Investigation of training data issues in ensemble classification based on margin concept : application to land cover mapping

Wei Feng

Université BORDEAUX MONTAIGNE

Bordeaux INP AQUITAINE

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ BORDEAUX MONTAIGNE

Discipline : Science et Technologie

Spécialité : Informatique

École doctorale Montaigne Humanités (ED 480)

présentée par

Wei FENG

pour obtenir le grade de :

Docteur de l'Université Bordeaux Montaigne

Investigation des problèmes des données d'apprentissage en classification ensembliste basée sur le concept de marge. Application à la cartographie d'occupation du sol

Investigation of training data issues in ensemble classification based on margin concept. Application to land cover mapping

---

Mme. Samia Boukir — Directrice de Thèse

Soutenance prévue le 19 juillet 2017 devant le jury composé de :

Mme	Samia Boukir, Professeur, Bordeaux INP	Examinateur
M.	Cyril de Runz, Maître de Conférences HDR, Université de Reims	Rapporteur
M.	Robin Genuer, Maître de Conférences, Université de Bordeaux	Examinateur
M.	Christian Germain, Professeur, Bordeaux Sciences Agro	Examinateur
M.	Gilles Richard, Professeur, Université Toulouse III	Rapporteur

Acknowledgments

At the end of this exciting and frustrating process, having finished this dissertation, I would like to take the opportunity to express my gratitude to all who have supported me through this long journey.

I express my profound thanks to my supervisor Samia Boukir, Professor of Computer Science at IPB (Bordeaux Institute of Technology), for her supervision and correction of my thesis.

I would also like to thank Gilles Richard, Professor of Computer Science at Paul Sabatier University (UPS-Toulouse 3), and Cyril de Runz, Associate-Professor of Computer Science at Reims Champagne-Ardenne University, for having accepted to review my thesis, as well as Robin Genuer, Associate-Professor at in Statistics at the University of Bordeaux, and Christian Germain, Professor of Computer Science at Bordeaux Sciences Agro, for participating in my thesis committee.

I am grateful to China Scholarship Council (CSC) for providing me with a scholarship for three years.

I would like to express my gratitude to all my friends for their concern and encouragement especially when I was ill and in trouble.

Finally, my sincere thanks go to my family: my parents and my wonderful brother, for their encouragement, attention and unconditional support in academic and non-academic issues, as well as my boyfriend Qiang, for his infinite concern, encouragement and help in my life and work. I would not have accomplished my PhD without your support and patience.

Résumé

L'apprentissage supervisé est un domaine de recherche majeur en apprentissage automatique. La qualité et la quantité des données d'apprentissage sont très importantes en apprentissage supervisé. Cependant, dans la plupart des cas, les échantillons utilisés pour l'apprentissage d'un modèle de classification sont insatisfaisants. Les données bruitées, déséquilibrées, de grande dimension ou complexes sont des défis majeurs en apprentissage automatique. Pour la plupart des classifieurs, la performance de classification décroit plus ou moins selon le niveau de bruit et le taux de déséquilibre.

L'apprentissage d'ensemble est une méthode efficace pour développer des systèmes de classification précis. Ce paradigme d'apprentissage est attrayant car il est capable de booster des classifieurs faibles, dont la performance est légèrement meilleure qu'une prédiction aléatoire, en des agrégations de classifieurs performantes capables d'effectuer des prédictions très précises. En outre, la capacité de généralisation d'un ensemble (ou classifieur multiple) est souvent plus forte que celle des classifieurs de base le composant. Le *boosting*, le *bagging* et les forêts aléatoires sont des méthodes d'apprentissage d'ensemble majeures. Les méthodes d'ensemble ont été appliquées avec succès dans de nombreuses applications du monde réel telles que le diagnostic médical et la cartographie d'occupation du sol.

La notion de marge, qui a été initialement utilisée pour développer la théorie des SVM (*Support Vector Machine*) et pour expliquer le succès du *boosting*, joue un rôle majeur de nos jours en apprentissage automatique. La marge d'ensemble est un concept clé en apprentissage d'ensemble. Elle a été utilisée aussi bien pour l'analyse théorique que pour la conception d'algorithmes d'apprentissage automatique. De nombreuses études ont montré que la performance de généralisation d'un classifieur ensembliste est liée à la distribution des marges de ses exemples d'apprentissage. Récemment, la marge d'ensemble a été utilisée pour l'échantillonnage de données déséquilibrées, la suppression d'erreurs d'étiquetage, la sélection de données, la sélection d'attributs et la conception de classifieurs.

Ce travail se focalise sur l'exploitation du concept de marge pour améliorer la qualité de l'échantillon d'apprentissage et ainsi augmenter la précision de classification de classifieurs sensibles au bruit, et pour concevoir des ensembles de classifieurs efficaces capables de gérer des données déséquilibrées.

Dans cette thèse, nous introduisons tout d'abord une définition alternative de la marge d'ensemble.

Une nouvelle définition de la marge d'ensemble

Nous proposons une nouvelle marge d'ensemble. C'est une version non supervisée d'une marge d'ensemble populaire, elle ne requière pas d'étiquettes de classe. Par conséquent, elle est potentiellement plus robuste au bruit puisqu'elle n'est pas affectée par les erreurs d'étiquetage. Une instance de forte marge est une instance qui a été classifiée par la majorité des classifieurs de base en la même classe. Plus la marge d'une instance est forte, plus la classification associée est fiable. De plus, la marge proposée a l'avantage par rapport à une marge supervisée d'être utilisable pour l'évaluation ou la conception de classifieurs en apprentissage d'ensemble semi ou non supervisé.

Notre marge d'ensemble, ainsi que trois autres marges d'ensemble, sont à la base de ce travail dont les contributions sont présentées dans ce qui suit :

Filtrage du bruit d'étiquetage utilisant la marge d'ensemble

Les données d'apprentissage mal étiquetées sont un défi majeur pour la construction d'un classifieur robuste que ce soit un ensemble ou pas. Une instance mal étiquetée est une instance dont la valeur attribuée et la valeur de l'étiquette ne sont pas compatibles. Les limitations des travaux relatifs actuels suggèrent clairement qu'une piste différente devrait être suivie pour le filtrage des données. La distribution des marges des données d'apprentissage reflète efficacement la performance d'un algorithme d'ensemble. Lorsqu'un modèle classifie correctement un jeu de données avec une forte probabilité, ces instances devraient avoir de fortes marges. La présence de bruit peut affaiblir la performance d'un classifieur et conduire à une distribution de marges plus faibles.

Pour gérer le problème d'étiquetage, une méthode d'identification et d'élimination du bruit d'étiquetage utilisant la marge d'ensemble est proposée. Elle est basée sur un algorithme existant d'ordonnancement d'instances erronées selon un critère de marge. Cette méthode peut atteindre un taux élevé de détection des données mal étiquetées tout en maintenant un taux de fausses détections aussi bas que possible. Elle s'appuie sur les valeurs de marge des données mal classifiées, considérant quatre différentes marges d'ensemble, incluant la nouvelle marge proposée. De plus, elle est étendue à la gestion de la correction du bruit d'étiquetage qui est un problème plus complexe.

La principale différence entre notre filtre de bruit d'étiquetage et ceux existants, basés aussi sur une approche ensembliste, est dans le fait qu'il n'adopte pas seulement le vote d'ensemble pour distinguer les données mal classifiées de celles qui sont bien classifiées, mais aussi, il prend en compte explicitement, à travers la marge d'ensemble, la probabilité que des instances mal classifiées soient identifiables comme du bruit. Ainsi, cette méthode pourrait aussi être considérée comme un filtre probabiliste. L'efficacité de nos méthodes d'élimination et de correction du bruit d'étiquetage, basées sur l'ordonnancement d'instances, est démontrée sur la classification de données. Une analyse comparative est menée par rapport au filtre basé sur le vote majoritaire, un filtre de bruit d'étiquetage ensembliste de référence. Deux types de bruit d'étiquetage artificiel, bruit aléatoire et bruit basé sur la matrice de confusion, sont utilisés dans nos expérimentations. Le bruit

basé sur la matrice de confusion affecte principalement les bordures de classes. Par conséquent, il est plus difficile à identifier que le bruit aléatoire.

Elimination du bruit d'étiquetage basée sur la marge d'ensemble

La première étape de notre méthode de suppression de bruit implique un classifieur ensembliste robuste : le *bagging* qui est construit en utilisant tout l'échantillon d'apprentissage. La marge de chaque instance d'apprentissage est alors calculée. Notre méthode ordonne les instances d'apprentissage mal classifiées selon leurs valeurs de marge. Plus la marge est élevée (en valeur absolue), plus la probabilité que l'instance correspondante mal classifiée soit mal étiquetée est forte. Les deux étapes suivantes de notre algorithme s'appuient sur un classifieur ensembliste sensible au bruit : le *boosting*. La seconde étape vise à sélectionner le meilleur échantillon d'apprentissage filtré, dont le bruit d'étiquetage aura été éliminé. Le taux de bruit d'étiquetage est estimé automatiquement par une procédure itérative qui supprime une quantité $\boldsymbol{M}$ (de 0 à progressivement 40% de la taille totale de l'échantillon d'apprentissage) d'instances potentiellement mal étiquetées ordonnées de l'échantillon d'apprentissage et évalue la précision de classification du *boosting*, construit avec l'échantillon d'apprentissage filtré, sur un échantillon de validation. Cette stratégie adaptative sélectionne alors l'échantillon d'apprentissage filtré qui a conduit à la précision maximale sur l'échantillon de validation. A la dernière étape, le *boosting* est impliqué encore pour évaluer la qualité de l'échantillon d'apprentissage filtré obtenu via une procédure d'évaluation de la précision de classification. KNN (*K-Nearest Neighbours*), un classifieur individuel sensible au bruit, est également utilisé (au lieu du *boosting*) dans les deux dernières étapes de notre filtre de données d'apprentissage mal étiquetées.

Une nouvelle version étendue de notre méthode d'élimination du bruit basée sur la marge d'ensemble est proposée. Ce filtre itératif s'appuie sur un calcul adaptatif des valeurs de marge de chaque instance d'apprentissage. Dans la version originale de notre algorithme, les marges d'apprentissage sont déterminées en une seule fois, dans la première étape. Les marges d'apprentissage étant à la base de notre procédure d'évaluation du bruit, une stratégie sensée consisterait à les mettre à jour à chaque étape d'élimination du bruit.

Correction du bruit d'étiquetage basée sur la marge d'ensemble

L'élimination du bruit peut écarter certaines données utiles, c'est pourquoi la correction automatique des instances d'apprentissage, identifiées comme mal étiquetées (instances mal classifiées de forte marge), est aussi tentée. La correction du bruit a engendré de meilleurs résultats que la simple élimination du bruit des données dans certains cas. Dans un schéma de correction de données, les instances mal étiquetées sont identifiées mais au lieu de les supprimer elles sont corrigées en substituant les étiquettes erronées par des étiquettes plus appropriées. Les étiquettes des instances mal étiquetées les plus probables sont modifiées en utilisant les étiquettes de classe prédites. Ensuite, ces instances corrigées sont réintroduites dans l'échantillon d'apprentissage. Notre méthode de correction du bruit d'étiquetage s'appuie sur une stratégie adaptative similaire à celle de notre

méthode d'élimination du bruit d'étiquetage. Mais, au lieu de supprimer une quantité $\boldsymbol{M}$ de bruit de l'échantillon d'apprentissage, elle corrige automatiquement le bruit détecté en utilisant les étiquettes prédites par l'ensemble construit par *bagging*. Ainsi, contrairement au schéma d'élimination, le nombre total d'échantillons d'apprentissage reste le même.

Une version itérative de notre algorithme de correction du bruit est aussi proposée dans cette thèse. Cette méthode automatique de correction du bruit s'appuie sur un calcul répétitif des marges d'apprentissage qui est similaire à l'étape de mise à jour de la distribution de marges de notre méthode de suppression du bruit d'étiquetage.

Les résultats de notre évaluation empirique démontrent que notre méthode basée sur la marge est plus performante que le filtre basé sur le vote majoritaire. En outre, les marges supervisées sont généralement plus performantes que les marges non supervisées, et les marges basées sur un calcul de somme sont plus efficaces pour la gestion du bruit d'étiquetage que les marges basées sur un calcul de maximum. De plus, notre approche d'élimination du bruit est plus performante que le schéma de correction associé. Notre méthode de filtrage de bruit basée sur un calcul itératif des marges d'apprentissage s'est avérée utile pour améliorer la performance de classification de l'algorithme *AdaBoost.M1*.

Classification multiple de données déséquilibrées utilisant la marge d'ensemble

Une distribution de classes déséquilibrées dans un échantillon d'apprentissage est un défi à la conception d'un classifieur. Les méthodes d'ensemble sont plus efficaces que les techniques d'échantillonnage de données pour améliorer la performance de classification des données déséquilibrées. Les techniques de *bagging* sont non seulement robustes au bruit mais aussi simples à développer. C'est pourquoi, nous avons choisi de fonder notre nouvel algorithme de classification multiple de données déséquilibrées sur le *bagging*.

Sélectionner les instances d'apprentissage les plus pertinentes pour chaque classifieur de base d'un ensemble est important pour gérer le problème de déséquilibre des données et éviter la perte d'information. Les instances informatives, telles que les échantillons en bordure de classes ou ceux appartenant à des classes difficiles, jouent un rôle majeur en classification, en particulier dans un contexte de déséquilibre des données. Ces instances ont généralement de faibles marges d'ensemble et sont plus importantes que les instances de forte marge pour la construction d'un classifieur fiable. En conséquence, un nouvel algorithme, basé sur une fonction d'évaluation de l'importance des données, qui s'appuie encore sur la marge d'ensemble, est proposé pour traiter le problème de déséquilibre des données. Dans cet algorithme, l'accent est mis sur les échantillons de faible marge. De plus, en classification équilibrée, se focaliser sur les instances de faible marge selon un ordonnancement global des marges devrait être bénéfique pour la performance d'un classifieur ensembliste. Cependant, ce schéma n'est pas approprié pour améliorer un modèle construit à partir d'un échantillon d'apprentissage déséquilibré. Même si la plupart des instances de classes minoritaires ont des valeurs de marge faibles, la sélection d'instances pertinentes à partir d'un tri global des marges risque d'engendrer une perte

d'échantillons des classes partiellement minoritaires, voire une détérioration de la performance de classification. Par conséquent, notre algorithme sélectionne les instances pertinentes de chaque classe de manière indépendante. Notre méthode est évaluée, en utilisant encore une fois quatre différentes marges d'ensemble, vis à vis de sa capacité à traiter le problème de déséquilibre des données, en particulier dans un contexte multi-classes.

La méthode proposée consiste en trois étapes principales : 1) calculer les valeurs de marge des échantillons d'apprentissage via un classifieur ensembliste; 2) construire des sous-échantillons d'apprentissage équilibrés en se focalisant plus sur les instances de faible marge; 3) entrainer les classifieurs de base sur les sous-échantillons d'apprentissage équilibrés et construire un nouvel ensemble avec une meilleure capacité à gérer les données déséquilibrées.

Notre méthode est inspirée de l'algorithme *SMOTEBagging* (*Synthetic Minority Over-sampling Technique*), une méthode d'ensemble majeure de sur-échantillonnage de données déséquilibrées. Un taux de ré-échantillonnage $\boldsymbol{a}$ est également utilisé pour contrôler le nombre d'instances à sélectionner dans chaque classe pour engendrer un échantillon équilibré. Cependant, notre algorithme combine l'apprentissage d'ensemble avec du sous-échantillonnage, adoptant ainsi un schéma de combinaison similaire à celui de l'algorithme *UnderBagging*. Mais, contrairement à ce dernier qui ré-équilibre les classes de manière aléatoire, notre méthode s'attache à construire des sous-échantillons équilibrés de meilleure qualité pour chaque classifieur de base. Cette approche pourrait éviter les principaux inconvénients des algorithmes *SMOTEBagging* et *UnderBagging*. Elle a une complexité calculatoire plus faible que celle de *SMOTEBagging* et se focalise plus sur les instances importantes pour des tâches de classification que *UnderBagging*.

Pour évaluer l'efficacité de notre approche, le *bagging* standard, ainsi que *UnderBagging* et *SMOTEBagging* qui ont inspiré notre méthode, ont été utilisés dans une analyse comparative. Cette étude a mis en évidence la supériorité de la nouvelle méthode proposée dans la prise en charge du problème de déséquilibre des données par rapport au *bagging*, *UnderBagging* et *SMOTEBagging*. Les marges basées sur un calcul de somme sont généralement plus performantes que les marges basées sur un calcul de maximum en termes de précision moyenne. En revanche, ces dernières ont une meilleure performance de classification minimum par classe. Les marges supervisées et les marges non supervisées atteignent des performances similaires. En outre, l'efficacité de la nouvelle marge proposée dans la gestion du problème des données déséquilibrées est démontrée.

Finalement, les méthodes d'ensemble proposées sont appliquées à la cartographie d'occupation du sol, une tâche de classification majeure en télédétection.

Application à la cartographie d'occupation du sol

En télédétection, les erreurs d'étiquetage sont inévitables car les données d'apprentissage sont typiquement issues de mesures de terrain. La présence de bruit dans les images de télédétection dégrade la capacité d'interprétation des données. Le déséquilibre des données d'apprentissage est un autre problème fréquent en télédétection. Les deux méthodes

d'ensemble proposées, intégrant la définition de marge la plus pertinente face à chacun de ces deux problèmes majeurs affectant les données d'apprentissage, sont appliquées à la cartographie d'occupation du sol.

Les forêts aléatoires sont des méthodes d'ensemble puissantes qui sont particulièrement pertinentes pour la classification de données de télédétection grâce à leur robustesse au bruit et leur efficacité pour les données volumineuses et de grande dimension. C'est pourquoi, nous avons opté pour les forêts aléatoires, au lieu du *bagging*, comme ensemble robuste dans la conception de nos algorithmes, basés sur la marge d'ensemble, pour résoudre les problèmes du bruit d'étiquetage et du déséquilibre des données d'apprentissage dans le contexte difficile de la classification de données de télédétection. En outre, seuls les taux de ré-échantillonnage $\boldsymbol{a}$ pouvant induire les meilleurs résultats sont adoptés dans notre méthode de classification multiple de données déséquilibrées basée sur la marge.

Nos résultats expérimentaux montrent que notre approche de gestion du bruit d'étiquetage des données d'apprentissage basée sur la marge est efficace pour la cartographie d'occupation du sol. Cette méthode est significativement plus performante, aussi bien pour l'élimination que pour la correction du bruit d'étiquetage, que la méthode de filtrage du bruit d'étiquetage basée sur le vote majoritaire, en présence de bruit artificiel et de bruit réel. De plus, le schéma itératif de calcul des marges est généralement plus efficace pour la prise en charge du bruit d'étiquetage que la version basée sur une seule passe de calcul. En outre, notre extension des forêts aléatoires, intégrant la marge d'ensemble et des taux de ré-échantillonnage optimaux, est efficace pour la cartographie d'occupation du sol dans un contexte de déséquilibre des données. Elle s'avère plus performante que les forêts aléatoires traditionnelles et deux autre extensions, les forêts aléatoires combinées avec du sous-échantillonnage de données et celles combinées avec du sur-échantillonnage de données de type *SMOTE*, selon deux mesures d'évaluation, la précision moyenne et la précision minimum par classe.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter introduces the research directions of this thesis and points out the investigated issues that will be addressed in subsequent chapters. Section 1.1 introduces machine learning concepts. Section 1.2 provides a brief introduction to ensemble learning. Section 1.3 presents the application of ensemble methods to remote sensing data classification. Section 1.4 describes the research questions raised in this thesis. Section 1.5 summarizes the main contributions of this work. Section 1.6 outlines the content of each subsequent chapter.

1.1 Machine Learning

Over the past two decades, Machine Learning (ML) has been recognized as central to the success of Artificial Intelligence which involves the study and development of computational models capable of learning processes [144]. It has been successfully used in many application fields such as web page ranking [90], collaborative filtering [195], entity recognition [16], speech recognition [67] and remote sensing [112, 185].

Supervised learning is a major research area in machine learning [144]. In supervised learning, a set of training examples (normally described by several features) with known output values is used by a learning algorithm to generate a model. This model is intended to approximate the mapping between the inputs and outputs. Finally, this model can be used to predict the instances with unseen labels. The goal of supervised learning is to provide a model which has low prediction error on future data that has not been observed during training. Some well-known supervised classification systems are: k-nearest neighbours (k-NN) [60], decision trees [27], neural networks [94], support vector machines (SVM) [30] and ensemble learning [21]. In this work, we focus on ensemble learning.

1.2 Ensemble learning

Classification has been widely studied in machine learning. Ensemble learning, also called committee-based learning, is an effective method to develop accurate classification systems [21]. The ensemble concept originates from the famous Condorcet theorem (1785) [43], which states that:

Even if the members of a group have just 50% of chance to individually take the right decision, a majority voting of the same group has nearly 100% of chance to take the right decision.

Ensemble learning has become a major learning paradigm since the 1990s. This method is appealing because it is able to boost weak learners which are slightly better than random guess to strong aggregated learners which can make very accurate predictions [212]. Furthermore, the generalization ability of an ensemble (or multiple classifier) is usually much stronger than that of base learners [212].

An ensemble is itself, in most case, a supervised learning algorithm (unsupervised [41] or semi-supervised [81] ensembles exist but more marginally), because it can be trained and then used to make predictions. An ensemble is constructed in two steps, i.e., generating the base learners, and then combining them [51], [212]. Base learners are usually generated from training data by a base learning algorithm which can be decision tree, neural network or other kinds of machine learning algorithms. Ensemble methods have already achieved great success in many real-world tasks, such as medical diagnosis [210], [213] and remote sensing [36], [79], [89], [82].

Ensemble margin is a key concept in ensemble learning [176]. It has been applied to both the theoretical analysis and the design of machine learning algorithms. Several studies have shown that the generalization performance of an ensemble classifier is related to the distribution of its margins on the training examples. Recently, the ensemble margin has been used in imbalanced data sampling [64], noise removal [82], instance selection [82], [136], feature selection [7] and classifier design [76], [82], [137]. This major concept in ensemble learning will be at the core of our ensemble learning framework.

1.3 Application to remote sensing data classification

Remote sensing image classification, which is an important topic in the field of remote sensing, is an approach to distinguish class attributes and distribution of ground objects based on the features of material electromagnetic radiation information in the remote sensing images [139]. It can be used for information extraction, dynamic change monitoring, cartography, remote sensing database construction and so on. The purpose of classification is to estimate the different species of each geographic region in remote sensing images.

In the past few decades, experts have been working on ways to improve the classification paradigm to obtain high remote sensing image classification accuracy [46, 91]. However, the classification accuracy is directly influenced by the quality of the training data used and real-world data often suffers from many problems which could de-

grade the interpretation ability of the remote sensing data. In recent years, studies have demonstrated the successful application of ensemble machine learning classifiers, such as Random Forests [26] integrating remote sensing and ancillary spatial data, to improve supervised classification accuracy of land cover maps [166], for which conventional parametric statistical classification techniques might not be appropriate [143]. Random forests is a powerful ensemble technique which is particularly suitable for remote sensing classification [59, 80]. Thanks to its noise robustness and its efficiency for large size and high dimensionality data.

1.4 Research questions and motivations

1.4.1 Research questions

The quality and quantity of training data is very important in supervised learning. However, in most cases, samples used for training a classification model are unsatisfactory. Noisy, imbalanced, high dimensionality and complex data are major challenges in machine learning [82]. Mislabeled training data is a challenge to face in order to build a robust classifier whether it is an ensemble or not. In remote sensing, where training data are typically ground-based, mislabeled training data is inevitable. Imbalanced training data is another problem frequently encountered in remote sensing. For most classifiers, the classification performance more or less decreases with noise level and imbalance ratio.

Generally, there are two methods to deal with the above mentioned problems: data addressing and algorithm improvement [29, 95]. Data addressing methods [37, 104] improve the classification performance of a model by changing the distribution of the training set without modification of the construction of the classifier. Algorithm level schemes [9], also named internal methods, try to adapt existing classifier learning algorithms to obtain high classification accuracy. Ensemble learning can be considered as a combination of data addressing and algorithm level schemes. It has been shown to be more suitable to classify noisy and imbalanced data than single classifiers [73, 183]. Nevertheless, it still receives some negative effects from such abnormal datasets to some extent. In this work, we focus on the exploration of ensemble margin to improve the performance of ensemble models in handling class noise and class imbalance training data issues.

1.4.2 Motivations

The main motivations for our work are:

1. Ensemble margin plays a crucial role in machine learning research as it provides a strong indication of a learners performance in practice. Recently, there has been a growing line of research in utilizing the concept of margin for algorithm design. Our ensemble learning algorithms exploit the characteristics of the ensemble margin to determine the type and quality of training instances.

2. The limitations of existing research clearly suggest that we should pursue a different framework in data cleaning. Hence, this work focuses on exploiting the margin concept to improve the quality of the training set and therefore to increase the classification accuracy of noise sensitive classifiers.

3. Using an appropriate training data selection preprocessing step is essential when the prediction model is trained on an imbalanced data set. In such cases, traditional ensemble methods fail to account for imbalanced class distributions, leading to poor predictions for minority class samples. We select important instances for each classifier in an ensemble using again the margin theory to face the class imbalance issue and avoid loss of information.

1.5 Thesis Contributions

The contributions of this thesis are:

1. A novel ensemble margin definition. It is an unsupervised version of a popular ensemble margin. Indeed, it does not involve the class labels.

2. An ensemble margin-based class noise identification and elimination method based on an existing margin-based class noise ordering to handle the mislabeling problem. This method can achieve a high mislabeled instance detection rate while keeping the false detection rate as low as possible.

3. An extension of the margin based noise elimination method to tackle the class noise correction which is a more challenging issue.

4. A novel bagging algorithm based on a data importance evaluation function relying again on the ensemble margin to deal with the class imbalance problem.

The proposed ensemble algorithms, involving random forests, were applied to land cover mapping tasks after being validated on diverse image and non-image data sets using bagging as a robust ensemble.

1.6 Organization of the thesis

This thesis is divided into 6 chapters, besides the introduction and the conclusion, organized as follows:

- **Chapter 2**

  This chapter presents some theories in ensemble learning and commonly-used ensemble approaches. Three ensemble creation methods are presented first. Then ensemble diversity which is one of the major fundamental concepts in ensemble learning is introduced. The focus of this chapter is on the margin theory as well as on its application in machine learning.

- **Chapter 3**

  This chapter gives an overview of noise filtering methods. It presents class noise handling methods especially the ensemble-based class noise handling methods including class noise removal, correction and ensemble margin based methods, as well as noise robust ensemble learners. Noise addressing methods for remote sensing data are finally introduced.

- **Chapter 4**

  In this chapter, we propose a novel unsupervised margin definition and study the suitability of two popular ensemble margins as well as the proposed new margin for class noise identification. Then an ensemble margin based method, which relies on class noise ordering based on ensemble margin and involves noise removal and correction, is presented to address the mislabeling problem.

- **Chapter 5**

  This chapter gives a review on ensemble methods for the class imbalance problem. It first introduces oversampling and undersampling based methods for imbalance learning, then presents and highlights ensemble-based class balancing methods. Imbalance learning methods for remote sensing data are finally presented.

- **Chapter 6**

  This chapter proposes a novel algorithm based on ensemble margin to deal with the class imbalance problem. We first carry out a feasibility study on adopting the margin concept for imbalance ensemble learning. Then a novel algorithm, that is based on a data importance function relying on the margin and forcing on the usage of lowest margin samples, is presented in detail.

- **Chapter 7**

  This chapter mainly focuses on the application of previously presented ensemble learning methods in remote sensing. Both proposed ensemble methods involving the best margin definition, for handling class noise and class imbalance training data issues, and random forests are applied to the mapping of land covers.

Chapter 2

An introduction to ensemble learning

This chapter presents some theories in ensemble learning and commonly-used ensemble approaches. Section 2.2 presents three ensemble creation methods. Section 2.3 introduces ensemble diversity which is one of the major fundamental concepts in ensemble learning. Section 2.4 focuses on the margin theory as well as on its application in machine learning. The final section gives a conclusion of this chapter.

2.1 Introduction

Ensemble methods use multiple models to obtain better predictive performance than could be obtained from any of the constituent models. They can improve the classification accuracy and reduce the generalization error effectively. They can be built at four different levels: data level, feature level, classifier level and combination level [82]. An ensemble contains a number of learners which are usually called base learners (single classifiers) [51], [212]. The generalization ability of an ensemble (multiple classifier) is usually much stronger than that of base learners [212]. Ensemble learning is appealing because it is able to boost weak learners which are slightly better than random guess to strong aggregated learners which can make very accurate predictions. So, base learners are also referred to as weak learners. It is noteworthy, however, that although most theoretical analyses work on weak learners (typically high variance classifiers such as decision trees), base learners used in practice are not necessarily weak since using not-so-weak base learners often results in better classification performance though of higher complexity.

Diversity among the members of an ensemble is known to be an important factor in classifier combination. In other words, the key to the success of ensemble algorithms is that, intuitively at least, they build a set of diverse classifiers. Intuitively, if there are many different classifiers, it is sensible to expect an increase in the overall performance

when combining them. Then, it is generally accepted that classifiers to be combined should be diverse, since there is clearly no advantage to be gained from an ensemble that is composed of a set of identical classifiers. Diversity has been utilized to determine ensemble generalization error [145] and design classification models [35].

In the field of machine learning, the margin plays an important role. This concept was first proposed by Vapnik, who applied it to build Support Vector Machines (SVM) [44]. It can be used to measure the degree of confidence of the classification [136] and its theory can also be used to guide the design of classification algorithms. Ensemble margin, which is one kind of margin, is an interesting and important factor to the generalization performance of voting classifiers. It has been argued that the smaller ensemble margin instances have a major influence in forming an appropriate training set to build a reliable ensemble classifier [82].

2.2 Ensemble creation methods

An ensemble classifier can be built through five steps: choice of base classifier, processing of the training data, processing of the input features, fusion of base classifiers' decisions, and injecting some randomness. Let us emphasize that the processing of the training data and the processing of the input features themselves usually inject randomness into the ensemble learner. *Boosting* [175], *bagging* [21] and *random forests* [26] are major ensemble learning methods.

2.2.1 Boosting

The term boosting refers to a family of algorithms that are able to convert weak learners to strong learners. Intuitively, a weak learner is just slightly better than random guess, while a strong learner is very close to perfect performance. The birth of boosting algorithms originated from the answer to an interesting theoretical question posed by Kearms and Valiant in 1989 [212]. That is, whether two complexity classes, *weakly learnable* and *strongly learnable* problems, are equal. This question is of fundamental importance, since if the answer is positive, any weak learner is potentially able to be boosted to a strong learner, particularly if we note that in real practice it is generally very easy to obtain weak learners, but difficult to get strong learners [212]. Schapire proved that the answer is positive, and the proof is a construction,i.e., boosting [175].

There are many variants of this powerful ensemble approach. It works by training a set of learners sequentially and combining them for prediction, where the later learners focus more on the mistakes of the earlier learners [212]. Adaboost [70] is the most influential boosting algorithm, it is summarized in Algorithm 1.

Algorithm 1: Adaboost

Input: Data set $S = (x_1, y_1), (x_2, y_2,), \cdots, (x_n, y_n)$;
Base learning algorithm ζ;

Number of learning rounds T;
Initialization:
$D_1(x) = 1/m$. %Initialize the weight distribution
Iterative process:
for $t = 1$ to T **do**
1. $h_t = \zeta(S, D_t)$; %Construct base learner h_t from S using distribution D_t
2. $\varepsilon_t = P_{x \sim D_t}(h_t(x) \neq y)$; %Evaluate the error of h_t
3. if $\varepsilon_t > 0.5$ **then break**
4. $\alpha_t = \frac{1}{2}\ln(\frac{1-\varepsilon_t}{\varepsilon_t})$; %Determine the weight of h_t
5. Update:

$$D_{t+1}(x) = D_t(x) \cdot \frac{e^{-\alpha_t h_t(x) y}}{Z_t} \tag{2.1}$$

Z_t is a normalization factor which enables D_{t+1} to be a distribution
end
Output: $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$

After obtaining the base learners, boosting combines them by majority voting, a typical combination strategy, and the most-voted class is predicted.

2.2.2 Bagging

The name Bagging came from the abbreviation of *Bootstrap AGGregatING* [21]. As the name implies, the two key ingredients of bagging are bootstrap and aggregation [212]. In bagging, the individual classifiers can be built in parallel, independently of one another. Bagging trains a number of base learners each from a different bootstrap sample by calling a base learning algorithm. Algorithm 2 summarizes the bagging procedure. A bootstrap sample is obtained by uniformly subsampling the training data with replacement. To predict a test instance, bagging feeds the instance to its base classifiers and collects all of their outputs, and then uses the most popular strategies *voting* to aggregate the outputs and takes the winner label as the prediction [212].

It is worth mentioning that the bootstrap sampling also offers bagging another advantage. As Breiman indicated, a bootstrap sample is obtained by uniformly subsampling the training data with replacement [21]. For a given bootstrapped sample, an instance in the training set has typically a probability of approximately 63.2 % of being selected at least once. The remaining 36.8% examples called out-of bag (OOB) will not be picked up [22], [209]. The performance of the base learners can be estimated by using these out-of-bag examples, and thereafter the generalization error of the bagged ensemble can be estimated [212].

Algorithm 2: Bagging

Input: Data set $S = (x_1, y_1), (x_2, y_2,), \cdots, (x_n, y_n)$;

Base learning algorithm ζ;
Number of learning rounds T;
Iterative process:
for $t = 1$ to T **do**
$h_t = \zeta(S, D_{bs})$; $\%D_{bs}$ is the booststrap distribution
end
Output: $H(x) = sign(\sum_{t=1}^{T} h_t(x))$

2.2.3 Random forests

A variant of bagging, *random forests* [26], has been deemed as one of the most powerful ensemble methods up to date [212]. *Random forests* is a combination of tree predictors in which decision trees are constructed using resampling with replacement, they randomly sample the attributes and choose the best split among those variables rather than the best split among all attributes. The suggested value of the number of randomly selected features is the logarithm or the square root of the total number of features [26]. Hence, randomness is not only injected into data sampling, as in bagging, but also introduced into the feature selection process. The assignment of class label of an unknown instance is generally performed using majority voting. Important advantages such as running efficiently on large data bases, handling thousands of input variables without variable deletion and low time cost make *random forests* widely attract the interest of researchers.

2.2.4 Comparative analysis

There are many ensemble classification methods with good performance. However, existing ensemble techniques have different drawbacks [209].

- Quinlan applied boosting and bagging to C4.5 [159] decision tree-based ensembles. Experimental results show that they can reduce the generalization error, and boosting has better effect than bagging. But in some cases, boosting can create overfitting [14].

- In boosting, each base classifier is trained on data that is weighted based on the performance of the previous classifier. The next base classifier focuses on the current samples which are classified with difficulty.

- Boosting can not only reduce the bias but also reduce the variance [55], but bagging can only reduce the variance. Bagging does nothing purposely to reduce the bias so that any bias reduction is achieved solely by chance [21].

- Boosting is sensitive to noise and outliers [14]. The noise sensitivity of AdaBoost is generally attributed to the exponential loss function which specifies that if an instance were not classified as the same as its given label, the weight of the instance

will increase drastically. Consequently, when a training instance is associated with a wrong label, AdaBoost still tries to make the prediction resemble the given label, and thus degenerates the performance [212].

- Random forest combines Breiman's "bagging" idea and the random selection of features. Randomness is introduced into the feature selection process [212]. Hence, random forest generates more diversity than bagging.

2.3 Ensemble diversity

Ensemble diversity is a property of an ensemble with respect to a set of data. It has been recognized as an important characteristic in classifier combination [122]. Diversity is the difference among the individual learners [122]. It is greater when, all other factors being equal, the classifiers that make incorrect decisions for a given example spread their decisions more evenly over the possible incorrect decisions. The more uniformly distributed the errors are, the greater the diversity, and vice versa [212]. Hence, ensemble methods can effectively make use of such diversity to reduce the variance-error without increasing the bias-error. In other words, ensemble learning is very effective, mainly due to the phenomenon that base classifiers have different "biases" [212]. Accuracy is one of the standard evaluations of the ensemble classification [205].

Though, measuring diversity is not straightforward because there is no generally accepted formal definition [122], there are effective popular heuristic mechanisms for diversity generation in ensemble construction [212], including processing the training samples [21], manipulating the representation of the target attributes [26, 48], selecting parameters [129], and output representations [25]. Their main idea is to inject randomness into the learning process:

1. Data sampling is the most popular method to increase the ensemble diversity. Multiple different training sets are obtained by sampling approaches, then the individual learners are trained from the generated different data sets [21].

2. In input feature manipulation, several classifiers with different and usually simpler representations of the target attributes are induced. Different subsets of features provide different views on the training data [169]. Therefore, individual learners trained from different subsets of features are usually diverse.

3. Parameters learning tries to generate diverse individual learners by using different parameter settings for the base learning algorithm [129].

4. In output methods, diverse individual learners are generated by combining different output representations [25].

2.4 Ensemble margin

2.4.1 Margin concept

Margins, which were originally applied to explain the success of boosting [176] and to develop the Support Vector Machines (SVM) theory [196], play a crucial role in modern machine learning research. The ensemble margin [176] is a fundamental concept in ensemble learning. Several studies have shown that the generalization performance of an ensemble classifier is related to the distribution of its margins on the training examples [176]. A good margin distribution means that most examples have large margins [103].

Ensemble margin has been used for improving the performance of Boosting [176]. Adaboost is the most successful method in improved methods of boosting [212]. It is known not to overfit since it tends to enlarge the ensemble margin even after the training set error reaches zero [76]. Schapire et al. [176] attempted to explain this phenomenon in terms of the ensemble margins the classifier achieves on training examples. Arc-Gv [24] is a variant of Adaboost. It was designed by Breiman to doubt Schapire's explanation. It is very similar to AdaBoost, but different in calculating the coefficient associated with each weak classifier. Arc-Gv always had a minimum ensemble margin that was provably larger than AdaBoost but performed worse in terms of test error. Thus, Breiman concluded that the ensemble margin theory could not explain AdaBoost completely.

Reyzin and Schapire reproduced Breiman's main finding and found that a better ensemble margin distribution was more important than the maximisation of the minimum ensemble margin [163]. It was of importance to have a large minimum ensemble margin and necessarily at the expense of other factors.

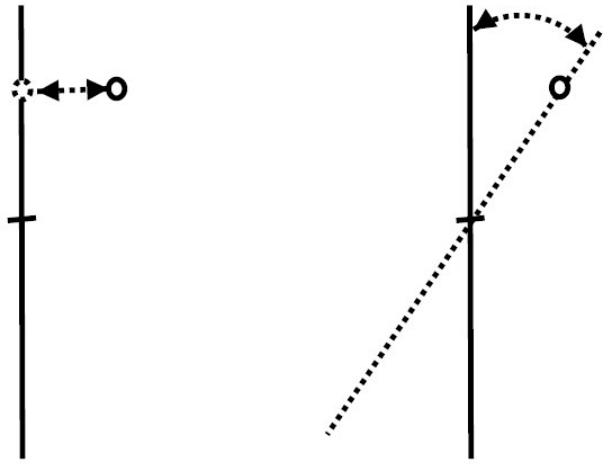


Figure 2.1: Sample margin (left) and hypothesis margin (right).

The margin region is the set of the contradiction samples of the information system.

There are two main ways to define margins: sample margin and hypothesis margin [47] (see Figure 2.1). The sample margin is defined as the distance between the instance and the decision boundary induced by the classifier. For example, support vector machines [44] aim to find the separating hyper-plane with the sample margin. However, the hypothesis margin requires the existence of a distance measure on the hypothesis class, it measures how much can the hypothesis travel before it hits an instance without changing the way it labels any of the sample points. This definition requires a distance measure between classifiers [7], [47]. This type of margin is the ensemble margin used in AdaBoost [70].

Margin-based classification is a growing line of research. In machine learning, the ensemble margin has been used in imbalanced data sampling [64], noise removal [82], [197], instance selection [82], [125], [136], feature selection [7] and classifier design [76], [82], [126], [137], [163], [180], [206].

2.4.2 Ensemble margin definitions

Different definitions of ensemble margin have been proposed [47], [76], [82], [126], [176].

2.4.2.1 Ensemble votes based definitions

The decision by an ensemble for each instance is made by voting. The ensemble margin can be calculated as a difference between the votes [82] according to two different well-known definitions [113] in both supervised [176] and unsupervised [83, 86] ways.

1. A popular ensemble margin, which has been introduced by Shapire et al. [176], is defined by equation (2.2), where v_y is the number of votes for the true class y and v_c is the number of votes for any other class c. This ensemble margin is in the range [-1, +1] and the examples which are correctly classified have positive margin values. A large positive ensemble margin can be interpreted as a confident correct classification.

$$margin(x) = \frac{v_y - \max_{c=1,\ldots,L \cap c \neq y}(v_c)}{\sum_{c=1}^{L}(v_c)} \tag{2.2}$$

   where L represents the number of classes.

2. The ensemble margin of a sample can also be obtained by the difference between the fraction of classifiers voting correctly and incorrectly, as in equation (2.3) [82, 113]. This second popular ensemble margin definition follows the same idea introduced by Schapire [176] but instead of using a max operation, it uses a sum operation [113].

$$margin(x) = \frac{v_y - \sum_{c=1,\ldots,L \cap c \neq y}(v_c)}{\sum_{c=1}^{L}(v_c)} \tag{2.3}$$

This ensemble margin is also in the range [-1, +1]. However, correctly classified samples have not necessarily positive margin values.

3. In [83, 86], the authors proposed an unsupervised version of Schapire's margin (equation (2.2)). This ensemble margin's range is from 0 to 1. It is defined by equation (2.4), where v_{c_1} is the votes number of the most voted class c_1 for sample x, and v_{c_2} is the votes number of the second most popular class c_2.

$$margin(x) = \frac{v_{c_1} - v_{c_2}}{\sum_{c=1}^{L}(v_c)} \tag{2.4}$$

Naturally, for two-class problems these definitions are quite similar. However, a major concern needs to be solved in relation to multi-class problems. For example, by equation (2.3), the margins can represent a lower bound, since they can assume negative values even when the correct label gets the most of votes (when there is a plurality, but not a majority) [113].

