Large-scale Machine Learning Algorithms for Big Data

A DISSERTATION PRESENTED
BY
YAN YAN
TO
CENTRE FOR ARTIFICIAL INTELLIGENCE
FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE SUBJECT OF
COMPUTER SCIENCE

UNIVERSITY OF TECHNOLOGY SYDNEY
ULTIMO, NEW SOUTH WALES
MAY 2018

CERTIFICATE OF ORIGINAL AUTHORSHIP

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Student: Production Note:
Signature removed prior to publication.

Date: 12 Aug 2018

ALL PUBLICATIONS DURING THE CANDIDATURE

Refereed Journal Publications

1. Yan Yan, Feiping Nie, Wen Li, Chenqiang Gao, Yi Yang, Dong Xu. "Image Classification by Cross-Media Active Learning with Privileged Information." *IEEE Transactions on Multimedia* 18, no. 12 (2016): 2494-2502.

2. Yahong Han, Yi Yang, Yan Yan, Zhigang Ma, Nicu Sebe and Xiaofang Zhou. "Semi-Supervised Feature Selection via Spline Regression for Video Semantic Recognition." *IEEE Transactions on Neural Networks and Learning Systems* 26, no. 2 (2015): 252-264.

3. Xingzhong Du, Yan Yan, Pingbo Pan, Guodong Long and Lei Zhao. "Multiple Graph Unsupervised Feature Selection." *Signal Processing* 120 (2016): 754-760.

4. Yan Yan, Gaowen Liu, Sen Wang, Jian Zhang and Kai Zheng. "Graph-Based Clustering and Ranking for Diversified Image Search." *Multimedia Systems* 23, no. 1 (2017): 41-52.

Refereed Conference Publications

1. Xuanyi Dong, Yan Yan, Wanli Ouyang and Yi Yang. Style Aggregated Network for Facial Landmark Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2018.

2. Yu Wu, Yutian Lin, Xuanyi Dong, Yan Yan, Wanli Ouyang and Yi Yang. Exploit the Unknown Gradually: One-Shot Video-Based Person Re-Identification by Stepwise Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2018.

3. Yan Yan, Tianbao Yang, Yi Yang, Jianhui Chen. A Framework of Online Learning with Imbalanced Streaming Data. In *Thirty-First AAAI Conference on Artificial Intelligence (AAAI)* 2017.

4. Yi Xu*, Yan Yan*, Qihang Lin, Tianbao Yang. Homotopy Smoothing for Non-Smooth Problems with Lower Complexity than O(1/ε). In *Neural Information Processing Systems (NIPS)* 2016.

5. Yan Yan, Zhongwen Xu, Ivor W. Tsang, Guodong Long, Yi Yang. Robust Semi-supervised Learning through Label Aggregation. In *Thirtieth Conference on Artificial Intelligence (AAAI)* 2016.

6. Mingkui Tan, Yan Yan, Li Wang, Anton Van Den Hengel, Ivor W. Tsang, Qinfeng (Javen) Shi. Learning Sparse Confidence-Weighted classifier on Very High Dimensional Data. In *Thirtieth Conference on Artificial Intelligence (AAAI)* 2016.

7. Yan Yan, Mingkui Tan, Ivor W. Tsang, Yi Yang, Chengqi Zhang and Qinfeng (Javen) Shi. Scalable Maximum Margin Matrix Factorization by Active Riemannian Subspace Search. In *International Joint Conference on Artificial Intelligence (IJCAI)* 2015, 3988-3994.

8. Joey Tianyi Zhou, Sinno Jialin Pan, Ivor W. Tsang and Yan Yan. Hybrid Heterogeneous Transfer Learning through Deep Learning. In *Twenty-Eighth Conference on Artificial Intelligence (AAAI)* 2014, 2213-2219.

Thesis advisor: Professor Yi Yang Yan Yan

Large-scale Machine Learning Algorithms for Big Data

Abstract

Machine learning is a research area in artificial intelligence which aims to learn a model from data. On one hand, the target is to learn a model yielding superior performance. On the other hand, as the rapid increase of the size of the collected data, there emerges a demand for machine learning algorithms to deal with large-scale problems.

Recent years have witnessed a sharp increase of the scale of the collected data. Taking recommender systems as an example, the Yahoo Music dataset includes more than 262 million ratings. In image classification, Imagenet contains more than 100 million images from the Internet. Such a large scale brings a great challenge to machine learning algorithms: how could the machine learning algorithms achieve satisfactory performance with less computational cost? In this dissertation, I mainly focus on several specific machine learning tasks and their scalability issues in either computation or storage aspects.

Computational cost plays a crucial role in machine learning algorithms. For instance, iteration complexity is a commonly-used theoretical metric to evaluate how fast an optimization algorithm converges. An example is the full singular value decompositions (SVDs) in the nuclear norm minimization for low-rank matrix completion. Its computational complexity can be $O(n^3)$ where n is the size of the matrix. It would be computationally unfordable when n scales up.

Memory cost is also a typical concern in machine learning. Recently deep neural networks have captured much attention and been successfully applied to a variety of applications. These deep models are known to be hungry for data, so training them usually requires a large number of training samples. When the entire training set cannot be loaded into the memory simultane-

ously, online (stochastic) learning can be applied. In such a memory-restricted scenario, both theoretical analysis and empirical investigation are expected.

Targeting on the above two aspects in large-scale machine learning tasks, in this dissertation, I investigate a variety of machine learning tasks and analyze their specific characteristics. Specifically, I mainly focus on four tasks, i.e., matrix factorization for ordinal ratings, semi-supervised learning, active learning for image classification, online learning for imbalanced streaming data. For the first three tasks, I analyze the specific characteristics of the underlying problems and design new algorithm to optimize the objective. Theoretical verification such as computational complexity is provided. For the last task, I propose an online learning algorithm to deal with imbalanced problems under the strict memory constraint.

Contents

Listing of figures

This thesis is dedicated to my beloved parents and grandparents.

Acknowledgments

First and foremost, I would like to thank my supervisor Professor Yi Yang. I feel extremely fortunate to work with Professor Yi Yang and have learned and progressed so much since the first day when I arrived in Australia. He guided me to investigate a variety of exciting research projects and always encouraged me to look for my own research interests. More importantly, he made me think critically and independently, about research, career and even life. Without this great advice and help during the recent years, it is not possible for me to imagine where I would be and what I would be doing right now. I am sure that I will keep progress under his deep and valuable guidance in the future.

I would like to thank Professor Ivor W. Tsang, Professor Mingkui Tan, Professor Tianbao Yang, Professor Feiping Nie, Professor Dong Xu and Dr Wen Li. I enjoyed the discussions with them. They were always patient to answer my questions. I also learned very much from them, and gradually gained the domain knowledge of many attractive research areas. Their help and advice guided me to find much more interests during my research. The guidance made me increasingly more enthusiastic and eager to contribute more time and energy to my future research.

I want to thank my colleagues and friends in University of Technology Sydney and the University of Iowa. Especially, I would like to thank Zhongwen Xu, Xiaojun Chang, Linchao Zhu, Pingbo Pan, Bo Han, Guoliang Kang, Liang Zheng, Yanbin Liu, Xuanyi Dong, Hu Zhang, Zhun Zhong, Yu Wu, Zhedong Zheng, Yi Xu, Zhe Li, Mingrui Liu, Zaiyi Chen, etc. I was for-

tunate to learn much from them and participate in the wonderful seminars with them. I acquire much knowledges and skills from them, not only the fields that I am working on, but also the areas that I have never touched.

0 Introduction

0.1 Background of Machine Learning

Machine learning plays a crucial role in artificial intelligence. It concentrates on induction or other types of algorithms that take as input specific training instances and *learn* a model that generalizes beyond these training data [70, 110]. This term was first used by Arthur Samuel when he was at IBM [70].

Depending on different tasks, machine learning algorithms can be categorized into two main types, i.e., supervised learning [125] and unsupervised learning [49]. The key difference between the two types of algorithms lies on whether there is learning feedbacks. In the supervised learning category, based on how much supervision can be provided, one can also classify supervised

learning algorithms into sub-groups. 1) Supervised learning [125]: the supervision feedbacks are completely available. 2) Semi-supervised learning [159]: the supervision feedbacks are incompletely available. 3) Active learning [108]: Limited supervision is initially available. The algorithm is able to actively choose unlabeled instances and interact with human users to provide supervision. 4) Reinforcement learning [119]: learning feedbacks are provided once the algorithms make an action in a dynamic environment. In unsupervised learning [49], learning feedbacks are unavailable.

Recent years have witnessed the increasing prevalence of machine learning in many real-world applications [134, 61, 117, 118][155, 86, 158, 28, 7, 75, 128, 9, 86, 65, 111, 116]. In recommender systems, for example, collaborative filtering is a widely-used machine learning approach to predict the potential preferences of users based on the existing ratings. Typically, a collaborative filtering model takes the ratings from users on products as inputs. These ratings compose the observed elements of a rating matrix, whose two dimensions present users and products, respectively. The aim of collaborative filtering is to complete the missing elements in this matrix. In machine learning, this tasks can be modeled as low-rank matrix completion. The assumption is the underlying low-rank structure of the rating matrix, which comes from the fact that a user tends to have the similar taste on a product with another user, if their preferences agree on many other products.

In computer vision tasks, machine learning algorithms have been widely applied [71, 99, 41, 74, 152, 63, 62]. For instance, Imagenet, a large-scale image datasets for classification, detection and segmentation, requires intensive human labor to provide the ground-truth annotations. Active learning can be used to decrease the human labor for labeling the training data. By exploiting an initial labeled training set, it produces a ranking list for unlabeled data. This list predicts the potential benefit if users provide the label for unlabeled data. Naturally, the best choice is to find the most beneficial unlabeled data for user-labeling, which is the target of active learning.

Similar issues happen in person re-identification (REID) [139]. Person REID requires temporal and spatial annotations of multiple subjects in a cross-camera scenario. It is usually expensive

to acquire such cross-camera annotations. There has been recently an expectation in the person REID community that one can also learn a sufficiently good model with only a few training instances and abundant unlabeled data. This task can be viewed as semi-supervised learning.

There are many real-world applications that can be coped with machine learning algorithms. In this dissertation, I mainly consider four scenarios where machine learning algorithms are applied. I summarized them in Section 0.3.

0.2 Machine Learning Algorithms for Large-Scale Data

Due to the advances of the Internet, recent years have seen a dramatically increase of the scale of the collected data. Here I take a number of examples. In recommender systems, many released datasets contain millions of ratings. In the Netfix dataset, the rating matrix consists of 17,770 movies, 480,189 users and totally 100,480,507 ratings. In the Yahoo! Music Track 1 dataset, the rating matrix consists of 624,961 music products, 1,000,990 users and 262,810,175 ratings. In computer vision tasks, the scale of data has also a sharp increase. In Large Scale Visual Recognition Challenge 2012 (ILSVRC2012), the training set includes 10,000,000 hand-labeled images depicting more than 10,000 object categories. Such large scale of training data imposes two main challenges, i.e., high computational cost and high memory expense.

Computational cost plays a vital role when analyzing a machine learning algorithm. For instance, a typical and theoretical way to measure the speed of an optimization algorithm is the iteration complexity (or convergence rate). It is usually a function of ε, the accuracy of the achieved solution to the optimal solution. When the required accuracy is sufficiently small, e.g., $\varepsilon = 10^{-4}$, there can be a huge gap of the computational cost between two iteration complexity of $O(1/\sqrt{epsilon})$ and $O(1/\varepsilon)$. The gap will be very significant especially when the scale of the training data is large.

