf{y}}_n = \{\hat{y}_{ni}\}_{i=1}^{C}$. We use the sum of labels present in a cluster, weighted by penalty values $\gamma(\mathbf{q}_n, \mathbf{q}_k)$, to assign pseudo-labels to unlabelled instances in such cluster [Chen and Wang, 2011]. If there are no labelled points in a cluster, equal probabilities will be assigned to each class. For class i of unlabelled instance n in cluster Ψ we have:

$$\hat{y}_{ni} = \frac{\sum_{k \in \Psi} I_{kL} * \gamma(\mathbf{q}_n, \mathbf{q}_k) * y_{ki}}{\sum_{k \in \Psi} I_{kL} * \gamma(\mathbf{q}_n, \mathbf{q}_k)}. \tag{3.10}$$

A pre-training procedure, with the pseudo-label values assigned to unlabelled in-

stances, is performed for a certain number of iterations and $\hat{y}_{ni}$ is not updated throughout such iterations. In this study, we use 10 iterations of pre-training, as different numbers did not improve performance in preliminary experiments.

3.3 Cluster-based Regularisation with Radial Basis Functions Network

In this work, we instantiate ClusterReg with RBFN. We chose cross entropy as the loss function and softmax as the output activation function [Bishop, 2006], since they form a natural pairing that leads to more accurate generalisation [Dunne Campbell, 1997]. Additionally, cross entropy might be robust to datasets with limited amounts of data [Kline and Berardi, 2005].

RBFN is efficient [Nabney, 1999] and can be easily adapted to our method. The training of a RBFN consists of two phases: unsupervised training, where the centres of the nodes in the hidden layer are selected; and supervised training [Nabney, 1999], where the weights of the output nodes are trained.

In the unsupervised training, we employ the gaussian activation function for the hidden nodes. The centres of such gaussians coincide with the available training instances. And the gaussian widths are tuned as described in Section 3.5.1.

In the supervised training phase of the RBFN, we adapt the loss function in Equation 3.3 and add a weight regularisation term. The output function is the softmax function $f_{ni} = \text{softmax}(z_{ni})$, as in Equation 3.9. Equation 3.11 presents the loss function of our classifier.

$$\mathcal{L}(\mathbf{f}_n, \mathbf{y}_n) = -\sum_{i=1}^{C} \left\{ \frac{I_{nL}}{L} y_{ni} \log\left(f_{ni}\right) + \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} u_{nj} \log\left(f_{ni}\right) - \alpha \frac{\mathbf{w}_i^T \mathbf{w}_i}{2} \right\}, \tag{3.11}$$

where $\mathbf{w}_i$ is the weight vector for output node (class) i and α controls the amount of

weight regularisation.

The training algorithm for our base learner is the Iterative Reweighted Least Squares (IRLS) [Bishop, 2006]. IRLS consists of T Newton-Raphson steps to update network weights, as in Equation 3.12.

$$\Delta \mathbf{w}_j = -\mathbf{H}^{-1} * \left[\frac{\partial \mathcal{L}(\mathbf{f}, \mathbf{y})}{\partial \mathbf{w}_j} \right], \tag{3.12}$$

where $\mathbf{H}$ is the Hessian matrix and $\frac{\partial \mathcal{L}}{\partial \mathbf{w}_j}$ is the gradient of loss function $\mathcal{L}$ with respect to (w.r.t.) the weight vector $\mathbf{w}_j$.

In order to use the IRLS method, we calculate the first and second derivatives of $\mathcal{L}$ w.r.t. $\mathbf{w}_j$. We consider $\alpha \frac{\mathbf{w}_i^T \mathbf{w}_i}{2}$ separately. The gradient $\frac{\partial \mathcal{L}(\mathbf{f},\mathbf{y})}{\partial \mathbf{w}_j}$ can be obtained with the chain rule as follows.[1]

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}_j} = \frac{\partial \mathcal{L}}{\partial f_i} * \frac{\partial f_i}{\partial z_j} * \frac{\partial z_j}{\partial \mathbf{w}_j} + \frac{\partial \left(\sum_{i=1}^{C} \alpha \frac{\mathbf{w}_i^T \mathbf{w}_i}{2} \right)}{\partial \mathbf{w}_j}$$

The first factor of the right-hand side becomes:

$$\frac{\partial \mathcal{L}}{\partial f_i} = -\sum_{i=1}^{C} \left\{ \frac{I_{nL}}{L} \frac{y_i}{f_i} + \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} \frac{u_i}{f_i} \right\}$$

Then, the second factor is

$$\frac{\partial f_i}{\partial z_j} = \begin{cases} f_i(1 - f_i), & \text{if } i = j \\ -f_i f_j, & \text{if } i \neq j, \end{cases}$$

which can be rewritten as

$$\frac{\partial f_i}{\partial z_j} = f_i(\delta_{ij} - f_j),$$

[1] We suppress the subscript that indicates instance n when the context is clear.

where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

Finally, the third factor is

$$\frac{\partial z_j}{\partial \mathbf{w}_j} = \frac{\partial \sum_j w_{ij} * \boldsymbol{\phi}_j}{\partial \mathbf{w}_j} = \frac{\partial \boldsymbol{\phi} * \mathbf{w}_i}{\partial \mathbf{w}_j} = \boldsymbol{\phi}$$

The gradient of the weight regularization term is:

$$\frac{\partial \left(\sum_{i=1}^{C} \alpha \frac{\mathbf{w}_i^T \mathbf{w}_i}{2} \right)}{\partial \mathbf{w}_j} = \alpha \mathbf{w}_j$$

Then, rewriting and adding the weight regularization, the gradient becomes:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}_j} = -\frac{I_{nL}}{L} \sum_{i=1}^{C} \left\{ \frac{y_i}{f_i} * f_i(\delta_{ij} - f_j) \right\} \boldsymbol{\phi} - \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} \sum_{i=1}^{C} \left\{ \frac{u_i}{f_i} * f_i(\delta_{ij} - f_j) \right\} \boldsymbol{\phi} + \alpha \mathbf{w}_j$$

$$= -\frac{I_{nL}}{L} \left(\sum_{i=1}^{C} y_i \delta_{ij} - \sum_{i=1}^{C} y_i f_j \right) \boldsymbol{\phi} - \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} \left(\sum_{i=1}^{C} u_i \delta_{ij} - \sum_{i=1}^{C} u_i f_j \right) \boldsymbol{\phi} + \alpha \mathbf{w}_j,$$

since $\sum_{i=1}^{C} y_i = 1$ and $\sum_{i=1}^{C} u_i = 1$, we have:

$$\frac{\partial \mathcal{L}(\mathbf{f}_n, \mathbf{y}_n)}{\partial \mathbf{w}_j} = \frac{I_{nL}}{L} \left(f_{nj} - y_{nj} \right) \boldsymbol{\phi}_n + \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} \left(f_{nj} - u_{nj} \right) \boldsymbol{\phi}_n + \alpha \mathbf{w}_j \tag{3.13}$$

In this sense, cross-entropy and softmax activation function are a natural pairing due to the fact that $\frac{\partial \mathcal{L}}{\partial z_j}$ is of the form $f_j - y_j$ [Bishop, 2006].

For the Newton-Raphson method, we also calculate the second derivative $\frac{\partial^2 \mathcal{L}}{\partial \mathbf{w}_j \partial \mathbf{w}_k}$ with the chain rule, then we have:

$$\frac{\partial^2 \mathcal{L}}{\partial \mathbf{w}_j \partial \mathbf{w}_k} = \frac{\partial^2 \mathcal{L}}{\partial \mathbf{w}_j \partial f_j} * \frac{\partial f_j}{\partial z_k} * \frac{\partial z_k}{\partial \mathbf{w}_k} + \frac{\partial \alpha \mathbf{w}_j}{\partial \mathbf{w}_k}$$

The first factor on the right-hand side is:

$$\frac{\partial^2 \mathcal{L}}{\partial \mathbf{w}_j \partial f_j} = \left(\frac{I_{nL}}{L} + \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} \right) \boldsymbol{\phi}.$$

The second factor becomes

$$\frac{\partial f_j}{\partial z_k} = f_j(\delta_{jk} - f_k).$$

The third factor is

$$\frac{\partial z_k}{\partial \mathbf{w}_k} = \boldsymbol{\phi}.$$

And, finally, the weight regularisation term becomes

$$\frac{\partial \alpha \mathbf{w}_j}{\partial \mathbf{w}_k} = \alpha.$$

Then, the Hessian matrix is a block matrix $\mathbf{H} = [H_{jk}]_{MC \times MC}$ (M is the number of hidden nodes), where each block is

$$H_{jk} = \left[\frac{\partial^2 \mathcal{L}}{\partial \mathbf{w}_j \partial \mathbf{w}_k} \right] = \sum_{n=1}^{N} \left\{ \left(\frac{I_{nL}}{L} + \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} \right) * f_{nj}(\delta_{jk} - f_{nk}) * \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T + \alpha \right\}. \tag{3.14}$$

The update rule in Equation 3.12 is iterated until a stopping criterion (for example, increase of validation error) is met. Algorithm 1 describes the ClusterReg method.

In ClusterReg, we can apply any clustering algorithm. Four algorithms from various clustering approaches, namely k-means, STSC, GMM and Fuzzy GK Clustering are employed in this Chapter.[1]

Since the first two clustering algorithms do not produce posterior probabilities, we employ a simple procedure to transform the original output g_{ni} for instance n and cluster i into the probability q_{ni}. For k-means, we estimate the posterior probabilities of cluster

[1] k-means and GMM are sensitive to the initialisation of centroids and components, respectively. We run these algorithms 5 times and choose the result with the least intra-cluster variance.

Algorithm 1 ClusterReg algorithm with RBFN.

Input: Training set $\mathbf{X} = \mathbf{L} \cup \mathbf{U}$, where $\mathbf{L} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{L}$, $\mathbf{U} = \{\mathbf{x}_n\}_{n=L+1}^{L+U}$ and $N = L + U$, often $U \gg L$.

Output: Posterior class probabilities $\mathbf{f}$.

1: Produce matrix $\mathbf{Q}$ of cluster probabilities with a clustering algorithm.

2: Compute pairwise penalties with

$$\gamma(\mathbf{q}_n, \mathbf{q}_k) = \sin\left(\frac{\pi}{2}\left(s(\mathbf{q}_n, \mathbf{q}_k)\right)^{\kappa}\right).$$

3: Compute initial pseudo-labels $\hat{y}_{ni}$ for each instances n and node i in cluster Ψ with

$$\hat{y}_{ni} = \frac{\sum_{k\in\Psi} I_{kL} * \gamma(\mathbf{q}_n, \mathbf{q}_k) * y_{ki}}{\sum_{k\in\Psi} I_{kL} * \gamma(\mathbf{q}_n, \mathbf{q}_k)}.$$

4: Find the V nearest neighbours of instance n according to highest penalties $\gamma(\mathbf{q}_n, \mathbf{q}_k)$.

5: Perform pre-training. Update the network weight $\mathbf{w}_j$ with a certain number of Newton-Raphson steps, as in line 10, with fixed $\hat{y}_{ni}$ provided by the initialisation above:

6: **while** termination criterion is not met **do**

7: Update pseudo-labels $\hat{y}_{ni}$ with

$$\hat{y}_{ki} = \begin{cases} y_{ki}, & \text{if } k \text{ is labelled} \\ f_{ki}, & \text{if } k \text{ is unlabelled.} \end{cases}$$

8: Update desired probabilities u_{ni} for unlabelled instances with

$$u_{ni} = \frac{\sum_{k\in\nu(n)} \gamma(\mathbf{q}_k, \mathbf{q}_n)\hat{y}_{ki}}{\sum_{k\in\nu(n)} \gamma(\mathbf{q}_k, \mathbf{q}_n)}.$$

9: Update the network weight $\mathbf{w}_j$ according to the Newton-Raphson method.

$$\Delta\mathbf{w}_j = -\mathbf{H}^{-1} * \left[\frac{\partial\mathcal{L}(\mathbf{f}, \mathbf{y})}{\partial\mathbf{w}_j}\right].$$

10: **end while**

11: **Output:** Trained RBFN

membership with the proportion of each distance from an instance n to each centroid over the sum of all centroid distances from n. Such scaling equation is as follows:

$$q_{ni} = \frac{1 - \left(\frac{g_{ni}}{\sum_{k=1}^{K} g_{nk}}\right)}{K - 1},$$

where g_{ni} is the distance from instance n to the cluster centroid i.

STSC outputs a matrix of K transformed eigenvectors with N dimensions. We generate posterior probabilities with the proportion of each ith transformed eigenvector with respect to all transformed eigenvectors at the nth position. g_{ni} is the nth position of the ith eigenvector, then we have:

$$q_{ni} = \frac{|g_{ni}|}{\sum_{k=1}^{K} |g_{nk}|}.$$

3.4 Cluster-based Regularisation with Multilayer Perceptron

In this Chapter, we propose a single classifier that will be used as a base learner for the ensemble methods in next Chapters. In this sense, we instantiate ClusterReg with another popular algorithm of neural networks: Multilayer Perceptron (MLP) network with a single hidden layer.

We use the Scaled Conjugate Gradient (SCG) algorithm to train the MLP network, since it does not depend on user parameters [Moller, 1993]. In order to apply the SCG algorithm to ClusterReg, we use the gradient (in Equation 3.13) and Hessian matrix (in Equation 3.14) of the proposed loss function.

3.5 Experimental studies

In this Section, we present our experimental analysis to compare the proposed classifier to existing methods in literature and study the performance of the instantiations of ClusterReg with both RBFN (ClusterReg-RBFN) and MLP (ClusterReg-MLP) networks.[1]

We perform experiments with transductive and inductive settings. We show the selection of parameters and discuss results with artificial and real-world datasets.[2] We also present a comparison of computational time with state-of-the-art algorithms.

3.5.1 Methods and parameter tuning

In order to tune the parameters of the methods in our experiments, we performed grid search with predefined parameter values using 10-fold cross-validation and the best result is reported.

For SGT algorithm, we performed a broader search than suggested in Joachims [2003] with all the parameter combinations in the following manner: the number of neighbours was searched in $k \in \{10, 50, 100\}$; the number of first eigenvectors was $d \in \{10, 40, 80, 100\}$ and the error parameter was $c \in \{10^0, 10^2, 10^3, 10^4\}$. Although, in Joachims [2003], the parameter c was set between 3200 and 12800, our preliminary experiments generated better results with our setting.

Similarly to ClusterReg, TSVM possesses the cluster assumption. Therefore, if the dataset has a meaningful cluster structure, we expect ClusterReg to deliver more accurate results. If such a structure is not present, both algorithms may have similar performance. For the parameters in TSVM, we followed Chapelle et al. [2006]. We used a RBF kernel and its width was selected as the median of the pairwise distances between instances [Chapelle et al., 2006]. Unlike in Chapelle et al. [2006], we decided to perform a broader

[1] We denote ClusterReg as the general algorithm, and ClusterReg-MLP and ClusterReg-RBFN as specific instantiations of ClusterReg with MLP and RBFN, respectively.

[2] All datasets were standardized with zero mean and standard deviation of one.

search of the soft margin parameter C (it controls the trade-off between margin size and misclassified training instances) with $C \in \{10^0, 10^1, 10^2, 10^3\}$. In preliminary experiments, lower values of C increased the computational time and reduced the generalisation accuracy of TSVM; and higher values did not improve results. We performed a grid search with all combinations of these parameters and selected the ones with the best result for each dataset.

Since Multi-Class Semi-Supervised Boosting (MCSSB) [Valizadegan et al., 2008] uses all three SSL assumptions, we expect ClusterReg to outperform MCSSB only on datasets where the cluster assumption holds, that is, datasets that possess a clear cluster structure that relates to the class distribution. MCSSB would deliver better results on datasets where there is an unclear or no cluster structure. As its base classifier, we chose SVM since it delivered the best results in our preliminary experiments. We fixed the parameter[1] $C = 10000$. The ratio of the range of distances used for kernel construction was searched in $\sigma \in \{0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.5, 0.8, 1\}$. We set the sample size as a ratio $\{0.1, 0.5, 0.8, 1\}$ of the total number of instances for transductive and inductive contexts. The number of base learners was searched in $\{20, 50\}$.

For RegBoost, Chen and Wang [2011] also suggested a grid search for the best combination of parameters. The number of iterations was tuned with 20 and 50. The number of neighbours was searched in $\{3, 4, 5, 6\}$. The resampling rate in the first iteration was set to 0.1. And the resampling rate in the rest of iterations was searched in $\{0.1, 0.25, 0.5\}$. Following Chen and Wang [2011] and our preliminary experiments, we chose SVM as base classifier.

For ClusterReg, the parameter λ controls the amount of regularisation in the algorithm. Thus, we perform a grid search in $\{0.2, 0.4, 0.6, 0.8, 1\}$, as we do not know whether the data hold the cluster assumption. It is advisable to set this value between 0 and 1.

[1] As demonstrated in Valizadegan et al. [2008] and confirmed by our preliminary experiments, this value should be set to 10000. Lower and higher values did not improve the performance.

Our preliminary experiments showed that the number of neighbours V can be set to 30 for most datasets used in this work. For datasets not larger than 1500 instances, this number might represent a comprehensive search for labels in the neighbourhood of an instance. For a small number of neighbours, ClusterReg may not capture the correct label structure of the neighbourhood. For datasets with more than 1500 instances, V could be set to 2% of the number of instances.

We employed four clustering methods from different clustering approaches: k-means, STSC, GMM and Fuzzy GK. We also selected the clustering algorithm by grid search, since the performance of these algorithms varies depending on the real underlying class structure in the dataset and the type of partition that such methods attempt to find.[1] However, our experiments demonstrated that ClusterReg with STSC usually obtains good generalisation ability for most datasets. This fact might indicate that most of these datasets have clusters with arbitrary shapes that other algorithms might not be able to find. Therefore, we suggest the use of STSC as the clustering algorithm for ClusterReg.

For the number of clusters K, we recommend to set such a parameter to, at least, the number of classes. We intend to generate clusters as compact as possible. If the class structure is not captured by the clustering algorithm, we can increase the number of clusters, so that one class is composed of multiple clusters. ClusterReg will avoid dividing these clusters and, therefore, may be able to produce the decision boundary outside the class. According to our preliminary experiments, we recommend, in general, to set K to two times the number of classes (or greater multiples of the number of classes).

The parameter κ controls the importance of each neighbour according to their similarity (conforming to the clustering algorithm) to an instance. With a larger κ, we relax the cluster assumption by allowing the decision boundary to cut through relatively distant neighbours. It regulates the size of the portion of a cluster that we allow the decision

[1] k-means tends to generate hyperspherical clusters Xu and Wunsch [2005]. GMM and Fuzzy GK are able to obtain elliptical clusters. Whereas STSC is capable of finding clusters with arbitrary shapes.

Clustering algorithm	Grid search with K-means, GMM, STSC or Fuzzy GK
λ	Grid search in $\{0.2, 0.4, 0.6, 0.8, 1\}$
K	Grid search in $\{1, 2, 3, 4\}$ times the number of classes
Centre widths	Grid search with ratio of $\{0.2, 0.5\}$ of the median of pairwise distances

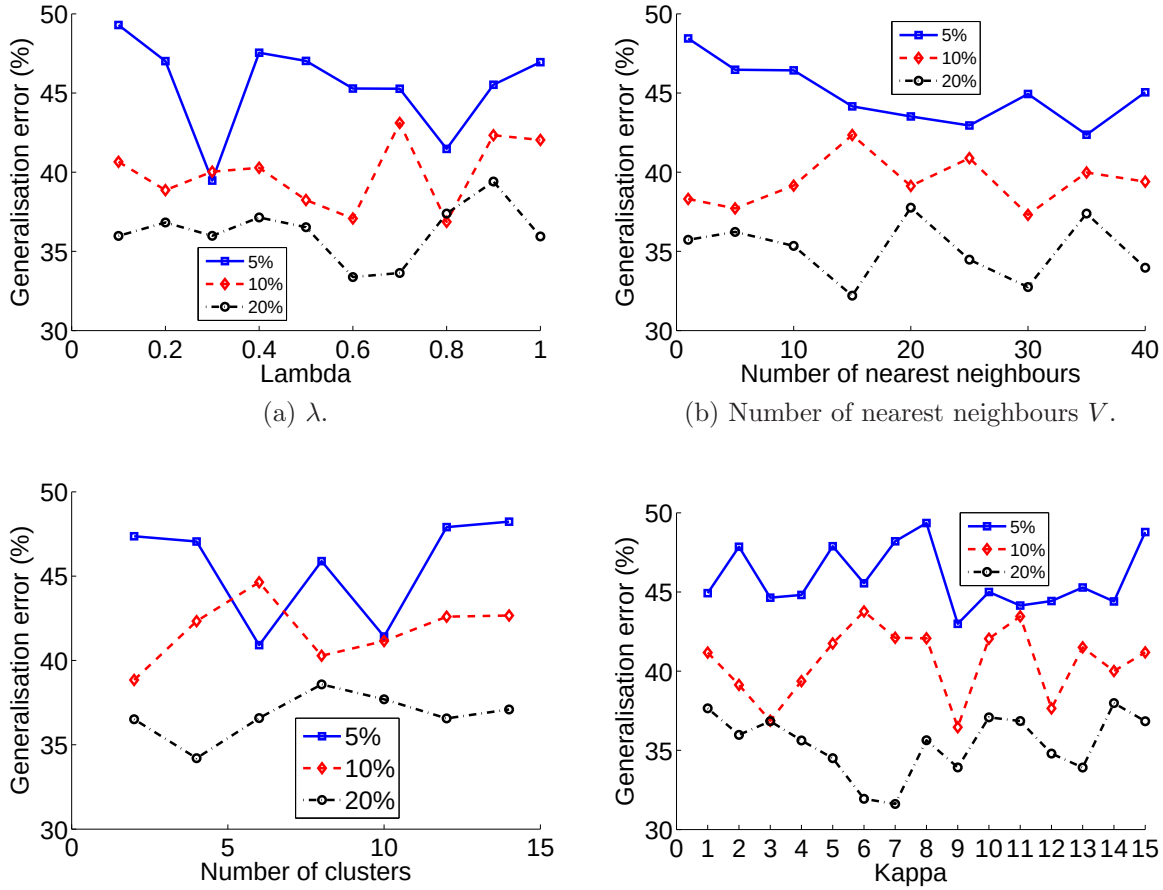
Table 3.1: Summary of tuned parameters for ClusterReg.

boundary to traverse. According to our preliminary experiments, it should be set between 1 and 12 – values in the middle of this range often deliver good performance. The performance of ClusterReg degrades, for all datasets, with values outside this range. Thus, we fixed $\kappa = 5$, although further tuning might produce better results.

For ClusterReg-MLP, specifically, the number of hidden nodes was fixed at 15, as larger numbers did not improve generalisation in our preliminary experiments due to overfitting and smaller values did not produce sufficiently complex networks for our datasets. And the number of epochs in SCG algorithm was 50.

In ClusterReg-RBFN, the centres (hidden nodes) of RBFN coincide with the instances of the entire dataset. Except when the number of instances is larger than 1000, in that case we randomly select 100 instances to be assigned to the centres. The width of centres was calculated as a ratio of the median of all pairwise Euclidean distances between instances. Such a ratio was searched in $\{0.2, 0.5\}$, as different values produced lower generalisation accuracy. The parameter α for weight regularisation was fixed at $\alpha = 0.5$ for both ClusterReg-MLP and ClusterReg-RBFN.

In Figure 3.6, we show the behaviour of the generalisation error for different values of λ, V, K and κ across three different percentages of labelled data in BUPA dataset [Frank and Asuncion, 2010]. We selected only a subset of the values that roughly yielded good performance in Figure 3.6 to be used in our experiments. Thus, Table 3.1 summarises the selection of each tuned parameter in ClusterReg. Further tuning might improve generalisation accuracy.



(a) λ.

(b) Number of nearest neighbours V.

(c) Number of Clusters K.

(d) κ.

Figure 3.6: Generalisation error from 10-fold cross-validation with different values of λ, V, K and κ across three different percentages of labelled data (5%, 10% and 20% in relation to the total number of instances) in BUPA dataset [Frank and Asuncion, 2010].

3.5.2 Transductive setting

We aim to establish the advantages of ClusterReg over classifiers with different assumptions using datasets with different underlying class structures (assumptions). Additionally, we compare ClusterReg-RBFN with ClusterReg-MLP. Thus, in this section, we compare ClusterReg to state-of-the-art algorithms on transductive learning.

In the transductive setting, the test instances are used as unlabelled data during the training phase of a classifier – the generalisation error is the training error on unlabelled

data. Several benchmarks have been designed and used for this setting in Chapelle et al. [2006]. We selected three artificial datasets – g241c, g241d and Digit1 – and four real-world datasets – USPS, COIL, BCI and Text – from Chapelle et al. [2006] to evaluate the proposed algorithm and other state-of-the-art methods using datasets with different SSL assumptions.

Among the artificial datasets, the cluster assumption holds in g241c, as it was designed so that classes correspond to clusters. Whereas g241d was especially built so that the cluster structure is misleading and the manifold assumption does not hold. Digit1 was generated with a low-dimensional manifold embedded into a high-dimensional space, and it does not possess a cluster structure. It is also expected that both cluster and manifold assumptions hold in USPS dataset. Transductive datasets have equally balanced classes and are summarised in Table 3.2. The details of the generation of such datasets can be found in Chapelle et al. [2006, Chapter 21].

Datasets	# classes	# instances	# attributes
g241c	2	1500	214
g241d	2	1500	214
Digit1	2	1500	214
BCI	2	400	114
COIL	6	1500	214
USPS	2	1500	214
Text	2	1500	11960

Table 3.2: Summary of datasets for transductive setting.

Each dataset has 12 subsets of 10 and 100 labelled instances, and the algorithms are run 12 times with 10 and 100 labels and the mean error is reported. We compare ClusterReg-MLP and ClusterReg-RBFN with various existing algorithms reported in Chapelle et al. [2006], Chen and Wang [2011], Zhu et al. [2009]. The details of the tuning procedure for such classifiers can be found in Chapelle et al. [2006, Chapter 21] and Chen and Wang [2011], Zhu et al. [2009]. All results shown in Tables 3.3 and 3.4 were reported

in Chapelle et al. [2006, Chapter 21], except for AdaBoost, ASSEMBLE and RegBoost, which were produced in Chen and Wang [2011]. The results of SAMME, ClusterReg-MLP and ClusterReg-RBFN were obtained in our experiments.

Figure 3.7 shows two-dimensional projections of true classes and predictions from ClusterReg for g241c and g241d with 10 labelled instances. The predictions of ClusterReg, for the first subset of 10 labelled points of g241c and g241d, are presented in Figures 3.7b and 3.7d, respectively.

As performed in Chapelle et al. [2006, Chapter 21], the test sets are fixed and we directly compare the mean of generalisation errors. In order to contextualise ClusterReg and existing algorithms, the results in Tables 3.3 and 3.4 are grouped according to the assumptions of each classifier. Thus, we compare ClusterReg with manifold-based, cluster-based, ensembles and methods with multiple assumptions. Tables 3.3 and 3.4 report the generalisation errors with 10 and 100 labelled points, respectively.

3.5.3 Inductive setting

In contrast to transductive learning, classifiers in inductive learning must be able to predict the label of unseen instances. We selected 20 datasets from the UCI machine learning repository [Frank and Asuncion, 2010]. Table 3.5 summarises the datasets employed.

Since the amount of labelled instances has a great impact on the performance of the classifiers, in this setting, we generate three versions of each dataset. The proportion of labelled data $\frac{L}{N}$ in each version is 5%, 10% and 20%. We transformed these datasets into semi-supervised problems by randomly selecting a stratified sample of labelled instances for each dataset according to the ratio $\frac{L}{N}$. The labelled instances of each dataset are different for each version, so that each version is, in fact, a different problem.

We performed 10-fold cross-validation for all datasets. In order to have the best error estimate as possible, all labels in the test set were available. In real-world datasets, it

Algorithm	g241c	g241d	Digit1	USPS	COIL	BCI	Text
Ensembles and multiple-assumptions algorithms							
AdaBoost	40.12	43.05	28.92	25.57	71.16	47.08	47.42
SAMME	50.09	50.07	50.07	19.98	70.25	50.30	*n/a*
ASSEMBLE	40.62	44.41	23.49	21.77	65.49	48.96	49.13
RegBoost	**38.22**	**42.90**	**17.94**	**17.41**	**65.39**	**46.73**	**34.96**
Manifold-based algorithms							
1NN	44.05	43.22	23.47	19.82	65.91	48.74	39.44
MVU+1NN	48.68	47.28	11.92	14.88	65.72	50.24	39.40
LEM+1NN	47.47	45.34	12.04	19.14	67.96	49.94	40.48
QC+CMN	39.96	46.55	9.80	**13.61**	59.63	50.36	40.79
Discrete Reg.	49.59	49.05	12.64	16.07	63.38	49.51	40.37
SGT	**22.76**	**18.64**	8.92	25.36	*n/a*	49.59	29.02
Laplacian RLS	43.95	45.68	**5.44**	18.99	**54.54**	48.97	**33.68**
CHM (normed)	39.03	43.01	14.86	20.53	*n/a*	**46.90**	*n/a*
Cluster-based algorithms							
SVM	47.32	46.66	30.60	20.03	68.36	49.85	45.37
TSVM	24.71	50.08	17.77	25.20	67.50	49.15	31.21
Cluster-Kernel	48.28	42.05	18.73	19.41	67.32	48.31	42.72
Data-Rep. Reg.	41.25	45.89	12.49	17.96	63.65	50.21	*n/a*
LDS	28.85	50.63	15.63	**15.57**	**61.90**	49.27	**27.15**
ClusterReg-MLP	**16.90**	40.82	12.06	19.42	65.51	**45.36**	40.48
ClusterReg-RBFN	26.94	**27.95**	**10.64**	19.98	69.13	49.19	40.48

Table 3.3: Average of errors (%) of runs with 12 subsets of 10 labelled instances. For all the algorithms, the test sets are fixed. The table reports only the mean of the results, as in Chapelle et al. [2006, Chapter 21]. All results shown in Tables 3.3 and 3.4 were reported in Chapelle et al. [2006, Chapter 21], except for AdaBoost, ASSEMBLE and RegBoost, which were produced in Chen and Wang [2011]. The results of SAMME, ClusterReg-MLP and ClusterReg-RBFN were obtained in our experiments. Bold face denotes the best result among each group of algorithms. And *n/a* denotes the absent results in Chapelle et al. [2006, Chapter 21].

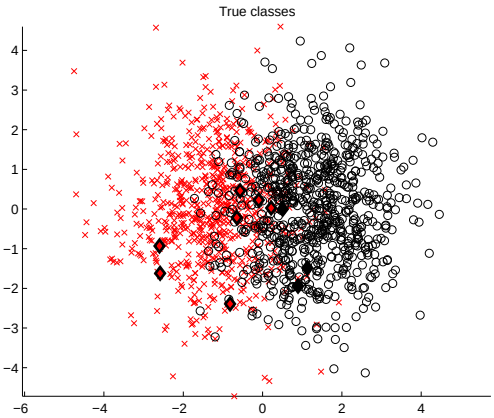
is not possible to know in advance the true class structure and the corresponding SSL assumption that such datasets possess. The success of a classifier will depend on the right matching between their assumptions and the actual class structure present in the data [Chapelle et al., 2006]. Intuitively, if the dataset has a manifold-like structure, it is expected that classifiers that use manifold assumption deliver better performance when compared to other SSC algorithms [Chapelle et al., 2006].

Algorithm	g241c	g241d	Digit1	USPS	COIL	BCI	Text
Ensembles and multiple-assumptions algorithms							
AdaBoost	24.82	26.97	9.09	9.68	22.96	24.02	26.31
SAMME	36.75	38.70	19.55	16.94	53.79	41.64	*n/a*
ASSEMBLE	27.19	27.42	6.71	8.12	21.84	28.75	27.77
RegBoost	**20.54**	**23.56**	**4.58**	**6.31**	**21.78**	**23.69**	**23.25**
Manifold-based algorithms							
1NN	40.28	37.49	6.12	7.64	23.27	44.83	30.77
MVU+1NN	44.05	43.21	3.99	6.09	32.27	47.42	30.74
LEM+1NN	42.14	39.43	**2.52**	6.09	36.49	48.64	30.92
QC+CMN	22.05	28.20	3.15	6.36	10.03	46.22	25.71
Discrete Reg.	43.65	41.65	2.77	**4.68**	**9.61**	47.67	24.00
SGT	**17.41**	**9.11**	2.61	6.80	*n/a*	45.03	**23.09**
Laplacian RLS	24.36	26.46	2.92	**4.68**	11.92	**31.36**	23.57
CHM (normed)	24.82	25.67	3.79	7.65	*n/a*	36.03	*n/a*
Cluster-based algorithms							
SVM	23.11	24.64	5.53	9.75	22.93	34.31	26.45
TSVM	18.46	22.42	6.15	9.77	25.80	**33.25**	24.52
Cluster-Kernel	13.49	4.95	3.79	9.68	21.99	35.17	24.38
Data-Rep. Reg.	20.31	32.82	**2.44**	5.10	**11.46**	47.47	*n/a*
LDS	18.04	28.74	3.46	**4.96**	13.72	43.97	**23.15**
ClusterReg-MLP	**13.38**	**4.36**	3.45	5.25	24.73	33.92	32.09
ClusterReg-RBFN	19.54	17.07	7.20	16.53	36.35	48.11	32.09

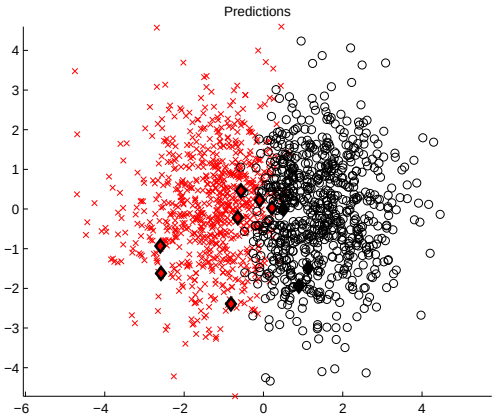
Table 3.4: Average of errors (%) of runs with 12 subsets of 100 labelled instances. For all the algorithms, the test sets are fixed. The table reports only the mean of the results, as in Chapelle et al. [2006, Chapter 21]. All results shown in Tables 3.3 and 3.4 were reported in Chapelle et al. [2006, Chapter 21], except for AdaBoost, ASSEMBLE and RegBoost, which were produced in Chen and Wang [2011]. The results of SAMME, ClusterReg-MLP and ClusterReg-RBFN were obtained in our experiments. Bold face denotes the best result among each group of algorithms. And *n/a* denotes the absent results in Chapelle et al. [2006, Chapter 21].

Thus, we compare our method to state-of-the-art single algorithms with different assumptions (all methods employ the smoothness assumption): one single classifier based on the manifold assumption – SGT; and one based on the cluster assumption – TSVM.

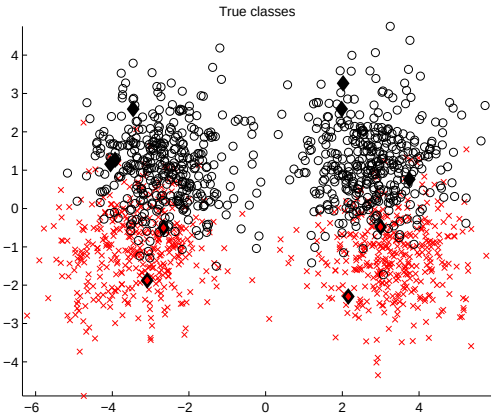
Ensemble-based algorithms with multiple assumptions may deliver higher average performance throughout various datasets [Chen and Wang, 2011], that is, such methods are more likely to deliver better predictions than a specialist algorithm that implements the



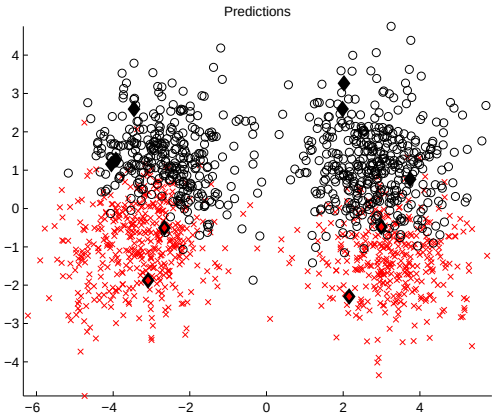
(a) True classes of g241c.