Gao and Zhou show that although AdaBoost effectively maximizes the minimum ensemble margin, compared with previous statistics on ensemble margin theory, the average ensemble margin is one of the statistics that considers the whole ensemble margin distribution and thus includes more information [76]. They proved that in the context of binary classification (2.5):

$$\begin{aligned} \overline{margin}(x) &< \frac{1}{m}\sum_{i=1}^{m} margin(x_i) \\ &= \frac{\sum_{t=1}^{T}\sum_{i=1}^{m} y_i \alpha_t h_t(x_i)}{m\sum_{t=1}^{T}|\alpha_t|} \end{aligned} \tag{2.5}$$

where $margin(x_i)$ is the margin of instance x_i and y_i its label, h_t is the base learner, α_t is the corresponding weight, T is the number of learning rounds (base classifiers in AdaBoost), m is the number of instances.

2.4.2.2 Other definitions

1. Crammer et al. [47] compute the ensemble margin of an instance x with respect to a set of instances A by equation (2.6) where $nearest\ miss_A(x)$ and $nearest\ hit_A(x)$ denote the nearest instance to x in A with the same and different label, respectively.

$$margin_A(x) = \frac{1}{2}\Big(||x - nearest\ miss_A(x)|| - ||x - nearest\ hit_A(x)||\Big) \tag{2.6}$$

2. According to the well-established relationship *the higher the classification confidence provided by the classifier, the higher the probability that the classifier has*

correctly classified this sample, Li et al. gave a new definition of ensemble margin [126]. The ensemble margin of sample x_i based on classification confidence is denoted by equation (2.7):

$$margin(x_i) = S(y_i) - max\{S(y_j)|i \neq j\} \tag{2.7}$$

where $S(w_i)$ means the sum of classification confidences whose corresponding classification decision is w_i which is the true label of x_i.

The authors proposed then the definition of the ensemble margin loss of x_i based on two different loss functions [126], the functions are respectively denoted as (2.8) and (2.9):

$$l_1(x_i) = \big(1 - margin(x_i)\big)^2 \tag{2.8}$$

$$l_2(x_i) = log\Big(1 + \exp\big(-margin(x_i)\big)\Big) \tag{2.9}$$

2.5 Conclusion

In this chapter, we first introduced ensemble learning, and presented three popular ensemble methods. Next, we described ensemble diversity. Accuracy and diversity are standard evaluations of the ensemble classification [205]. According to the literature, if we want to get a classifier with higher accuracy and good diversity, mainly two methods could be considered: choosing a good training set and/or a good ensemble classifier. Finally, we provided an in depth description of the concept of ensemble margin. The ensemble margin got more and more attention, and has been used in many fields of machine learning. Many definitions of ensemble margin have been given. Different definitions have different effects on the classification results. Hence, conducting a comprehensive analysis and comparison of these definitions can be useful to ensemble learning.

Chapter 3

Addressing the mislabeling problem in ensemble learning: a review

This chapter gives an overview of noise filtering methods. Section 3.1 presents the types of noise and the consequences of class noise on learning. Section 3.2 introduces class noise handling methods at both data level and algorithm level. Section 3.3 presents ensemble-based class noise handling methods including class noise removal, correction and ensemble margin based methods, as well as noise robust ensemble learners. Noise addressing methods for remote sensing data are presented in section 3.4. Section 3.5 gives a summary of existing data cleaning techniques.

3.1 Introduction

Classification has been widely studied in machine learning. The standard approach consists in obtaining a model inferred from training data to predict the class of new samples. There exist important applications of classification in fields such as pattern recognition, bioinformatics, physics, medecine, economics, etc. and are used for meteorological forecasts, text classification, disease diagnosis, remote sensing, to name a few [2]. The classification accuracy of a classifier is directly influenced by the quality of the training data used [171]. However, real-world data is never perfect and often suffers from noise [214]. The presence of noise in data is a common problem that produces several negative consequences in classification problems [78]. However, effective noise handling is one of the most difficult problems in inductive machine learning [75].

Mislabeled training data (class noise) is a challenge to face in order to build a robust classifier whether it is an ensemble or not. It is particularly troublesome in supervised problems, where it alters the relationship between the informative features and the measure outputs. Furthermore, learning from noisy data can create overfitting [197]. Labeling training instances is a costly and rather subjective task that usually induces some label-

ing errors in the training set [29, 75]. Therefore, how to reduce noise consequences and form an efficient training set is a major issue in supervised classification [142].

Several approaches in literature have been devoted to the handling of class noise and the development of robust techniques achieving higher classification accuracies on test data [68]. They can be categorized into three main approaches [82]: making algorithms that are more robust to noise [111, 158], filtering out the noise [29, 84] and correcting noisy instances [192].

3.1.1 Types of noise

The class labels and the attribute values of training data directly influence the quality of a classification although a large number of components can determine the quality of a data set [214]. In literature, two types of noise are distinguished [75, 199, 214]:

1 **Class noise** (also referred to as label noise). It occurs when an example is incorrectly labeled. Class noise can be attributed to several causes, such as subjectivity during the labeling process, data entry errors, or inadequacy of the information used to label each example. Two types of class noise exist [34, 78] :

   - *Contradictory examples* which refer to duplicate or similar examples in the data set having different class labels [97].
   - *Misclassification examples* which are labeled with class labels different from their true labels [215].

   In addition, some authors also consider outliers that are correctly labeled as class noise [153]. Mislabeled instances may be outliers if their labels have a low probability of occurrence in their vicinity. Similarly, some instances may also look abnormal, with respect to the class that corresponds to their incorrect label. Hence, it is natural that many techniques in the class noise literature are very close to outlier and anomaly detection techniques. Many of the methods that have been developed to deal with outliers and anomalies can also be used for class noise. However, it must be highlighted that mislabeled instances are not necessarily outliers, or anomalies, which are subjective concepts [68]. For example, if labeling errors occur in a boundary region where all classes are equiprobable, the mislabeled instances neither are rare events nor look anomalous. Similarly, an outlier is not necessarily a mislabeled sample since it can be due to attribute noise or simply be a low-probability event [68].

2 **Attribute noise** It refers to corruptions in the values of one or more attributes. Examples of attribute noise are: erroneous attribute values, missing or unknown attribute values, and incomplete values [78].

It has been proved that class noise is potentially more harmful than attribute noise on classifier capability [158]. Quinlan shows that removing higher levels of noise from

attribute information decreases the predictive accuracy of the resulting classifier if the same attribute noise is present when the classifier is subsequently used, but for class noise, cleaning the training data will result in a classifier with a higher predictive accuracy [158]. In addition, the prevalence of the impact of class noise is explained by the fact that [68]:

1. There are many features, whereas there is only one class label.
2. The importance of each feature for learning is different, whereas labels always have a large impact on learning.

Hence, this work put the emphasis on class noise and the different ways of addressing it.

3.1.2 Consequences of class noise on learning

Class noise is ubiquitous in real-world datasets and has several negative consequences [68]:

- Class noise can decrease classification performances, which has been theoretically proved for simple models like kNN, linear or quadratic classifiers [68].
- The presence of class noise can lead to the failure of supervised classifiers, including ensemble classifiers [6]. Learning through multiple models becomes harder for large levels of class noise, where some samples become more difficult for all models and are therefore seldom correctly classified by an individual model [68].
- Class noise can increase the number of necessary training instances, the complexity of learned models, the number of nodes of decision trees, the number of support vectors in SVMs (Support Vector Machines) [68] and the size (number of base classifiers) of an ensemble.
- Class noise can increase the difficulty to identify relevant features [214].
- Class noise can affect the estimated error rate in multiclass problems [102].
- Learning from noisy data can create overfitting [197].

3.2 Class noise handling methods

3.2.1 Dealing with class noise at data level

3.2.1.1 Class noise identification and removal

Problems corrupted by noise are complex and accurate solutions are often difficult to achieve without using specialized techniques, particularly for noise-sensitive methods [78]. Noise filters, which are preprocessing mechanisms to detect and eliminate noisy instances in the training set, are commonly applied to improve the classification performance [173].

The preprocessing of noisy data is illustrated in Figure 3.1 [29]. The effectiveness of a noise filter, i.e. whether its usage induces an improvement in classifier performance, depends on the noise-robustness, the generalization capabilities of the classifier used and the characteristics of the data especially the level of noise that affects them [173].

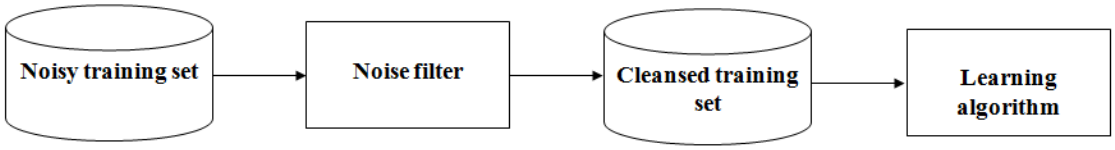


Figure 3.1: General procedure for learning in the presence of class noise with training data cleansing [29].

A simple filtering method to deal with class noise is to remove instances that appear to be mislabeled. Many such cleansing methods exist in the class noise literature. For example, *classification filtering* uses the predictions of classifiers to identify mislabeled instances [68]. Thongkam et al. learn a SVM using the available training data then all training instances which are misclassified are simply removed [194]. The filtering approach is easy to implement and induces relatively low computational costs. Moreover, it can simplify resulting models [178]. One major drawback of filtering the data is that some valuable instances might be dropped from the data set [192] which can be particularly harmful on small and imbalanced data sets. However, according to [29], keeping bad mislabeled instances may hinder performance more than removing too many correctly labeled samples. In addition, the noise filtering approach cannot fully overcome the errors in the data for noise levels of 30% or greater [10].

Hughes et al. extend the above method, but instead of removing the misclassified instances, they delete the label of these instances, for which experts are less reliable, then use semi-supervised learning with both the labelled and the new unlabelled instances [105]. The advantage of this method is keeping the distribution of the instances unaltered [68]. However, this method also suffers from a *chicken-and-egg* dilemma. Good classifiers are necessary for *classification filtering*, but learning in noisy environment may precisely produce poor classifiers.

The Blame-Based Noise Reduction (BBNR) algorithm removes all instances that contribute to the misclassification of their nearest neighbours and whose removal does not cause any instance to be misclassified [49]. This method does not need more accurate classifiers for filtering the noise but can increase the computational complexity.

3.2.1.2 Class noise identification and correction

All the above algorithms are mainly used for improving the data quality by detecting and eliminating its class noise. Teng shows that a classifier built from corrected data

should have a higher predictive power than filtered data [192]. He describes a different approach called *polishing*. When noisy instances are identified, instead of removing them, they are repaired by replacing the corrupted class label values with predicted class labels then the corrected instances are reintroduced into the data set. In [10], the *depuration* algorithm was proposed to iteratively modify the class label of the examples whose class label disagrees with the class labels of most of their neighbours.

Clustering can be used to detect mislabeled instances, exploiting neighbourhood consistency, as done in the *depuration* algorithm [10]. An instance whose label is not consistent with the labels of nearby clusters is likely to be mislabeled [68]. In [155], this neighbourhood consistency constraint is exploited to design a clustering-based algorithm to correct the labels of mislabeled instances.

However, noise correction is only viable when data sets are small because it is generally time consuming [78]. Although several works claim that complete or partial noise correction in training data, with test data still containing noise, improves test performance results in comparison with no preprocessing [78, 192], it can introduce more noise (correction failure) into the training data if too many truly clean examples are mislabeled [161].

3.2.2 Dealing with class noise at classifier level

3.2.2.1 Robustness of learning algorithms

Robustness is the capability of an algorithm to build models that are insensitive to data corruptions and suffer less from the impact of noise. Thus, a classification algorithm is said to be more robust than another if the former builds classifiers which are less influenced by noise than the latter, i.e. the more robust an algorithm is, the more similar the models built from clean and noisy data are [78]. Robustness is considered more important than performance results when dealing with noisy data, because it allows to know a priori the expected behavior of a learning method against noise in cases where the characteristics of noise are unknown [78].

3.2.2.2 Robust algorithms against noise

Decision trees are greatly impacted by class noise. This instability makes them very suitable for ensemble methods though. A method for building robust decision trees consists to carefully select an appropriate splitting criterion. In [1], different node split criteria are compared for ensembles of decision trees in the presence of class noise. The Imprecise Information-Gain based on imprecise probabilities and uncertainty measures, is shown to improve accuracy, with respect to the Gini index, the Information Gain Ratio and the Information Gain. Another approach typically described as useful to deal with noise in decision trees is postpruning [1]. Tree pruning is a technique in machine learning that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances. This procedure reduces the complexity of the final classifier and hence improves predictive accuracy by the reduction of overfitting

caused by the overspecialization over the isolated (and usually noisy) examples [78]. However, Gamberger et al. show that this approach is less effective than noise removal and correction [75]. If the noise level is relatively high, even a robust learner may have a poor performance [78].

The kNN (K Nearest Neighbours) classifiers [60] are sensitive to class noise in particular for small neighbourhood sizes [150]. It is unlikely to obtain a good performance without data preprocessing in noisy environments. Sáez et al. proposed a computation of data complexity measures to predict in advance when the usage of a noise filter will statistically improve the prediction results of 1-NN [173].

3.3 Ensemble-based class noise handling methods

3.3.1 Ensemble methods for class noise filtering

3.3.1.1 Ensemble-based class noise removal

Classification filtering faces the risk to remove too many instances containing valuable information. To alleviate this problem, the ensemble approach is widely used to filter out mislabeled instances [29, 82, 84, 117, 182, 197, 215]. It attempts to improve the quality of the training data as a preprocessing step in classification, by detecting and eliminating mislabeled instances. The detection of mislabeled instances is performed by considering the vote of each base classifier in the ensemble to each instance [82].

Two typical ensemble approaches that address the mislabeling problem are the majority vote filter and the consensus filter [29]. In the majority vote method, if more than half of all the base classifiers of the ensemble classify an instance incorrectly, then this instance is tagged as mislabeled. The consensus filter [29] requires that all base classifiers fail to classify an instance as the class given by its training label for it to be eliminated from the training data [82]. However, a majority vote filter not only eliminates mislabeled instances but also all the clean training instances that have been wrongly classified by the underlying ensemble classifier. It cannot distinguish these *false positives* from the mislabeled instances (*true positives*). This is an important limitation as the clean training instances wrongly identified as noise contain critical information such as boundary instances which play an important role in classifier design [82]. On the contrary, the criterion of the consensus filter [29] is too strict. It removes just a little portion of noise. Nonetheless, neither the majority filter nor the consensus filter are satisfying answers to effective mislabeled instance filtering [82].

Verbaeten and Assche considered the problem of mislabeled training examples by preprocessing the training set based on some well-known ensemble classification methods (bagging and boosting) [197] using C4.5 as base classifier [159]. They proposed two approaches:

1. Filtering based on voting (consensus vote and majority vote) of base classifiers of a bagging ensemble.

2. Filtering based on removing training examples that obtained high weights in the boosting process. Indeed, mislabeled examples are assumed to have high weights.

Results show that majority vote filters are more accurate than consensus filters as discussed in the previous paragraph. In addition, bagging - majority vote filters outperform the boosting filters. Boosting filter tends to incorrectly remove many important correctly labeled instances with large weights.

Zhu et al. proposed a method for identifying and eliminating mislabeled instances in large or distributed datasets by partitioning a dataset into subsets [215]. The main idea is to partition a large dataset $\boldsymbol{E}$ into some subsets firstly, and learn a set of classification rules $\boldsymbol{R_i}$ for any subset of $\boldsymbol{E}$, then a special rule set $\boldsymbol{GR_i}$ is selected from $\boldsymbol{R_i}$ and used to evaluate all instances in $\boldsymbol{E}$. For any instance in $\boldsymbol{E}$, two error count variables, local error count and global error count, are used to record how this instance behaves with the good rule sets from all subsets. Due to the fact that exceptions usually do not fire $\boldsymbol{GR_i}$ and noise more likely denies $\boldsymbol{GR_i}$, the noise has a higher probability of receiving large error values in comparison with non-noisy examples. They adopted two schemes, majority and non-objection, to identify noise. After each round, the identified noise and a certain portion of good examples are removed, and if the filtering result is not satisfactory, the procedure can be repeated. This method was shown to be effectively useful for large datasets.

Miranda et al. combine four classifiers, which are induced by different machine learning techniques, and hence constitute a heterogeneous ensemble, by voting to detect mislabeled instances [147]. The identified noise was removed. Their results show that the noise removal technique was significant to increase the accuracy. However, the disadvantage of this method is that it eliminates instances that lie on the wrong side of the classification boundary, which can be dangerous [68, 88]. Moreover, lots of parameters must be considered when choosing different techniques as base classifiers for a given data set.

Edge analysis can be used to detect mislabeled instances [204]. The definition of the edge of an instance is the sum of the weights of weak classifiers, composing a boosting ensemble, that misclassified the instance [23]. Hence, it is the contrary of the ensemble margin proposed by Schapire et al. [176]. An instance with a large edge is often misclassified by the weak classifiers, i.e. it has a low confidence. In this method, the observations which were initially classified correctly are classified incorrectly in later rounds to classify harder observations correctly. Mislabeled data have edge values which remain high due to persistent misclassification. It is therefore proposed to remove the instances corresponding to the top edge values (typically 5%) [68].

In Outlier Removal Boosting (ORBoost) [114], data cleansing is performed while learning and not after learning. Instance weights which are above a certain threshold are set to zero during boosting. This method gets more robustness than adaboost because of paying less attention to class noise. However, ORBoost only has good performance in the case of low noise level. Furthermore, it is sensitive to the choice of the threshold, which is performed using a validation set [68].

Among all the mentioned ensemble-based class noise filters, it remains difficult to appropriately select one of them for effective noise handling. Recently, Sluban et al. attempted an answer to this major issue in ensemble learning. They investigated the relationship between the diversity of heterogeneous ensembles and their class noise detection performance, with the hypothesis that ensemble diversity may be used as guidance for selection of well performing noise detection ensembles [183]. The majority and the consensus ensemble voting schemes were studied in empirical analysis. Their results show that increased diversity in ensembles using the majority voting method does not lead to better performance of heterogeneous ensembles in noise detection and may even degrade the noise detection performance. On the other hand, for consensus voting-based noise detection heterogeneous ensembles, more diverse ensembles achieve higher precision in class noise detection.

3.3.1.2 Ensemble-based class noise correction

As discussed in section 3.2.1, noise removal can discard some useful data and noise correction has been shown to give better results than simply removing the noise from the data set in some cases [192]. Rebbapragada et al. use active learning to deal with the problem of class noise [161]. They proposed two scores *ALC (Active Label Correction)-Mislabeled* and *ALC-Disagreement* to identify mislabeled data. The *ALC-Mislabeled* score estimates how likely an example $\boldsymbol{x}$ is mislabeled by calculating the difference in probabilities of the existing and predicted labels. The larger the score, the more likely it is that $\boldsymbol{x}$ is mislabeled. They sort the instances in descending order according to their scores and choose the largest $\boldsymbol{k}$ scores for expert review. *ALC-Disagreement* chooses examples for relabeling that are not obviously mislabeled; it selects examples that exhibit a large degree of confusion as to the predicted label and thus can be viewed as selecting hard to classify examples. This confusion is reflected by the probability distribution over the class labels - the closer it is to a uniform distribution the more confusion [161]. Finally, mislabeled examples are updated by their predicted class labels, which are the labels receiving highest vote. Two fully-automated cleaning techniques (single-pass discarding and correcting) are used for comparison. Single-pass discarding removes examples whose probabilities or committee votes on the existing label are less than the probabilities or votes on the predicted label. Single-pass correcting simply updates the misclassified examples to their predicted labels. The results show that active learning outperforms both automated data cleaning methods. However, like any active leaning strategy, human expertise is needed, which is a major shortcoming compared to automated class noise handling.

Miranda et al. correct mislabeled data by extending their noise detection method described in section 3.3.1.1 [147]. Instances identified as noise are relabelled by the classes which are the most predicted by the noise-detection classifiers. As a comparison, the authors also proposed a hybrid technique: if the data is identified as class noise, kNN is used to decide whether it should be removed or corrected. The results demonstrate that the classifiers constructed using both class noise handling methods can achieve a higher accuracy than those using the original training set. Furthermore, the classification

accuracies achieved by noise correction and hybrid methods are similar for most data sets although the authors hope the latter can obtain a better performance. However, both methods are less effective than their noise removal technique described in section 3.3.1.1. Moreover, as discussed in section 3.3.1.1, their noise detection algorithm tends to identify a lot of important correctly classified samples as noise. In other words, noise detection plays a key role in the process of noise removal and correction. Imprecise noise identification method will lead to less effective noise removal and correction no matter how reasonable the noise identification strategy is.

3.3.1.3 Exploiting the ensemble margin for class noise filtering

Ensemble margins, which have been described detailedly in chapter 2, can be used for noise filter design. In [82], noisy instances are defined as instances that are either mislabeled in the training data, or are inherently ambiguous and hard to categorize because their label value conflicts with most of the other instance label values while having similar attribute values. Using an unsupervised version of ensemble margin, Guo defined a class noise as an instance that most base classifiers in the ensemble classified as another class. In other words, this instance was classified wrongly with high margin. Guo's ensemble margin-based noise removal algorithm removes a portion of the highest margin examples that have been misclassified. In this method, all the training instances, that have been misclassified, are sorted in descending order according to their unsupervised margin values. Then, two noise removal strategies, adaptive filtering (adaptive estimation of the noise level) and fixed filtering (removal of a fixed amount of noise that is assumed to be known) are experimented to estimate or confirm the amount of noise. The results show that ensemble margin is relevant to identify class noise. Moreover, although the classification accuracies obtained by adaptive filtering and by fixed filtering are similar for most data sets, the adaptive strategy has a great advantage over the fixed alternative in case of uncertain amount of noise which is generally the case in real-world applications.

A reverse boosting algorithm is proposed in [32]. In this method, safe, noisy and borderline patterns are distinguished, whose weights are respectively increased, decreased and unaltered during boosting. Samples are classified into these three categories using parallel perceptrons, a specific type of committee machine whose ensemble margin allows to separate the input space into three regions: a safe (beyond the margin), a noisy (before the margin), and a borderline region (inside the margin). The approach improves the results of parallel perceptrons in the presence of noisy labels, but is most often dominated by classical perceptrons [68].

3.3.2 Class noise tolerant ensemble learners

AdaBoost [69] is one of the most popular techniques for generating ensembles due to its adaptability and simplicity [33]. However, AdaBoost tends to overfit class noise because it obtains large weights for mislabeled instances in late stages of learning [68]. Several methods propose to update weights more carefully to reduce the sensitivity of

boosting to class noise [68, 198]. For example, AveBoost2 [151] algorithms replace the weight ${w_i}^{(t+1)}$ of ith instance at step $t+1$ by the following expression (equation 3.1):

$$\frac{t{w_i}^{(t)} + {w_i}^{(t+1)}}{t+1} \tag{3.1}$$

AveBoost2 obtains larger training errors, but smaller generalization errors than AdaBoost. Besides, it is more robust to class noise than AdaBoost through slowing down the growth of the weights of misclassified instances. Similarly, MadaBoost [54] imposes an upper bound for each instance weight, which is equal to the initial probability. Hence, the weights of the examples cannot become arbitrarily large as it happens in AdaBoost. This method has been shown not to overfit on noisy data. Averaged Boosting (A-Boost) [118] uses the average of the product of the base hypotheses and weights while AdaBoost uses the sum of it, and calculates the weight based on the error rate of the current hypothesis on the original training examples while the AdaBoost algorithm uses the updated weights. This approach performs similarly to bagging on noisy data. However, common losses (modification of weights) in machine learning are not always effective, especially in the case of high noise level [12].

Cao et al. proposed a new boosting approach named Noise-Detection based Adaboost (ND-Adaboost) [33]. They gave an analysis of class noise detection based loss function and ensemble margin respectively, then proposed a new loss function. The proposed algorithm was designed by integrating the class noise-detection based loss function into Adaboost to adjust the weight distribution at each iteration and added a regeneration condition to control the ensemble training error bound [33].

In [121], two approaches are proposed to reduce the consequences of class noise in boosting. One way to prevent Adaboost from overfitting is to limit the number of iterations. However, the authors did not investigate any effective ways of choosing the appropriate number of iterations. A second approach is to *smooth* the boosted classifier by bagging. This algorithm combines bagging and boosting paradigms in the following way: 1) K bootstrapped training subsets consisting of p percents of the training set are created; 2) K boosted classifiers are trained for M iterations; 3) the K predictions are aggregated. This method can increase the diversity of boosting and be less sensitive to class noise than Adaboost [121].

Bagging has better performance than boosting in the presence of class noise. Since each mislabeled sample impacts the classifier, bagging gets some quite different models by repeatedly selecting different subsets of training set using bootstrap sampling. Hence, the diversity of base classifiers is improved in bagging [68]. Abellan and Masegosa show that the employment of bagging ensembles of *credal decision trees*, a special type of decision trees, which are based on imprecise probabilities and information based uncertainty measures, can be a successful tool in classification problems with a high level of noise in the class variable [2]. A comparative empirical analysis about the accuracy of both bagging-C4.5 and bagging-credal decision trees show that bagging-C4.5 wins in most data sets with 0~20% of noise and fails when 30% of noise is added.

Several studies show that the choice of sampling size is arbitrary in terms of generalization performance of the ensemble. The optimal size of the bootstrap samples is

application dependent in noisy tasks, so it is worth to explore the possibility of subsampling [170]. Sabzevari et al. show that bagging composed of unpruned decision trees trained on bootstrap samples whose size is between 10% and 40% of the size of the original training set are more robust to class noise than standard bagging (i.e. using a 100% sampling ratio of the size of the original data) [170].

The more classes in a problem, the more complex it is. Multi-class learning can also increase the chances of incorrect classifications and hinder the prediction capabilities of the classifiers in noisy problems [78]. Several works have demonstrated that decomposing the multi-class problem into several binary subproblems is an easy way to reduce their complexity and the effects caused by noise [171]. This method includes two steps:

- *Problem division*: the problem is decomposed into several binary subproblems which are solved by independent binary classifiers,
- *Combination of the outputs*

The *One-Vs-One* (OVO) [171] decomposition strategy consists of dividing a classification problem with M classes into $M(M-1)/2$ binary subproblems. A classifier is trained for each new subproblem only considering the examples from the training data corresponding to the pair of classes (λ_i, λ_j), with $i < j$, considered. In [171], the C4.5 and RIPPER (Repeated Incremental Pruning to Produce Error Reduction) [42] robust learners and the noise-sensitive k-NN method are evaluated with and without the usage of OVO. Their results show that the OVO decomposition improves all the baseline classifiers in terms of accuracy when the data are corrupted by class noise, which could be due to the distribution of the noisy examples in the subproblems, and collection of information from different classifiers [171].

3.4 Addressing the mislabeling problem in remote sensing

.

Remote sensing with sensors mounted on satellites or aircrafts is much needed for disaster response, homeland defense, resource management and environmental monitoring [40]. Remote sensing data considered include those from multispectral, hyperspectral, radar, optical, and infrared sensors.

Classification is often one or the major tasks in remote sensing information processing. However, the presence of noise in remote sensing data degrades the interpretation ability of the data. In particular, mislabeled training data is inevitable in remote sensing where training data sources are typically ground-based [107, 142]. The quality of supervised land cover mapping methods depends on the accuracy of the classifier used to produce the map, and the quality of the labeled data used to train the classifier [162]. Unfortunately, labeling instances to create training sets is time-consuming. Human interpretation and labeling of training sites can require several person-hours of effort per site, depending

on the complexity of the regional land cover and the quality of source data. Despite this level of labor intensity, labeling errors (class noise) still occur, the source of which are subjectivity in data interpretation, inadequacy of source information used for labeling, and other human errors [143]. As a result, creation of training data is one of the most costly, error-prone, and subjective parts of the classification process [162]. Several class noise handling methods presented above have been successfully used in remote sensing. Their goal is to improve the ability to interpret the remote sensing data, often of high complexity, and increase the classification accuracy.

3.4.1 Class noise handling methods

Artificial Neural Network (ANN) is a nonparametric method, which has been used successfully for the classification of diverse remote sensing data, mainly because neural network classifiers are believed to out-perform standard statistical classifiers [101, 202]. Rogan et al. gave a comparison of the performance of a fuzzy artificial neural network and two classification tree algorithms (S-Plus and C4.5) in the context of mapping land-cover changes using Landsat-5 TM (Thematic Mapper) multispectral images [168]. The performances of three methods were assessed using classification accuracy measures and the model data set was modified to test the effect of training data errors (i.e., class noise). Their results show that artificial neural network is the most robust and most accurate classifier in mapping land-cover changes in the presence of class noise compared to S-Plus and C4.5 when the noise level is below 30%.

Hosseini et al. eliminate class noise in remote sensing imagery through a moving window and majority local filter. A multi-spectral image (low spatial resolution Landsat ETM+ (Enhanced Thematic Mapper Plus)), including cultivation, forest and urban sites, is used in this study [101]. However, the size of neighbourhood for the noise filter has to be very large for noise to be sufficiently removed, while a large size neighbourhood may alter the boundaries between classes and create zigzag bounding polygons. Moreover, meaningful information in classified data is ignored because of the geometric and dimensional non-correspondence to real objects with the moving window implementation matrix [157]. To avoid these drawbacks, Qian et al. use the K-mutual neighbours graph to separate the data points into noisy and true data. K-nearest neighbours and majority voting methods are used to correct the class labels of noisy data [157].

Class noise in remote sensing imagery can also be tackled by semi-supervised leaning. In [124], context-sensitive semi-supervised SVMs (Support Vector Machines) first use labeled instances to label unlabeled instances that are spatially close to them and then these new semi-labels and a fixed window-based postfiltering are used to reduce the effects of mislabeled training instances in the classification map. Semi-supervised learning can keep the distribution of the instances unaltered. Hence, it is more suitable for remote sensing data processing. Indeed, training data imbalance in remote sensing data can hinder the efficiency of data cleansing methods. This is due to the fact that minority instances, which are also more likely to be misclassified, may be more likely to be wrongly (false positives) removed by classification filtering, which makes learning even more difficult in noisy classification tasks [68].

3.4.2 Ensemble-based class noise handling methods

Studies have demonstrated the successful application of ensemble classification techniques to land cover mapping [19, 87, 167]. Du et al. [59] evaluated the effectiveness of two popular ensemble algorithms, bagging and boosting, against noise on multi-source remotely sensed images, including a QuickBird multispectral image, a hyperspectral image (OMISII) and a multi-spectral image (low spatial resolution Landsat ETM+). The accuracy assessment demonstrates that bagging is more robust to class noise in remote sensing image classification [59].

As discussed in section 3.3.2, boosting methods tend to overfit class noise. Le Saux tried to find an appropriate loss function to improve the robustness of boosting [123]. The capacities to handle mislabeled data of five different loss functions are tested on a QuickBird multispectral image. The results show that DoomII loss [184] has the highest performance with a limited amount of labeling noise, while with an increased mislabeling level ($>$ 20% of mislabeled inputs) Savage loss [138] performs better. However, Frenay and Verleysen show that the class noise-robust methods are adequate only for simple cases of class noise that can be safely managed by overfitting avoidance [68].

Random forest [26], a powerful machine learning classifier, combines the bagging idea and random selection of features [3, 80, 87, 208]. It has been successfully applied to multi-class classification tasks in remote sensing [3, 57, 87, 143]. Some studies demonstrated that random forest has very good performance in noisy multispectral remote sensing imagery. [119, 167] which does not come as a surprise as random forest is well-known in ensemble learning for its robustness to noise [17]. In [167], the effectiveness of random forests for land-cover classification on Landsat-5 TM data is assessed. The results show that random forest has low overtraining probability and provides robustness to class noise. In [119], the authors also successfully addressed the presence of class noise in multispectral aerial imagery. Their study confirms that random forests can tolerate some mislabeling of the data.

Ensemble noise filtering has been successfully used on remote sensing data. An empirical evaluation of the consensus filter, which has been presented in section 3.3.1.1, demonstrated that this class noise removal method improves classification accuracy for a land-cover mapping task on a high resolution multi-source image (each pixel is described by a time series of twelve NDVI (Normalized Difference Vegetation Index) values and by its latitude) for which the training data contains mislabeled samples [28]. For noise levels up to 20%, noise filtering allows the base-line accuracy to be retained. However, an evaluation of the precision of the approach shows that consensus filters are conservative in throwing away good data at the expense of keeping mislabeled data. Hence, it removes just a little portion of noise [82] making this method less effective in presence of high levels of class noise. Besides, retaining noisy data hinders performance more than throwing out good data [29].

The effectiveness of an ensemble margin-based class noise removal method [82, 84], described in section 3.3.1.3, is demonstrated in performing mapping of land covers [82]. An airborne urban image of 25cm spatial resolution, a multi-source dataset combining lidar and image data and a multispectial Landsat data of 80m spatial resolution are used

to evaluate this algorithm. *Boosting*, a well-known noise sensitive ensemble classifier, is used to assess the quality of the resulting filtered training sets. The results show that this ensemble method for noise removal is effective for land cover mapping from noisy remotely sensed data with noise levels between 10 and 30 % [82].

3.5 Conclusion

Class noise is a complex phenomenon with many potential consequences on classification outcome especially in remote sensing where training data usually present a significant amount of mislabeled instances. There exist many different techniques to address class noise, which can be classified as noise removal, noise correction or class noise-robust methods. Not a single method is completely effective for all noisy data. So, the machine learning practitioner has to choose the method whose definition of class noise appears as the most relevant in his particular field of application. For example, if class noise is only marginal, class noise-robust methods could be sufficient. Eventually, most data cleansing methods are easy to implement and have been shown to be efficient and to be good candidates in many noisy situations. In addition, in several works, it has been observed that simply removing mislabeled instances is more efficient than correcting them. However, instance removal methods may remove too many uncorrupted instances. The overcleansing problem is of particular importance for imbalanced datasets which are common in remote sensing. On the other hand, keeping mislabeled instances may harm more than removing too many correctly labeled samples. Therefore, a compromise has to be found.

Ensemble learners, especially random forest, are more robust than single classifiers and have been successfully used to deal with the mislabeling problem. Indeed, all ensemble-based class noise handling methods can be interpreted as making particular assumptions. First, in data cleansing methods, different heuristics are used to distinguish mislabeled instances from exceptions. Each heuristic is a definition of what is class noise. Second, in class noise-robust methods, overfitting avoidance is assumed to be sufficient to deal with class noise. Therefore, the success of ensemble-based methods against noise contributes to a good compromise between directly using instances as they are and finding any instance that is possibly mislabeled.

There are many open research questions related to class noise and many avenues remain to be explored. Semi-supervised learning has the advantage of not altering the distribution of the instances and it could be interesting to investigate whether this scheme does improve the class noise handling performances with respect to simply removing suspicious instances from noisy data. Decomposition in multiclass problems can change the distribution of noisy examples in resulting subproblems and increase the separability of the classes, and hence can be used for noise detection or, more generally, for data selection. Ensemble margin has also great potential for classifier design against noise and for noise identification as demonstrated by some recent work that appeared in literature. Indeed, the generalization performance of an ensemble classifier is related to the distribution of its margins on the training examples. In addition, random forest

has been proved to be the most robust method in ensemble learning. The effectiveness of noise filtering in random forest classification is an interesting research direction to explore.

Chapter 4

Class noise filtering using ensemble margin

This chapter presents an ensemble margin-based method to address the mislabeling problem. Two kinds of artificial class noise which will be used in our experiments are described in section 4.2. A novel unsupervised margin definition is proposed in section 4.3. In section 4.4, we study the suitability of two popular ensemble margins as well as the proposed new margin for class noise identification. Section 4.5 presents the ensemble margin based class noise ordering which is an important step in noise filter design. Sections 4.6 and 4.7 describe our margin based noise removal and correction schemes respectively. The experimental results are reported in section 4.9. Section 4.10 summarizes our work.

4.1 Introduction

This chapter focuses on the classification of noisy datasets. In the previous chapter, we have presented different data cleaning algorithms such as ensemble based class noise filtering [29], [204], [33], which considers the fusion of classifiers to address the classification of noisy instances and has already been highlighted as having a better behavior with noisy data in the field of classification compared with single filters [68], [194] as well as noise robust classifiers [1], [121].

In ensemble learning, ensemble margin [86] is acknowledged as an important factor for improving the generalization performance of classifiers [4, 83]. In the following, we present an ensemble margin-based class noise elimination method to deal with the class noise problem of real world data sets. This method can achieve a high mislabeled instance detection rate (*true positives*) while keeping the false detection rate (*false positives*) as low as possible. The main difference between our margin based and existing ensemble-based noise filters [29], [197] is that it not only adopts the ensemble vote to distinguish misclassified and correctly classified instances, but also explicitly takes into

account, through the ensemble margin, the probability of the misclassified instances being identified as noise. Hence, this method could also be considered as a probability based noise filter.

There are two major ensemble margin definitions, both supervised, which are presented in chapter 2. But, *which margin is the most suited for noise filter design?* We propose a novel unsupervised margin in this chapter. This margin has the appealing property of not involving the class labels and thus should be potentially more robust to class noise.

Since noise removal is known for its risk of removing useful instances, noise correction methods which are designed via relabeling noisy instances [147], [161] are proposed to avoid the reduction of informational samples. However, the noise correction methods reported in the previous chapter do not fully satisfy the demands of most applications due to a variety of reasons such as low automation or imprecise identification of noise. Our margin based noise removal method has the advantage of more accurately distinguishing mislabeled instances and corrected classified instances. Therefore, we extend the margin based mislabeled instance elimination by tackling the class noise correction.

Two well-known noise sensitive classifiers, *boosting* [175] and *K-Nearest Neighbors (KNN)* [45], are used to assess the quality of the resulting filtered training sets. A comparative analysis is conducted with respect to the majority vote filter [29], also an ensemble-based mislabeled training data identification approach. Although the majority vote filter [29] misidentifies some clean instances as noise, its easy implementation makes this popular method still attractive [174, 183, 197].

4.2 Introducing artificial noise into the data

A class noise (or mislabeling) is an instance whose label value conflicts with most of the other instance label values while having the same or similar attribution values. It implies that most base classifiers in the ensemble classified this instance as another class. In other words, this instance was classified wrongly with high confidence [82]. Noisy distributions are application dependent and are generally unknown. In our research work, two kinds of artificial noise are used to simulate mislabeled data.

4.2.1 Standard approaches for artificial noise generation

In the binary classification mislabelling experiment of [143], training data mislabelling was undertaken by randomly re-assigning a proportion of $\boldsymbol{A}$ class instances as $\boldsymbol{B}$ class and $\boldsymbol{B}$ instances as $\boldsymbol{A}$. While introducing artificial noise into binary class training data is a straightforward process, it is not always the case for multiclass problems. Studies typically use random class-label switching to simulate noise in a classification [29, 85, 182, 197, 214]. They randomly chose a subset of $\boldsymbol{a\%}$ from the whole training set. The class label values of these randomly selected examples are randomly labeled to another label.

4.2.2 Confusion-matrix based artificial noise

In [143], an alternative approach is proposed for artificial mislabeling designed to replicate realistic real-world operator misclassification (mislabelling) of reference data instances [134], that results in a more reliable analysis of noise effects on a supervised classifier performance. In its multi-class mislabelling experiment, a preliminary random forest model was built using a multi-class training data. A confusion matrix derived from the OOB (Out of Bag) data (not used in the bootstrap training samples) was used to determine, for each class c_i, the class to which it was most frequently misclassified l_i , i.e. the most frequent erroneous class predicted by the model from the OOB data. For the multi-class classification, starting with a training data set with more or less "real" noisy labels whose amount is unknown in practice, the introduction of artificial noise in class labels is performed by mislabelling a proportion of each class c_i to l_i.

In the process of producing random noise, all instances have the same probability to be mislabeled. However, confusion matrix based noise mainly affects class decision boundaries. Hence, confusion matrix based noise is more difficult to be identified than random noise.

4.3 New ensemble margin

In this section, we propose a novel unsupervised ensemble margin alternative defined as equation (4.1), where v_{c_1} is the votes number of the most voted class for sample x and T represents the number of base classifiers in the ensemble. The proposed margin is an unsupervised version of the classic sum-margin referred to as equation (2.3), it does not require the true class label of instance x. Hence, it is potentially more robust to class noise. This new margin will be named as *unsupervised sum-margin*. The range of this margin is from 0 to 1.

$$\begin{aligned} margin(x) &= \frac{v_{c_1} - \sum_{c=1,\ldots,L \cap c \neq c_1}(v_c)}{\sum_{c=1}^{L}(v_c)} \\ &= \frac{2v_{c_1} - T}{T} \end{aligned} \tag{4.1}$$

The proposed margin also has the advantage to be considered as a classifier evaluation function or adopted for classifier design in unsupervised or semi-supervised ensemble learning.