Another factor that heavily influences the computational cost is the computational complexity of the necessary operations. Nuclear norm minimization is a convex relaxation of the rank minimization problems, but the computational complexity of singular value decompositions (SVDs) for a n by n full matrix can be $O(n^3)$. It can be computationally unaffordable if n is

large.

Memory cost is another important restriction on large-scale data. Deep learning has been successfully applied to a wide range of real-world applications, including computer vision, natural language processing and information retrieval [72, 55, 112]. Suppose that one would like to train a convolutional neural networks (CNNs) on the entire Imagenet dataset. It is not practical to load all the images into memory and perform back-propagation to train CNNs. A solution is to use the *stochastic* method to update the model, which is exactly what most researchers are doing when training CNNs. Therefore, memory cost is sometimes a critical restriction of machine learning algorithms on large-scale data. Online (stochastic) algorithms can be applied to deal with the case where only a (batch of) data instance(s) are used to update the model. It would be interesting to investigate both the theoretical analysis and empirical performance of these algorithms.

0.3 Considered Machine Learning Tasks

This section summarizes a number of realistic machine learning tasks, i.e., matrix completion by maximum margin matrix factorization, semi-supervised learning by label aggregation, active learning for image classification by privileged information and online learning for imbalanced data, that are considered to deal with in this dissertation. For each of the specific tasks, I would like to provide a brief introduction to each of their backgrounds first. Then I directly indicate and analyze the main challenges that the existing approaches are faced with. Lastly, I present the general ideas to deal with the large-scale issue. The detailed solutions to these tasks are presented in the subsequent Chapters. Specifically, the proposed algorithms for maximum margin matrix factorization, semi-supervised learning, active learning and online learning are introduced in Chapter 2 [147], Chapter 3 [148], Chapter 4 [146] and Chapter 5 [149], respectively.

0.3.1 Maximum Margin Matrix Factorization for Ordinal Ratings

The rapid increase of Web services has witnessed an increasing demand for predicting the preferences of users on products of interest, such as movies and music tracks [118]. This task, also

known as the collaborative filtering (CF), is a principal task in recommender systems [134, 61]. In general, the user ratings are given in discrete values, including binary ratings and ordinal ratings [117]. The binary ratings can be either "+1" (*like*) or "-1" (*dislike*); while the ordinal ratings are in discrete values such as 1-5 "stars", which are more popular in applications.

Given a small number of user ratings $\mathbf{Y} \in \mathbb{R}^{m \times n}$ (from m users on n items), the aim of CF is to reconstruct the unobserved ratings. Let Ω be a subset containing the indices of the observed entries. To perform the reconstruction, a common approach is to learn a low-rank matrix $\mathbf{X}$ to fit $\mathbf{Y}$ by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{X}} \quad & f(\mathbf{X}) \\ s.t. \quad & \text{rank}(\mathbf{X}) \leq k, \end{aligned} \tag{1}$$

where k denotes the number of latent factors (i.e. the rank of $\mathbf{X}$) and $f(\mathbf{X})$ denotes some loss functions. The low-rank property has been studied in a variety of applications [142, 141, 151]. In many studies, such as matrix completion, the least-square loss function $f(\mathbf{X}) = \sum_{ij \in \Omega} (\mathbf{X}_{ij} - \mathbf{Y}_{ij})^2$ is used [21, 20, 124]. Despite of its popularity, the least-square loss may not perform well when the ratings are discrete values [117].

To deal with rating data, the maximum margin matrix factorization (M^{3}F) is proposed using the hinge loss [117, 102, 134]. For binary ratings, the objective function can be written as

$$\min_{\mathbf{X}} \quad f(\mathbf{X}) = \min_{\mathbf{X}} \quad \sum_{ij \in \Omega} h(\mathbf{Y}_{ij} \mathbf{X}_{ij}), \tag{2}$$

where $h(z) = \max(0, 1 - z)$. The hinge loss $h(z)$ for binary ratings can be easily extended to general ordinal ratings where $\mathbf{Y}_{ij} \in \{1, 2, \ldots L\}$ by applying $L + 1$ thresholds $\vartheta_0 \leq \vartheta_1 \leq \ldots \leq \vartheta_L$ learned from data [102]. For the discrete valued rating data, hinge loss would achieve better performance compared to the least square loss.

Problem (1) is known to be NP-hard. Many researchers [44, 100] thus propose to solve its nuclear-norm convex relaxation $\min_{\mathbf{X}} \lambda ||\mathbf{X}||_* + f(\mathbf{X})$, where $||\mathbf{X}||_*$ denotes the nuclear norm of $\mathbf{X}$ and λ is a regularization parameter. Many convex optimization methods, such as proximal

gradient methods [121, 96] can be adopted to solve this problem. However, these methods may scale poorly due to the requirement of singular value decompositions (SVDs) of large ranks.

To improve the scalability, some researchers assumes that the rank of $\mathbf{X}$ (i.e. k) is known, and $\mathbf{X}$ can be explicitly factorized as $\mathbf{X} = \mathbf{U}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{n\times k}$ and $\mathbf{V} \in \mathbb{R}^{m\times k}$ [102, 92]. They then solve the following variational formulation instead:

$$\min_{\mathbf{U},\mathbf{V}} \quad \frac{\lambda}{2}(||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2) + f(\mathbf{U}\mathbf{V}^\top), \tag{3}$$

where λ is a regularization parameter. Many methods, such as the stochastic gradient descent (SGD), can be used to solve this problem. However, in real applications, the prior knowledge about k is not likely accessible. Consequently, these algorithms may have to perform expensive model selections to determine k, which is unaffordable in computation [143, 144]. Additionally, since problem (3) is non-convex w.r.t. $\mathbf{U}$ and $\mathbf{V}$ simultaneously, most methods may face the premature convergence problem [59].

Regarding the scalability issue and the latent factor detection issue of existing methods, in Chapter 2 [147], I propose an active Riemannian subspace search for M^{3}F (ARSS-M^{3}F). The main contributions of this chapter are as follows:

- Leveraging the nonlinear Riemannian conjugate gradient, I propose an efficient block-wise nonlinear Riemannian conjugate gradient (BNRCG) algorithm, which reconstructs $\mathbf{X}$ and learns multiple thresholds ϑ in M^{3}F in a joint framework. Compared to existing M3F algorithms, the proposed algorithm is much more efficient.
- Based on BNRCG, I proposed the ARSS-M^{3}F method which applies a simple and efficient pursuit scheme to automatically compute the number of latent factors, which avoids expensive model selections.
- Extensive experiments on both synthetic data sets and real-world data sets demonstrate the superior efficiency and effectiveness of the proposed methods.

0.3.2 Semi-Supervised Learning by Label Aggregation

Massive data can be easily collected from social networks and online services due to the explosion of Internet development. However, the vast majority of collected data are usually unlabeled and unstructured. Labeling a large amount of unlabeled data can be expensive. Therefore, it is natural to consider exploiting the abundance of unlabeled data to further improve the performance of algorithms. This has led to a rising demand for semi-supervised learning methods that leverage both labeled data and unlabeled data [155, 86, 158, 28].

Semi-supervised learning (SSL) is an active research area and a variety of SSL algorithms have been proposed [11, 14, 29, 115, 7, 75, 128]. However, many existing algorithms are faced with the scalability issue owing to the high complexity. For example, the complexity of LapSVM [9] is $O(n^3)$ due to the requirement for the inverse of a dense Gram matrix. TSVM in [66] treats the SVM problem as a sub-problem and infers the labels of unlabeled data via a label switch procedure, which may lead to a large number of iterations.

In addition to the scalability issue, SSL algorithms may suffer from label noise, leading to unreliable performance. In the SSL setting, there are usually only small amount of labeled data and a large proportion of unlabeled data. Even small mistakes in the human (non-expert) annotation process are likely to result in label noise. Thus robustness is particularly critical for SSL methods in many applications [86, 65].

In this dissertation, I focuses on the two aforementioned challenges of SSL, i.e. scalability and robustness. Inspired by crowdsourcing [111, 116], in Chapter 3 [148], I propose an efficient RObust Semi-Supervised Ensemble Learning (ROSSEL) method to approximate ground-truth labels of unlabeled data through aggregating a number of pseudo-labels generated by low-cost *weak annotators*, such as linear SVM classifiers. Meanwhile, based on the aggregated labels, ROSSEL learns an inductive SSL classifier by Multiple Label Kernel Learning (MLKL) [77]. Unlike most existing SSL algorithms, the proposed ROSSEL requires neither expensive graph Laplacian nor iterative label switching. Instead, it only needs *one* iteration for label aggregation and can be solved by an SVM solver very efficiently. The major contributions are listed as fol-

lows,

- Leveraging an ensemble of low-cost supervised weak annotators, I propose ROSSEL to efficiently obtain a weighted combination of pseudo-labels of unlabeled data to approximate ground-truth labels to assist semi-supervised learning.
- Instead of simple label aggregation strategies used in crowdsourcing (*e.g.*majority voting), ROSSEL performs a weighted label aggregation using MLKL. Meanwhile it learns an inductive SSL classifier, which only requires *one* iteration and linear time complexity w.r.t. number of data and features.
- Complexity analysis of several competing SSL methods and the proposed method is provided.

0.3.3 Active Learning for Image Classification by Privileged Information

With the advance of network technology and web services, numerous photos are uploaded to the Internet every day, which makes the Internet becomes a huge repository of images. Therefore, collecting web images as the training data has become a popular way to learn models for image classification [71, 99, 41, 74]. Labeling large scale images is time consuming and labor intensive. A more practical way is to actively sample and label a small subset of training images which are the most informative [106, 45, 122, 33].

In Chapter 4, I propose a novel active sample selection approach (*a.k.a.* active learning) for image classification by using web images. Previous research has shown that cross-media modeling of various media types is beneficial for multimedia content analysis [154, 136, 130, 151, 25]. The web images are often associated with rich textual descriptions (*e.g.*, surrounding texts, captions, *etc*). While such text information is not available in testing images, I show that text features are useful for learning robust classifiers, enabling better active learning performance of image classification. Typical active sampling methods only deal with one media type [56, 63, 62, 152, 93], which cannot simultaneously utilize different media types. The new supervised learning paradigm, namely learning using privileged information (LUPI), can be used to solve

this problem [126, 74]. In a LUPI scenario, in addition to main features, there is also privileged information available in the training procedure. Privileged information can only be used in training, and is not available in testing.

Uncertainty sampling is the most frequently used strategy in the active learning [152]. In this work, I propose to exploit both visual and text features for active sample selection by taking text as privileged information. By LUPI, I train SVMs on visual features and slacking function on text features. I present five strategies to combine the uncertainty measure of these two classifiers.

To ensure the selected samples to be representative, in Chapter 4 [146], I exploit the diversity measurement, such that the selected samples are less similar to each other. I formulate a ratio objective function to maximize cross-media uncertainty and minimize the similarity of selected data. Then I propose to measure uncertainty and diversity for training sample selection [152]. A new optimization method is proposed to solve the proposed model, which automatically learns the optimal ratio of uncertainty to similarity. In this way, I avoid introducing the trade-off parameter between the two types of measurements. I summarize the main contributions of this work as follows:

- By exploiting privileged information, I propose a new notion of cross-media uncertainty measurement, which measures the uncertainty of unlabeled images by jointly considering visual features as the main information and text features as the privileged information.