(b) Predictions of ClusterReg with K-means, $K = 2$, $\kappa = 5$, $V = 20$, for g241c.



(c) True classes of g241d.



(d) Predictions of ClusterReg with K-means, $K = 4$, $\kappa = 5$, $V = 20$, for g241d.

Figure 3.7: Two-dimensional projections of true classes and predictions from ClusterReg for g241c and g241d with 10 labelled instances, denoted by dark diamonds.

wrong assumption for a given dataset. Therefore, we compare ClusterReg to algorithms with two ensemble classifiers that use all SSL assumptions – MCSSB and RegBoost.

In order to select the instantiation of ClusterReg that produces the highest generalisation accuracy, we also compare ClusterReg-RBFN and ClusterReg-MLP.

Tables 3.6, 3.7 and 3.8 show the mean and standard deviation of generalisation error of all algorithms for all datasets with 5%, 10% and 20% of labelled data, respectively. We

Datasets	# classes	# instances	# attributes
Australian credit	2	690	16
Balance scale	3	625	6
Bupa	2	345	8
Contraceptive	3	1473	11
Dermatology	6	366	36
Ecoli	5	327	8
German credit	2	1000	26
Glass	6	214	11
Haberman	2	306	5
Heart cleveland	5	303	15
Horse colic	2	368	28
House votes	2	435	18
Ionosphere	2	351	35
Mammographic masses	2	961	7
Pima indians diabetes	2	768	10
SPECT	2	267	24
Vehicle silhouettes	4	846	20
Transfusion	2	748	6
WDBC	2	569	32
Yeast	9	1479	10

Table 3.5: Summary of datasets for inductive setting.

employ a pairwise t-test with 95% of significance level to compare the selected algorithms to ClusterReg-RBFN, as shown in these tables.

3.5.4 Computational time

We also measured the computational time of ClusterReg-RBFN and ClusterReg-MLP. In Figure 3.7, we plot the CPU time of each method used in inductive setting with 5%, 10% and 20% of labelled data, so that we can compare the efficiency of each method under different amounts of labels for each dataset. Each computational time reported is the average time and its standard deviation of the 10-fold cross-validation executions that delivered the error rates already shown in Tables 3.6, 3.7 and 3.8. Specifically, we selected eight datasets where either ClusterReg-MLP or ClusterReg-RBFN obtained superior performance to present computation time, namely: Contraceptive, Vehicle silhouettes, Iono-

Datasets	SGT	TSVM	MCSSB	RegBoost	ClusterReg-MLP	ClusterReg-RBFN
Australian credit	23.91 ± 6.20 ○	18.84 ± 6.90 ○	44.52 ± 4.87 ●	18.15 ± 3.74 ○	21.16 ± 5.08 ○	41.88 ± 17.14
Balance scale	11.20 ± 4.10	13.28 ± 4.23 ●	26.06 ± 5.58 ●	57.30 ± 11.24 ●	16.65 ± 5.67 ●	9.82 ± 1.90
Bupa	44.95 ± 7.89 ●	44.60 ± 6.21 ●	38.91 ± 10.85 ●	47.45 ± 10.83 ●	37.70 ± 6.76 ●	30.50 ± 2.21
Contraceptive	57.65 ± 4.93 ●	52.15 ± 5.67	57.07 ± 4.59 ●	67.76 ± 8.20 ●	52.15 ± 5.02	49.85 ± 1.27
Dermatology	11.50 ± 7.29 ○	7.88 ± 8.38 ○	11.12 ± 5.82 ○	58.24 ± 5.63 ●	8.18 ± 5.99 ○	23.39 ± 7.44
Ecoli	25.09 ± 6.62 ●	15.37 ± 8.69	18.66 ± 5.96	37.62 ± 6.83 ●	17.20 ± 9.02	16.54 ± 4.74
German credit	31.50 ± 2.68 ●	31.30 ± 3.92 ●	31.46 ± 5.59 ●	52.62 ± 21.26 ●	30.00 ± 4.27 ●	23.27 ± 1.91
Glass	38.40 ± 10.80 ○	50.93 ± 7.07 ○	60.31 ± 10.60	77.53 ± 17.87 ●	51.02 ± 8.90 ○	58.40 ± 9.29
Haberman	34.34 ± 7.67 ●	40.20 ± 7.37 ●	33.15 ± 11.00 ●	31.53 ± 17.19 ●	33.32 ± 10.11 ●	16.91 ± 3.06
Heart cleveland	44.89 ± 9.96	45.23 ± 8.50	47.34 ± 15.06	61.06 ± 7.89 ●	42.58 ± 9.07	40.85 ± 3.53
Horse colic	37.21 ± 6.62 ●	42.93 ± 7.37 ●	30.38 ± 10.08	48.44 ± 19.57 ●	33.97 ± 7.47	31.06 ± 5.61
House votes	8.04 ± 5.80	9.41 ± 4.71	61.57 ± 7.24 ●	56.10 ± 12.64 ●	8.74 ± 5.37	7.81 ± 3.06
Ionosphere	35.34 ± 9.77 ●	25.06 ± 9.76 ●	35.64 ± 12.78 ●	50.55 ± 19.80 ●	11.71 ± 6.10	12.97 ± 2.51
M. masses	23.73 ± 2.36 ●	22.17 ± 2.75 ●	46.34 ± 4.80 ●	25.42 ± 4.94 ●	22.26 ± 6.33 ●	12.73 ± 3.19
Pima diabetes	32.28 ± 5.13 ●	36.33 ± 5.12 ●	34.82 ± 4.62 ●	34.21 ± 7.51 ●	35.40 ± 5.67 ●	27.05 ± 1.96
SPECT	28.86 ± 5.07 ●	20.17 ± 6.96 ●	79.51 ± 10.71 ●	31.99 ± 4.27 ●	15.31 ± 7.28 ●	11.09 ± 1.78
V. silhouettes	42.32 ± 2.81 ○	40.30 ± 5.45 ○	49.47 ± 6.09	69.71 ± 5.89 ●	36.64 ± 4.79 ○	52.11 ± 5.51
Transfusion	29.81 ± 19.24	29.16 ± 6.48 ●	23.88 ± 6.03	34.59 ± 23.03 ●	22.87 ± 4.70	19.65 ± 6.21
WDBC	8.44 ± 2.96	11.07 ± 5.22	37.25 ± 5.37 ●	18.93 ± 5.67 ●	4.57 ± 3.90 ○	8.69 ± 1.17
Yeast	49.63 ± 3.31 ○	44.96 ± 4.85 ○	56.58 ± 3.03 ●	68.63 ± 3.68 ●	45.98 ± 5.64 ○	53.35 ± 2.12
Win/Tie/Loss	10/5/5	10/5/5	13/6/1	19/0/1	7/7/6	/

Table 3.6: Mean and standard deviation (%) of 10-fold cross-validation error at 5% of labelled data. ●/○ indicates whether ClusterReg-RBFN is statistically superior/inferior to the compared method, according to pairwise t-test at 95% of significance level. Win/Tie/Loss denotes the number of datasets where ClusterReg-RBFN is significantly superior/comparable/inferior to the compared algorithm.

Datasets	SGT	TSVM	MCSSB	RegBoost	ClusterReg-MLP	ClusterReg-RBFN
Australian credit	13.77 ± 3.43	14.35 ± 3.16	44.58 ± 6.90 ●	13.38 ± 2.54	17.83 ± 3.75 ●	12.76 ± 1.46
Balance scale	10.56 ± 4.92 ●	11.03 ± 3.84 ●	23.40 ± 5.29 ●	46.80 ± 9.48 ●	14.40 ± 3.45 ●	5.47 ± 2.69
Bupa	34.43 ± 8.73	40.83 ± 6.96 ●	43.64 ± 9.92 ●	47.11 ± 12.00 ●	33.30 ± 5.24	33.22 ± 2.43
Contraceptive	55.06 ± 2.73 ●	52.21 ± 3.41 ●	53.35 ± 3.51 ●	61.00 ± 4.59 ●	50.58 ± 3.47 ●	45.71 ± 1.91
Dermatology	1.91 ± 1.32 ○	7.38 ± 6.53 ○	9.97 ± 6.31 ○	69.25 ± 5.95 ●	7.64 ± 6.05 ○	20.12 ± 6.85
Ecoli	19.64 ± 7.79	15.00 ± 5.29	18.59 ± 6.63	35.11 ± 7.51 ●	16.84 ± 6.52	18.90 ± 5.93
German credit	28.10 ± 6.05 ●	34.70 ± 6.31 ●	32.35 ± 5.22 ●	48.28 ± 16.27 ●	31.90 ± 6.19 ●	22.55 ± 1.54
Glass	37.06 ± 12.68	42.64 ± 12.72	52.54 ± 11.18 ●	67.30 ± 12.24 ●	38.44 ± 14.47	43.09 ± 9.44
Haberman	32.39 ± 11.42 ●	37.62 ± 9.99 ●	42.59 ± 10.20 ●	29.91 ± 10.65 ●	29.44 ± 7.96 ●	22.64 ± 5.44
Heart cleveland	38.92 ± 3.77	50.44 ± 6.44 ●	52.73 ± 11.12 ●	72.12 ± 12.89 ●	47.24 ± 6.38 ●	37.81 ± 2.34
Horse colic	32.36 ± 6.59	35.02 ± 6.10	25.35 ± 9.32	57.12 ± 18.39 ●	29.61 ± 7.54	30.10 ± 6.89
House votes	6.19 ± 3.04 ○	9.40 ± 4.69	61.35 ± 8.08 ●	58.12 ± 11.63 ●	8.26 ± 3.40 ○	11.76 ± 1.24
Ionosphere	24.75 ± 8.14 ●	19.10 ± 7.29 ●	35.90 ± 6.75 ●	44.85 ± 15.40 ●	8.27 ± 5.63	10.48 ± 2.08
M. masses	21.96 ± 2.97 ●	21.13 ± 3.28 ●	46.21 ± 6.15 ●	21.11 ± 2.72 ●	19.46 ± 3.55 ●	12.26 ± 1.50
Pima diabetes	31.38 ± 5.23 ●	25.65 ± 4.41	34.84 ± 6.50 ●	32.75 ± 5.10 ●	24.35 ± 3.38 ○	27.90 ± 2.30
SPECT	21.35 ± 8.95 ●	19.10 ± 7.91	79.60 ± 8.61 ●	49.55 ± 32.36 ●	15.34 ± 7.40	15.45 ± 1.49
V. silhouettes	40.08 ± 5.18 ○	32.86 ± 4.66 ○	43.46 ± 7.23 ○	74.44 ± 2.74 ●	31.21 ± 5.99 ○	55.63 ± 3.75
Transfusion	20.98 ± 4.58 ●	29.55 ± 5.45 ●	23.79 ± 6.93 ●	35.07 ± 7.15 ●	21.78 ± 5.03 ●	15.87 ± 1.94
WDBC	8.97 ± 3.19 ●	6.33 ± 3.83 ●	37.37 ± 7.19 ●	13.86 ± 6.47 ●	4.39 ± 3.72	2.77 ± 1.49
Yeast	40.57 ± 3.46 ○	43.48 ± 5.12 ○	53.90 ± 3.70	68.63 ± 2.94 ●	42.94 ± 4.09 ○	52.09 ± 3.50
Win/Tie/Loss	10/6/4	10/7/3	15/3/2	19/1/0	8/7/5	/

Table 3.7: Mean and standard deviation (%) of 10-fold cross-validation error at 10% of labelled data. ●/○ indicates whether ClusterReg-RBFN is statistically superior/inferior to the compared method, according to pairwise t-test at 95% of significance level. Win/Tie/Loss denotes the number of datasets where ClusterReg-RBFN is significantly superior/comparable/inferior to the compared algorithm.

Datasets	SGT	TSVM	MCSSB	RegBoost	ClusterReg-MLP	ClusterReg-RBFN
Australian credit	13.48 ± 3.42 ○	15.07 ± 2.91	44.34 ± 7.04 ●	17.37 ± 5.21	16.38 ± 4.88	16.14 ± 3.12
Balance scale	5.92 ± 2.95 ●	9.11 ± 2.90 ●	23.85 ± 8.12 ●	55.06 ± 5.23 ●	8.64 ± 3.21 ●	3.45 ± 1.08
Bupa	33.03 ± 9.26 ●	35.66 ± 7.53 ●	38.25 ± 10.96 ●	52.16 ± 11.77 ●	31.03 ± 6.69 ●	20.41 ± 5.00
Contraceptive	50.38 ± 2.95 ●	51.39 ± 3.87 ●	54.15 ± 6.38 ●	57.22 ± 6.41 ●	47.05 ± 3.69	45.80 ± 3.09
Dermatology	2.16 ± 2.79 ○	3.01 ± 2.40 ○	6.52 ± 3.99 ○	59.61 ± 8.20 ●	4.10 ± 3.69 ○	14.71 ± 4.92
Ecoli	20.47 ± 4.46	12.53 ± 7.21 ○	17.59 ± 7.73	37.52 ± 13.14 ●	14.95 ± 5.40	18.37 ± 3.34
German credit	26.00 ± 5.42	30.20 ± 4.64 ●	33.83 ± 6.82 ●	37.56 ± 16.99 ●	28.40 ± 3.47 ●	23.90 ± 2.73
Glass	34.72 ± 11.53 ●	43.96 ± 10.31 ●	61.69 ± 12.82 ●	67.06 ± 9.44 ●	39.72 ± 13.23 ●	19.42 ± 6.93
Haberman	24.91 ± 11.19 ●	30.10 ± 8.24 ●	32.57 ± 9.23 ●	25.95 ± 7.57 ●	26.22 ± 11.79 ●	17.47 ± 5.46
Heart cleveland	37.31 ± 6.91	46.87 ± 11.75	52.30 ± 12.83 ●	56.29 ± 16.76 ●	43.55 ± 9.65	41.09 ± 6.02
Horse colic	27.99 ± 5.12 ○	33.94 ± 8.40	40.87 ± 10.84	47.22 ± 14.98 ●	29.33 ± 6.68 ○	37.05 ± 3.09
House votes	6.21 ± 4.18	5.98 ± 3.11	61.29 ± 7.43 ●	50.04 ± 10.84 ●	5.29 ± 3.77	6.87 ± 2.88
Ionosphere	16.54 ± 6.02 ●	13.95 ± 5.43 ●	36.03 ± 10.85 ●	38.46 ± 13.65 ●	10.53 ± 4.64	8.59 ± 1.78
M. masses	23.93 ± 5.44 ●	18.73 ± 5.40 ●	46.45 ± 4.85 ●	46.73 ± 5.50 ●	18.21 ± 6.32 ●	10.52 ± 1.72
Pima diabetes	29.18 ± 7.15 ●	25.13 ± 5.75	34.88 ± 7.24 ●	31.74 ± 5.47 ●	22.91 ± 4.63	22.98 ± 3.42
SPECT	16.89 ± 7.23 ●	18.75 ± 5.98 ●	79.53 ± 5.20 ●	30.85 ± 12.03 ●	18.75 ± 6.78 ●	8.07 ± 2.53
V. silhouettes	31.91 ± 4.50 ○	23.88 ± 4.62 ○	33.47 ± 4.32 ○	72.05 ± 5.28 ●	22.33 ± 3.08 ○	50.83 ± 5.46
Transfusion	20.32 ± 4.41 ●	25.94 ± 4.05 ●	23.78 ± 5.44 ●	26.61 ± 4.43 ●	21.13 ± 4.56 ●	16.63 ± 2.24
WDBC	9.31 ± 4.21 ●	5.45 ± 3.03 ●	37.28 ± 6.42 ●	28.99 ± 5.33 ●	2.82 ± 1.90 ●	1.32 ± 1.14
Yeast	38.95 ± 4.00 ○	42.12 ± 2.16 ○	52.47 ± 4.27	68.65 ± 2.65 ●	40.84 ± 3.80 ○	51.35 ± 2.79
Win/Tie/Loss	11/4/5	11/5/4	15/3/2	19/1/0	9/7/4	/

Table 3.8: Mean and standard deviation (%) of 10-fold cross-validation error at 20% of labelled data. ●/○ indicates whether ClusterReg-RBFN is statistically superior/inferior to the compared method, according to pairwise t-test at 95% of significance level. Win/Tie/Loss denotes the number of datasets where ClusterReg-RBFN is significantly superior/comparable/inferior to the compared algorithm.

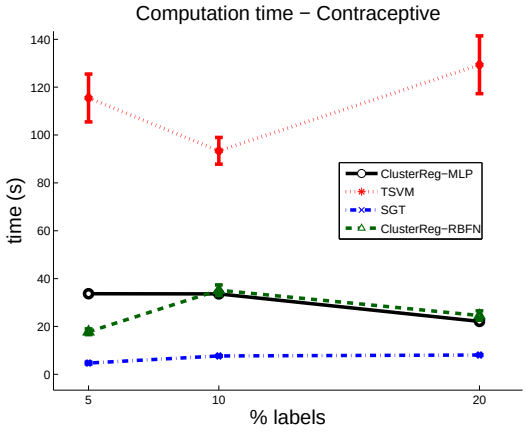
sphere, WDBC, BUPA, Transfusion, SPECT and Yeast (Figures 3.8a–3.7h, respectively). In order to produce a fair comparison, we used only the single classifiers employed in the inductive setting: SGT, TSVM, ClusterReg-MLP and ClusterReg-RBFN.

We measured the CPU time of all algorithms in an Intel Core 2 Quad CPU Q8200 with 2 gigabytes of memory. ClusterReg was implemented using *Matlab*. Its implementation can be further optimised.

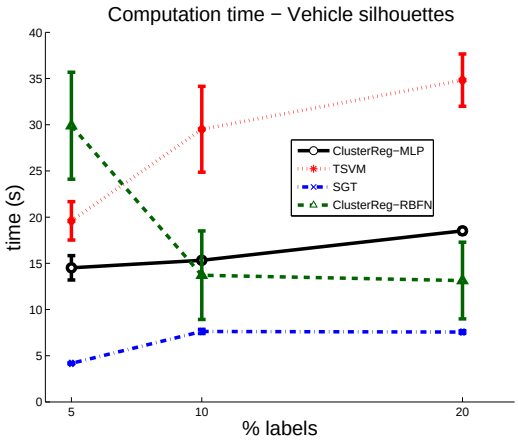
3.6 Discussions

In the transductive experiments, we analyse two types of algorithms: manifold and cluster-based classifiers. Both g241c and g241d present a challenging task for manifold-based algorithms, since they do not satisfy manifold assumption. In contrast, g241c is designed as a suitable problem to cluster-based classifiers, while g241d and Digit1 are challenging datasets to cluster-based algorithms due to either misguiding or absent cluster structure for classes.

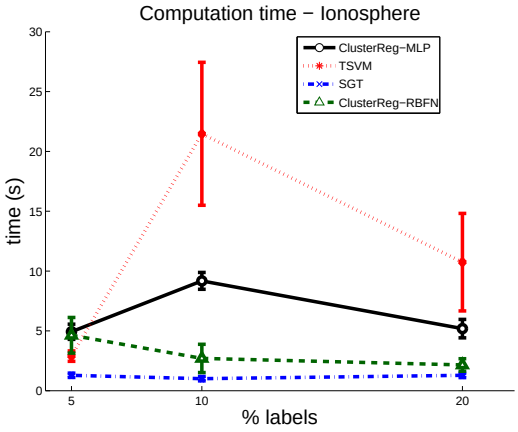
Therefore, both ClusterReg-MLP and ClusterReg-RBFN outperformed the manifold-



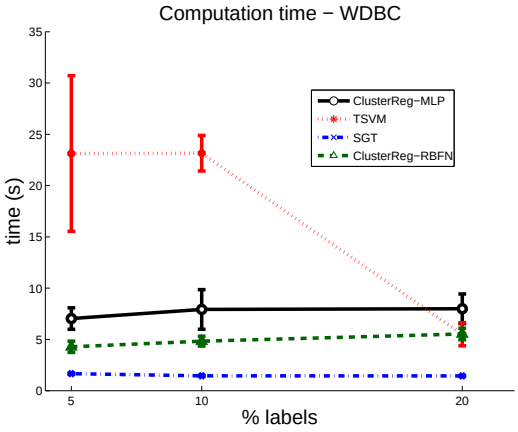
(a) Contraceptive.



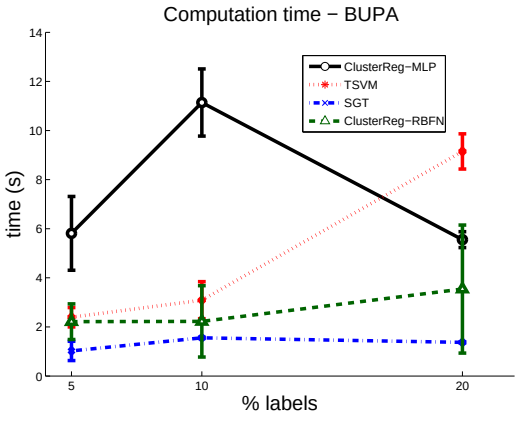
(b) Vehicle silhouettes.



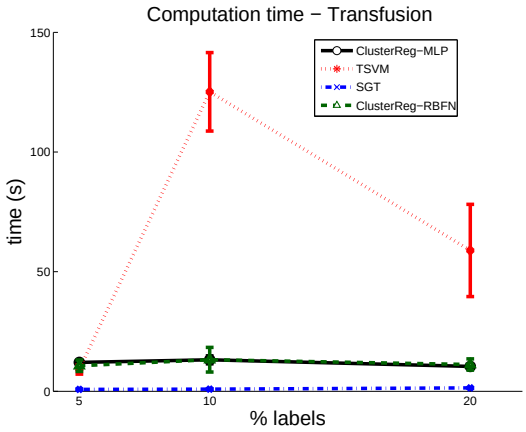
(c) Ionosphere.



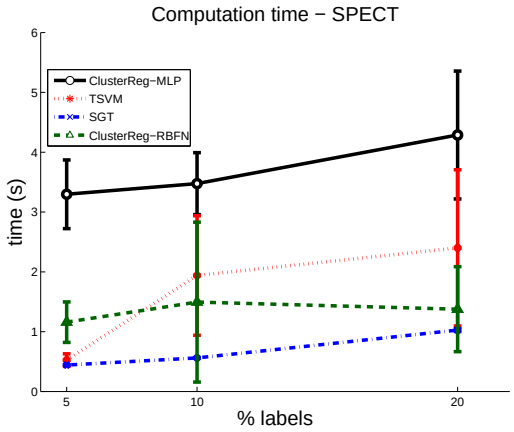
(d) WDBC.



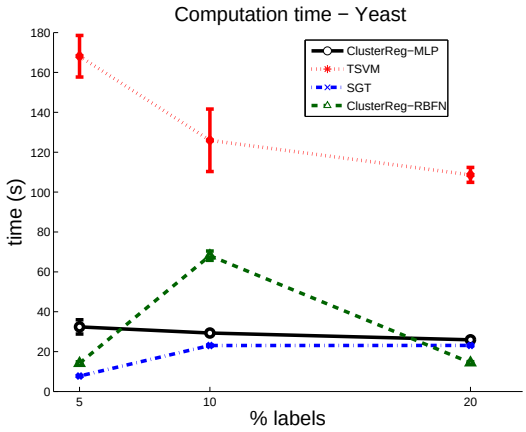
(e) BUPA.



(f) Transfusion.



(g) SPECT.



(h) Yeast.

Figure 3.7: Plots of mean and standard deviation of the computation time of 10-fold cross-validation executions for 5%, 10% and 20% of labelled data, on the datasets where ClusterReg obtained the best results.

based algorithms on g241c and g241d. And, as expected, delivered better generalisation performance than all other cluster-based classifiers on both datasets. The exception was SGT in g241d with 10 labels. This might indicate that, for this case, the graph neighbourhood built by SGT properly represents the underlying class structure. Both ClusterReg-MLP and ClusterReg-RBFN also yielded competitive performance among cluster-based and manifold-based algorithms on the real-world datasets with 10 and 100 labelled instances. Particularly, ClusterReg-MLP produced good performance on BCI when compared to both manifold and cluster-based algorithms. This might indicate that, in this case, ClusterReg-MLP was able to properly use the information of scarce labelled instances.

The g241c dataset possesses a clear cluster structure for classes, where the cluster-based methods should perform sufficiently well. The datasets g241d and Digit1 are tailored to misguide such algorithms. So, when compared to these classifiers, ClusterReg improves the use of the cluster structure and the few labelled instances available to find a suitable decision boundary. It is important to highlight that the results presented in

Tables 3.3 and 3.4 were achieved using the k-means algorithm with 5 replicates with random initialisation. This may indicate that the other methods fail to find the correct gap between classes while a simple clustering algorithm is able to find the clusters. This fact demonstrates how useful clustering techniques can be for semi-supervised classification with the cluster assumption.

It is important to notice that ClusterReg is more robust than TSVM when classes do not correspond to clusters, which is the case for g241d and Digit1, shown in Tables 3.3 and 3.4. This fact may indicate that the proposed classifier is able to exploit the information from the limited labelled data in a more effective way than TSVM, since the unlabelled data do not bring very useful knowledge to cluster-based classifiers.

ClusterReg-MLP produced better generalisation than ClusterReg-RBFN in most datasets. However, for g241d with 10 labelled instances, ClusterReg-RBFN delivered higher predictive accuracy. For such relatively simple datasets, the MLP architecture (the number of weights of a MLP is greater than that of a RBFN due to the weights in the hidden layer of a MLP) was more suitable to learn the class structure of these datasets than the RBFN with locally tuned hidden nodes.

For the inductive setting, Tables 3.6, 3.7 and 3.8 show the generalisation error and statistical test results for the employed algorithms with the presence of 5%, 10% and 20% of labelled data, respectively.

When compared to the ensemble methods, MCSSB and RegBoost, ClusterReg-MLP and ClusterReg-RBFN delivered significantly better results with all amounts of labelled data, as confirmed by pairwise t-test, in Tables 3.6–3.8. Besides being ensemble approaches, these classifiers differ from ClusterReg mainly in the use of SSL assumptions. They use both manifold and cluster assumptions. When only one of them holds and/or the other assumptions are misguiding, a more specialised algorithm, like ClusterReg, might be more effective. Moreover, RegBoost seemed to be affected by the number of

classes: for binary problems, it delivered more competitive generalisation than in multi-class problems. In contrast, ClusterReg is inherently multi-class and did not present such a shortcoming.

ClusterReg-RBFN produced superior results in most datasets when compared to SGT with amounts of labelled data. This fact indicates that ClusterReg is more robust to few labelled instances. We expected to have contrasting results to SGT across the datasets, as the actual structures of real-world datasets are unknown. However, these results suggest that ClusterReg might be able to use labelled instances more effectively than SGT, when the data distribution does not help to infer the correct class distribution.

Similarly to ClusterReg, TSVM also possesses the cluster assumption. However, in the case that the cluster assumption holds, we expect ClusterReg to perform better when very few instances are available. As mentioned before, ClusterReg is more robust than TSVM to the position of the scarce labelled instances in the cluster, as it uses the clustering partition to find the decision boundary. Whereas TSVM seeks the largest margin between classes, which can lead to the wrong decision boundary in the presence of overlapping classes.

In fact, the pairwise t-test confirms our expectations. ClusterReg-RBFN produced significantly superior generalisation performance in most datasets for all amounts of labelled data. And ClusterReg-MLP performed statistically better than TSVM on 3 problems for 20% of labelled data. With 10%, ClusterReg-MLP delivered significantly superior generalisation in 5 cases. For 5% of labelled data, ClusterReg-MLP performed statistically better on 7 datasets. Therefore, when compared to the cluster-based algorithm (TSVM), the proposed method is able to use labelled instances more effectively and it is more robust to overlapping classes and misleading cluster structures with limited labelled points.

Even when the datasets do not follow cluster assumption, the experiments suggest that ClusterReg could still outperform TSVM. This is due to the balance of two terms

in the loss function. When the cluster assumption does not hold, the second term in Equation 3.8 might not be reliable; however the first term may be able to compensate such a misleading term more effectively than TSVM. That is, the experiments indicate that the supervised learning in ClusterReg may be more effective than in TSVM.

In the inductive setting, ClusterReg-RBFN produced superior generalisation than ClusterReg-MLP in most datasets for all amounts of labelled data. Such a fact might indicate that the hidden nodes in RBFN might be able to effectively identify and represent clusters, which facilitates the training of the network weights associated to a given cluster and encourages the regularisation mechanism of ClusterReg in its second phase of training.

In contrast, ClusterReg-MLP delivered better generalisation than that of ClusterReg-RBFN in the transductive setting. Such a result was expected since the MLP networks can identify and use the dimensions that are useful for training the network weights [Bishop, 2006], which is an important benefit due to the nature of these datasets. For example, the datasets g241c and g241d have only two informative dimensions of a total of 241. Digit1 was designed to consist of points close to a low-dimensional manifold embedded into a high-dimensional space. The authors of Digit1 applied a sequence of transformations to the instances. Its data lie close to a five-dimensional manifold. The dimensions of USPS and COIL were masked. Since ClusterReg-MLP possesses weights linked to each dimension of the dataset, the network is able to learn the low-dimensional underlying structures of the data. That is, ClusterReg-MLP is more robust than ClusterReg-RBFN in the presence of irrelevant dimensions. These results indicate that ClusterReg-MLP might produce better generalisation than that of ClusterReg-RBFN for datasets with a manifold structure.

Regarding the computation time, SGT is the least time consuming method. However, since SGT has the manifold assumption, it may not be suitable for datasets where

there is a cluster structure for classes [Chapelle et al., 2006]. Focusing on the context of cluster-based methods, ClusterReg presented a competitive performance when compared to TSVM on most datasets with different amounts of labelled data, as shown in Figure 3.7. Furthermore, we can notice that the difference of execution time across 5%, 10% and 20% of labelled data for ClusterReg is not as high as in TSVM, which might indicate that the computation time of ClusterReg is more stable under different amounts of labels.

All the experiments of ClusterReg were performed with a fixed number of hidden nodes and epochs. The results may be improved with a fine tuning of these parameters. However, the computation time is likely to increase as these parameters change to greater numbers.

Due to the evidences of superior generalisation in Tables 3.6–3.8 and efficiency in Figure 3.7, the proposed method should, therefore, be instantiated with RBFN.

3.7 Conclusions

We proposed a new multi-class semi-supervised classification algorithm that exploits soft partitions produced by a clustering algorithm, and uses such information to regularise the training of a classifier. The transductive experimental setting, with synthetic and real-world datasets, assessed the generalisation ability of the new method in different scenarios where the cluster assumption holds and when it is misguiding. In the inductive case, we used real-world datasets with different ratios of labelled data to evaluate ClusterReg, along with other methods with various approaches to the SSL assumptions.

Both sets of experiments confirmed that the proposed method is able to improve generalisation performance under various scenarios, when the cluster assumption holds. The gain in generalisation accuracy in multi-class datasets was particularly encouraging. Among the reasons for these improvements, we can highlight the ClusterReg's ability to handle the potential presence of overlapping classes and its robustness to the particular

situation of each labelled instance in the clusters.

In this Chapter, we successfully addressed the research question raised in the Section 1.5.1 with the introduction of ClusterReg algorithm. In order to answer the next research questions, ClusterReg-RBFN will be used as the multi-class base learner in the ensemble approaches proposed in the remainder of this Thesis. And its loss function will be employed as the objective function of such approaches.

CHAPTER 4

A Fully Semi-supervised Ensemble for Multi-class Classification

In this Chapter, we address the research question raised in the Section 1.5.2 with the introduction of a new ensemble classifier. Such a method will be used to evaluate the impact of unlabelled instances on ensemble design in SSC.

Among various classification methods, ensemble algorithms have been widely and successfully employed in both supervised and semi-supervised problems. As for all other SSC techniques, the performance of an ensemble is strongly affected by how unlabelled data are used. In this sense, an important question arises: at what level an ensemble should consider using unlabelled data. Such data can be considered either at the ensemble level, the base classifier level, or both. To our knowledge, such an issue has not been addressed until now, as most algorithms use unlabelled data only at the ensemble level and employ supervised base classifiers. In this Chapter, we present a study on the usefulness of employing unlabelled data at both ensemble and base learner levels, comparing it to using such data at the ensemble level only. We propose the Cluster-based Boosting (CBoost) algorithm for multi-class classification. Such a method extends the ClusterReg

algorithm and, unlike other semi-supervised ensembles in the literature, is composed of semi-supervised base classifiers.

CBoost is able to learn from the clustering neighbourhood structure of pseudo-labels assigned by the ensemble, which leads to better generalisation when compared to learning the exact pseudo-labels individually. The proposed ensemble can overcome incorrect pseudo-label assignments used in the training of a new base classifier. CBoost is robust to the position of labelled data within a cluster and is able to deal with the potential presence of overlapping classes. Our experiments confirmed that the proposed method is significantly superior to state-of-the-art ensemble methods and can improve the generalisation ability of single classifiers.

The remainder of this Chapter is organised as follows. Next section presents a review of existing methods. Section 4.3 introduces the proposed algorithm in details. Then, we present the experimental results and discuss our algorithm in Section 4.4. Finally, Section 4.5 discusses our contributions and Section 4.6 presents the conclusions.

4.1 Introduction

Ensemble techniques are widely employed in classification due to the ability of reducing individual errors produced by base classifiers. Therefore, combining a group of suitable classifiers, as an ensemble of classifiers, can improve the generalisation performance when compared with a single classifier in both supervised [Nguyen et al., 2006] and semi-supervised classification [Valizadegan et al., 2008, Chen and Wang, 2011]. This work will investigate ensemble learning in SSC context.

In semi-supervised ensemble learning, an important question arises: at what level of an ensemble one should consider using unlabelled instances. To our knowledge, such an issue was not addressed in literature. Therefore, we present a study about the usefulness of employing unlabelled data at both ensemble and base learner levels, comparing to using

such data at ensemble level only.

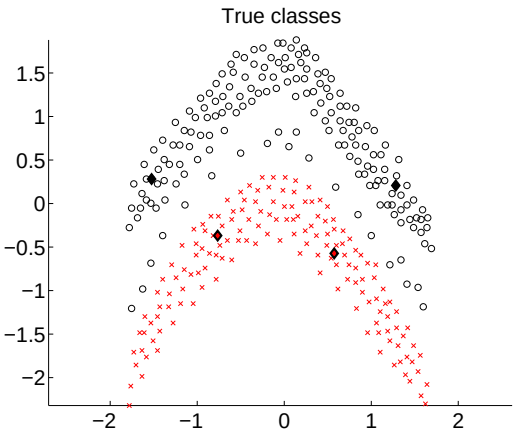
In SSC literature, most ensemble methods optimise a semi-supervised loss function at ensemble level and use supervised base classifiers [Valizadegan et al., 2008, Chen and Wang, 2011]. That is, the unlabelled data is only considered for the ensemble algorithm and supervised base classifiers receive such data as pseudo-labelled instances.[1] However, if the ensemble algorithm (for example, the boosting framework [Friedman, 2001]) does not predict an instance correctly, relying on pseudo-labels for unlabelled instances might reinforce errors in the optimisation process. Such a fact is due to supervised base classifiers learning the exact label that is assigned to a given instance.

In this context, the use of semi-supervised base learners might alleviate such an issue by handling the pseudo-labelled instances as actual unlabelled data. That is, the pseudo-labelled data, as presented to the base learner, would be learnt taking into account the pseudo-labels in the neighbourhood structure, instead of learning each instance and its possibly erroneous pseudo-label individually. Therefore, the optimisation process would not propagate a previous error caused by an incorrect pseudo-label.