4.4 Ensemble margin for class noise filtering

4.4.1 Effect of class noise on ensemble margin distribution

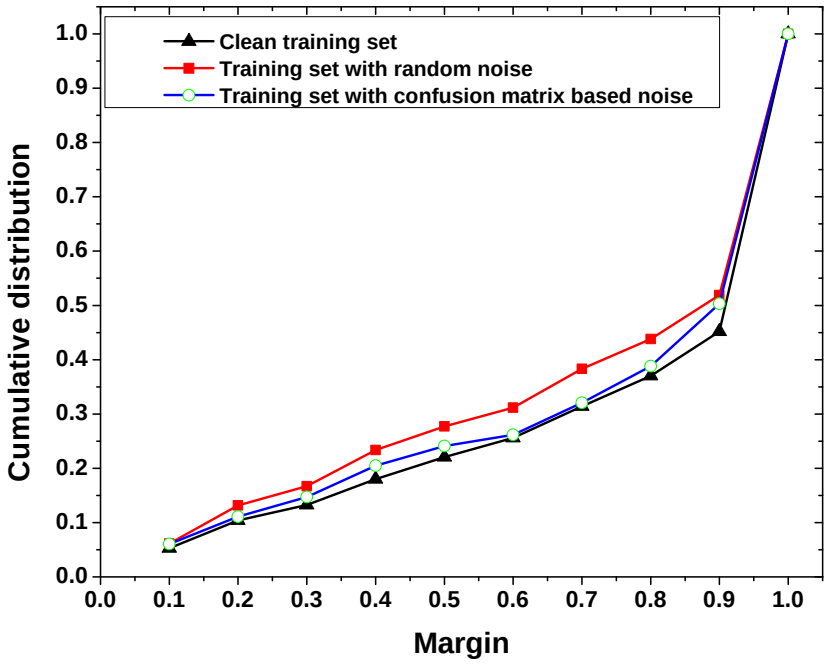
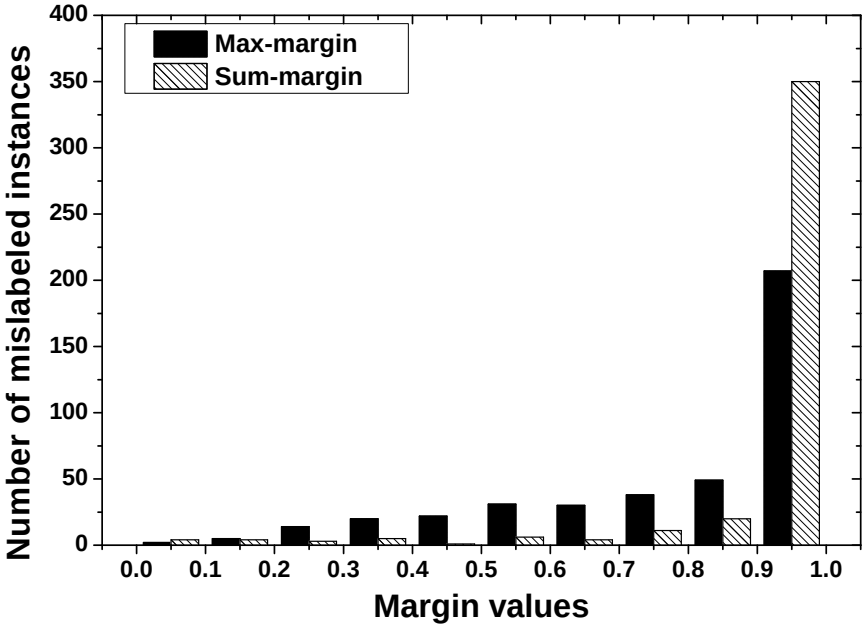


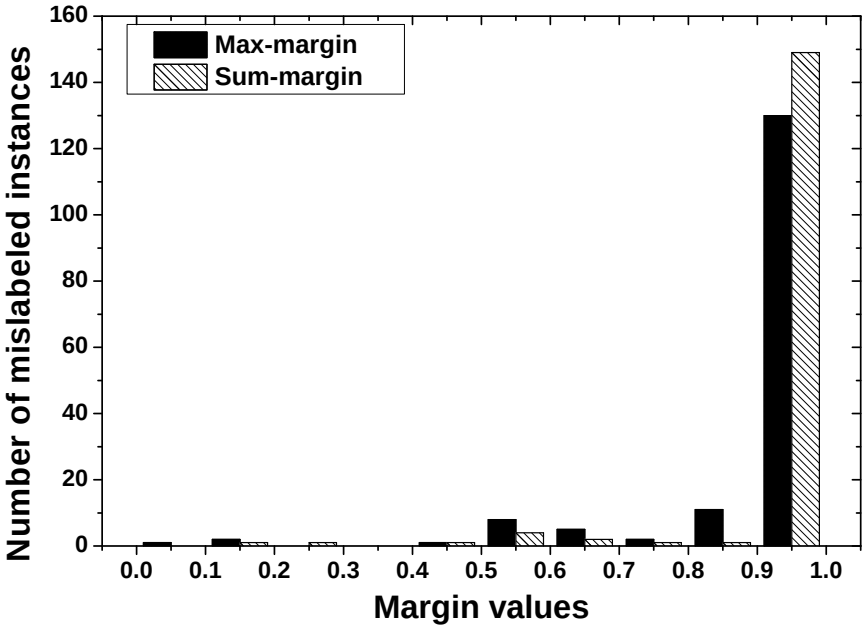
Figure 4.1: Training margin distribution of bagging with clean and noisy training set on data set *Pendigit* using a new ensemble margin.

The margin distribution of training instances effectively reflects the performance of an ensemble algorithm [176]. When a model classifies a data set correctly with high probability, these instances should obtain high margin values. The presence of noise can weaken the performance of a classifier and lead to a smaller margin distribution. Figure 4.1 shows the training margin distribution of bagging involving decision tree as base learner on data set *Pendigit* (table 7.1) using the new ensemble margin in the case of both clean and altered with 20% of random and confusion-matrix based artificial noise respectively. From the margin plots, it can be seen that both kinds of noise result expectedly in smaller training margins compared with the use of a clean training set for ensemble construction. Moreover, random noise leads to more instances with lower margin values compared with confusion-matrix based class noise in the new margin distribution, i.e. the former leads to a lower mean margin than the latter.

4.4.2 Max-margin versus sum-margin

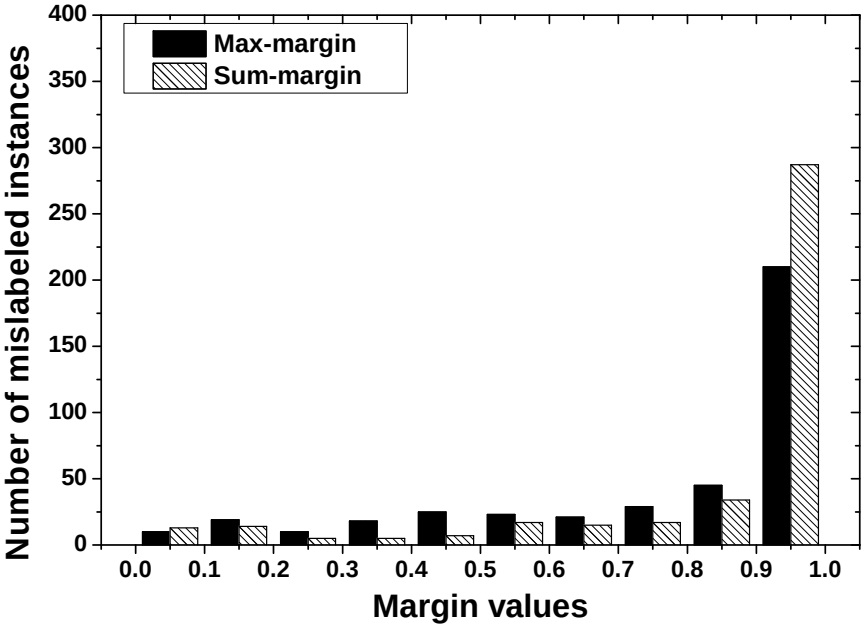


(a) Pendigit

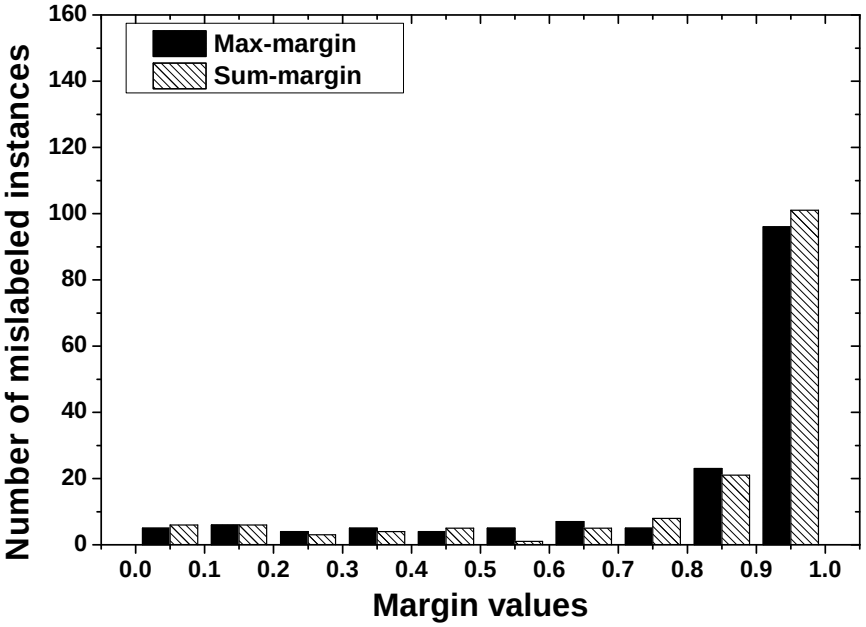


(b) Segment

Figure 4.2: Max-margin and sum-margin distributions of true random class noise on data sets *Pendigit* and *Segment*.



(a) Pendigit



(b) Segment

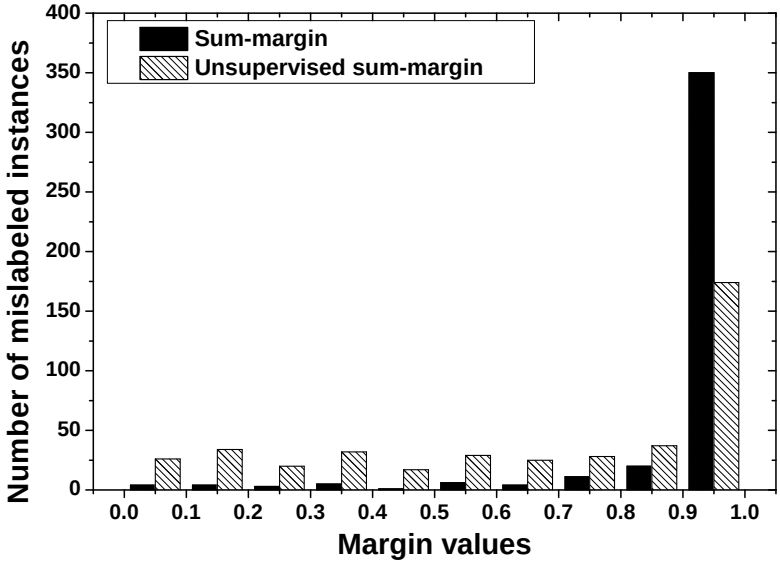
Figure 4.3: Max-margin and sum-margin distributions of true confusion-matrix based artificial noise on data sets *Pendigit* and *Segment*.

The main difference between max-margin (equation 2.2) and sum-margin (equation 2.3) is indicated in [113]. While the definition of sum-margin applies a sum operation

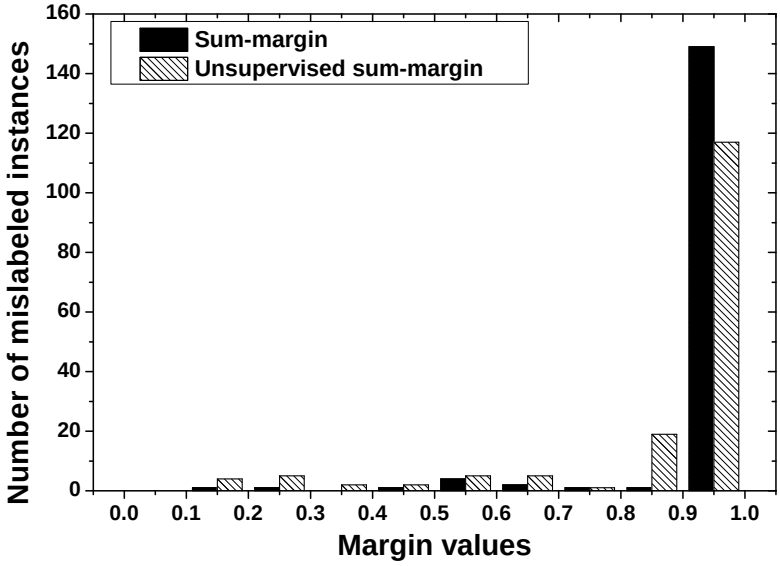
on ensemble votes, the definition of max-margin uses a maximum operation. In this section, we aim to compare the performances of max-margin and sum-margin in class noise identification. The ideal margin distribution for effective class noise identification is a margin distribution where all true mislabeled samples obtain high absolute values. According to the definitions of the two margins (equations 2.2 and 2.3), a true mislabeled instance could obtain a higher margin (in absolute value) with a sum operation and thus be identified with higher probability. For example, in a multi-class problem, let us assume the number of ensemble votes of a noisy sample for the true class is 20, and the number of votes for any other class is 40, with an ensemble size of 100. Then, while the max-margin (in absolute value) of the sample is 0.2, the sum-margin (in absolute value) of the sample achieves 0.6. Consequently, among the two possible definitions of ensemble margin, the second one, based on a sum operation, should be potentially more successful in mislabeled data identification. Figures 4.2 and 4.3, which show the margin distribution histograms of true class noise on data sets *Pendigit* and *Segment* (table 7.1) with 20% noise level in the cases of random and confusion-matrix based artificial noise respectively, confirm our assumption. The usage of sum-margin could result in cleaner training set as well as more accurate classification as will be shown in our experimental results later.

4.4.3 Supervised versus unsupervised margin

The main objective of this section is to carry out a feasibility analysis of exploiting the margin for noise identification. Because an unsupervised margin does not require the true class label of an instance, it has an appealing advantage over a supervised margin: *it can be involved in a semi-supervised ensemble learning scheme*. Moreover, an unsupervised margin is potentially more robust to noise as it is not affected by errors occurring on the class label itself. Figures 4.4 and 4.5 perform a comparison of the sum-based margin and its unsupervised version that we have proposed (equation 4.1) on data sets *Pendigit* and *Segment* which are corrupted by random and confusion-matrix based artificial noise with 20% level respectively. The supervised margin tends to make more mislabeled instances obtain high margin (in absolute value) with respect to the unsupervised margin on the two data sets, suggesting potentially greater capability in mislabeled data identification.

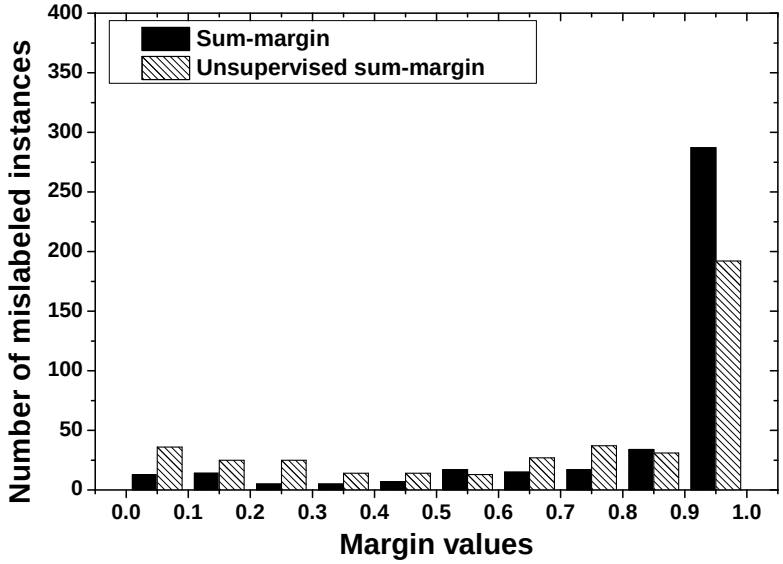


(a) Pendigit

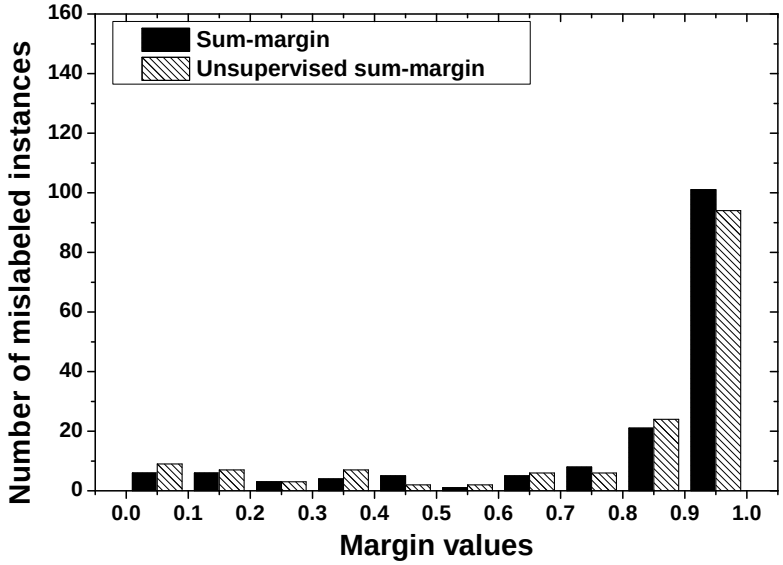


(b) Segment

Figure 4.4: Supervised and unsupervised margin distributions of true random class noise on data sets *Pendigit* and *Segment*.



(a) Pendigit



(b) Segment

Figure 4.5: Supervised and unsupervised margin distributions of true confusion-matrix based artificial noise on data sets *Pendigit* and *Segment*.

4.5 Ensemble Margin-based class noise ordering

Each training instance has a probability of being mislabeled. However, these probabilities are different depending on instance features and behavior in the training process.

The objective of noise removal is to eliminate the most likely noisy instances. Ordering training instances according to their probability of being mislabeled is a simple and efficient method for noise removal [82]. In [82], these probabilities rely on the margin values of training instances involving an unsupervised ensemble margin (equation (2.3)) which is an unsupervised version of the classic max-margin [82].

Let us consider an ensemble classifier C, and a set of n training data denoted as $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where x_i is a vector with feature values and y_i is the value of the class label. The mislabeled instance ordering approach, introduced in [82], simply relies on an ensemble margin's definition as a class noise evaluation function, slightly modified here, defined as (4.2). This method assesses only the training instances x_i whose attribution and label values are not consistent.

$$N(x_i) = |margin(x_i)| \quad \forall (x_i, y_i) \in S \mid C(x_i) \neq y_i \tag{4.2}$$

The higher $N(x_i)$, the higher the probability of x_i being mislabeled.

4.6 Ensemble margin based class noise removal

4.6.1 Noise filter

Our mislabeled training data identification method is based on noise ordering [82] and relies on the ensemble margin. The first step of our noise removal method involves a robust ensemble classifier: *bagging* (with pruned decision trees) which is constructed using the whole training set. The margin value of each training instance is then calculated. Our method orders misclassified training instances according to their margin values. The higher the margin (in absolute value) of a misclassified instance, the higher the probability this instance is being mislabeled. The two following steps of our algorithm rely on a noise-sensitive ensemble classifier: *boosting*. The second step aims at selecting the best filtered training set cleaned out of any mislabeled data. Using a more robust ensemble classifier such as bagging or random forests to estimate the noise rate and select the best training set is not the best choice. Indeed, robust ensembles tolerate a certain amount of noise and therefore would fail to detect it.

The class noise rate is automatically estimated through an iterative procedure that removes an amount M (from 0 to gradually 40% of the total training set size) of ordered potential mislabeled instances from training set and assesses the classification accuracy of *boosting*, built with the filtered training set, on a validation set. This adaptive strategy selects then the filtered training set that led to maximum accuracy on validation set. In the last step, *boosting* is involved again to assess the quality of the resulting filtered training set via a classification accuracy assessment procedure. A single noise-sensitive classifier, namely *KNN*, is also used (instead of *boosting*) in the last steps of our mislabeled training data filter.

Relying on the margin-based noise evaluation function 4.1, the ordering-based mislabeled instance elimination algorithm consists of the following steps:

Algorithm 3: Margin based noise removal

Inputs:

1. Training set $S = (x_1, y_1), (x_2, y_2,), \cdots, (x_n, y_n)$;
2. Validation set V;
3. Ensemble creation algorithm C;
4. Number of iterations M (noise amount);

Process:

1. Constructing an ensemble classifier C with all the n training data $(x_i, y_i) \in S$.
2. Computing the margin of each training instance x_i.
3. Ordering all the training instances x_i, that have been misclassified, according to their noise evaluation values $N(x_i)$, in descending order.
4. **For** $m = 1$ to M **do**
   a) Eliminating the first m most likely mislabeled instances x_i to form a new cleaner training set S'_m.
   b) Evaluating the cleaned training set S'_m by classification performance, on a validation set V.

   End
5. Selecting the best filtered training set S' which led to maximum accuracy on validation set V.

Output: Best filtered training set S'.

4.6.2 Iterative guided noise filter

A novel extended version of our margin-based noise removal method is proposed here. This iterative guided noise filter relies on an adaptive calculation of the margin values of each training instance. In the original version of our algorithm (Algorithm 3), the training margins are determined once only, in the first step. The training margins being at the core of our noise evaluation procedure, a sensible strategy (though more costly) would consist in updating them at each noise removal step (second step of our algorithm). Hence, a robust ensemble is constructed again (at each iteration) with the reduced filtered training set, cleaned out of a fixed (1% of the current training set size) amount of potential noise. New training margins are then calculated to use as inputs in

the next step. This adaptive strategy selects then the filtered training set that led to maximum accuracy on validation set. In the last step, *boosting* is involved again to assess the quality of the resulting filtered training set via a classification accuracy assessment procedure.

Algorithm 4: Iterative margin-based class noise removal

Input:

1. Training set $S = (x_1, y_1), (x_2, y_2,), \cdots, (x_n, y_n)$;
2. Validation set V;
3. Ensemble creation algorithm C;
4. Number of iterations M (noise amount);

Iterative process:

1. **For** $m = 1$ to M **do**

   If m=1

   a) Constructing an ensemble classifier C_1 with the n training data $(x_i, y_i) \in S$.

   b) Evaluating the original training set S by classification performance, on a validation set V.

   c) Training data S'_1=S

   else

   a) Constructing an ensemble classifier C_m with the training set S'_m.

   b) Computing the margin of each training instance x_i of S'_m.

   c) Ordering all the training instances x_i of S'_m, that have been misclassified, according to their noise evaluation values $N_m(x_i)$, in descending order.

   d) Eliminating the first 1% most likely mislabeled instances x_i to form a new cleaner training set S''_m.

   e) Evaluating the cleaned training set S''_m by classification performance, on a validation set V.

   f) New training set S'_{m+1}=S''_m

   End

2. Select the best filtered training set S'' which led to maximum accuracy on validation set V.

Output: Best filtered training set S''.

4.7 Ensemble margin based class noise correction

4.7.1 Noise filter

Noise removal can discard some useful data, so we also attempt to automatically correct the training instances that have been identified as mislabeled (highest margin misclassified instances). Noise correction has been shown to give better results than simply removing the noise from the data set in some cases [192]. In a data correction scheme, the noisy instances are identified, but instead of removing these instances out, they are repaired by replacing corrupted values with more appropriate ones [192]. The labels of the most likely mislabeled instances are changed to the predicted classes. Then, these corrected instances are reintroduced into the training set. Our class noise correction method relies on an adaptive strategy that is similar to our class noise removal method. But, instead of removing an amount M of noise from training set at each step, it automatically corrects the detected noise using the predicted labels by the constructed bagging ensemble. This ordering-based mislabeled instance correction algorithm consists of the following steps summarized in Algorithm 5:

Algorithm 5: Margin based noise correction

Inputs:

1. Training set $S = (x_1, y_1), (x_2, y_2,), \cdots, (x_n, y_n)$;
2. Validation set V;
3. Ensemble creation algorithm C;
4. Number of iterations M (noise amount);

Process:

1. Constructing an ensemble classifier C with all the n training data $(x_i, y_i) \in S$.
2. Computing the margin of each training instance x_i.
3. Ordering all the training instances x_i, that have been misclassified, according to their noise evaluation values $N(x_i)$, in descending order.
4. **For** $m = 1$ to M **do**
   a) Correcting the labels of the first m most likely mislabeled instances x_i to form a new cleaner training set.
   b) Evaluating the cleaned training set S'_m by classification performance, on a validation set V.

   End

5. Select the best corrected training set S' which led to maximum accuracy on validation set V.

Output: Best corrected training set S'.

4.7.2 Iterative guided noise filter

An iterative version of our noise correction algorithm is also proposed in this chapter. This *iterative guided noise correction* method relies on a repetitive calculation of the training margins that is similar to the margin distribution update step involved in our iterative class noise removal method. But, instead of removing a portion of potential mislabeled instances (at each iteration), it relabels the detected class noise using the predicted labels by the constructed bagging ensemble. Hence, unlike the removal scheme, the total number of training samples remains the same. The main steps of our iterative guided noise correction method are summarized in the following algorithm.

Algorithm 6: Iterative margin based noise correction

Inputs:

1. Training set $S = (x_1, y_1), (x_2, y_2,), \cdots, (x_n, y_n)$;
2. Validation set V;
3. Ensemble creation algorithm C;
4. Number of iterations M (noise amount);

Iterative process:

1. **For** $m = 1$ to M **do**

   If m=1

   a) Constructing an ensemble classifier C_1 with the n training data $(x_i, y_i) \in S$.

   b) Evaluating the original training set S by classification performance, on a validation set V.

   c) Training data S'_1=S

   else

   a) Constructing an ensemble classifier C_m with the training set S'_m.

   b) Computing the margin of each training instance x_i of S'_m.

   c) Ordering all the training instances x_i of S'_m, that have been misclassified, according to their noise evaluation values $N_m(x_i)$, in descending order.

d) Correcting the labels of the first **1%** most likely mislabeled instances x_i to form a new cleaner training set S''_m.

e) Evaluating the cleaned training set S''_m by classification performance, on a validation set V.

f) New training data S'_{m+1}=S''_m

End

2. Select the best corrected training set S' which led to maximum accuracy on validation set V.

Output: Best corrected training set S'.

4.8 Discussion

1. Unlike traditional class noise processing methods which remove hypothetical mislabeled instances according to some simple rules such as the majority vote or the consensus noise filters [29], the margin based algorithm, which can also be categorized as a probability based method, identifies noisy samples depending on the associated margin values that result from a majority voting of a noise robust ensemble.

2. Class decision boundary and minority class instances, which are the most informative in classification, are easy to be misclassified in machine learning. Thus, they have a high risk of being treated as noise and removed by simple noise filters such as the filter reported in [194] where all the misclassified instances are removed or the weight based ensemble based filter [197] which removes all the high weight samples consequently affecting boundary data and minority class samples. In the margin ordering based approach, class decision boundary and minority samples have low margin values. Therefore, they are at low risk of being discarded while the mislabeling problem is alleviated effectively by targeting high margin misclassified instances.

3. Robust ensembles such as bagging are more suitable to obtain with high accuracy misclassified samples as well as their margin values in the proposed margin-based class noise identification algorithm. Noise sensitive learners have higher risk of misclassifying difficult instances in corrupted environment, that will lead to some unnecessary complications to data cleaning work and poorer class noise identification performance.

4. Theoretically, the margin based noise correction approach should achieve better performance than the margin based noise removal because it keeps all the training samples. However, the correction scheme has a high requirement on the predicted labels of class noise, i.e. the produced label from an ensemble should be close to

the true class of a mislabeled instance with high confidence. Hence, the ensemble learner whose main task is to produce an appropriate training margin distribution for noise identification has a significant impact on the success of our noise correction scheme.

5. The extended iterative guided approach identifies class noise by taking into consideration the margin distribution of a different training set in each iteration. In other words, instead of using one ensemble to produce margin values of potential noise once, the iterative scheme aims to improve the accuracy of noise identification via the fusion of diverse ensembles. It is noteworthy, however, that the iterative scheme has higher computational costs.

A comparative analysis is conducted between our margin-based mislabeled data identification method and the majority filter [29]. Both class noise removal and correction schemes are involved in the comparison. Each of the three different ensemble margins, defined in chapter 2 as well as the new margin introduced in this chapter, are involved in the validation of our algorithms.

4.9 Experimental results

4.9.1 Data sets

We applied the class noise removal and correction methods on 10 data sets including 5 image data sets (top 5) and 5 non image data sets (bottom 5) from UCI Machine Learning repository [8] (table 7.1) including two imbalanced data sets (*Glass* and *Wine quality-red*). Each data set has been divided into three parts: training set, validation set and test set, as shown on table 7.1. For a fairer comparison, we included the validation in the training data when the validation set was not necessary (fixed and majority filters).

In all the tables of the following experiments, the best performance for each data set is highlighted in bold. The asterisk is utilized to mark the margin which has the best performance among the four margin definitions in adaptive noise filtering.

Dataset	Training set	Validation set	Test set	Variables	Classes
Letter	5000	2500	5000	16	26
Optdigits	1000	500	1000	64	10
Pendigit	2000	1000	2000	16	10
Vehicle	200	100	200	18	4
Segment	800	400	800	19	7
Abalone	1500	750	1500	8	3
Glass	80	40	80	10	6
Waveform	2000	1000	2000	21	3
Wine quality-red	600	300	600	11	6
Texture	2000	1000	2000	40	11

Table 4.1: Data sets.

4.9.2 Class noise filtering using random noise

In this section, we test the performances of both noise removal and correction based on one step training margins calculation in the presence of random noise. We randomly chose a subset of 20% from the whole training set and whole validation set respectively. The class label values of these selected examples were randomly labeled to another label. Two noise filtering strategies are experimented. The first one is adaptive and involves the elimination (Algorithm 3) or correction (Algorithm 5) of an amount of ordered potential mislabeled instances equal to the one that led to maximum accuracy on validation set. The second one just eliminates or corrects a fixed amount equal to the noise rate (20%). The margin-based approach involves the two popular definitions of ensemble margin: equations 2.2 and 2.3 and their unsupervised versions including the novel proposed margin (equations 2.4 and 4.1).

4.9.2.1 Noise removal assessment

4.9.2.1.1 Overall classification accuracy

Tables 4.2 and 4.3 show respectively the accuracy of *AdaBoost.M1* and *1-NN* without noise filtering, and by noise filtering for both majority vote and margin-based methods in the presence of random noise. These tables show that the margin-based mislabeled data removal scheme significantly outperforms the majority vote filter. The accuracies achieved by adaptive filtering and by a fixed amount of filtering are slightly in favor of the fixed strategy (at least for the best performances, indicated in bold). However, the adaptive strategy does not require the knowledge of the noise rate (which is generally unknown) and leads to a more automated noise filtering procedure.

Supervised margins are more effective for class noise identification than unsupervised margins. Among the two possible definitions of ensemble margin, the second one, based

on a sum operation, is the most successful for mislabeled data removal. This result is rather expected as a sum operation is more robust to noise than a max operation, at the core of the $\mathbf{1}st$ definition of ensemble margin. The new introduced margin (equation (4.1)), although not as efficient as its supervised counterpart (equation (2.3)), is also effective to identify mislabeled data and outperforms the majority vote filter. Additionally, the two unsupervised margins achieve similar performances for noise removal.

Data	No filter	Majority filter	Margin-based noise removal							
			Max-margin		Unsupervised max-margin		Sum-margin		Unsupervised sum-margin	
			Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.
Letter	46.7	47.8	49.9	52.0	49.9	49.6	52.4	**56.9***	48.5	50.5
Optdigit	89.3	90.8	94.5	93.4	93.1	93.2	**94.7**	94.1*	93.9	93.2
Pendigit	90.3	93.0	93.2	95.3	92.7	94.2	**95.9**	95.4*	92.8	93.9
Vehicle	72.2	73.7	73.5	72.3	72.6	72.1	**74.1**	72.1	72.8	73.0*
Segment	92.1	91.1	93.6	94.0	93.5	94.2	93.4	**94.9***	93.3	94.2
Abalone	54.2	54.1	53.9	54.5*	54.3	54.1	54.7	54.4	**54.7**	54.0
Glass	97.7	97.5	97.5	96.8*	**98.8**	96.6	97.5	96.3	97.5	96.5
Waveform	81.6	79.0	82.2	82.4*	81.6	81.8	**82.7**	80.9	81.8	82.1
Wine qual -ity-red	60.7	59.8	**62.0**	60.5	**62.0**	59.7	61.4	60.6*	61.6	59.3
Texture	86.3	89.5	88.5	91.7	86.9	90.4	**94.0**	93.9*	87.2	90.5
Average accuracy	77.1	77.6	78.9	79.3	78.5	78.6	**80.1**	79.9*	78.4	78.7

Table 4.2: Accuracy of $AdaBoost.M1$ classifier with no filter, with majority vote filtered and with four margin-based filtered training sets, using both an adaptive and a fixed amount of filtering in the presence of random noise.

Data	No filter	Majority filter	Margin-based noise removal							
			Max-margin		Unsupervised max-margin		Sum-margin		Unsupervised sum-margin	
			Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.
Letter	74.6	59.1	81.1	79.9	78.1	78.2	**87.1**	85.1*	79.2	77.6
Optdigit	77.9	93.3	92.5	93.0	91.1	92.9	**93.6**	93.1*	91.0	93.0
Pendigit	79.8	95.1	94.9	**96.3***	94.1	95.9	96.2	95.9	93.5	94.2
Vehicle	59.0	66.5	68.5	63.0	68.5	64.0	**70.0**	**70.0***	69.0	66.0
Segment	81.5	90.3	90.3	91.6	90.5	91.5	92.9	**93.3***	90.0	91.8
Abalone	45.1	**53.2**	51.3	49.7	49.9	49.7	52.7	51.3*	49.8	49.5
Glass	63.8	73.8	**75.0**	71.3*	**75.0**	68.8	**75.0**	68.8	**75.0**	65.0
Waveform	62.8	**78.1**	76.6	75.5	74.6	77.6*	76.9	75.8	74.7	76.7
Wine qual -ity-red	49.7	**60.3**	56.3	57.8*	56.5	56.8	55.3	56.5	55.5	55.7
Texture	80.2	94.4	93.4	93.9*	92.9	93.2	**95.5**	93.8	92.5	93.2
Average accuracy	67.4	76.4	78.0	77.2	77.1	76.8	**79.5**	78.3*	77.0	76.2

Table 4.3: Accuracy of $\mathbf{1-NN}$ classifier with no filter, with majority vote filtered and with four margin-based filtered training sets, using both an adaptive and a fixed amount of filtering in the presence of random noise.

4.9.2.1.2 Per-class classification accuracy

Tables 4.4 and 4.5 compare respectively the classification accuracy achieved by *AdaBoost.M1* and *1-NN*, on test set, for the most difficult class with no filter, with majority vote filtered and with four margin-based filtered training sets, using both an adaptive and a fixed amount of filtering. These tables show that the margin-based mislabeled data removal scheme significantly increases the accuracy on the most difficult class for almost all the data sets while the majority vote method is less effective and even decreases the per-class accuracy of some data sets. Difficult class instances have typically low margin values and hence are at low risk of being removed, our potential mislabeled training data being the highest margin misclassified instances. The new unsupervised margin outperforms the unsupervised max-margin in boosting per-class accuracy performances but is significantly less effective for *1-NN* classifier. In addition, the accuracies of the most difficult class achieved by an adaptive filtering are higher than by a fixed amount of filtering for boosting classification but significantly lower for *1-NN* classification. Furthermore, tables 4.2, 4.3, 4.4 and 4.5 show that the sum operation based margin noise filter induces a faster rise in classification accuracy and per-class accuracy than other margin based noise filters for both ensemble and single classifiers.

Data	No filter	Majority filter	Margin-based noise removal							
			Max-margin		Unsupervised max-margin		Sum-margin		Unsupervised sum-margin	
			Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.
Letter	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Optdigit	77.0	79.3	87.4	82.4	84.6	82.9*	**90.2**	81.4	86.0	79.5
Pendigit	75.4	79.4	78.9	82.5	78.4	82.0	86.6	**89.3***	78.6	82.1
Vehicle	47.3	**51.4**	43.6	43.1*	43.2	42.6	48.0	39.3	49.6	39.6
Segment	77.8	77.0	84.7	85.0	83.7	85.7	80.7	**86.5***	83.6	85.8
Abalone	26.4	26.2	34.9	40.2	34.7	39.6	34.7	**44.0***	33.3	33.5
Glass	5.0	0.0	0.0	**50.0**	**50.0**	**50.0**	0.0	**50.0**	0.0	**50.0**
Waveform	78.2	73.9	79.6	**82.1***	78.6	80.5	80.2	79.4	78.9	80.2
Wine qual -ity-red	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.4**	0.0
Texture	63.1	64.2	69.8	85.6	65.1	74.2	84.9	**89.6***	71.2	81.5
Average accuracy	45.0	45.1	47.9	55.1	51.8	53.7	50.5	**55.9***	48.2	53.2

Table 4.4: Classification accuracy of ***AdaBoost.M1*** classifier for the most difficult class with no filter, with majority vote filtered and with four margin-based filtered training sets, using both an adaptive and a fixed amount of filtering in the presence of random noise.

Data	No filter	Majority filter	Margin-based noise removal							
			Max-margin		Unsupervised max-margin		Sum-margin		Unsupervised sum-margin	
			Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.
Letter	66.1	0.0	38.0	63.2	38.0	64.7	73.1	**75.4***	40.9	64.2
Optdigit	58.9	**83.0**	74.1	81.3	77.7	82.1*	**83.0**	81.3	73.2	81.3
Pendigit	70.8	83.4	86.1	89.8*	81.9	86.5	**90.6**	89.3	80.0	85.1
Vehicle	30.9	**41.8**	**41.8**	32.7	**41.8**	30.9	**41.8**	38.2*	40.0	32.7
Segment	70.3	76.9	75.4	81.8	79.0	77.9	**82.5**	84.3*	79.0	81.4
Abalone	39.0	30.3	35.6	38.4	32.0	34.3	**42.6**	40.7*	31.4	35.4
Glass	**50.0**	0.0	**50.0**	33.3*	**50.0**	33.3*	**50.0**	33.3*	**50.0**	0.0
Waveform	61.1	73.7	73.5	75.0*	71.6	75.0*	74.0	74.6	71.5	73.7
Wine qual -ity-red	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Texture	72.5	85.1	86.2	81.6	81.6	78.7	**90.8**	82.2*	81.6	79.3
Average accuracy	52.0	47.4	56.1	57.7	55.4	56.4	**62.8**	59.9*	54.8	53.3

Table 4.5: Classification accuracy of $\mathbf{1-NN}$ classifier for the most difficult class with no filter, with majority vote filtered and with four margin-based filtered training sets, using both an adaptive and a fixed amount of filtering in the presence of random noise.

4.9.2.2 Noise correction assessment

4.9.2.2.1 Overall classification accuracy

In tables 4.6 and 4.7, organized as tables 4.2 and 4.3, we attempt to correct the training data identified by margin-based or majority vote methods as mislabeled. A comparison of the overall accuracies of *AdaBoost.M1* and *1-NN* on the original training data (no noise correction) and on the corrected training data reveals that margin-based algorithms reach higher accuracy while the ability of the majority vote correction to retain the baseline accuracy decreases particularly for *AdaBoost.M1*. Unlike in noise removal, while the fixed scheme still outperforms the adaptive one for noise correction in *1-NN*, the adaptive scheme turns out more effective in boosting classification. While the sum-based definition of ensemble margin (equation (2.3)) remains beyond all doubt the most appropriate for *1-NN* classifier, it is not the case for *AdaBoost.M1* classifier for which the max-margin achieves the best noise correction performance. Unsurprisingly, the class noise removal scheme outperforms its correction counterpart, for both majority vote and margin-based methods, the noise correction being a more challenging task. Indeed, noise correction algorithms are at high risk of inducing additional noise, and retaining bad data hinders performance more than throwing out good data.

Data	No filter	Majority filter	Margin-based noise correction							
			Max-margin		Unsupervised max-margin		Sum-margin		Unsupervised sum-margin	
			Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.
Letter	46.7	45.6	43.7	**50.4***	43.6	**50.4***	43.6	48.4	43.6	50.3
Optdigit	89.3	88.0	91.7	**93.2***	89.2	92.3	92.4	92.2	87.8	91.3
Pendigit	90.3	89.6	91.9	93.6*	90.5	91.8	**94.0**	93.6*	89.3	91.6
Vehicle	72.2	**74.3**	72.2	73.9*	72.3	72.0	72.6	73.6	72.1	72.8
Segment	92.1	91.1	94.0	94.0	93.8	**94.4***	93.6	94.1	94.0	**94.4***
Abalone	54.2	54.5	53.7	54.5*	54.6	54.3	54.3	54.1	**54.7**	54.0
Glass	**97.7**	97.5	97.5	96.5	97.5	96.9*	97.5	96.9*	97.5	96.6
Waveform	81.6	78.9	81.9	**82.2***	80.6	81.8	82.0	81.3	80.4	82.1
Wine qual -ity-red	60.7	60.6	61.8	61.0	61.1	61.1*	**62.3**	58.5	61.2	60.4
Texture	86.3	85.8	87.5	89.5	86.2	87.9	**91.4**	91.2*	86.3	88.1
Average accuracy	77.1	76.6	77.6	**78.9***	76.9	78.3	78.4	78.4	76.7	78.2

Table 4.6: Accuracy of ***AdaBoost.M1*** classifier with no corrected, with majority vote corrected and with four margin-based corrected training sets, using both an adaptive and a fixed amount of filtering in the presence of random noise.

Data	No filter	Majority filter	Margin-based noise correction							
			Max-margin		Unsupervised max-margin		Sum-margin		Unsupervised sum-margin	
			Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.
Letter	74.6	63.2	73.4	76.4	71.3	75.7	78.5	**80.4***	72.7	76.0
Optdigit	77.9	88.7	90.4	89.1	89.2	89.2	**91.1**	90.4*	89.7	88.0
Pendigit	79.8	89.8	93.5	92.6	92.1	91.8	**94.3**	93.7*	91.4	92.1
Vehicle	59.0	66.5	67.5	64.0	67.0	63.5	**68.0**	64.5*	65.5	61.5
Segment	81.5	89.1	90.8	92.3	89.3	91.1	91.4	**93.0***	89.4	92.3
Abalone	45.1	53.5	53.0	51.4	52.3	50.5	**54.4**	51.9*	52.3	50.6
Glass	63.8	**76.3**	**76.3**	71.3*	**76.3**	68.8	**76.3**	68.8	**76.3**	65.0
Waveform	62.8	75.4	76.6	76.4*	75.3	75.5	**76.9**	75.8	75.3	74.9
Wine qual -ity-red	49.7	**61.2**	57.5	58.5	58.2	58.7*	57.2	56.5	57.7	54.3
Texture	80.2	87.3	90.7	90.0	90.0	88.7	**93.2**	91.7*	89.9	87.5
Average accuracy	67.4	75.1	76.9	76.2	76.1	75.3	**78.1**	76.7*	76.0	74.2

Table 4.7: Accuracy of $\mathbf{1-NN}$ classifier with no corrected, with majority vote corrected and with four margin-based corrected training sets, using both an adaptive and a fixed amount of filtering in the presence of random noise.