- I propose a new method to optimize the objective without using the trade-off parameter between diversity and uncertainty.

0.3.4 Online Learning for Imbalanced Data

Streaming data are pervasive in many domains, including online social media [87, 3], clickbait prediction [13], ad placement [84], *etc.*. In these scenarios, data are coming sequentially. Mining the streaming data requires the learner to make a prediction instantly after receiving an example and update the model based on the received true label. As the increasing popularity of stream-

ing data, it becomes critical to design effective learning algorithms for mining streaming data and making accurate predictions on the fly.

Online learning has emerged to be an important learning paradigm due to its ability to handle streaming data. Different from traditional batch learning, in online learning, data arrive sequentially, and the prediction is made before getting a feedback about the true label. Thus, the online performance of a learner is a critical concern in online learning, since it measures how much the predictions are consistent with the true label.

In most existing studies of online learning, a challenge for mining large-scale streaming data is that examples are usually skew-distributed over different classes. Particularly for binary problems, the number of positive examples is usually significantly smaller than that of negative ones in many applications. Therefore, the zero-one loss and its surrogates commonly used in traditional online learning algorithms are not appropriate for imbalanced data. This issue has been long recognized as cost asymmetry, i.e., the cost for a false negative should be different from that for a false positive. To deal with it, cost-sensitive algorithms, one of the most popular approaches for tackling imbalanced data, have been recently studied in the online setting [129], which usually assign fixed costs, or ad-hoc costs based on the distribution of data received so far to different classes. However, it would not necessarily achieve superior performance measures including F-measure, area under ROC curve (AUROC), area under precision and recall curve (AUPRC).

Another line of research for learning with the imbalanced streaming data is to directly optimize target measures in an online fashion, which attracts increasing attention recently [47, 156]. However, there are two main limitations. Firstly, measures applied in imbalanced problems, *e.g.*, F-measure, AUROC and AUPRC, are usually not decomposable, which makes it significantly challenging to directly optimize these measures in the online setting. Moreover, an algorithm designed for optimizing a specific measure (*e.g.*, F-measure) is usually not applicable for optimizing another certain measure (*e.g.*, AUROC).

To address these issues, in Chapter 5 [149], I present a unified framework for learning with imbalanced streaming data that is easily adapted to different performance measures. The pro-

posed framework simultaneously learns multiple classifiers with various cost vectors. In particular, at each iteration, the prediction is made by a classifier which is selected randomly according to a sampling distribution, which is updated based on the current performance measures of classifiers, similarly to the well-know exponential weighted average algorithm [83]. The selection of the optimal classifier is adaptive and evolving according to the streaming data. I would like to emphasize that the proposed approach is different from the cross-validation approach, which replies on a separate validation set. Furthermore, the proposed framework enjoys a rigorous theoretical justification for the F-measure maximization. Empirical studies demonstrate that the proposed algorithm is more effective than previous online learning algorithms for imbalanced streaming data.

1

Literature Review

In this dissertation, I investigate the possibility of machine learning algorithms to decrease the computational and memory cost. It is difficult to propose a general approach to achieving this target. Therefore, I analyze a number of specific machine learning tasks and design new methods to solve the original problems. Below I summarize a literature survey on the considered tasks respectively.

1.1 Matrix Completion by Maximum Margin Matrix Factorization

The maximum margin matrix factorization (M^3F) problem can be formulated as a semi-definite programming (SDP) problem, thus it can be solved using standard SDP solvers [117]. However,

the SDP solver scales very poorly. To improve the scalability, a fast M^{3}F method is proposed to solve problem (3) by investigating the gradient-based optimization method [102]. A low-rank matrix fitting algorithm (LMAFIT) is proposed to solve (3) with the least square loss [135]. More recently, a lock-free approach to parallelizing stochastic gradient descent is proposed [101]. However, it is nontrivial for them to solve M^{3}F.

Note that the fixed-rank matrices belong to a smooth matrix manifold [1, 124]. Manifold has been also exploited in a range of applications [26, 51, 85]. Many manifold optimization methods have been proposed to solve (3) [90, 15, 124], such as the Riemannian trust-region method for MC (RTRMC) [15], the low-rank geometric conjugate gradient method (LRGeomCG) [124], the quotient geometric matrix completion method (qGeomMC) [91], Grassmannian rank-one update subspace estimation (GROUSE) and the method of scaled gradients on Grassmann manifolds for matrix completion (ScGrassMC) [94]. However, all these methods are not applicable to solve M^{3}F.

A number of M^{3}F extensions have been introduced in the last decades [134, 133, 69]. For example, the authors in [133] presented a method using M^{3}F to optimize ranking rather than ratings. Some researcher further improved the performance of M^{3}F by casting it within ensemble approaches [35, 137].

The importance of automatic latent factor detection (i.e. the model selection problem) has been recognized by many researchers [143, 144, 92]. For example, a probabilistic M^{3}F model is proposed in [143, 144], where the number of latent factors can be inferred from data. However, these methods are usually very expensive as the probabilistic model requires a large amount of computation, which is avoided in our method.

1.2 Semi-Supervised Learning by Label Aggregation

As large scale data are easily accessible, it is usually difficult to obtain sufficient supervision in practice. For instance, a feature selection algorithm is proposed in [53] for video recognition where the number of labeled videos are limited. In [46], an action recognition method is proposed which does not exploit any positive examplars. The authors in [78] propose a method to

deal with weak-label learning tasks. In this section, we focus on SSL problems.

Among SSL algorithms, graph-based methods are commonly used [28]. Many graph-based algorithms introduce the manifold structure by leveraging manifold regularization [160, 157, 8, 114, 9, 123, 113, 145, 155]. However, the complexity of building graph Laplacian is at least $O(n^2)$. Consequently, these graph-based algorithms are usually difficult to handle large scale datasets. Recently, the authors in [128] propose an adaptive SSL to optimize the weight matrix of the model and the label matrix simultaneously, which avoids expensive graph construction. There are some SSL methods exploiting pseudo-labels of unlabeled data. For instance, in [73], pseudo-labels are used to make deep neural networks able to handle unlabeled data. The authors in [4] propose to exploit pseudo-ensembles to produce models that are robust to perturbation. In [37], pseudo-labels are exploited in an image reranking framework regularized by multiple graphs. The authors in [27] formulate multi-label semi-supervised feature selection as a convex problem and propose an efficient optimization algorithm. A semi-supervised ranking and relevance feedback framework is proposed for multimedia retrieval in [153]. In [76], the authors propose a SVM-based SSL algorithm by exploiting the label mean. A cost-sensitive semi-supervised SVM is proposed in [75]. Although these methods avoid expensive graph Laplacian, they still require a number of iterations for training.

Ensemble learning is a supervised learning paradigm that trains a variety of learners on a given the training set, and derives a prediction from the votes of all its learners [38]. There are a number of most commonly used ensemble algorithms, including bagging [17], random forests [18] and boosting [103]. Bagging is one of the most commonly used ensemble algorithms, where a number of bootstrap replicates are generated on the training set by bootstrap sampling, and a learner is trained on each bootstrap replicate. Ensemble learning methods can only handle labeled data.

1.3 Active Learning for Image Classification by Privileged Information

Active learning aims to obtain better performance when learning with fewer labeled training samples by actively selecting a portion of the training data from a pool of unlabeled data [107].

Uncertainty sampling is the most frequently used approach to active sample selection [107, 122, 5], which selects queries the unlabeled data that the learner is most uncertain.

There are some other criteria in addition to uncertainty, such as diversity [152], representativeness [33, 62] and density [95, 63]. In [95], pre-clustering method was proposed to avoid repeatedly labeling samples in the same cluster, by which diversity can be introduced. The authors in [33] propose an active sampling strategy based on a hierarchical clustering of unlabeled data. However, the performance of these methods likely depends on the performance of clustering. If the result of clustering is not consistent with the target model, their active learning performance may degrade accordingly [62]. Some works consider representativeness. Representative unlabeled data are those that best represent the underlying distribution of data [62, 107]. In [62], the authors proposed an algorithm that takes both informativeness and representativeness of unlabeled data into consideration. A probabilistic variant of K-Nearest-Neighbor is used to extend active learning when the number of classes is large [63].

In a multi-view scenario, each sample is represented by multiple features. It is assumed that a concept is possible to learn from a single feature type [93, 2]. A web page on Wikipedia, for example, may contain various types of features, including images and texts. Co-testing is studied in [93]. It queries the samples that cause disagreement of the learners from various views, which are named contention points [93]. The motivation of co-testing is that at least one learner can lead to improvement from the queried data. A combination of multi-view active sampling and semi-supervised learning is proposed in [132].

Learning using privileged information (LUPI) is proposed in [126]. Compared to conventional multi-view learning, in the LUPI scenario, privileged information is only available as auxiliary features in the training process rather than the testing process. LUPI has shown promising results in many works. Various types of privileged information can be exploited to assist learning. Image attributes can be used as middle-level semantic features bridging the gap between visual features and high-level object classes [52]. Textual descriptions, which are rather abundant particularly for Web data, are frequently leveraged in classification tasks [74] and retrieval tasks [30]. In contrast to the traditional computer vision tasks such as image classification, the

authors in [81] proposed a new framework by infering knowledge in the multimedia domain from the semantic domain.

1.4 Online Learning for Imbalanced Data

In traditional online learning, studies revolve around the regret analysis of algorithms for sequential prediction problems (*e.g.*, prediction with expert advice, online classification) [23, 57, 82]. In these studies, many online algorithms have been developed, *e.g.*, the exponentially weighted average algorithm [83] and the online gradient descent [161]. In the last ten years, we observe substantial applications of these algorithms in machine learning and data analytics, *e.g.*, online classification [48, 32].

Learning with cost asymmetry has attracted much attention recently. Most studies cast the problem into cost-sensitive learning that assigns different costs to mistakes of different classes [42, 89, 104]. While there exist a long list of literatures on batch learning with cost-sensitivity, few studies were devoted to online learning with cost-sensitivity [31, 129]. These studies assume a given cost vector (or matrix) and modify conventional loss functions to incorporate the given cost vector/matrix. The issue with this approach is that the cost vector/matrix is usually unknown when applying to imbalanced data. Recent studies have found that the optimal costs assigned to different classes have an explicit relationship with the optimal performance measure [97]. Besides the cost-sensitive approach, some resampling based methods are proposed to deal with imbalanced data. However, most of them focus on batch learning, *e.g.*, [79], while there are a few works concerning the online setting, *e.g.*, [131].

Recently, there emerge some works about online optimization for a particular performance measure, *e.g.*, F-measure, AUROC. For example, [156, 47] proposed online learning algorithms for AUROC optimization. However, both works focus on the offline performance evaluation. In [19], the authors proposed an online learning algorithm for F-measure optimization with an automatic threshoding strategy based on the online F-measure. However, they innocently ignored the strategy for updating the model by simply assuming a given algorithm that can learn the posterior probability $\Pr(y|\mathbf{x})$. In [60], a method is proposed to directly optimize AUROC,

but requires extra resources to store the learned support vectors. The authors in [68] proposed an online learning framework for non-decomposable loss functions based on the structural SVM. The drawback of this method is that their online learning algorithm needs to solve a difficult optimization problem at each iteration. As for AUPRC, there still lacks of efforts. Recent studies [50, 34] have found that when dealing with highly skewed datasets, Precision-Recall (PR) curves might give a more informative picture of an algorithm's performance, which gives the measure of AUPRC.