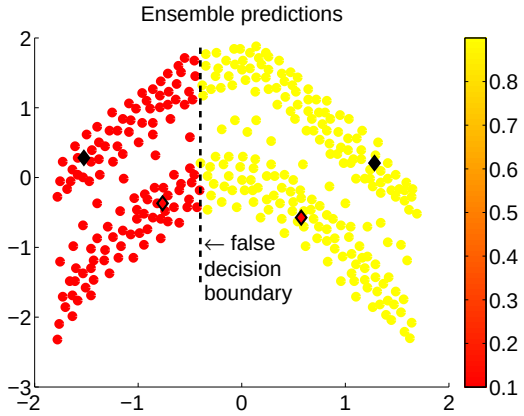
In order to illustrate the impact of incorrect pseudo-labels on the training of base classifiers, Figure 4.1 presents the learning steps of a gradient boosting procedure [Friedman, 2001] for SSC. Figure 4.1a shows an artificial dataset with two half-moon shaped classes, such a dataset has only four labelled instances, denoted as ⧫. In this method, the ensemble assigns pseudo-labels to unlabelled instances that will be used to train the base learner. In Figure 4.1b, we arbitrarily assigned very low quality predictions that will be seen as the current ensemble predictions.[2] Such incorrect predictions follow an erroneous threshold at -0.4 on the horizontal axis, therefore we have a predefined decision boundary (and a large number of incorrect pseudo-labels) in the current optimization

[1] In this work, pseudo-labels are posterior class probabilities artificially assigned to unlabelled instances by some method, indicating that they are not true labels.

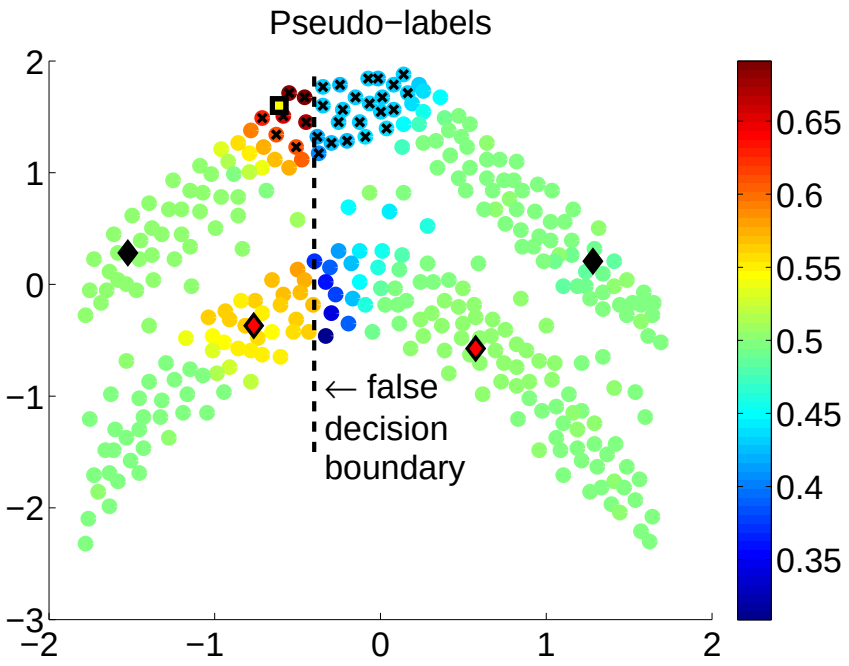
[2] Predictions are the posterior probabilities of classes and the colour scale denotes probabilities for the class regarded as the bottom half-moon.



(a) True classes of two half-moons dataset.

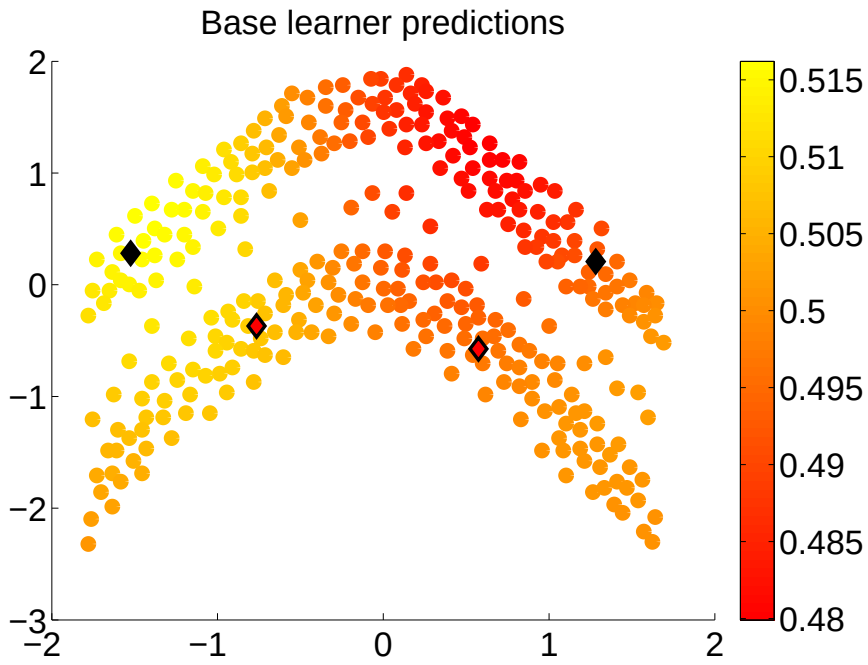


(b) Predefined incorrect decision boundary that traverses classes.

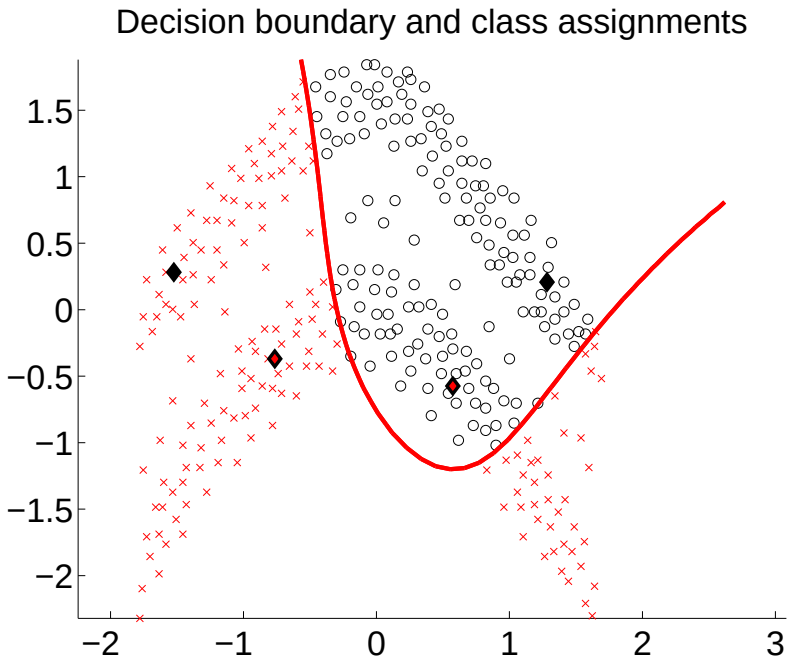


(c) Pseudo-labels generated by Gradient boosting. × represent the neighbours of ■.

Figure 4.1: Steps of ensemble learning using the two half-moon dataset. ♦ represents the labelled instances. Figure 4.1a represents the true class assignments. Figure 4.1b shows a predefined incorrect decision boundary as the ensemble output. And Figure 4.1c denotes the pseudo-labels (posterior class probabilities) generated by Gradient boosting that will be used to train a base learner.



(a) Posterior class probabilities from a supervised base learner trained with pseudo-labels from Figure 4.1c.



(b) Resulting decision boundary.

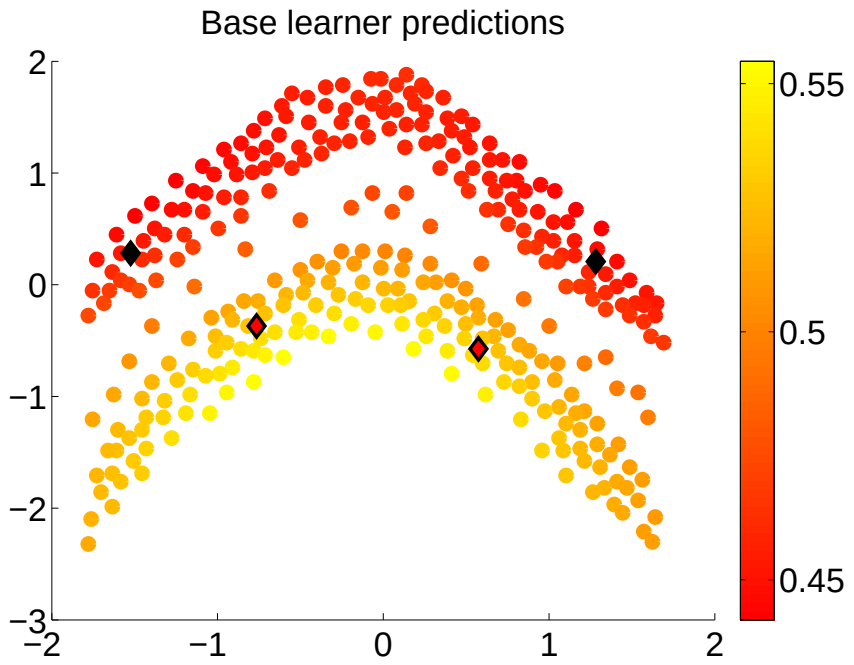
Figure 4.2: Posterior class probabilities (Figure 4.2a) and resulting decision boundary (Figure 4.2b) of a supervised base classifier.

stage.[1] Such a decision boundary was specifically designed to separate the half-moons and the corresponding labelled instances. Gradient boosting is based on the gradient descent algorithm, that is, the pseudo-labels are the actual gradient of the loss function with respect to (w.r.t.) the current ensemble output. Then, Figure 4.1c shows the pseudo-labels (probabilities) assigned by the ensemble algorithm that will be used in the training of a base learner.

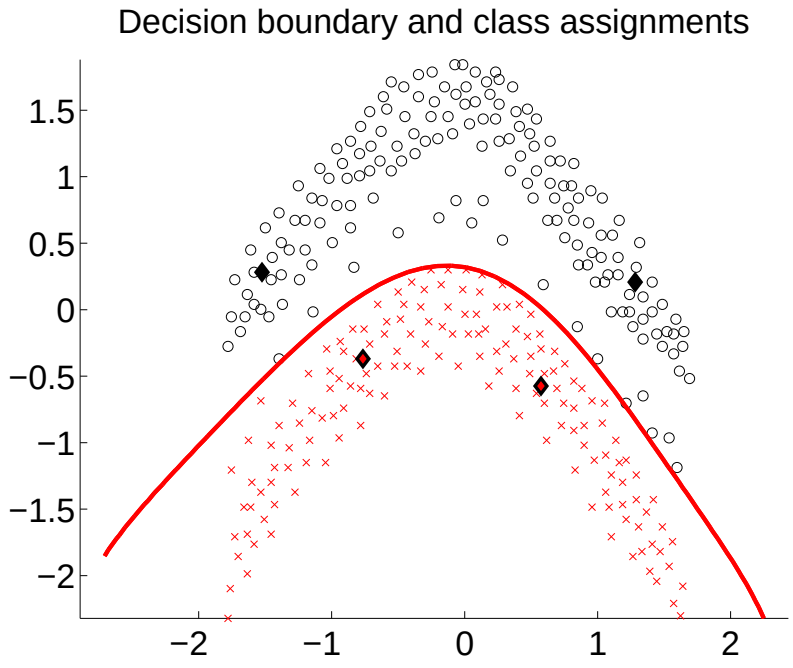
The instance denoted as ■ and its neighbours (marked as ×) should belong to the same class (that is, top half-moon), however these instances were assigned with different pseudo-labels, which is a consequence of the false decision boundary that we designed. Figure 4.2a depicts the class probabilities and Figure 4.2b shows the decision boundary delivered by a supervised base learner. As expected, a supervised base classifier will learn exactly the labels that are presented to it. Therefore, the base classifier will learn an incorrect decision boundary and will propagate the errors to the remainder of the ensemble optimisation process.[2] Therefore, in such cases, using a supervised base learner can degrade the generalisation performance of the ensemble.

In contrast, semi-supervised base classifiers based on cluster assumption will address the incorrect pseudo-label according to its situation in the dataset. If the instance is in a high-density region, the pseudo-labels of its neighbours should be shared with such a point. That is, the distribution of pseudo-labels would be employed to assess whether two instances should belong to the same class. Therefore, semi-supervised base learners can alleviate the problem of incorrect decision boundary (especially when it traverses high-density regions) by considering the distribution of unlabelled data, instead of using only the pseudo-label assigned to an instance. In this sense, we would have more reliable use of pseudo-labels than learning the exact pseudo-label of each instance. This fact can be demonstrated in Figure 4.3, where we show how a semi-supervised algorithm would

[1] This situation can occur in the boosting training stage as we observed in the preliminary experiments.
[2] In gradient boosting, the ensemble output is a weighted sum of every base classifier predictions.



(a) Posterior class probabilities from a semi-supervised base learner trained with pseudo-labels from Figure 4.1c.



(b) Resulting decision boundary.

Figure 4.3: Posterior class probabilities (Figure 4.3a) and resulting decision boundary (Figure 4.3b) of a semi-supervised base classifier.

perform when given such incorrect pseudo-labels presented in Figure 4.1c. In particular, Figure 4.3a depicts the class probabilities and Figure 4.3b shows the decision boundary delivered by a semi-supervised base learner. As expected, the base learner was able to overcome incorrect pseudo-labels (wrong class distribution provided by the ensemble), in this case, by avoiding the decision boundary to be generated in a high-density region (traversing the half-moons). In Figure 4.1c, the point ■ has neighbours (denoted as ×) on both classes (according to the misleading ensemble pseudo-labels). In order to learn from that specific instance, a semi-supervised base classifier will consider the pseudo-labels of its neighbours by using a weighted average of labels and will avoid assigning different labels to instances in that high-density region. The outcome is a better decision boundary, as shown in Figure 4.3b, which will improve the final ensemble predictions.

In this work, we propose a boosting ensemble method for multi-class SSC based on cluster regularisation: Cluster-based Boosting (CBoost). We employ a semi-supervised loss function that incorporates the cluster assumption. We selected the loss function introduced in Chapter 3, since it is able to effectively avoid decision boundaries in high-density regions and it is robust to the position of the few labelled instances in a given cluster in a multi-class context. These facts are due to the use of clustering algorithms as a component in a regularisation mechanism. The clustering method is employed to find high-density regions, allowing CBoost to define a neighbourhood and to assign penalties for unlabelled instances.

We selected the gradient boosting framework [Friedman, 2001], since the algorithm is relatively fast [Sun and Yao, 2010], produces highly robust ensemble classifiers and its instantiation is straightforward (it is based on steepest descent method) [Friedman, 2001].

And for base learners, we instantiated ClusterReg with Radial Basis Functions Networks (RBFN). We selected RBFN networks due to its efficiency [Nabney, 1999]. We employed the Iteratively Reweighted Least Squares (IRLS) to train the weights of the

networks [Bishop, 2006], since an iterative algorithm is necessary for such a non-linear multi-class optimisation [Bishop, 2006].

By employing ensemble learning, we expect to obtain a more robust algorithm, when compared to a single classifier, which is capable of overcoming the aforementioned issues with pseudo-labelling of unlabelled instances in boosting methods. In this sense, we aim to investigate the usefulness of employing semi-supervised base classifiers in semi-supervised ensemble learning.

CBoost has the following advantages. (i) It inherits the robustness of ClusterReg to overlapping classes and to the position of the few labelled instances in a given cluster when the cluster assumption holds. (ii) Since there might be many valid decision boundaries that do not divide clusters, CBoost relies, differently from ClusterReg, on an effective combination of various classifiers to generate an improved decision boundary. (iii) It is designed for multi-class problems, so that it does not depend on decomposition techniques (such as *one-vs-all* or *one-vs-one*). (iv) Both the ensemble algorithm and base classifiers optimise a semi-supervised loss function (introduced in Chapter 3), and the base classifier will also consider the neighbourhood of an instance when learning its pseudo-label, so that the base learner may be able to overcome potential errors in pseudo-labels.

4.2 Background

Combining several suitable classifiers, as an ensemble of classifiers, can enhance the generalisation performance of the entire group when compared to a single classifier [Nguyen et al., 2006]. In this section, we review relevant SSC ensemble methods. Particularly, MCSSB and RegBoost will be used in our experimental analysis.

MCSSB [Valizadegan et al., 2008] performs multi-class classification. Such a method combines the similarity information among the instances with the classifier predictions to obtain more reliable pseudo-labels. It is a graph-based ensemble approach. Its objective

function has the smoothness, manifold and cluster assumptions. And it uses supervised base learners.

SSMB, ASSEMBLE and the algorithm in Zheng et al. [2009] are designed for binary classification, therefore they depend on suboptimal decomposition methods to deal with multi-class problems. Since such methods, along with MCSSB and Song et al. [2011], attempt to find the largest margin between classes, they might be sensitive to overlapping classes and to the position of labelled data in high-density regions. SSMB and ASSEMBLE do not handle semi-supervised assumptions explicitly [Chen and Wang, 2011].

RegBoost [Chen and Wang, 2011] employs three semi-supervised assumptions in its boosting algorithm. RegBoost uses a kernel density estimation approach, which penalises the classifier if it does not assign the same label to a pair of neighbour instances in a high-density region, to implement cluster assumption. However, if overlapping high-density regions are present RegBoost might not establish a good separation between these regions. Moreover, this algorithm is designed only for binary classification. As mentioned before, a decomposition technique, such as *one-vs-all* [Valizadegan et al., 2008], can be employed to extend the algorithm to multi-class problems. However, as expected, our experiments showed that RegBoost delivers inferior results when applied to multi-class real-world datasets.

The aforementioned methods and the ensembles described in Section 2.6 may generate classifiers that can be very certain about the class of unlabelled points, even though these unlabelled points are misclassified. Moreover, except for MCSSB these ensembles are not specifically designed for multi-class problems, depending on decomposition techniques that do not exploit the fact that each example is only assigned to one class. And adapting semi-supervised binary classifiers to multi-class context involves tackling imbalanced classification and different output scales of different binary classifiers [Valizadegan et al., 2008].

These methods employ, directly or indirectly, the cluster assumption and they attempt to find the largest margin between classes, then they might be sensitive to overlapping classes and to the position of labelled data in high-density regions, as shown in the previous Chapter. Moreover, except for SSMB[1], these algorithms rely on supervised base classifiers. In this sense, the base method will learn the pseudo-labels that are presented to it. In case of an incorrect pseudo-label occurs, this error will be reinforced by the base classifier and, therefore, will degrade the entire ensemble generalisation performance.

In order to overcome such drawbacks, we extend the framework introduced in Chapter 3 and propose a multi-class boosting algorithm based on cluster regularisation (CBoost) with semi-supervised base classifiers.

4.3 Cluster-based Boosting algorithm

In this section, we introduce the CBoost algorithm. We present the gradient boosting framework and a general description of our algorithm. Later, we describe the cluster-based loss function introduced in Chapter 3 and instantiate the gradient boosting framework. Finally, we introduce an instantiation of ClusterReg for RBFN to work as base classifiers.

CBoost presents a robust ensemble method for multi-class problems with overlapping classes, where cluster assumption holds. It provides an effective regularisation technique based on clustering partitions that penalises instances with different predictions in the same cluster. Such a framework uses the posterior probabilities generated by a clustering algorithm to define neighbourhood and pairwise penalisation for every instance. CBoost regards the structure arising from the clustering algorithm as a soft partition. Each instance is assigned a probability of belonging to a given cluster, unlike hard partition where clusters are strictly disjoint. By using soft partitions (also known as soft clustering),

[1]SSMB uses Mixture models as semi-supervised base learners, which does not consider the structure of unlabelled by only enlarging the pseudo-margin, as in ASSEMBLE. This technique might reinforce errors during training.

we can address uncertain instances (likely in low density region, that is, in the border of clusters) differently from the more confident ones (likely in the most dense region of clusters).

4.3.1 Gradient boosting

Gradient boosting is a machine learning technique for regression and classification. It produces a predictive machine in the form of an ensemble of base learners. This algorithm trains an ensemble in a greedy stage-wise fashion with steepest descent minimisation. It allows the optimization of an arbitrary differentiable loss function.

Such a framework was originally designed for regression. The current ensemble Z^t at iteration t is a linear combination of base learners z. Each base learner is trained with the residuals of the direction of the negative gradient $r_n = -\left[\frac{\partial \mathcal{L}(Z_n^{t-1}, y_n)}{\partial Z_n^{t-1}}\right]$ of a loss function $\mathcal{L}$. The multiplier β^t is the result of a line-search $\beta^t = \text{argmin}_\beta \sum_i \mathcal{L}(Z_n^{t-1} + \beta^t z_n, y)$ along the direction of the new base learner. Base learners are added to the ensemble proportionally to β^t, with the rule $Z_n^t = Z_n^{t-1} + \eta\beta^t z_n$, where η is a learning rate to avoid overfitting. In this sense, at each iteration, gradient boosting finds the steepest descent, performs a line-search along that direction and includes a base learner that will further minimise the loss function. Algorithm 2 depicts the original gradient boosting algorithm for regression.

In multi-class classification, the ensemble outputs posterior class probabilities $\mathbf{F} = \{\mathbf{F}_n\}_{n=1}^N$, where $\mathbf{F}_n = \{F_{ni}\}_{i=1}^C$ and $\sum_{i=1}^C F_{ni} = 1$, which is a transformation of the linear combination Z, as in Equation 4.3. The residual r_n should be transformed into posterior probabilities $\tilde{\mathbf{y}}_n$. And the multiplier β^t becomes a vector with a weight associated to each class. Next Sections will present a version of this method for multi-class classification.

Algorithm 2 Original gradient boosting for regression.

Input: Training set $\{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$, number of iterations T and learning rate η.
Output: Predicted targets Z^t.

1: Initialise the ensemble with a constant $Z^0 = 0$.
2: **for** $t = 1$ to T, $n = 1$ to N and $j = 1$ to C **do**
3: Find residuals of $\mathcal{L}$ w.r.t. Z_n with rule

$$r_n = -\left[\frac{\partial \mathcal{L}(Z_n^{t-1}, y_n)}{\partial Z_n^{t-1}}\right]$$

4: Fit a base learner z_n to r_n
5: Compute multiplier β^t by solving

$$\beta^t = \text{argmin}_\beta \sum_i \mathcal{L}(Z_n^{t-1} + \beta^t z_n, y)$$

6: Update the ensemble $Z_n^t = Z_n^{t-1} + \eta\beta^t z_n$
7: **end for**

4.3.2 General architecture and notations

CBoost is an extension of ClusterReg to ensemble learning. Our proposed algorithm consists of training a combination Z, where $\mathbf{Z} = \{\mathbf{Z}_n\}_{n=1}^{N}$, and $\mathbf{Z}_n = \{Z_{ni}\}_{i=1}^{C}$, of multiple base learners using steepest gradient descent in order to perform predictions in the form of posterior probabilities $\mathbf{F}_n$. In this method, each new ensemble member contributes to improving the current ensemble predictions by learning the direction of the functional gradient that the ensemble is minimising. This algorithm is able to produce a better decision boundary than a single classifier [Friedman, 2001].

Each new base classifier, denoted as $\mathbf{f} = \{\mathbf{f}_n\}_{n=1}^{N}$, $\mathbf{f}_n = \{f_{ni}\}_{i=1}^{C}$ and $\sum_{i=1}^{C} f_{ni} = 1$, is trained with the labels, $\tilde{\mathbf{y}} = \{\tilde{\mathbf{y}}_n\}_{n=1}^{N}$ and $\tilde{\mathbf{y}}_n = \{\tilde{y}_{ni}\}_{i=1}^{C}$, that correspond to the direction of the steepest descent of the loss function $\mathcal{L}$. The value $\tilde{\mathbf{y}}_n$ is defined in Equation 4.10. We use the loss function $\mathcal{L}$ presented in Section 3.2. Since we are using ClusterReg as the base learner, the quantity $\tilde{\mathbf{y}}$ satisfy $\sum_{i=1}^{C} \tilde{y}_{ni} = 1$.

In order to learn unlabelled instances, ClusterReg consider the values $\tilde{\mathbf{y}}$ as fixed

pseudo-labels. That is, these quantities remain the same for all iterations of ClusterReg. Unlike Equation 3.1, where labels change at each iteration, the estimated desired output $\mathbf{u}_n$ for ClusterReg is fixed throughout the training of a new base learner. A weighted average of the available $\hat{\mathbf{y}}_n$ is used to obtain the estimated desired output $\mathbf{u}_n$, as in Equation 4.1.

$$u_{ni} = \frac{\sum_{k \in \nu(n)} \gamma(\mathbf{q}_k, \mathbf{q}_n) \tilde{y}_{ki}}{\sum_{k \in \nu(n)} \gamma(\mathbf{q}_k, \mathbf{q}_n)}, \tag{4.1}$$

It is important to notice that pseudo-labels are posterior probabilities assigned to unlabelled instances. Both $\tilde{\mathbf{y}}_n$ and $\hat{\mathbf{y}}_n$ can be regarded as pseudo-labels for instance n. The value $\tilde{\mathbf{y}}_n$ is a pseudo-label assigned by the steepest descent optimisation in the gradient boosting framework, as in Equation 4.10, and is learned by a base classifier with Equation 4.1. It is used at the base learner level. This values represent the direction of the negative gradient of the loss function, which the base classifier will learn. In this Chapter, the quantity $\hat{\mathbf{y}}_n$ denotes the pseudo-labels of an instance n in the optimisation at the ensemble level. Such a value is defined in Equation 4.7. It is updated at each ensemble iteration. At ensemble level, this value is used to compose the estimated desired output $\mathbf{u}_n$ as in Equation 4.6.

The generated ensemble member f_{ni} is guaranteed to be parallel to the functional gradient, which improves the generalization performance [Friedman, 2001]. That is, f_{ni} is the most highly correlated solution to $-\left[\frac{\partial \mathcal{L}}{\partial \mathbf{Z}}\right]$ over the data distribution.

The training of f with the labels $\tilde{\mathbf{y}}$ assures that each new base classifier provides the correct direction for the minimisation of the loss function. The multiplier $\boldsymbol{\beta} = \{\beta_i\}_{i=1}^{C}$ weights the importance of $\mathbf{f}$ to the ensemble. The weight $\boldsymbol{\beta}$ is optimised with a linear search as in Equation 4.4.

The general architecture of CBoost is depicted in Figure 4.4 and its steps are as follows.

1. Extract matrix of posterior probabilities generated by a clustering algorithm.

2. Pairwise penalty is calculated according to the output of a clustering algorithm.

3. Initialisation procedure assigns the initial pseudo-labels to unlabelled instances according to the labels and penalty values associated with each labelled instances of each cluster.

4. Penalty values are employed to find the nearest neighbours of each instance. The neighbourhood of a given instance is defined as those instances with the highest penalty values relative to such an instance.

5. With the initial pseudo-labels, penalty values and nearest neighbours at hand, the training of the first base classifier is performed, which is regarded as the initial ensemble. A number of semi-supervised base classifiers are trained with the pseudo-labels produced by the ensemble algorithm.

6. The ensemble method combines all trained base classifiers to form the current ensemble predictions and updates the pseudo-labels for the training of a new base classifier.

We present the details of these steps in the following sections.

4.3.3 Gradient boosting for multi-class classification

Gradient boosting is a general gradient descent framework suitable for minimising a loss function $\mathcal{L}(\mathbf{F}_n, \mathbf{y}_n)$, where $\mathbf{F}_n$ denotes posterior class probabilities for instance n generated by the ensemble. Such a framework demonstrated competitive, highly robust and straightforward instantiations of gradient boosting for both regression and classification [Friedman, 2001]. Due to such characteristics, we selected this framework to train the proposed ensemble method.

Such a procedure starts by assigning a constant to the initial linear combination $\mathbf{Z}^0$, where $\mathbf{Z}_n = \{Z_{ni}\}_{i=1}^{C}$ is the linear output vector of the ensemble for instance n. At

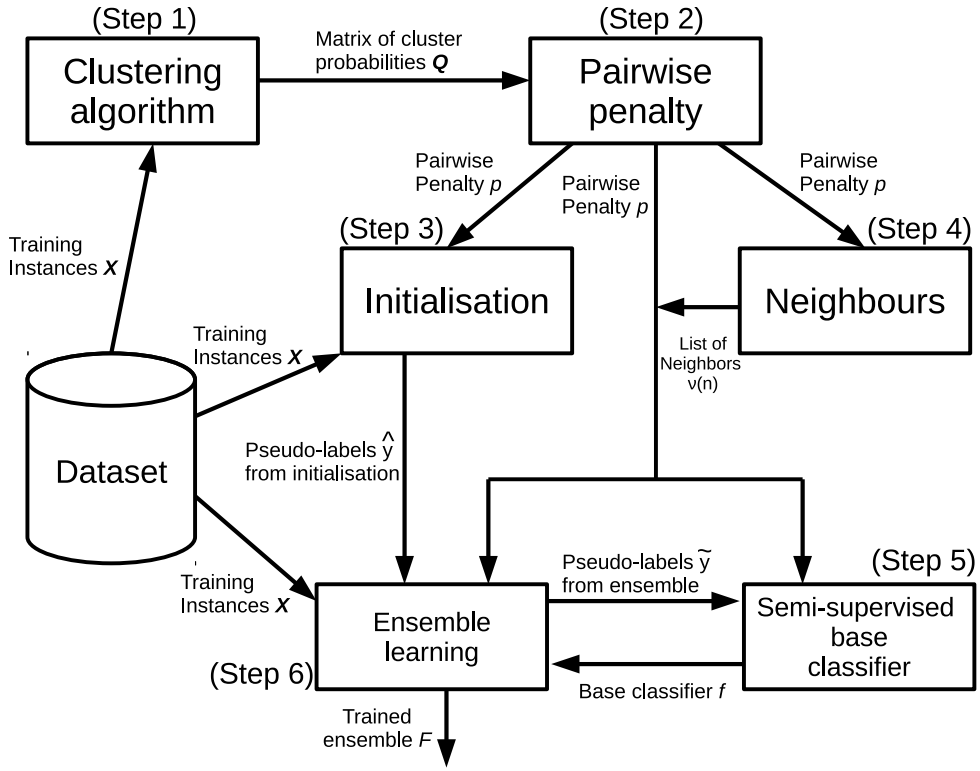


Figure 4.4: CBoost's architecture.

each iteration, pseudo-labels (current residuals) of instance n and class j are assigned as opposite direction of gradient of the loss function w.r.t. the linear output Z^t_{nj} of the current ensemble, as shown in Equation 4.2. Superscripts indicate iteration number.

$$r_{nj} = -\left[\frac{\partial \mathcal{L}(\mathbf{F}^{t-1}_n, \mathbf{y}_n)}{\partial Z^{t-1}_{nj}}\right]. \tag{4.2}$$

Since such an ensemble is a multi-class classification algorithm, it is appropriate to use the cross-entropy cost function [Bishop, 2006] and softmax activation function. Then, the ensemble output $\mathbf{F}$ represents posterior class probabilities and F_{ni} is calculated with the softmax function, as in Equation 4.3.

$$F_{ni} = F(Z_{ni}) = \text{softmax}(Z_{ni}) = \frac{\exp(Z_{ni})}{\sum_j^C \exp(Z_{nj})}. \tag{4.3}$$

A new base learner $\mathbf{f}$ is trained with the newly generated pseudo-labels. The base classifier $\mathbf{f}$ is assumed to have the form $f_{ni} = \text{softmax}(z_{ni})$, where z_{ni} is its output for instance n and class i and the softmax function is used as an output activation function to generate posterior class probabilities.

Since the ensemble is a weighted sum of all base classifiers, a weight vector $\boldsymbol{\beta}^t = \{\beta_i^t\}_{i=1}^C$ for the new base learner is calculated as in Equation 4.4.

$$\boldsymbol{\beta}^t = \text{argmin}_{\boldsymbol{\beta}^t} \sum_n^N \sum_i^C \mathcal{L}\left(F\left(Z_{ni}^{t-1} + \beta_i^t * z_{ni}\right), y_{ni}\right) \tag{4.4}$$

The base classifier is included in the current ensemble following the rule in Equation 4.5, where η is a learning rate that might reduce overfitting by diminishing the influence of newly trained base learner on the ensemble.

$$Z_{ni}^t = Z_{ni}^{t-1} + \eta \beta_i^t z_{ni} \tag{4.5}$$

Several greedy steps of gradient descent are performed until a stopping criterion is met, for example: a fixed number of iterations T, increase of training or validation error rates.

4.3.4 Multi-class cluster-based loss function

The loss function introduced in Chapter 3 consists of two terms: supervised loss and cluster regularisation. A loss function measures how predictions and desired output (true labels) differ. For unlabelled data, however, there are no true labels. In this algorithm, we assign labels to unlabelled data according to penalty values and neighbourhood defined through the use of a clustering algorithm. Equation 4.6 denotes the estimated desired output for an unlabelled instance.

$$u_{ni} = \frac{\sum_{k \in \nu(n)} \gamma(\mathbf{q}_k, \mathbf{q}_n) \hat{y}_{ki}}{\sum_{k \in \nu(n)} \gamma(\mathbf{q}_k, \mathbf{q}_n)}, \tag{4.6}$$

where

$$\hat{y}_{ki} = \begin{cases} y_{ki}, & \text{if } k \text{ is labelled} \\ F_{ki}, & \text{if } k \text{ is unlabelled.} \end{cases} \tag{4.7}$$

The current estimate u_{ni} is the probability of class i given instance n. Such estimates are updated at each ensemble iteration. The posterior F_{ki} is the current ensemble output Z_{ki} transformed into class probabilities [Bishop, 2006]. $\hat{y}_{kj}$ is also known as pseudo-label of k. $\nu(n)$ represents the set of nearest neighbours of n. The penalty $\gamma(\mathbf{q}_k, \mathbf{q}_n)$ is calculated according to the partition provided by the cluster algorithm. Basically, if instances n and k are similar (according to the structure of clustering method), a higher penalty will be assigned to such a pair. Then, u_{ni} becomes a weighted average of current pseudo-labels of the neighbourhood of n, which is more reliable than using $\hat{y}_{kj}$ directly.

We use the loss function defined for ClusterReg in Equation 3.8. Then, Equation 4.8 defines multi-class loss function $\mathcal{L}(\mathbf{F}_n, \mathbf{y}_n)$ for CBoost with cross entropy cost function.

$$\mathcal{L}(\mathbf{F}_n, \mathbf{y}_n) = -\sum_{i=1}^{C} \left\{ \frac{I_{nL}}{L} y_{ni} \log\left(F_{ni}\right) + \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} u_{ni} \log\left(F_{ni}\right) \right\}, \tag{4.8}$$

As in ClusterReg, the regularisation term in Equation 4.8 will penalise the classifier if it assigns different labels to similar instances, as denoted by the product $-u_{ni} \log\left(F_{ni}\right)$. Thus, if the algorithm assigns different outputs for two similar instances, penalty and loss will be high, causing high regularisation. On the other hand, if the penalty is low (the instances are not similar according to the clustering algorithm), the assignment of distinct labels for a couple of instances will not have a significant impact on training. Therefore, CBoost also implements the smoothness assumption.

CBoost follows the cluster assumption by using the density information in $\mathbf{Q}$ to regularise the classifier. We add the maximum value of probability vector $\max(\mathbf{q}_n)$ in the regularisation term of the loss function. The probability $\max(\mathbf{q}_n)$ can be interpreted as

an estimate of density of the region of n. It weights the importance of n in the training.

With the k-means algorithm, it is important to highlight that $\max(\mathbf{q}_n)$ is not the unconditional density. In fact, it is the maximum probability of cluster membership. However, this will distinguish instances by grading their membership to a given cluster. In this case, such a function is an estimate of the proximity of an instance to a given cluster, which can be used as an estimate of the difference of density of two points.

The penalty function will regularise the training if the classifier assigns two different labels to the instance to be learned n and its neighbour k. And the regularisation will be even higher if n is in a high density region, according to the clustering algorithm. Therefore, the classifier will avoid generating a decision boundary that divides a cluster.

4.3.5 Multi-class boosting with cluster regularisation

In this section, we present a multi-class gradient descent boosting algorithm with cluster regularisation.

Unlike original gradient boosting, a base classifier trained with original labels is assigned to initial ensemble $\mathbf{Z}^0$. As indicated in our preliminary experiments, this initialisation delivered better results than simply assigning a constant $\mathbf{Z}^0 = 0$. Such a base classifier is trained with labels generated by the initialisation procedure described in Section 4.3.6.