4.9.2.2.2 Per-class classification accuracy

Tables 4.8 and 4.9 compare respectively the classification accuracy achieved by *AdaBoost.M1* and *1-NN*, on test set, for the most difficult class with no corrected, with majority vote corrected and with four margin-based corrected training sets, using both an adaptive and a fixed amount of filtering. These tables show that the margin-based mislabeled data correction scheme significantly increases the accuracy on the most difficult class for almost all the data sets with respect to no correction classification as well as the majority vote correction. Moreover, tables 4.6, 4.7, 4.8 and 4.9 confirm again the effectiveness of supervised margins in data cleaning. In addition, although the positive performance of the new margin based mislabeled instances correction is not very obvious in *1-NN* classification, tables 4.6 and 4.8 demonstrate the success of using our new margin in the design of noise filter in increasing the classification accuracy and per-class accuracy for ensemble classifiers.

Data	No filter	Majority filter	Margin-based noise correction							
			Max-margin		Unsupervised max-margin		Sum-margin		Unsupervised sum-margin	
			Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.
Letter	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Optdigit	77.0	78.0	80.5	80.8*	79.4	78.9	**81.3**	78.8	75.6	77.5
Pendigit	75.4	69.5	76.9	82.2	72.4	78.7	82.4	**83.2***	74.8	78.4
Vehicle	47.3	44.3	42.3	43.6	43.6	46.0	44.1	43.6	42.7	**47.8***
Segment	77.8	77.9	86.8	86.1	86.3	86.7	85.2	85.2	**88.0**	87.3*
Abalone	26.4	23.5	33.3	39.8	33.8	29.9	31.6	**41.1***	34.0	39.8
Glass	5.0	0.0	0.0	**50.0**	0.0	**50.0**	0.0	**50.0**	0.0	**50.0**
Waveform	78.2	74.3	79.5	80.8	77.2	81.2	80.2	79.8	77.0	**81.5***
Wine quality-red	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Texture	63.1	51.8	73.9	68.1	68.6	69.0	**83.5**	78.7*	68.2	65.7
Average accuracy	45.0	41.9	47.3	53.1	46.1	52.0	48.8	**54.1***	46.0	52.8

Table 4.8: Classification accuracy of ***AdaBoost.M1*** classifier for the most difficult class with no corrected, with majority vote corrected and with four margin-based corrected training sets, using both an adaptive and a fixed amount of filtering in the presence of random noise.

Data	No filter	Majority filter	Margin-based noise correction							
			Max-margin		Unsupervised max-margin		Sum-margin		Unsupervised sum-margin	
			Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.
Letter	**66.1**	18.4	23.4	61.0	26.9	59.3	50.9	63.6*	32.2	63.6*
Optdigit	58.9	67.0	69.6	69.6	**73.2**	69.6	70.5	72.3*	70.5	69.6
Pendigit	70.8	65.3	81.9	79.5	78.2	77.6	**84.2**	82.3*	76.3	77.7
Vehicle	30.9	**45.5**	**45.5**	36.4	43.6	32.7	43.6	38.2*	34.6	29.1
Segment	70.3	76.3	77.2	81.8	72.8	75.4	78.1	**85.1***	73.7	81.8
Abalone	39.0	20.1	31.8	32.6	30.5	30.1	**37.5**	36.4*	30.1	28.4
Glass	**50.0**	0.0	0.0	33.3*	0.0	33.3*	0.0	33.3*	0.0	0.0
Waveform	61.1	69.9	70.5	**73.9***	69.9	71.6	70.8	73.0	70.0	71.1
Wine quality-red	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Texture	72.5	58.1	75.3	67.2	73.6	61.5	**83.3**	81.3*	73.0	58.1
Average accuracy	52.0	42.1	47.5	53.5	46.9	51.1	51.9	**56.6***	46.0	47.9

Table 4.9: Classification accuracy of $\mathbf{1-NN}$ classifier for the most difficult class with no corrected, with majority vote corrected and with four margin-based corrected training sets, using both an adaptive and a fixed amount of filtering in the presence of random noise.

4.9.3 Class noise filtering using confusion matrix based noise

The confusion matrix based noise affects more class decision boundary instances than in the case of random noise. Therefore, it is more difficult to be identified and addressed than random noise. In this section, we test the performances of our margin based noise filter in the presence of this second type of class noise. The considered level of noise is 20% as in the former experiments involving random noise.

4.9.3.1 Noise removal assessment

4.9.3.1.1 Overall classification accuracy

Tables 4.10 and 4.11 display respectively the overall accuracy of *AdaBoost.M1* and *1-NN* without noise filtering, and by noise filtering for both majority vote and margin-based methods in the case of confusion matrix based noise. Compared with the no filter case, the results achieved by our class noise removal method confirm once again that the classification accuracy would improve when noise is removed from the misclassified instances which are ordered in decreasing order based on their margin values. The best

increase in accuracy with respect to the unfiltered case is over **17%** on data set *Optdigits* with *1-NN* classifier. Our margin-based mislabeled data removal scheme still achieve a better performance than the majority vote filter. While the unsupervised margins outperform supervised margins with *AdaBoost.M1*, it is the opposite for *1-NN* classifier. Moreover, sum operation based margins are statistically more effective for class noise identification than max-margins as in our random noise study. Furthermore, in the process of adaptive noise removal for *AdaBoost.M1* classification, the novel introduced margin (unsupervised sum-margin) clearly outperforms the three other margins. Finally, while the fixed scheme turns out slightly more effective than the adaptive one for confusion matrix based noise removal in *AdaBoost.M1*, they achieve similar performances in *1-NN* classification.

Data	No filter	Majority filter	Margin-based noise removal							
			Max-margin		Unsupervised max-margin		Sum-margin		Unsupervised sum-margin	
			Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.
Letter	45.7	47.2	45.7	49.8	48.8	48.5	51.1	**52.9***	46.0	48.2
Optdigits	88.8	92.6	**92.9**	89.0	92.8	89.4	92.7	88.0	92.6	89.5*
Pendigit	88.7	90.3	91.1	**92.5***	92.3	92.4	**92.5**	91.7	91.3	92.0
Vehicle	69.2	72.6	**74.8**	66.0	**74.8**	66.1	**74.8**	66.3	**74.8**	66.4*
Segment	93.4	92.3	92.5	91.4	93.1	91.8	92.4	90.8	**93.6**	93.0*
Abalone	54.9	54.7	54.8	54.1	55.0	55.0	55.0	53.4	55.1	**55.2***
Glass	87.0	**95.0**	89.1	82.3	89.3	88.5	88.1	78.4	88.4	91.3*
Waveform	78.5	81.1	82.2	82.1	82.5	**82.7***	82.0	82.6	**82.7**	82.5
Wine quality-red	50.7	59.6	59.4	59.6	59.7	59.9*	59.7	57.6	**60.2**	58.7
Texture	87.6	88.4	90.6	90.4	89.9	**90.7***	89.7	90.2	90.1	**90.7***
Average accuracy	74.5	77.4	77.3	75.7	**77.8**	76.5	**77.8**	75.2	77.5	76.8*

Table 4.10: Accuracy of ***AdaBoost.M1*** classifier with no filter, with majority vote filtered and with four margin-based filtered training sets, using both an adaptive and a fixed amount of filtering in the presence of confusion matrix based noise.

Data	No filter	Majority filter	Margin-based noise removal							
			Max-margin		Unsupervised max-margin		Sum-margin		Unsupervised sum-margin	
			Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.
Letter	74.1	60.1	76.2	75.2	76.3	75.5	78.1	**78.5***	76.4	76.6
Optdigits	76.5	93.6	94.0	91.8	91.3	92.0	**94.1**	92.1*	91.6	91.3
Pendigit	81.7	92.1	92.5	**93.8***	92.1	93.0	93.1	93.6	92.5	92.7
Vehicle	63.5	69.0	**70.5**	65.0*	**70.5**	64.5	**70.5**	64.0	**70.5**	64.5
Segment	77.8	**92.5**	91.5	88.9*	92.3	88.1	91.5	86.0	**92.5**	87.1
Abalone	46.9	**55.0**	52.7	54.1	52.1	53.9	52.8	53.4	52.1	54.3*
Glass	56.3	**71.3**	70.0	60.0	70.0	65.0*	70.0	58.8	70.0	63.8
Waveform	63.9	78.2	76.1	78.1	75.6	77.9	76.4	**78.3***	75.6	78.1
Wine qual -ity-red	50.7	**58.8**	56.8	56.7	56.7	56.8	57.3	56.8	57.2	57.3*
Texture	78	92.7	92.4	**93.2***	91.5	92.8	92.4	92.6	91.7	91.9
Average accuracy	66.9	76.3	77.3	75.7	76.8	75.9*	**77.6**	75.4	77.0	75.7

Table 4.11: Accuracy of $\mathbf{1-NN}$ classifier with no filter, with majority vote filtered and with four margin-based filtered training sets, using both an adaptive and a fixed amount of filtering in the presence of confusion matrix based noise.

4.9.3.1.2 Per-class classification accuracy

Tables 4.12 and 4.13 present respectively the classification accuracy of *AdaBoost.M1* and *1-NN* for the most difficult class, on test set. Those tables show that although the positive effect of margin-based noise removal scheme is hidden to some extent in the performance of average accuracy mainly because of the worst recognition for the most difficult class of the imbalanced data *Glass*, our margin-based class noise removal method is still effective in increasing the accuracy of the most difficult class for most data sets compared with the no filter case (increase in accuracy of up to **24%** on data set *Abalone* with *AdaBoost.M1* classifier) and statistically outperforms the majority vote method (increase in accuracy of up to **28%** on data set *Abalone* in the classification of *AdaBoost.M1*). Hence, the ability of our algorithm in the classification of difficult classes is demonstrated again. A similar conclusion as in section 4.9.3.1.1 can be drawn again here: sum operation based margins still lead to better results compared with the margins depending on a max operation. Additionally, supervised margins statistically exhibit better per-class performances with respect to their unsupervised versions.

Data	No filter	Majority filter	Margin-based noise removal							
			Max-margin		Unsupervised max-margin		Sum-margin		Unsupervised sum-margin	
			Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.
Letter	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Optdigits	76.7	83.8	**85.6**	76.6	83.6	81.6*	82.5	71.0	**85.6**	81.3
Pendigit	71.6	69.1	75.2	73.9	71.1	74.3	78.4	**81.2***	72.2	73.5
Vehicle	54.2	**55.1**	54.9	43.8*	54.9	43.5	54.9	42.7	54.9	43.6
Segment	**86.5**	80.6	82.0	83.8	82.6	81.3	82.2	84.0*	84.1	83.6
Abalone	20.7	17.0	33.0	42.3	27.5	40.8	39.1	**45.3***	28.3	43.7
Glass	**50.0**	**50.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	28.6*
Waveform	71.6	73.9	79.3	75.1	80.0	76.4*	78.2	76.1	**81.5**	76.3
Wine quality-red	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Texture	75.8	**80.6**	79.0	79.7*	78.7	79.4	71.3	77.6	76.9	78.9
Average accuracy	50.7	**51.0**	48.9	47.5	47.9	47.7	48.7	47.8	48.4	50.9*

Table 4.12: Classification accuracy of ***AdaBoost.M1*** classifier for the most difficult class with no filter, with majority vote filtered and with four margin-based filtered training sets, using both an adaptive and a fixed amount of filtering in the presence of confusion matrix based noise.

Data	No filter	Majority filter	Margin-based noise removal							
			Max-margin		Unsupervised max-margin		Sum-margin		Unsupervised sum-margin	
			Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.
Letter	65.5	0.0	51.9	65.2	50.3	64.7	66.8	**67.5***	47.5	65.2
Optdigits	64.3	75.9	79.5	76.8	76.8	79.5*	79.5	79.5*	**82.1**	77.7
Pendigit	78.3	73.5	76.7	78.1	76.3	79.1	79.1	**80.5***	75.8	78.6
Vehicle	43.6	**47.3**	45.5	38.2	45.5	41.8*	45.5	40.0	45.5	38.2
Segment	69.0	80.2	76.9	77.9*	**81.8**	73.5	76.9	71.7	**81.8**	71.1
Abalone	43.6	20.6	33.9	42.6	31.6	38.6	35.2	**44.3***	31.8	39.0
Glass	**28.6**	**28.6**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	14.3
Waveform	60.4	**73.4**	70.6	71.9	70.2	71.6	70.5	72.1*	69.7	71.9
Wine quality-red	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Texture	72.8	79.9	83.9	82.8*	81.0	82.2	**84.5**	80.5	81.0	80.7
Average accuracy	52.6	47.9	51.9	53.4	51.3	53.1	**53.8**	53.6	51.5	53.7*

Table 4.13: Classification accuracy of $\mathbf{1-NN}$ classifier for the most difficult class with no filter, with majority vote filtered and with four margin-based filtered training sets, using both an adaptive and a fixed amount of filtering in the presence of confusion matrix based noise.

4.9.3.2 Noise correction assessment

4.9.3.2.1 Overall classification accuracy

Tables 4.14 and 4.15 present the accuracy of *AdaBoost.M1* and *1-NN* without noise correction, and with noise correction for both majority vote and margin-based methods in the case of confusion matrix based noise, respectively. The comparison of our margin based class noise correction and no filter performances demonstrates the feasibility and effectiveness of adopting the ensemble margin for the correction of mislabeled instances that lie likely on class decision boundaries (confusion matrix based artificial noise). Our method significantly outperforms the majority vote noise correction scheme with *AdaBoost.M1* (**8** wins over 10). However, the majority vote filter performs better than our method in *1-NN* classification (**6** wins over 10). Nevertheless, our method is still effective to distinguish the confusion matrix based noise and significantly increases the prediction accuracy of *1-NN* with respect to the no correction case. The fixed filtering scheme achieves a similar performance with respect to the adaptive one for both *AdaBoost.M1* and *1-NN* classifications. Moreover, two concordant conclusions with random noise based analysis (section 4.9.2) can be drawn: 1) the class noise removal scheme

outperforms its correction counterpart, for both majority vote and margin-based methods, 2) supervised margins are more effective than unsupervised margins. Additionally, while the sum margins outperform the max margins for *1-NN*, it is the opposite for *AdaBoost.M1*, which is coincident with our discussion in section 4.9.2.2.1. Furthermore, in the process of adaptive noise correction for *AdaBoost.M1* classification, the unsupervised max-margin achieves better performance than the other margins.

Data	No filter	Majority filter	Margin-based noise correction							
			Max-margin		Unsupervised max-margin		Sum-margin		Unsupervised sum-margin	
			Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.
Letter	45.7	45.7	43.1	47.8	41.7	**49.0***	40.9	46.3	44.9	46.3
Optdigits	88.8	89.5	**91.4**	89.5*	91.5	89.4	91.2	89.2	88.5	89.1
Pendigit	88.7	88.6	91.0	**92.1***	89.5	91.7	91.6	91.4	89.1	91.5
Vehicle	69.2	**73.3**	72.5	66.4	72.5	68.3*	72.5	66.4	72.5	66.0
Segment	**93.4**	92.2	92.5	92.8*	92.2	92.6	91.6	91.0	92.8	92.4
Abalone	54.9	54.8	**55.2**	54.7	54.9	54.9*	**55.2**	53.9	55.0	54.9*
Glass	87.0	**95.0**	89.5	82.6	89.5	85.6*	89.5	82.4	89.5	84.0
Waveform	78.5	79.6	**82.4**	81.6	81.5	81.7*	**82.4**	81.7*	81.5	81.4
Wine qual -ity-red	50.7	58.0	**60.2**	**60.2***	60.1	58.9	59.8	58.9	59.8	58.4
Texture	87.6	85.8	89.1	90.3	89.6	**90.7***	89.2	90.5	88.9	**90.7***
Average accuracy	74.4	76.2	**76.7**	75.8	76.3	76.3*	76.4	75.2	76.3	75.5

Table 4.14: Accuracy of ***AdaBoost.M1*** classifier with no filter, with majority vote filtered and with four margin-based corrected training sets, using both an adaptive and a fixed amount of filtering in the presence of confusion matrix based noise.

Data	No filter	Majority filter	Margin-based noise correction							
			Max-margin		Unsupervised max-margin		Sum-margin		Unsupervised sum-margin	
			Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.
Letter	74.1	73.9	69.0	74.3	69.1	74.0	69.8	**75.1***	68.6	74.0
Optdigits	76.5	**91.4**	91.1	87.7*	89.9	86.8	91.2	86.7	89.0	85.8
Pendigit	81.7	**92.6**	91.0	91.5	90.7	90.9	91.4	91.8*	90.4	90.9
Vehicle	63.5	**70.0**	69.5	65.0	69.5	64.0	69.5	67.0*	69.5	64.0
Segment	77.8	90.1	89.5	89.8	89.8	89.8	89.9	90.0*	**90.6**	89.6
Abalone	46.9	**55.3**	52.7	53.9	52.7	53.9	54.3	54.3*	52.6	53.7
Glass	56.3	**68.8**	67.5	61.3*	67.5	61.3*	67.5	61.3*	67.5	60.0
Waveform	63.9	76.1	76.0	77.7	75.7	77.6	76.6	**77.8***	75.7	77.7
Wine quality-red	50.7	**59.5**	58.5	57.2	58.5	57.5*	57.8	57.2	58.0	56.3
Texture	78.0	88.2	**90.9**	90.1*	89.4	89.7	90.2	89.3	88.8	89.4
Average accuracy	66.9	**76.6**	75.6	74.8	75.3	74.5	75.8	75.0*	75.1	74.1

Table 4.15: Accuracy of $\mathbf{1-NN}$ classifier with no filter, with majority vote filtered and with four margin-based corrected training sets, using both an adaptive and a fixed amount of filtering in the presence of confusion matrix based noise.

4.9.3.2.2 Per-class classification accuracy

Tables 4.16 and 4.17 present the classification accuracy of *AdaBoost.M1* and *1-NN* for the most difficult class on test set, respectively. A failure of our approach to recognize the most difficult class of the imbalanced data *Glass* makes the mean accuracy become unsuitable as a fair evaluation measure. When compared with the no filter case, the effectiveness of our margin-based class noise correction method is clearly demonstrated (**6** wins over 8 on *AdaBoost.M1* and **5** wins over 9 on *1-NN*). Moreover, our method still significantly outperforms the majority vote filter (**7** wins over 8 for *AdaBoost.M1* and **7** wins over 9 for *1-NN*). Additionally, with respect to no filter classification, while the majority vote filter obtains lower classification accuracy of the most difficult class on most of the datasets, the new margin effectively leads to more positive results, particularly for *AdaBoost.M1* classification (**5** wins over 8).

With respect to unsupervised margins, supervised margins achieve relatively better performances in *AdaBoost.M1*. However, both kinds of margins provide quite similar performances for *1-NN*. In addition, while max-margins slightly outperform sum-margins in *AdaBoost.M1*, sum-margins lead to better results in *1-NN*. Finally, tables 4.12, 4.13, 4.16 and 4.17 demonstrate that both our margin based mislabeled data removal and

correction algorithms are statistically more effective in increasing the prediction accuracy of the most difficult class with respect to the no filter case and the majority vote method in artificially corrupted data with confusion matrix based noise. Meanwhile, the effectiveness and the contribution of our new margin is demonstrated and emphasized once again.

Data	No filter	Majority filter	Margin-based noise correction							
			Max-margin		Unsupervised max-margin		Sum-margin		Unsupervised sum-margin	
			Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.
Letter	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Optdigits	76.7	80.8	**83.8**	83.1*	83.1	82.2	81.1	77.3	71.5	75.3
Pendigit	71.6	71.1	73.0	75.9	70.8	74.4	77.2	**78.0***	70.7	74.3
Vehicle	54.2	53.1	**57.1**	43.8	**57.1**	41.6	**57.1**	45.3*	**57.1**	40.2
Segment	**86.5**	79.7	83.8	82.6	80.7	82.6	81.0	83.5*	84.3	83.5*
Abalone	20.7	16.1	**47.4**	40.6*	39.6	35.9	33.2	40.0	43.9	34.3
Glass	**50.0**	**50.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Waveform	71.6	71.7	78.2	78.4	77.9	**79.2***	78.8	**79.2***	77.8	**79.2***
Wine quality-red	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Texture	75.8	51.8	75.5	78.2	77.4	**80.1***	74.3	79.2	75.1	78.9
Average accuracy	**50.7**	47.4	49.9	48.3*	48.7	47.6	48.3	48.3*	48.0	46.6

Table 4.16: Classification accuracy of ***AdaBoost.M1*** classifier for the most difficult class with no corrected, with majority vote corrected and with four margin-based corrected training sets, using both an adaptive and a fixed amount of filtering in the presence of confusion matrix based noise.

Data	No filter	Majority filter	Margin-based noise correction							
			Max-margin		Unsupervised max-margin		Sum-margin		Unsupervised sum-margin	
			Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.	Fixed	Adapt.
Letter	**65.5**	24.0	36.1	59.9	32.4	60.4	54.2	62.6*	36.9	59.9
Optdigits	64.3	77.7	**78.6**	76.8*	75.0	76.8*	73.2	66.1	75.0	73.2
Pendigit	78.3	74.9	80.0	80.3	77.2	**80.8***	78.6	79.5	77.2	**80.8***
Vehicle	43.6	**45.5**	40.0	40.0*	40.0	40.0*	40.0	38.2	40.0	40.0*
Segment	69.0	77.0	76.0	75.2	77.7	75.2	76.0	75.2	**77.9**	75.2
Abalone	**43.6**	20.3	31.1	34.3	30.9	33.5	33.9	36.9*	31.4	33.9
Glass	**28.6**	**28.6**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Waveform	60.4	69.6	70.8	71.6	69.6	72.2	70.9	72.4	69.6	**72.5***
Wine quality-red	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Texture	72.8	63.2	**77.6**	66.7	71.3	67.8*	77.0	67.8*	71.3	63.2
Average accuracy	**52.6**	48.1	49.0	50.5	47.4	50.7*	50.4	49.9	47.9	49.9

Table 4.17: Classification accuracy of $\mathbf{1-}NN$ classifier for the most difficult class with no corrected, with majority vote corrected and with four margin-based corrected training sets, using both an adaptive and a fixed amount of filtering in the presence of confusion matrix based noise.

4.9.4 Iterative guided versus one step training margin calculation noise filtering

To test the performances of our iterative guided noise filter based on a repetitive calculation of training margins (Algorithm 4), this section carries out a comparison of this algorithm and our one step training margin calculation noise filter (Algorithm 3) for both noise removal and correction using confusion matrix based artificial class noise. Because of the need of a validation set for the iterative method, only the adaptive filtering strategy is considered for the one step noise filter in the comparison. The margins that have achieved the best performances for *AdaBoost.M1* classification in section 4.9.3.1.1 (unsupervised sum-margin) and section 4.9.3.2.1 (unsupervised max-margin) are utilized for the design of both iterative and one step noise filters.

4.9.4.1 Comparative study on noise removal

Tables 4.18 and 4.19 respectively present the overall accuracy and prediction accuracy on the most difficult class using *AdaBoost.M1* without noise filtering, and with

noise filtering for both our iterative guided and our one step training margin calculation methods using the novel unsupervised sum-margin and confusion matrix based noise. Table 4.18 shows that although the one step filter outperforms the iterative filter, the latter still outperforms without noise filtering classification. In the process of iterative guided noise removal scheme, the removal of the instances which are identified as mislabeled data in the current training set may result in more false positive samples in the next iteration. Consequently, the iterative filter faces a higher risk of throwing out good samples than the one step filter. Table 4.19 shows that the iterative noise filter statistically outperforms the one step noise removal and the classification without any filtering. Indeed, difficult class instances have low margin values. Hence, while these samples are difficult to be accurately distinguished from noisy samples by the one step filter, these instances may be retained with a higher probability because of the conservative nature of the iterative filter.

Data	No filter	One step removal	Iterative removal
Letter	45.7	48.2	**51.7**
Optdigit	88.8	**89.5**	**89.5**
Pendigit	88.7	92.0	**92.7**
Vehicle	**69.2**	66.4	67.0
Segment	**93.4**	93.0	90.6
Abalone	54.9	**55.2**	54.4
Glass	87.0	**91.3**	82.3
Waveform	78.5	**82.5**	81.8
Wine quality-red	50.7	**58.7**	55.8
Texture	87.6	90.7	**91.0**
Average accuracy	74.4	**76.8**	75.7

Table 4.18: Classification accuracy of ***AdaBoost.M1*** classifier with no filtered, one step filtered and iteratively filtered training sets using the unsupervised sum-margin and confusion-matrix based noise.

4.9.4.2 Comparative study on noise correction

Tables 4.20 and 4.21 give the overall accuracy and prediction accuracy on the most difficult class using *AdaBoost.M1* without noise correction, and with noise correction for both iterative guided and one step training margin calculation noise filters using the unsupervised max-margin and confusion matrix based artificial noise. These tables exhibit some similar conclusions to the ones drawn in the previous comparative study

Data	No filter	One step removal	Iterative removal
Letter	0.0	0.0	**13.5**
Optdigit	76.7	**81.3**	78.9
Pendigit	71.6	73.5	**77.0**
Vehicle	**54.2**	43.6	53.8
Segment	**86.5**	83.6	84.3
Abalone	20.7	**43.7**	36.0
Glass	**50.0**	28.6	14.3
Waveform	71.6	76.3	**79.7**
Wine quality-red	0.0	0.0	0.0
Texture	75.8	78.9	**79.7**
Average accuracy	50.7	50.9	**51.7**

Table 4.19: Classification accuracy of ***AdaBoost.M1*** classifier for the most difficult class with no filtered, one step filtered and iteratively filtered training sets using the unsupervised sum-margin and confusion-matrix based noise.

on noise removal. Table 4.20 shows that the iterative filter still statistically outperforms the classification without noise filtering. Moreover, while the one step filter achieves relatively better performance than the iterative method on most of the data sets, the iterative correction obtains the best results on the two imbalanced noisy data (*Glass* and *Wine quality-red*). Table 4.21 shows that the iterative noise correction statistically leads to a faster increase in accuracy for the most difficult class compared with the one step correction as well as without correction classifications. Although the mislabeled instance correction technique significantly improves the data quality, it still has a risk of producing additional noise which really needs to be overcame.

4.10 Conclusion

Ensemble margin is acknowledged as an important factor for improving the generalization performance of classifiers in ensemble learning. This chapter presented an ensemble margin-based method to address the mislabeling problem. We have proposed a novel ensemble margin definition. This margin is an unsupervised version of the classic sum-margin and was used throughout all our experiments.

Our mislabeled training data identification algorithm exploits the ensemble margin and handles both the removal and the correction of noisy labels by both one step and iterative training margins calculation schemes. The effectiveness of our method is eval-

Data	No filter	One step correction	Iterative correction
Letter	45.7	**49.0**	46.4
Optdigit	88.8	**89.4**	88.1
Pendigit	88.7	**91.7**	91.1
Vehicle	**69.2**	68.3	66.6
Segment	**93.4**	92.6	90.9
Abalone	**54.9**	**54.9**	54.0
Glass	87.0	85.6	**87.6**
Waveform	78.5	81.7	**82.0**
Wine quality-red	50.7	58.9	**59.7**
Texture	87.6	**90.7**	90.6
Average accuracy	74.4	**76.3**	75.7

Table 4.20: Classification accuracy of ***AdaBoost.M1*** classifier with no corrected, one step corrected and iteratively corrected training sets using the unsupervised max-margin and confusion-matrix based noise..

Data	No filter	One step correction	Iterative correction
Letter	0.0	0.0	0.0
Optdigit	76.7	**82.2**	77.1
Pendigit	71.6	74.4	**74.5**
Vehicle	54.2	41.6	**61.3**
Segment	**86.5**	82.6	79.7
Abalone	20.7	**35.9**	35.7
Glass	**50.0**	0.0	29.3
Waveform	71.6	79.2	**79.7**
Wine quality-red	0.0	0.0	0.0
Texture	75.8	80.1	**81.2**
Average accuracy	50.7	47.6	**51.8**

Table 4.21: Classification accuracy of ***AdaBoost.M1*** classifier for the most difficult class with no corrected, one step corrected and iteratively corrected training sets using the unsupervised max-margin and confusion-matrix based noise.

uated via analyzing the classification performances of two noise sensitive classification models *AdaBoost.M1* and *1-NN*, which are trained from our margin based filtered data.

The results of our empirical evaluation demonstrated that, our margin based method outperforms the majority vote noise filter. Moreover, we compared the performances of 4 different ensemble margins including the novel unsupervised margin in our margin based noise filter design. The analysis results show that supervised margins generally outperform unsupervised margins, and sum-operation based margins are more effective in class noise handling than max-margins.

In our work, we also compared the performances of our noise removal and noise correction methods. Our experimental results show that our noise removal outperforms its corresponding correction. Although our noise correction achieves better classification accuracy for noise sensitive classifiers than the majority vote method due to providing a more accurate prediction label for identified mislabeled instances, it still has a risk of producing additional noise. This weakness has to be alleviated as retaining bad data hinders performance more than throwing out good data.

Finally, we tested the performance of our iterative guided training margin calculation noise filtering method. This method is demonstrated as useful to improve the classification performance of *AdaBoost.M1*. In addition, iterative data clean algorithms are conservative. Such characteristic can benefit the quality of prediction of small and difficult class instances. Hence, in our experiments, the iterative filter was effective for the classification of the most difficult class in diverse classification problems.

Chapter 5

A review on ensemble methods for the class imbalance problem

This chapter gives a review on ensemble methods for the class imbalance problem. Section 5.1 presents the characteristic of imbalanced data. Section 5.2 introduces oversampling and undersampling methods for imbalance learning. Section 5.3 presents ensemble-based class balancing methods. The limitations of each of these existing imbalance learning techniques are also described. Imbalance learning methods for remote sensing data are presented in section 5.4. Finally, a summary of existing imbalance classification methods is presented in section 5.5.

5.1 Introduction

Class distribution, i.e., the proportion of instances belonging to each class in a data-set, plays a key role in any kind of machine-learning and data-mining research. Binary imbalanced data classification problems occur when one class, usually the one that refers to the concept of interest (positive or minority class), is underrepresented in the data-set; in other words, the number of negative (majority) instances outnumbers the amount of positive class instances [73, 104, 108]. Processing minority class instances as noise can reduce classification accuracy. In addition, the degradation of classification performance is also linked to other difficulty factors related to data distribution, such as decomposition of the minority class into many rare sub-concepts [110], the effect of too strong overlapping between the classes [172] or the presence of too many minority examples inside the majority class regions [18, 200]. When these factors occur together with class imbalance, the recognition of the minority class is hindered more seriously [18]. Moreover, dealing with multi-class tasks with different misclassification costs of classes is harder than dealing with two-class ones [120, 172, 200].

Some traditional classification algorithms, such as K-Nearest Neighbors (KNN) [45],

Support Vector Machines (SVM) [44], and decision trees [159], which show good behavior in problems with balanced classes, do not necessarily achieve good performance in class imbalance problems. There are several reasons behind this behavior [133, 135], some causes are:

(1) Patterns that predict the minority class are often highly specific and thus their support is very low, hence they are prone to be discarded in favor of more general patterns that predict the majority class.

(2) Many approaches, like divide-and-conquer [189], divide the training sample into small partitions, which contain even less instances from the minority class, which makes more difficult the extraction of regularities.

(3) The use of global performance measures for guiding the learning process, such as the standard classification accuracy rate, may bias the classification results towards the majority class.

(4) Some minority class instances might be identified as noise, and therefore they could be wrongly discarded by the classifier. Conversely, some actual noisy instances can degrade the identification of the minority class, since it has only a few instances to train.

The class imbalance case has been reported to exist in a wide variety of real-world domains, such as face recognition [132], text mining [148], software defect prediction [181], and remote sensing [187]. Moreover, it is generally more important to accurately predict or identify the rarer case than the more common case, and this is reflected in the costs associated with errors in the predictions and classifications [95]. Consequently, how to classify imbalanced data effectively has emerged as one of the biggest challenges in machine learning.

Typically, there are four methods for imbalanced learning [96]: sampling methods [77], cost-sensitive methods [108, 128], kernel-based methods [108] and active learning methods [62].

- **Sampling methods** The objective of these non-heuristic methods is to provide a balanced distribution by considering the representative proportions of class examples. They are carried out before training starts. These methods will be presented in detail in section 5.2.

- **Cost-sensitive methods** These methods incorporate both data level transformations (by adding costs to instances) and algorithm level modifications (by modifying the learning process to accept costs). They generally use the cost matrix to consider the costs associated with misclassifying samples [96]. Cost-sensitive neural network [211] with threshold-moving technique was proposed to adjust the output threshold toward inexpensive classes, such that high-cost samples are unlikely to be misclassified. Three cost-sensitive boosting methods, AdaC1, AdaC2, and AdaC3 were proposed [188] and cost items were used to weight the updating

strategy in the boosting algorithm. The disadvantage of these approaches is the need to define misclassification costs, which are not usually available in the data sets [73].

- **Kernel-based methods** The principles of kernel-based learning are centered on the theories of statistical learning and Vapnik-Chervonenkis dimensions [196]. In kernel-based methods, there have been many works to apply sampling and ensemble techniques to the support vector machine (SVM) concept [30]. Different error costs [5] were suggested for different classes to bias the SVM to shift the decision boundary away from positive instances and make positive instances more densely distributed.

- **Active learning methods** Traditional active learning methods were used to solve the imbalanced training data problem. Recently, various approaches on active learning from imbalanced data sets were proposed [62]. Active learning effectively selects the instances from a random set of training data, therefore significantly reducing the computational costs when dealing with large imbalanced data sets. The major drawback of these approaches is large computation costs for large datasets [62].

5.2 Sampling methods for learning from imbalanced data

The sampling approach rebalances the class distribution by resampling the data space. This method avoids the modification of the learning algorithm by trying to decrease the effect caused by data imbalance with a preprocessing step, so it is usually more versatile than the other imbalance learning methods. Many works have been studying the suitability of data preprocessing techniques to deal with imbalanced data-sets [63, 73]. Their studies have shown that for several base classifiers, a balanced data set provides an improved overall classification performance compared to an imbalanced data set. He [96] and Galar et al. [73] give a good overview of these sampling methods, among which random undersampling [77] and random oversampling [37] are the most popular.

5.2.1 Oversampling techniques

Random oversampling tries to balance class distribution by randomly replicating minority class instances. But several authors agree that this method can increase the likelihood of occurring overfitting, since it makes exact copies of existing instances [13], [73], [108].

SMOTE (Synthetic Minority Over-sampling Technique), the most popular over-sampling method, was proposed by Chawla et al. [37]. Its main idea is to create new minority class examples by interpolating several minority class instances that lie together. SMOTE creates new instances by randomly selecting one (or more depending on the oversampling ratio) of the K-Nearest Neighbors (KNN) of a minority class instance and generating the

new instance values from a random interpolation of both (or more) instances. SMOTE can avoid the over fitting problem [13]. However, its procedure is inherently dangerous since it blindly generalizes the minority class without regard to the majority class and this strategy is particularly problematic in the case of highly skewed class distributions since, in such cases, the minority class is very sparse with respect to the majority class, thus resulting in a greater chance of class mixture [65].

Many improved oversampling algorithms attempt to retain SMOTE's advantages and reduce its shortcomings. MSMOTE (Modified SMOTE) [104] is a modified version of SMOTE. The main idea of this algorithm is to divide the instances of the minority class into three groups, safe, border and latent noise instances, by the calculation of distances among all examples. When MSMOTE generates new examples, the strategy to select the nearest neighbors is changed with respect to SMOTE and depends on the group previously assigned to the instance. For safe instances, the algorithm randomly selects a data point from the K nearest neighbors; for border instances, it only selects the nearest neighbor; finally, for latent noise instances, it does nothing. This method is effective to reduce the risk of introducing artificially mislabeled instances. Hence, it can lead to more accurate classification than SMOTE. Sáez et al. try to increase the effectiveness of SMOTE by dividing the data set into four groups: safe, borderline, rare and outliers [172]. In fact, it is another version of MSMOTE which considers a fourth group in the underlying instance categorisation: rare instances. Their results show that borderline examples are usually preprocessed. The preprocessing of outliers depends on whether the safe examples are representative enough within the core of the class: if the amount of safe examples is rather low, preprocessing outliers is usually a good alternative. Finally, the preprocessing of rare examples mainly depends on the amounts of safe examples and outliers.

5.2.2 Undersampling techniques

Random undersampling aims to balance class distribution through the random elimination of majority class examples. Its major drawback is that it can discard potentially useful data, which could be important for the induction process [13], [73], [108].

Zhang and Mani used the KNN classifier [45] to achieve undersampling [207]. Based on the characteristics of the given data distribution, four KNN undersampling methods were proposed in [207], namely, NearMiss-1, NearMiss-2, NearMiss-3, and the most distant method. Instead of using the entire set of over-represented majority training examples, a small subset of these examples is selected such that the resulting training data is less skewed. The NearMiss-1 method selects those majority examples whose average distance to the three closest minority class examples is the smallest, while the NearMiss-2 method selects the majority class examples whose average distance to the three farthest minority class examples is the smallest. NearMiss-3 selects a given number of the closest majority examples for each minority example to guarantee that every minority example is surrounded by some majority examples. Finally, the most distant method selects the majority class examples whose average distance to the three closest minority class examples is the largest. Experimental results suggest that the NearMiss-2

method can provide competitive results with respect to SMOTE and random undersampling methods for imbalanced learning. This method is effective to clean the decision surface by increasing the distance between minority class and majority class. In addition, it is useful to reduce class overlapping.

5.2.3 Oversampling versus undersampling

At first glance, the oversampling and undersampling methods appear to be functionally equivalent since they both alter the size of the original data set and can actually provide the same proportion of class balance. However, this commonality is only superficial, each method introduces its own set of problematic consequences that can potentially hinder learning [100], [141]. In the case of undersampling, the problem is relatively obvious: removing examples from the majority class may cause the classifier to miss important concepts pertaining to the majority class. In regards to oversampling, the problem is a little more opaque: the computational complexity is increased rapidly with the production of more positive samples, especially in dealing with large data such as remote sensing data. In addition, oversampling has risk of over-fitting [13]. For example, since random oversampling simply appends replicated data to the original data set, multiple instances of certain examples become *tied* leading to overfitting [13]. In particular, overfitting in oversampling occurs when classifiers produce multiple clauses in a rule for multiple copies of the same example which causes the rule to become too specific; although the training accuracy will be high in this scenario, the classification performance on the unseen testing data is generally far worse [100].

Despite some limitations, oversampling and undersampling schemes have their own strengths. For example, one of the main advantages of undersampling techniques lies in the reduction of the training time, which is especially significant in the case of highly imbalanced large data sets [66]. Oversampling can provide a balanced distribution without losing information on majority class. Furthermore, both approaches provide competitive results compared with more complex methods such as ensemble methods [73]. However, when considering whether it is preferable to "add" or "remove" instances from the training set, several authors have shown the superiority of oversampling over undersampling [13, 66].

5.3 Ensemble-based imbalanced data classification methods

5.3.1 Class imbalance ensemble learning at data level

Ensemble classifiers are known to increase the accuracy of single classifiers by combining several of them and have been successfully applied to imbalanced data-sets [64, 130, 156]. Ensemble learning methods have been shown to be more effective than data sampling techniques to enhance the classification performance of imbalanced data [115]. However, as the standard techniques for constructing ensembles are rather too

overall accuracy oriented, they still have difficulty to sufficiently recognize the minority class [18]. So, the ensemble learning algorithms have to be designed specifically to effectively handle the class imbalance problem [73]. The combination of ensemble learning with imbalanced learning techniques (such as sampling methods presented in section 5.2) to tackle the class imbalance problem has led to several proposals in the literature, with positive results [73]. Hence, aside from conventional categories such as kernel-based methods, ensemble-based methods can be classified into a new category in imbalanced domains [73]. In addition, the idea of combining multiple classifiers itself can reduce the probability of overfitting [154]. In the following, we will present two popular approaches combining the ensemble paradigm and the sampling scheme to deal with the class imbalance problem.

5.3.1.1 Oversampling combined ensembles

In oversampling combined ensembles, a new minority training set is sampled (with or without replacement) from the original minority class training instances such that $|\boldsymbol{N}_{min}| = |\boldsymbol{N}_{maj}|$, where $|\boldsymbol{N}_{min}|$ is the size of the minority class and $|\boldsymbol{N}_{maj}|$ is the size of the majority class, then ensemble learning algorithms such as adaboost, bagging or random forests can be trained from the new balanced dataset [96]. He and Garcia compared the performances of random forests and adaboost, combined with random oversampling, on binary class imbalanced data sets [96]. Their results show random forests outperforms adaboost.

For multi-class imbalance problems, except using oversampling to balance the number of samples for each class, another approach [66, 201] is decomposing the multi-class problem into several binary subproblems by one-versus-one [92] or one-versus-all approaches [164]. Wang and Yao compared the performances of adaboost.NC and adaboost combined with random oversampling with or without using classes decomposition for multi-class imbalanced data sets [201]. Adaboost.NC [201] is an improved version of adaboost algorithm. It updates the weights of training examples by considering both difference among the current classifiers and the misclassification information. Their results in the case of classes decomposition show adaboost.NC and adaboost have similar performance. One-versus-all decomposition approach does not provide any advantages for both boosting ensembles in their multi-class imbalance learning experiments. The reason seems to be the loss of global information of class distributions in the process of class decomposition. The results achieved without using classes decomposition show although adaBoost.NC outperforms adaboost, their performances are degraded as the number of imbalanced classes increases. For the data sets with more classes, despite the increased quantity of minority class examples by oversampling, the class distribution in data space is still imbalanced, which seems to be dominated by the majority class [201].

The methods consisting of first pre-processing data and then using standard ensembles on balanced data cannot absolutely avoid the shortcomings of sampling. Moreover, internal imbalance sampling based ensemble approaches should work better [131]. This technique balances the data distribution in each iteration when constructing the ensemble. It can obtain more diversity than the mere use of an sampling process before learning

a model [73]. SMOTEBoost [38] proposed by Chawla et al. improves the over-sampling method SMOTE [37] by combining it with AdaBoost.M2 [177]. They used SMOTE data preprocessing algorithm before evaluating the prediction error of the base classifier. The weights of the new instances are proportional to the total number of instances in the new data-set. Hence, their weights are always the same. Whereas original data-sets instances weights are normalized in such a way that they form another distribution with the new instances. After training a classifier, the weights of the original data-set instances are updated; then another sampling phase is applied (again, modifying the weight distribution). The basic idea is to let the base learners focus more and more on difficult yet rare class examples. In each round, the weights for minority class examples are increased. However, SMOTE has high risk of producing mislabeled instances in noisy environment, and boosting is very sensitive to class noise. Hence, how to increase its robustness should not be overlooked.