Finally, we note that the proposed algorithm is different from online Bayesian learning that maintains and updates the posterior distribution of model parameters [39], and is also different from the online ensemble algorithm in [127] that aggregates all classifiers for prediction. The synthesis of online gradient descent for updating individual classifiers and the exponential weighted average algorithm for updating probabilities is similar to the work of online kernel selection [150]. However, the two work have different focuses. In particular, their goal is to select the best kernel classifier among multiple kernel classifiers for optimizing traditional measures while our goal is to select the best cost-sensitive classifier among multiple cost-sensitive classifiers for optimizing a target measure suited for imbalanced data. Therefore, their analysis can not be borrowed for our purpose.

2

Matrix Completion by Maximum Margin Matrix Factorization

2.1 Introduction

Targeting on the scalability issue and the latent factor detection issue of existing methods for maximum margin matrix factorization, in this chapter, I propose an active Riemannian subspace search for M^{3}F (ARSS-M^{3}F). The main contributions of this chapter [*] are as follows:

[*] The main results of this chapter were previously published in Yan Yan, Mingkui Tan, Ivor W. Tsang, Yi Yang, Chengqi Zhang and Qinfeng (Javen) Shi. Scalable Maximum Margin Matrix Factorization by Active Riemannian Subspace Search. In *International Joint Conference on Artificial Intelligence (IJCAI)* 2015, 3988-3994.

- Leveraging the nonlinear Riemannian conjugate gradient, I propose an efficient block-wise nonlinear Riemannian conjugate gradient (BNRCG) algorithm, which reconstructs $\mathbf{X}$ and learns multiple thresholds ϑ in M^{3}F in a joint framework. Compared to existing M3F algorithms, the proposed algorithm is much more efficient.
- Based on BNRCG, I proposed the ARSS-M^{3}F method which applies a simple and efficient pursuit scheme to automatically compute the number of latent factors, which avoids expensive model selections.
- Extensive experiments on both synthetic data sets and real-world data sets demonstrate the superior efficiency and effectiveness of the proposed methods.

2.2 M^{3}F on Fixed-rank Manifold

Without loss of generality, I first study M^{3}F where the rank of the rating matrix $\mathbf{X}$ to be recovered is known. I propose the BNRCG method by exploiting the Riemannian geometries to address it.

2.2.1 Notations

Throughout the chapter, I denote by the superscript $\top$ the transpose of a vector/matrix, $\mathbf{0}$ a vector/matrix with all zeros, $\mathrm{diag}(\mathbf{v})$ a diagonal matrix with a vector of diagonal entries equal to $\mathbf{v}$. Let $\mathbf{A} \odot \mathbf{B}$ and $\langle \mathbf{A}, \mathbf{B} \rangle = tr(\mathbf{A}\mathbf{B}^\top)$ represent the element-wise product and inner product of the matrices $\mathbf{A}$ and $\mathbf{B}$, respectively. The singular value decomposition (SVD) of matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ is given by $\mathbf{X} = \mathbf{U}(\mathrm{diag}(\sigma))\mathbf{V}^\top$. Based on the SVD, the nuclear norm (or trace-norm) of $\mathbf{X}$ is defined as $\|\mathbf{X}\|_* = \|\sigma\|_1 = \sum_i |\sigma_i|$, and the Frobenius norm of $\mathbf{X}$ is defined as $\|\mathbf{X}\|_F = \|\sigma\|_2$.

2.2.2 The Proposed Model

In collaborative filtering tasks, the preference scores are often ordinal ratings, where $\mathbf{Y}_{ij} \in \{1, 2, \ldots L\}$. To generalize the hinge loss for binary case to ordinal ratings, I introduce $L + 1$

thresholds $\vartheta_0 \leq \vartheta_1 \leq ..., \leq \vartheta_L$. By default, I have $\vartheta_0 = -\infty$ and $\vartheta_L = +\infty$. Therefore, there are $L-1$ free threshold parameters to be determined, namely $\vartheta = [\vartheta_1, \vartheta_2, ..., \vartheta_{L-1}]^\intercal \in \mathbb{R}^{L-1}$. In a hard-margin case, $\mathbf{X}$ must satisfy the following conditions on observed entries

$$\vartheta_{\mathbf{Y}_{ij}-1} + 1 \leq \mathbf{X}_{ij} \leq \vartheta_{\mathbf{Y}_{ij}} - 1.$$

In a soft-margin setting, the hinge loss error for each entry of $\mathbf{X}$ can be written as

$$\xi_{ij} = \sum_{z=1}^{L-1} h(T_{ij}^z \cdot (\vartheta_z - \mathbf{X}_{ij})), \forall ij \in \Omega, \tag{2.1}$$

where $T_{ij}^z = \begin{cases} +1 & \text{for} \quad z \geq Y_{ij} \\ -1 & \text{for} \quad z < Y_{ij} \end{cases}$ and $h(z) = \max(0, 1-z)$.

Principally, I propose to reconstruct $\mathbf{X}$ by minimizing the squared hinge loss error

$$\ell(\mathbf{X}, \vartheta) = \frac{1}{2} \sum_{ij \in \Omega} \xi_{ij}^2.$$

Additionally, to prevent from over-fitting, I regularize $\ell(\mathbf{X}, \vartheta)$ by a regularizer $\Upsilon(\mathbf{X}) = \frac{1}{2}(||\mathbf{X}||_F^2 + \nu||\mathbf{X}^\dagger||_F^2)$, where $\mathbf{X}^\dagger$ denotes the pseudo-inverse and $\nu > 0$ is a small scalar (e.g., $\nu = 0.0001$ in this chapter by default) and $||\mathbf{X}^\dagger||_F^2$ is a barrier to avoid decreasing of the rank of $\mathbf{X}$ [124]. The M^{3}F problem is formulated as the following optimization problem

$$\min_{\mathbf{X}, \vartheta} f(\mathbf{X}, \vartheta), \quad \text{s.t.} \quad \text{rank}(\mathbf{X}) = k, \tag{2.2}$$

where $f(\mathbf{X}, \vartheta) = \lambda\Upsilon(\mathbf{X}) + \ell(\mathbf{X}, \vartheta)$ and $0 < \lambda < 1$ denotes the regularization parameter. Note that this regularizer is different from that used in [124], and it is very important for preventing from the over-fitting issue in the context of M^{3}F (see more details in experimental studies).

After addressing problem (2.2), the prediction can be easily made by

$$\mathbf{Y}_{ij}^{*} = \max\{z|\mathbf{X}_{ij} \geq \vartheta_z, z = 1, ..., L\}. \tag{2.3}$$

Unfortunately, since $f(\mathbf{X}, \vartheta)$ is non-convex due to the constraint $\text{rank}(\mathbf{X}) = k$, the optimization of (2.2) is very difficult. Noting $\mathbf{X}$ is restricted on fixed-rank matrices, I accordingly propose to address it by exploiting the Riemannian geometries on fixed-rank matrices.

2.2.3 Riemannian Geometry of Fixed-rank Matrices

Suppose $\text{rank}(\mathbf{X}) = r$ with r being known, then $\mathbf{X}$ lies on a smooth manifold of fixed rank-r matrices [124], which is defined as

$$\begin{aligned} \mathcal{M}_r &= \{\mathbf{X} \in \mathbb{R}^{m\times n} : \text{rank}(\mathbf{X}) = r\} \\ &= \{\mathbf{U}\text{diag}(\sigma)\mathbf{V}^{\mathsf{T}} : \mathbf{U} \in \text{St}_r^m, \mathbf{V} \in \text{St}_r^n, ||\sigma||_0 = r\} \end{aligned}$$

with $\text{St}_r^m = \{\mathbf{U} \in \mathbb{R}^{m\times r} : \mathbf{U}^{\mathsf{T}}\mathbf{U} = \mathbf{I}\}$ the Stiefel manifold of $m \times r$ real and orthonormal matrices. The tangent space $T_{\mathbf{X}}\mathcal{M}_r$ of $\mathcal{M}_r$ at $\mathbf{X} = \mathbf{U}\text{diag}(\sigma)\mathbf{V}^{\mathsf{T}} \in \mathbb{R}^{m\times n}$ is given by

$$\begin{aligned} T_{\mathbf{X}}\mathcal{M}_r = \{&\mathbf{U}\mathbf{M}\mathbf{V}^{\mathsf{T}} + \mathbf{U}_p\mathbf{V}^{\mathsf{T}} + \mathbf{U}\mathbf{V}_p^{\mathsf{T}} : \mathbf{M} \in \mathbb{R}^{r\times r}, \\ &\mathbf{U}_p \in \mathbb{R}^{m\times r}, \mathbf{U}_p^{\mathsf{T}}\mathbf{U} = 0, \mathbf{V}_p \in \mathbb{R}^{n\times r}, \mathbf{V}_p^{\mathsf{T}}\mathbf{V} = 0\}. \end{aligned} \tag{2.4}$$

By defining a metric $g_{\mathbf{X}}(\mathbf{A}, \mathbf{B}) = \langle \mathbf{A}, \mathbf{B}\rangle$ on $\mathcal{M}_r$, where $\mathbf{X} \in \mathcal{M}_r$ and $\mathbf{A}, \mathbf{B} \in T_{\mathbf{X}}\mathcal{M}_r$, then $\mathcal{M}_r$ becomes a Riemannian manifold by restricting $\langle \mathbf{A}, \mathbf{B}\rangle$ to the *tangent bundle*, which is defined as the disjoint union of all tangent spaces $T\mathcal{M}_r = \bigcup_{\mathbf{X}\in\mathcal{M}_r}\{\mathbf{X}\} \times T_{\mathbf{X}}\mathcal{M}_r = \{(\mathbf{X}, \mathbf{E}) \in \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} : \mathbf{X} \in \mathcal{M}_r, \mathbf{E} \in T_{\mathbf{X}}\mathcal{M}_r\}$.

Let $\mathbf{G}$ be the gradient of any smoothing function $f(\mathbf{X})$ in Euclidian space at $\mathbf{X} = \mathbf{U}\text{diag}(\sigma)\mathbf{V}^{\mathsf{T}}$. The Riemannian gradient of $f(\mathbf{X})$ on $\mathcal{M}_r$ is given as the orthogonal projection of $\mathbf{G}$ onto the

tangent space at $\mathbf{X}$:

$$\text{grad}f(\mathbf{X}) = P_{T_{\mathbf{X}}\mathcal{M}_r}(\mathbf{G}). \tag{2.5}$$

Here $P_{T_{\mathbf{X}}\mathcal{M}_r}(\mathbf{Z}) : \mathbf{Z} \mapsto P_U\mathbf{Z}P_V + P_U^{\perp}\mathbf{Z}P_V + P_U\mathbf{Z}P_V^{\perp}$ denotes the orthogonal projection of any $\mathbf{Z} \in \mathbb{R}^{m\times n}$ onto the tangent space at $\mathbf{X} = \mathbf{U}\text{diag}(\sigma)\mathbf{V}^{\mathsf{T}}$, where $P_U = \mathbf{U}\mathbf{U}^{\mathsf{T}}$ and $P_U^{\perp} = \mathbf{I} - \mathbf{U}\mathbf{U}^{\mathsf{T}}$ for any $\mathbf{U} \in \text{St}_r^m$.