We calculate the gradient of $\mathcal{L}(\mathbf{F}_n^{t-1}, \mathbf{y}_n)$ w.r.t. Z_{nj}^{t-1} to obtain the current residuals r_{nj} for class j that will be used to train a new base classifier $\mathbf{f}$. By performing the training with such residuals, each base classifier learns the opposite direction of the gradient of $\mathcal{L}$. Thus, each new base learner directs the training of the ensemble towards a minimum of $\mathcal{L}$. Such residuals are computed as in Equation 4.9. Obtaining Equation 4.9 is similar to the derivation of Equation 3.13, thus we omit such steps.

$$r_{nj} = -\frac{\partial \mathcal{L}(\mathbf{F}_n^{t-1}, \mathbf{y}_n)}{\partial Z_{nj}^{t-1}} = -\frac{I_{nL}}{L} * (F_{nj}^{t-1} - y_{nj}) - \frac{I_{nU}\lambda \max(\mathbf{q}_n)}{U} * (F_{nj}^{t-1} - u_{nj}) \tag{4.9}$$

We assume that the input y_{nj} of base classifiers are class probabilities (true labels are denoted as value one and other classes as zeros in the probability vector). That is, $0 \leq y_{nj} \leq 1,\ i = 1 \ldots C$ and $\sum_{j=1}^{C} y_{nj} = 1$. Therefore, the residuals r_{nj} should be transformed into the proper target values in probability scale with

$$\tilde{y}_{nj} = \text{softmax}(r_{nj}), \tag{4.10}$$

where the function $\text{softmax}(r_{nj})$ is defined in Equation 4.3. These pseudo-labels $\tilde{\mathbf{y}}$ are used to train a new base learner $\mathbf{f}$.

Since there is no closed form in the line search in Equation 4.4, we use a single Newton-Raphson step to search for multiplier vector $\boldsymbol{\beta}^t = \{\beta_j^t\}_{j=1}^{C}$ [Friedman, 2001], which is initially 0, as shown in Equation 4.11.[1]

$$\beta_j^t = -\mathbf{H}^{-1} * \frac{\partial \mathcal{L}(\mathbf{F}^{t-1}, \mathbf{y})}{\partial \beta_j^t}, \tag{4.11}$$

where the gradient of each instance n, with the chain rule, is[2]

$$\frac{\partial \mathcal{L}(\mathbf{F}_n^{t-1}, \mathbf{y}_n)}{\partial \beta_j^t} = \frac{\partial \mathcal{L}}{\partial F_{ni}^t} * \frac{\partial F_{ni}^t}{\partial Z_{nj}^t} * \frac{\partial Z_{nj}^t}{\partial \beta_j^t}.$$

The first factor on the right-hand side is

$$\frac{\partial \mathcal{L}}{\partial F_i^t} = -\sum_{i=1}^{C} \left\{ \frac{I_{nL}}{L} \frac{y_i}{F_i^t} + \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} \frac{u_i}{F_i^t} \right\}.$$

The second factor becomes:

$$\frac{\partial F_i^t}{\partial Z_j^t} = F_i^t(\delta_{ij} - F_j^t)$$

[1] We suppress the subscript that indicates instance n and iteration number t when the context is clear.
[2] We derive β_j^t w.r.t $\mathbf{F}_n^{t-1}$ since initially $\beta_j^t = 0$ and hence $\mathbf{F}_n^t = \mathbf{F}_n^{t-1}$.

The third factor is

$$\frac{\partial Z_j^t}{\partial \beta_j^t} = \frac{\partial \left(Z_j^{t-1} + \beta_j^t z_j \right)}{\partial \beta_j^t} = z_j.$$

Then, we can write

$$\frac{\partial \mathcal{L}(\mathbf{F}_n^{t-1}, \mathbf{y}_n)}{\partial \beta_j^t} = -\sum_{i=1}^{C} \left\{ \frac{I_{nL}}{L} \frac{y_i}{F_i^t} + \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} \frac{u_i}{F_i^t} \right\} * F_i^t (\delta_{ij} - F_j^t) * z_j$$

$$= -\frac{I_{nL}}{L} \sum_{i=1}^{C} \left\{ \frac{y_i}{F_i^t} * F_i^t (\delta_{ij} - F_j^t) \right\} z_j - \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} \sum_{i=1}^{C} \left\{ \frac{u_i}{F_i^t} * F_i^t (\delta_{ij} - F_j^t) \right\} z_j$$

$$= -\frac{I_{nL}}{L} \left(\sum_{i=1}^{C} y_i \delta_{ij} - \sum_{i=1}^{C} y_i F_j^t \right) z_j - \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} \left(\sum_{i=1}^{C} u_i \delta_{ij} - \sum_{i=1}^{C} u_i F_j^t \right) z_j,$$

since $\sum_{i=1}^{C} y_i = 1$ and $\sum_{i=1}^{C} u_i = 1$, we have:

$$\frac{\partial \mathcal{L}(\mathbf{F}_n^{t-1}, \mathbf{y}_n)}{\partial \beta_j^t} = \frac{I_{nL}}{L} \left(F_{nj}^{t-1} - y_{nj} \right) z_{nj} + \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} \left(F_{nj}^{t-1} - u_{nj} \right) z_{nj}. \tag{4.12}$$

For the Newton-Raphson method, we also calculate the second derivative $\frac{\partial^2 \mathcal{L}}{\partial \beta_j^t \partial \beta_k^t}$ with the chain rule, then we have:

$$\frac{\partial^2 \mathcal{L}(\mathbf{F}^{t-1}, \mathbf{y})}{\partial \beta_j^t \partial \beta_k^t} = \frac{\partial^2 \mathcal{L}}{\partial \beta_j^t \partial F_j^t} * \frac{\partial F_j^t}{\partial Z_k^t} * \frac{\partial Z_k^t}{\partial \beta_k^t}$$

The first factor on the right-hand side is:

$$\frac{\partial^2 \mathcal{L}}{\partial \beta_j^t \partial F_j^t} = \left(\frac{I_{nL}}{L} + \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} \right) z_j,$$

then, the second factor becomes

$$\frac{\partial F_j^t}{\partial Z_k^t} = F_j^t (\delta_{jk} - F_k^t).$$

And the third factor is

$$\frac{\partial Z_k^t}{\partial \beta_k^t} = z_k.$$

Then, the hessian matrix is $\mathbf{H} = [H_{jk}]_{C \times C}$, where

$$H_{jk} = \frac{\partial^2 \mathcal{L}(\mathbf{F}^{t-1}, \mathbf{y})}{\partial \beta_j^t \partial \beta_k^t} = \sum_{n=1}^{N} \left(\frac{I_{nL}}{L} + \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} \right) * F_{nj}^{t-1} (\delta_{jk} - F_{nk}^{t-1}) * z_{nj} z_{nk}. \quad (4.13)$$

The ensemble is updated as Equation 4.5 for each class. The optimisation is terminated according to some stopping criterion, such as an increase of validation error. In order to produce the posterior class probabilities as the ensemble outputs, we apply Equation 4.3. The proposed ensemble technique is summarised in Algorithm 3.

4.3.6 Radial Basis Functions Network as base learner and initialisation procedure

In our proposed method, we use ClusterReg as base learner due to its robustness to overlapping classes and presence of few labelled instances in borders of clusters, as demonstrated in Chapter 3.

As discussed in Section 3.6, we instantiate ClusterReg with RBFN since such networks produce high generalisation accuracy and can be efficient and easily adapted to our method. In fact, the experimental analysis in Section 3.5 confirmed that RBFN was more efficient and delivered higher predictive accuracy than MLP networks. Therefore, we employ the training procedure described in Section 3.3 to train ClusterReg with RBFN.

In order to initialise the ensemble, we train the initial base learner exclusively with the pre-training procedure described in Section 3.2.5. It consists of training the initial base classifier for number of iterations with the pseudo-labels $\hat{y}_{ni}$ delivered by Equation 3.10. During such a training, the pseudo-labels $\hat{y}_{ni}$ are fixed.

At early iterations, this technique helps the ensemble algorithm to start with a better

Algorithm 3 CBoost algorithm.

Input: Training set $\mathbf{X} = \mathbf{L} \cup \mathbf{U}$, where $\mathbf{L} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{L}$, $\mathbf{U} = \{\mathbf{x}_n\}_{n=L+1}^{L+U}$ and $N = L + U$, often $U \gg L$.

Output: Posterior class probabilities $\mathbf{F}^t$.

Fit base learner $f_{nj} = \text{softmax}(z_{nj})$ to initial labels $\hat{y}_{nj}$ assigned by

$$\hat{y}_{nj} = \frac{\sum_{k \in \Psi} I_{kL} * P(\mathbf{q}_n, \mathbf{q}_k) * y_{kj}}{\sum_{k \in \Psi} I_{kL} * P(\mathbf{q}_n, \mathbf{q}_k)}.$$

Assign initial ensemble $Z_{nj}^0 = z_{nj}$ and $F_{nj}^0 = \text{softmax}(Z_{nj}^0)$.

for $t = 1$ to T, $n = 1$ to N and $j = 1$ to C **do**

Update $\hat{y}_{nj}$ using F_{nj}^t with

$$\hat{y}_{kj} = \begin{cases} y_{kj}, & \text{if } k \text{ is labelled} \\ F_{kj}^t, & \text{if } k \text{ is unlabelled.} \end{cases}$$

Update class probabilities for unlabelled instances with

$$u_{ni} = \frac{\sum_{k \in \nu(n)} \gamma(\mathbf{q}_k, \mathbf{q}_n) \hat{y}_{ki}}{\sum_{k \in \nu(n)} \gamma(\mathbf{q}_k, \mathbf{q}_n)}.$$

Find residuals of $\mathcal{L}$ w.r.t. Z_{nj} with rule

$$r_{nj} = -\frac{I_{nL}}{L} * (F_{nj}^{t-1} - y_{nj}) - \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} * (F_{nj}^{t-1} - u_{nj})$$

Calculate pseudo-labels $\tilde{y}_{nj} = \text{softmax}(r_{nj})$

Fit semi-supervised base learner $f_{nj} = \text{softmax}(z_{nj})$ to $\tilde{y}_{nj}$

Find multiplier β_j^t using rule

$$\beta_j^t = -\mathbf{H}^{-1} * \left[\frac{\partial \mathcal{L}(\mathbf{F}^{t-1}, \mathbf{y})}{\partial \beta_j^t} \right]$$

Update the posterior class probabilities with

$$Z_{nj}^t = Z_{nj}^{t-1} + \eta \beta_j^t z_{nj}$$

$$F_{nj}^t = \text{softmax}(Z_{nj}^t)$$

end for

solution for base classifiers, due to more reliable initial pseudo-labels (without such a procedure, pseudo-labels would represent equal probabilities to all classes). As expected, our preliminary experiments showed that such a procedure improves generalisation ability.

4.4 Experimental studies

In this Section, we perform experiments with two settings: transductive and inductive. We show the selection of parameters and present results with artificial and real-world datasets.

4.4.1 Methods and parameter tuning

In order to tune the parameters of the methods in our experiments, we performed grid search with predefined parameter combinations using 10-fold cross-validation and the best result is reported.

Since MCSSB [Valizadegan et al., 2008] uses all three SSL assumptions, we expect CBoost to outperform MCSSB only on datasets where the cluster assumption holds, that is, a meaningful cluster structure is, in fact, present in the data. MCSSB would deliver better results on datasets where there is either an unclear or no cluster structure. As its base classifier, we chose SVM, since it delivered the best results in our preliminary experiments. We fixed the parameter[1] $C = 10000$. The percentage of the range of distances used for kernel construction was searched in $\sigma \in \{0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.5, 0.8, 1\}$. We set sample size $s \in \{0.1, 0.5, 0.8, 1\}$. The number of base learners was 20 and 50. All the parameter combinations were tested and the best result for each dataset is reported.

For RegBoost [Chen and Wang, 2011], the number of iterations was searched with 20 and 50. The number of neighbours was search in $\{3, 4, 5, 6\}$. The resampling rate in the first iteration was set to 0.1. And the resampling rate in the rest of iterations was

[1] As demonstrated in [Valizadegan et al., 2008] and confirmed by our preliminary experiments, this value should be set to 10000. Lower and higher values did not improve the performance.

searched in $\{0.1, 0.25, 0.5\}$. Following the recommendation in Chen and Wang [2011] and our preliminary experiments, we chose SVM as its base classifier.

Besides MCSSB and RegBoost, we compare the proposed CBoost algorithm (denoted as CBoost-Semi) to ClusterReg instantiated with RBFN and CBoost with a supervised RBFN as base learner (referred as CBoost-Sup). The remainder of this Section will describe the parameters of CBoost-Semi. ClusterReg uses the same parameter setting as the base classifiers of CBoost-Semi. We also use the same parameter values for CBoost-Sup, except for the absence of semi-supervised parameters for its base learner: λ, κ, K, V and clustering algorithm.

In CBoost, λ is the regularisation parameter in the algorithm. For ensemble level, we perform a grid search in $\{0.2, 0.4, 0.6, 0.8, 1\}$, as we do not know in advance whether a meaningful cluster structure is, in fact, present in the dataset. We suggest setting this value between 0 and 1 since different values might degrade generalisation performance. In order to provide diversity to the ensemble, λ is uniformly drawn from $[0.2, 1]$ for each base classifier.[1]

As in ClusterReg, the number of neighbours V should be set to 30 for the datasets used in this work. With 30 neighbours, CBoost is likely to perform a comprehensive and reliable search within the neighbourhood of an instance in datasets with less than 1500 instances. CBoost may not capture the correct local label structure with a smaller number of neighbours. For datasets with more than 1500 instances, we can set V to around 2% of the number of instances.

We employed Self-Tuning Spectral Clustering (STSC) [Manor and Perona, 2004] to generate the partition used in cluster-based regularisation, as such a method is able to find clusters with arbitrary shapes. We also recommend the use of STSC as the clustering algorithm for CBoost. For transductive settings we also employed k-means [Xu

[1] Assigning same λ for all base classifiers did not improve generalisation error according to our preliminary experiments.

and Wunsch, 2005] algorithm[1] and performed cross-validation to choose between both algorithms.

The parameter K should be set to, at least, the number of classes. We also tried larger number of clusters (multiples of the number of classes). In the case where class structure is not captured by the clustering algorithm, we can increase the number of clusters, so that one class is composed of multiple clusters. The algorithm will avoid traversing these clusters and may generate a decision boundary that does not divide such a class. We performed a grid search with $\{1, 2, 3, 4\}$ times the number of classes for both experimental settings.

We set κ to 5 throughout all experiments (value in the middle of the range suggested for ClusterReg). Further tuning of this parameter might lead to better results.

For ClusterReg, the centres of RBFN coincide with the instances of the entire dataset. Except when the number of instances are larger than 1000, in that case we randomly select 100 instances to be assigned to the centres. Our preliminary experiments showed that assigning different centre widths to each base classifier delivered better results than using a single value to all base learners. Such a fact is expected since the centre widths have a great impact on RBFN predictive ability and it is important to possess a certain degree of diversity. Thus, centre widths of RBFN are uniformly drawn from between 20% and 80% of the median of all pairwise Euclidean distances between instances. The weight regularisation parameter α is uniformly drawn from $[0.2, 0.5]$ and λ is uniformly drawn from $[0.2, 1]$ for each base classifier.[2]

We fixed the number of base classifiers to 20, η was fixed to 0.5 and the number of IRLS iterations for RBFN was set to 50 (further optimisation on these values can improve results). The remaining parameters for ClusterReg (K and the clustering algorithm) are equal to the respective parameters of the ensemble algorithm. In Table 4.1, we summarise

[1] K-means generates hyperspherical clusters [Xu and Wunsch, 2005].

[2] Ranges for λ, α and centre widths were empirically tested in our preliminary experiments.

Tuned parameters of CBoost	
Clustering algorithm (algorithms used to produce matrix $\mathbf{Q}$)	K-means and STSC
λ (controls the amount of regularisation)	Grid search in $\{0.2, 0.4, 0.6, 0.8, 1\}$
K (number of clusters)	Grid search with $\{1, 2, 3, 4\}$ times the number of classes

Table 4.1: Summary of tuned parameters for CBoost.

the parameters that are tuned in CBoost.

4.4.2 Transductive setting

Firstly, we aim to establish the advantages of ensemble learning over single classifiers. Secondly, we assess the impact of employing semi-supervised base learners in a semi-supervised ensemble. In this section, we compare CBoost-Semi to ClusterReg and CBoost-Sup based on transductive learning.

In the transductive setting, test instances are regarded as unlabelled data and the generalisation error is the training error on unlabelled data. We use the datasets and the setting described in Section 3.5.2 and summarised in Table 3.2 to evaluate CBoost-Semi and CBoost-Sup.

Tables 4.2 and 4.3 present the generalisation errors with 10 and 100 labelled instances, respectively. We compare CBoost-Sup, CBoost-Semi with the algorithms described in Section 3.5.2. Such algorithms are grouped according to their assumptions: manifold-based, cluster-based, ensemble and methods with multiple assumptions. All results shown in Tables 3.3 and 3.4 were reported in Chapelle et al. [2006, Chapter 21], except for AdaBoost, ASSEMBLE and RegBoost, which were produced in Chen and Wang [2011]. The results of SAMME, ClusterReg-MLP and ClusterReg-RBFN were obtained in our experiments.

In order to analyse the improvement of CBoost-Semi when compared to ClusterReg and CBoost-Sup, Figure 4.4 presents box plots of the generalisation performance of such algorithms.

Algorithm	g241c	g241d	Digit1	USPS	COIL	BCI	Text
Manifold-based algorithms							
1NN	44.05	43.22	23.47	19.82	65.91	48.74	39.44
MVU+1NN	48.68	47.28	11.92	14.88	65.72	50.24	39.40
LEM+1NN	47.47	45.34	12.04	19.14	67.96	49.94	40.48
QC+CMN	39.96	46.55	9.80	**13.61**	59.63	50.36	40.79
Discrete Reg.	49.59	49.05	12.64	16.07	63.38	49.51	40.37
SGT	**22.76**	**18.64**	8.92	25.36	*n/a*	49.59	**29.02**
Laplacian RLS	43.95	45.68	**5.44**	18.99	**54.54**	48.97	33.68
CHM (normed)	39.03	43.01	14.86	20.53	*n/a*	**46.90**	*n/a*
Cluster-based algorithms							
SVM	47.32	46.66	30.60	20.03	68.36	49.85	45.37
TSVM	24.71	50.08	17.77	25.20	67.50	49.15	31.21
Cluster-Kernel	48.28	42.05	18.73	19.41	67.32	48.31	42.72
Data-Rep. Reg.	41.25	45.89	12.49	17.96	63.65	50.21	*n/a*
LDS	28.85	50.63	15.63	**15.57**	**61.90**	49.27	**27.15**
ClusterReg (MLP)	**16.90**	40.82	12.06	19.42	65.51	**45.36**	40.48
ClusterReg (RBFN)	26.94	**27.95**	**10.64**	19.98	69.13	49.19	40.48
Ensembles and multiple-assumptions algorithms							
AdaBoost	40.12	43.05	28.92	25.57	71.16	47.08	47.42
SAMME	50.09	50.07	50.07	19.98	70.25	50.30	*n/a*
ASSEMBLE	40.62	44.41	23.49	21.77	65.49	48.96	49.13
RegBoost	38.22	42.90	17.94	**17.41**	65.39	**46.73**	**34.96**
CBoost-Sup	44.65	45.76	15.64	19.98	77.61	47.37	44.49
CBoost-Semi	**22.76**	**23.07**	**14.72**	19.98	**64.33**	48.50	43.77

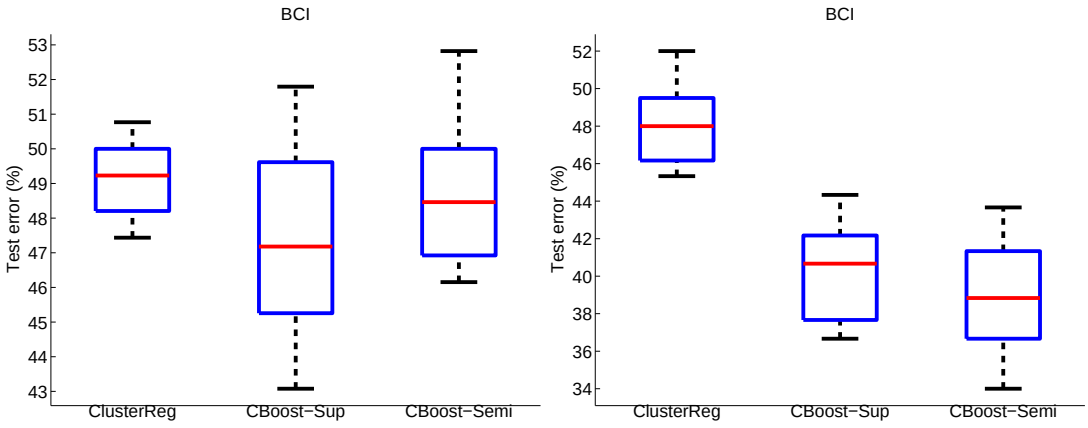
Table 4.2: Average of errors (%) of runs with 12 subsets of 10 labelled instances. For all algorithms, the test sets are fixed. The table reports only the mean of the results, as in Chapelle et al. [2006, Chapter 21]. All results shown in Tables 3.3 and 3.4 were reported in Chapelle et al. [2006, Chapter 21], except for AdaBoost, ASSEMBLE and RegBoost, which were produced in Chen and Wang [2011]. The results of SAMME, ClusterReg-MLP and ClusterReg-RBFN were obtained in our experiments. Bold face denotes the best result among each group of algorithms. And *n/a* denotes the absent results in Chapelle et al. [2006, Chapter 21].

Algorithm	g241c	g241d	Digit1	USPS	COIL	BCI	Text
Manifold-based algorithms							
1NN	40.28	37.49	6.12	7.64	23.27	44.83	30.77
MVU+1NN	44.05	43.21	3.99	6.09	32.27	47.42	30.74
LEM+1NN	42.14	39.43	**2.52**	6.09	36.49	48.64	30.92
QC+CMN	22.05	28.20	3.15	6.36	10.03	46.22	25.71
Discrete Reg.	43.65	41.65	2.77	**4.68**	**9.61**	47.67	24.00
SGT	**17.41**	**9.11**	2.61	6.80	*n/a*	45.03	**23.09**
Laplacian RLS	24.36	26.46	2.92	**4.68**	11.92	**31.36**	23.57
CHM (normed)	24.82	25.67	3.79	7.65	*n/a*	36.03	*n/a*
Cluster-based algorithms							
SVM	23.11	24.64	5.53	9.75	22.93	34.31	26.45
TSVM	18.46	22.42	6.15	9.77	25.80	**33.25**	24.52
Cluster-Kernel	13.49	4.95	3.79	9.68	21.99	35.17	24.38
Data-Rep. Reg.	20.31	32.82	**2.44**	5.10	**11.46**	47.47	*n/a*
LDS	18.04	28.74	3.46	**4.96**	13.72	43.97	**23.15**
ClusterReg (MLP)	**13.38**	**4.36**	3.45	5.25	24.73	33.92	32.09
ClusterReg (RBFN)	19.54	17.07	7.20	16.53	36.35	48.11	32.09
Ensembles and multiple-assumptions algorithms							
AdaBoost	24.82	26.97	9.09	9.68	22.96	24.02	26.31
SAMME	36.75	38.70	19.55	16.94	53.79	41.64	*n/a*
ASSEMBLE	27.19	27.42	6.71	8.12	21.84	28.75	27.77
RegBoost	20.54	23.56	4.58	**6.31**	**21.78**	**23.69**	**23.25**
CBoost-Sup	20.92	28.35	4.87	8.78	63.78	40.25	30.76
CBoost-Semi	**12.71**	**6.99**	**4.34**	7.20	30.67	38.83	25.58

Table 4.3: Average of errors (%) of runs with 12 subsets of 100 labelled instances. For all algorithms, the test sets are fixed. The table reports only the mean of the results, as in Chapelle et al. [2006, Chapter 21]. All results shown in Tables 3.3 and 3.4 were reported in Chapelle et al. [2006, Chapter 21], except for AdaBoost, ASSEMBLE and RegBoost, which were produced in Chen and Wang [2011]. The results of SAMME, ClusterReg-MLP and ClusterReg-RBFN were obtained in our experiments. Bold face denotes the best result among each group of algorithms. And *n/a* denotes the absent results in Chapelle et al. [2006, Chapter 21].

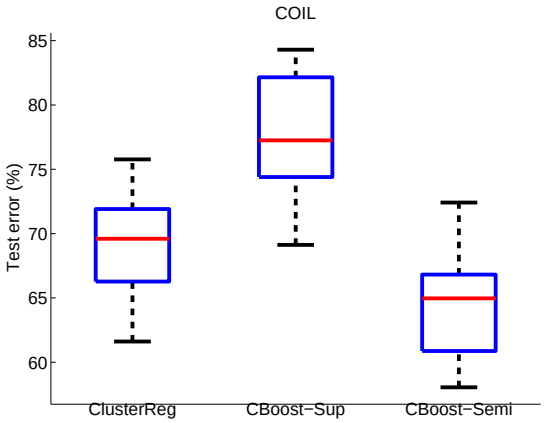
4.4.3 Inductive setting

Inductive learning is the scenario where algorithms can predict the label of unseen instances. We use this setting to evaluate CBoost along with other existing algorithms, namely, MCSSB and RegBoost. We employ the datasets summarised in Table 3.5 and

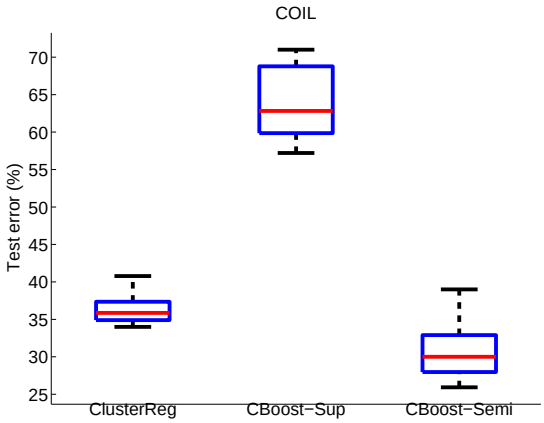


(a) BCI - 10 labels.

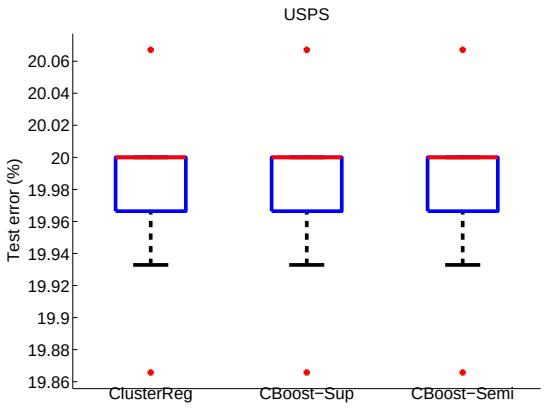
(b) BCI - 100 labels.



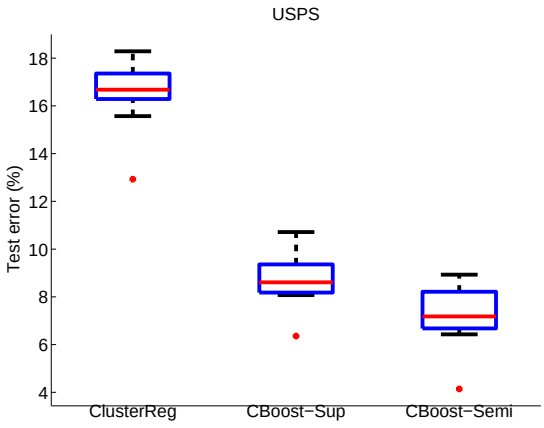
(c) COIL - 10 labels.



(d) COIL - 100 labels.



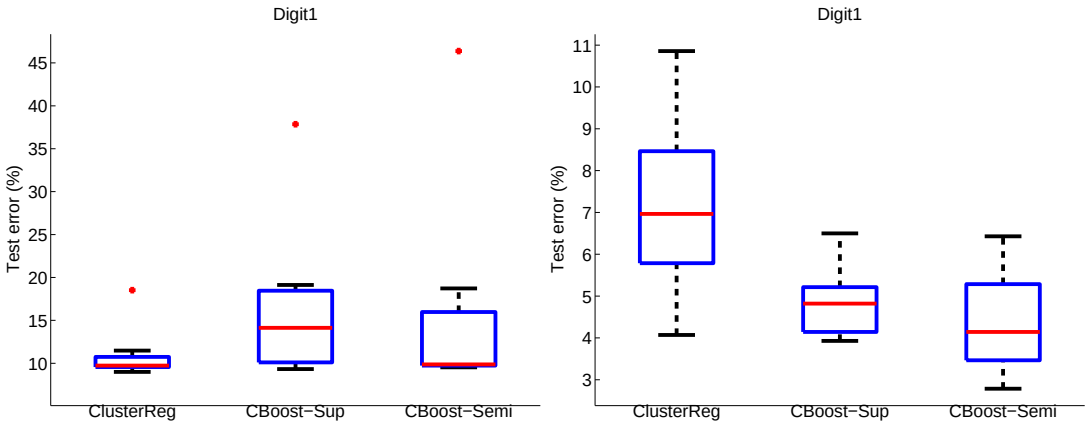
(e) USPS - 10 labels.



(f) USPS - 100 labels.

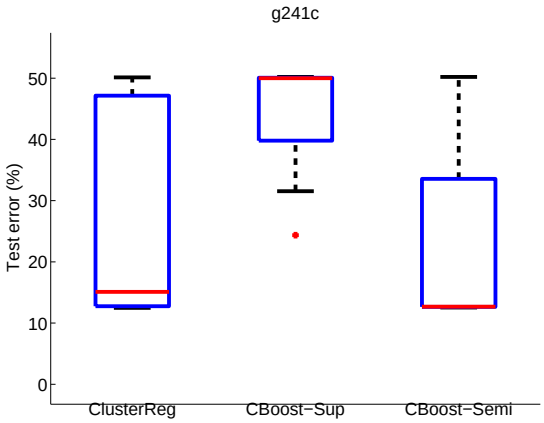
the setting described in Section 3.5.3.

It is not possible do not know in advance the true class structure and the corresponding

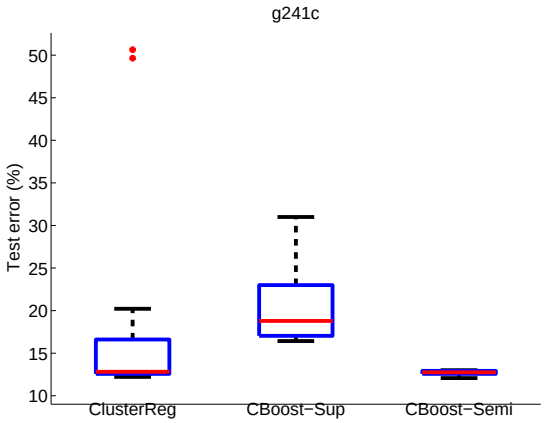


(g) Digit1 - 10 labels.

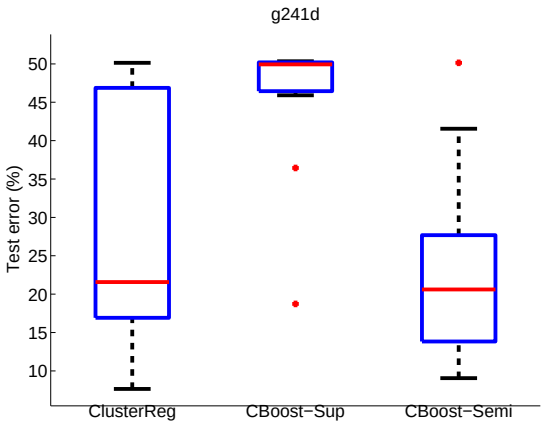
(h) Digit1 - 100 labels.



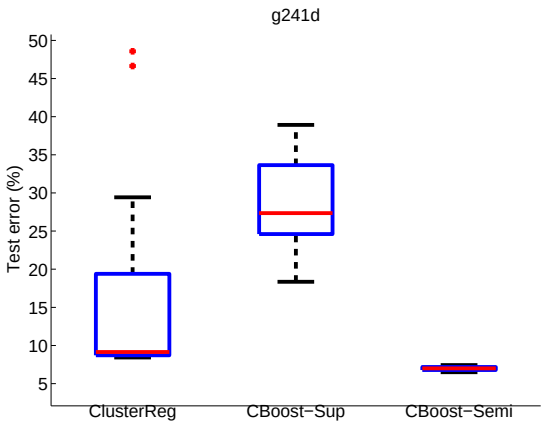
(i) g241c - 10 labels.



(j) g241c - 100 labels.



(k) g241d - 10 labels.



(l) g241d - 100 labels.

Figure 4.4: Boxplot of test errors (%) of ClusterReg, CBoost-Sup and CBoost-Semi.

SSL assumption that these real-world datasets possess. The success of a classifier will depend on the right matching between their assumptions and the actual class structure present in the data [Chapelle et al., 2006]. Ensemble-based algorithms with multiple assumptions may deliver higher average performance throughout various datasets [Chen and Wang, 2011], that is, such methods are more likely to deliver better predictions than a specialist algorithm that implements the wrong assumption for a given dataset.

In this sense, we compare CBoost-Semi to algorithms with two ensemble classifiers that work on three SSL assumptions – MCSSB and RegBoost. We also employ the base classifier used in CBoost, ClusterReg with RBFN, in this experiment to assess the improvement in generalisation ability over a single classifier. And, finally, we compare the proposed method to a similar ensemble with supervised base learners in order to investigate the difference in performance of supervised and semi-supervised base classifiers.

Tables 4.4, 4.5 and 4.6 show the mean and standard deviation of the generalisation error of all algorithms for all datasets with 5%, 10% and 20% of labelled data, respectively. We employ a pairwise t-test with 95% of significance level to compare the algorithms to CBoost-Sup. Symbols •/◦ indicate whether CBoost-Semi is statistically superior/inferior and Win/Tie/Loss denotes the number of datasets where CBoost-Semi is significantly superior/comparable/inferior to the compared algorithm.

4.4.4 Computational time

We compare the computational time of CBoost-Semi to a single classifier ClusterReg and to CBoost-Sup. In Figure 4.4, we plot the CPU time of each algorithm across 5%, 10% and 20% of labelled data of each dataset. We report the average time and its standard deviation of the 10-fold cross-validation executions that delivered the error rates shown in Tables 4.4, 4.5 and 4.6.