Thanathamathee et al. proposed a method combining synthetic boundary data generation and boosting procedures to handle imbalanced data sets [193]. They first eliminate imbalanced error domination effect by measuring the distance between class sets with Hausdorff distance [203], and identify all relevant class boundary data, which have minimum distance value with the instances of other classes. Then, they synthesize new boundary data using a bootstrapping re-sampling technique on original boundary instances [61]. Finally, they proceed to learning the synthesized data by a boosting neural network [93]. Their method outperforms KNN, AdaBoost.M1 and SMOTEBoost. However, the method relies mainly on boundary definition; if the boundary is not correctly detected, the results may be deteriorated.

Bagging significantly outperforms boosting over noisy and imbalanced data [116]. Moreover, bagging techniques are not only easy to develop, but also powerful when dealing with class imbalance if they are properly combined [73]. Most of related works in the literature indicate good performance of bagging extensions versus the other ensembles [106, 131]. OverBagging [200] is a method for the management of class imbalance that merges bagging and data preprocessing. It increases the cardinality of the minority class by replication of original examples (random oversampling), while the examples in the majority class can be all considered in each bag or can be resampled to increase the diversity. This method outperforms original bagging in dealing with binary imbalanced data problems [73].

SMOTEBagging has been proposed to deal with multi-class imbalance problems [200]. It is another different manner to oversample minority class instances by the usage of the SMOTE preprocessing algorithm [37]. But the way it creates each bag is significantly different. A SMOTE resampling rate (a) is set in each iteration (ranging from 10% in the first iteration to 100% in the last, always being multiple of 10) and this ratio defines the number of minority class instances ($a \cdot N_{maj}$) randomly resampled (with replacement) from the original data-set in each iteration. The rest of the minority class instances are generated by the SMOTE algorithm. The reported results show that this method can get better performance than OverBagging for both binary class and multi-class imbalance problems [73, 106].

Blaseczynski and Stefanowski proposed a Neighbourhood Balanced Bagging [18] for

binary class imbalance problems. In this method, the sampling probabilities of training examples are modified according to the class distribution in their neighbourhood. Then it consists in keeping a larger size of bootstrap samples by a probability-based oversampling. Their experiments prove that their extended bagging is significantly better than OverBagging and SMOTEBagging.

5.3.1.2 Undersampling combined ensembles

Undersampling combined ensembles are similar to oversampling combined ensembles, but instead of increasing the number of minority class instances, random sampling from majority class instances is used in data pre-processing. The results reported in [96] show that random forests outperforms adaboost when combined with under-sampling for binary class imbalance problems. In addition, undersampling outperforms random oversampling when combined with random forest for binary imbalanced data sets in [96].

Wang and Yao studied the performance of undersampling combined adaboost for two types of multi-class imbalance problems, multi-minority and multi-majority cases, and used oversampling combined adaboost and adaboost.NC as comparison [201]. Their results show that undersampling combined adaboost can obtain better recognition for minority classes but produce worse performance for majority classes than oversampling combined adaboost and adaboost.NC. In other word, if the evaluation of an imbalance classification algorithm is simply based on the recognition ability of the minority class, undersampling combined adaboost is the winner among them. Moreover, this method is sensitive to the number of minority classes. This is because undersampling explicitly empties some space for recognizing minority classes by removing examples from the majority class region. When there is only one minority class, a classifier is very likely to assign the space to this class. When there are many minority classes, they have to share the same space. Hence, the effect of undersampling is reduced [201]. In addition, the results related to undersampling combined adaboost seem to be more sensitive to multi-minority than multi-majority class.

RUSBoost (Random UnderSampling Boosting) [179] is an algorithm that combines data sampling and boosting. It realizes a random undersampling by removing examples from the majority class while SMOTEBoost creates synthetic examples for the minority class by using SMOTE. Compared to SMOTEBoost, this algorithm is less complex and time-consuming, and easier to be operated [73]. Moreover, it is reported as the best approach in [73] with less computational complexity and higher performances than many other more complex algorithms such as BalanceCascade (presented in section 5.3.1.3) in dealing with binary class imbalance problems [73]. Further, it outperforms other two best methods, SMOTEBagging and UnderBagging, in [73].

UnderBagging was first proposed by Barandela et al. [11]. In this method, the number of the majority class examples in each bootstrap sample is randomly reduced to the cardinality of the minority class. Simple versions of undersampling combined with bagging are proved to work better than more complex solutions such as EasyEnsemble and BalanceCascade [131] (presented in section 5.3.1.3) [18]. Another popular extended version of bagging is Roughly Balanced Bagging (RBBag) [98]. It results from the crit-

ics of the original UnderBagging algorithm and its variants which use exactly the same number of majority and minority class examples in each bootstrap sample. Instead of fixing a constant sample size, RBBag equalizes the sampling probability of each class. For each iteration, the size of the majority class in the bootstrap sample is set according to the minority class binomial distribution. The class distribution of the resulting bootstrap samples may be slightly imbalanced and varies over iterations. This approach is more consistent with the nature of the original bagging and better uses the information about the minority examples. Both under-sampling bagging extensions outperform SMOTEBagging and OverBagging for binary class imbalance problems in [18]. In addition, according to [18, 98], RBBag performs better than the original UnderBagging. However, the performances of the two methods were not tested for multi-class imbalance learning.

Neighbourhood Balanced Bagging has another version [18]. The difference with the presented method in the previous section is in reducing the sample size with a probability-based undersampling. The reported experiments prove that this method is competitive with RBBag for binary-class imbalance tasks and outperforms the first version that was involving an oversampling scheme.

Inverse Random Under Sampling (IRUS) was proposed by Tahir et al. for the binary class imbalance problem in which the ratio of the majority and minority class cardinalities is inversed [190]. The main idea is to severely undersample the majority class multiple times with each subset having fewer examples than the minority class. For each training set, the authors assume a decision boundary which should separate the majority class from the minority class [190]. As the number of minority class samples in each training set is greater than the number of majority class samples, the focus in machine learning is on the minority class and consequently it can invariably be successfully separated from the majority class training samples. By combining multiple classifiers, the authors construct a composite decision boundary between the majority class and the minority class. Their results show this method outperforms UnderBagging, OverBagging, SMOTE and EasyEnsemble [130]. Hence, inversing the ratio of different class is effective to strengthen the decision boundary and improve the performance of bagging (subagging [72]) in imbalance learning. However, this method performs effectively only in the case of a small ratio (<11) between majority and minority classes.

Random forest is a major ensemble method [26]. It is more effective than data sampling techniques to enhance classification performance, especially for big data imbalanced classification, and its success does not rely on the choice of base learner [115, 165]. It is the most successful version of bagging resulting from the combination of the latter with random subspaces [99]. Consequently, random forest dealing with imbalanced data should be extended from all bagging-based imbalance learning methods, and may outperform these methods especially for multi-class tasks. Balanced Random Forests (BRF) [39] adapts random forests to imbalanced data. This method is similar to UnderBagging. To learn a single tree (CART) in each iteration, it first draws a bootstrap sample from the minority class, and then draws the same number of examples from the majority class. BRF outperforms the original random forest ensemble in binary-class imbalance learning. However, Liu et al. report that it does not perform as well as the BalanceCascade

algorithm (introduced in the following section) when dealing with two-class imbalance problems [130, 131].

5.3.1.3 Hybrid combined ensembles

Random balance boost [52] follows the same philosophy as SMOTEBoost and RUSBoost. Each base classifier is trained with a data set obtained through random balance. The random balance is designed to be used in an ensemble and relies on randomness and repetition. It conserves the size of the original dataset but varies the class proportions in the training sample of each base classifier using a random ratio. This includes the case where the minority class is overrepresented and the imbalance ratio is inverted. SMOTE and random undersampling (resampling without replacement) are used to respectively increase or reduce the size of the classes to achieve the desired ratios. The combination of SMOTE and undersampling provides more diversity and leads to better performance compared with other state-of-the-art combined ensemble methods such as SMOTEBoost and RUSBoost for binary-class imbalance problem [52, 53] .

Qian et al. proposed a resampling bagging algorithm [156] which is another version of UnderOverBagging [200], a combination of UnderBagging and OverBagging. In that method, small classes are oversampled and large classes are undersampled. The resampling scale is determined by the ratio of the minimum class size and the maximum class size. For binary class imbalance problems, the bayesian classifier algorithm [56] is used as base learner. The reported experimental results show that this method is more efficient than bagging, adaboost, random forests and some popular extended versions of bagging (UnderBagging, SMOTEBagging, OverBagging) and some hybrid ensembles [131]. However, the algorithm performance is highly related to the ratio of minority class size and features number. When this ratio is less than 3, the probability of obtaining a worse performance can increase significantly. For multi-class imbalance problems, KNN, Bayes, and BP (Back Propagation) neural networks are performed as base learning algorithms separately (homogeneous ensembles) and combined together (heterogeneous ensemble). The heterogeneous ensemble has the best performances in the reported experiments

EasyEnsemble [131] was proposed by Liu and Zhou in the context of imbalanced data sampling. The main motivation of this method was to keep the high efficiency of under-sampling but reduce the risk of ignoring potentially useful information contained in majority class examples. It adopts a very simple strategy. First, it randomly generates multiple subsamples S_{maj1},S_{maj2},...,S_{majn} from the majority class sample. The size of each subsample is the same as that of the minority class sample S_{min}, i.e., $|S_{maji}|$ = $|S_{min}|$, $1 \leq i \leq n$. Then, the union of each possible pair (S_{maji}, S_{min}) is used to train an adaboost ensemble. The final ensemble is formed by combining all the base learners in all the adaboost ensembles. It can get better results than adaboost, bagging, random forest, SMOTEBoost and BRF for binary imbalance problems [130]. It seems that using an ensemble as base classifier is more effective (though less efficient) for imbalance classification than using a single classifier.

BalanceCascade [131] tries to use *guided* rather than random deletion of majority class examples. In contrast to EasyEnsemble, it works in a supervised manner. In the

ith round, a subsample $\boldsymbol{S}_{maji}$ is randomly generated from the current majority class data set $\boldsymbol{S}_{maj}$ with sample size $|\boldsymbol{S}_{maji}|$=$\boldsymbol{S}_{min}$. Then, an ensemble $\boldsymbol{H}_i$ is trained from the union of $\boldsymbol{S}_{maji}$ and $\boldsymbol{S}_{min}$ by adaboost. After that, the majority class data examples that are correctly classified by $\boldsymbol{H}_i$ are removed from $\boldsymbol{S}_{maj}$. Since BalanceCascade removes correctly classified majority class examples in each iteration, it should be more efficient on highly imbalanced data sets. The method outperforms adaboost and random forest combined with both random undersampling and oversampling schemes on binary-class imbalanced data sets. But, despite the underlying guided sampling procedure, the reported results are not better than those achieved by EasyEnsemble. Furthermore, some borderline instances of majority class face the risk of being removed.

5.3.1.4 Discussion

(1) Compared to binary classification data imbalance problems, multi-class imbalance problems increase the data complexity and negatively affect the classification performance regardless of whether the data is imbalanced or not. Hence, multi-class imbalance problems cannot be simply solved by rebalancing the number of examples among classes in the pre-processing step [200]. A hybrid sampling strategy solution, which can overcome the problems of oversampling but not by cutting down the size of majority classes through undersampling should be investigated.

(2) Oversampling combined ensembles construct larger trees (base classifiers) and undersampling combined ensembles need more base classifiers to minimize the loss of information. Hence, the trade-off between computational complexity and performance of ensemble learning algorithms should be considered, especially in dealing with the imbalance problem of big datasets such as remote sensing images.

(3) There are many other boosting-based algorithms designed to address imbalance problems at data level such as EUSBoost (Evolutionary UnderSampling Boosting) [74], cost-sensitive boosting [149, 188] and so on. However, most boosting-based methods face the threat of noise as the original boosting method [52]. In addition, most boosting-based imbalanced learning techniques only focus on two-class imbalance problems and are difficult to extend to multi-class imbalance problems. They generally rely on class decomposition to simplify the multi-class imbalance problem. However, each individual classifier is trained without full data knowledge. Consequently, class decomposition can cause classification ambiguity or uncovered data regions [109, 191].

(4) The combination of bagging with data preprocessing techniques has shown competitive results, the key issue of these methods residing in properly exploiting the diversity when each bootstrap replica is formed [73]. As the most successful version of bagging, random forest has the best performance when balancing training instances before learning. But, there is relatively less investigation of random forest when combined with internal sampling schemes.

5.3.2 Class imbalance ensemble learning at classifier level

Classifier level approaches try to adapt existing classifier learning algorithms to bias the learning toward the minority class [127]. Sometimes these methods require special knowledge of both the corresponding classifier and the application domain, comprehending why the classifier fails when the class distribution is uneven [73]. For example, Park et Ghosh introduce a method by bagging a novel kind of decision α-*Tree* for imbalanced classification problems [152]. First, a novel splitting criterion parameterized by a scalar α is shown to generalize several well-known splitting criteria such as those used in C4.5 [159]. When applied to imbalanced data, different values of α induce different splitting variables in a decision tree. Those introduced decision trees tend to be less correlated. This increased diversity in an ensemble of such trees improves imbalance classification performance across a range of minority class priors. Experimental results show that their approach has better performance than bagging C4.5 and UnderBagging C4.5 in dealing with binary imbalance problems. However, base classifier variation based approaches have a disadvantage of difficultly being carried out and improved.

Weighted Random Forest (WRF) [39] has been proposed to make random forest more suitable for learning from extremely imbalanced data which follows the idea of cost sensitive learning [128]. Since the random forests classifier tends to be biased towards the majority class, Chen et al. place a heavier penalty on misclassifying the minority class. They assign a weight to each class, with the minority class given a larger weight (i.e., higher misclassification cost). The class weights are incorporated into the random forests algorithm in two places. In the tree induction procedure, class weights are used to weight the Gini criterion [146] on finding splits. In the terminal nodes of each tree, class weights are again taken into consideration. The class prediction of each terminal node is determined by weighted majority vote; i.e., the weighted vote of a class is the weight for that class multiplied by the number of cases for that class at the terminal node. The final class prediction for random forests is then determined by aggregating the weighted vote from each individual tree, where the weights are average weights in the terminal nodes. This method has similar performance as Balanced Random Forests. However, it is computationally less efficient.

5.3.3 Exploiting the ensemble margin for imbalanced data

Fan et al. showed that there was an inherently potential risk associated with the over-sampling algorithms in terms of the large ensemble margin principle [64]. Some over-sampling methods would decrease neighbor-based ensemble margins for the majority class. For example, SMOTE [37] would not only bias towards the minority class but also might be detrimental to the majority class. So, Fan et al. proposed a new synthetic over-sampling method, based on the ensemble margin. They seek a good balance between maximizing the ensemble margins gain for the minority class and minimizing the ensemble margins loss for the majority class. This algorithm generates the synthetic instances using the SMOTE algorithm, then chooses some instances as the final training set according to the margin-based imbalanced data sampling method. The resulting

balance was better than when only maximizing the ensemble margins gain for the minority class or only minimizing the ensemble margins loss for the majority class and the computational complexity was reduced compared to SMOTE [37].

5.4 Addressing the class imbalance problem in remote sensing

5.4.1 Imbalanced data classification methods

Although class imbalance has been extensively studied for binary classification problems, few approaches deal with multi-class imbalanced data sets, as is usually the case in remote sensing applications [31]. Johnson et al. adopt SMOTE (presented in section 5.2.1) to generate artificial training samples for the minority classes to address the challenge of multi-class imbalanced remote sensing imagery classification [112]. Landsat 8 Normalized Difference Vegetation Index (NDVI) image with 30 m resolution is used in their experiment. Naive bayes [71] and decision tree are used to get land cover mapping. The reported result shows both classifiers can get higher classification accuracy with balanced training set. And Naive bayes outperforms decision tree when combined with SMOTE. It is worth mentioning that, although SMOTE increases the time complexity compared with undersampling in dealing with such big data problem, it should offer more advantages in the case of remote sensing classification with limited labeled and imbalanced training samples.

5.4.2 Ensemble-based imbalanced data classification methods

Sampling combined ensembles have been introduced for remote sensing applications and demonstrated promising results [112]. For example, SMOTE combined random forest is used to classify a multi-class imbalanced remote sensing imagery [112]. The reported result shows this combination is more accurate than training a random forest with original imbalanced training set and also outperforms using a single classifier on the same balanced data set. Moreover, this combination, to some extent, alleviates the problem of high time complexity produced by SMOTE.

Stumpf et al. proposed an iterative sampling method to deal with two-class imbalance problems on very-high resolution optical remote sensing images with multispectral 4-bands and 10 m spatial resolution [185]. To estimate the class ratio in the training sample that leads to a balance of commission and omission errors, an iterative procedure was implemented and tested for landslide mapping, where the training set was split repeatedly into subsets for training and validation. The parameter α_i was defined as the ratio of minority class and majority class in the current training set at step i, and changed systematically to reach a target value α_n. In each iteration, all the minority class $|S_{min}|$, and α_i-fold number of the majority class, $|S_{maj\ i}| = \alpha_i \cdot |S_{min}|$, which had been sampled randomly from training set, were used to train a random forest and assess the classification accuracies on the remaining validation set. The procedure started from a

balanced class distribution ($\boldsymbol{\alpha_i}$=1) and in each step $\boldsymbol{\alpha_i}$ increased by 0.1. Notice that, with the update of $\boldsymbol{\alpha}$, the imbalance ratio of validation set was decreasing. Finally, the best class sampling ratio is applied to adjust the class-balance for the entire training set. The results show that this method can improve random forest's classification performance compared to the use of the original imbalanced training data.

Stumpf et al. extended their iterative class balancing method by combining active learning with random forests and applying it on the same remote sensing images [186]. In each iteration, Balanced Random Forests is used to generate a tree ensemble with the current training set. The vote entropy for each unlabeled sample is computed by the votes of random forest. Then, the mean local vote entropy at each position of the image is obtained with a sliding window, whose size is close to the mean effective range. Afterwards, the windows maximizing the mean local vote entropy are found. The labels of samples contained within those windows are queried. All new labeled samples are added to the training set as inputs in the next step.

The ensemble margin, which has been described detailedly in chapter 2, is a fundamental concept in ensemble learning. Mellor et al. introduce new ensemble margin criteria to evaluate the performance of random forests in the context of large area land cover classification and examine the effect of imbalanced training data characteristics on classification accuracy and uncertainty [143]. They also proposed a new margin weighted confusion matrix, which used in combination with the traditional confusion matrix, provides confidence estimates associated with correctly and misclassified instances in the random forests classification model. Landsat TM satellite imagery, topographic and climate ancillary data are used to build binary (forest/non-forest) and multiclass (forest canopy cover classes) classification models. For the binary classification, the imbalance experiment involved adjusting balance as a ratio of minority class to majority class. For the multi-class imbalance experiment, two sampling methods were used to get training sets. The first one adjusted the ratio of best to worst class training samples in each random forest model. The second method involved generating imbalance in the training data samples by increasing the proportion of the worst class while simultaneously decreasing the proportion of the best class by the same amount. For every iteration of both multi-class imbalance experiments, the number of samples representing the remaining classes was kept constant. Both binary and multi-class imbalance experiments maintain the same total number of original training samples. The reported experiments show that this margin based classification performance evaluation method is successful for the investigation of imbalance problems in land cover ensemble classification.

Sun et al. proposed an Ensemble Method based on the Maximum Margin (EMMM) for binary imbalanced hyperspectral image classification [187]. This method involves bagging for ensemble construction. The authors expected that the final ensemble scheme corresponds to the largest maximum margin. To keep the same size between the minority class and the majority class, the EMMM algorithm partitions the majority class sample into different subsets with kernelized K-means clustering [50]. Then, two different sampling methods are adopted by comparing the size of each subset and the minority class size: random under-sampling is adopted for the subsets with larger sizes and SMOTE oversampling algorithm is adopted for the subsets with smaller sizes. The one-versus-one

decomposition rule is used in the multiclassification solver. The EMMM method obtains better performances than other competing imbalance learning methods namely SMOTE and under-sampling for multi-class hyperspectral images. However, the clustering approach, involved in the partition of the majority class, may be not applicable to the class overlapping problem.

5.5 Conclusion

Imbalanced data is a challenge in machine learning. The skewed distribution makes many conventional machine learning algorithms less effective, especially in predicting minority class examples. Hence, the objective of imbalance learning can be generally described as *obtaining a classifier that will provide high accuracy for the minority class without severely jeopardizing the accuracy of the majority class*. There exist many different techniques to address the class imbalance problem. The simplest techniques are random oversampling and random undersampling. Despite some limitations, they are still competitive with other rebalancing methods.

Ensemble learners are more robust than single classifiers and have been certificated more effective than sampling methods to deal with the imbalance problem. Depending on how to deal with the imbalance problem, ensemble-based imbalance learning techniques can be categorized into two major groups, data level, including *oversampling combined ensembles* and *undersampling combined ensembles*, and classifier level. According to the used ensemble method, they can also be divided into three sub-categories: boosting-based, bagging-based and random forests-based extended ensembles. Among them, boosting-based methods are the most sensitive to noisy instances. Hence, data cleaning methods might be necessary for boosting-based algorithms to handle noisy samples. Bagging-based methods are the most popular in dealing with class imbalance problems thanks to the good generalization ability, the easy operation, the robustness and the high scalability of bagging.

Multi-class imbalance classification is not as well-developed as its binary-class counterpart. Simple re-balancing towards the biggest or smallest class is not a proper approach. So, new sampling strategies should be investigated for multi-class imbalance problems. Additionally, some methods rely on a multi-class decomposition scheme in addressing multi-class imbalanced data classification. Hence, it seems worthwhile to design new fusion approaches suitable for cases with skewed distributions. This way it may be possible to compensate for the class imbalance both on decomposed class level (data level) and on final output combination level (classifier level).

There are many other open research questions related to imbalanced data learning and many avenues remain to be explored. The understanding of the relationship between data imbalance ratio and learning model complexity, and the best levels of balance ratio for a given base learning algorithm, especially for multi-class imbalance problems, will be useful to provide fundamental insights into the imbalance learning problem and critical technical tools to many practical real imbalance learning applications like in remote sensing. Another interesting future line of research would be to investigate whether it

is possible to find an optimal combination of class balancing and diversity techniques. Ensemble margin has great potential for classifier design by identifying important instances as demonstrated by some recent work that appeared in the literature. Minority class instances having small ensemble margin values, the effectiveness of combining ensemble learning with margin theory for imbalanced data is also an interesting research direction to explore.

Chapter 6

Class imbalance ensemble learning based on the margin theory

This chapter proposes a novel algorithm based on ensemble margin to deal with the class imbalance problem. In section 6.2, we carry out a feasibility study on adopting the margin concept for imbalance ensemble learning. In section 6.3, a data importance function relying on the margin is proposed. Then a novel data importance ordering based algorithm is presented in detail. The experimental results are reported in section 6.4. The conclusion of this chapter is given in section 6.5.

6.1 Introduction

The proportion of instances belonging to each class in a data-set plays an important role in machine learning. However, the real world data ofter suffer from class imbalance. In this chapter, we propose a new algorithm to handle the class imbalance problem. Several methods proposed in the literature to address the problem of class imbalance as well as their strengths and weaknesses have been presented in the previous chapter. Undersampling and oversampling are two of the most popular data preprocessing techniques dealing with imbalanced data-sets [108], [37], [73]. However ensemble classifiers have been shown to be more effective than data sampling techniques to enhance the classification performance of imbalanced data [115]. Moreover, the combination of ensemble learning with sampling methods to tackle the class imbalance problem has led to several proposals in the literature, with positive results [73].

Ensemble-based imbalance learning techniques can be divided mainly into boosting-based and bagging-based extended ensembles [73]. However, as mentioned in the previous chapter, boosting based methods are sensitive to noise. On the contrary, bagging techniques are not only robust to noise but also easy to develop. Galar et al. pointed out that bagging ensembles would be powerful when dealing with class imbalance if they

are properly combined [73], [106]. Consequently, we chose to found our new imbalance ensemble learning method on bagging.

Ensemble margin theory is a proven effective way to improve the performance of classification models [32], [64]. It can be used to detect the most important instances and thus help ensemble classifiers to avoid the negative effects of redundant and noisy samples. The margin concept has been successfully used to address the class noise problem in chapter 4. Low margin instances are more important than high margin instances for training an accurate ensemble classifier as stated in chapter 4. In this chapter, we propose a novel ensemble margin based algorithm, which handles imbalanced classification by employing more low margin examples which are more informative than high margin samples. This algorithm combines ensemble learning with undersampling, but instead of balancing classes randomly such as UnderBagging [11] and margin theory, our method pays attention to construct higher quality balanced sets for each base classifier. In order to demonstrate the effectiveness of the proposed method in handling class imbalanced data, UnderBagging [11] and SMOTEBagging [200], which have been presented detailedly in the previous chapter, are used in a comparative analysis. As in our class noise handling work, we also compare the performances of different ensemble margin definitions, including the new margin proposed in chapter 4, in class imbalance learning.

6.2 Ensemble margin for imbalance learning

The main purpose of this section is to carry out a feasibility study on exploiting the ensemble margin concept for imbalanced classification as was done in section 4.4 of chapter 4 to explore the margin for class noise filtering. In this section, we first analyze the effect of class imbalance on the margin distribution of training data. Then the relationships of different kind of instances with their corresponding margin values are explored.

6.2.1 Effect of class imbalance on ensemble margin distribution

Class	Balanced data	Imbalanced data
Class 1	218	218
Class 2	212	50
Class 3	217	217
Class 4	199	199
Total samples	846	684

Table 6.1: Imbalanced and balanced versions of data set *Vehicle*.

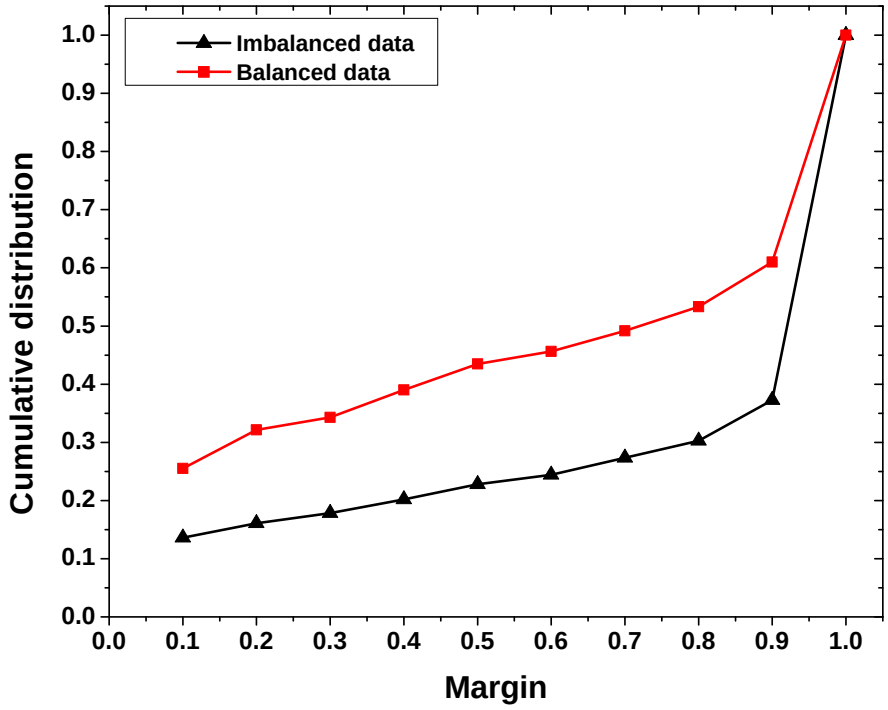


Figure 6.1: Margin distribution of correctly classified training instances by bagging with both balanced and imbalanced versions of data set *Vehicle* using a new ensemble margin.

We have previously stated that the margin distribution of training instances effectively reflects the performance of an ensemble algorithm. In this section, we analyze the effect of class imbalance on the margin distribution of the training set, as what we did in our class noise handling work (section 4.4.1 of chapter 4). During the process of classifying a balanced multi-class data, each class has the same number of instances. However, class imbalance makes the learning task more complex. Figure 6.1 shows the margin distribution of correctly classified training instances by bagging involving decision tree as base learner on data set *Vehicle* (table 6.1) in both balanced and imbalanced cases, using our new ensemble margin (equation 4.1). The margin values should be as high as possible for correctly classified instances. From the margin plot, we can see that imbalanced data lead to more instances obtaining high margin values and less instances with low margin values. In fact, the existing of one or more minority classes in a classification task results in majority classes obtaining more space. Thus this makes a classifier bias to the classification of majority classes and causes an illusory optimized margin distribution for imbalance learning.

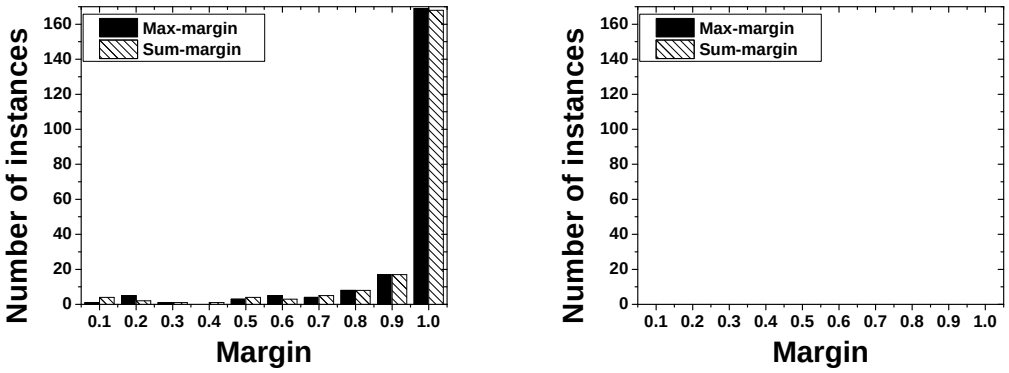
6.2.2 Max-margin versus sum-margin

The difference between max-margin (equation 2.2) and sum-margin (equation 2.3) has been indicated when handling the mislabeling problem in section 4.4.2 of chapter 4. In this section, we analyze the relevance of the two major margin definitions for imbalance learning design. According to the definitions of the max-margin and the sum-margin, the margin values of easily classified instances are robust to the margin calculation method, i.e. those data obtain high margin values no matter what margin definition is used. The difference between the two margin definitions should appear in the margin values of class decision boundary instances and minority class instances which are difficultly predicted by a classification model. In chapter 2 (section 2.4.2.1), we have mentioned that when dealing with multi-class problems, the sum-margin can represent a lower bound, since it can assume negative margin values even when the correct label gets the most of votes (when there is a plurality, but not a majority) [113] unlike the max-margin for which all the correctly classified instances get positive margin values. Hence, for the difficult instances, while their margins are near zero with a max operation, their margin values can be more or less far from zero with a sum operation.

In order to illustrate the differences between the max-margin and the sum-margin in imbalance learning, figures 6.2 and 6.3 respectively exhibit the margin distribution histograms of correctly and wrongly classified training instances for each class of the imbalanced data set *Vehicle*. The smallest class (class 2) is misclassified completely, and class 4 is the easiest class to predict. The histograms of observable training margin distributions clearly indicate that the differences between the max-margin and sum-margin are mainly reflected in the classification of the minority class (class 2) and a small proportion of majority class instances. Sum-margin leads to significantly more misclassified instances with high margins (in absolute value), especially for the minority class. In a good margin distribution, the margins should be as high as possible for correctly classified instances but as low as possible for misclassified instances. Hence, the max-margin exhibits a better margin distribution than sum-margin for data set *Vehicle*.

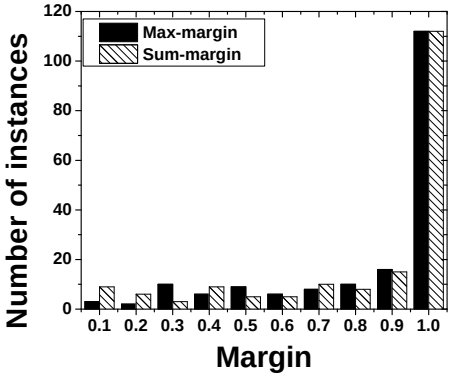
6.2.3 Supervised versus unsupervised margin

The main objective of this section is to compare the relevance of supervised and unsupervised ensemble margins for imbalance learning, as done in our class noise handling work in section 4.4.3 of chapter 4. When an instance is correctly classified by an ensemble, the votes number of the most voted class equals to the number of votes for the true class of this instance. According to the definitions of the supervised and unsupervised sum-margins (equation 2.3 and 4.1), there is no difference between both margin definitions on the margin values of correctly predicted instances. In imbalance learning, minority class instances are easily misclassified. For a misclassified sample, the votes number of the most voted class is greater than the number of votes for the true class, according to equations 2.3 and 4.1, the supervised sum-margin could lead to bigger margin (in absolute value) for the instance. Figures 6.4 and 6.5 perform a comparison of

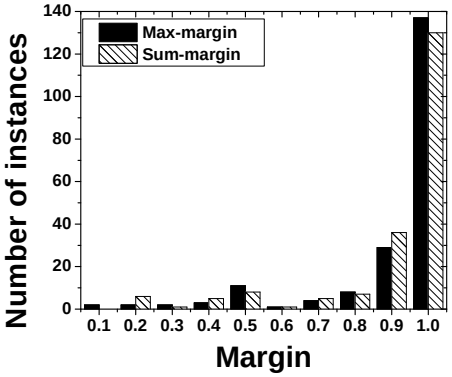


(a) Class 1

(b) Class 2 (minority class)



(c) Class 3



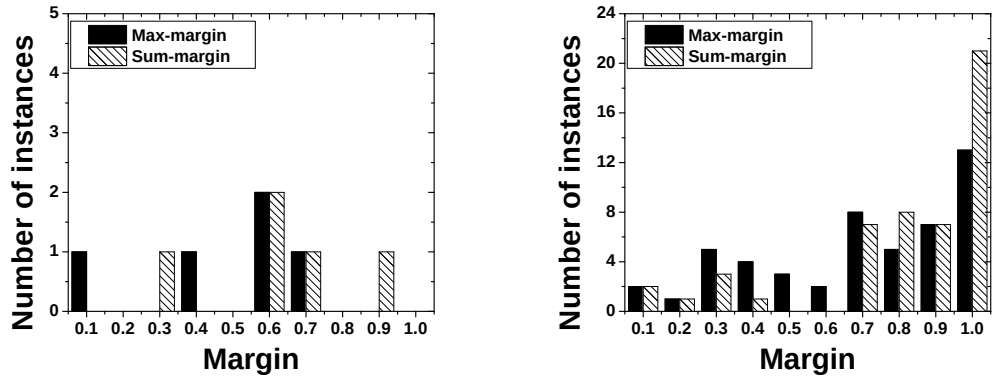
(d) Class 4

Figure 6.2: Max-margin and sum-margin distributions of correctly classified training instances using bagging with imbalanced data *Vehicle*

the sum-based margin and its unsupervised version that we have proposed (equation 4.1) respectively on the correctly predicted and misclassified instances of the imbalanced data set *Vehicle*. The supervised margin tends to make more misclassified instances obtain high margin (in absolute value) compared to the unsupervised margin especially for the minority class. Hence, the unsupervised margin might be more suitable to distinguish the important samples (lower margins) from other instances (higher margins) in margin based imbalance classifier design.

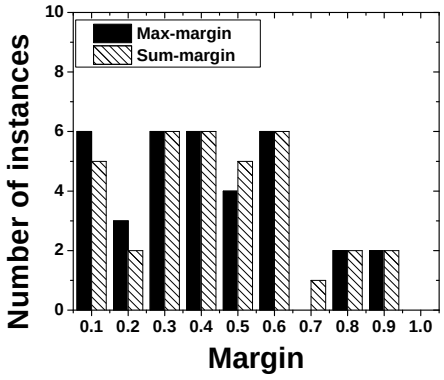
6.3 Ensemble margin based imbalanced data classification

Enhancing the classification of class decision boundary instances is useful to improve the classification accuracy. Hence, for a balanced classification, focusing on the usage of

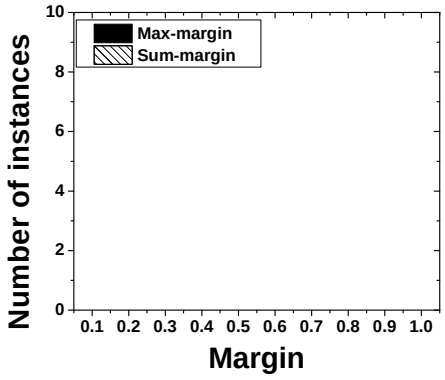


(a) Class 1

(b) Class 2 (minority class)



(c) Class 3



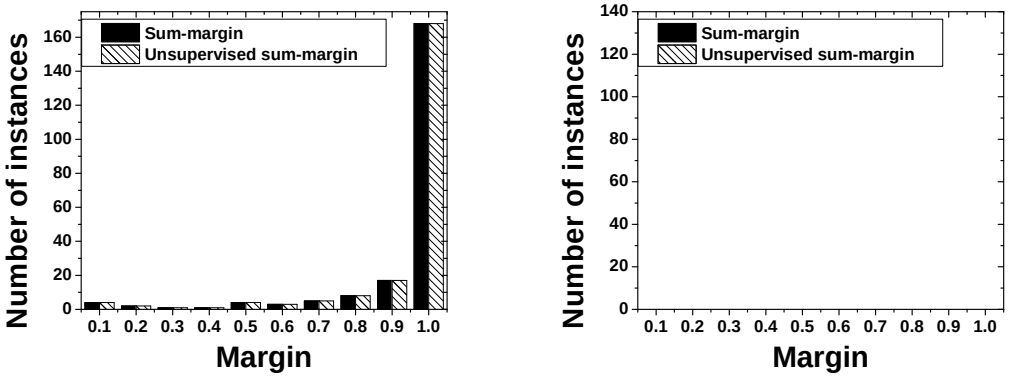
(d) Class 4

Figure 6.3: Max-margin and sum-margin distributions of wrongly classified training instances using bagging with imbalanced data *Vehicle*

the small margin instances of a global margin ordering should benefit the performance of an ensemble classifier. However, the same scheme is not suited to improve the model built from an imbalanced training set. Although most of the minority class instances have low margin values, selecting useful instances from a global margin sorting still has a risk to lose partial minority class samples, even causes the classification performance to deteriorate. Hence, the most appropriate method for the improvement of imbalanced classification is to choose useful instances from each class independently.

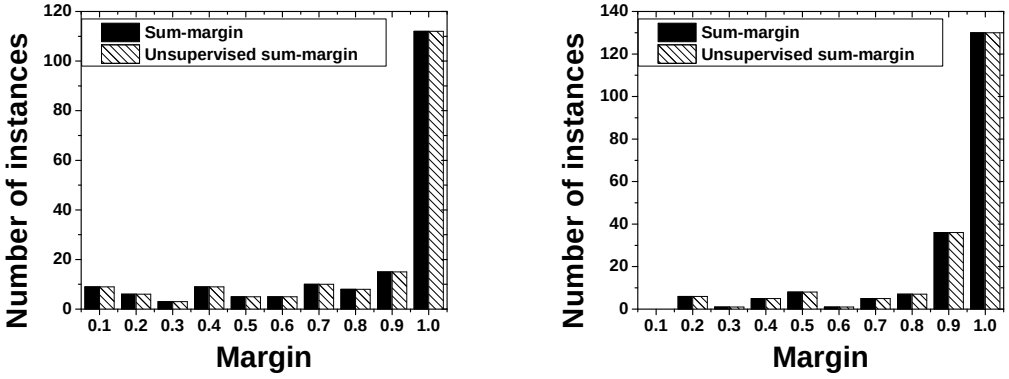
6.3.1 Ensemble margin based data ordering

In chapter 4, we have used a data ordering method based on a class noise evaluation function, which relies on an ensemble margin's definition, to identify noise. Two characteristics of the data ordering are 1) confirming the importance of small margin instances 2) focusing on identifying the high margin instances among the misclassified



(a) Class 1

(b) Class 2 (minority class)



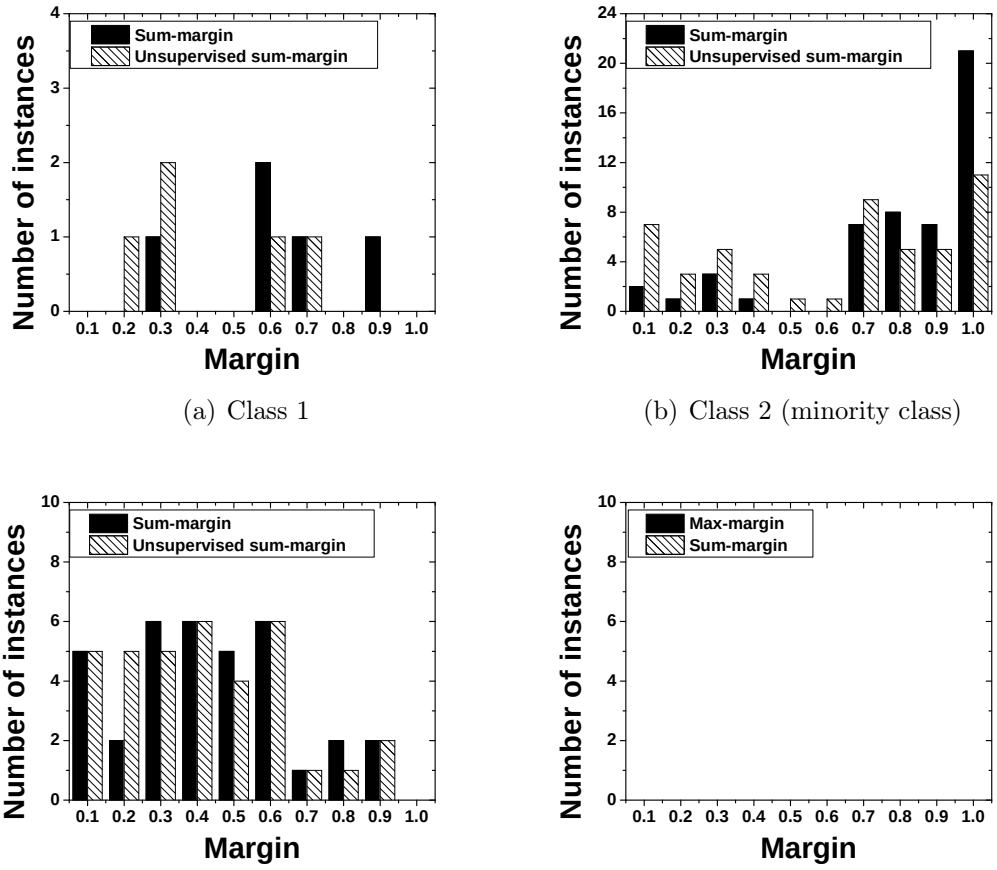
(c) Class 3

(d) Class 4

Figure 6.4: Supervised and unsupervised margin distributions of correctly classified training data using bagging with imbalanced data *Vehicle*

instances which are considered as potential class noise. The informative instances such as class decision boundary samples and difficult class instances play an important role in classification particularly when it is imbalanced classification. These instances generally have low ensemble margins, that is coincident to the first characteristic of the noise evaluation function. To utilize the relationship between the importance of instances and their margins effectively in imbalance learning, we designed our class imbalance sampling algorithm based on margin ordering. However, unlike the data ordering procedure used for noise identification, the margin ordering for imbalance learning pays more attention to low margin instances.