With prior knowledge about differential geometries on fixed-Rank matrices, I can compute the Riemannian gradient of $f(\mathbf{X}, \vartheta)$ w.r.t. $\mathbf{X}$ on $\mathcal{M}_r$. Let $\text{grad}f(\mathbf{X}, \vartheta)$ denote the Riemannian gradient. To compute $\text{grad}f(\mathbf{X}, \vartheta)$, I need to calculate the gradient of $f(\mathbf{X}, \vartheta)$ on Euclidean space. Firstly, the gradient of $\ell(\mathbf{X}, \vartheta)$ w.r.t. $\mathbf{X}$, denoted by $\widehat{\mathbf{G}}$, can be calculated by

$$\widehat{\mathbf{G}}_{ij} = \frac{\partial \ell(\mathbf{X}, \vartheta)}{\partial \mathbf{X}_{ij}} = \sum_{z=1}^{L-1} T_{ij}^z \cdot h(T_{ij}^z \cdot (\vartheta_z - \mathbf{X}_{ij})). \tag{2.6}$$

where $ij \in \Omega$. Note that the gradient of $\Upsilon(\mathbf{X})$ w.r.t. $\mathbf{X}$ is $\mathbf{U}\text{diag}(\sigma - \nu/\sigma^3)\mathbf{V}^{\mathsf{T}}$ at $\mathbf{X} = \mathbf{U}\text{diag}(\sigma)\mathbf{V}^{\mathsf{T}}$. The gradient of $f(\mathbf{X}, \vartheta)$ w.r.t. $\mathbf{X}$ in Euclidian space, denoted by $\mathbf{G}$, can be computed by

$$\mathbf{G} = \widehat{\mathbf{G}} + \lambda\mathbf{U}\text{diag}(\sigma - \nu/\sigma^3)\mathbf{V}^{\mathsf{T}}. \tag{2.7}$$

Once $\mathbf{G}$ is computed, $\text{grad}f(\mathbf{X}, \vartheta)$ can be calculated according to equation (2.5). The details of computation can be found in Appendix A.

Finally, the gradient of $f(\mathbf{X}, \vartheta)$ w.r.t. ϑ, denoted by $\mathbf{g} = [g_1, g_2, ..., g_{L-1}]^{\mathsf{T}}$, can be calculated by

$$g_z = \frac{\partial f(\mathbf{X}, \vartheta)}{\partial \vartheta_z} = \sum_{ij\in\Omega} -T_{ij}^z \cdot h(T_{ij}^z \cdot (\vartheta_z - \mathbf{X}_{ij}), \tag{2.8}$$

where $z \in \{1, 2, ..., L-1\}$.

2.2.4 Block-wise Nonlinear Riemannian Conjugate Gradient Descent for M³F

The objective function in (2.2) involves two types of variables, namely the rating matrix $\mathbf{X} \in \mathcal{M}_r$ and the thresholding parameter $\vartheta \in \mathbb{R}^{L-1}$. Accordingly, I propose a Block-wise Nonlinear Riemannian Conjugate Gradient (BNRCG) to solve problem (2.2), which is shown in Algorithm 1. The basic idea is that, at each iteration, I first minimize $f(\mathbf{X}, \vartheta)$ w.r.t. $\mathbf{X}$ with fixed ϑ by a Nonlinear Riemannian Conjugate Gradient method (Steps 1-3), and then minimize $f(\mathbf{X}, \vartheta)$ w.r.t. ϑ with fixed $\mathbf{X}$ by applying a standard gradient descent method (Steps 4-5). I will illustrate Steps 2-5 in details.

Algorithm 1 BNRCG for Fixed-rank M³F.

Given $\text{rank}(\mathbf{X}) = r$. Initialize $\mathbf{X}_1$, η_0, and ϑ_1. Let $t = 1$.
1: Compute $\mathbf{E}_t = -\text{grad} f(\mathbf{X}_t, \vartheta_t)$ according to (2.5).
2: Compute the conjugate direction with PR+ rule:
$\eta_t = \mathbf{E}_t + \beta_t \mathcal{T}_{\mathbf{X}_{t-1} \to \mathbf{X}_t}(\eta_{t-1}) \in T\mathcal{M}_r$.
3: Choose a step size α_t and set $\mathbf{X}_{t+1} = R_{\mathbf{X}_t}(\alpha_t \eta_t)$.
4: Compute $\mathbf{g}_t$ according to (2.8).
5: Choose a step size γ_t and set $\vartheta_{t+1} = \vartheta_t - \gamma_t \mathbf{g}_t$.
6: Quit if stopping conditions achieve.
7: Let $t = t + 1$ and go to step 1.

When updating $\mathbf{X}$, different from the classical gradient methods on Euclidean space, the search direction in manifold optimization needs to follow a path on the manifold. Let $\mathbf{X}_t$ be the iteration variable in the BNRCG method on Euclidean space, the search direction η_t is calculated by

$$\eta_t = -\text{grad} f(\mathbf{X}_t) + \beta_t \eta_{t-1}, \tag{2.9}$$

where β_t can be calculated by a Polak-Ribière (PR+) rule [124]:

$$\beta_t = \frac{\text{grad} f(\mathbf{X}_t)^\top (\text{grad} f(\mathbf{X}_t) - \text{grad} f(\mathbf{X}_{t-1}))}{\langle \text{grad} f(\mathbf{X}_{t-1}), \text{grad} f(\mathbf{X}_{t-1}) \rangle}. \tag{2.10}$$

Unfortunately, since grad$f(\mathbf{X}_t)$, grad$f(\mathbf{X}_{t-1})$ and η_{t-1} are in different tangent spaces $T_{\mathbf{X}_t}\mathcal{M}$ and $T_{\mathbf{X}_{t-1}}\mathcal{M}$, the above two equations are not applicable on Riemannian manifolds. To address this issue, I need two geometric operations, namely, *Retraction* and *Vector Transport*. With the retraction mapping, one can move points in the direction of a tangent vector and stay on the manifold. In [124], the retraction on $\mathcal{M}$ can be computed in a closed form by

$$R_{\mathbf{X}}(\mathbf{E}) = P_{\mathcal{M}}\ (\mathbf{X}+\mathbf{E}) = \sum_{i=1} \sigma_i p_i \mathbf{q}_i^{\mathsf{T}}, \tag{2.11}$$

where $\sum_{i=1} \sigma_i p_i \mathbf{q}_i^{\mathsf{T}}$ denotes the best rank-ϱ approximation to $\mathbf{X}+\mathbf{E}$. In addition, the following *Vector Transport* makes the calculations of (2.9) and (2.10) meaningful. A vector transport $\mathcal{T}$ on a manifold $\mathcal{M}$ is a smooth map which transports tangent vectors from one tangent space to another. For convenience, let $\mathcal{T}_{\mathbf{X}\to\mathbf{Y}}(\eta_{\mathbf{X}})$ denote the transport from one tangent space $T_{\mathbf{X}}\mathcal{M}$ to another tangent space $T_{\mathbf{Y}}\mathcal{M}$, where $\eta_{\mathbf{X}}$ denotes the tangent vector on $\mathbf{X}$. The step size in the Step 3 and Step 5 is computed by the line search method. When updating $\mathbf{X}_{k+1}$, given a descent direction $\eta_k \in T_{\mathbf{X}_k}\mathcal{M}_r$, the step size α_k is determined such that

$$f(R_{\mathbf{X}_k}(\alpha_k\eta_k)) \leq f(\mathbf{X}_k) + c_1\alpha_k\langle \text{grad}f(\mathbf{X}_k), \eta_k\rangle, \tag{2.12}$$

where c_1 is the parameter. When updating ϑ_{k+1} by the standard gradient descent method, the step size γ_t can be computed by the line search on the following condition

$$f(\mathbf{X}_{k+1}, \vartheta_{k+1}) \leq f(\mathbf{X}_{k+1}, \vartheta_k) + c_2\gamma_k g_t, \tag{2.13}$$

where c_2 is the parameter and $0 < c_1 < c_2 < 1/2$.

Lastly, Algorithm 1 is guaranteed to converge to a stationary point of $f(\mathbf{X}, \vartheta)$.

Proposition 1. *The BNRCG algorithm is guaranteed to converge to a stationary point* $(\mathbf{X}^*, \vartheta^*)$ *of* $f(\mathbf{X}, \vartheta)$ *where* grad$f(\mathbf{X}^*, \vartheta^*) = 0$ *and* $\nabla_\vartheta f(\mathbf{X}^*, \vartheta^*) = 0$.

The proof can be found in Appendix B.

2.2.5 Automatic Latent Factor Detection by Active Subspace Search

Based on BNRCG for fixed-rank M³F, I propose an active subspace search method to detect the number of latent factors automatically presented in Algorithm 2.

Starting from $\mathbf{X} = 0$ where $\xi^0 = \mathbf{b}$, ARSS-M³F iterates with two main steps: to identify the most-active subspace through the worst-case analysis in Step 1, and to find the solution of the fixed-rank M³F problem by BNRCG in step 2. In the following, I present the details of the two main steps.

In the first step, I compute the gradient $\mathbf{G}$ of $f(\mathbf{X}, \vartheta)$ w.r.t. $\mathbf{X}$ and the active subspace can be found by performing a truncated SVD on $\mathbf{G}$ with the dimensionality of ϱ. In the second step, I initialize $\mathbf{X}^k = R_{\mathbf{X}^{k-1}}(-t_{\min}\bar{\mathbf{X}}\quad)$ where the step size t_{min} is determined by the line search method on the following condition:

$$f(R_{\mathbf{X}^{k-1}}(-t_{\min}\mathbf{G}^{k-1})) \leq f(\mathbf{X}^{k-1}) - \frac{t_{\min}}{2}\langle \mathbf{G}^{k-1}, \mathbf{G}^{k-1}\rangle \tag{2.14}$$

Then, the initialized $\mathbf{X}^k$ is used as the input of the Algorithm 1, namely BNRCG, by which $\mathbf{X}^k$ and ϑ^k can be updated iteratively. Note that after initializing $\mathbf{X}^k$ in the step 2(a), I increase the estimated rank of BNRCG by ϱ. Due to (2.14), the objective value $f(\mathbf{X}^k)$ monotonically decrease w.r.t. k. Therefore, I stop Algorithm 2 once the following condition is achieved

$$(f(\mathbf{X}^{k-1}) - f(\mathbf{X}^k))/(\varrho f(\mathbf{X}^{k-1})) \leq \varepsilon, \tag{2.15}$$

where ε is a stopping tolerance. In this way, as the algorithm is performed iteratively, I are able to detect the rank of the matrix to be recovered.

2.3 Empirical Studies

I demonstrate the performance of the proposed methods, namely BNRCG-M³F with fixed-rank problems and ARSS-M³F, by comparing with several related state-of-the-art methods, including FM³F [102], GROUSE [6], LMAFIT [135], ScGrassMC [94], LRGeomCG [124]

Algorithm 2 Active Riemannian Subspace Search for M³F .