The CPU time was measured in an Intel Core 2 Quad CPU Q8200 with 2 gigabytes

Datasets	MCSSB	RegBoost	ClusterReg	CBoost-Sup	CBoost-Semi
Australian credit	44.52 ± 4.87 •	18.15 ± 3.74	41.88 ± 17.14 •	21.47 ± 3.47 •	18.67 ± 1.27
Balance scale	26.06 ± 5.58 •	57.30 ± 11.24 •	9.82 ± 1.90 ◦	23.07 ± 8.41 •	15.98 ± 4.93
Bupa	38.91 ± 10.85	47.45 ± 10.83 •	30.50 ± 2.21 ◦	49.58 ± 9.77 •	38.90 ± 4.90
Contraceptive	57.07 ± 4.59 •	67.76 ± 8.20 •	49.85 ± 1.27 ◦	49.38 ± 4.02 ◦	52.74 ± 2.84
Dermatology	11.12 ± 5.82 •	58.24 ± 5.63 •	23.39 ± 7.44 •	19.80 ± 4.87 •	5.34 ± 5.31
Ecoli	18.66 ± 5.96 •	37.62 ± 6.83 •	16.54 ± 4.74 •	14.59 ± 4.25 •	11.68 ± 2.81
German credit	31.46 ± 5.59	52.62 ± 21.26 •	23.27 ± 1.91 ◦	21.25 ± 1.73 ◦	29.03 ± 2.61
Glass	60.31 ± 10.60 •	77.53 ± 17.87 •	58.40 ± 9.29 •	38.96 ± 12.01	36.31 ± 10.36
Haberman	33.15 ± 11.00	31.53 ± 17.19	16.91 ± 3.06 •	28.00 ± 6.03	29.09 ± 2.71
Heart cleveland	47.34 ± 15.06	61.06 ± 7.89 •	40.85 ± 3.53 ◦	39.75 ± 5.14 ◦	53.42 ± 3.79
Horse colic	30.38 ± 10.08 •	48.44 ± 19.57 •	31.06 ± 5.61 •	25.87 ± 6.27	26.23 ± 6.17
House votes	61.57 ± 7.24 •	56.10 ± 12.64 •	7.81 ± 3.06	10.73 ± 3.69 •	7.84 ± 2.30
Ionosphere	35.64 ± 12.78 •	50.55 ± 19.80 •	12.97 ± 2.51	9.05 ± 2.04	13.35 ± 7.78
Mammographic masses	46.34 ± 4.80 •	25.42 ± 4.94 •	12.73 ± 3.19	14.91 ± 1.34	15.24 ± 3.36
Pima indians diabetes	34.82 ± 4.62 •	34.21 ± 7.51 •	27.05 ± 1.96 •	32.81 ± 4.21 •	29.66 ± 2.86
SPECT	79.51 ± 10.71 •	31.99 ± 4.27 •	11.09 ± 1.78	10.69 ± 2.74	11.08 ± 3.12
Vehicle silhouettes	49.47 ± 6.09 •	69.71 ± 5.89 •	52.11 ± 5.51 •	45.60 ± 3.18 •	35.33 ± 5.59
Transfusion	23.88 ± 6.03 •	34.59 ± 23.03	19.65 ± 6.21 ◦	22.80 ± 6.15 ◦	29.46 ± 5.48
WDBC	37.25 ± 5.37 •	18.93 ± 5.67 •	8.69 ± 1.17 •	7.02 ± 1.86	6.86 ± 2.68
Yeast	56.58 ± 3.03 •	68.63 ± 3.68 •	53.35 ± 2.12 •	53.06 ± 1.76 •	48.78 ± 0.99
Win/Tie/Loss	16/4/0	17/3/0	10/4/6	9/7/4	–

Table 4.4: Mean and standard deviation (%) of 10-fold cross-validation error with 5% of labelled data. •/◦ indicates whether CBoost-Semi is statistically superior/inferior to the compared method, according to pairwise t-test at 95% of significance level. Win/Tie/Loss denotes the number of datasets where CBoost-Semi is significantly superior/comparable/inferior to the compared algorithm.

Datasets	MCSSB	RegBoost	ClusterReg	CBoost-Sup	CBoost-Semi
Australian credit	44.58 ± 6.90 •	13.38 ± 2.54 ◦	12.76 ± 1.46 ◦	19.96 ± 2.67 •	16.18 ± 2.73
Balance scale	23.40 ± 5.29 •	46.80 ± 9.48 •	5.47 ± 2.69	9.38 ± 5.18 •	4.45 ± 1.72
Bupa	43.64 ± 9.92 •	47.11 ± 12.00 •	33.22 ± 2.43 •	26.80 ± 3.20 •	23.55 ± 4.31
Contraceptive	53.35 ± 3.51 •	61.00 ± 4.59 •	45.71 ± 1.91	45.37 ± 2.97	46.65 ± 1.80
Dermatology	9.97 ± 6.31 •	69.25 ± 5.95 •	20.12 ± 6.85 •	5.49 ± 4.84 •	1.26 ± 1.64
Ecoli	18.59 ± 6.63	35.11 ± 7.51 •	18.90 ± 5.93	25.98 ± 4.51 •	19.17 ± 4.57
German credit	32.35 ± 5.22 •	48.28 ± 16.27 •	22.55 ± 1.54	22.02 ± 2.36	22.83 ± 3.67
Glass	52.54 ± 11.18 •	67.30 ± 12.24 •	43.09 ± 9.44 •	19.01 ± 7.31	19.54 ± 4.36
Haberman	42.59 ± 10.20 •	29.91 ± 10.65	22.64 ± 5.44 ◦	28.79 ± 4.67 ◦	34.62 ± 5.88
Heart cleveland	52.73 ± 11.12	72.12 ± 12.89	37.81 ± 2.34 •	48.21 ± 8.02	48.32 ± 3.71
Horse colic	25.35 ± 9.32	57.12 ± 18.39 •	30.10 ± 6.89 •	23.45 ± 5.23	22.52 ± 5.19
House votes	61.35 ± 8.08 •	58.12 ± 11.63 •	11.76 ± 1.24 •	4.86 ± 1.93 •	1.78 ± 1.23
Ionosphere	35.90 ± 6.75 •	44.85 ± 15.40 •	10.48 ± 2.08 •	6.41 ± 4.06	8.27 ± 2.20
Mammographic masses	46.21 ± 6.15 •	21.11 ± 2.72	12.26 ± 1.50 ◦	14.49 ± 2.14 ◦	23.02 ± 4.94
Pima indians diabetes	34.84 ± 6.50 •	32.75 ± 5.10 •	27.90 ± 2.30	26.49 ± 2.74	28.39 ± 3.34
SPECT	79.60 ± 8.61 •	49.55 ± 32.36 •	15.45 ± 1.49 •	12.12 ± 1.06	11.70 ± 1.58
Vehicle silhouettes	43.46 ± 7.23 •	74.44 ± 2.74 •	55.63 ± 3.75 •	49.38 ± 3.32 •	37.90 ± 2.31
Transfusion	23.79 ± 6.93	35.07 ± 7.15	15.87 ± 1.94 ◦	19.98 ± 4.38	26.77 ± 16.47
WDBC	37.37 ± 7.19 •	13.86 ± 6.47 •	2.77 ± 1.49 ◦	5.12 ± 1.99	5.31 ± 2.05
Yeast	53.90 ± 3.70 •	68.63 ± 2.94 •	52.09 ± 3.50 •	50.26 ± 0.98 •	47.57 ± 1.66
Win/Tie/Loss	16/4/0	15/4/1	10/5/5	8/10/2	–

Table 4.5: Mean and standard deviation (%) of 10-fold cross-validation error with 10% of labelled data. •/◦ indicates whether CBoost-Semi is statistically superior/inferior to the compared method, according to pairwise t-test at 95% of significance level. Win/Tie/Loss denotes the number of datasets where CBoost-Semi is significantly superior/comparable/inferior to the compared algorithm.

Datasets	MCSSB	RegBoost	ClusterReg	CBoost-Sup	CBoost-Semi
Australian credit	44.34 ± 7.04 •	17.37 ± 5.21	16.14 ± 3.12	18.35 ± 3.03 •	15.82 ± 3.44
Balance scale	23.85 ± 8.12 •	55.06 ± 5.23 •	3.45 ± 1.08	2.21 ± 1.84	2.62 ± 2.45
Bupa	38.25 ± 10.96 •	52.16 ± 11.77 •	20.41 ± 5.00	20.24 ± 6.20	21.22 ± 4.65
Contraceptive	54.15 ± 6.38 •	57.22 ± 6.41 •	45.80 ± 3.09 •	48.45 ± 6.44 •	43.52 ± 2.12
Dermatology	6.52 ± 3.99	59.61 ± 8.20 •	14.71 ± 4.92 •	3.54 ± 3.00	4.13 ± 2.24
Ecoli	17.59 ± 7.73	37.52 ± 13.14 •	18.37 ± 3.34 •	12.64 ± 4.19	12.93 ± 7.42
German credit	33.83 ± 6.82 •	37.56 ± 16.99 •	23.90 ± 2.73 •	20.92 ± 3.18	19.73 ± 2.34
Glass	61.69 ± 12.82 •	67.06 ± 9.44 •	19.42 ± 6.93	18.84 ± 6.88	20.32 ± 7.66
Haberman	32.57 ± 9.23 •	25.95 ± 7.57	17.47 ± 5.46 ○	24.31 ± 6.08	25.60 ± 7.41
Heart cleveland	52.30 ± 12.83 •	56.29 ± 16.76 •	41.09 ± 6.02	42.66 ± 5.73	39.83 ± 5.19
Horse colic	40.87 ± 10.84 •	47.22 ± 14.98 •	37.05 ± 3.09 •	29.11 ± 4.99	29.12 ± 5.04
House votes	61.29 ± 7.43 •	50.04 ± 10.84 •	6.87 ± 2.88 •	9.68 ± 5.17 •	3.11 ± 1.99
Ionosphere	36.03 ± 10.85 •	38.46 ± 13.65 •	8.59 ± 1.78	8.27 ± 1.85	8.59 ± 1.78
Mammographic masses	46.45 ± 4.85 •	46.73 ± 5.50 •	10.52 ± 1.72	9.61 ± 1.80	10.36 ± 2.33
Pima indians diabetes	34.88 ± 7.24 •	31.74 ± 5.47 •	22.98 ± 3.42 ○	24.06 ± 3.63 ○	26.55 ± 2.69
SPECT	79.53 ± 5.20 •	30.85 ± 12.03 •	8.07 ± 2.53	8.07 ± 2.53	8.23 ± 4.06
Vehicle silhouettes	33.47 ± 4.32	72.05 ± 5.28 •	50.83 ± 5.46 •	37.02 ± 4.43	34.26 ± 4.72
Transfusion	23.78 ± 5.44	26.61 ± 4.43 •	16.63 ± 2.24 ○	18.44 ± 2.32	20.81 ± 3.69
WDBC	37.28 ± 6.42 •	28.99 ± 5.33 •	1.32 ± 1.14	2.41 ± 1.65	1.97 ± 1.28
Yeast	52.47 ± 4.27 •	68.65 ± 2.65 •	51.35 ± 2.79 •	49.54 ± 2.95 •	46.57 ± 2.61
Win/Tie/Loss	16/4/0	18/2/0	8/9/3	4/15/1	–

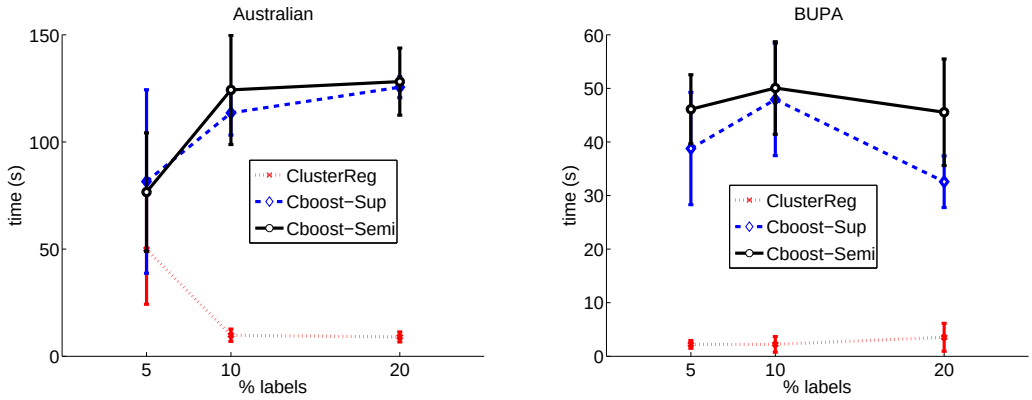
Table 4.6: Mean and standard deviation (%) of 10-fold cross-validation error with 20% of labelled data. •/○ indicates whether CBoost-Semi is statistically superior/inferior to the compared method, according to pairwise t-test at 95% of significance level. Win/Tie/Loss denotes the number of datasets where CBoost-Semi is significantly superior/comparable/inferior to the compared algorithm.

of memory. All algorithms were implemented in Matlab. The implementations of CBoost and ClusterReg can be further optimised.

4.5 Discussions

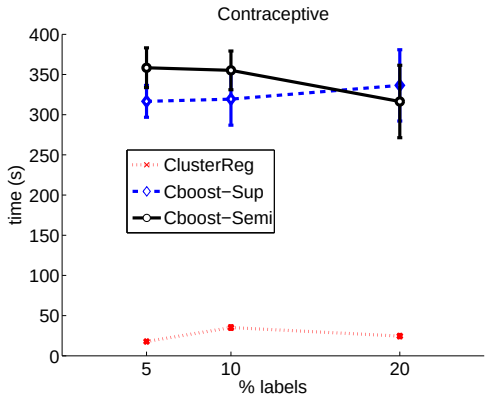
In the transductive setting (Tables 4.2 and 4.3), CBoost-Semi was superior to existing methods with 10 and 100 labelled instances, when the cluster assumption holds (as in g241c). Such performance is expected since CBoost-Semi improves the use of the cluster structure and it is robust to the few labelled instances available to generate a suitable decision boundary. Moreover, the proposed ensemble was superior to existing ensemble methods, which indicated that, in contrast to other ensembles, CBoost-Semi was able to overcome incorrect pseudo-labels during training.

When the cluster structure is misleading (as in g241d), CBoost-Semi could still produce superior generalisation than all other methods with 10 labelled instances. Such a fact

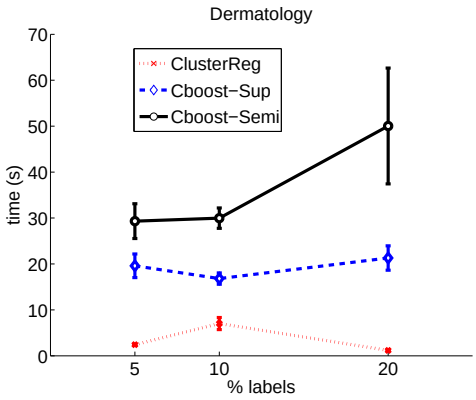


(a) Australian.

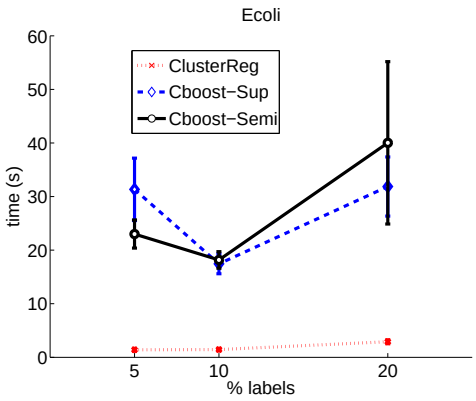
(b) BUPA.



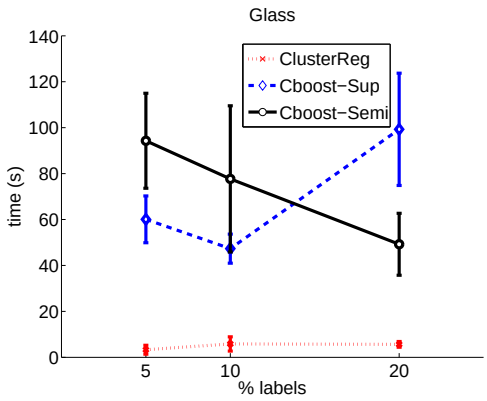
(c) Contraceptive.



(d) Dermatology.

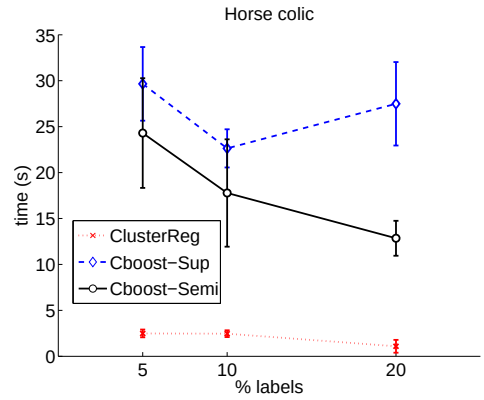


(e) Ecoli.

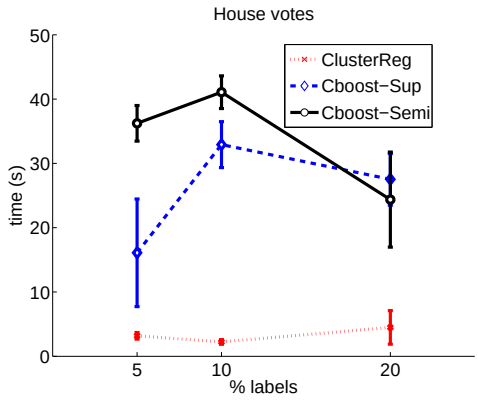


(f) Glass.

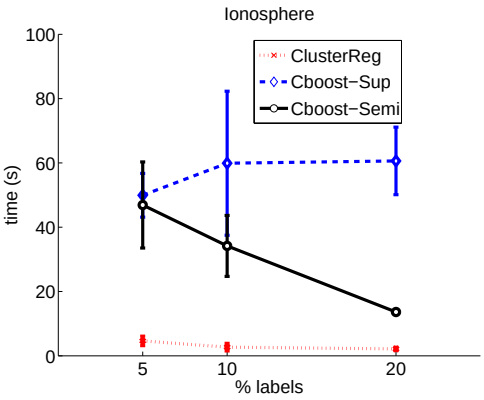
might indicate that, in the absence of a clear cluster structure, CBoost-Semi was able to learn from the few labelled instances more effectively than other algorithms. With 100 labelled instances, CBoost-Semi obtained superior generalisation than all other methods,



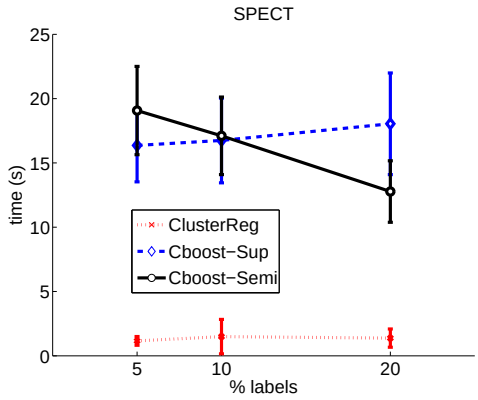
(g) Horse colic.



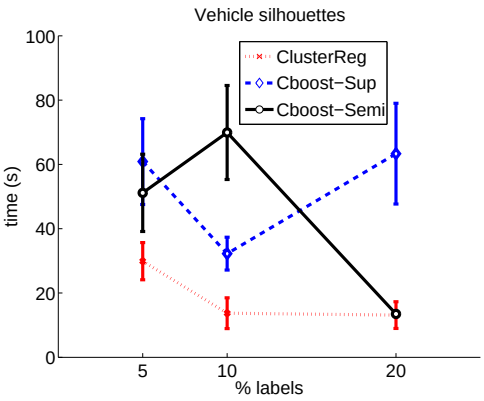
(h) House votes.



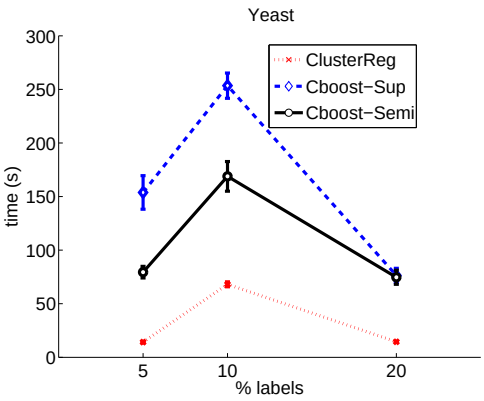
(i) Ionosphere.



(j) SPECT.



(k) Statlog.



(l) Yeast.

Figure 4.4: Plots of mean and standard deviation of the computational time of 10-fold cross-validation executions for 5%, 10% and 20% of labelled data.

except for Cluster-Kernel. Such performance demonstrates that the loss function and training algorithm of CBoost-Semi was suitable for cases where labelled data should have

a greater impact on the generation of decision boundaries.

As expected, manifold-based algorithms delivered highest generalisation for datasets where the manifold assumption holds (Digit1 and USPS) with 10 and 100 labelled points. Despite the unmatched assumptions, our proposed method obtained superior generalisation ability when compared with cluster-based and ensemble methods for Digit1. Such a fact also indicates that CBoost-Semi was able to use labelled instances more effectively in misleading datasets. Additionally, CBoost-Semi produced comparable accuracy among algorithms with the cluster assumption and multiple assumptions for USPS.

The performance of manifold-based algorithms in COIL with 10 and 100 labelled instances indicates that there is a manifold underlying the data, which also explains the results for methods with the cluster assumption. RegBoost delivered superior performance for BCI. Since RegBoost implements both cluster and manifold assumptions, this result suggests that such a dataset possesses both cluster and manifold structures [Chen and Wang, 2011]. For Text with 100 labelled instances, CBoost-Semi obtained comparable results to ensemble methods, however, with 10 labelled points, the proposed classifier could not produce a suitable decision boundary. A fine tuning of its parameters might improve the generalisation.

In order to evaluate the impact of the proposed ensemble algorithm and the use of semi-supervised base learners, Figure 4.4 shows the generalisation of CBoost-Sup, CBoost-Semi and their base classifier, ClusterReg.

CBoost-Semi was significantly superior when compared to ClusterReg for most datasets, especially g241d and COIL, where there is no useful cluster structure. In the presence of meaningful cluster structures, CBoost-Semi also delivered better predictive accuracy. These facts demonstrated the usefulness of our proposed ensemble technique in overcoming errors of individual classifier by training new base classifiers with the gradient direction of the current ensemble (residual errors represent the gradient of the loss

function). Such an ensemble technique produced better predictions regardless the SSL assumptions (both CBoost and ClusterReg implement cluster and smoothness assumptions in a similar manner). Such an improvement was also verified in real-world datasets.

Intuitively, when cluster structure is misleading or absent, it is common that classifiers with the cluster assumption generate sub-optimal[1] decision boundaries. Then, their performance will strongly depend on labelled data. In such a case, we expected CBoost-Sup to perform better than CBoost-Semi, since the use of yet another semi-supervised classifier in the framework may twist a potentially good decision boundary generated by the supervised term of the algorithm. This was the case for USPS dataset with 10 labels and Digit1 with both amounts of labels. However, for g241d with 10 and 100 labels and USPS with 100 labels, CBoost-Semi was superior. This fact indicates that, since the impact of unlabelled data on semi-supervised base classifier (ClusterReg) is weighted by λ, CBoost-Semi was not as affected as CBoost-Sup, where supervised base classifiers exactly learned all incorrect pseudo-labels provided. For g241c (where cluster assumption holds) and COIL, CBoost-Semi was superior, which denote that ClusterReg was more suitable as base learner than supervised classifier.

In the inductive setting, when compared to ClusterReg, CBoost-Semi was significantly superior in most datasets with 5%, 10% and 20% of labelled data. Except for Australian Credit dataset, where boosting might have degraded the decision boundary generated by its base learner. These results might denote that ensemble learning, as in supervised learning, could improve performance over single classifier in SSC.

Regarding MCCSB and RegBoost, which implement all mentioned SSC assumptions, CBoost-Semi yielded better predictive accuracy for most problems. MCSSB and RegBoost relied on pseudo-labels based on the current ensemble predictions. If current predictions had been incorrect, the supervised base learner would have reinforced such errors on

[1]When the employed SSL assumption does not match the actual structure in the data, predictions made by semi-supervised algorithms may be less accurate than predictions from fully supervised methods.

the ensemble. And both methods attempted to find the largest margin between classes. Thus, the results of the inductive setting indicate that such algorithms were sensitive to overlapping classes and to the position of labelled data in high-density regions. Moreover, RegBoost seemed to be affected by the number of classes: for all multi-class problems, it was less accurate than MCSSB, which was designed for multi-class classification.

In contrast, CBoost-Semi was robust to overlapping classes and to the position of few labelled instances in a given cluster, when the cluster assumption held, due to the use of ClusterReg. Additionally, the proposed method was specifically designed for multi-class classification. And both ensemble boosting algorithm and base classifiers optimise a semi-supervised function, so that base classifiers also consider the neighbourhood of an instance when learning its pseudo-label. These characteristics allowed base learners to overcome possible errors in pseudo-labels, as indicated by the inductive setting.

The exception was for the Australian Credit dataset with 10% and BUPA with 5%, where RegBoost and MCSSB obtained superior performance, respectively. In such particular cases, Australian Credit and BUPA datasets might possess a manifold structure that favours algorithms that implement the manifold assumption.

CBoost-Semi was also significantly superior to CBoost-Sup in most datasets. It is important to notice that, with 5% and 10% of labelled data, CBoost-Semi obtained a larger number of dataset where it delivered statistically better performance than CBoost-Sup. This fact indicates that using semi-supervised base classifiers improved predictive accuracy when there were very few labelled points. When the proportion of labelled point increased, the difference between these approaches seemed to decrease (as observed in Vehicle Silhouettes dataset). With 20% of labelled data, CBoost-Semi delivered better generalisation in four cases and was not inferior in any dataset. These results confirmed that employing ClusterReg as base learner improves the predictive performance of a cluster-based semi-supervised ensemble.

As we can notice, the use of unlabelled data had a great impact on the effectiveness of semi-supervised algorithms. Particularly for ensemble techniques, an important issue arises: at what level of an ensemble one should consider using unlabelled instances. In order to answer such a question, we proposed CBoost-Semi, a semi-supervised ensemble classifier that employs unlabelled data at both levels. And we compared its performance with CBoost-Sup, a similar ensemble technique that uses supervised base classifier. CBoost-Semi and CBoost-Sup share their loss function and boosting algorithm, so that we have a precise measure of the impact of using unlabelled data at both levels. In fact, our experiments confirmed the difference of performance of these two approaches.

Besides answering such a question, we also used ClusterReg in our analysis to verify the improvements in prediction accuracy of an ensemble over a single classifier. Additionally, we compared CBoost to existing ensemble methods in the literature. Our experiments validated the advantages of CBoost in several real-world datasets.

CBoost presented the following contributions: (i) a study of the impact of semi-supervised base learner in a semi-supervised ensemble; (ii) a cluster-based ensemble that shows robustness to overlapping classes and to the position of labelled instances in a cluster; (iii) an effective extension of ClusterReg to ensemble learning.

If either the dataset does not present a meaningful cluster structure or such a structure is misleading, the performance of CBoost might degrade. As expected, in our computational time analysis (depicted in Figure 4.4), we verified that the cost of obtaining high quality predictions, through the use of ensemble technique, is the increase of computational time. In all cases, ClusterReg was the fastest method. And, as anticipated, since CBoost-Semi uses semi-supervised base learners, which incurs the consideration of neighbourhoods, it requires more computational time than CBoost-Sup. In this case, computational time is the trade-off for better generalisation performance.

4.6 Conclusions

In this Chapter, we addressed the research question raised in Section 1.5.2. We introduced a robust multi-class ensemble, based on ClusterReg, in order to evaluate the impact of unlabelled data in ensemble design. According to our experimental analysis, we concluded that the use of semi-supervised base classifiers (ClusterReg) is beneficial to the generalisation ability of the ensemble.

In particular, we proposed a new multi-class semi-supervised classifier, CBoost, which extends ClusterReg algorithm by employing gradient boosting to improve ClusterReg's predictive accuracy. CBoost inherits ClusterReg's robustness to overlapping classes and to the position of few labelled instances in a given cluster when the cluster assumption holds. The proposed method is especially designed for multi-class problems, avoiding depending on decomposition techniques. And it uses a powerful semi-supervised base learner – ClusterReg instantiated with RBFN. Such a base classifier considers the neighbourhood of an instance when learning its pseudo-label. This approach leads to a more robust ensemble, since base learners may be able to overcome possible incorrect pseudo-labels.

Two experimental settings were investigated: transductive and inductive scenarios. Both confirmed the usefulness of extending ClusterReg to ensemble learning and the impact of using semi-supervised base learners. Our analysis demonstrated the improvement in performance of CBoost over single ClusterReg, CBoost with supervised base learners and other state-of-the-art methods, when the cluster assumption holds. Results were particularly encouraging for multi-class datasets.

As highlighted in Section 4.5, CBoost-Semi presented a higher computational time when compared to CBoost-Sup, due to the more time consuming training of semi-supervised base learners. In SSC, we often have large-scale datasets due to abundant unlabelled data. In the next Chapter, we investigate techniques that improve both time and memory

efficiencies of the proposed ensemble method, so that cluster-based SSC can be performed in large datasets.

CHAPTER 5

Efficient Boosting for Semi-supervised Classification

In this Chapter, we follow the motivation described in Section 1.5.3 and propose an efficient boosting algorithm for multi-class SSC. Such a method improves the time and memory complexities of CBoost, so that SSC can be performed in large-scale datasets.[1]

Due to a large number of available unlabelled data, a semi-supervised training set can often have tens of thousands of instances. Therefore, the learning algorithms must be able to handle such large datasets. Recently, various ensemble algorithms have been introduced with better generalisation performance when compared with single classifiers. However, existing ensemble algorithms are unable to handle large scale datasets. In this work, we propose an Efficient Cluster-based Boosting (ECB) algorithm. ECB uses a regularisation technique, based on posterior probabilities generated by a clustering algorithm, to avoid generating a decision boundary in high-density regions. In order to reduce the computational time, base learners are trained with a subset of the unlabelled data uniformly sampled from unlabelled set along with all available labelled instances at each iteration. We also employ an algorithm that automatically selects a suitable approx-

[1] In this Chapter, we consider large-scale datasets as relatively large datasets with tens of thousands of instances.

imation technique to increase the efficiency in the computation of nearest neighbours. We theoretically discuss the reason why ECB is able to achieve good performance with small amounts of sampled data and a relatively small number of base learners. Our experiments confirmed that ECB scales well to large datasets whilst delivering comparable generalisation to state-of-the-art methods.

The remainder of this Chapter is organized as follows. Next section discusses existing gaps in literature. Section 5.3 introduces the ECB algorithm and Section 5.4 reports the experimental studies with ECB. Section 5.5 discusses our results and contributions. In Section 5.6, we provide theoretical analysis on the amounts of sampled data and base learners employed in ECB. And the last section presents our conclusions.

5.1 Introduction

In SSC, we often find large datasets due to the abundant unlabelled data. Hence, a classification algorithm must be able to scale well for such datasets. The typical large number of unlabelled instances has a great impact on the efficiency of existing semi-supervised classifiers. Amongst methods that implement the cluster assumption (cluster-based algorithms), Transductive Support Vector Machines (TSVM) [Joachims, 2002] is a popular choice, however it requires time $\mathcal{O}(N^3)$ where N is the number of instances. Classifiers based on the manifold assumption (manifold-based algorithms) are also time consuming with computational complexity of $\mathcal{O}(N^3)$ or $\mathcal{O}(VN^2)$ where V is the number of neighbours [Szummer and Jaakkola, 2002, Zhu and Ghahramani, 2002]. The scalability issue is aggravated in multi-class classification, where suboptimal decomposition approaches are employed, such as *one-vs-all* or *one-vs-one*. These approaches require additional computational time.

Ensemble learning has been successfully employed in both supervised [Nguyen et al., 2006] and semi-supervised [Valizadegan et al., 2008, Chen and Wang, 2011] classification

to improve generalisation performance when compared with single classifiers. However, the use of existing ensemble techniques in large-scale SSC datasets is limited due to time and memory requirements. For example, RegBoost [Chen and Wang, 2011] is a binary ensemble classifier that, if implemented with SVM[1], requires time of $\mathcal{O}(VN \log N + TS^3 + TVU)$, where T is the number of base learners and S is the sample size. Due to the computation of nearest neighbours, RegBoost requires memory of $\mathcal{O}(N^2)$, which also might prevent its application to large datasets.

Additionally, the computational complexity of binary ensembles will increase for multi-class classification. A few multi-class ensemble approaches have been proposed [Valizadegan et al., 2008, Saffari et al., 2009]. However, despite having implemented the cluster assumption, these algorithms do not take advantage of the entire cluster structure.

We propose the Efficient Cluster-based Boosting (ECB) algorithm. ECB is a boosting approach that improves the scalability of ensemble learning for large-scale SSC datasets, while maintaining the generalisation ability of ensemble-based methods. ECB employs posterior cluster probabilities (soft partitions derived from a clustering algorithm) in its regularisation procedure in order to avoid generating the decision boundary in high-density regions. We tackle large datasets by sampling unlabelled points, along with pseudo-labels[2], to form the training set for each base learner. Our experiments and theoretical analysis demonstrate that the sampling technique delivers good predictive accuracy, despite a small number of base learners and sampled instances. ECB has the following benefits.

- ECB tackles large-scale datasets by simply uniformly sampling unlabelled instances to compose the training set of each new base classifier in a boosting procedure.

- Both ensemble and base classifiers optimise a semi-supervised loss function. Hence,

[1]SVM is the base classifier recommended by Chen and Wang [2011].

[2]In this work, pseudo-labels are posterior class probabilities that are systematically assigned to unlabelled instances by some classifier. Pseudo-labels might be different from true labels.

the base classifier will also consider the neighbourhood of an instance when learning its pseudo-label, so that the base learner may be able to correct potential errors from pseudo-labels.

- ECB employs efficient clustering algorithm and approximates nearest neighbours to reduce time and memory requirements. Our experiments demonstrated that, despite using sampling and approximation techniques, its predictive accuracy is similar to that of the state-of-the-art ensemble algorithms that use all available data at each iteration.

- ECB is designed for multi-class problems. Hence, it does not depend on decomposition techniques.

- ECB is robust to overlapping classes and to the position of the few labelled instances in a given cluster when the cluster assumption holds.

- The use of gradient boosting with uniform random sampling often lead to a good performance, despite the small number of iterations and sampled instances, as discussed in Section 5.6.

5.2 Background

In this section, we discuss the importance of proposing a cluster-based ensemble for multi-class SSC that is able to handle large amounts of data.

The computational complexity of various popular SSC methods prevents their applications to large datasets [Mann and McCallum, 2007]. Manifold-based algorithms require large computational effort due to the construction of a graph to represent the data. Such a graph has labelled and unlabelled points as vertices and labels are assigned to unlabelled vertices based on their neighbours. Zhu and Ghahramani [2002] introduced label propagation, where labelled instances are used to assign labels to unlabelled instances in

its neighbourhood according to the graph. In Joachims [2003], the authors use graphs to train a transductive version of the k-NN classifier. Szummer and Jaakkola [2002] employ random walks in a graph to assign labels to unlabelled data. The computational complexity of such methods is $\mathcal{O}(N^3)$ or $\mathcal{O}(VN^2)$, where V is the number of neighbours [Mann and McCallum, 2007]. Moreover, such manifold-based algorithms depend on the graph construction. The computational complexity issue has not been extensively studied yet [Zhu, 2008]. Additionally, these procedures usually cannot deal with unseen (test) data, that is, they are inherently transductive. This could prevent applications of graph-based methods in problems requiring fully inductive classifiers.

Most existing cluster-based algorithms are also computationally intensive. TSVM [Joachims, 2002] attempts to find the largest margin between classes by searching for different label assignments for unlabelled data and calculating margins between dense regions of similarly labelled instances. Such search is expensive and TSVM requires time of $\mathcal{O}(N^3)$. Such a method might not find the correct decision boundary between such regions (clusters) if dense regions are overlapping. TSVM might be sensitive to the few labelled points in the dense regions.

The aforementioned algorithms are binary classifiers. In order to perform multi-class classification, such methods rely on decomposition techniques, for example *one-vs-one* and *one-vs-all*. Thus, applying these costly algorithms to multi-class classification requires multiple and expensive runs. Such a drawback has a great impact on large-scale datasets [Saffari et al., 2009].

Ensemble algorithms, in particular boosting techniques, were successfully employed in SSC [Chen and Wang, 2011, Saffari et al., 2009, Valizadegan et al., 2008]. RegBoost [Chen and Wang, 2011] employs three assumptions in its boosting algorithm. In order to implement the cluster assumption, RegBoost uses a kernel density estimation approach, which will penalise the classifier if it does not assign the same label to a pair of neigh-

bour instances in a high-density region. However, if overlapping high-density regions are present, RegBoost might not establish a good separation between these regions. RegBoost requires time of $\mathcal{O}(VCN \log N + CTS^3 + TCVU)$ for multi-class classification and, due to search for nearest neighbours, demands memory of $\mathcal{O}(N^2)$, which might be prohibitive for large datasets. Yu et al. [2012] investigated the use of ensembles in high-dimensional SSC. Such algorithm divides features into several subspaces, builds a graph for each subspace, trains a linear algorithm (base classifier) on each graph and combines such classifiers as an ensemble. The computational complexity of such method is $\mathcal{O}(N^2D + NDD_{sub} + DD_{sub}^3)$, where D is the original dimensionality and D_{sub} is the subspace dimensionality.

The ensembles in Chen and Wang [2011], Yu et al. [2012] are binary classification algorithms and depend on the reduction of multi-class classification to multiple two-class problems. This exacerbated the high computational complexity issue for multi-class SSC.