Let us consider a training set denoted as $S = \{(x_1, y_1), \ldots, (x_n, y_m)\}$, where x_i is a vector with feature values and y_i is the value of the class label. The importance of a training instance x_i could be assessed by an importance evaluation function which relies on an ensemble margin's definition and is defined by equation 6.1. *The lower the margin value (in absolute value), the more informative the instance* x_i *is and the more important it is*



(a) Class 1

(b) Class 2 (minority class)

(c) Class 3

(d) Class 4

Figure 6.5: Supervised and unsupervised margin distributions of wrongly classified training data using bagging with imbalanced data *Vehicle*

considered for our imbalance sampling scheme.

$$W(x_i) = 1 - |margin(x_i)| \tag{6.1}$$

To solve the problem previously mentioned related to the margins (both supervised and unsupervised) based on a sum operation, a shift is performed before data importance calculation. The shifted margin values are achieved by subtracting the minimum margin value of the samples of the training set which are correctly classified from their original margin values. An example is used to explain the margin shift procedure in figure 6.6.

6.3.2 A novel bagging method based on ensemble margin

We propose a novel ensemble margin based imbalance learning method to the quest for a classifier that is more robust when dealing with imbalanced datasets. This method is

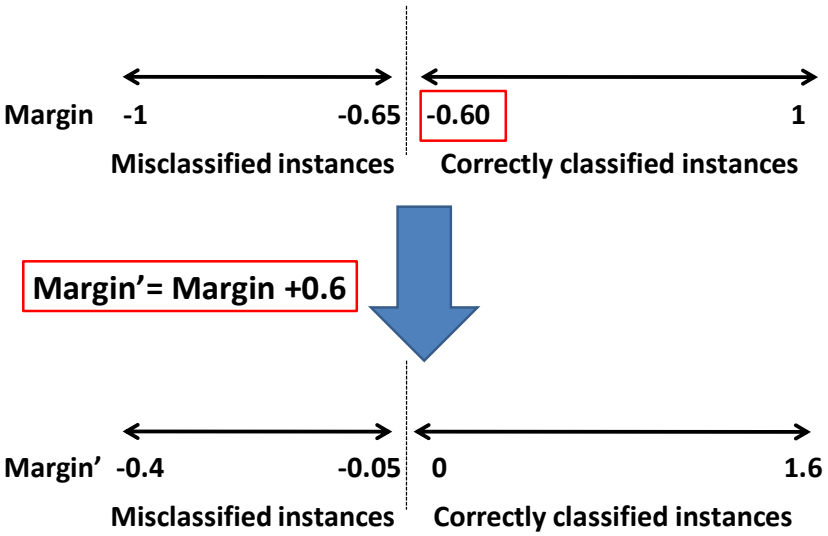


Figure 6.6: Shift procedure for sum operation based margin.

inspired by SMOTEBagging [200] a major oversampling method which has been defined in the previous chapter. It combines under sampling, ensemble and margin concepts. It could overcome the shortcomings of both *SMOTEBagging* [200] and *UnderBagging* [11]. This method has lower computational complexity than *SMOTEBagging* and focuses more on important instances for classification tasks than *UnderBagging*. Thus, our method is a boosting-like strategy which pays more attention on low margin instances.

The proposed method has three main steps:

1. Computing the ensemble margin values of the training samples via an ensemble classifier.

2. Constructing balanced training subsets by focusing more on small margin instances.

3. Training base classifiers on balanced training subsets and constructing a new ensemble with a better capability for imbalance learning.

Denote $S = \{X, Y\} = \{x_i, y_i\}_{i=1}^{n}$ as training samples. The first step of our method involves a robust ensemble classifier: *bagging* which is constructed using the whole training set. The margin value of each training instance is then calculated. In the second phase, we aim to select the most significant training samples for classification to form several new balanced training subsets. Suppose L is the number of classes and N_i the number of training instances of the ith class. We sort those classes in descending order according

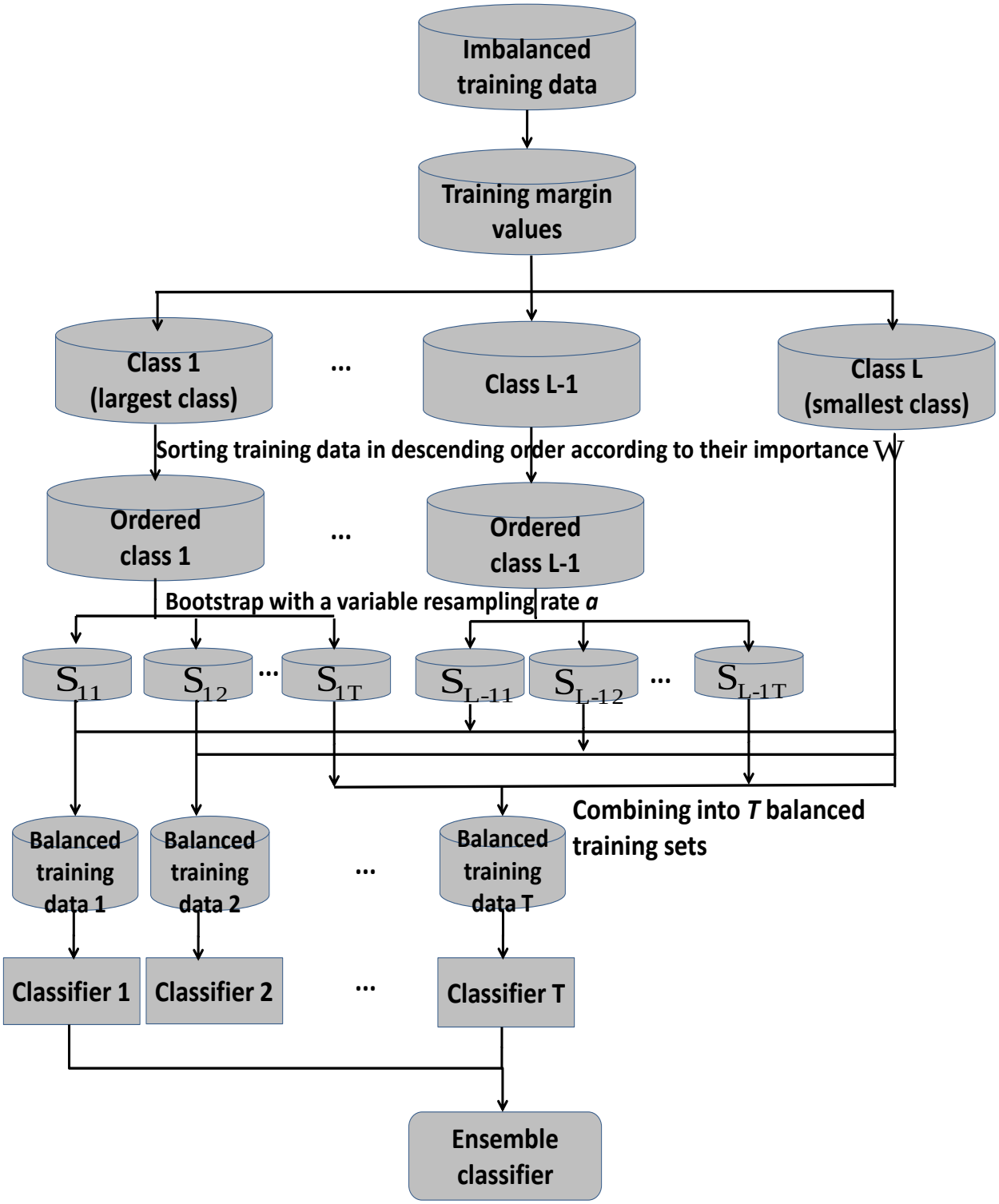


Figure 6.7: Flowchart of margin based imbalanced ensemble classification (ensemble size T, range of resampling rate a $10\% - 100\%$).

to their number of instances. Therefore, N_L is the training size of class L, which is the smallest, and N_1 is the training size of class 1 which is the largest. The training instances of each class, $1 \leqslant c \leqslant L$, are sorted in descending order according to the margin based importance evaluation function (equation 6.1) previously introduced. For each class c, the higher the importance value $W(x_i)$ of an instance $x_i \in c$, the more important this instance is for classification decision. Then, as in *SMOTEBagging* [200], a resampling rate a is used to control the amount of instances which should be chosen in each class to contract a balanced data set. All the instances of the smallest class are kept.

The range of a is set from 10 to 100 first, always being multiple of 10. For each class $c \neq L$, L representing the smallest class, N_L instances are bootstrapped from the first $N_1 \cdot a\%$ of the importance ordered samples of class c to construct subset S_{c1}. All the subsets are balanced. When the amount of class c $(2 \leqslant c \leqslant L-1)$ is under $N_1 \cdot a\%$, N_L instances are bootstrapped from the first N_c samples of class c, which is the same as in *UnderBagging*. Then the N_L smallest class samples are combined with S_{c1} $(c = 1,...,L-1)$ to construct the first balanced data. In the next phase, the first base classifier is built using the obtained balanced training set. Figure 6.7 presents the flowchart of our method with an ensemble size T and a range of $10\% - 100\%$ for a. The elements in the range of a could construct a geometric progression denoted as A. If we build $T = 100$ classifiers as ensemble members, every 10 classifiers will be built with different resampling rates a ranging from 10% to 100%, as in *SMOTEBagging*. However, while *SMOTEBagging* uses N_1, the training size of the largest class 1, as a standard for carrying out oversampling (SMOTE) on other relative minority classes, our method use N_L, the training size of the smallest class L, as a standard for performing an instance importance based undersampling on other relative majority classes.

6.3.3 Algorithm

Algorithm 7: A novel bagging method based on ensemble margin

Inputs:

1. Training set $S = (x_1, y_1), (x_2, y_2,), \cdots, (x_n, y_n)$;
2. Number of classes L;
3. N_i is the number of training instances of ith class $N_L \leqslant N_i \leqslant N_1$ $(L = smallest\ class, 1 = largest\ class)$;
4. Ensemble creation algorithm ζ;
5. Number of classifiers T;
6. Range of resampling rate a.

Iterative process:

1. Construct an ensemble classifier H with all the n training data $(x_i, y_i) \in S$ and compute the margin of each training instance x_i.
2. Obtain the weight $W(x_i)$ of each training instance x_i.
3. Order separately the training instances x_i of each class, according to the instance importance evaluation function $W(x_i)$, in descending order.
4. **For** $t = 1$ to T **do**
   a) Keep all the N_L instances of the smallest class L
   b) **For** $c = 1$ **to** $L - 1$
      i. **If** $N_c > a\% \cdot N_1$
      Get a subset S_{ct} of size N_L by performing a boostrap from first $N_1 \cdot a\%$ ordered samples of the training set S_c.
      ii. **else**
      Get a subset S_{ct} of size N_L by performing a boostrap from N_c samples of S_c.

      End
   c) Construct a new balanced data set S_t by combining the N_L smallest class training instances with S_{ct} $(c = 1, ..., L - 1)$.
   d) Train a classifier $h_t = \zeta(S_t)$.
   e) Change percentage a%.

   End

Output: $H(x) = sign(\sum_{t=1}^{T} h_t(x))$

6.3.4 Discussion

1. Imbalanced classification can not be simply treated as a data redundancy problem. While our imbalance learning algorithm tries its utmost to achieve the main objective of imbalanced classification, *improve a classifier's recognition on minority class instances meanwhile keep the accuracy of majority class not decreasing*, it does not need to remove any instances from training set as in training data reduction algorithms.
2. We have mentioned in previous chapter that classic undersampling based ensemble approach [179], [190] such as *UnderBagging* [11] samples instances randomly from majority classes to achieve a balance ratio. However, in imbalance learning, not only the imbalance ratio needs to be considered, but also the quality of the sampled instances. Our method focuses more on class decision boundary and difficult instances (lower margin instances) which are more informative for imbalance learning while safe samples (higher margin instances) give less contribution.

3. Most methods presented in previous chapter such as [52], [38] deal with binary imbalanced problems. Due to the difficult extension of these methods, class decomposition, such as OVO (One-vs-One) [92] or OVA (One-vs-All) [164], is the way to extend these methods to multi-class classification. However, those class decomposition based schemes are not suitable when a large number of classes is considered. The novel proposed method trains each base classifier with the most important instances selected from each class, hence, this method has better generalization ability for addressing both binary and multi-class imbalance problems.

4. The change in ensemble diversity [122] depends on many factors, such as ensemble learning algorithm, size of training data set and training data complexity. Both the size and the distribution of the training set for constructing a base classifier are different in the margin ordering based bagging ensemble with respect to the original training set. Hence, our algorithm can result in increased diversity compared with the bagging built on original imbalanced data. Furthermore, under the condition of training base classifiers with a fixed amount of the training set, the employment of low margin instances can provide more diversity compared with random sampling involved in *UnderBagging*.

5. Our algorithm selects important instances from each class according to their margin values and does not produce additional instances in the training process. Therefore, our method avoids the potential noise effect induced by new interpolated samples (SMOTE) which is difficultly addressed in *SMOTEBagging* [200].

6.4 Experimental results

6.4.1 Data sets

We applied our margin-based imbalance learning method on 10 UCI [8] data sets including 9 multi-class and 1 binary data (table 6.2). Among these imbalanced data, *Optdigit*, *Pendigit* and *Vehicle* are artificially imbalanced data with imbalanced *Vehicle* used, as in [82]. The top 5 data are image data sets and the bottom 5 are non image data sets. The 10 data sets are characterized by different sizes, class numbers and features. Furthermore they differ in class imbalance ratio.

Table 6.2 summaries the properties of the selected data-sets, including the number of classes (CL), the number of attributes (AT), the number of examples (EX) as well as the number of instances for each class ($\boldsymbol{C}_i$).

6.4.2 Experimental setup

In all our experiments, Classification and Regression Trees (CART) [27] are used as base classifiers for training all the classification models. Standard *Bagging* [21] is utilized to obtain the margin values of training instances. All the ensembles are implemented with 100 trees. Each data set has been randomly divided into two parts: training set and

Table 6.2: Imbalanced data sets

Data	EX	AT	CL	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
Covtype	8000	54	7	2985	3843	481	33	139	241	278			
Optdigit	1642	64	10	187	224	196	191	210	197	180	20	197	40
Pendigit	3239	16	10	20	426	408	379	437	397	362	20	394	396
Vehicle	684	17	4	218	50	217	199						
Wilt	4839	5	2	4578	261								
Cleveland	297	13	5	160	54	35	35	13					
Hayes-roth	160	4	3	65	64	31							
Newthyroid	215	5	3	150	35	30							
Glass	214	10	6	70	76	17	13	9	29				
Wine quality-red	1599	11	6	10	53	681	638	199	18				

test set. The range of sampling parameter a is set to [10-100]. All the reported results are mean values of a 10-time calculation.

.

6.4.3 Evaluation methods

In the framework of imbalanced data-sets, standard metrics such as overall accuracy are not the most appropriate, since they do not distinguish between the classification rates of different classes, which might lead to erroneous conclusions [66]. Therefore we adopt ***minimum accuracy per class*** (also used for the evaluation of class noise identification algorithms in chapter 4) and ***average accuracy*** as performance measures in our experiments.

- **Recall**, also called per class accuracy, is the percentage of instances correctly classified in each class. [172] strongly recommends to use the dedicated performance measure $Recall$ to evaluate classification algorithms, especially when dealing with multi class imbalance problems. Let n_{ii} and n_{ij} represent the true prediction of the ith class and the false prediction of the ith class into jth class respectively. The per class accuracy for class i can be defined as (6.2).

  $$Recall_i = \frac{n_{ii}}{\sum_{j=1}^{L} n_{ij}} \tag{6.2}$$

  where L stands for the number of classes

- **Average accuracy** is a performance metric that gives the same weight to each of the classes of the problem, independently of the number of examples it has. It can be calculated as the following equation:

$$AverageAccuracy = \frac{\sum_{i=1}^{L} Recall_i}{L} \quad (6.3)$$

6.4.4 Imbalance learning performance comparative analysis

These experiments evaluate the classification performance of the proposed ensemble margin based imbalance learning algorithm, and its comparison to original bagging as well as state of the art algorithms UnderBagging [11] and SMOTEBagging [200]. In addition, the performances of four ensemble margin definitions including the new margin (equation 4.1) in our margin based ensemble are compared as in chapter 5. The best results are marked in bold and the asterisk is utilized to highlight the margin definition with the best performance in the proposed method.

6.4.4.1 Average accuracy

Table 6.3 shows the average accuracy achieved by the proposed margin based extended bagging algorithm, bagging, UnderBagging as well as SMOTEBagging on the 10 imbalanced data sets of table 6.2. The experimental results in this table show that all the imbalance learning algorithms lead to an improved classification with respect to traditional bagging. Moreover, undersampling based ensemble classifiers such as margin based bagging and UnderBagging outperform oversampling based ensemble classifiers (SMOTEBagging). This result is consistent with the state-of-the-art work presented in the previous chapter, where we have explained that oversampling based methods have a risk to inject additional noise into the training set. The ensemble model based on margin achieves the best performance, especially in addressing the imbalance problem of many-majority and less-minority classes, that often occurs in the real world. These results put a clear emphasis on the importance of preprocessing the training set prior to building a base classifier by focusing on the examples with low margin values and not treating them uniformly. Although there are not obvious differences between the performances of the four ensemble margin definitions, sum margins, both supervised and unsupervised, perform slightly better than max margins, with a slight advantage to the supervised sum-margin. Supervised margins have very similar performances with unsupervised margins unlike in the class noise handling performance analysis conducted in chapter 4.

6.4.4.2 Minimum accuracy per class

Table 6.4 organized as the previous table, presents the results on minimum accuracy per class obtained on the 10 imbalanced data sets of table 6.2 by margin based bagging, traditional bagging, UnderBagging as well as SMOTEBagging. This table shows that our extended bagging algorithm significantly outperforms traditional bagging on the recognition of the most difficult class. With respect to UnderBagging, the win frequency of our method is **7/10** and its improvement in per class classification accuracy is up to **15%** (data set *Hayes-roth*). When compared with SMOTEBagging, the margin based

Data	Bagging	Under-Bagging	SMOTE-Bagging	Margin-based bagging			
				Max-margin	Unsupervised max-margin	Sum-margin	Unsupervised sum-margin
Covtype	32.0	67.9	65.7	67.4	67.6	67.9	**68.1***
Optdigit	69.4	87.5	80.4	89.7	**90.5***	89.6	90.0
Pendigit	62.4	88.0	76.9	90.2	90.3	**90.4***	90.0
Vehicle	71.2	72.8	73.4	76.1	76.4	76.2	**76.6***
Wilt	87.2	94.7	95.0	95.5	95.5	**95.6***	95.5
Cleveland	28.1	29.2	28.9	29.2	28.0	**29.5***	28.4
Hayes-roth	77.3	76.8	76.1	79.2	79.9	**82.9***	79.9
Newthyroid	81.7	93.6	85.6	94.0	94.0	94.2	**94.3***
Glass	91.6	92.9	91.2	**93.4***	**93.4***	93.1	93.1
Wine quality-red	27.9	33.8	**36.7**	33.3*	31.6	30.6	33.1
Mean accuracy	62.9	73.7	71.0	74.8	74.7	**75.0***	74.9

Table 6.3: Average accuracy of standard bagging, UnderBagging, SMOTEBagging and margin-based bagging with four margins.

method obtains also a win frequency of **7/10** and improves the minimum accuracy per class of up to **39%** (data set *Pendigit*). Unlike in the previous average accuracy margin analysis, max margins perform better than sum margins in our margin based method for the classification of the most difficult class. In addition, supervised margins slightly outperform unsupervised margins.

6.4.5 Influence of model parameters on classification performance

6.4.5.1 Influence of the ensemble size

In order to study the influence of ensemble size on bagging construction, we present in figure 6.8 and 6.9 the evaluation of the average accuracy and minimum accuracy per class on data sets *Wilt* (image data) and *Hayes-roth* (non image data) with respect to ensemble size throughout the bagging induction processes, i.e. from 1 up to 200 trees for all the bagging methods.

From figure 6.8, we can see that our margin based bagging shows higher average accuracies than *Bagging* as well as the other two state of the art methods *UnderBagging* and *SMOTEBagging* for both datasets. In our margin based method, unsupervised margins present relatively smoother curves with respect to supervised margin definitions

Data	Bagging	Under-Bagging	SMOTE-Bagging	Margin-based bagging			
				Max-margin	Unsupervised max-margin	Sum-margin	Unsupervised sum-margin
Covtype	0.0	41.0	**46.4**	31.4	30.8	31.8*	31.0
Optdigit	0.0	71.4	61.3	78.1	**79.6***	76.7	79.3
Pendigit	0.0	70.8	33.3	**72.8***	71.9	71.0	70.9
Vehicle	31.3	44.0	**47.0**	40.8	39.1	41.7*	39.3
Wilt	74.0	92.8	94.4	**95.4***	95.3	95.2	95.3
Cleveland	0.0	0.0	0.0	**7.4***	4.4	5.7	3.4
Hayes-roth	47.6	53.5	41.1	68.1	**69.2***	67.8	64.4
Newthyroid	61.8	**88.0**	72.4	85.0*	85.0*	84.2	84.2
Glass	**80.0**	79.8	**80.0**	**80.0***	**80.0***	79.8	79.8
Wine quality-red	0.00	15.9	0.00	15.9	**19.6***	14.2	16.9
Mean accuracy	19.5	55.7	47.6	**57.5***	**57.5***	56.8	56.4

Table 6.4: Minimum accuracy per class of standard bagging, UnderBagging, SMOTE-Bagging and margin-based bagging with four margins.

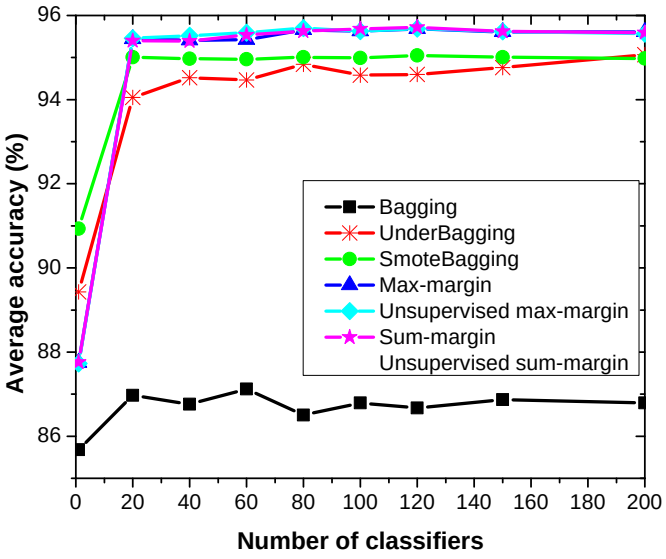
for *Hayes-roth*. Additionally, in section 2.4.2.1 of chapter 2, we have stated that, the different margin definitions under consideration for our margin-based ensemble learning framework are quite similar for two-class problems. Hence, expectedly, the four margins present very similar performances in our margin based method on the binary class data set *Wilt*.

Figure 6.9 shows that our method obtains significantly higher minimum accuracy per class with respect to *Bagging*, *UnderBagging* and *SMOTEBagging* for both datasets. Unlike the above average accuracy margin analysis, while the curves of the four margins are still very similar on data set *Wilt*, the curves associated to sum margin definitions are relatively smoother than those of max margins on data set *Hayes-roth*, and sum-margin (supervised) obtains the smoothest curve.

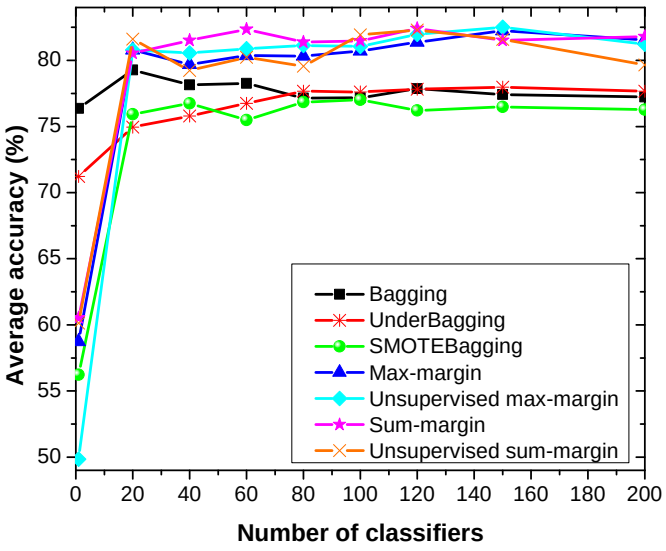
6.4.5.2 Influence of the resampling rate

This section aims to study the influence of the resampling rate $\boldsymbol{a}$ on margin-based bagging performance in imbalanced classification. We first employ the following example to illustrate our experimental design. The maximum value of the resampling rate $\boldsymbol{a}$ should be equal to or less than 100. When the size of $\boldsymbol{A}$, the associated set of $\boldsymbol{a}$ values, is set to 5, the elements of $\boldsymbol{A}$ are $\{\mathbf{20,40,60,80,100}\}$, i.e. the range of $\boldsymbol{a}$ is 20-100. When $\boldsymbol{A} = \{\mathbf{1}\}$, our margin based method becomes similar to *UnderBagging*.

In this experiment, the size $\boldsymbol{T}$ of the bagging ensemble is set to 100 and the tested number of elements in $\boldsymbol{A}$ is set from 1 to 40. Figure 6.10 exhibits the optimal range of $\boldsymbol{a}$

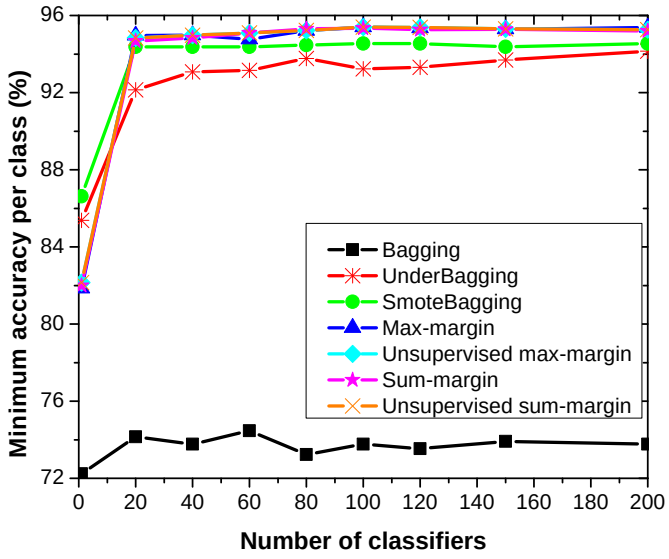


(a) Wilt

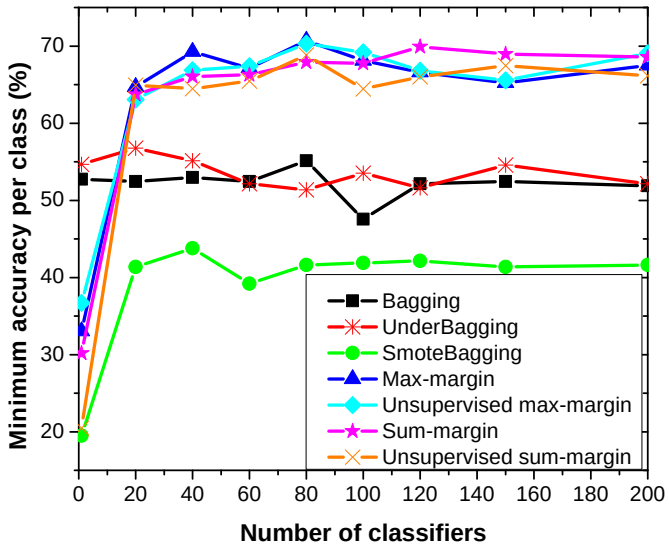


(b) Hayes-roth

Figure 6.8: Evolution of the average accuracy according to the ensemble size.



(a) Wilt



(b) Hayes-roth

Figure 6.9: Evolution of the minimum accuracy per class according to the ensemble size.

which leads to the best average accuracy for each of the four margin definitions, on all the data sets. Almost all the classification results are improved compared with those of tables 6.3 and 6.4. The best increase in average accuracy is about 1.5% for most of the non image data (bottom of table 6.3). The best increase in minimum accuracy per class is about **10%** for image datasets *Covtype* and *Vehicle*. Hence, it is interesting to further optimize our algorithm by the selection of an optimal resampling range.

Tables 6.5 and 6.6 respectively present the average accuracy and minimum accuracy per class, achieved by our margin-based bagging algorithm using respectively max-margin, unsupervised max-margin, sum-margin and unsupervised sum-margin with optimal resampling ranges, on the ten data sets. The exhibited results correspond to the classification results presented in figure 6.10. From these tables, we can see that while sum margins obtain slightly better results compared with max margins for the improvement of average accuracy, it is the opposite in minimum accuracy per class performances. Supervised and unsupervised margins achieve relatively similar performances.

	Max-margin	Unsupervised max-margin	Sum-margin	Unsupervised sum-margin
Covtype	67.8	**68.1**	67.9	**68.1**
Optdigit	90.5	**90.9**	**90.9**	**90.9**
Pendigit	**91.8**	91.5	**91.8**	91.3
Vehicle	77.1	**77.5**	77.0	**77.5**
Wilt	**96.0**	**96.0**	95.6	**96.0**
Cleveland	30.6	28.0	**31.0**	29.4
Hayes-roth	**84.3**	83.8	**84.3**	83.4
Newthyroid	95.8	95.7	**95.9**	95.1
Glass	94.8	**94.9**	94.2	94.7
Wine quality-red	34.0	34.4	34.5	**34.8**
Mean accuracy	**76.3**	76.1	**76.3**	76.1

Table 6.5: Average accuracy of margin-based bagging involving four margins with optimal resampling range.

	Max-margin	Unsupervised max-margin	Sum-margin	Unsupervised sum-margin
Covtype	40.1	39.9	39.9	**40.8**
Optdigit	80.4	**81.1**	80.4	**81.1**
Pendigit	**75.8**	73.3	72.7	73.7
Vehicle	**51.2**	48.6	50.4	47.7
Wilt	**95.5**	**95.5**	**95.5**	**95.5**
Cleveland	**11.2**	10.1	10.3	8.3
Hayes-roth	72.8	**73.1**	72.6	70.7
Newthyroid	90.9	90.9	92.0	**92.2**
Glass	**80.0**	**80.0**	**80.0**	**80.0**
Wine quality-red	17.5	20.3	**27.3**	19.9
Mean accuracy	61.5	61.3	**62.1**	61.0

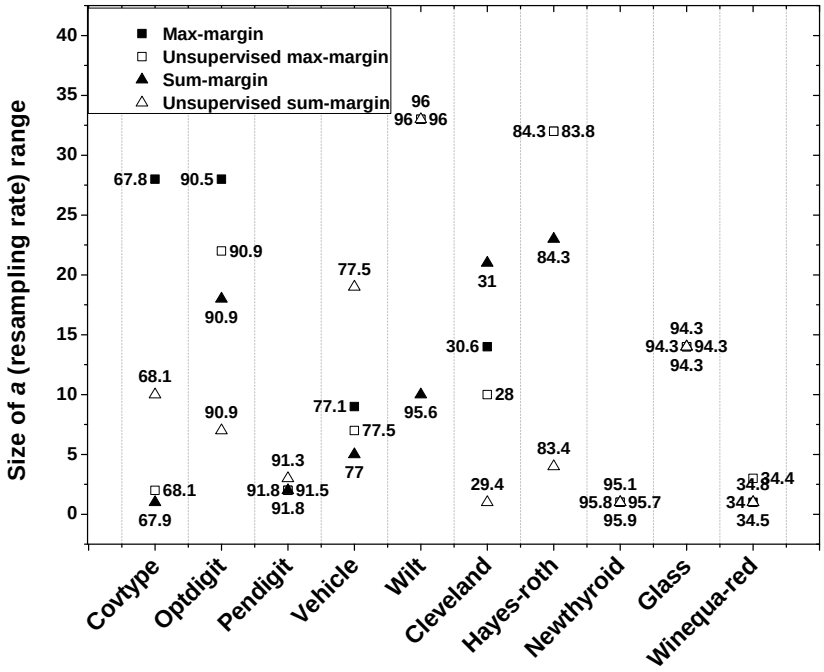
Table 6.6: Minimum accuracy per class of margin-based bagging involving four margins with optimal resampling range.

6.5 Conclusion

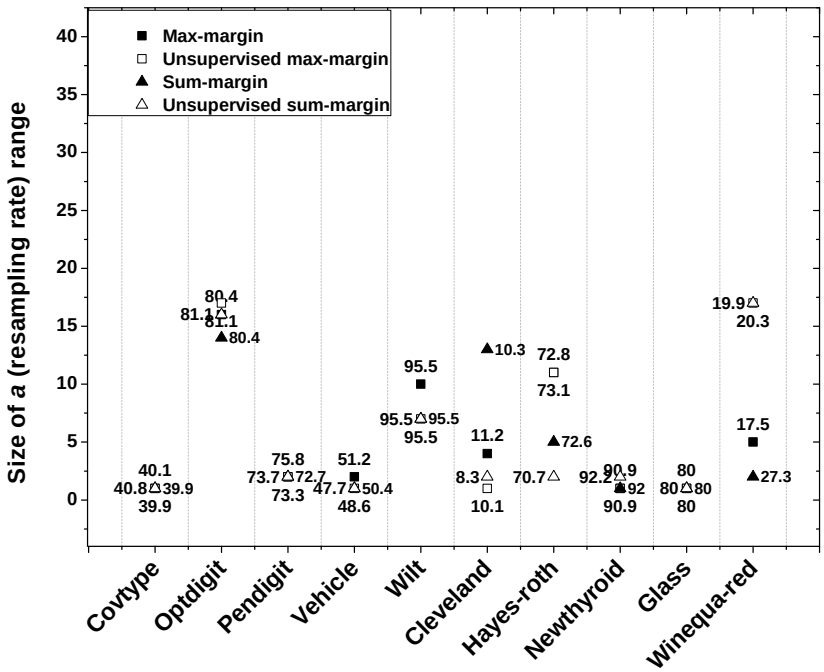
Ensembles of classifiers have shown very good properties for addressing the problem of imbalanced classification. They work in line with baseline solutions for this task such as data preprocessing for an ensemble or for each classifier of the ensemble. However, selecting more informative instances should benefit ensemble construction and better handle multi class imbalanced classification. Our answer to this data selection problem consists of carrying out an estimation of instance importance which relies on the ensemble margin. More specifically, instances can be focused on or not by an ensemble of base classifiers according to their margin values. We consider the lowest margin instances as the most informative in classification tasks.

In this work, we have proposed a novel margin ordering based bagging method based on an under sampling scheme for imbalanced classification. To evaluate the effectiveness of our approach, standard bagging as well as two state of the art imbalance learning ensemble methods UnderBagging and SMOTEBagging that inspired our method were used in comparative analysis. From this study, we have emphasized the superiority of the new proposed method, in handling the imbalance learning problem compared with bagging, UnderBagging and SMOTEBagging.

The performances of four margin definitions involved in our algorithm were also compared. While the sum-margins generally outperform max margins in terms of average accuracy, the latter have better performance in minimum accuracy per class. The supervised margins achieve similar performance with the unsupervised margins. In addition,



(a) Average accuracy



(b) Minimum accuracy per class

Figure 6.10: Optimal range of resampling rate a in margin based bagging involving four different margins for all the data sets.

the effectiveness of the new proposed margin in addressing the class imbalance problem is demonstrated.

As future research we plan to extend the margin-based ensemble framework to an oversampling scheme, such as producing minority class instances by adopting the *SMOTE* procedure on the small margin instances.

Chapter 7

Application to land cover mapping

This chapter mainly focuses on the application of previously presented ensemble learning methods to land cover mapping. The application of margin based class noise filtering involving random forests to the classification of noisy remote sensing data is presented in section 7.2. Section 7.3 applies the proposed imbalance learning ensemble algorithm based on margin concept and random forests to the classification of imbalanced remotely sensed data. Conclusions are drawn in section 7.4.

7.1 Introduction

Remotely sensed image data is widely used in a range of oceanographic, terrestrial, and atmospheric applications, such as land cover mapping, environmental modeling and monitoring, and the updating of geographical databases [140]. Supervised classification is an important task in remote sensing image analysis. A significant attention has focused on multiple classifier systems or ensemble classifiers [58]. However, the problems of class mislabeling and class imbalance often exist in remote sensing data and result in negative effect on the performance of supervised classifiers even in ensemble models.

The presence of noise in remote sensing imagery degrades the interpretation ability of the data. In particular, mislabeled training data is inevitable in remote sensing where training data sources are typically ground-based [107, 142, 143]. Several methods such as [123], [101], [124], [28] for addressing the problem of class noise on remote sensing datasets have been presented in chapter 3. However, those methods are either only effective for simple cases of class noise that can be safely managed by overfitting avoidance, or tend to treat too many clean instances as noise.

Class imbalance decreases the prediction accuracies of the minority classes, which are usually more important than the majority classes in land-cover mapping. In chapter 5, we introduced the existing algorithms proposed for the imbalanced classification of remote

sensing data [112, 185–187]. Among these methods, the sampling combined ensemble methods are the most popular [112]. However, the simple sampling combined ensemble method such as [112] just increases the training data size of minority classes by over sampling without considering the negative effect of sampling schemes such as producing additional noise. Some improved methods such as [185], [187] have been proposed for addressing binary imbalance problems. For multi-class classification, these methods can be carried out only by *One versus One* and *One versus All* schemes which have been mentioned in chapter 5 and proved not necessary for multi-class imbalance learning in [201].

In chapters 4 and 6, we respectively proposed ensemble margin based methods to handle the class noise and class imbalance training data issues. These methods have potential effectiveness to improve the classification of the remote sensing data, which is more difficultly addressed because of various reasons such as big size and high imbalance ratio. Hence, this chapter will further test the generalization abilities of the proposed algorithms on remote sensing imagery. Additionally, although the novel unsupervised sum-margin (equation 4.1) has been demonstrated effective to address both class mislabeling and class imbalance problems, this chapter will apply for remote sensing classification the margin definition *sum-margin* (equation 2.3), which led to the best performances in handling both training data issues as reported in chapters 4 and 6.

Random forests, presented detailedly in chapter 2, is a powerful ensemble technique in particular for remote sensing classification [59, 80]. It has the advantages of noise robustness and fast for classification of big data. So, in this chapter, we adopt random forests instead of bagging as a robust ensemble to design our margin based ensemble learning algorithms to solve the class mislabeling and class imbalance problems in the difficult context of remote sensing classification.

7.2 Dealing with mislabeled training data using margin based ensemble methods for land cover classification

This section handles the mislabeling problem of remote sensing data by using the proposed ensemble margin based methods including both class noise removal and correction (chapter 4). The effectiveness of the class noise handling methods is demonstrated in performing mapping of land covers. As in chapter 4, *boosting* [175] and *K-Nearest Neighbors (KNN)* [45], are used to assess the quality of the resulting filtered training sets. A comparative analysis is also conducted with respect to the majority vote filter [29]. Two cases, artificial class noise and actual class noise, are considered in the experiments. The considered artificial class noise has been presented in chapter 4. The actual class noise always exists in real world data, especially in remotely sensed data, but its amount is unknown, hence, it is more difficult to be addressed than artificial class noise whose amount is controlled.

7.2.1 Material

Four remote sensing datasets (table 7.1) were involved in the experiments.