Initialize $\mathbf{X}^0 = 0, r = 0, \xi^0 = \mathbf{b}$ and ϑ. Let $k = 1$.
1: Find active subspaces as follows:
(a): Compute $\mathbf{G} = \frac{\partial f(\mathbf{X}^k, \quad)}{\partial \mathbf{X}^k}$;
(b): Do thin SVD on $\mathbf{G}$: $[\mathbf{P} \quad, \Sigma \quad, \mathbf{Q} \quad] = \text{SVD}(\mathbf{G}, \varrho)$.
2: Let $\bar{\mathbf{X}} \quad = \mathbf{P} \quad \Sigma \quad \mathbf{Q}^\mathsf{T}$, do master problem optimization:
(a): Find an appropriate step size $t_{\min}$ by (2.14) and initialize $\mathbf{X}^k \quad =$ $R_{\mathbf{X}^{k-1}}(-t_{\min}\bar{\mathbf{X}} \quad)$ (Warm Start).
(b): Let $r = r + \varrho$ and update $\mathbf{X}^k$ and ϑ^k by Algorithm 1
3: Quit if stopping conditions are achieved. Let $k = k + 1$ and go to step 1.

and RTRMC [15] , on both synthetic and real-world CF tasks. Seven data sets are used in the experiments, including three synthetic data sets and four real-world data sets, Movielens 1M, Movielens 10M [58], Netflix [10] and Yahoo! Music Track 1 data set [40].

The root-mean-square error (RMSE) on both training and testing set will be used as the comparison metric: RMSE $= \sqrt{\sum_{ij\in\Pi}(\mathbf{Y}^*_{ij} - \mathbf{Y}_{ij})^2/|\Pi|}$, where $\mathbf{Y}^*$ denoted the reconstructed ratings according to (2.3), and $|\Pi|$ denotes number of emblements in the set Π. All the experiments are conducted in Matlab on a work station with an Intel(R) CPU (Xeon(R) E5-2690 v2 @ 3.00GHz) and 256GB memory.

2.3.1 Synthetic Experiments

In the synthetic experiments where I know the ground-truth, I will demonstrate four points: 1) The sensitivity of the regularization of the proposed M³F methods; 2) The scalability of BNRCG-M³F and ARSS-M³F over other methods; 3) The importance of the squared hinge loss measure over other measures for rating data, e.g., the least square error; 4) The effectiveness of latent factor detection by ARSS-M³F. To demonstrate the above points, I study three synthetic problems of two scales.

Synthetic Problem

For each of the three synthetic problems, motivated by [94, 120], I first generate a ground-truth low-rank matrix by $\widehat{\mathbf{X}} = \widehat{\mathbf{U}}\text{diag}(\widehat{\partial})\widehat{\mathbf{V}}^\mathsf{T}$, where ∂ is a r-sparse vector with each nonzero entry

sampled from *Gaussian* distribution $\mathcal{N}(0, 1000)$, $\widehat{\mathbf{U}} \in \mathrm{St}_r^m$ and $\widehat{\mathbf{V}} \in \mathrm{St}_r^n$. In the both two small-scale problems, $\widehat{\mathbf{X}}$ is of size $1,000 \times 1,000$ with $r = 20$, while the large-scale problem $\widehat{\mathbf{X}}$ is of size $20,000 \times 20,000$ with $r = 50$. After sampling the original entries, I respectively produce the binary ratings by $\widehat{\mathbf{Y}}_{ij} = \mathrm{sgn}(\widehat{\mathbf{X}}_{ij})$, and the ordinal ratings $\{1, 2, 3, 4, 5\}$ by projecting the entries of $\widehat{\mathbf{X}}$ into five bins according to their values, which results in a rating matrix $\widehat{\mathbf{Y}}$. Once $\widehat{\mathbf{Y}}$ is generated, I sample $l = r(m + n - r) \times \zeta_{os}$ entries from $\widehat{\mathbf{Y}}$ uniformly to form the observed ratings $\mathbf{Y}$, where ζ_{os} is the oversampling factor [80]. In the experiments I set $\zeta_{os} = 3.5$.

Sensitivity of Regularization Parameter

In this section, to demonstrate the sensitivity of regularization, I perform experiments on the small-scale binary matrix. To illustrate the impact of the regularization in the proposed methods, I test BNRCG-M^{3}F with various regularization parameters λ. Figure 2.1 reports the training RMSE and testing RMSE. The convergence is shown in Figure 2.2a. As can be seen, the regularization is crucial for preventing overfitting.

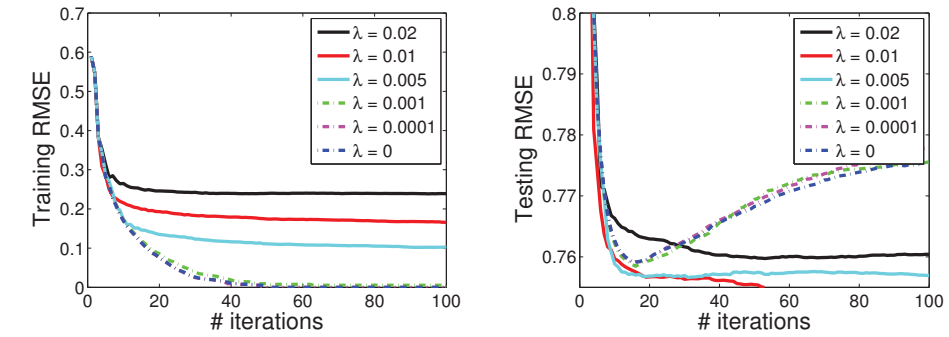


Figure 2.1: RMSE of BNRCG-M^{3}F on binary rating data.

Convergence of M^{3}F on Ordinal Rating Data

In this section, I perform experiments on the small-scale ordinal matrix. I compare the proposed algorithms with the six baseline methods and collect the convergence behavior of the three M^{3}F methods. The ground-truth rank is used as the estimated rank for all methods excluding ARSS-M^{3}F.

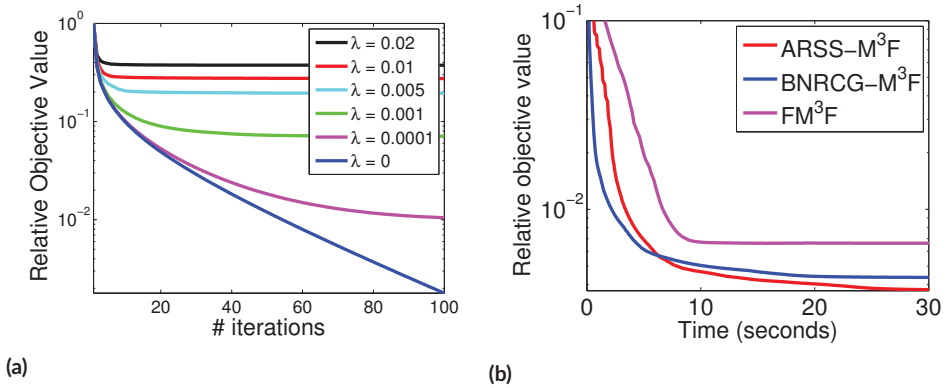


Figure 2.2: Relative objective values of various methods.

The convergence behavior of our methods and FM³F is illustrated in Figure 2.2b, which shows that our methods can converge better and faster. Table 2.2 reports the resultant RMSE on the testing set and the computational time of each method on the small-scale synthetic ordinal rating data set.

Scalability of M³F on Ordinal Rating Data

In this section, I perform experiments on the large-scale ordinal matrix. I compare our methods with the 5 baseline algorithms. I use the ground-truth rank as the estimated rank for all methods except ARSS-M³F. The average estimated rank of ARSS-M³F is 42, which is close to the groundtruth rank of 50. According to the estimated rank in the two synthetic datasets, the latent factor detection of ARSS-M³ is effective. The RMSE on the testing set and computational time of each algorithm are listed in Table 2.2.

Table 2.1: Statistics of the Real-world Data Sets.

Data Sets	# users	# items	# ratings
Movielens 1M	6,040	3,952	1,000,209
Movielens 10M	71,567	10,681	10,000,054
Netflix	480,189	17,770	100,480,507
Yahoo! Music Track 1	1,000,990	624,961	262,810,175

Table 2.2: Experimental results on synthetic and real-world data sets. Computational time is recorded in seconds.

Methods	Small Synthetic*		Large Synthetic*		Movielens 1M†		Movielens 10M†		Netflix†		Yahoo Music†	
	RMSE	Time	RMSE	Time	RMSE	Time	RMSE	Time	RMSE	Time	RMSE	Time
FM^{3}F [102]	0.3811	11.99	0.3899	2186	0.9344	212.2051	0.9143	13001	1.0971	65662	-	-
GROUSE [6]	0.4718	27.84	0.512	11214	0.9225	39.4184	0.8653	3853	-	-	-	-
LMAFIT [135]	0.4701	6.08	0.4973	827	0.9373	19.9465	0.8424	832	0.9221	4374	24.222	24349
ScGrassMC [94]	0.4638	10.19	0.4714	2149	0.9372	21.3109	0.8427	917	0.9192	5787	24.7982	37705
LRGeomCG [124]	0.4679	6.01	0.4904	814	0.9321	10.2484	0.849	312	0.9015	3151	25.2279	8666
RTRMC [15]	0.4676	8.68	0.4715	884	0.9311	14.1038	0.846	673	0.9102	6465	24.5971	32592
BNRCG-M^{3}F	0.3698	5.34	0.3915	635	0.9285	13.4437	0.8437	714	0.9022	4118	23.8573	24631
ARSS-M^{3}F	0.3693	5.33	0.3684	542	0.9222	9.5482	0.8411	650	0.9001	3583	23.7902	22065

* No cost of model selections is included for all fix-rank methods as the ground-truth rank is available.

† The rank detected by ARSS-M^{3}F is used as the estimated rank for other methods. Thus no model selection is considered. The average ranks estimated by ARSS-M^{3}F on Movielens 1M, Movielens 10M, Netflix and Yahoo Music are 8, 14, 16 and 28 respectively.

2.3.2 Real-world Experiments

In real-world data experiments, to demonstrate the significance of the hinge loss to the rating data and effectiveness of latent factor estimation of our method, I study four real-world large scale data sets, namely Movielens 1M, Movielens 10M data set, Netflix data set and Yahoo! Music Track 1 data set. The baseline methods include FM^{3}F, GROUSE, LMAFIT, ScGrassMC, LRGeomCG and RTRMC.

Table 2.1 lists the size statistics of the four data sets. The vast majority (99.71%) of ratings in Yahoo! Music Track 1 are multiples of ten. For convenience, I only consider these ratings. For Movielens 10M and Yahoo! Music Track 1, I map the ratings to ordinal integer values before the experiment. For each data set, I sample 80% of data into the training set and the rest into the testing set.

Table 2.2 reports the computational time of all comparison methods and testing RMSE on the four data sets. According to the resultant RMSE, compared to other loss measure, i.e. least square loss, our method can recover the matrix with lower error. Note that in all experiments in both synthetic and real-world data, no model selection cost is included for all comparison methods. If model selections are considered, the comparison methods will cost much more time. Some results for GROUSE and M^{3}F are not available due to their high computation cost. From the table, ARSS-M^{3}F and BNRCG-M^{3}F recover the rating matrix efficiently and outperform

other comparison methods in terms of RMSE on the four real-world data sets. It is worth mentioning that though LRGeomCG shows faster speed on Yahoo data set, it achieves much worse RMSE than M^{3}F based methods.

2.4 Conclusion

To deal with the ordinal discrete ratings in recommendation systems, M^{3}F is proposed. However, existing M^{3}F methods is faced with the scalability and latent factor detection issues. To address the two challenges, I present ARSS-M^{3}F, a scalable M^{3}F method based on active Riemannian subspace search. Specifically, the proposed algorithm first treat the M^{3}F problem as the fixed number of latent factors and solve it using BNRCG. In the meantime, a simple and efficient active subspace search approach is applied to automatically compute the number of latent factors. Experiments on both synthetic and real-world data demonstrate that the proposed method can provide competitive performance.