In order to perform ensemble learning in multi-class SSC, Valizadegan et al. [2008], Saffari et al. [2009], Song et al. [2011] proposed multi-class boosting techniques. Valizadegan et al. [2008] introduced a Multi-Class Semi-Supervised Boosting (MCSSB) algorithm. MCSSB is a multi-class version of the SemiBoost algorithm proposed in Mallapragada et al. [2009]. Such an algorithm combines the similarity information among the instances with the classifier predictions to obtain more reliable pseudo-labels. It is a graph-based approach, and its objective function possesses three SSL assumptions and it uses supervised base classifiers. The computational complexity of MCSSB is $\mathcal{O}(TCU^2 + TS^3)$, where T is the number of boosting iterations, C is the number of classes and S is the number of sampled instances. MCSSB stores a similarity matrix that requires memory of $\mathcal{O}(N^2)$. Such requirements might limit its application to large datasets.

Applying the state-of-the-art algorithms described here to large-scale datasets is a challenging task due to their high computational complexity. Delalleau et al. [2006] proposed a sampling technique to reduce computational complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(S^2N)$,

where S is the number of sampled instances. However, such a technique is designed for transductive graph-based algorithms and the experimental results in Chapelle et al. [2006] show that the difference between such an algorithm and uniform random sampling is marginal. Other techniques for increasing efficiency can reduce the time complexity to $\mathcal{O}(S^3)$, where $S \ll N$, but also reduce performance [Zhu and Lafferty, 2005, Mann and McCallum, 2007].

In order to address such limitations, we propose an efficient cluster-based multi-class boosting technique that maintains comparable generalisation ability to state-of-the-art methods. We instantiated the gradient boosting framework [Friedman, 2001] to obtain an efficient ensemble method. Gradient boosting produces highly robust ensemble classifiers and its instantiation is straightforward [Friedman, 2001].

In our ECB, in order to handle a large amount of data, each base classifier is trained with a uniform random subset of unlabelled instances along with all labelled points.[1] Unlike other semi-supervised ensembles, ECB is composed of semi-supervised base learners. In this sense, it is able to learn from the clustering neighbourhood structure of pseudo-labels assigned by the ensemble, instead of using supervised base classifiers to learn the exact pseudo-labels individually. As a result, ECB is able to overcome incorrect pseudo-label assignments used in the training of a new base classifier.

Since ECB depends on the output of a clustering algorithm, we employed the Landmark-based Spectral Clustering (LSC) algorithm to compute posterior cluster probabilities [Chen and Cai, 2011]. We also used the approximation technique introduced in Muja and Lowe [2009] to efficiently obtain nearest neighbours and avoid the expensive computation of pairwise distance matrix. We use ClusterReg as the base learner due to its robustness and efficiency, as demonstrated in Chapter 3.

[1] In SSC, the number of labelled instances is orders of magnitude smaller than the number of labelled points. Therefore, we can avoid sampling and safely use all available labelled data in the training set of base learners.

In order to assess the impact of using a sampling technique and an approximation approach for nearest neighbours, we employ CBoost in our experimental analysis. Similar to ECB, CBoost is able to overcome possible errors in pseudo-labels assignments from the ensemble procedure and is robust to overlapping classes and to the position of few labelled instances in a given cluster when the cluster assumption holds. However, the complexity of CBoost is $\mathcal{O}(T_{base}C^2M^2U + TCVU)$, where T_{base} is the number of boosting iterations, C is the number of classes, M ($M \ll U$) is the number of hidden nodes of a RBFN, U is the number of unlabelled instances and V is the number of neighbours. In other words, CBoost is not efficient for large-scale datasets.

5.3 Efficient Cluster-based Boosting

In this section, we introduce the ECB algorithm. First, we present the general steps of ECB. Then, we show the loss function that is minimised. Later, we describe the regularisation mechanism and initialisation procedure. We highlight the techniques used to reduce both time and memory requirements of CBoost. Finally, we show the base learner and the proposed ensemble algorithm.

5.3.1 General architecture and notations

ECB is an extension of CBoost to large-scale datasets. It uses the regularisation mechanism of ClusterReg and employs the ensemble algorithm of CBoost. Our proposed method uses gradient boosting (described in Section 4.3) as its ensemble training algorithm. At each step of ECB, a new ensemble member is trained with the direction of the steepest gradient descent. The training set of a new base learner is composed of a uniform random sample $\mathbf{E}$ of size B from $\mathbf{U}$ and all L labelled instances. The number of instances in the training set is $S = B + L$. A combination of such members delivers the predictions. ECB also minimises the semi-supervised loss function $\mathcal{L}$ presented in Section 3.2.

ECB reduces time and memory complexities by (i) sampling the training set for each base leaner, (ii) emplying a large-scale clustering algorithm [Chen and Cai, 2011], and (iii) using approximation techniques for the nearest neighbours [Muja and Lowe, 2009].

Figure 5.1 shows the general architecture of the proposed method. It consists of the following steps.

1. Extract posterior probabilities of every point from an efficient clustering algorithm.

2. An approximation technique, based on the automatic selection and configuration of randomized kd-trees and hierarchical k-means tree algorithms [Muja and Lowe, 2009], is employed to find the nearest neighbours for each unlabelled instance.

3. Penalties are assigned to every neighbour of each instance according to the similarity between the posterior cluster probabilities of such instances.

4. Initialisation procedure assigns initial pseudo-labels to unlabelled instances according to labels present in the neighbourhood structure of each cluster. With the initial pseudo-labels, penalty values and nearest neighbours at hand, the training of the first base classifier is performed with the initial training set, which is composed of uniformly sampled unlabelled points and all labelled instances.

5. The ensemble algorithm generates pseudo-labels for each instance and trains a number of semi-supervised base classifiers with unlabelled instances that are sampled at each iteration. Then, the ensemble combines all trained base classifiers to form the final ensemble predictions.

6. A sample of unlabelled instances is uniformly drawn from the set of unlabelled data $\mathbf{U}$ and, along with the complete set of labelled instances $\mathbf{L}$, compose the training set for a base learner.

7. With the pseudo-labels generated by the ensemble algorithm, penalty values and nearest neighbours, several semi-supervised base classifiers are trained and included in the ensemble.

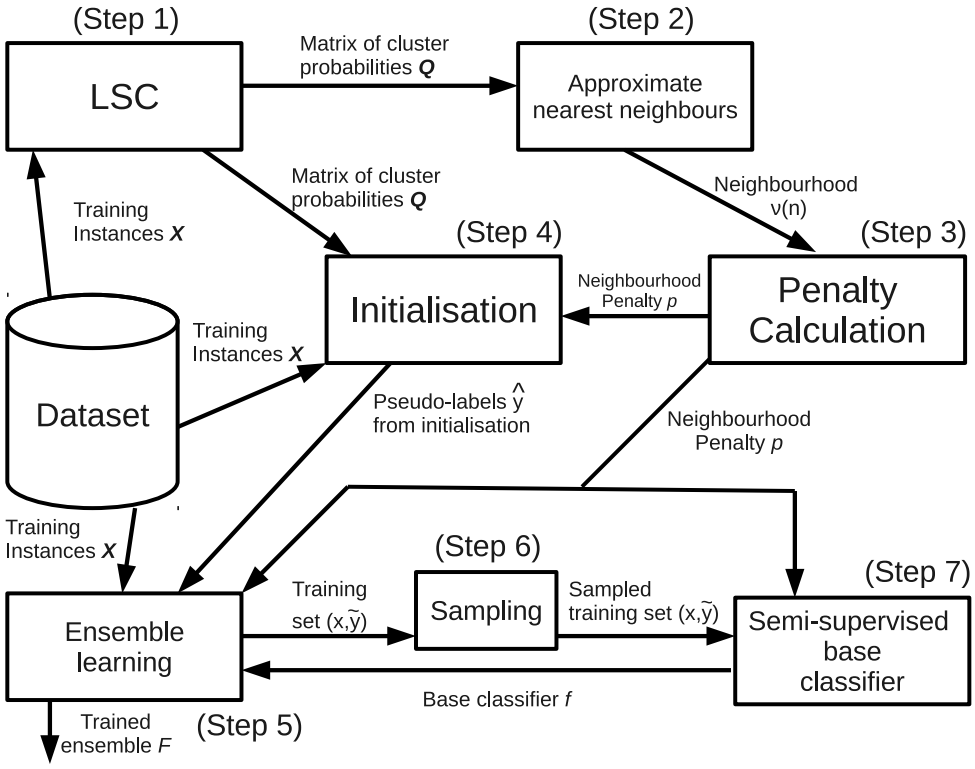


Figure 5.1: ECB's architecture.

5.3.2 Multi-class loss function with cluster regularisation

The loss function in ECB consists of two terms: supervised loss and cluster regularisation. In multi-class classification, it is useful to use cross-entropy cost and softmax output function [Bishop, 2006].

We assume that the output of a clustering algorithm is a partition $\mathbf{Q} = [q_{nk}]_{N \times K}$ with K clusters and N instances, where the row vector $\mathbf{q}_n$ contains the probabilities of instance n belonging to each one of the K clusters. Such vector sums to one and n is associated to the cluster with the highest probability.

Equation 5.1 defines the adopted multi-class loss function with cross entropy.

$$\mathcal{L}(\mathbf{F}_n, \mathbf{y}_n) = -\sum_{i=1}^{C} \left\{ \frac{I_{nL}}{L} y_{ni} \log\left(F_{ni}\right) + \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} u_{ni} \log\left(F_{ni}\right) \right\}, \tag{5.1}$$

where $I_{nL} = 1$ if $n \in \mathbf{L}$ and 0 otherwise, and $I_{nU} = 1$ if $n \in \mathbf{U}$ and 0 otherwise. $\mathbf{F}_n = \{F_{ni}\}_{i=1}^{C}$ denotes the output vector of the ensemble for instance n and C is the number of classes. We use the number of labelled L and unlabelled U instances to scale both terms of our loss function, since the number of unlabelled points is much greater than L. Without such scaling, the semi-supervised regularisation would dominate the total loss and the labelled error would not be considered. The parameter λ denotes the trade-off between the supervised loss and semi-supervised regularisation and $\max(\mathbf{q}_n)$ returns the maximum value in vector $\mathbf{q}_n$.

For unlabelled instances, we assign an estimated label to each unlabelled point according to its penalty values and neighbourhood (derived from posterior cluster probabilities). Equation 5.2 denotes the estimated desired output for an unlabelled instance.

$$u_{ni} = \frac{\sum_{k \in \nu(n)} \gamma(\mathbf{q}_k, \mathbf{q}_n) \hat{y}_{ki}}{\sum_{k \in \nu(n)} \gamma(\mathbf{q}_k, \mathbf{q}_n)}, \tag{5.2}$$

where

$$\hat{y}_{ki} = \begin{cases} y_{ki}, & \text{if } k \text{ is labelled} \\ F_{ki}, & \text{if } k \text{ is unlabelled.} \end{cases}$$

The current estimate u_{ni} is the probability of class i assigned to instance n. $\nu(n)$ represents the set of nearest neighbours of n. The penalty $\gamma(\mathbf{q}_k, \mathbf{q}_n)$ is calculated according to posterior cluster probabilities. A higher similarity between i and k incurs a higher penalty for that pair. When k is unlabelled, $\hat{y}_{kj}$ is also known as the pseudo-label of k. Then, u_{ni} is the weighted average of current pseudo-labels of the neighbourhood of n.

5.3.3 Cluster-based regularisation

Equation 5.3 computes the penalty between n and k. Similarity is denoted by $s(\mathbf{q}_n, \mathbf{q}_k)$ and is normalised in $[0, 1]$. Then, we use the first quarter of the sine function, as suggested in Chen and Wang [2011], with its slope determined by κ, to map similarity into penalisation.

$$\gamma(\mathbf{q}_n, \mathbf{q}_k) = \sin\left(\frac{\pi}{2}\left(s(\mathbf{q}_n, \mathbf{q}_k)\right)^{\kappa}\right). \tag{5.3}$$

The parameter κ regulates the steepness of the mapping from similarity to penalisation.[1] With a higher κ, only most similar instances will have a high penalty. It controls the extent in which the decision boundary avoids clusters. Equation 3.5 defines the similarity measure $s(\mathbf{q}_n, \mathbf{q}_k)$.

5.3.4 Initialisation procedure

For many SSC algorithms, if the classifier assigns the same class to every unlabelled instance, the training error will be in a local optimum [Mann and McCallum, 2007]. This fact is due to the loss function comparing a predicted output with similar outputs of its neighbours. In order to overcome such a local optimum, we use an initialisation procedure that employs the distribution of labelled points in a cluster to assign initial pseudo-labels to unlabelled data.

We use the sum of labels present in a cluster, weighted by penalty values $\gamma(\mathbf{q}_n, \mathbf{q}_k)$, to assign pseudo-labels to unlabelled instances in such cluster [Chen and Wang, 2011]. If there is no labelled points in a cluster, equal probabilities will be assigned to each class. For class i of unlabelled instance n in cluster Ψ we have:

$$\hat{y}_{ni} = \frac{\sum_{k \in \Psi} I_{kL} * \gamma(\mathbf{q}_n, \mathbf{q}_k) * y_{ki}}{\sum_{k \in \Psi} I_{kL} * \gamma(\mathbf{q}_n, \mathbf{q}_k)}. \tag{5.4}$$

[1] In this Chapter, κ is set to 5 for all experiments.

The initialisation procedure consists of training an initial base classifier for a small number of iterations[1] with the pseudo-labels $\hat{y}_{ni}$.

5.3.5 Approximate nearest neighbours and large-scale clustering

In our proposed method, the clustering algorithm has a great impact on both generalisation and efficiency. Our preliminary experiments showed that LSC [Chen and Cai, 2011] can lead to good generalisation accuracy when compared to other clustering algorithms used in Chapter 3, for example k-means and the spectral method in Manor and Perona [2004]. Such an algorithm can handle large datasets. Its time complexity is $\mathcal{O}(T_{km}PND + P^3 + P^3D)$, where T_{km} is the number of iterations in k-means, P is the number of landmarks and $P \ll N$. Thus, we select LSC as the clustering algorithm employed to produce matrix $\mathbf{Q}$.

Semi-supervised methods often seek the labels in the neighbourhood of an instance in order to assign pseudo-labels. The construction of such a neighbourhood requires the calculation of all pairwise distances in a $N \times N$ matrix and the search for all neighbours, which requires time of $\mathcal{O}(VN \log N)$ and memory of $\mathcal{O}(N^2)$ [Vaidya, 1989], where V is the number of neighbours.

Since there is a number of approximation techniques that can be employed to reduce computational complexity, we chose a method that automatically selects a suitable approximation method to find nearest neighbours for each instance (row) represented in $\mathbf{Q}$ with less time requirement [Muja and Lowe, 2009]. In order to approximate nearest neighbours, we used Fast Library for Approximate Nearest Neighbours (FLANN) [Muja and Lowe, 2009]. This algorithm automatically chooses between randomized kd-tree and hierarchical k-means tree algorithms in order to approximate the nearest neighbours. The

[1] Throughout this Chapter, we use 10 iterations of pre-training, as different numbers did not improve performance in preliminary experiments.

selection of the approximation algorithm and its parameters is performed with the minimisation of a cost function that considers both time and memory requirements. With the use of such an approximation technique, ECB reduces the memory requirement to $\mathcal{O}(VN)$, and $V \ll N$.

FLANN considers the nearest neighbour method as a parameter in an optimisation procedure. Such optimisation searches for parameters that minimise the proposed cost function. This cost function is a combination of the search time, tree build time and memory usage. The trade-off between accuracy and efficiency is controlled by a user-defined algorithm. FLANN selects the best nearest neighbour algorithm and their optimum parameters in a two-step approach: a global exploration of the parameters and a local tuning. The first step is performed with a sampling procedure. And the second step employs a simplex method to locally optimise the parameters obtained in the first step.

The output of FLANN is a matrix with the distances from each instance in $\mathbf{Q}$ to its V neighbours. We use such matrix to calculate the penalty values employed in the regularisation term of our loss function. The soft partition arising from the clustering algorithm is used to generate regularisation and, therefore, to implement the cluster assumption in our algorithm. ECB employs the smoothness assumption by penalising the classifier if it assigns different labels to similar instances, as denoted by $-u_{ni} \log (f_{ni})$.

5.3.6 Sampling procedure

The training of multiple base classifiers limits the use of existing ensemble algorithms in large-scale classification. This shortcoming is exacerbated in the multi-class context [Saffari et al., 2009]. Delalleau et al. [2006] proposed a sampling procedure to tackle large-scale datasets. However, their technique is designed for transductive graph-based algorithms. In the experimental analysis of Chapelle et al. [2006], such a sampling technique did not show considerable improvement over uniform random sampling. Moreover, many SSC

methods (ensemble and single classifiers) depend on the calculation of a pairwise distance matrix [Valizadegan et al., 2008] that requires memory in the order of $\mathcal{O}(N^2)$, which also restrain the application of ensemble methods to large datasets.

We propose to use uniform random sampling of unlabelled data to compose the training set of base classifiers and solve both issues highlighted above. We uniformly sample B unlabelled instances and use all L labelled points to form a training set of size S for each base learner, where $S = B + L$. Apart from forming a smaller training set, sampling will also decrease memory usage for storing penalties from an unlabelled instance to its neighbours.

Such a simple sampling procedure reduces the time complexity of the base learner from $\mathcal{O}(T_{base}C^2M^2U)$ to $\mathcal{O}(T_{base}C^2M^2S)$, where $S \ll U$ and T_{base} is the number of epochs for RBFN. Despite the use of sampling, ECB is still able to achieve good generalisation ability when compared to other ensemble methods. Section 5.6 provides a theoretical discussion on the reasons for such a performance. In fact, our experiments confirmed that ECB maintains comparable performance with the state-of-the-art algorithms. Our experimental analysis also presents comparisons with CBoost, which does not use sampling procedure and computes the exact nearest neighbours.

The calculation of loss function requires time of $\mathcal{O}(TCVU)$, where T is the number of base learners (iterations) in ECB. Then, along with the time complexity of the base learner, the time complexity of ECB becomes $\mathcal{O}(T_{base}C^2M^2S+TCVU)$. Therefore, unlike existing state-of-the-art ensemble methods [Valizadegan et al., 2008, Chen and Wang, 2011, Yu et al., 2012], the time complexity of ECB grows linearly with the number of unlabelled instances.

5.3.7 Radial Basis Function Network as the base learner

In our proposed method, we use ClusterReg as base learner due to its robustness to overlapping classes and presence of few labelled instances in borders of clusters, as demonstrated in Chapter 3.

As discussed in Section 3.6, we instantiate ClusterReg with RBFN since such networks produce high generalisation accuracy and can be efficient and easily adapted as base learner in our method. In fact, the experimental analysis in Section 3.5 confirmed that RBFN was more efficient and delivered higher predictive accuracy than MLP networks.

Since the number of hidden nodes has an impact on the efficiency of RBFN, we use a small number of instances as centres in the hidden layer of RBFN for large datasets. However, these points should be useful for the training of ClusterReg, that is, such instances should be representative for the dataset. Thus, we employ the algorithm introduced in Engel et al. [2004] to select meaningful centres from the training set of size S of each base learner. The maximum number of centres is tuned according to Section 5.4.1.

And, in the supervised training phase of RBFN, we employ the procedure described in Section 3.3 to train the weights of ClusterReg. Due to the calculation of the Hessian matrix in Equation 3.14, each base classifier requires time of $\mathcal{O}(T_{base}C^2M^2S)$.

In order to initialise the ensemble, we train the initial base learner assigned to $\mathbf{Z}^0$ exclusively with the pre-training procedure described in Section 5.3.4. It consists of training the initial base classifier for number of iterations with the pseudo-labels $\hat{y}_{ni}$ delivered by Equation 5.4. During such a training, the pseudo-labels $\hat{y}_{ni}$ are fixed.

5.3.8 Boosting for large-scale multi-class classification

Following Friedman [2001], Bishop [2006], we employ the softmax function (Equation 5.5) to transforms linear outputs $\mathbf{Z}$ into posterior class probabilities $\mathbf{F}$.

$$F_{ni} = \text{softmax}(Z_{ni}) = \frac{\exp(Z_{ni})}{\sum_j^C \exp(Z_{nj})}. \tag{5.5}$$

Multi-class base classifiers have the form $\mathbf{f}_n = \text{softmax}(\mathbf{z}_n)$, where $\mathbf{f}_n$ is the predicted posterior class probabilities of n.

Unlike original gradient boosting, a base classifier, trained with pseudo-labels $\hat{y}$ delivered by the initialisation procedure, is assigned to the initial ensemble $\mathbf{Z}^0$. Based on our preliminary experiments, such an approach delivered better results than simply assigning a constant to the initial ensemble, for example $\mathbf{Z}^0 = 0$.

We calculate the derivative of $\mathcal{L}(F_{nj}^{t-1})$ w.r.t. Z_{nj}^{t-1} to obtain the current residuals r_{nj} for class j that will be used to train a new base classifier $\mathbf{f}$. Such residuals are computed as in Equation 5.6. Obtaining Equation 5.6 is similar to the derivation of Equation 3.13, thus we omit such steps.

$$r_{nj} = -\frac{I_{nL}}{L} * (F_{nj}^{t-1} - y_{nj}) - \frac{I_{nU}\lambda \max(\mathbf{q}_n)}{U} * (F_{nj}^{t-1} - u_{nj}) \tag{5.6}$$

We uniformly sample a subset $\mathbf{E}$ of size B from $\mathbf{U}$ and include all labelled instances, $\mathbf{S} = \mathbf{E} \cup \mathbf{L}$, to form the training set of a base learner. The number of instances in the training set becomes $S = B + L$.

We assume that labels y_{nj} are class probabilities (true labels are denoted as value one and other classes as zeros in the probability vector). The residuals r_{nj}, then, must be transformed into the proper target values in probability scale, that is, $\tilde{y}_{nj} = \text{softmax}(r_{nj})$. The pseudo-labels $\tilde{\mathbf{y}}$ assigned to the reduced training set $\mathbf{S}$ are used to train a new base

learner $\mathbf{f}$.

We use a single Newton-Raphson step to search for multiplier vector $\boldsymbol{\beta}^t = \{\beta_j^t\}_{j=1}^C$. For each class, β_j^t is calculated as in Equation 5.7. β_j^t is initially 0.

$$\beta_j^t = -\mathbf{H}^{-1} * \frac{\partial \mathcal{L}(\mathbf{F}^{t-1}, \mathbf{y})}{\partial \beta_j^t}. \tag{5.7}$$

The gradient $\frac{\partial \mathcal{L}}{\partial \beta_j^t}$ for each instance is[1]

$$\frac{\partial \mathcal{L}(\mathbf{F}_n^{t-1}, \mathbf{y}_n)}{\partial \beta_j^t} = \frac{I_{nL}}{L}\left(F_{nj}^{t-1} - y_{nj}\right) z_{nj} + \frac{I_{nU}\lambda \max(\mathbf{q}_n)}{U}\left(F_{nj}^{t-1} - u_{nj}\right) z_{nj}, \tag{5.8}$$

and Hessian matrix $\mathbf{H}$ is

$$H_{jk} = \sum_{n=1}^{N}\left\{\left(\frac{I_{nL}}{L} + \frac{I_{nU}\lambda \max(\mathbf{q}_n)}{U}\right) * F_{nj}^{t-1}\left(\delta_{jk} - F_{nk}^{t-1}\right) * z_{nj} z_{nk}\right\}, \tag{5.9}$$

where $\delta_{jk} = 1$ if $j = k$ and 0 otherwise. Obtaining Equations 5.8 and 5.9 is similar to the derivations of Equations 4.12 and 4.13, respectively, thus we omit such steps.

The base classifier is included in the current ensemble following the rule in Equation 5.10, where η is a learning rate that might reduce overfitting by decreasing the influence of newly trained base learners on the ensemble.

$$Z_{nj}^t = Z_{nj}^{t-1} + \eta \beta_j^t z_{nj} \tag{5.10}$$

Several greedy steps of gradient descent are performed until a stopping criterion is met, for example, a fixed number of iterations T, or the increase of training or validation errors. In order to produce posterior class probabilities as the ensemble outputs, we apply Equation 5.5. The proposed ensemble technique is summarised in Algorithm 4.

[1] We derive β_j^t w.r.t $\mathbf{F}^{t-1}$ since initially $\beta_j^t = 0$ and hence $\mathbf{F}^t = \mathbf{F}^{t-1}$.

Algorithm 4 ECB algorithm with RBFN.

Input: Training set $\mathbf{X} = \mathbf{L} \cup \mathbf{U}$, where $\mathbf{L} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{L}$, $\mathbf{U} = \{\mathbf{x}_n\}_{n=L+1}^{L+U}$ and $N = L + U$, often $U \gg L$.

Output: Posterior class probabilities $\mathbf{F}^t$.

1: Calculate initial pseudo-labels as

$$\hat{y}_{ni} = \frac{\sum_{k \in \Psi} I_{kL} * \gamma(\mathbf{q}_n, \mathbf{q}_k) * y_{ki}}{\sum_{k \in \Psi} I_{kL} * \gamma(\mathbf{q}_n, \mathbf{q}_k)}.$$

2: Randomly generate a subset $\mathbf{E}$ of size B from $\mathbf{U}$ and include all labelled instances $\mathbf{S} = \mathbf{E} \cup \mathbf{L}$, the number of instances in training set becomes $S = B + L$.

3: Train a RBFN $f_{ni} = \text{softmax}(z_{ni})$ with $\mathbf{S}$ and initial labels $\hat{y}_{ni}$.

4: Assign initial ensemble $Z_{nj}^0 = z_{nj}$ and $F_{nj}^0 = \text{softmax}(Z_{nj}^0)$.

5: **for** $t = 1$ to T, $n = 1$ to N and $j = 1$ to C **do**

6: Update $\hat{y}_{nj}$ using F_{nj}^t with

$$\hat{y}_{kj} = \begin{cases} y_{kj}, & \text{if } k \text{ is labelled} \\ F_{kj}^t, & \text{if } k \text{ is unlabelled.} \end{cases}$$

7: Update class probabilities u_{nj} for unlabelled instances with

$$u_{nj} = \frac{\sum_{k \in \nu(n)} \gamma(\mathbf{q}_k, \mathbf{q}_n) \hat{y}_{kj}}{\sum_{k \in \nu(n)} \gamma(\mathbf{q}_k, \mathbf{q}_n)}.$$

8: Find residuals of $\mathcal{L}$ w.r.t. $\mathbf{F}$ with rule

$$r_{nj} = -\left[\frac{\partial \mathcal{L}(\mathbf{F}_n, \mathbf{y}_n)}{\partial Z_{nj}^{t-1}}\right] = -\frac{I_{nL}}{L} * (F_{nj}^{t-1} - y_{nj}) - \frac{I_{nU} \lambda \max(\mathbf{q}_n)}{U} * (F_{nj}^{t-1} - u_{nj}).$$

9: Find pseudo-labels $\tilde{y}_{nj} = \text{softmax}(r_{nj})$.

10: Randomly generate a subset $\mathbf{E}$ of size B from $\mathbf{U}$ and include all labelled instances $\mathbf{S} = \mathbf{E} \cup \mathbf{L}$.

11: Train RBFN $f_{nj} = \text{softmax}(z_{nj})$ with $\mathbf{S}$ and $\tilde{y}_{nj}$.

12: Find multiplier β_j^t using rule $\beta_j^t = -\mathbf{H}^{-1} * \frac{\partial \mathcal{L}(\mathbf{F}^{t-1}, \mathbf{y})}{\partial \beta_j^t}$.

13: Update linear combination with

$$Z_{nj}^t = Z_{nj}^{t-1} + \eta \beta_j^t z_{nj}.$$

14: Update the posterior class probabilities with $F_{nj}^t = \text{softmax}(Z_{nj}^t)$.

15: **end for**

5.4 Experimental studies

In this section, we perform extensive experiments with two settings: transductive and inductive. We show the selection of parameters and discuss results with artificial and real-world datasets. We also present a comparison in terms of efficiency and effectiveness to state-of-the-art algorithms, including CBoost, using large-scale datasets.[1]

5.4.1 Methods and parameter tuning

Since MCSSB [Valizadegan et al., 2008] uses all three SSL assumptions, we expect ECB to outperform MCSSB only on datasets where the cluster assumption holds, that is, datasets that possess a clear cluster structure that relates to the class distribution. MCSSB would deliver better results on datasets where there is an unclear or no cluster structure. As its base classifier, we chose SVM, since it delivered the best results in our preliminary experiments for large-scale datasets. We fixed the parameter[2] $C = 10000$. The ratio of the range of distances used for kernel construction was searched in $\sigma \in \{0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.5, 0.8, 1\}$. We set the sample size as a ratio $\{0.1, 0.5, 0.8, 1\}$ of the total number of instances for transductive and inductive contexts and for large-scale datasets we fixed the sample size at 0.1. The number of base learners was set to 200 for large-scale datasets in order to have a fair comparison with ECB. Such a parameter was tuned with 20 and 50 for the remainder of the experiments. We performed grid search with parameter combinations and the best 10-fold cross-validation result is reported.[3]

For RegBoost [Chen and Wang, 2011], the authors suggested a grid search for the best combination of parameters. The number of iterations was 20 and 50 for inductive and transductive settings and 200 for the large-scale experiments. The number of neighbours

[1] All datasets were standardized with zero mean and standard deviation of one.

[2] As demonstrated in Valizadegan et al. [2008] and confirmed by our preliminary experiments, this value should be set to 10000. Lower and higher values did not improve the performance.

[3] Throughout this work, grid search is performed with 10-fold cross-validation.

was searched in $\{3, 4, 5, 6\}$. The resampling rate in the first iteration was set to 0.1. The resampling rate in the rest of iterations was searched in $\{0.1, 0.25, 0.5\}$. For large-scale datasets, we fixed the resampling rate at 0.1. Following the results from our preliminary experiments, we chose SVM as the base classifier.

In ECB, λ balances supervised loss and regularisation from unlabelled data. We perform a grid search in $\{0.2, 0.4, 0.6, 0.8, 1\}$, as we do not know in advance whether cluster assumption is indeed present in the dataset. We suggest setting this value between 0 and 1 since different values might degrade generalisation performance. In order to provide diversity to the ensemble, λ is uniformly drawn from the interval $[0.2, 1]$ for base classifiers.[1]

As in Chapter 3, the number of neighbours V was set to 30 for most datasets used in this work. Further tuning might improve generalisation accuracy. As mentioned in Section 5.3.5, we used the FLANN algorithm as the approximation method for producing nearest neighbours. The quality of the obtained neighbours is controlled by the precision parameter, which is the only user-defined parameter of FLANN. This parameter corresponds to number of leaf nodes (instances) that will be examined in both randomized kd-tree and hierarchical k-means tree. A higher precision produces more exact nearest neighbours and incurs a greater computational effort. The precision denotes the desired percentage of exact neighbours (other neighbours are approximations). As in Muja and Lowe [2009], we set the target precision parameter to 0.7 (that is, 70% of exact nearest neighbours and 30% of approximate neighbours) and the other parameters were set as default [Muja and Lowe, 2009]. From the available distances in Muja and Lowe [2009], we selected Euclidean distance in the search for nearest neighbours. Other distances may be used to improve performance for particular datasets.

In addition, we have developed an approach for automatic algorithm selection and

[1] Assigning same λ for all base classifiers did not improve generalisation error according to our preliminary experiments.

configuration, which allows the best algorithm and parameter settings to be determined automatically for any given dataset.

We employed LSC to generate the partition used in cluster-based regularisation, as such method is able to find clusters with arbitrary shapes and can be employed to large-scale datasets. We chose the default configuration of LSC where the selection of landmarks is performed by the k-means algorithm. We fixed the number of landmarks to 200 for all settings, as we found it a good trade-off between efficiency and generalisation ability.

The parameter K should be set to, at least, the number of classes. We also tried a larger number of clusters (multiples of the number of classes). In the case where the class structure is not captured by the clustering algorithm, we can increase the number of clusters, so that one class is composed of multiple clusters. The algorithm will avoid cutting through these clusters and may generate a decision boundary that does not divide such class. We performed a grid search in $\{1, 2, 3, 4\}$ times the number of classes for all experimental settings.

We fixed κ to 5 throughout all experiments (value in the middle of the range suggested in Chapter 3). Further tuning of this parameter might lead to better results.

The subset size B was searched in $\{200, 1500\}$ for transductive and inductive settings and fixed to 1500 for large-scale experiments.

We employed the algorithm introduced in Engel et al. [2004] to select the centres of RBFN. Then, we set the maximum number of centres to B for transductive and inductive settings. For large-scale datasets, such a parameter was fixed to 100. Our preliminary experiments showed that assigning different centre widths to each base classifier delivered better results than using the same value for all base learners. Such fact is expected since the centre widths have a great impact on RBFN's predictive ability and it is important to possess a certain degree of diversity. Thus, centre widths of RBFNs are uniformly drawn from between 20% and 80% of the median of all pairwise Euclidean distances between

Parameters	Inductive and transductive settings	Large datasets
λ	Grid search in $\{0.2, 0.4, 0.6, 0.8, 1\}$	1
K	Grid search in $\{1, 2, 3, 4\}$ times the number of classes	two times the number of classes
B	Grid search in $\{200, 1500\}$	1500
T	20	200

Table 5.1: Summary of tuned parameters for ECB.

sampled instances. A similar approach was adopted to set α, which is uniformly drawn from $[0.2, 0.5]$ for each base classifier.[1]

For transductive and inductive settings, we fixed the number of base classifier to 20, η was fixed to 0.5 and the number of IRLS iterations for RBFN was set to 50 (further optimisation on these values can improve results). We used these values for large-scale datasets, except for the number of base classifiers that was set to 200.

For CBoost, we used LSC as the clustering algorithm and we calculated the exact nearest neighbours (no approximation techniques). For the rest of the parameters, we followed the tuning scheme as for ECB. In Table 5.1, we summarise the tuning of each parameter in ECB.

5.4.2 Transductive setting

We aim to establish that ECB is able to produce comparable generalisation to state-of-the-art algorithms, despite the use of approximate nearest neighbours and sampled data for the training of base learners. We evaluate ECB with datasets with various underlying structures and compare the proposed method with single and ensemble algorithms, along with ClusterReg and CBoost.

In the transductive setting, test instances are regarded as unlabelled data and the generalisation error is the training error on unlabelled data. We use the datasets and the

[1]Ranges for λ, α and centre widths were empirically tested in our preliminary experiments.

setting described in Section 3.5.2 and summarised in Table 3.2 to evaluate ECB.

Tables 5.2 and 5.3 present the generalisation error with 10 and 100 labelled instances, respectively. We compare ECB with the algorithms described in Section 3.5.2. Such algorithms are grouped according to their assumptions: manifold-based, cluster-based, ensemble and methods with multiple assumptions. All results shown in Tables 3.3 and 3.4 were reported in Chapelle et al. [2006, Chapter 21], except for AdaBoost, ASSEMBLE and RegBoost, which were produced in Chen and Wang [2011]. The results of SAMME, ClusterReg-MLP and ClusterReg-RBFN were obtained in our experiments.

5.4.3 Inductive setting

It is also important to evaluate ECB in a scenario where predictions are performed for unseen instances, therefore we study the generalisation of ECB in an inductive setting. We use this setting to evaluate ECB along with existing algorithms, namely, ClusterReg, MCSSB, RegBoost. We employ the datasets summarised in Table 3.5 and the setting described in Section 3.5.3.

We performed 10-fold cross-validation for all datasets. In order to have the best error estimate as possible, all labels in the test set were available. It is not possible to know in advance the true class structure and the corresponding SSL assumption that these real-world datasets possess. The success of a classifier will depend on the right matching between their assumptions and the actual class structure present in the data [Chapelle et al., 2006]. Ensemble-based algorithms with multiple assumptions may deliver higher average performance throughout various datasets [Chen and Wang, 2011], that is, such methods are more likely to deliver better predictions than a specialist algorithm that implements the wrong assumption for a given dataset. In this sense, we compare ECB to algorithms with two ensemble classifiers that use all SSL assumptions – MCSSB and RegBoost – and a cluster-based ensemble, CBoost.