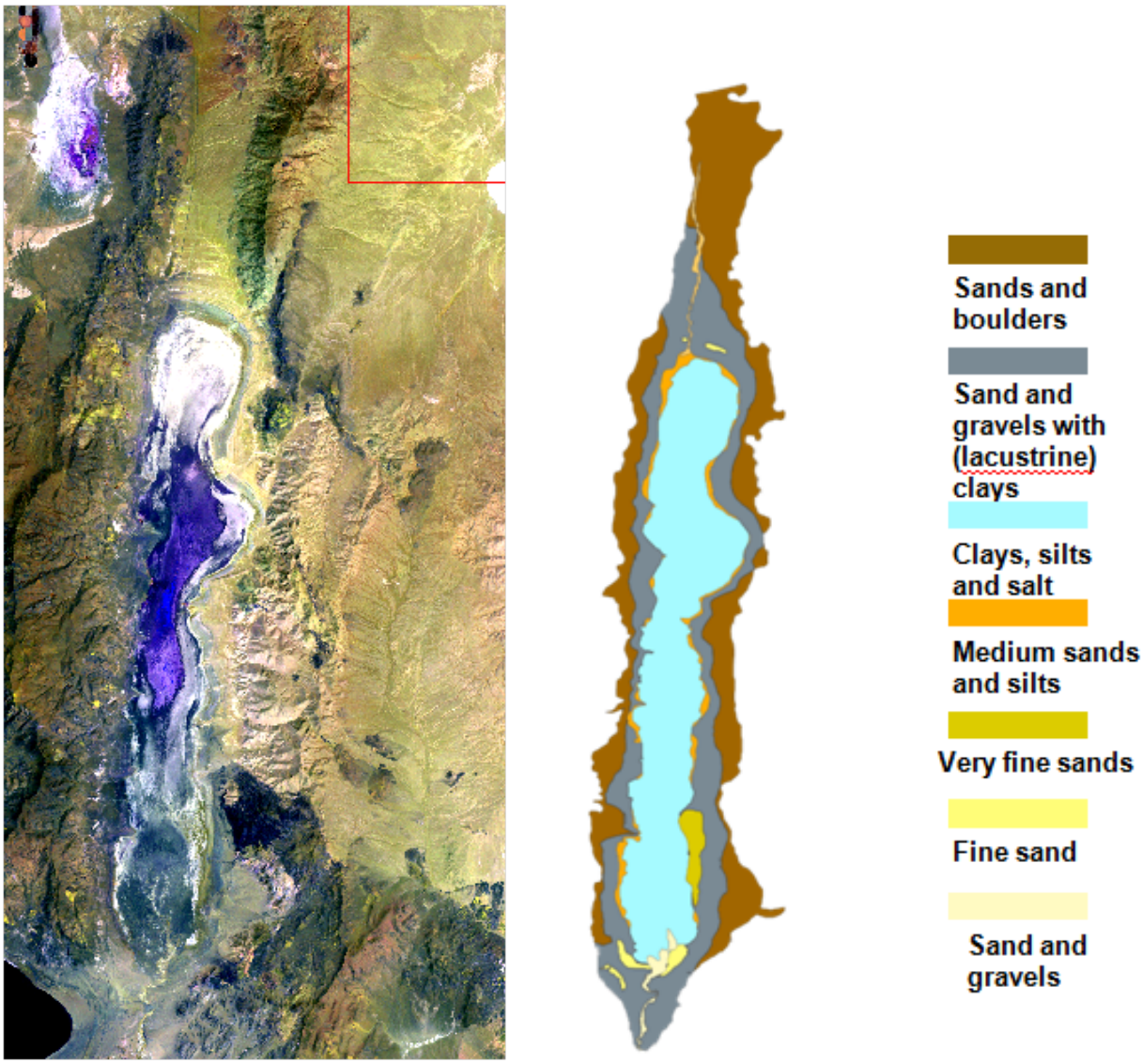
1 The first dataset is a Quickbird multispectral forest image of 2.4 m spatial resolution with 4 spectral bands: Red, Green, Blue and Near-Infra-Red. Five forest structure classes have been defined to describe different forest structure growth stages [15]. The variables of this highly imbalanced data set include five $1st$ order texture features extracted from the four spectral bands using a window size of 7×7 pixels: mean, variance, median, kurtosis, skewness.

2 The second dataset is an airborne urban image of 25cm spatial resolution. This orthoimage is composed of three spectral bands in the visible domain: Red, Green, Blue. It exhibits 4 classes to identify [19]: buildings, artificial ground, natural ground and vegetation which have been defined by photo-interpretation.

3 The third dataset is a Landsat MSS (Multispectral Scanner System) satellite multispectral data from UCI Repository [8] with 4 spectral bands (Red, Green and two Near-Infra-Red bands) and 80m spatial resolution. It represents different soils in relation to cropping practices: red soil, cotton crop, gray soil, damp gray soil, soil with vegetation stubble and very damp gray soil. This data consists of the multispectral reflectance values of pixels in 3×3 neighbourhoods in the satellite image, and the classification associated with the central pixel in each neighbourhood. This data is given in random order and certain lines of data have been removed so the original image cannot be reconstructed from this data set.

4 The fourth dataset is a highly imbalanced Landsat Satellite multispectral Lake image (figure 7.1) of size 1190×2351 pixels with 7 spectral bands (Red, Green, Blue, Near-infrared, mid-infrared, far-infrared) and a spatial resolution of 28.5 m/pixel. Seven classes have been defined by geologist expertise to describe different geological structures. The variables of this highly imbalanced data set include, as in the first dataset, 5 $1st$ order texture features extracted from the spectral bands using a window size of 3×3 pixels: mean, variance, median, kurtosis, skewness of the mid-infrared band, and means of Red, Green, Blue, Near-infrared and far-infrared bands.

All datasets were randomly selected from the original data. Each data set was divided into three parts: training set, validation set and test set. Artificial class noise was injected by randomly choosing a subset of 20% (noise rate) from original datasets, then randomly labeling the class label values of these selected examples to another label.



(a) (b)

Figure 7.1: (a) Three-band color composite of Lake Landsat Satellite image (b) Ground truth.

Dataset	Training	Validation	Test	Variables	classes
Forest	2512	1238	2512	20	5
Urban	6800	3400	6800	3	4
Statlog	2000	1000	2000	36	6
Lake	22630	11317	22630	10	7

Table 7.1: Data sets.

7.2.2 Results and discussion

In our experiments, we used *random forests* [26] to create an ensemble involving Classification and Regression Trees (CART) [27] as base classifiers. *Boosting* [69] and *KNN* [45] noise-sensitive classifiers were used to assess the quality of both class noise removal and class noise correction algorithms. *Random forests* and *boosting* ensembles were implemented with 100 pruned trees. Minimum size of terminal nodes of pruned trees was 7. $\boldsymbol{K}$ value of *KNN* was set to 1. Other parameters of *random forests*, *boosting* and *KNN* were kept to their default values in R-project packages [160]. All the reported results are mean values of 10-time calculations. According to the comparative analysis involving four different ensemble margins in chapter 4, sum-margin (equation 2.3), which turned out the most effective margin for class noise handling, is adopted in these experiments. We applied our margin-based class noise removal and correction methods on the 4 satellite image data sets (table 7.1). For a fairer comparison, the validation set was included in the training data when the majority vote method was used.

7.2.2.1 Experiments on artificial class noise

Tables 7.2 and 7.3 show the accuracies of *boosting* and *KNN* on satellite image test sets, using randomly mislabeled training data (noise rate of 20%), without class noise filtering, and by both noise removal and noise correction using respectively margin-based and majority vote filters. The margin-based result columns of these tables (as well as in all the tables to follow) are split into two sub-columns. The first one indicates that the training margins calculation is done in a single step, and the second one refers to a repetitive calculation of the training margins (iterative scheme).

Table 7.2 shows that, for the classification of *boosting*, margin-based mislabeled data removal and correction have significantly better performances than majority vote filters for the four datasets. The increase in accuracy compared with the majority vote method is over **4%** for all the remote sensing data. Our iterative scheme always outperforms the single step margins calculation alternative (up to **3%** of increase in accuracy) for both noise removal and noise correction thanks to its greater flexibility and ability to distinguish between correct and noisy samples. The increase in accuracy with respect to the unfiltered case is over **5%** for most data sets. Table 7.3 demonstrates that the margin based method is still the most effective for the classification of *KNN* compared to both

no filtering and majority vote based class noise filtering. The best increase in accuracy achieved by our method with respect to the unfiltered case and the majority vote filter is about **16%** for data set *Lake*. Moreover, our iterative scheme still outperforms the single step margins calculation alternative (up to **5%** of increase in accuracy) for both noise removal and noise correction, which is consistent with the previous class noise handling performance assessment based on boosting. Our margin-based noise correction appears overall as a better alternative than our noise removal.

Data	No filter	Removal			Correction		
		Majority	Margin		Majority	Margin	
			One step	Iterative		One step	Iterative
Forest	84.0	82.9	85.1	86.5	82.6	85.7	**87.1**
Urban	64.4	64.1	68.5	**69.8**	64.3	68.6	69.3
Statlog	83.6	84.2	86.3	**88.7**	83.2	84.8	88.4
Lake	73.9	74.1	74.8	76.7	73.1	75.1	**78.2**

Table 7.2: Accuracy of boosting classifier with no filter, majority vote filtered and margin-based filtered training sets on artificially corrupted data sets (noise rate=20%)

Data	No filter	Removal			Correction		
		Majority	Margin		Majority	Margin	
			One step	Iterative		One step	Iterative
Forest	67.2	**82.4**	74.4	80.1	81.0	75.2	80.4
Urban	54.8	65.4	68.4	70.0	64.0	68.5	**70.2**
Statlog	73.4	85.2	82.7	86.5	82.0	83.5	**88.0**
Lake	65.0	65.4	76.0	79.7	60.4	77.1	**81.2**

Table 7.3: Accuracy of *KNN* classifier with no filter, majority vote filtered and margin-based filtered training sets on artificially corrupted data sets (noise rate=20%)

Tables 7.4 compares the classification accuracy achieved by *boosting*, on test set, for the most difficult class of each dataset with noisy training data. This table shows that our method significantly increases the accuracy of *boosting* on the most difficult class for the three datasets (up to **33%** for *Forest*) while the majority vote method is less effective and even decreases the per-class accuracy of data set *Urban*. Difficult class instances have typically low margin values and hence are at low risk of being removed or incorrectly repaired, our potential mislabeled training data being the highest margin

misclassified instances. The iterative class noise removal method generally outperforms its single step counterpart in per-class accuracy (increase in accuracy of up to **25%**). The iterative calculation of the training margins also turned out more successful, in terms of per-class accuracy, than the single step alternative in our class noise correction algorithm. Furthermore, both our removal and correction schemes appear useful for the handling of the mislabeling problem in our satellite data. Let us notice that the particularly low minimum per-class accuracy for data set *Forest* is related to a rare forest structure class (only over 2% of the training data) of this highly imbalanced data set [20]. The imbalance ratio of data *Lake* is even higher (up to 53.8) leading to the failure of all alternatives in the identification of the most difficult class of this data set.

In table 7.5, we carried out a comparative analysis of the classification accuracy achieved by *KNN* for the most difficult class of each noisy dataset for both majority vote and margin-based class noise filters. As in the previous per-class performance comparative analysis, while the majority vote method decreases the minimum accuracy per class of most datasets, our method gets similar or better accuracy (increases the accuracy of up to **9%** for *Statlog*) on the most difficult class for most datasets. However, the iterative margin-based method is not as effective for *KNN*. Indeed, it outperforms its single step counterpart (for both noise removal and correction) on just half the data sets (which are the most imbalanced).

In conclusion, tables 7.2, 7.3, 7.4 and 7.5 show that with respect to our single step noise filter, our iterative noise filter induces a higher rise in per-class accuracy especially for ensemble classification and generally obtains higher overall classification accuracy for both ensemble and single classifiers at a high level of noise.

Data	No filter	Removal			Correction		
		Majority	Margin		Majority	Margin	
			One step	Iterative		One step	Iterative
Forest	0.3	2.7	8.7	**33.2**	0	13.2	**33.2**
Urban	43.6	41.6	**56.6**	55.5	41.4	53.6	51.3
Statlog	27.5	35.5	43.9	48.5	16.0	36.3	**54.8**
Lake	0	0	0	0	0	0	0

Table 7.4: Classification accuracy of boosting classifier for the most difficult class with no filter, majority vote filtered and margin-based filtered training sets on artificially corrupted data sets (noise rate=20%)

Data	No filter	Removal			Correction		
		Majority	Margin		Majority	Margin	
			One step	Iterative		One step	Iterative
Forest	**18.9**	0	**18.9**	16.2	0	**18.9**	16.2
Urban	46.7	44.2	51.6	**55.1**	41.6	52.9	54.4
Statlog	51.3	21.0	55.4	**61.0**	8.7	54.4	**61.0**
Lake	14.5	**15.6**	9.0	0.6	12.0	8.8	0

Table 7.5: Classification accuracy of *KNN* classifier for the most difficult class with no filter, majority vote filtered and margin-based filtered training sets on artificially corrupted data sets (noise rate=20%)

7.2.2.2 Experiments on actual class noise

In our experiments, the actual class noise rate, which is unknown, was assumed to be smaller than the considered artificial class noise. Consequently, $\boldsymbol{M}$ (ratio of the removed or corrected instances) was set to a lower range (from 0 to 20%).

In table 7.6, organised as table 7.2, we attempt to identify, then remove or correct the actual noisy labels, whose amount is unknown, eventually contained in the satellite image data sets. A comparison of the results of *boosting* with no filter shows that our margin-based algorithms get higher or similar accuracy. While our method was generally successful to handle actual mislabeled data, using both removal and correction schemes, (maximum gain in accuracy of over **3%** on *Urban* data set), it does not degrade the accuracy on data set *Statlog*, which might be less corrupted by noisy labels. Moreover, the margin based method significantly outperforms the majority vote scheme for both class noise removal and correction. The best increase in accuracy with respect to the majority vote method is about **7%** on data set *Urban*. Unlike the class noise simulation results presented in table 7.2, the results of our margin-based filters are relatively similar in the case of actual noisy data. This might be attributed to the small amount of actually misclassified instances. Indeed, *random forests* is robust against small to moderate class noise rates but is sensitive to higher levels of noise [166].

Table 7.7 shows the accuracy of *KNN* on the four original satellite image datasets. The obtained results are consistent with the artificial class noise experiment results (table 7.3). The best performance achieved by our margin-based approach is still for the *Urban* data set (increase in accuracy of about **6%** with respect to the majority vote method and of about **7%** with respect to the no filtering alternative).

Data	No filter	Removal			Correction		
		Majority	Margin		Majority	Margin	
			One step	Iterative		One step	Iterative
Forest	88.0	85.0	88.4	88.3	84.2	**88.5**	88.4
Urban	67.2	63.6	69.9	69.9	63.9	69.3	**70.4**
Statlog	**90.0**	88.2	89.9	89.8	85.2	89.9	**90.0**
Lake	75.9	75.8	76.6	77.4	73.8	76.8	**77.6**

Table 7.6: Accuracy of boosting classifier with no filter, majority vote filtered and margin-based filtered training sets on original data sets

Data	No filter	Removal			Correction		
		Majority	Margin		Majority	Margin	
			One step	Iterative		One step	Iterative
Forest	81.1	**83.4**	80.3	80.5	81.3	79.8	80.5
Urban	64.4	65.7	**71.3**	71.2	64.0	**71.3**	70.8
Statlog	89.5	89.1	89.7	89.7	86.0	89.4	**89.8**
Lake	80.0	80.1	80.8	81.6	72.2	80.9	**81.7**

Table 7.7: Accuracy of *KNN* classifier with no filter, majority vote filtered and margin-based filtered training sets on original data sets

Table 7.8 presents the classification accuracy of *boosting* for the most difficult class, on test set, by using the original training sets. It shows that margin-based methods significantly outperform majority vote methods, especially for data set *Forest* (gain in per-class accuracy of almost **40%**). Our margin-based noise removal methods generally outperform our noise correction method for improving *boosting*'s recognition ability on difficult classes. However, in contrast to table 7.4, our single step noise removal method has generally a better performance (the maximum improvement in accuracy being over **4%**) than the iterative scheme. Indeed, the latter is more appropriate for higher levels of noise (see table 7.4 which exhibits a gain in minimum per-class accuracy with respect to the single step filter of up to **25%**), for which the uncertainty in noise rate estimation would have less impact than for low levels of noise. Nonetheless, the iterative noise filter is still effective to increase the accuracy on difficult classes compared with the no filter case.

The predictive performances of *KNN* for the most difficult class in actual noise case are displayed in table 7.9. It can be noticed that, as in the previous table, the iterative

scheme is less effective than the single step margin calculation method. With respect to no filtering classification, while the majority vote filter decreases the prediction accuracy of the most difficult class for most data sets, the margin based method obtains similar or even better results for most remote sensing data. The increase in minimum accuracy per class with respect to the majority vote filter is over **11%** for data set *Urban*. In addition, our margin-based noise correction obtains a relatively better performance (increase in per-class accuracy of up to **3%** for *Urban* data set) than our noise removal methods compared with no filter performances. Hence, the correction of mislabeled data has relatively more impact than throwing assumptive noise on *KNN* performances in low noise level classification tasks.

Data	No filter	Removal			Correction		
		Majority	Margin		Majority	Margin	
			One step	Iterative		One step	Iterative
Forest	46.8	13.5	**52.7**	50.8	3.2	48.4	44.3
Urban	55.8	40.9	**59.0**	58.0	41.3	58.8	57.9
Statlog	59.8	52.9	59.9	60.8	49.9	59.7	**63.1**
Lake	0	0	0	0	0	0	0

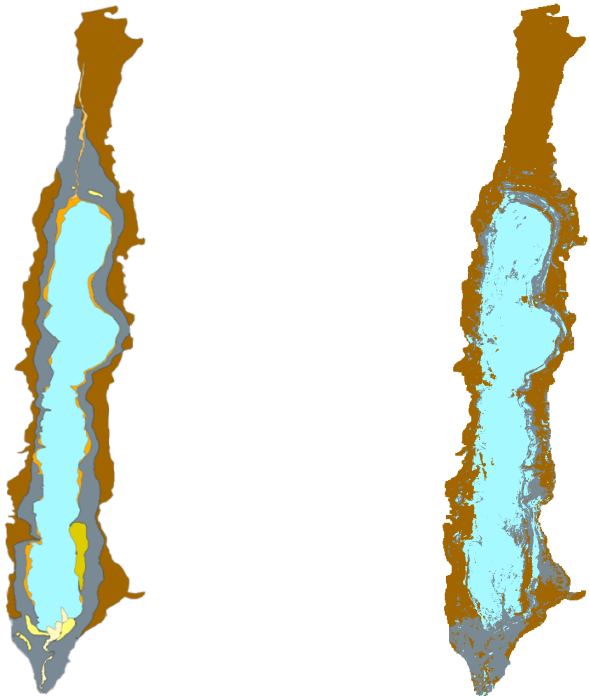
Table 7.8: Classification accuracy of boosting classifier for the most difficult class with no filter, majority vote filtered and margin-based filtered training sets on original data sets

Data	No filter	Removal			Correction		
		Majority	Margin		Majority	Margin	
			One step	Iterative		One step	Iterative
Forest	24.3	18.9	24.3	**27.0**	8.1	**27.0**	**27.0**
Urban	55.6	47.5	58.6	57.3	44.3	**58.7**	56.8
Statlog	**72.3**	63.6	69.2	70.3	49.7	**72.3**	68.7
Lake	16.3	**17.0**	7.2	2.9	13.8	6.1	0.6

Table 7.9: Classification accuracy of *KNN* classifier for the most difficult class with no filter, majority vote filtered and margin-based filtered training sets on original data sets

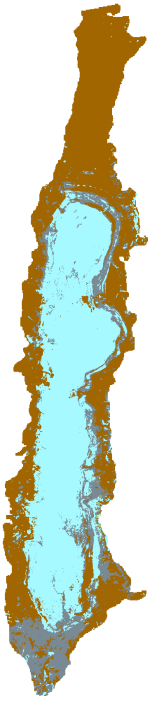
Finally, tables 7.4, 7.5, 7.8 and 7.9 highlight the strength of our margin-based approach in handling difficult and/or rare classes. Indeed, our per-class accuracy results outperform majority vote and no filtering per-class accuracy performances especially for ensemble classifiers, in presence of both artificial and actual noise.

To visually exhibit the effectiveness of our margin based class noise handling method on the improvement of data quality, in the final part of these experiments, we carry out a comparison of our method with no filtering and majority vote filtering on the classification map of remote sensing data. Figure 7.2 presents the classification maps of *boosting* obtained without class noise filtering and by both majority vote based and margin based noise correction methods on *Lake* Landsat Satellite image with artificial class noise rate of 20%. As shown in this figure, with respect to no filtering, the margin based filter reduces the class noise and achieves a better classification map quality. When compared with the classification map provided by the majority vote method, the classification map obtained by the margin based method has more homogeneous regions and is closer to the ground truth of the clean *Lake* image.

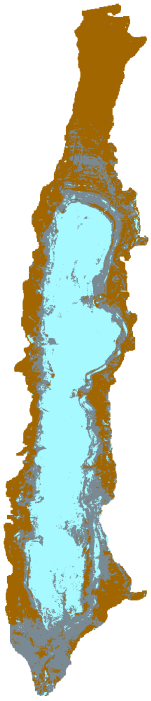


(a) Reference geological map

(b) No filtering



(c) Majority vote correction



(d) Margin based noise correction

Figure 7.2: Classification maps of *boosting* with no filtering, majority vote based and margin based noise correction methods on *Lake* Landsat Satellite image with artificial class noise rate of 20%.

7.3 Margin-based ensemble method for imbalanced land cover classification

In this section, the proposed margin based class imbalance ensemble learning method is applied to classify imbalanced remote sensing data. As in chapter 6, undersampling [13, 108] and SMOTE oversampling [37] combined ensembles will be used in the comparative analysis. However, instead of using bagging for ensemble construction, random forests will be used, as in our approach.

7.3.1 Material

Four multi-class remote sensing data sets, which have been previously used for class noise handling (table 7.1) but in reduced size, are presented in table 7.10, along with the number of features, the number of classes and the number of examples. These data sets present an unequal distribution of the number of examples among the classes. The number of examples for every one of the classes of each dataset are sorted from the class with the lowest amount of examples (smallest class) up to the class with the highest amount of examples (largest class). In our experiments, each data set was divided into two parts: training set and test set.

Data	Train.	Test.	Var.	Class	C1	C2	C3	C4	C5	C6	C7
Forest	3131	3131	20	5	106	604	1130	1194	2430		
Urban	98556	98556	3	4	4323	8597	91138	93054			
Statlog	2500	2500	36	6	485	539	540	1061	1169	1206	
Lake	28288	28289	10	7	408	453	738	1024	14972	17033	21949

Table 7.10: Imbalanced data sets.

7.3.2 Results and discussion

As in the previous experiments related to the mislabeling problem, we used *random forests* [26] to create an ensemble involving Classification and Regression Trees (CART) [27] as base classifiers. For all the imbalanced ensemble classification methods, the size of the ensemble was set to 100. The range of the sampling ratio $\boldsymbol{a}$ for the margin based ensemble method was set from 10 to 100. All of the results shown in this section are the mean values of 10 time calculations. According to the comparative analysis involving four different ensemble margins carried out in chapter 6, sum-margin (equation 2.3), which turned out the most effective margin for class imbalance learning, is adopted in these experiments. Average accuracy and minimum accuracy per class are employed to assess the performance of the imbalance ensemble learning algorithms.

7.3.2.1 Average accuracy

Table 7.11 presents the average classification accuracy obtained on the four imbalanced remote sensing data sets of table 7.10 by margin based extended random forests (Margin based RF) with optimal resampling ranges, traditional random forests (RF), under sampling combined (Under-RF) as well as SMOTE combined random forests (SMOTE-RF). The results show that, all the three imbalance ensemble learning schemes are effective to increase the average accuracy with respect to traditional random forests. However, our margin based undersampling RF ensemble method outperforms random under sampling based and SMOTE oversampling based random forests. Our margin based algorithm is particularly effective on the highly imbalanced dataset *Lake* with an improved average accuracy of over **22%** with respect to RF.

Data	RF	Under-RF	SMOTE-RF	Margin based RF
Forest	81.0	81.9	**84.3**	83.4
Urban	63.7	72.5	71.1	**73.9**
Statlog	87.8	87.4	88.7	**89.0**
Lake	42.1	60.4	57.1	**64.3**

Table 7.11: Average accuracy of margin based random forests, traditional random forests, under sampling combined as well as SMOTE combined random forests.

7.3.2.2 Minimum accuracy per class

Table 7.12 organized as the previous table, exhibits the results on minimum accuracy per class obtained on the four imbalanced remote sensing data sets of table 7.10 by margin based extended random forests (Margin based RF) with optimal resampling ranges, traditional random forests (RF), under sampling combined (Under-RF) as well as SMOTE combined random forests (SMOTE-RF). This table shows that our extended random forests algorithm significantly outperforms traditional random forests on per-class imbalance learning performance. Our method is also better than Under-RF (for all data sets), exhibiting an improvement in per class classification accuracy of up to **10%** (data set *Forest*). With respect to SMOTEBagging, the margin based method obtains a win frequency of **3/4** and improves the minimum accuracy per class of up to **12%** (data set *Lake*).

Figure 7.3(a) exhibits the ground truth of three minority classes (very fine sands, fine sand, sand and gravels) of the *Lake* Landsat Satellite image (figure 7.1) and figures ,7.3(b), 7.3(c), 7.3(d) and 7.3(e) show the classification maps respectively obtained by standard random forests, random undersampling based, SMOTE based random forests and our margin based method. Figure 7.3(b) shows that although random forest is popular and effective for the classification of remote sensing data [3, 57, 143], it is not as effective for highly imbalanced classification. Under sampling based imbalance

Data	RF	Under-RF	SMOTE-RF	Margin based RF
Forest	59.3	58.9	68.3	**69.5**
Urban	42.8	59.0	52.4	**59.2**
Statlog	61.5	70.1	65.5	**76.1**
Lake	0	33.7	22.4	**34.5**

Table 7.12: Minimum accuracy per class of margin based random forests, standard random forests, under sampling combined as well as SMOTE combined random forests.

learning ensemble methods including the random undersampling based and our margin based random forests are more accurate to classify minority class instances than the oversampling based scheme (SMOTE combined RF). However, our margin based method results in a better classification map (figure 7.3(e)), which is closer to the ground truth than the classification map provided by the random undersampling based imbalance learning ensemble method.

7.3.2.3 Influence of model parameters on classification performance

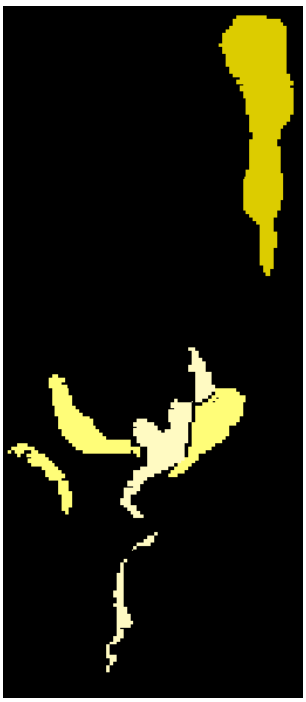
In order to study the influence of ensemble size on random forests construction, we present in figure 7.4 the evaluation of the average accuracy and minimum accuracy per class on Landsat Satellite data *Statlog* with respect to ensemble size throughout the random forests induction processes, i.e. from 20 up to 200 trees for all the random forests methods.

From figure 7.4(a), we can see that our margin based random forests show higher average accuracies than standard random forests as well as the other two state of the art methods undersampling based and SMOTE oversampling based random forests for data set *Statlog* especially when the ensemble size is over 140. In addition, our margin based method presents a relatively smoother curve with respect to standard RF and Under-RF.

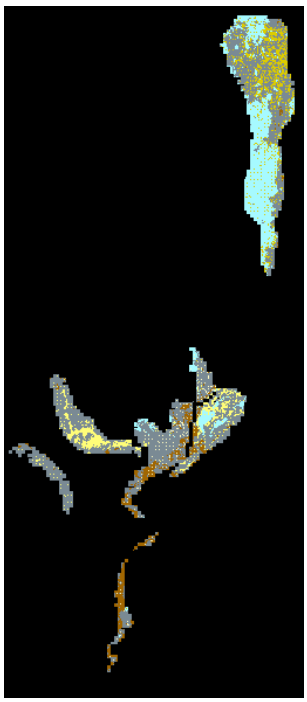
Figure 7.4(b) shows that our method obtains a significantly higher minimum accuracy per class with respect to standard random forests, undersampling based and SMOTE based random forests for data set *Statlog*. In addition, the curve of the margin based imbalance learning method is once again smoother than the curve associated to random undersampling combined random forests.

7.4 Conclusion

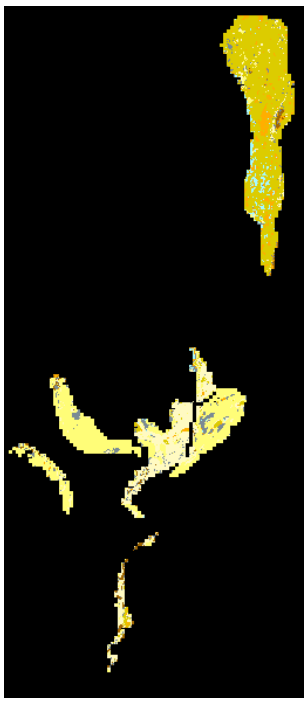
In this chapter, we first applied our ensemble margin based class noise filters to deal with the mislabeling problem in land cover classification. Several conclusions can be summarized based on our experimental evaluations and comparative studies.



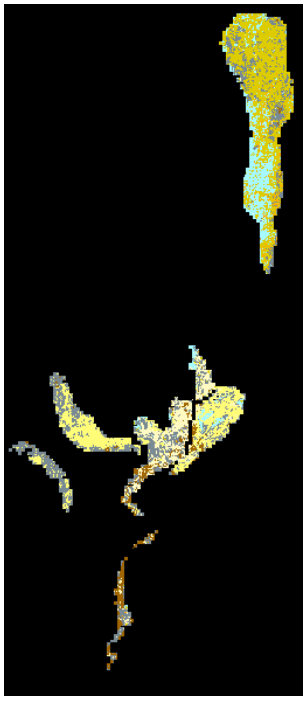
(a) Reference geological map



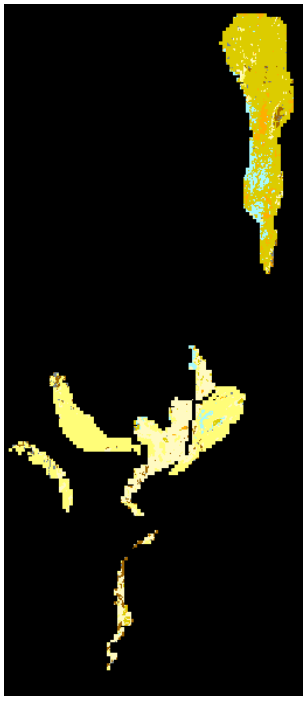
(b) Standard random forests



(c) Undersampling based random forests

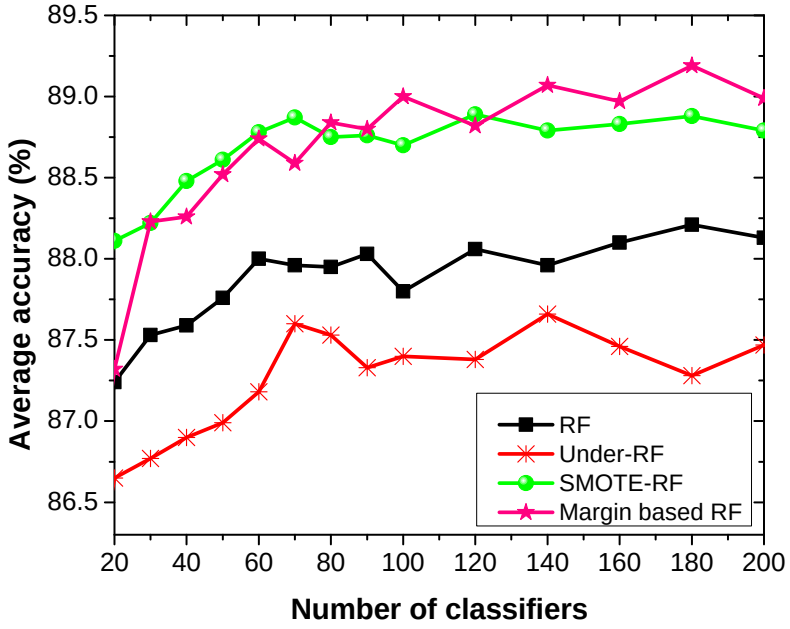


(d) SMOTE based random forests

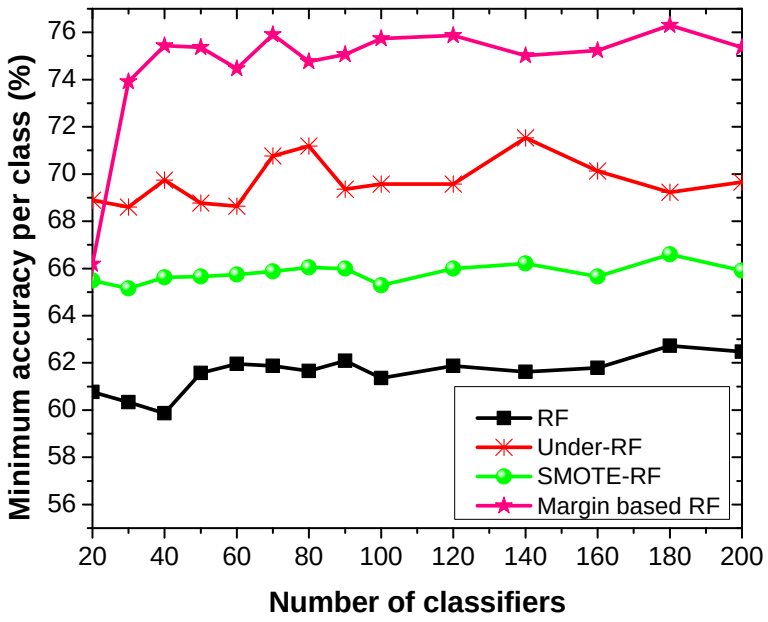


(e) Margin based random forests

Figure 7.3: Ground truth, and classification maps of margin based random forests, standard random forests, undersampling based and SMOTE based random forests, on minority classes of *Lake* Landsat Satellite image.



(a) Average accuracy



(b) Minimum accuracy per class

Figure 7.4: Evolution of the average accuracy and the minimum accuracy per class on data set *Statlog* according to the random forests size.

- Our margin based class noise handling approach is effective for land cover mapping.
- Our method is significantly more effective for both class noise removal and correction than the majority vote class noise filtering method in the cases of both *artificial class noise* and *actual class noise*.
- The margin calculation iterative scheme is generally more effective for class noise handling than its one step-based version.
- The usage of the random forests makes our method distinguish clean data from mislabeled instances more accurately, hence leading to a better noise correction.
- Our noise filter adopts automatic noise detection and cleaning schemes, hence, it has better applicability in remote sensing than traditional noise filters.

Then our novel proposed margin based extended random forests method was applied to address the imbalance problem in remote sensing classification. The conclusions of our experimental results and comparative analysis could be summarized as follows:

- Our margin based imbalanced ensemble classification method is effective for imbalanced land cover mapping.
- Our method outperforms traditional random forests in class imbalance remote sensing classification tasks. With respect to other two state of the art methods, undersampling combined and SMOTE oversampling combined random forests, our extended random forests, involving optimal resampling rates is the most effective through both average accuracy and minimum accuracy per class evaluation measures.

Chapter 8

Conclusion

There are two major challenges in machine learning, *class noise* and *class imbalance*, which are encountered in many real world applications. The goal of this thesis was to investigate the role of the margin theory in ensemble classifier design, and thus to handle the two problems. The final part of this dissertation summarizes the contributions of the thesis and gives directions for future work. This thesis consists of two main parts handling the two major training data issues in the context of ensemble learning:

1. The first part focuses on the main current research issues arising in class noise filtering, and claims that data cleaning for supervised machine learning tasks can be effectively accomplished on the basis of the training margin distribution produced by a robust ensemble algorithm. In particular, this work investigates the hypothesis that the true class noise is a misclassified instance which obtains a high ensemble margin (in absolute value).

2. The second part proposes a novel imbalance learning algorithm by merging ensemble learning and margin theory with undersampling. We argue that this combination for imbalanced classification is more effective than more traditional imbalance learning schemes for several reasons. Firstly, ensemble margin has been used for both algorithm evaluation and as a guideline for finding class decision boundaries, and thus, a natural goal would be to improve the distribution of the training set by utilizing the instances with small margin values. Secondly, current sampling-based ensemble methods alter the overall training data distribution by using data sampling schemes to decrease the imbalance ratio among the different classes simply without exploring the potential for imbalance learning of the instances composing the training set such as class decision boundary instances.

8.1 Main contributions

This thesis enhances the current state-of-the-art in class noise filtering and class imbalance learning and makes key contributions to the following areas:

1. A novel ensemble margin definition is proposed. This margin is an unsupervised version of the classic sum-margin.

2. We give an overview of the noise removal and correction state-of-the-art methods and a summary of existing data cleaning techniques that are related to our research. This is essential to raise the limitations of each of these techniques. The famous ensemble-based *majority vote filter* has been chosen for our comparative analysis.

3. Class noise filtering using ensemble margin

   a) An ensemble margin-based method is proposed to address the mislabeling problem. Our mislabeled training data identification algorithm exploits the ensemble margin and handles both the removal and the correction of noisy labels by both one step and iterative training margins calculation schemes. The effectiveness of our method is evaluated via analyzing the classification performances of two noise sensitive classification models *Adaboosting.M1* and *1-NN*, which are trained from our margin based filtered data. The results of our empirical evaluation demonstrated that, our margin based method is effective for the classification of both image data and non image data, and significantly outperforms the majority vote noise filter.

   b) Two popular ensemble margin definitions, as well as their unsupervised alternatives including the novel unsupervised margin, were assessed in our margin-based handling of the mislabeling problem. The comparative analysis results show that supervised margins generally outperform unsupervised margins, and sum-operation based margins are more effective in class noise handling than max operation based margins.

   c) We compared the performances of our noise removal and noise correction methods. Our experimental results show that, expectedly, our noise removal algorithm outperforms our noise correction algorithm. Although the latter achieves better classification accuracy for noise sensitive classifiers than the majority vote method due mainly to providing a more accurate prediction label for identified mislabeled instances, it still has a risk of producing additional noise. This weakness has to be alleviated as retaining bad data hinders performance more than throwing out good data.

   d) We tested the performance of our iterative guided training margin calculation noise filtering method. This method was demonstrated as useful to improve the classification performance of *Adaboosting.M1*. In addition, iterative data clean algorithms are conservative. Such characteristic can benefit the quality of prediction of small and difficult class instances. Hence, in our experiments,

the iterative filter was effective for the classification of the most difficult class in diverse classification problems.

4. We introduced the research ground of imbalance learning and gave a review of existing methods for imbalanced classification. Then, the limitations of each of these imbalance learning techniques were raised, as done for the analysis of the class noise handling issue. Finally, a summary of existing imbalance learning approaches was presented and two of the most famous algorithms were highlighted and selected for our comparative analysis.

5. A novel bagging method based on ensemble margin for imbalanced classification

   a) We proposed an original margin ordering based bagging method based on an under sampling scheme for imbalanced classification. This algorithm selects more informative training instances, which should benefit ensemble construction and better handle multi class imbalanced classification, by carrying out an estimation of training instance importance which relies on the ensemble margin. We consider the lowest margin instances as the most informative in classification tasks. In other words, instances can be focused on or not by an ensemble of base classifiers according to their margin values. From that study, we have emphasized the superiority of the new proposed bagging method, in handling the imbalance learning problem compared with bagging, UnderBagging and SMOTEBagging.

   b) The performances of four margin definitions involved in our algorithm were also compared. While the sum-margins generally outperform max margins in terms of average accuracy, the latter have better performance in minimum accuracy per class. The supervised margins achieve similar performance with the unsupervised margins. In addition, the effectiveness of the new proposed margin in addressing the class imbalance problem was demonstrated.

6. Application to land cover mapping

   a) We applied our ensemble margin based class noise filters to deal with the mislabeling problem in land cover classification. Our experimental evaluations and comparative studies show that our margin based class noise handling approach is not only effective for land cover mapping but also significantly more effective for both class noise removal and correction than the majority vote class noise filtering method in the cases of both *artificial class noise* and *actual class noise*. Moreover, our noise filter adopts automatic noise detection and cleaning schemes, hence, it has better applicability in remote sensing than traditional noise filters. In addition, the margin calculation iterative scheme turned out generally more effective for class noise handling than its one step-based version. The usage of the random forests makes our method distinguish clean data from mislabeled instances more accurately, hence leading to a better noise correction.

b) The novel proposed margin based extended random forests was applied to address the imbalance problem in remote sensing classification. Our experimental results showed that our method is effective for imbalanced land cover mapping and outperforms the traditional random forests in class imbalance remote sensing classification tasks. Furthermore, with respect to two state of the art methods, undersampling combined and SMOTE oversampling combined random forests, our extended random forests, based on an optimal resampling rate, obtains better results in both average accuracy and minimum accuracy per class performance evaluation measures for all the considered remote sensing data.

8.2 Future work

8.2.1 Class noise filtering

The proposed ensemble margin-based class noise filter is an effective method to improve the classification accuracy on a corrupted data. There are still a few explorations that can be carried out in future work, in particular the following research directions:

- Traditional noise filters tend to remove too much significant data from the training set such as small class instances. Our extended ensemble margin based approach has lower risk of discarding minority class instances, however, it still faces the challenge of addressing the mislabeling problem in imbalanced data. Hence, it can be relevant to investigate how to improve the classification performance on noisy imbalanced data by combining our margin based filter with a data sampling scheme.

- Semi-supervised learning is a learning paradigm which improves the learning performance by taking into account both labeled and unlabeled data. The main idea of semi-supervised learning is increasing the training set size by utilizing labeled data to mark unlabeled data with high confidence. The combination of ensemble margin theory and semi-supervised learning could be interesting to investigate whether this semi-supervised ensemble learning scheme can improve the class noise handling performances with respect to simply removing suspicious instances from noisy training data. In this context, our new ensemble margin would be particularly relevant as it is unsupervised.

- Decomposition in multi-class problems can change the distribution of noisy examples in resulting subproblems and increase the separability of the classes. Hence, a potential research direction is to incorporate a decomposition scheme in our ensemble approach to produce a stronger class noise filter.

8.2.2 Class imbalance learning

Our margin-based class imbalance learning method has been demonstrated as effective for the classification of imbalanced data sets. In future work, there are some potential extensions of the proposed method to investigate.

- We plan to extend the margin-based ensemble framework for class imbalance handling to an oversampling scheme, such as producing additional minority class instances by adopting the *SMOTE* procedure on small margin instances. Producing minority class instances according to class decision boundary samples (small margin values) can result in more diversity in the created ensembles. This method would alleviate some of the weaknesses of SMOTE which tends to inject redundant and noisy samples into the training set. This extension of our algorithm would be relevant for the classification of imbalanced data sets with limited labels encountered in many application domains especially in remote sensing.