2.5 Appendix A: Computation of grad$f(\mathbf{X}, \vartheta)$

According to [124], a tangent vector $\eta \in \mathcal{TM}_r$ is represented as $\eta = \mathbf{UMV}^\mathsf{T} + \mathbf{U}_p\mathbf{V}^\mathsf{T} + \mathbf{UV}_p^\mathsf{T}$ (see equation (2.4) for details). By definition, the Riemannian gradient of $f(\mathbf{X}, \vartheta)$ w.r.t. $\mathbf{X}$, denoted by grad$f(\mathbf{X}, \vartheta)$, at $\mathbf{X} = \mathbf{U}\mathrm{diag}(\sigma)\mathbf{V}^\mathsf{T}$ can be calculated by $P_{T_\mathbf{X}\mathcal{M}_r}(\mathbf{G})$, where $P_{T_\mathbf{X}\mathcal{M}_r}(\mathbf{Z}) = P_U\mathbf{Z}P_V + P_U^\perp\mathbf{Z}P_V + P_U\mathbf{Z}P_V^\perp$ is the projection of $\mathbf{G}$ onto the tangent space $\mathcal{TM}_r$. Let $\Xi = \lambda\mathrm{diag}(\sigma - \nu/\sigma^3)$. For convenience, I first present the computation of grad$f(\mathbf{X}, \vartheta)$ in Algorithm 3.

Lemma 1. *Suppose* $\mathbf{U}_p$, $\mathbf{V}_p$, *and* $\mathbf{M}$ *are obtained from Algorithm 3, then grad*$f(\mathbf{X}, \vartheta) = \mathbf{UMV}^\mathsf{T} + \mathbf{U}_p\mathbf{V}^\mathsf{T} + \mathbf{UV}_p^\mathsf{T}$.

Proof. To verify the validity of Algorithm 3, I just need to show that, $P_{T_\mathbf{X}\mathcal{M}_r}(\mathbf{G}) = \mathbf{UMV}^\mathsf{T} + \mathbf{U}_p\mathbf{V}^\mathsf{T} + \mathbf{UV}_p^\mathsf{T}$.

Notice that, $\mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}$. On one hand, I have $P_{T_\mathbf{X}\mathcal{M}_r}(\mathbf{G}) = P_{T_\mathbf{X}\mathcal{M}_r}(\widehat{\mathbf{G}} + \mathbf{U}\Xi\mathbf{V}^\mathsf{T}) = \widehat{\mathbf{G}}P_V + P_U\widehat{\mathbf{G}} - P_U\widehat{\mathbf{G}}P_V + \mathbf{U}\Xi\mathbf{V}^\mathsf{T}$. On the other hand, according to Algorithm

Algorithm 3 Compute Riemannian gradient grad$f(\mathbf{X})$.

1: Let $\Xi = \lambda \text{diag}(\sigma - \nu/\sigma^3)$, and compute $\widehat{\mathbf{G}}$ via (2.6).
2: Compute $\mathbf{G}_u = \widehat{\mathbf{G}}^\top \mathbf{U}$, and $\mathbf{G}_v = \widehat{\mathbf{G}}\mathbf{V}$.
3: Compute $\widehat{\mathbf{M}} = \mathbf{U}^\top \mathbf{G}_v$.
4: Compute $\mathbf{U}_p = \mathbf{G}_v - \mathbf{U}\widehat{\mathbf{M}}$, and $\mathbf{V}_p = \mathbf{G}_u - \mathbf{V}\widehat{\mathbf{M}}^\top$.
6: Update $\mathbf{M} = \widehat{\mathbf{M}} + \Xi$.
6: Output $\mathbf{U}_p$, $\mathbf{V}_p$, and $\mathbf{M}$, and grad$f(\mathbf{X}, \vartheta) = \mathbf{U}\mathbf{M}\mathbf{V}^\top + \mathbf{U}_p\mathbf{V}^\top + \mathbf{U}\mathbf{V}_p^\top$.

3, I have $\eta = \mathbf{U}\mathbf{M}\mathbf{V}^\top + \mathbf{U}_p\mathbf{V}^\top + \mathbf{U}\mathbf{V}_p^\top = \mathbf{U}\widehat{\mathbf{M}}\mathbf{V}^\top + \mathbf{U}_p\mathbf{V}^\top + \mathbf{U}\mathbf{V}_p^\top + \mathbf{U}\Xi\mathbf{V}^\top = \widehat{\mathbf{G}}\mathbf{V}\mathbf{V}^\top + \mathbf{U}\mathbf{U}^\top\widehat{\mathbf{G}} - \mathbf{U}\widehat{\mathbf{M}}\mathbf{V}^\top + \mathbf{U}\Xi\mathbf{V}^\top$, which actually equals to $P_{T_{\mathbf{X}}\mathcal{M}_r}(\mathbf{G})$. This completes the proof. □

2.6 Appendix B: Proof of Proposition 1

The proof parallels the proof in [124]. Notice that, the optimization on ϑ is conducted in Euclidian space $\mathbb{R}^{L-1}$. Moreover, $\{\vartheta_t\}$ is bounded; otherwise $\ell(\mathbf{X}, \vartheta)$ will go to infinity according to (2.1). Without loss of generality, suppose $\vartheta_t \in [-l, l]^{L-1}$, where $l > 0$ is a finite number. Following [124], I can also show that $\{\mathbf{X}_t\}$ stay in a closed and bounded subset of $\mathcal{M}_r$.

Let $\Psi = \{\mathbf{X} \in \mathcal{M}_r, f(\mathbf{X}, \vartheta) \leq f(\mathbf{X}_0, \vartheta_0)\}$ be the level set at $(\mathbf{X}_0, \vartheta_0)$. Due to the line search, I have $\ell(\mathbf{X}_t, \vartheta_t) + \frac{}{2}(||\mathbf{X}_t||_F^2 + \nu||\mathbf{X}_t^\dagger||_F^2) \leq f(\mathbf{X}_0, \vartheta_0)$. Therefore, I have $\frac{}{2}||\mathbf{X}_t||_F^2 \leq f(\mathbf{X}_0, \vartheta_0)$, which implies $\sigma_1 = \sqrt{||\mathbf{X}_t||_F^2} \leq \sqrt{2f(\mathbf{X}_0, \vartheta_0)/\lambda}$. Here, σ_1 denotes the largest singular value of $\mathbf{X}_t$. Similarly, I have $\frac{}{2}||\mathbf{X}_t^\dagger||_F^2 = \sum_{i=1}^r \frac{}{2\ \ i^2} \leq f(\mathbf{X}_0, \vartheta_0)$, which implies that $\frac{}{2\ \ i^2} \leq f(\mathbf{X}_0, \vartheta_0), \forall i \in \{1, ..., r\}$. This further implies that $\sigma_r \geq \sqrt{\nu\lambda/2f(\mathbf{X}_0, \vartheta_0)}$, where σ_r is the least singular value of $\mathbf{X}_t$.

Clearly, all $\mathbf{X}_t$ stay inside the set $\mathcal{S} = \{\mathbf{X} \in \mathcal{M}_r : \sigma_1 \leq \sqrt{2f(\mathbf{X}_0, \vartheta_0)/\lambda}, \sigma_r \geq \sqrt{\nu\lambda/2f(\mathbf{X}_0, \vartheta_0)}\}$, which is closed and bounded, hence compact.

Now I complete the proof by contradiction. Without loss of generality, suppose

$$\lim_{t\to\infty} ||\text{grad}f(\mathbf{X}_t, \vartheta_t)||_F + ||\nabla_{\vartheta_t} f(\mathbf{X}_t, \vartheta_t)||_2 \neq 0,$$

then there exists an $\varepsilon > 0$, and a subsequence in $\{(\mathbf{X}_t, \vartheta_t)\}_{t\in\Gamma}$ such that $||\text{grad}f(\mathbf{X}_t, \vartheta_t)||_F +$

$||\nabla_{\vartheta} f(\mathbf{X}_t, \vartheta_t)||_2 \geq \varepsilon > 0$ for all $t \in \Gamma$. Since $\mathbf{X}_t \in \mathcal{S}$ and ϑ_t is constrained in $[-l, l]^{L-1}$, the subsequence $\{(\mathbf{X}_t, \vartheta_t)\}_{t \in \Gamma}$ should have a limit point $(\mathbf{X}^*, \vartheta^*)$ in $\mathcal{S} \times [-l, l]^{L-1}$. By continuity of grad$f(\mathbf{X}, \vartheta)$ and $\nabla_{\vartheta} f(\mathbf{X}, \vartheta)$ (which can be easily verified for squared hinge loss), this implies that $||\text{grad} f(\mathbf{X}_t, \vartheta_t)||_F \geq \varepsilon$ which contradicts Theorem 4.3.1 in [1] that every accumulation point is a critical point of $f(\mathbf{X}, \vartheta)$. I therefore conclude that $\lim_{t \to \infty} ||\text{grad} f(\mathbf{X}_t, \vartheta_t)||_F = 0$ and $\lim_{t \to \infty} ||\nabla_{\vartheta} f(\mathbf{X}_t, \vartheta_t)||_2 = 0$.

3

Semi-Supervised Learning by Label Aggregation

3.1 Introduction

This chapter [*] focuses on the two aforementioned challenges of SSL, i.e. scalability and robustness. Inspired by crowdsourcing [111, 116], I propose an efficient RObust Semi-Supervised Ensemble Learning (ROSSEL) method to approximate ground-truth labels of unlabeled data

[*] The main results of this chapter were previously published in Yan Yan, Zhongwen Xu, Ivor W. Tsang, Guodong Long, Yi Yang. Robust Semi-supervised Learning through Label Aggregation. In *Thirtieth Conference on Artificial Intelligence (AAAI)* 2016.

through aggregating a number of pseudo-labels generated by low-cost *weak annotators*, such as linear SVM classifiers. Meanwhile, based on the aggregated labels, ROSSEL learns an inductive SSL classifier by Multiple Label Kernel Learning (MLKL) [77]. Unlike most existing SSL algorithms, the proposed ROSSEL requires neither expensive graph Laplacian nor iterative label switching. Instead, it only needs *one* iteration for label aggregation and can be solved by an SVM solver very efficiently. The major contributions of this chapter are listed as follows,

- Leveraging an ensemble of low-cost supervised weak annotators, I propose ROSSEL to efficiently obtain a weighted combination of pseudo-labels of unlabeled data to approximate ground-truth labels to assist semi-supervised learning.
- Instead of simple label aggregation strategies used in crowdsourcing (*e.g.*majority voting), ROSSEL performs a weighted label aggregation using MLKL. Meanwhile it learns an inductive SSL classifier, which only requires *one* iteration and linear time complexity w.r.t. number of data and features.
- Complexity analysis of several competing SSL methods and the proposed method is provided.

3.2 The Proposed Model

Inspired by crowdsourcing methods [111, 116], I propose a new SSL algorithm that efficiently learns a classifier by leveraging both labeled and unlabeled data. Our proposed method consists of the two steps, namely label generation and label aggregation, illustrated in Figure 3.1. In the first stage, a set of weak annotators are trained and applied to unlabeled data to generate a set of pseudo-labels. In the second stage I combine the pseudo-labels to approximate the optimal labels of unlabeled data. In the meantime, weight vectors is derived, which enables ROSSEL to handle unseen data.