Algorithm	g241c	g241d	Digit1	USPS	COIL	BCI	Text
Manifold-based algorithms							
1NN	44.05	43.22	23.47	19.82	65.91	48.74	39.44
MVU+1NN	48.68	47.28	11.92	14.88	65.72	50.24	39.40
LEM+1NN	47.47	45.34	12.04	19.14	67.96	49.94	40.48
QC+CMN	39.96	46.55	9.80	**13.61**	59.63	50.36	40.79
Discrete Reg.	49.59	49.05	12.64	16.07	63.38	49.51	40.37
SGT	**22.76**	**18.64**	8.92	25.36	*n/a*	49.59	29.02
Laplacian RLS	43.95	45.68	**5.44**	18.99	**54.54**	48.97	**33.68**
CHM (normed)	39.03	43.01	14.86	20.53	*n/a*	**46.90**	*n/a*
Cluster-based algorithms							
SVM	47.32	46.66	30.60	20.03	68.36	49.85	45.37
TSVM	24.71	50.08	17.77	25.20	67.50	49.15	31.21
Cluster-Kernel	48.28	42.05	18.73	19.41	67.32	48.31	42.72
Data-Rep. Reg.	41.25	45.89	12.49	17.96	63.65	50.21	*n/a*
LDS	28.85	50.63	15.63	**15.57**	**61.90**	49.27	**27.15**
ClusterReg-MLP	**16.90**	40.82	12.06	19.42	65.51	**45.36**	40.48
ClusterReg-RBFN	26.94	**27.95**	**10.64**	19.98	69.13	49.19	40.48
Ensembles and multiple-assumptions algorithms							
AdaBoost	40.12	43.05	28.92	25.57	71.16	47.08	47.42
SAMME	50.09	50.07	50.07	19.98	70.25	50.30	*n/a*
ASSEMBLE	40.62	44.41	23.49	21.77	65.49	48.96	49.13
RegBoost	38.22	42.90	17.94	**17.41**	65.39	**46.73**	**34.96**
CBoost	22.76	23.07	14.72	19.98	**64.33**	48.50	43.77
ECB	**19.90**	**20.84**	**12.76**	19.98	65.22	48.29	43.66

Table 5.2: Average of errors (%) of runs with 12 subsets of 10 labelled instances. For all the algorithms, the test sets are fixed. The table reports only the mean of the results, as in Chapelle et al. [2006, Chapter 21]. All results shown in Tables 3.3 and 3.4 were reported in Chapelle et al. [2006, Chapter 21], except for AdaBoost, ASSEMBLE and RegBoost, which were produced in Chen and Wang [2011]. The results of SAMME, ClusterReg-MLP and ClusterReg-RBFN were obtained in our experiments. Bold face denotes the best result among each group of algorithms. And *n/a* denotes the absent results in Chapelle et al. [2006, Chapter 21].

Tables 5.4, 5.5 and 5.6 show the mean and standard deviation of the generalisation error of all algorithms for all datasets with 5%, 10% and 20% of labelled data, respectively. We employ a pairwise t-test with 95% of significance level to compare the algorithms to ECB. Symbols •/◦ indicate whether ECB is statistically superior/inferior and Win/Tie/Loss denotes the number of datasets where ECB is significantly supe-

Algorithm	g241c	g241d	Digit1	USPS	COIL	BCI	Text
Manifold-based algorithms							
1NN	40.28	37.49	6.12	7.64	23.27	44.83	30.77
MVU+1NN	44.05	43.21	3.99	6.09	32.27	47.42	30.74
LEM+1NN	42.14	39.43	**2.52**	6.09	36.49	48.64	30.92
QC+CMN	22.05	28.20	3.15	6.36	10.03	46.22	25.71
Discrete Reg.	43.65	41.65	2.77	**4.68**	**9.61**	47.67	24.00
SGT	**17.41**	**9.11**	2.61	6.80	*n/a*	45.03	**23.09**
Laplacian RLS	24.36	26.46	2.92	**4.68**	11.92	**31.36**	23.57
CHM (normed)	24.82	25.67	3.79	7.65	*n/a*	36.03	*n/a*
Cluster-based algorithms							
SVM	23.11	24.64	5.53	9.75	22.93	34.31	26.45
TSVM	18.46	22.42	6.15	9.77	25.80	**33.25**	24.52
Cluster-Kernel	13.49	4.95	3.79	9.68	21.99	35.17	24.38
Data-Rep. Reg.	20.31	32.82	**2.44**	5.10	**11.46**	47.47	*n/a*
LDS	18.04	28.74	3.46	**4.96**	13.72	43.97	**23.15**
ClusterReg (MLP)	**13.38**	**4.36**	3.45	5.25	24.73	33.92	32.09
ClusterReg (RBFN)	19.54	17.07	7.20	16.53	36.35	48.11	32.09
Ensembles and multiple-assumptions algorithms							
AdaBoost	24.82	26.97	9.09	9.68	22.96	24.02	26.31
SAMME	36.75	38.70	19.55	16.94	53.79	41.64	*n/a*
ASSEMBLE	27.19	27.42	6.71	8.12	21.84	28.75	27.77
RegBoost	20.54	23.56	4.58	**6.31**	**21.78**	**23.69**	**23.25**
CBoost	**12.71**	**6.99**	**4.34**	7.20	30.67	38.83	25.58
ECB	15.12	7.27	4.65	7.25	30.15	38.86	25.63

Table 5.3: Average of errors (%) of runs with 12 subsets of 100 labelled instances. For all the algorithms, the test sets are fixed. The table reports only the mean of the results, as in Chapelle et al. [2006, Chapter 21]. All results shown in Tables 3.3 and 3.4 were reported in Chapelle et al. [2006, Chapter 21], except for AdaBoost, ASSEMBLE and RegBoost, which were produced in Chen and Wang [2011]. The results of SAMME, ClusterReg-MLP and ClusterReg-RBFN were obtained in our experiments. Bold face denotes the best result among each group of algorithms. And *n/a* denotes the absent results in Chapelle et al. [2006, Chapter 21].

rior/comparable/inferior to the compared algorithm.

5.4.4 Large-scale datasets

In this section, we present a scalability study between ECB and other methods. First, we show the generalisation error and the computational time required for large datasets.

Datasets	MCSSB	RegBoost	ClusterReg	CBoost	ECB
Australian credit	44.52 ± 4.87 ○	18.15 ± 3.74	41.88 ± 17.14 ○	18.67 ± 1.27 ○	20.03 ± 1.46
Balance scale	26.06 ± 5.58 ●	57.30 ± 11.24 ●	9.82 ± 1.90	15.98 ± 4.93 ●	11.84 ± 3.53
Bupa	38.91 ± 10.85	47.45 ± 10.83	30.50 ± 2.21 ○	38.90 ± 4.90	41.33 ± 6.99
Contraceptive	57.07 ± 4.59 ●	67.76 ± 8.20 ●	49.85 ± 1.27	52.74 ± 2.84 ●	49.53 ± 2.93
Dermatology	11.12 ± 5.82	58.24 ± 5.63 ●	23.39 ± 7.44 ●	5.34 ± 5.31	7.03 ± 5.36
Ecoli	18.66 ± 5.96 ●	37.62 ± 6.83 ●	16.54 ± 4.74 ●	11.68 ± 2.81	11.73 ± 2.62
German credit	31.46 ± 5.59 ●	52.62 ± 21.26 ●	23.27 ± 1.91	29.03 ± 2.61 ●	25.25 ± 2.58
Glass	60.31 ± 10.60	77.53 ± 17.87 ●	58.40 ± 9.29	36.31 ± 10.36 ○	57.33 ± 11.01
Haberman	33.15 ± 11.00	31.53 ± 17.19	16.91 ± 3.06 ○	29.09 ± 2.71 ○	35.35 ± 3.26
Heart cleveland	47.34 ± 15.06	61.06 ± 7.89 ●	40.85 ± 3.53 ○	53.42 ± 3.79 ●	45.71 ± 6.22
Horse colic	30.38 ± 10.08	48.44 ± 19.57 ●	31.06 ± 5.61	26.23 ± 6.17	27.34 ± 6.33
House votes	61.57 ± 7.24 ●	56.10 ± 12.64 ●	7.81 ± 3.06	7.84 ± 2.30	7.58 ± 2.10
Ionosphere	35.64 ± 12.78 ●	50.55 ± 19.80 ●	12.97 ± 2.51 ●	13.35 ± 7.78	11.15 ± 1.46
Mammographic masses	46.34 ± 4.80 ●	25.42 ± 4.94 ●	12.73 ± 3.19	15.24 ± 3.36 ●	11.96 ± 2.79
Pima indians diabetes	34.82 ± 4.62 ●	34.21 ± 7.51 ●	27.05 ± 1.96	29.66 ± 2.86 ●	26.76 ± 2.53
SPECT	79.51 ± 10.71 ●	31.99 ± 4.27 ●	11.09 ± 1.78	11.08 ± 3.12	9.89 ± 2.82
Vehicle silhouettes	49.47 ± 6.09 ●	69.71 ± 5.89 ●	52.11 ± 5.51 ●	35.33 ± 5.59	34.41 ± 4.34
Transfusion	23.88 ± 6.03 ●	34.59 ± 23.03 ●	19.65 ± 6.21	29.46 ± 5.48 ●	19.07 ± 6.28
WDBC	37.25 ± 5.37 ●	18.93 ± 5.67 ●	8.69 ± 1.17	6.86 ± 2.68 ○	8.69 ± 0.74
Yeast	56.58 ± 3.03 ●	68.63 ± 3.68 ●	53.35 ± 2.12 ●	48.78 ± 0.99	48.22 ± 1.82
Win/Tie/Loss	13/6/1	17/3/0	5/11/4	7/9/4	–

Table 5.4: Mean and standard deviation (%) of 10-fold cross-validation error at 5% of labelled data. ●/○ indicates whether ECB is statistically superior/inferior to the compared method, according to pairwise t-test at 95% of significance level. Win/Tie/Loss denotes the number of datasets where ECB is significantly superior/comparable/inferior to the compared algorithm.

Datasets	MCSSB	RegBoost	ClusterReg	CBoost	ECB
Australian credit	44.58 ± 6.90 ●	13.38 ± 2.54 ○	12.76 ± 1.46 ○	16.18 ± 2.73	15.64 ± 2.37
Balance scale	23.40 ± 5.29 ●	46.80 ± 9.48 ●	5.47 ± 2.69	4.45 ± 1.72	4.60 ± 1.43
Bupa	43.64 ± 9.92 ●	47.11 ± 12.00 ●	33.22 ± 2.43 ●	23.55 ± 4.31	24.85 ± 5.26
Contraceptive	53.35 ± 3.51 ●	61.00 ± 4.59 ●	45.71 ± 1.91	46.65 ± 1.80 ●	43.83 ± 3.55
Dermatology	9.97 ± 6.31 ●	69.25 ± 5.95 ●	20.12 ± 6.85 ●	1.26 ± 1.64	1.33 ± 2.36
Ecoli	18.59 ± 6.63	35.11 ± 7.51 ●	18.90 ± 5.93 ●	19.17 ± 4.57 ●	14.71 ± 2.71
German credit	32.35 ± 5.22 ●	48.28 ± 16.27 ●	22.55 ± 1.54 ●	22.83 ± 3.67	21.01 ± 1.87
Glass	52.54 ± 11.18 ●	67.30 ± 12.24 ●	43.09 ± 9.44 ●	19.54 ± 4.36	19.48 ± 3.00
Haberman	42.59 ± 10.20 ●	29.91 ± 10.65	22.64 ± 5.44 ○	34.62 ± 5.88	32.88 ± 2.68
Heart cleveland	52.73 ± 11.12 ●	72.12 ± 12.89 ●	37.81 ± 2.34	48.32 ± 3.71 ●	40.43 ± 4.78
Horse colic	25.35 ± 9.32	57.12 ± 18.39 ●	30.10 ± 6.89 ●	22.52 ± 5.19	23.11 ± 4.21
House votes	61.35 ± 8.08 ●	58.12 ± 11.63 ●	11.76 ± 1.24 ●	1.78 ± 1.23 ○	6.39 ± 3.34
Ionosphere	35.90 ± 6.75 ●	44.85 ± 15.40 ●	10.48 ± 2.08	8.27 ± 2.20	8.54 ± 3.80
Mammographic masses	46.21 ± 6.15 ●	21.11 ± 2.72 ●	12.26 ± 1.50	23.02 ± 4.94 ●	12.61 ± 1.87
Pima indians diabetes	34.84 ± 6.50 ●	32.75 ± 5.10 ●	27.90 ± 2.30 ●	28.39 ± 3.34 ●	24.22 ± 2.80
SPECT	79.60 ± 8.61 ●	49.55 ± 32.36 ●	15.45 ± 1.49 ●	11.70 ± 1.58	12.12 ± 1.06
Vehicle silhouettes	43.46 ± 7.23 ●	74.44 ± 2.74 ●	55.63 ± 3.75 ●	37.90 ± 2.31	36.33 ± 3.34
Transfusion	23.79 ± 6.93 ●	35.07 ± 7.15 ●	15.87 ± 1.94	26.77 ± 16.47	17.73 ± 3.23
WDBC	37.37 ± 7.19 ●	13.86 ± 6.47 ●	2.77 ± 1.49	5.31 ± 2.05 ●	2.81 ± 2.12
Yeast	53.90 ± 3.70 ●	68.63 ± 2.94 ●	52.09 ± 3.50 ●	47.57 ± 1.66	46.43 ± 1.50
Win/Tie/Loss	18/2/0	18/1/1	11/7/2	6/13/1	–

Table 5.5: Mean and standard deviation (%) of 10-fold cross-validation error at 10% of labelled data. ●/○ indicates whether ECB is statistically superior/inferior to the compared method, according to pairwise t-test at 95% of significance level. Win/Tie/Loss denotes the number of datasets where ECB is significantly superior/comparable/inferior to the compared algorithm.

Datasets	MCSSB	RegBoost	ClusterReg	CBoost	ECB
Australian credit	44.34 ± 7.04 •	17.37 ± 5.21	16.14 ± 3.12 ◦	15.82 ± 3.44 ◦	18.52 ± 2.68
Balance scale	23.85 ± 8.12 •	55.06 ± 5.23 •	3.45 ± 1.08	2.62 ± 2.45	3.47 ± 1.79
Bupa	38.25 ± 10.96 •	52.16 ± 11.77 •	20.41 ± 5.00	21.22 ± 4.65	18.10 ± 6.09
Contraceptive	54.15 ± 6.38 •	57.22 ± 6.41 •	45.80 ± 3.09	43.52 ± 2.12	44.86 ± 3.58
Dermatology	6.52 ± 3.99 •	59.61 ± 8.20 •	14.71 ± 4.92 •	4.13 ± 2.24	3.46 ± 1.77
Ecoli	17.59 ± 7.73	37.52 ± 13.14 •	18.37 ± 3.34	12.93 ± 7.42 ◦	19.06 ± 6.43
German credit	33.83 ± 6.82 •	37.56 ± 16.99 •	23.90 ± 2.73 •	19.73 ± 2.34	20.85 ± 2.87
Glass	61.69 ± 12.82 •	67.06 ± 9.44 •	19.42 ± 6.93	20.32 ± 7.66 •	14.40 ± 6.16
Haberman	32.57 ± 9.23 •	25.95 ± 7.57 •	17.47 ± 5.46	25.60 ± 7.41	19.98 ± 7.60
Heart cleveland	52.30 ± 12.83 •	56.29 ± 16.76 •	41.09 ± 6.02	39.83 ± 5.19	36.69 ± 5.98
Horse colic	40.87 ± 10.84 •	47.22 ± 14.98 •	37.05 ± 3.09 •	29.12 ± 5.04	29.17 ± 4.75
House votes	61.29 ± 7.43 •	50.04 ± 10.84 •	6.87 ± 2.88 •	3.11 ± 1.99	4.30 ± 2.11
Ionosphere	36.03 ± 10.85 •	38.46 ± 13.65 •	8.59 ± 1.78	8.59 ± 1.78	7.93 ± 2.31
Mammographic masses	46.45 ± 4.85 •	46.73 ± 5.50 •	10.52 ± 1.72	10.36 ± 2.33	10.94 ± 2.13
Pima indians diabetes	34.88 ± 7.24 •	31.74 ± 5.47 •	22.98 ± 3.42	26.55 ± 2.69 •	22.26 ± 2.92
SPECT	79.53 ± 5.20 •	30.85 ± 12.03 •	8.07 ± 2.53	8.23 ± 4.06	7.64 ± 2.77
Vehicle silhouettes	33.47 ± 4.32	72.05 ± 5.28 •	50.83 ± 5.46 •	34.26 ± 4.72	30.81 ± 4.48
Transfusion	23.78 ± 5.44 •	26.61 ± 4.43 •	16.63 ± 2.24	20.81 ± 3.69	18.18 ± 3.14
WDBC	37.28 ± 6.42 •	28.99 ± 5.33 •	1.32 ± 1.14	1.97 ± 1.28 •	0.89 ± 1.15
Yeast	52.47 ± 4.27 •	68.65 ± 2.65 •	51.35 ± 2.79 •	46.57 ± 2.61	46.64 ± 3.12
Win/Tie/Loss	18/2/0	19/1/0	6/13/1	3/15/2	–

Table 5.6: Mean and standard deviation (%) of 10-fold cross-validation error at 20% of labelled data. •/◦ indicates whether ECB is statistically superior/inferior to the compared method, according to pairwise t-test at 95% of significance level. Win/Tie/Loss denotes the number of datasets where ECB is significantly superior/comparable/inferior to the compared algorithm.

Then, we present an analysis of the impact of the size of sampled subset. Finally, we show the number of base learners required for convergence of ECB.

We compare the computational time and generalisation error of ECB to MCSSB, RegBoost and CBoost.[1] We used 3 datasets: SecStr [Chapelle et al., 2006], Acoustic [Duarte and Hu, 2004] and Shuttle [Hsu and Lin, 2002]. Table 5.7 summarises such datasets. We randomly selected 100 labelled instances for each dataset.

Datasets	# classes	# instances	# attributes
SecStr	2	83679	315
Acoustic	3	98528	50
Shuttle	7	58000	9

Table 5.7: Summary of large datasets.

Figures 5.2, 5.3 and 5.4 present the generalisation error and computational time for

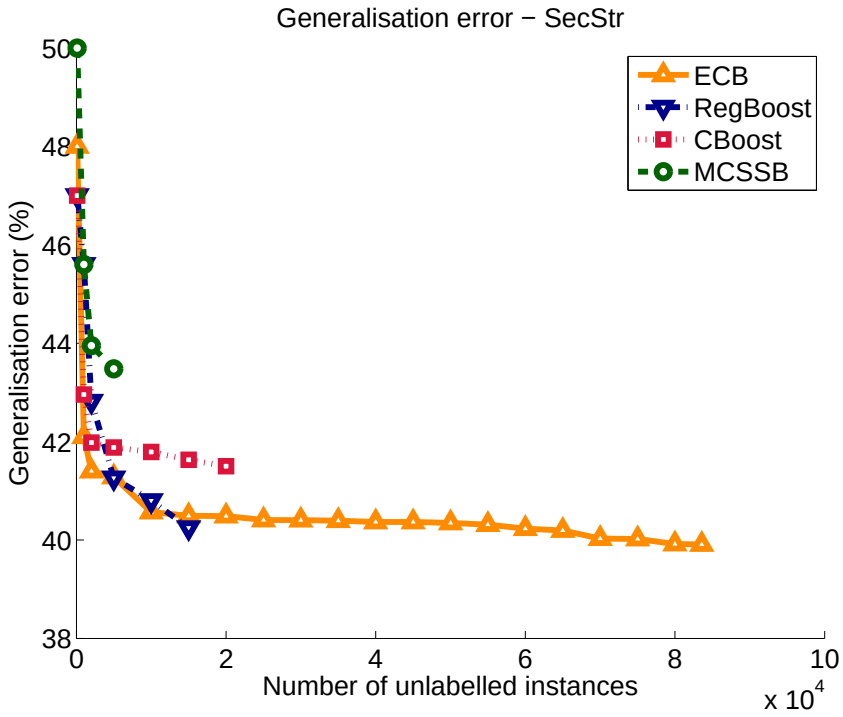
[1]The CPU time was measured in an Intel(R) Xeon(R) CPU at 2.20GHz with 64 gigabytes of memory. All algorithms were implemented in Matlab(R). The implementation of ECB can be further optimised.

the SecStr, Acoustic and Shuttle datasets, respectively. For each step in these plots, new unlabelled instances are randomly chosen and included in the previous training set.

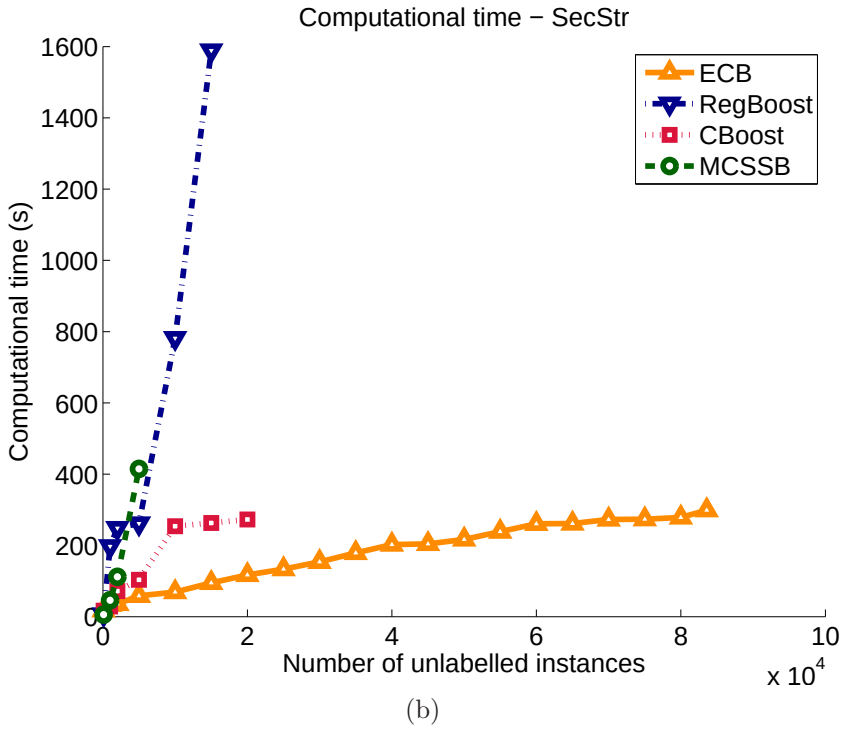
We also analyse the influence of the amount of sampled unlabelled instances. Figures 5.5 and 5.6 show the impact of different sample sizes on the generalisation error and computational time. In order to evaluate the number of base learners required to converge ECB, we plot the generalisation error in Figure 5.7.

5.5 Discussions

In order to analyse the generalisation ability of ECB, we performed experiments with both transductive and inductive settings. In the transductive setting (Tables 5.2 and 5.3), we compared ECB to existing algorithms described in Chapelle et al. [2006], Chen and Wang [2011]. For the g241c dataset (with 10 and 100 labelled instances), ECB obtained, as expected, superior performance to all manifold-based algorithms since such a dataset holds the cluster assumption. When compared to cluster-based classifiers applied to g241c with 10 labelled instances, ECB obtained better performance than most algorithms, except for the single classifier ClusterReg, which indicates that the ensemble approach might have overfit the data. With 100 labelled points, ECB outperformed most algorithms, except for CBoost. Such fact might indicate that the use of approximate nearest neighbours had an impact on the generalisation accuracy. Despite of g241d having a misleading cluster structure, ECB achieved comparable results to SGT (best performance) and was superior to all other cluster-based algorithms, with 10 labelled instances. With 100 labelled instances, ECB also obtained comparable results to the best algorithm (ClusterReg). Such a performance is explained by the cluster regularisation technique inherited from ClusterReg. In such a method, classes can be represented by more than one cluster and, even though the data distribution does not match the class distribution, these classes can be identified by clustering algorithms using multiple clusters for each class. Therefore,

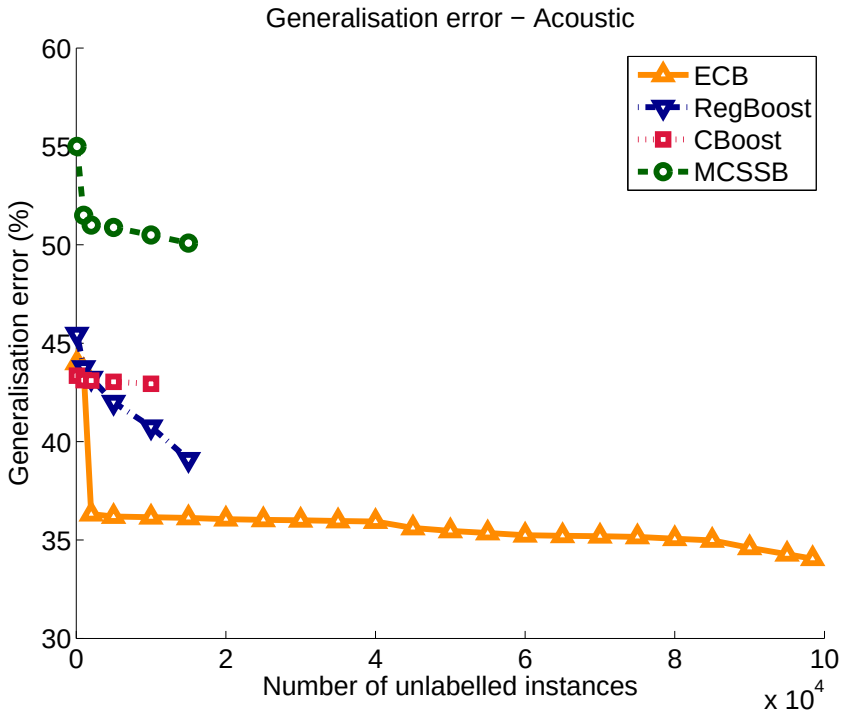


(a)

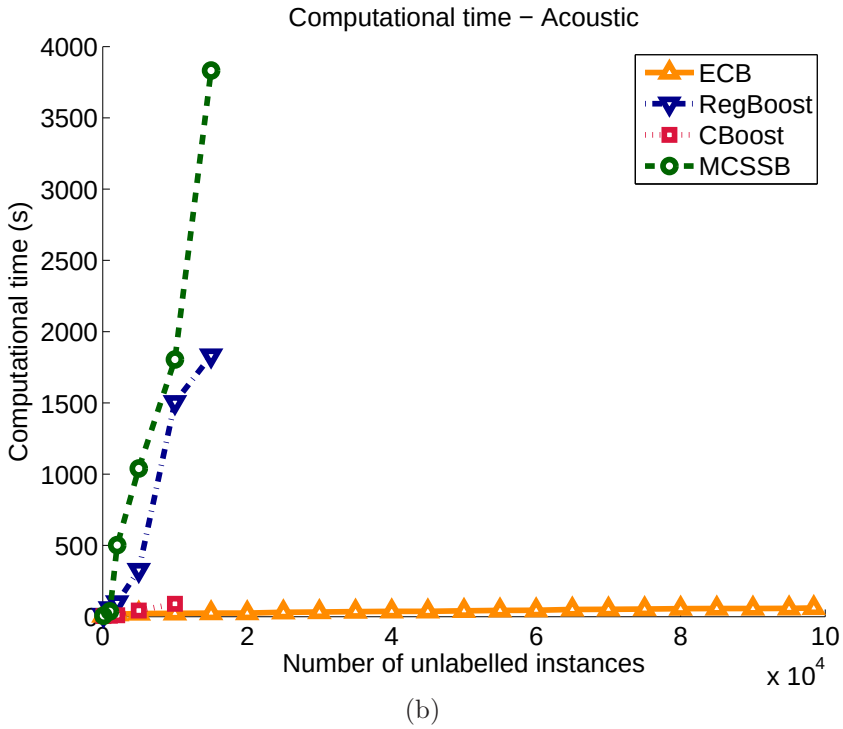


(b)

Figure 5.2: Plots of generalisation error (5.2a) and computational time (5.2b) versus the increase of the number of unlabelled instances for the SecStr dataset. Points used in one run are also employed in next runs.

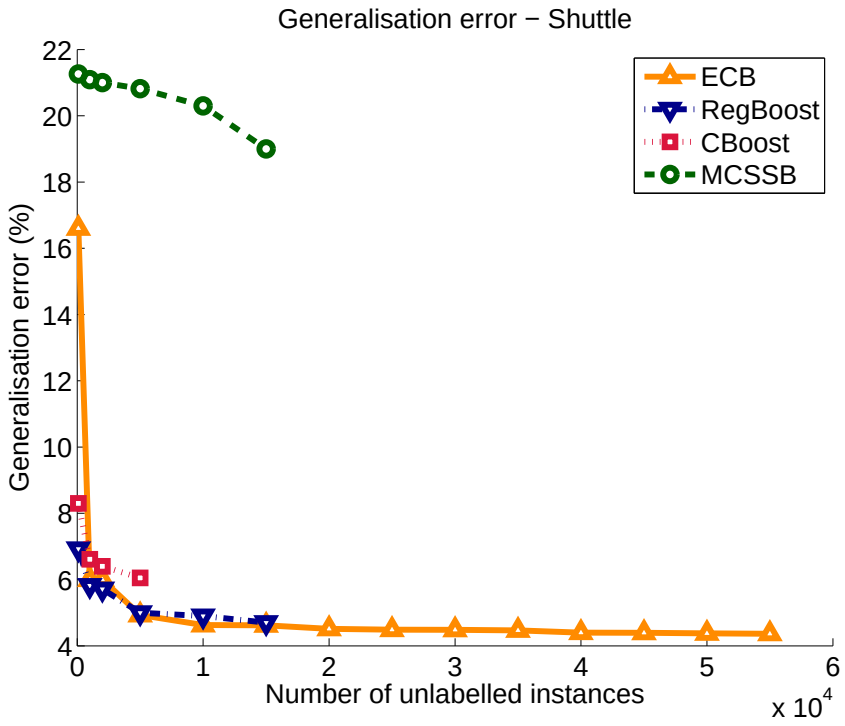


(a)

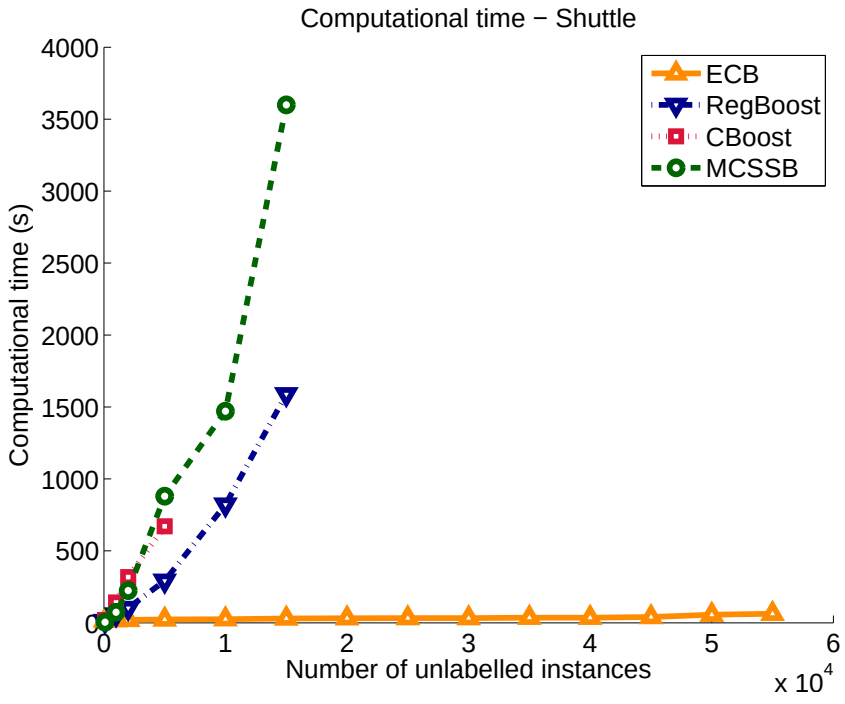


(b)

Figure 5.3: Plots of generalisation error (5.3a) and computational time (5.3b) versus the increase of the number of unlabelled instances for the Acoustic dataset. Points used in one run are also employed in next runs.

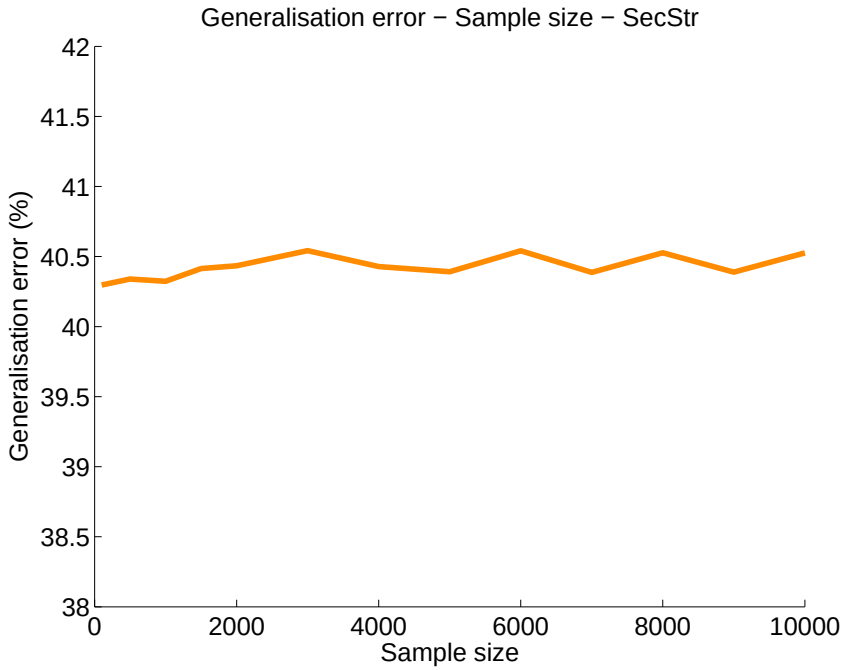


(a)

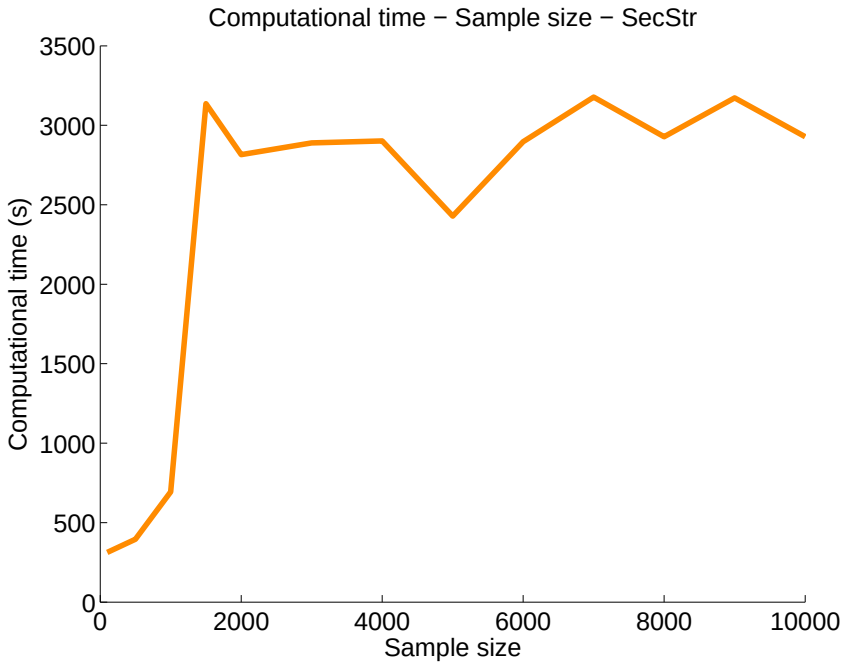


(b)

Figure 5.4: Plots of generalisation error (5.4a) and computational time (5.4b) versus the increase of the number of unlabelled instances for the Shuttle dataset. Points used in one run are also employed in next runs.