- The proposed method can also be extended by combining it with semi-supervised learning. The idea is to improve the classification of an imbalanced data by increasing the minority class size of the training set. In each ensemble learning iteration, the predicted labels of minority class instances with high margins could be added to the current training set and potentially improve its minority class information significance.

Bibliography

[1] J. Abellan and A. R. Masegosa, *An experimental study about simple decision trees for bagging ensemble on datasets with classification noise*, in Symbolic and Quantitative Approaches to Reasoning with Uncertainty, C. Sossai and G. Chemello, eds., vol. 5590 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 446–456.

[2] ——, *Bagging schemes on the presence of class noise in classification*, Expert Systems with Applications, 39 (2012), pp. 6827 – 6837.

[3] O. S. Ahmed, S. E. Franklin, M. A. Wulder, and J. White, *Characterizing stand-level forest canopy cover and height using landsat time series, samples of airborne lidar, and the random forest algorithm*, ISPRS Journal of Photogrammetry and Remote Sensing, 101 (2015), pp. 89 – 101.

[4] F. Aiolli and A. Sperduti, *A re-weighting strategy for improving margins*, Artificial Intelligence, 137 (2002), pp. 197 – 216.

[5] R. Akbani, S. Kwek, and N. Japkowicz, *Applying Support Vector Machines to Imbalanced Datasets*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 39–50.

[6] K. Ali and M. Pazzani, *Error reduction through learning multiple descriptions*, Machine Learning, 24 (1996), pp. 173–202.

[7] M. Alshawabkeh, *Hypothesis margin based weighting for feature selection using boosting: theory, algorithms and applications*, PhD thesis, Northeastern University, 2013.

[8] A. Asuncion and D. Newman, *UCI machine learning repository*, 2007.

[9] R. Banfield, L. Hall, K. Bowyer, and W. Kegelmeyer, *Ensemble diversity measures and their application to thinning*, Information Fusion, 6 (2005), pp. 49–62.

[10] R. Barandela and E. Gasca, *Decontamination of training samples for supervised pattern recognition methods*, in Advances in Pattern Recognition, vol. 1876 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2000, pp. 621–630.

[11] R. Barandela, J. S. Sánchez, and R. M. Valdovinos, *New applications of ensembles of classifiers*, Pattern Analysis & Applications, 6 (2003), pp. 245–256.

[12] P. L. Bartlett, M. I. Jordan, and J. D. Mcauliffe, *Convexity, classification, and risk bounds*, Journal of the American Statistical Association, 101 (2006), pp. pp. 138–156.

[13] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, *A study of the behavior of several methods for balancing machine learning training data*, SIGKDD Explor. Newsl., 6 (2004), pp. 20–29.

[14] E. Bauer and R. Kohavi, *An empirical comparison of voting classification algorithms: Bagging, boosting, and variants*, Machine Learning, 36 (1999), pp. 105–139.

[15] B. Beguet, S. Boukir, D. Guyon, and N. Chehata, *Modelling-based feature selection for classification of forest structure using very high resolution multispectral imagery*, in SMC'2013, IEEE, Int. Conf. on Systems, Man, and Cybernetics, Manchester, UK., 2013, pp. 4294–4299.

[16] B. Bhasuran, G. Murugesan, S. Abdulkadhar, and J. Natarajan, *Stacked ensemble combined with fuzzy matching for biomedical named entity recognition of diseases*, Journal of Biomedical Informatics, 64 (2016), pp. 1 – 9.

[17] G. Biau, *Analysis of a random forests model*, J. Mach. Learn. Res., 13 (2012), pp. 1063–1095.

[18] J. Blaszczyński and J. Stefanowski, *Neighbourhood sampling in bagging for imbalanced data*, Neurocomputing, 150, Part B (2015), pp. 529 – 542.

[19] S. Boukir, L. Guo, and N. Chehata, *Classification of remote sensing data using margin-based ensemble methods.*, in ICIP'2013, IEEE International Conference on Image Processing, 2013, pp. 2602–2606.

[20] S. Boukir, O. Regniers, L. Guo, L. Bombrun, and C. Germain, *Texture-based forest cover classification using random forests and ensemble margin*, in IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2015, pp. 3072–3075.

[21] L. Breiman, *Bias, variance, and arcing classifiers*, Tech. Report 460, Statistics Department, University of California at Berkeley, 1996.

[22] L. Breiman, *Out-of-bag estimation*, tech. report, Department of Statistics, University of California, Berkeley, 1996.

[23] L. Breiman, *Arcing the edge*, tech. report, Statist. Dept. Univ. California, Berkeley CA. USA. Tech. Rep. 486, 1997.

[24] ——, *Prediction games and arcing algorithms*, Neural Comput., 11 (1999), pp. 1493–1517.

[25] L. Breiman, *Randomizing outputs to increase prediction accuracy*, Machine Learning, 40 (2000), pp. 229–242.

[26] ——, *Random forests*, Mach. Learn., 45 (2001), pp. 5–32.

[27] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*, Wadsworth and Brooks, Monterey, CA, 1984.

[28] C. Brodley and M. Friedl, *Improving automated land cover mapping by identifying and eliminating mislabeled observations from training data*, in International Geoscience and Remote Sensing Symposium, 1996. IGARSS, vol. 2, May 1996, pp. 1379–1381.

[29] C. E. Brodley and M. A. Friedl, *Identifying mislabeled training data*, Journal of artificial intelligence research, 11 (1999), pp. 131–167.

[30] C. J. C. Burges, *A tutorial on support vector machines for pattern recognition*, Data Mining and Knowledge Discovery, 2 (1998), pp. 121–167.

[31] J. B. Campbell and R. H. Wynne, *Introduction to Remote Sensing*, The Guilford Press, New York, 5th ed., 2011.

[32] I. Cantador and J. Dorronsoro, *Boosting parallel perceptrons for label noise reduction in classification problems*, in Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, vol. 3562, Springer Berlin Heidelberg, 2005, pp. 586–593.

[33] J. J. Cao, S. Kwong, and R. Wang, *A noise-detection based adaboost algorithm for mislabeled data*, Pattern Recognition, 45 (2012), pp. 4451 – 4465.

[34] C. Catal, O. Alan, and K. Balkan, *Class noise detection based on software metrics and roc curves*, Information Sciences, 181 (2011), pp. 4867 – 4877.

[35] G. D. Cavalcanti, L. S. Oliveira, T. J. Moura, and G. V. Carvalho, *Combining diversity measures for ensemble pruning*, Pattern Recognition Letters, 74 (2016), pp. 38 – 45.

[36] J. C. W. Chan and D. Paelinckx, *Evaluation of random forest and adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery*, Remote Sensing of Environment, 112 (2008), pp. 2999 – 3011.

[37] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, *Smote: Synthetic minority over-sampling technique*, J. Artif. Int. Res., 16 (2002), pp. 321–357.

[38] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, *Smoteboost: Improving prediction of the minority class in boosting*, in Knowledge Discovery in Databases: PKDD 2003, vol. 2838 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2003, pp. 107–119.

[39] C. Chen, A. Liaw, and L. Breiman, *Using random forest to learn imbalanced data*, Tech. Report 666, Department of Statistics, University of California, Berkeley., 2004.

[40] C. H. Chen and P. P. Ho, *Statistical pattern recognition in remote sensing*, Pattern Recognition, 41 (2008), pp. 2731 – 2741.

[41] J. Chen and H. Yu, *Unsupervised ensemble ranking of terms in electronic health record notes based on their importance to patients*, Journal of Biomedical Informatics, 68 (2017), pp. 121 – 131.

[42] W. W. Cohen, *Fast effective rule induction*, in Twelfth International Conference on Machine Learning, Morgan Kaufmann, 1995, pp. 115–123.

[43] M. Condorcet, *Essay on the application of analysis to the probability of majority decisions*, (1785).

[44] C. Cortes and V. Vapnik, *Support-vector networks*, Machine Learning, 20 (1995), pp. 273–297.

[45] T. Cover and P. Hart, *Nearest neighbor pattern classification*, IEEE Transactions on Information Theory, 13 (1967), pp. 21–27.

[46] M. J. Cracknell and A. M. Reading, *Geological mapping using remote sensing data: A comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit spatial information*, Computers & Geosciences, 63 (2014), pp. 22 – 33.

[47] K. Crammer, R. Gilad-bachrach, A. Navot, and N. Tishby, *Margin analysis of the lvq algorithm*, in Advances in Neural Information Processing Systems 2002, MIT press, 2002, pp. 462–469.

[48] P. Cunningham and J. Carney, *Diversity versus quality in classification ensembles based on feature selection*, in 11th European Conference on Machine Learning, Springer, 2000, pp. 109–116.

[49] S. J. Delany and P. Cunningham, *An analysis of case-base editing in a spam filtering system.*, in Advances in Case-Based Reasoning, vol. 3155, Springer, 2004, pp. 128–141.

[50] I. S. Dhillon, Y. Guan, and B. Kulis, *Kernel k-means: Spectral clustering and normalized cuts*, in Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, New York, NY, USA, 2004, pp. 551–556.

[51] T. Dietterich, *An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization*, Machine Learning, 40 (2000), pp. 139–157.

[52] J. Díez-Pastor, J. Rodríguez, C. García-Osorio, and L. I. Kuncheva, *Random balance: Ensembles of variable priors classifiers for imbalanced data*, Knowledge-Based Systems, 85 (2015), pp. 96 – 111.

[53] J. F. Díez-Pastor, J. J. Rodríguez, C. I. García-Osorio, and L. I. Kuncheva, *Diversity techniques improve the performance of the best imbalance learning ensembles*, Information Sciences, 325 (2015), pp. 98–117.

[54] C. Domingo and O. Watanabe, *Madaboost: A modification of adaboost*, in Proceedings of the Thirteenth Annual Conference on Computational Learning Theory, San Francisco, CA, USA, 2000, Morgan Kaufmann Publishers Inc., pp. 180–189.

[55] P. Domingos, *A unified bias-variance decomposition for zero-one and squared loss*, in AAAI/IAAI(Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence), AAAI Press / The MIT Press, 2000, pp. 564–569.

[56] P. Domingos and M. Pazzani, *On the optimality of the simple bayesian classifier under zero-one loss*, Machine Learning, 29 (1997), pp. 103–130.

[57] P. Du, A. Samat, B. Waske, S. Liu, and Z. Li, *Random forest and rotation forest for fully polarized sar image classification using polarimetric and spatial features*, ISPRS Journal of Photogrammetry and Remote Sensing, 105 (2015), pp. 38 – 53.

[58] P. Du, J. Xia, J. Chanussot, and X. He, *Hyperspectral remote sensing image classification based on the integration of support vector machine and random forest*, in IEEE International Geoscience and Remote Sensing Symposium, 2012, pp. 174–177.

[59] P. Du, J. Xia, W. Zhang, K. Tan, Y. Liu, and S. Liu, *Multiple classifier system for remote sensing image classification: A review*, Sensors, 12 (2012), pp. 4764–4792.

[60] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, John Wiley & Sons, 2***nd*** ed., 2001.

[61] B. Efron and R. Tibshirani, *Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy*, Statistical Science, 1 (1986), pp. 54–75.

[62] S. Ertekin, J. Huang, L. Bottou, and C. L. Giles, *Learning on the border: active learning in imbalanced data classification.*, in CIKM (Conference on Information and Knowledge Management), M. J. Silva, A. H. F. Laender, R. A. Baeza-Yates, D. L. McGuinness, B. Olstad, H. Olsen, and A. O. Falcao, eds., ACM, 2007, pp. 127–136.

[63] A. Estabrooks, T. Jo, and N. Japkowicz, *A multiple resampling method for learning from imbalanced data sets*, Computational Intelligence, 20 (2004), pp. 18–36.

[64] X. N. Fan, K. Tang, and T. Weise, *Margin-based over-sampling method for learning from imbalanced datasets*, in Advances in Knowledge Discovery and Data Mining, vol. 6635, Springer Berlin Heidelberg, 2011, pp. 309–320.

[65] A. Fernández, S. García, and F. Herrera, *Addressing the Classification with Imbalanced Data: Open Problems and New Challenges on Class Distribution*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 1–10.

[66] A. Fernández, V. López, M. Galar, M. J. del Jesus, and F. Herrera, *Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches*, Knowledge-Based Systems, 42 (2013), pp. 97 – 110.

[67] A. E. Ferreira and D. Alarcao, *Real-time blind source separation system with applications to distant speech recognition*, Applied Acoustics, 113 (2016), pp. 170 – 184.

[68] B. Frenay and M. Verleysen, *Classification in the presence of label noise: A survey*, IEEE Transactions on Neural Networks and Learning Systems, 25 (2014), pp. 845–869.

[69] Y. Freund and R. Schapire, *Experiments with a new boosting algorithm*, in The 13th International Conference on Machine Learning, ICML'96, 1996, pp. 148–156.

[70] Y. Freund and R. E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting*, Journal of Computer and System Sciences, 55 (1997), pp. 119 – 139.

[71] D. G. M. Friedman, N.and Geiger, *Bayesian network classifiers*, Mach. Learn., 29 (1997), pp. 131–163.

[72] J. H. Friedman and P. Hall, *On bagging and nonlinear estimation*, Journal of Statistical Planning and Inference, 137 (2007), pp. 669 – 683.

[73] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, *A review on ensembles for the class imbalance problem: Bagging-,*

boosting-, and hybrid-based approaches, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 42 (2012), pp. 463–484.

[74] M. Galar, A. Fernández, E. Barrenechea, and F. Herrera, *Eusboost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling*, Pattern Recognition, 46 (2013), pp. 3460 – 3471.

[75] D. Gamberger, N. Lavrac, and S. Dzeroski, *Noise detection and elimination in preprocessing: Experiments in medical domains*, Applied Artificial Intelligence, 14 (2000), pp. 205–223.

[76] W. Gao and Z. H. Zhou, *The kth, median and average margin bounds for adaboost*, CoRR (Computing Research Repository), abs/1009.3613 (2010).

[77] S. García and F. Herrera, *Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy*, Evol. Comput., 17 (2009), pp. 275–306.

[78] S. Garcia, J. Luengo, and F. Herrera, *Dealing with noisy data*, in Data Preprocessing in Data Mining, vol. 72 of Intelligent Systems Reference Library, Springer, 2015, pp. 107–145.

[79] U. Gessner, M. Machwitz, C. Conrad, and S. Dech, *Estimating the fractional cover of growth forms and bare surface in savannas. a multi-resolution approach based on regression tree ensembles*, Remote Sensing of Environment, 129 (2013), pp. 90 – 102.

[80] P. Gislason, J. Benediktsson, and J. Sveinsson, *Random forests for land cover classification*, Pattern Recognition Letters, 27 (2006), pp. 294–300.

[81] S. Gu and Y. Jin, *Multi-train: A semi-supervised heterogeneous ensemble classifier*, Neurocomputing, 249 (2017), pp. 202 – 211.

[82] L. Guo, *Margin framework for ensemble classifiers. Application to remote sensing data*, PhD thesis, University of Bordeaux, 2011.

[83] L. Guo and S. Boukir, *Margin-based ordered aggregation for ensemble pruning*, Pattern Recognition Letters, 34 (2013), pp. 603–609.

[84] ——, *Ensemble margin framework for image classification*, in ICIP'2014, IEEE International Conference on Image Processing, 2014, pp. 4231–4235.

[85] ——, *Ensemble margin framework for image classification*, in ICIP'2014, IEEE International Conference on Image Processing, 2014, pp. 4231–4235.

[86] L. Guo, S. Boukir, and N. Chehata, *Support vectors selection for supervised learning using an ensemble approach*, in ICPR'2010, 20th IAPR International Conference on Pattern Recognition, 2010, pp. 37–40.

[87] L. Guo, N. Chehata, C. Mallet, and S. Boukir, *Relevance of airborne lidar and multispectral image data for urban scene classification using random forests*, ISPRS Journal of Photogrammetry and Remote Sensing, 66 (2011), pp. 56 – 66.

[88] I. Guyon, N. Matic, and V. Vapnik, *Advances in knowledge discovery and data mining*, American Association for Artificial Intelligence, 1996, ch. Discovering Informative Patterns and Data Cleaning, pp. 181–203.

[89] M. Han, X. R. Zhu, and W. Yao, *Remote sensing image classification based on neural network ensemble algorithm*, Neurocomputing, 78 (2012), pp. 133 – 138.

[90] S. Hariharan, S. Dhanasekar, and K. Desikan, *Reachability based web page ranking using wavelets*, Procedia Computer Science, 50 (2015), pp. 157 – 162.

[91] I. K. Haryana, V. N. Fikriyah, and N. V.Yulianti, *Application of remote sensing and geographic information system for settlement land use classification planning in bantul based on earthquake disaster mitigation (case study: Bantul earthquake, may 27th 2006)*, Procedia Environmental Sciences, 17 (2013), pp. 434 – 443.

[92] T. Hastie and G. E. Batista, *Classification by pairwise coupling*, Ann. Statist., 26 (1998), pp. 451–471.

[93] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd ed., 1998.

[94] ——, *Neural Networks: A Comprehensive Foundation (3rd Edition)*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2007.

[95] H. He and Y. Ma, *Imbalanced learning: foundations, algorithms, and applications*, Wiley-IEEE Press, 2013.

[96] H. B. He and E. A. Garcia, *Learning from imbalanced data*, IEEE Transactions on Knowledge and Data Engineering, 21 (2009), pp. 1263–1284.

[97] M. A. Hernandez and S. J. Stolfo, *Real-world data is dirty: Data cleansing and the merge/purge problem*, Data Mining and Knowledge Discovery, 2 (1998), pp. 9–37.

[98] S. Hido, H. Kashima, and Y. Takahashi, *Roughly balanced bagging for imbalanced data*, Stat. Anal. Data Min., 2 (2009), pp. 412–426.

[99] T. K. Ho, *The random subspace method for constructing decision forests*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20 (1998), pp. 832–844.

[100] R. C. Holte, L. E. Acker, and B. W. Porter, *Concept learning and the problem of small disjuncts*, in Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 1, IJCAI'89, San Francisco, CA, USA, 1989, Morgan Kaufmann Publishers Inc., pp. 813–818.

[101] M. Hosseini, M. Saradjian, A. Javahery, and S. Nadi, *Noise removal from land cover maps by post processing of classification results*, in RAST '07. 3rd International Conference on Recent Advances in Space Technologies, June 2007, pp. 309–314.

[102] A. v. d. Hout and P. G. M. v. d. Heijden, *Randomized response, statistical disclosure control and misclassification: A review*, International Statistical Review, 70 (2002), pp. pp. 269–288.

[103] Q. Hu, L. Li, X. Wu, G. Schaefer, and D. Yu, *Exploiting diversity for optimizing margin distribution in ensemble learning*, Knowledge-Based Systems, 67 (2014), pp. 90 – 104.

[104] S. G. Hu, Y. F. Liang, L. T. Ma, and Y. He, *Msmote: Improving classification performance when training data is imbalanced*, in Computer Science and Engineering, 2009. WCSE '09. Second International Workshop on, vol. 2, Oct 2009, pp. 13–17.

[105] N. Hughes, S. Roberts, and L. Tarassenko, *Semi-supervised learning of probabilistic models for ecg segmentation*, in Engineering in Medicine and Biology Society, 2004. 26th Annual International Conference of the IEEE, vol. 1, Sept 2004, pp. 434–437.

[106] J. J. Blaszczyński; Stefanowski and L. Idkowiak, *Extending bagging for imbalanced data.*, in Proceeding of the eighth CORES (Core Ordering and Reporting Enterprise System), Springer Series on Advances in Intelligent Systems and Computing, vol. 226, 2013, pp. 269–278.

[107] S. E. G. J. L. Morgan and N. C. Coops, *Aerial photography: A rapidly evolving tool for ecological management*, Biological Sciences, 60 (2010), pp. 47–59.

[108] N. Japkowicz and S. Stephen, *The class imbalance problem: A systematic study*, Intelligence Data Analysis, 6 (2002), pp. 429–449.

[109] R. Jin and J. Zhang, *Multi-class learning by smoothed boosting*, Mach. Learn., 67 (2007), pp. 207–227.

[110] T. Jo and N. Japkowicz, *Class imbalances versus small disjuncts*, ACM SIGKDD (Special Interest Group on Knowledge Discovery and Data Mining) Explorations Newsletter, 6 (2004), pp. 40–49.

[111] G. John, *Robust decision trees: Removing outliers from databases*, in First International Conference on Knowledge Discovery and Data Mining, 1995, pp. 174–179.

[112] B. A. Johnson and K. Iizuka, *Integrating openstreetmap crowdsourced data and landsat time-series imagery for rapid land use/land cover (lulc) mapping: Case study of the laguna de bay area of the philippines*, Applied Geography, 67 (2016), pp. 140 – 149.

[113] M. Kapp, R. Sabourin, and P. Maupin, *An empirical study on diversity measures and margin theory for ensembles of classifiers*, in 10th International Conference on Information Fusion, July 2007, pp. 1–8.

[114] A. Karmaker and S. Kwek, *A boosting approach to remove class label noise*, in Fifth International Conference on Hybrid Intelligent Systems, vol. 3, Nov 2005, pp. 169–177.

[115] T. M. Khoshgoftaar, A. Fazelpour, D. J. Dittman, and A. Napolitano, *Ensemble vs. data sampling: Which option is best suited to improve classification performance of imbalanced bioinformatics data?*, in IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), 2015, pp. 705–712.

[116] T. M. Khoshgoftaar, J. V. Hulse, and A. Napolitano, *Comparing boosting and bagging techniques with noisy and imbalanced data*, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 41 (2011), pp. 552–568.

[117] T. M. Khoshgoftaar, S. Zhong, and V. Joshi, *Enhancing software quality estimation using ensemble-classifier based noise filtering*, Intell. Data Anal., 9 (2005), pp. 3–27.

[118] Y. Kim, *Averaged boosting: A noise-robust ensemble method*, in Advances in Knowledge Discovery and Data Mining, vol. 2637 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2003, pp. 388–393.

[119] S. Kluckner and H. Bischof, *Semantic classification by covariance descriptors within a randomized forest*, in IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), Sept 2009, pp. 665–672.

[120] B. Krawczyk, *Learning from imbalanced data: open challenges and future directions*, Progress in Artificial Intelligence, (2016), pp. 1–12.

[121] A. Krieger, C. Long, and A. Wyner, *Boosting noisy data*, in Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, San Francisco, CA, USA, 2001, Morgan Kaufmann Publishers Inc., pp. 274–281.

[122] L. I. Kuncheva and C. J. Whitaker, *Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy*, Machine Learning, 51 (2003), pp. 181–207.

[123] B. Le Saux, *Interactive design of object classifiers in remote sensing*, in 22nd International Conference on Pattern Recognition (ICPR), Aug 2014, pp. 2572–2577.

[124] C. Li, B. Kuo, C. Lin, and C. Huang, *A spatial-contextual support vector machine for remotely sensed image classification*, IEEE Transactions on Geoscience and Remote Sensing, 50 (2012), pp. 784–799.

[125] L. Li, A. Pratap, H. T. Lin, and Y. S. Abu-Mostafa, *Improving generalization by data categorization*, in Knowledge Discovery in Databases: PKDD 2005, vol. 3721, Springer Berlin Heidelberg, 2005, pp. 157–168.

[126] L. J. Li, B. Zou, Q. H. Hu, X. Q. Wu, and D. R. Yu, *Dynamic classifier ensemble using classification confidence*, Neurocomputing, 99 (2013), pp. 581 – 591.

[127] Y. Lin, Y. Lee, and G. Wahba, *Support vector machines for classification in nonstandard situations*, Machine Learning, 46 (2002), pp. 191–202.

[128] C. X. Ling and V. S. Sheng, *Cost-sensitive Learning and the Class Imbalanced Problem*, Sammut C (ed) Encyclopedia of machine learning. Springer., Berlin, 2008.

[129] F. T. Liu, K. M. Ting, Y. Yu, and Z.-H. Zhou, *Spectrum of variable-random trees*, J. Artif. Int. Res., 32 (2008), pp. 355–384.

[130] T. Y. Liu, *Easyensemble and feature selection for imbalance data sets*, in Bioinformatics, Systems Biology and Intelligent Computing, 2009. IJCBS '09. International Joint Conference on, Aug 2009, pp. 517–520.

[131] X. Y. Liu and Z. H. Zhou, *Ensemble methods for class imbalance learning*, H. He and Y. Ma (Eds), Imbalanced Learning: Foundations, Algorithms,and Applications, Wiley-IEEE Press, (2013), pp. 61–82.

[132] Y. H. Liu and Y. T. Chen, *Total margin based adaptive fuzzy support vector machines for multiview face recognition*, in 2005 IEEE International Conference on Systems, Man and Cybernetics, vol. 2, 2005, pp. 1704–1711.

[133] V. López, A. Fernández, S. Garcia, V. Palade, and F. Herrera, *An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics*, Information Sciences, 250 (2013), pp. 113 – 141.

[134] K. Lowell, P. Woodgate, G. Richards, S. Jones, and L. Buxton, *Fuzzy reliability assessment of multi-period land-cover change maps*, Photogramm. Eng. Remote Sens., 71 (2005), pp. 939–945.

[135] O. Loyola-González, J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa, and M. García-Borroto, *Study of the impact of resampling methods for contrast pattern based classifiers in imbalanced databases*, Neurocomputing, 175, Part B (2016), pp. 935 – 947.

[136] E. Marchiori, *Class conditional nearest neighbor for large margin instance selection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 32 (2010), pp. 364–370.

[137] G. Martinez-Munoz and A. Suarez, *Pruning in ordered bagging ensembles*, in ICML'2006, 23rd International Conference on Machine Learning, 2006, pp. 609–616.

[138] H. Masnadi-shirazi and N. Vasconcelos, *On the design of loss functions for classification: theory, robustness to outliers, and savageboost*, in Advances in Neural Information Processing Systems 21, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds., Curran Associates, Inc., 2009, pp. 1049–056.

[139] P. Mather and B. Tso, *Classification Methods for Remotely Sensed Data*, CRC Press, 2 ed., 2009.

[140] ——, eds., *Classification Methods for Remotely Sensed Data*, CRC Press, 2 ed., 2016.

[141] D. Mease, A. J. Wyner, and A. Buja, *Boosted classification trees and class probability/quantile estimation*, Journal of Machine Learning Research, 8 (2007), pp. 409–439.

[142] A. Mellor, S. Boukir, A. Haywood, and S. Jones, *Using ensemble margin to explore issues of training data imbalance and mislabeling on large area land cover classification*, in ICIP'2014, IEEE International Conference on Image Processing, 2014, pp. 26–29.

[143] A. Mellor, S. Boukir, A. Haywood, and S. Jones, *Exploring issues of training data imbalance and mislabelling on random forest performance for large area land cover classification using the ensemble margin*, Journal of Photogrammetry and Remote Sensing, 105 (2015), pp. 155 – 168.

[144] A. Mellouk and A. Chebira, *Machine learning*, InTechOpen, 2009.

[145] P. Melville and R. J. Mooney, *Constructing diverse classifier ensembles using artificial training examples*, in Proceedings of the 18th International Joint Conference on Artificial Intelligence, IJCAI'03, San Francisco, CA, USA, 2003, Morgan Kaufmann Publishers Inc., pp. 505–510.

[146] B. H. Menze, B. M. Kelm, R. Masuch, U. Himmelreich, P. Bachert, W. Petrich, and F. A. Hamprecht, *A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data*, BMC (BioMed Central) Bioinformatics, 10 (2009), pp. 1–16.

[147] A. L. Miranda, L. P. Garcia, A. C. Carvalho, and A. C. Lorena, *Use of classification algorithms in noise detection and elimination*, in Hybrid Artificial Intelligence Systems, Springer Berlin Heidelberg, 2009, pp. 417–424.

[148] T. Munkhdalai, O.-E. Namsrai, and K. H. Ryu, *Self-training in significance space of support vectors for imbalanced biomedical event data*, BMC (BioMed Central) Bioinformatics, 16 (2015), pp. 1–8.

[149] V. Nikulin, G. J. McLachlan, and S. K. Ng, *Ensemble Approach for the Classification of Imbalanced Data*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 291–300.

[150] S. Okamoto and Y. Nobuhiro, *An average-case analysis of the k-nearest neighbour classifier for noisy domain*, in Proc. 15th Int. Joint Conf. Artif. Intell., vol. 1, Aug. 1997, pp. 238–243.

[151] N. C. Oza, *Aveboost2: Boosting for noisy data*, in Multiple Classifier Systems, vol. 3077 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2004, pp. 31–40.

[152] Y. Park and J. Ghosh, *Ensembles of* $(alpha)$*-trees for imbalanced classification problems*, IEEE Transactions on Knowledge and Data Engineering, 26 (2014), pp. 131–143.

[153] M. Pechenizkiy, A. Tsymbal, S. Puuronen, and O. Pechenizkiy, *Class noise and supervised learning in medical domains: The effect of feature extraction*, in 19th IEEE International Symposium on Computer-Based Medical Systems, 2006, pp. 708–713.

[154] M. P. Perrone and L. N. Cooper, *When networks disagree: Ensemble methods for hybrid neural networks*, in Artificial Neural Networks for Speech and Vision, R. Mammone, ed., New York, 1993, Chapman and Hall, pp. 126–142.

[155] M. Prasad and A. Sowmya, *Multi-class unsupervised classification with label correction of hrct lung images*, in Proceedings of International Conference on Intelligent Sensing and Information Processing, 2004, 2004, pp. 51–56.

[156] Y. Qian, Y. Liang, M. Li, G. Feng, and X. Shi, *A resampling ensemble algorithm for classification of imbalance problems*, Neurocomputing, 143 (2014), pp. 57 – 67.

[157] Y. Qian, K. Zhang, and F. Qiu, *Spatial contextual noise removal for post classification smoothing of remotely sensed images*, in Proceedings of the 2005 ACM Symposium on Applied Computing, SAC '05, New York, NY, USA, 2005, ACM, pp. 524–528.

[158] J. R. Quinlan, *Induction of decision trees*, Machine Learning, 1 (1986), pp. 81–106.

[159] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[160] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2013.

[161] U. Rebbapragada, C. Brodley, D. Sulla-Menashe, and M. Friedl, *Active label correction*, in IEEE 12th International Conference on Data Mining, Dec 2012, pp. 1080–1085.

[162] U. Rebbapragada, R. Lomasky, C. Brodley, and M. Friedl, *Generating high-quality training data for automated land-cover mapping*, in IEEE International Geoscience and Remote Sensing Symposium. IGARSS, vol. 4, July 2008, pp. IV – 546–IV – 548.

[163] L. Reyzin and R. E. Schapire, *How boosting the margin can also boost classifier complexity*, in Proceedings of the 23rd International Conference on Machine Learning, ICML '06, New York, NY, USA, 2006, ACM, pp. 753–760.

[164] R. Rifkin and A. Klautau, *In defense of one-vs-all classification*, J. Mach. Learn. Res., 5 (2004), pp. 101–141.

[165] S. Río, V. López, J. M. Benítez, and F. Herrera, *On the use of mapreduce for imbalanced big data using random forest*, Information Sciences, 285 (2014), pp. 112 – 137.

[166] V. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. Rigol-Sanchez, *An assessment of the effectiveness of a random forest classifier for land-cover classification*, ISPRS Journal of Photogrammetry and Remote Sensing, 67 (2012), pp. 93 – 104.

[167] ——, *An assessment of the effectiveness of a random forest classifier for land-cover classification*, ISPRS Journal of Photogrammetry and Remote Sensing, 67 (2012), pp. 93 – 104.

[168] J. Rogan, J. Franklin, D. Stow, J. Miller, C. Woodcock, and D. Roberts, *Mapping land-cover modifications over large areas: A comparison of machine learning algorithms*, Remote Sensing of Environment, 112 (2008), pp. 2272 – 2283.
Earth Observations for Terrestrial Biodiversity and Ecosystems Special Issue.

[169] L. Rokach, *Ensemble-based classifiers*, Artificial Intelligence Review, 33 (2010), pp. 1–39.

[170] M. Sabzevari, G. Martinez-Munoz, and A. Suarez, *Small margin ensembles can be robust to class-label noise*, Neurocomputing, (2015).

[171] J. A. Saez, M. Galar, J. Luengo, and F. Herrera, *Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition*, Knowledge and Information Systems, 38 (2014), pp. 179–206.

[172] J. A. Sáez, B. Krawczyk, and M. Woźniak, *Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets*, Pattern Recognition, 57 (2016), pp. 164 – 178.

[173] J. A. Saez, J. Luengo, and F. Herrera, *Predicting noise filtering efficacy with data complexity measures for nearest neighbour classification*, Pattern Recognition, 46 (2013), pp. 355 – 364.

[174] J. A. Sáez, J. Luengo, J. Stefanowski, and F. Herrera, *Smote-ipf: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering*, Information Sciences, 291 (2015), pp. 184 – 203.

[175] R. E. Schapire, *The strength of weak learnability*, Mach. Learn., 5 (1990), pp. 197–227.

[176] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, *Boosting the margin: A new explanation for the effectiveness of voting methods*, The Annals of Statistics, 26 (1998), pp. 1651–2080.

[177] R. E. Schapire and Y. Singer, *Improved boosting algorithms using confidence-rated predictions*, Machine Learning, 37 (1999), pp. 297–336.

[178] N. Segata, E. Blanzieri, S. Delany, and P. Cunningham, *Noise reduction for instance-based learning with a local maximal margin approach*, Journal of Intelligent Information Systems, 35 (2010), pp. 301–331.

[179] C. Seiffert, T. M. Khoshgoftaar, J. V. Hulse, and A. Napolitano, *Rusboost: A hybrid approach to alleviating class imbalance*, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 40 (2010), pp. 185–197.

[180] C. H. Shen and H. X. Li, *Boosting through optimization of margin distributions*, Trans. Neur. Netw., 21 (2010), pp. 659–666.

[181] M. Siers and M. Z. Islam, *Software defect prediction using a cost sensitive decision forest and voting, and a potential solution to the class imbalance problem*, Information Systems, 51 (2015), pp. 62 – 71.

[182] B. Sluban, D. Gamberger, and N. Lavrac, *Ensemble-based noise detection: noise ranking and visual performance evaluation*, Data Mining and Knowledge Discovery, (2013), pp. 1–39.

[183] B. Sluban and N. Lavrac, *Relating ensemble diversity and performance: A study in class noise detection*, Neurocomputing, 160 (2015), pp. 120 – 131.

[184] A. J. Smola and P. J. Bartlett, eds., *Advances in Large Margin Classifiers*, MIT Press, 2000.

[185] A. Stumpf and N. Kerle, *Object-oriented mapping of landslides using random forests*, Remote Sensing of Environment, 115 (2011), pp. 2564 – 2577.

[186] A. Stumpf, N. Lachiche, J. P. Malet, N. Kerle, and A. Puissant, *Active learning in the spatial domain for remote sensing image classification*, IEEE Transactions on Geoscience and Remote Sensing, 52 (2014), pp. 2492–2507.

[187] T. Sun, L. Jiao, J. Feng, F. Liu, and X. Zhang, *Imbalanced hyperspectral image classification based on maximum margin*, IEEE Geoscience and Remote Sensing Letters, 12 (2015), pp. 522–526.

[188] Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, *Cost-sensitive boosting for classification of imbalanced data*, Pattern Recognition, 40 (2007), pp. 3358 – 3378.

[189] Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, and Y. Zhou, *A novel ensemble method for classifying imbalanced data*, Pattern Recognition, 48 (2015), pp. 1623 – 1637.

[190] M. A. Tahir, J. Kittler, and F. Yan, *Inverse random under sampling for class imbalance problem and its application to multi-label classification*, Pattern Recognition, 45 (2012), pp. 3738 – 3750.

[191] A. C. Tan, D. Gilbert, and Y. Deville, *Multi-class protein fold classification using a new ensemble machine learning approach*, Genome Informatics, 14 (2003), pp. 206–217.

[192] C. Teng, *Correcting noisy data*, in Proceedings of the Sixteenth International Conference on Machine Learning, 1999, pp. 239–248.

[193] P. Thanathamathee and C. Lursinsap, *Handling imbalanced data sets with synthetic boundary data generation using bootstrap re-sampling and adaboost techniques*, Pattern Recognition Letters, 34 (2013), pp. 1339 – 1347.

[194] J. Thongkam, G. Xu, Y. Zhang, and F. Huang, *Support vector machine for outlier detection in breast cancer survivability prediction*, in Advanced Web and Network Technologies, and Applications, Y. Ishikawa, J. He, G. Xu, Y. Shi, G. Huang, C. Pang, Q. Zhang, and G. Wang, eds., vol. 4977 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2008, pp. 99–109.

[195] T. Tran, D. Phung, and S. Venkatesh, *Collaborative filtering via sparse markov random fields*, Information Sciences, 369 (2016), pp. 221 – 237.

[196] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag New York, Inc., 1995.

[197] S. Verbaeten and A. Assche, *Ensemble methods for noise elimination in classification problems*, in Multiple Classifier Systems, vol. 2709 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2003, pp. 317–325.

[198] A. Vezhnevets and O. Barinova, *Avoiding boosting overfitting by removing confusing samples*, in European Conference on Machine Learning: ECML 2007, vol. 4701 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 430–441.

[199] R. Y. Wang, V. C. Storey, and C. P. Firth, *A framework for analysis of data quality research*, IEEE Transactions on Knowledge and Data Engineering, 7 (1995), pp. 623–640.

[200] S. Wang and X. Yao, *Diversity analysis on imbalanced data sets by using ensemble models*, in IEEE Symposium on Computational Intelligence and Data Mining., March 2009, pp. 324–331.

[201] ——, *Multiclass imbalance problems: Analysis and potential solutions*, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 42 (2012), pp. 1119–1130.

[202] B. Waske, J. A. Benediktsson, and J. R. Sveinsson, *MCS 2009, 8th International Workshop on Multiple Classifier Systems, Reykjavik, Iceland, June 10-12, 2009. Proceedings*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, ch. Classifying Remote Sensing Data with Support Vector Machines and Imbalanced Training Data, pp. 375–384.

[203] U. Wattanachon and C. Lursinsap, *Spsm: A new hybrid data clustering algorithm for nonlinear data analysis*, International Journal of Pattern Recognition and Artificial Intelligence, 23 (2009), pp. 1701–1737.

[204] V. Wheway, *Using boosting to detect noisy data*, in Advances in Artificial Intelligence. Pacific Rim International Conference on Artificial Intelligence 2000 Workshop Reader, R. Kowalczyk, S. Loke, N. E. Reed, and G. Williams, eds., vol. 2112 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2001, pp. 123–130.

[205] T. Windeatt, *Diversity measures for multiple classifier system analysis and design.*, Information Fusion, 6 (2005), pp. 21–36.

[206] Z. X. Xie, Y. Xu, Q. H. Hu, and P. F. Zhu, *Margin distribution based bagging pruning*, Neurocomputing, 85 (2012), pp. 11 – 19.

[207] J. Zhang and I. Mani, *Knn approach to unbalanced data distributions: A case study involving information extraction*, in Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets, 2003.

[208] L. Zhang and P. N. Suganthan, *Random forests with ensemble of feature spaces*, Pattern Recognition, 47 (2014), pp. 3429 – 3437.

[209] Y. Zhang and W. N. Street, *Bagging with adaptive costs*, IEEE Transactions on Knowledge and Data Engineering, 20 (2008), pp. 577–588.

[210] Y. G. Zhang, B. L. Zhang, F. Coenenz, and W. J. Lu, *Highly reliable breast cancer diagnosis with cascaded ensemble classifiers*, in The 2012 International Joint Conference on Neural Networks, June 2012, pp. 1–8.

[211] Z. Zhou and X.-Y. Liu, *Training cost-sensitive neural networks with methods addressing the class imbalance problem*, IEEE Transactions on Knowledge and Data Engineering, 18 (2006), pp. 63–77.

[212] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*, Chapman and Hall/CRC, 1st ed., June 2012.

[213] Z. H. Zhou and Y. Jiang, *Medical diagnosis with c4.5 rule preceded by artificial neural network ensemble*, IEEE Transactions on Information Technology in Biomedicine, 7 (2003), pp. 37–42.

[214] X. Zhu and X. Wu, *Class noise vs. attribute noise: A quantitative study*, Artificial Intelligence Review, 22 (2004), pp. 177–210.

[215] X. Q. Zhu, X. D. Wu, and Q. J. Chen, *Eliminating class noise in large datasets*, in Proceeding of International Conference on Machine Learning, ICML2003, 2003, pp. 920–927.

Abstract

Classification has been widely studied in machine learning. Ensemble methods, which build a classification model by integrating multiple component learners, achieve higher performances than a single classifier. The classification accuracy of an ensemble is directly influenced by the quality of the training data used. However, real-world data often suffers from class noise and class imbalance problems.

Ensemble margin is a key concept in ensemble learning. It has been applied to both the theoretical analysis and the design of machine learning algorithms. Several studies have shown that the generalization performance of an ensemble classifier is related to the distribution of its margins on the training examples. This work focuses on exploiting the margin concept to improve the quality of the training set and therefore to increase the classification accuracy of noise sensitive classifiers, and to design effective ensemble classifiers that can handle imbalanced datasets. A novel ensemble margin definition is proposed. It is an unsupervised version of a popular ensemble margin. Indeed, it does not involve the class labels.

Mislabeled training data is a challenge to face in order to build a robust classifier whether it is an ensemble or not. To handle the mislabeling problem, we propose an ensemble margin-based class noise identification and elimination method based on an existing margin-based class noise ordering. This method can achieve a high mislabeled instance detection rate while keeping the false detection rate as low as possible. It relies on the margin values of misclassified data, considering four different ensemble margins, including the novel proposed margin. This method is extended to tackle the class noise correction which is a more challenging issue.

The instances with low margins are more important than safe samples, which have high margins, for building a reliable classifier. A novel bagging algorithm based on a data importance evaluation function relying again on the ensemble margin is proposed to deal with the class imbalance problem. In our algorithm, the emphasis is placed on the lowest margin samples. This method is evaluated using again four different ensemble margins in addressing the imbalance problem especially on multi-class imbalanced data.

In remote sensing, where training data are typically ground-based, mislabeled training data is inevitable. Imbalanced training data is another problem frequently encountered in remote sensing. Both proposed ensemble methods involving the best margin definition for handling these two major training data issues are applied to the mapping of land covers.

Keywords

Bagging, classification, ensemble learning, ensemble margin, imbalanced data, mislabeled data, random forests, remote sensing.