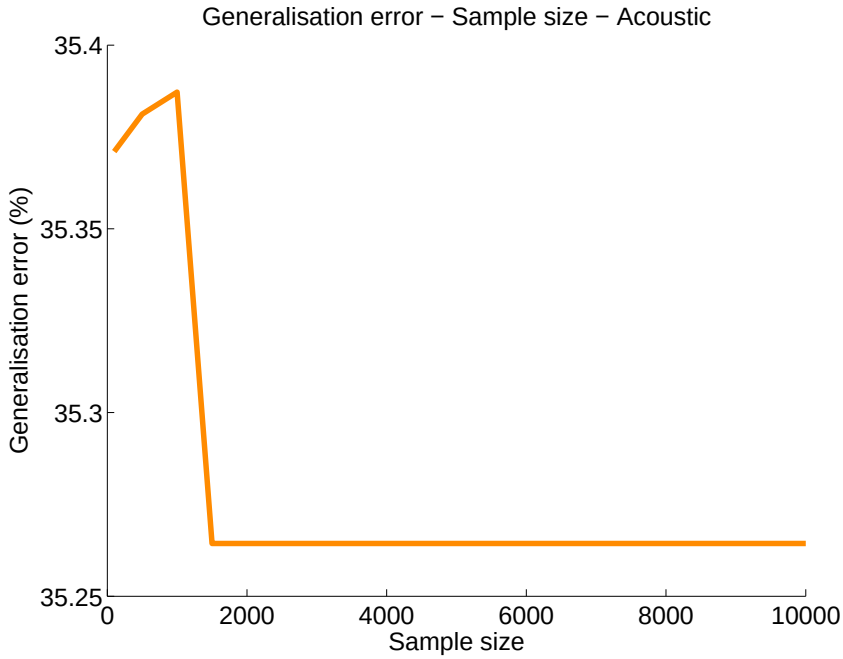


(a)

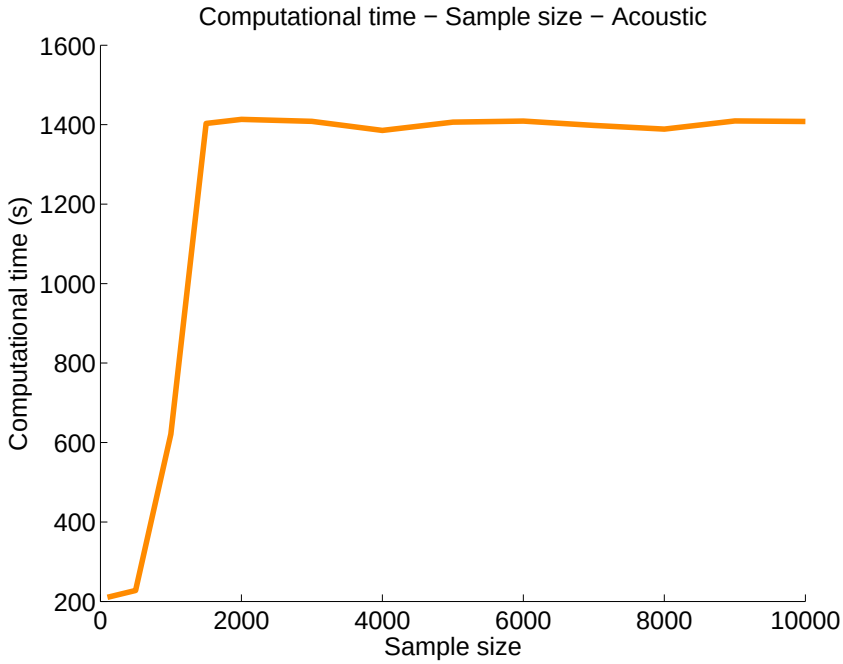


(b)

Figure 5.5: Performance of various sample sizes.

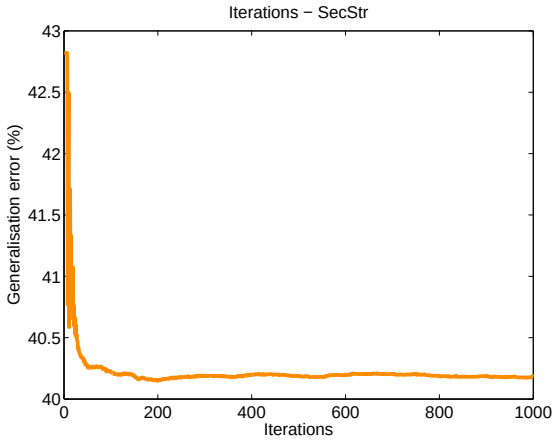


(a)

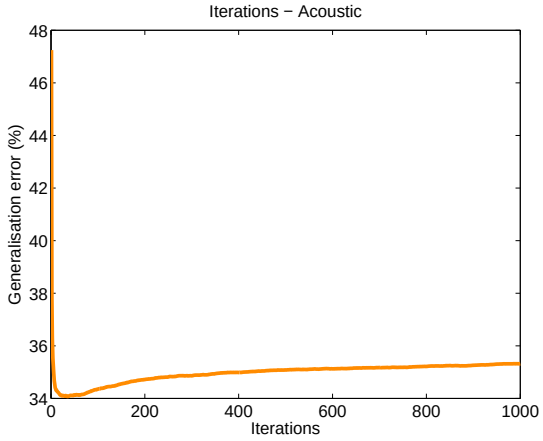


(b)

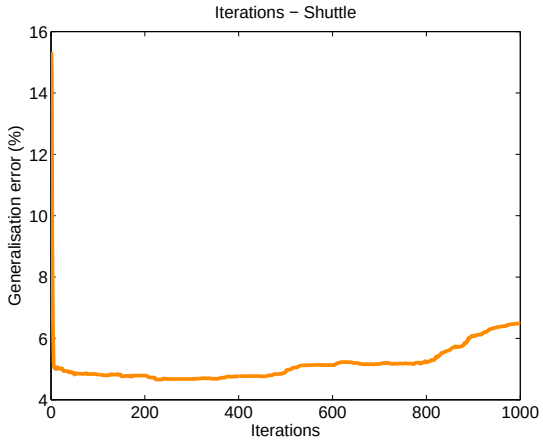
Figure 5.6: Performance of various sample sizes.



(a)



(b)



(c)

Figure 5.7: Plot of generalisation error and number of iterations (number of base learners) in ECB. Sample size was fixed to $B = 100$. Such a plot shows that ECB reaches its minimum test error with a small number of base learners despite the small amount of sampled data.

classifiers that make use such technique, ClusterReg and CBoost, could overcome the misleading structure.

Manifold-based algorithms are expected to deliver better generalisation for the Digit1 dataset [Chapelle et al., 2006]. With 10 labelled instances, we observed that manifold-based methods are indeed more accurate, in general, than cluster-based classifiers since the use of the structure of unlabelled data has more influence in the presence of fewer labelled instances. With 100 labelled instances, ECB obtained better generalisation ability than other cluster-based techniques, except for Data-Dependent Regularisation, ClusterReg and CBoost; which is explained by the use of approximate neighbours.

Both cluster and manifold assumptions are expected to hold for USPS dataset [Chapelle et al., 2006]. However, with 10 and 100 labelled instances, manifold-based algorithms delivered best performance (QC+CMN, Discrete Regularisation and Laplacian RLS). ECB obtained comparable accuracy with cluster-based methods for both amount of labelled data. Such results might indicate that, in fact, the manifold present in the data is more meaningful for classification.

The structures of COIL, BCI and Text datasets are unknown. Nonetheless, ECB yielded competitive performance among cluster-based and manifold-based algorithms on these real-world datasets with 10 and 100 labelled instances, although ClusterReg produced better generalisation in BCI when compared to both manifold and cluster-based algorithms. This might indicate that, in this case, ECB was able to properly use the information from the few labelled instances.

As expected, since ECB uses a sampling approach and approximate nearest neighbours, it did not obtain the best results among all classifiers in the transductive setting. However it produced comparable generalisation ability to other cluster-based methods when the cluster assumption holds.

In the inductive setting, we used only real-world datasets (Table 3.5) and their struc-

ture is unknown. When compared to MCSSB and RegBoost, ECB was statistically superior for most datasets across all amounts of labelled instances (Tables 5.4, 5.5 and 5.6). This fact indicates that our method could take advantage of its robustness to fewer labelled instances and overlapping classes. The decision boundary generated by ECB might have not been severely affected by the position of labelled points in a given high-density region. Such a performance in real-world datasets might be explained by the use of semi-supervised base learners. Such base learners are able to seek the neighbourhood for an appropriate label of an unlabelled instance and might recover from an incorrect pseudo-label assigned to such a point.

Both MCSSB and RegBoost implement all three semi-supervised assumptions. In cases where there is a clear cluster structure, the quality of the decision boundary generated by these algorithms might be limited by the search for a manifold in such datasets. In such situations, methods specialised in finding cluster structures may yield significantly superior generalisation performance, as shown in Tables 5.4, 5.5 and 5.6.

When compared to ClusterReg, ECB was able to significantly improve in many datasets with 10% and 20% of labelled data. Therefore, our ensemble approach was able to recover from errors of base classifiers, despite the use of approximate nearest neighbours. However, for 5% of labelled instances, ECB only statistically improved over ClusterReg in 5 datasets. This fact might indicate that ClusterReg could find appropriate decision boundaries and ECB was not able to deliver further improvements.

ECB obtained similar generalisation ability to CBoost across the majority of datasets. Despite the use of approximation techniques to calculate nearest neighbours and the sampling procedure, ECB was successful on relatively small real-world datasets when compared to algorithms that compute exact nearest neighbours and uses all data available at each iteration.

Apart from comparing the accuracy of ECB with state-of-the-art methods, we per-

formed experiments with datasets with tens of thousands instances to evaluate the scalability of ECB. In Figures 5.2, 5.3 and 5.4, we show the generalisation ability and efficiency of ECB along with MCSSB, RegBoost and CBoost across different amounts of unlabelled data. As depicted in Figure 5.2a, all methods reduce their test error with the increase of the number of unlabelled instances, which might denote the usefulness of unlabelled data in such dataset.

Regarding scalability, MCSSB, RegBoost and CBoost fail with a few thousands of instances (as shown in Figure 5.2b). MCSSB updates each instance weight with the consideration of all other unlabelled points, that is, it uses all instances to assign the pseudo-label of an unlabelled instance. This update leads to a quadratic growth of computational time with respect to the number of unlabelled points. Moreover, MCSSB stores a $N \times N$ similarity matrix. Such facts cause the algorithm to fail due to either memory shortage or time usage.

RegBoost requires the computation of exact nearest neighbours, which involves the use of a $N \times N$ distance matrix. As indicated by Figures 5.2b, 5.3b and 5.4b, such an algorithm starts to demand virtual memory with small amounts of data, which leads to a high increase in computation time at each step of the graph. Similar to MCSSB, RegBoost fails due to excessive running time and memory consumption.

Similarly, in Figure 5.3, the algorithms reduce their generalisation error with larger amounts of unlabelled data. However, as depicted in Figure 5.3b, only ECB was able to handle the full dataset. In Figure 5.4a, MCSSB did not deliver comparable accuracy to other algorithms. This fact may indicate that the effectiveness of its decision boundary was affected by the use of the manifold assumption when there might be a clear cluster structure in the dataset. Similar to the case of the Acoustic dataset, only ECB showed good scalability for Shuttle dataset (Figure 5.4b).

As shown in Figures 5.2b, 5.3b and 5.4b, the time requirement of ECB grows linearly

with the number of unlabelled instances. As depicted in Figures 5.2a, 5.3a and 5.4a, ECB can also produce comparable results with existing algorithms.

It is important to notice that the error of ECB is usually lower than that of CBoost, as shown in Figures 5.2a, 5.3a and 5.4a. The use of sampled data in each iteration of ECB leads to less trained base learners in comparison with CBoost. This early stop can be interpreted as a regularisation mechanism of the learning algorithm [Bishop, 2006]. This fact might indicate that ECB is able to avoid overfitting by stopping its training earlier than CBoost with the use of a sample of instances for each base classifier. In contrast, CBoost uses all available instances, which can lead further boosting iterations and might overfit the training points.

Based on the previous results, one could conclude that the employed clustering algorithm, LSC, is suitable for large datasets, delivering good partitions efficiently without compromising memory usage. The approximation technique increases efficiency in terms of both time and memory, which tackles the drawbacks of RegBoost and MCSSB with respect to high memory consumption (such drawbacks also have an impact on the execution time due to the overhead caused by accessing virtual memory). The sampling procedure also greatly reduces time complexity and allows training with large datasets in reasonable time. In fact, our experiments confirmed that the proposed method is suitable for large-scale datasets. Table 5.8 summarises time and memory complexities of the methods used in our experiments. It is important to highlight that $M \ll U$, $S \ll U$ and $V \ll U$. For example, for the Acoustic dataset used in Figure 5.7b, $U = 98428$, $S = 200$, $M = 100$ and $V = 30$.

Algorithms	Time	Memory
MCSSB	$\mathcal{O}(T_{base}CU^2 + TS^3)$	$\mathcal{O}(N^2)$
RegBoost	$\mathcal{O}(VCN \log N + CT_{base}S^3 + TCVU)$	$\mathcal{O}(N^2)$
CBoost	$\mathcal{O}(VCN \log N + T_{base}C^2M^2U + TCVU)$	$\mathcal{O}(N^2)$
ECB	$\mathcal{O}(T_{base}C^2M^2S + TCVU)$	$\mathcal{O}(VU)$

Table 5.8: Summary of the time and memory complexities.

In order to evaluate the sensitivity of ECB to the sample size B regarding accuracy and efficiency, Figures 5.5 and 5.6 present the generalisation error and CPU time for the SecStr and Acoustic datasets, respectively, for different amounts of sampled data. The amount of unlabelled data needed for SecStr is small, as shown in Figure 5.5a, which denotes that such a dataset does not possess a clear cluster structure. Hence, labelled data will be more important for the training algorithm. In contrast, ECB could improve its generalisation ability with larger amounts of sampled data for the Acoustic dataset (Figure 5.6a).

As shown in Figures 5.5b and 5.6b, the computational time stabilizes when the sample size reaches the limit of the number of hidden nodes in the RBFN. This behaviour is expected since the number of centres employed in RBFN increases with the sample size until it reaches a limit (in this case, 2000 hidden nodes).[1]

We also plot the generalisation error throughout 1000 iterations (base learners) of ECB on SecStr, Acoustic and Shuttle datasets (Figures 5.7a, 5.7b and 5.7c, respectively) without a termination criterion. We verify that the proposed algorithm converges with a small number of base learners, despite the small number of sampled instances, $B = 100$, at each iteration. As expected, the algorithm starts to overfit in later iterations. This figure suggests that ECB can be successfully used for large-scale datasets without compromising the execution time with a large number of base learners.

5.6 Theoretical discussion on Efficient Cluster-based Boosting

In the ECB algorithm, a subset $\mathbf{E}$ of size B is sampled from $\mathbf{U}$ at each iteration with replacement. A total T iterations will sample $T \times B$ points from the unlabelled set. In

[1]The complexity of RBFN grows quadratically with the number of centres and we limit such a parameter.

practical applications, it is necessary to know the number of distinct points presenting in the total TB points. The following will calculate the ratio of distinct points vs. the total points U in ECB.

Let ξ_{ij} be the binary indicator that the unlabelled data point $i \in \{1, \ldots, U\}$ was selected as the sampled point $j \in \{1, \ldots, k\}$, where $k = TB$. Since the sampling is with replacement, ξ_{ij} is an independent Bernoulli trial with the success probability $1/U$. Thus, the number of times point i was sampled,

$$\chi_i = \sum_{j=1}^{k} \xi_{ij},$$

has a Binomial$(k, 1/U)$ distribution. Therefore, the probability that the particular point i is not sampled, $P(\chi_i = 0)$, is calculated from the binomial mass function as

$$P(\chi_i = 0) = (1 - 1/U)^k.$$

Point i that is not sampled $\Pi_i = I(\chi_i = 0)$, is a Bernoulli trial with success probability $(1 - 1/U)^k$. The expected proportion of the population that is not sampled is

$$m_k = E\left(\frac{1}{U}\sum_{i=1}^{U}\Pi_i\right) = \frac{1}{U}\sum_{i=1}^{U}E(\Pi_i) = (1 - 1/U)^k. \tag{5.11}$$

Based on Equation (5.11), ECB will sample

$$distinct = 1 - (1 - 1/U)^{TB} \ * 100\%$$

distinct samples in the total sampled TB points.

For example, for the Acoustic dataset, $U = 98428$, $B = 1500$, $T = 200$, and the distinct ratio is 95%.

Although the sampling approach cannot cover 100% unlabelled data, and sometimes it can cover only a small part of unlabelled data, ECB can still achieve a promising generalization ability due to the fact that the generated ensemble member z_{nj} is guaranteed to be parallel to the functional gradient, which is beneficial for the generalization performance [Friedman, 2001]. That is, in the step (11) of Algorithm 4, ECB chooses $\{z_{nj}\}_1^{B+L}$ that is most parallel to the residual $-\left[\frac{\partial \mathcal{L}(\mathbf{F}_n, \mathbf{y}_n)}{\partial Z_{nj}^{t-1}}\right] \in R^{B+L}$ as the ensemble member. This z_{nj} is the most highly correlated solution with $-\left[\frac{\partial \mathcal{L}(\mathbf{F}_n, \mathbf{y}_n)}{\partial Z_{nj}^{t-1}}\right]$ over the data distribution.

Based on Friedman [2001] and confirmed by Figure 5.7, this formulation often leads to comparable generalization to state-of-the-art algorithms, which might be the reason why ECB can achieve a good performance even with a small number of iterations.

5.7 Conclusions

In this Chapter, we addressed the research question in Section 1.5.3. As demonstrated in our experiments, existing classifiers cannot handle large datasets. Therefore, we introduced an efficient boosting algorithm for multi-class SSC. By reducing time and memory complexities of CBoost, such a method allows SSC to be performed in large-scale datasets.

We proposed a cluster-based boosting algorithm for large multi-class datasets. We reviewed the literature and pointed out that state-of-the-art algorithms cannot be applied to large datasets due to their time requirements. Our proposed method handles large datasets by employing an efficient clustering algorithm, an approximation technique for nearest neighbours to avoid the computation of pairwise distance matrix and a sampling procedure to reduce the training set of base learners. Such improvements reduce both time and memory requirements for ECB.

We designed three experimental settings: transductive, inductive, and large-scale datasets. In both transductive and inductive settings, ECB could deliver comparable generalisation ability to state-of-the-art algorithms. In our analysis on large-scale datasets,

we evaluated and validated the scalability of the proposed algorithm for both binary and multi-class datasets.

The experimental analysis confirmed that (i) the use of uniform sampling along with approximation techniques for nearest neighbours and large-scale clustering can increase the efficiency of ECB and maintain comparable generalisation performance with other methods; (ii) ECB inherits the advantages of ClusterReg and presents robustness to the position of labelled data in a cluster and, (iii) as in CBoost, by using semi-supervised base learners, ECB is also robust to incorrect pseudo-label assignments during training.

CHAPTER 6

Conclusions and Future Work

In this Thesis, we studied ensemble techniques for multi-class semi-supervised classification. Firstly, we introduced ClusterReg, a multi-class classifier that possesses robustness to overlapping class and to the few labelled instances on low-density regions. Then, we used the concept of cluster regularisation of ClusterReg to develop a robust boosting technique, CBoost, which uses a semi-supervised classifier, ClusterReg, as base learners. Unlike existing methods, such an ensemble algorithm is able to overcome errors in pseudo-labels assignments. And, finally, we introduced ECB, an efficient and effective multi-class ensemble algorithm that uses approximate nearest neighbours and sampling to extend CBoost in order to allow SSC to be performed in large-scale datasets.

In this Chapter, we highlight the major contributions of this Thesis. We also describe further investigations that might extend this work. Our contributions are reported in Section 6.1 and the future works are presented in Section 6.2.

6.1 Contributions

We summarise the major contributions of this Thesis as follows.

- **Algorithms for multi-class semi-supervised classification**.

Most existing SSC methods in literature are binary classifiers, therefore such algorithms depend on suboptimal decomposition methods to perform multi-class classification and are prone to problems of imbalanced classification and different output scales of binary classifiers [Valizadegan et al., 2008]. Thus, we developed ensemble techniques that are inherently multi-class. And the base learners also perform multi-class classification.

- **Semi-supervised classification with cluster regularisation**.

  Most cluster-based methods attempt to find the largest margin between high-density regions (clusters). When overlapping clusters are present, with sparse labelled instances on their borders, such classifiers may not find the correct gap between classes to generate a decision boundary, although these inherent clusters might be easily identified.

  In this sense, we introduced in Chapter 3 a semi-supervised multi-class classifier (ClusterReg). ClusterReg uses soft clustering in a regularisation mechanism in order to avoid generating decision boundaries that traverses high-density regions. Due to such a regularisation technique, the proposed method is robust to overlapping classes and to the few labelled instances in a cluster.

  Therefore, in order to investigate ensemble learning for multi-class SSC, we successfully addressed relevant weaknesses in state-of-the-art algorithms (research question raised in the Section 1.5.1) with the introduction of ClusterReg and employed such a method as the base learner in the proposed ensemble algorithms.

- **A Fully semi-supervised ensemble for multi-class classification**.

  Semi-supervised ensembles can employ unlabelled data in their training algorithm (at ensemble level) and/or in the training of base learners (at base learner level). Therefore, the design of an ensemble and, hence, its performance may be strongly

affected by the use of unlabelled data at either of such levels. To our knowledge, such an issue was not addressed in literature. Thus, we presented a study on the usefulness of employing unlabelled data at both ensemble and base learner levels, comparing to using such data at ensemble level only.

In this sense, this Thesis investigated such an issue (research question raised in the Section 1.5.2) in Chapter 4. We presented a study on the usefulness of employing unlabelled data at both ensemble and base learner levels, comparing such an approach to using unlabelled instances at the ensemble level only. We proposed the Cluster-based Boosting (CBoost) algorithm for multi-class classification. Such a method extends ClusterReg and, unlike other semi-supervised ensembles in the literature, is composed of semi-supervised base classifiers.

CBoost extends ClusterReg by employing gradient boosting to improve ClusterReg's predictive accuracy. CBoost inherits ClusterReg's robustness to overlapping classes and to the position of few labelled instances in a given cluster when the cluster assumption holds. The proposed method is especially designed for multi-class problems, avoiding depending on decomposition techniques. And it uses an effective semi-supervised base learner, ClusterReg. Such a base classifier considers the neighbourhood of an instance when learning its pseudo-label. This approach leads to a more robust ensemble, since base learners may be able to overcome possible incorrect pseudo-labels.

- **Efficient boosting for semi-supervised classification**.

  Due to the available number of unlabelled data, a semi-supervised training set can often have tens of thousands of instances. Therefore, the learning algorithms must be able to handle such large datasets. However, existing ensemble algorithms are unable to handle typical large-scale datasets.

Therefore, in Chapter 5, we proposed the Efficient Cluster-based Boosting (ECB) algorithm. ECB extends CBoost by employing an efficient clustering algorithm, an approximation technique for nearest neighbours to avoid the computation of pairwise distance matrix and a sampling procedure to reduce the training set of base learners.

Chapter 5 demonstrated that (i) the use of uniform sampling along with approximation techniques for nearest neighbours and large-scale clustering can increase the efficiency of ECB and maintain comparable generalisation performance with other methods; (ii) ECB inherits the advantages of ClusterReg and presents robustness to the position of labelled data in a cluster and, (iii) as in CBoost, by using semi-supervised base learners, ECB is also robust to incorrect pseudo-label assignments during training.

Such improvements reduce both time and memory requirements for ECB, and maintain comparable generalisation ability with state-of-the-art algorithms. Therefore, ECB enables semi-supervised classification for large-scale datasets.

6.2 Future work

The classifiers proposed in this Thesis and described in literature use local-search training algorithms. Hence, such methods may find only local optima and may not produce the global optimum of their loss functions. In this sense, since population-based optimisers have been widely employed in the supervised context to train classifiers [Garcia-Pedrajas et al., 2005, Khare et al., 2005, Nguyen et al., 2006, Kim and Cho, 2008], we aim to investigate the use of such global-search methods for semi-supervised classification. We expect such training algorithms with global search to deliver better solutions, when compared to local-search methods, for our proposed classifiers and, hence, to produce higher generalisation accuracy.

ClusterReg, CBoost and ECB rely on the parameter λ to compromise between supervised loss and semi-supervised regularisation in our loss function. Thus, our future work is also to employ model selection techniques [Wasserman, 2000, Bishop, 2006], such as Bayesian techniques, to optimise such a trade-off. Multi-objective evolutionary algorithms [Chen et al., 1999, Deb et al., 2000, Chen and Yao, 2010] are also an alternative.

The proposed loss function is based on cluster regularisation. Since we may find good compromises between supervised and cluster-based regularisation with model selection techniques, we can also include a manifold regularisation term in our loss function. The main contribution would be the automatic selection of which SSL assumption (terms in the loss function) should have a higher impact on the training of a classifier for a given dataset with an unknown underlying structure. Such a systematic selection of assumptions would help to solve a limitation of existing methods in literature: the dependency on trade-off parameters to balance multiple assumptions.

List of References

M. Belkin and P. Niyogi. Using manifold stucture for partially labeled classification. In *Advances in Neural Information Processing Systems (NIPS)*, pages 953–960, 2003.

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 7:2399–2434, 2006.

K. P. Bennett and A. Demiriz. Semi-supervised support vector machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 368–374. MIT Press, 1998.

K. P. Bennett, A. Demiriz, and R. Maclin. Exploiting unlabeled data in ensemble methods. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 289–296, New York, USA, 2002. ACM.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pages 19–26, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 92–100. Morgan Kaufmann Publishers, 1998.

U. Brefeld, T. Gärtner, T. Scheffer, and S. Wrobel. Efficient co-regularised least squares regression. In *Proceedings of the 23rd International Conference on Machine learning (ICML)*, pages 137–144, New York, USA, 2006. ACM.

V. Castelli and T. Cover. The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Transactions on Information Theory*, 42(6):2102 –2117, November 1996.

V. Castelli and T. M. Cover. On the exponential value of labeled samples. *Pattern Recognition Letters*, 16(1):105–111, 1995.

S. Chakraborty. Bayesian semi-supervised learning with support vector machine. *Statistical Methodology*, 8(1):68–82, 2011.

O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, 2006.

H. Chen and X. Yao. Multiobjective neural network ensembles based on regularized negative correlation learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(12):1738–1751, December 2010.

K. Chen and S. Wang. Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):129–143, January 2011.

K. Chen, L. Xu, and H. Chi. Improved learning algorithms for mixture of experts in multiclass classification. *Neural Networks*, 12(9):1229–1252, 1999.

C. Constantinopoulos and A. Likas. Semi-supervised and active learning with the probabilistic rbf classifier. *Neurocomputation*, 71(13-15):2489–2498, 2008.

F. d'Alché Buc, Y. Grandvalet, and C. Ambroise. Semi-supervised marginboost. In *Advances in Neural Information Processing Systems (NIPS)*, pages 553–560, 2002.

R. Dara, S. Kremer, and D. Stacey. Clustering unlabeled data with soms improves classification of labeled real-world data. In *Proceedings of the 2002 International Joint Conference on Neural Networks (IJCNN)*, volume 3, pages 2237–2242, Honolulu, HI, USA, 2002.

V. R. de Sa. Learning classification with unlabeled data. In *Advances in Neural Information Processing Systems (NIPS)*, pages 112–119, 1993.

K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: Nsga-ii. In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature (PPSN)*, pages 849–858, London, UK, 2000. Springer-Verlag.

O. Delalleau, Y. Bengio, and N. L. Roux. Large-scale algorithms. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-supervised learning*, chapter 18. MIT Press, 2006.

A. Demiriz, K. Bennett, and M. J. Embrechts. Semi-supervised clustering using genetic algorithms. In *Artificial Neural Networks in Engineering (ANNIE)*, pages 809–814. ASME Press, 1999.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

M. F. Duarte and Y. H. Hu. Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):826–838, July 2004.

N. A. C. Dunne Campbell, R. A. Dunne. On the pairing of the softmax activation and cross entropy penalty functions and the derivation of the softmax activation function. In *Proceedings of the 8th Australasian Conference on Neural Networks*, pages 181–185, 1997.

Y. Engel, S. Mannor, and R. Meir. The kernel recursive least-squares algorithm. *IEEE Transactions on Signal Processing*, 52(8):2275–2285, 2004.

A. Frank and A. Asuncion. UCI machine learning repository, 2010. URL http://archive.ics.uci.edu/ml.

J. H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

N. Garcia-Pedrajas, C. Hervas-Martinez, and D. Ortiz-Boyer. Cooperative coevolution of artificial neural network ensembles for pattern classification. *IEEE Transactions on Evolutionary Computation*, 9(3):271–302, June 2005.

Y. Grandvalet, F. d'Alché Buc, and C. Ambroise. Boosting mixture models for semi-supervised learning. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pages 41–48, London, UK, 2001. Springer-Verlag.

D. Gustafson and W. Kessel. Fuzzy clustering with fuzzy covariance matrix. In *Proceedings of the IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, volume 17, pages 761–766, San Diego, USA, 1978.

M. Hady and F. Schwenker. Co-training by committee: A new semi-supervised learning framework. In *Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 563–572, Pisa, Italy, December 2008.

M. F. A. Hady, F. Schwenker, and G. Palm. Semi-supervised learning for tree-structured ensembles of rbf networks with co-training. *Neural Networks*, 23(4):497–509, 2010.

C. W. Hsu and C. J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

J. Zhu, H. Zou and T. Hastie. Multi-class adaboost. *Statistics and Its Interface (Special issue on data mining and machine learning)*, 2:349–360, 2009.

T. Joachims. Learning to classify text using support vector machines. Master's thesis, Kluwer, 2002.

T. Joachims. Transductive learning via spectral graph partitioning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 290–297, 2003.

V. Khare, X. Yao, B. Sendhoff, Y. Jin, and H. Wersing. Co-evolutionary modular neural networks for automatic problem decomposition. In *IEEE Congress on Evolutionary Computation*, volume 3, pages 2691–2698, September 2005.

K.-J. Kim and S.-B. Cho. Evolutionary ensemble of diverse artificial neural networks using speciation. *Neurocomputing*, 71(7–9):1604–1618, 2008.

M. Kline and L. Berardi. Revisiting squared-error and cross-entropy functions for training neural network classifiers. *Neural Computing and Applications*, 14:310–318, December 2005.

Y. Liu and X. Yao. Ensemble learning via negative correlation. *Neural Networks*, 12(10): 1399–1404, 1999.

J. Malkin, A. Subramanya, and J. Bilmes. A semi-supervised learning algorithm for multi-layered perceptrons. Technical report, University of Washington Electrical Engineering (UWEE), 2009.

P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu. Semiboost: Boosting for semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2000–2014, 2009.

G. S. Mann and A. McCallum. Simple, robust, scalable semi-supervised learning via expectation regularization. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pages 593–600, New York, USA, 2007. ACM.

L. Z. Manor and P. Perona. Self-tuning spectral clustering. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS) 2004*, pages 1601–1608. MIT Press, 2004.

S. Melacci and M. Belkin. Laplacian support vector machines trained in the primal. *The Journal of Machine Learning Research*, 12:1149–1184, March 2011.

M. F. Moller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4):525–533, 1993.

M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application (VISSAPP)*, pages 331–340. INSTICC Press, 2009.

I. T. Nabney. Efficient training of rbf networks for classification. In *Ninth International Conference on Artificial Neural Networks (ICANN)*, volume 1, pages 210–215, Edinburgh, UK, 1999.

M. H. Nguyen, H. A. Abbass, and R. I. Mckay. A novel mixture of experts model based on cooperative coevolution. *Neurocomputing*, 70(1-3):155–163, 2006.

K. P. Nigam. *Using unlabeled data to improve text classification.* PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2001.

K. Plunkett and J. Elman. *Exercises in Rethinking Innateness: A Handbook for Connectionist Simulations*. MIT Press, 1997.

J. Ratsaby, , J. Ratsaby, and S. S. Venkatesht. Learning from a mixture of labeled and unlabeled examples with parametric side information. In *Proceedings of the Eighth Annual Conference on Computational Learning Theory*, pages 412–417. ACM Press, 1995.

A. Saffari, C. Leistner, and H. Bischof. Regularized multi-class semi-supervised boosting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 967–974, June 2009.

A. Saffari, C. Leistner, M. Godec, and H. Bischof. Robust multi-view boosting with priors. In *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, pages 776–789, Heraklion, Greece, 2010. Springer-Verlag.

N. Seliya and T. M. Khoshgoftaar. Software quality analysis of unlabeled program modules with semisupervised clustering. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 37(2):201–211, March 2007a.

N. Seliya and T. M. Khoshgoftaar. Software quality estimation with limited fault data: a semi-supervised learning perspective. *Software Quality Control*, 15(3):327–344, 2007b.

N. Seliya, T. M. Khoshgoftaar, and S. Zhong. Semi-supervised learning for software quality estimation. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 183–190, November 2004.

V. Sindhwani, S. S. Keerthi, and O. Chapelle. Deterministic annealing for semi-supervised kernel machines. In *Proceedings of the 23rd International Conference on Machine learning (ICML)*, pages 841–848. ACM, 2006.

R. G. F. Soares, H. Chen, and X. Yao. Semisupervised classification with cluster regularization. *IEEE Transactions on Neural Networks and Learning Systems*, 23(11): 1779–1792, November 2012.

E. Song, D. Huang, G. Ma, and C.-C. Hung. Semi-supervised multi-class adaboost by exploiting unlabeled data. *Expert Systems with Applications*, 38(6):6720–6726, 2011.

P. Sun and X. Yao. Sparse approximation through boosting for learning large scale kernel machines. *IEEE Transactions on Neural Networks*, 21(6):883–894, June 2010.

M. Szummer and T. Jaakkola. Partially labeled classification with markov random walks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 945–952. MIT Press, 2002.

K. Tang, M. Lin, F. L. Minku, and X. Yao. Selective negative correlation learning approach to incremental learning. *Neurocomputing*, 72(13–15):2796–2805, 2009.

P. M. Vaidya. An o(n log n) algorithm for the all-nearest-neighbors problem. *Discrete and Computational Geometry*, 4(1):101–115, 1989.

H. Valizadegan, R. Jin, and A. K. Jain. Semi-supervised boosting for multi-class classification. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases - Part II (ECML/PKDD)*, pages 522–537, Berlin, Heidelberg, 2008. Springer-Verlag.

V. Vapnik. *Statistical learning theory*. Wiley-Interscience, 1998.

S. Wang, K. Tang, and X. Yao. Diversity exploration and negative correlation learning on imbalanced data sets. In *Proceedings of the 2009 International Joint Conference on Neural Networks (IJCNN)*, pages 3259–3266, Atlanta, GA, USA, June 2009.

Y. Wang, S. Chen, and Z.-H. Zhou. New semi-supervised classification method based on modified cluster assumption. *IEEE Transactions on Neural Networks and Learning Systems*, 23(5):689–702, May 2012.

L. Wasserman. Bayesian model selection and model averaging. *Journal of Mathematical Psychology*, 44(1):92–107, 2000.

X. Chen and D. Cai. Large scale spectral clustering with landmark-based representation. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 313–318, 2011.

R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, May 2005.

X. Yao and Y. Liu. Evolving neural network ensembles by minimization of mutual information. *International Journal of Hybrid Intelligent Systems*, 1(1–2):12–21, 2004.

G. Yu, G. Zhang, Z. Yu, C. Domeniconi, J. You, and G. Han. Semi-supervised ensemble classification in subspaces. *Applied Soft Computing*, 12(5):1511–1522, 2012.

M. Zanda and G. Brown. A study of semi-supervised generative ensembles. In *Proceedings of the 8th International Workshop on Multiple Classifier Systems (MCS)*, pages 242–251, Berlin, Heidelberg, 2009. Springer-Verlag.

L. Zheng, S. Wang, Y. Liu, and C.-H. Lee. Information theoretic regularization for semi-supervised boosting. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1017–1026, New York, USA, 2009. ACM.

Z.-H. Zhou. When semi-supervised learning meets ensemble learning. In *Proceedings of*

the 8th International Workshop on Multiple Classifier Systems (MCS), pages 529–538, Berlin, Heidelberg, 2009. Springer-Verlag.

Z.-H. Zhou and M. Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1529–1541, 2005.

X. Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin, 2008.

X. Zhu. Tutorial on semi-supervised learning. Machine Learning Summer School, University of Chicago. Chicago, USA, 2009.

X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Carnegie Mellon University (CMU), 2002.

X. Zhu and J. Lafferty. Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *Proceedings of the 22nd International Conference on Machine learning (ICML)*, pages 1052–1059, New York, USA, 2005. ACM.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 912–919, 2003.