3.2.1 Label Generation

Low-cost, less-than-expert labels are easy to obtain from weak annotators in crowdsourcing [111]. Following the crowdsourcing framework, ROSSEL firstly generates a set of pseudo-labels for unlabeled data using ensemble learning. In this chapter I focus on bagging to generate pseudo-labels.

Bagging is a simple and effective supervised ensemble learning algorithm, which produces a number of bootstrap replicates using bootstrap sampling. A weak learner is trained on each bootstrap replicate. By applying these weak learners on unlabeled data, a set of pseudo-labels can be derived. Bagging finally aggregates all the pseudo-labels by majority voting to generate predictions.

ROSSEL trains weak annotators using bootstrap sampling. Similar to crowdsourcing, I apply weak annotators on unlabeled data and obtain the resultant less-than-expert labels. The label generation procedure is illustrated in Figure 3.1.

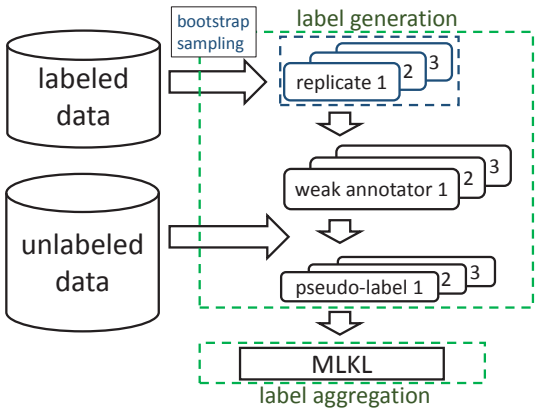


Figure 3.1: Illustration of the proposed ROSSEL.

3.2.2 Label Aggregation by MLKL

Considering a binary supervised learning scenario, let $\mathcal{D}_L = \{\mathbf{x}_i, y_i\}_{i=1}^{l}$ denotes the labeled set, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$ denotes the feature vector and the label of the i-th sample,

respectively. A general objective function is formulated as follows

$$\min_{\mathbf{w}} \Omega(\mathbf{w}) + C\ell(\mathbf{w}), \tag{3.1}$$

where $\mathbf{w} \in \mathbb{R}^d$ is the weight vector, $\Omega(\mathbf{w})$ is the regularization term, $\ell(\mathbf{w})$ is a loss function and C is the regularization parameter. I focus on the ℓ_2-regularized hinge loss. The objective function of hinge loss then can be specifically written as

$$\begin{aligned} &\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{i=1}^{l} \xi_i \\ &s.t. \quad y_i \mathbf{w}^\top \mathbf{x}_i \geq 1 - \xi_i, \xi_i \geq 0, i = 1, ..., l, \end{aligned} \tag{3.2}$$

where ξ_i is the slack variable of the i-th instance.

SSL is aimed to exploit the abundant unlabeled data. Hence let $\mathcal{D}_U = \{\mathbf{x}_i\}_{i=l+1}^{n}$ denote the unlabeled set and I incorporate the information of unlabeled data into the objective function, which can be written as,

$$\begin{aligned} &\min_{\tilde{\mathbf{y}} \in \mathcal{Y}} \min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||_2^2 + C_1 \sum_{i=1}^{l} \xi_i + C_2 \sum_{i=l+1}^{n} \xi_i \\ &s.t. \quad \tilde{y}_i \mathbf{w}^\top \mathbf{x}_i \geq 1 - \xi_i, \quad \xi_i \geq 0, i = 1, ..., n, \end{aligned} \tag{3.3}$$

where C_1 and C_2 are the regularization parameters that control the tradeoff between model complexity, the cost generated by the labeled data, and the cost generated by the unlabeled data, and $\mathcal{Y} = \{\tilde{\mathbf{y}} | \tilde{\mathbf{y}} = [\mathbf{y}_L; \tilde{\mathbf{y}}_U], \tilde{\mathbf{y}}_U \in \{-1, +1\}^{n-l}\}$, where $\mathbf{y}_L \in \mathbb{R}^l$ represents the ground-truth label vector of labeled data, and $\tilde{\mathbf{y}}_U$ represents any possible labels of unlabeled data. Thus there are exponential possible values for $\mathbf{y}_U$, i.e. the labels of unlabeled data, which is intractable to directly optimize.

By introducing dual variables $\alpha \in \mathbb{R}^n$, the Lagrangian of Equation (3.3) can be obtained by

$$\begin{aligned}\mathcal{L}(\mathbf{w}, \alpha) = \frac{1}{2}||\mathbf{w}||_2^2 + C_1 \sum_{i=1}^{l} \xi_i + C_2 \sum_{i=l+1}^{n} \xi_i \\ + \sum_{i=1}^{n} \alpha_i (1 - \xi_i - \tilde{y}_i \mathbf{w}^\top \mathbf{x}_i).\end{aligned} \tag{3.4}$$

By setting the derivatives of $\mathcal{L}$ w.r.t. $\mathbf{w}$ and ξ_i as 0, the Lagrangian can be updated as below,

$$\mathcal{L} = -\frac{1}{2}\alpha^\top \left((XX^\top) \odot \tilde{\mathbf{y}}\tilde{\mathbf{y}}^\top \right) \alpha + 1^\top \alpha, \tag{3.5}$$

where $\alpha \in \mathcal{A}$ and $\mathcal{A} = \{\alpha | 0 \leq \alpha_i \leq C_1, 0 \leq \alpha_j \leq C_2, 1 \leq i \leq l, l+1 \leq j \leq n\}$. I can then replace the inner minimization problem of Problem (3.3) by its dual as below,

$$\min_{\tilde{\mathbf{y}} \in \mathcal{Y}} \max_{\in \mathcal{A}} -\frac{1}{2}\alpha^\top \left((XX^\top) \odot \tilde{\mathbf{y}}\tilde{\mathbf{y}}^\top \right) \alpha + 1^\top \alpha, \tag{3.6}$$

where $X = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]^\top$. It is usually difficult to optimize $\tilde{\mathbf{y}}$ due to the significant number of possible values. Inspired by ideas from crowdsourcing, which obtain sufficiently qualified labels on unlabeled data by exploiting a set of weak annotators, I propose to solve Problem (3.6) by MLKL [78, 77].

Definition 1. *Given a size-M label set $\{\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2, ..., \tilde{\mathbf{y}}_M\}$, multiple label kernel learning (MLKL) refers to the problem as below,*

$$\min_{\in \mathcal{U}} \max_{\in \mathcal{A}} -\frac{1}{2}\alpha^\top \left((XX^\top) \odot (\sum_{m=1}^{M} \mu_m \tilde{\mathbf{y}}_m \tilde{\mathbf{y}}_m^\top) \right) \alpha + 1^\top \alpha, \tag{3.7}$$

which aims to find a weighted combination of the label kernels $\sum_{m=1}^{M} \mu_m \tilde{\mathbf{y}}_m \tilde{\mathbf{y}}_m^\top$ to approximate the ground-truth label kernel $\tilde{\mathbf{y}}^* \tilde{\mathbf{y}}^{*\top}$, where $\mathcal{U} = \{\mu | \sum_{m=1}^{M} \mu_m = 1, \quad \mu_m \geq 0\}$, $\mathcal{A} = \{\alpha | 0 \leq \alpha_i \leq C_1, 0 \leq \alpha_j \leq C_2, 1 \leq i \leq l, l+1 \leq j \leq n\}$, and $\mu = [\mu_1, \mu_2, ..., \mu_M]^\top$ denotes the weight vector of base label kernels.

Similar to crowdsourcing, a set of pseudo-labels of unlabeled data are generated in the first

Table 3.1: Comparison of complexity of the proposed method and other related SSL methods.

Mthods	LapSVM	LapRLS	meanS3VM	CS4VM	ASL	ROSSEL
Complexity	$O(n^3d)$	$O(n^3d)$	$O(n^2dT)$	$O(n^2dT)$	$O(nd^2T)$	$O(\mathcal{M}nd)$

In this table, n, d, $\mathcal{M}$ and T represent the number of data, the dimension of data, the number of weak annotators and the number of iterations of the algorithm respectively.

step by bootstrap sampling. In the second step, I propose to obtain the SSL classifier by MLKL. Assume that there are $\mathcal{M}$ pseudo-labels, namely $\mathcal{Y}_{\mathcal{M}} = \{\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2, ..., \tilde{\mathbf{y}}_{\mathcal{M}}\}$, then I can complete the primal formulation of Problem (3.7) as,

$$\begin{aligned} \min_{\in\mathcal{U},\mathbf{w}_m,} \quad & \frac{1}{2}\sum_{m=1}^{M}\frac{1}{\mu_m}||\mathbf{w}_m||_2^2 + C_1\sum_{i=1}^{l}\xi_i + C_2\sum_{i=l+1}^{n}\xi_i \\ s.t. \quad & \sum_{m=1}^{M}\tilde{y}_{mi}\mathbf{w}_m^\top\mathbf{x}_i \geq 1-\xi_i, \xi_i \geq 0, i = 1, ..., n, \end{aligned} \tag{3.8}$$

where $\tilde{y}_{mi}$ denotes the label for the i-th sample in $\mathbf{y}_m$.

By setting $\hat{\mathbf{w}} = [\frac{\mathbf{w}_1}{\sqrt{\mu_1}}, ..., \frac{\mathbf{w}_M}{\sqrt{\mu_M}}]^\top$, $\hat{\mathbf{x}}_i = [\sqrt{\mu_1}\mathbf{x}_i, \sqrt{\mu_2}\tilde{y}_{1i}\tilde{y}_{2i}\mathbf{x}_i, ..., \sqrt{\mu_T}\tilde{y}_{1i}\tilde{y}_{Mi}\mathbf{x}_i]^\top$, and $\hat{\mathbf{y}} = \tilde{\mathbf{y}}_1$, the primal problem of MLKL (3.7) becomes

$$\begin{aligned} \min_{\hat{\mathbf{w}},} \quad & \frac{1}{2}||\hat{\mathbf{w}}||_F^2 + C_1\sum_{i=1}^{l}\xi_i + C_2\sum_{i=l+1}^{n}\xi_i \\ s.t. \quad & \hat{y}_i\hat{\mathbf{w}}^\top\hat{\mathbf{x}}_i \geq 1-\xi_i, \xi_i \geq 0, i = 1, ..., n. \end{aligned} \tag{3.9}$$

Problem (3.9) is similar to the primal of a standard SVM problem, and can be easily solved by existing SVM packages, such as LIBLINEAR. Compared to Problem (3.3), Problem (3.9) can be solved very efficiently.

ROSSEL is easy to extend to cope with multiclass problems by applying the one-vs-all strategy. The detailed ROSSEL algorithm for a multiclass case can be found in Algorithm 4.

3.3 Complexity Analysis

There are two main stages in the proposed method, namely label generation and label aggregation. In the label generation step, $\mathcal{M}$ weak annotators are trained. Weak annotators can be any

Algorithm 4 RObust Semi-Supervised Ensemble Learning (ROSSEL)

1: Initialize $\mathcal{M}$, the number of weak annotators.
2: for $k = 1$ to K do
3: Sample $\mathcal{M}$ bootstrap replicates $\{(\bar{\mathbf{X}}_1, \bar{\mathbf{y}}_{k1}), (\bar{\mathbf{X}}_2, \bar{\mathbf{y}}_{k2}), ..., (\bar{\mathbf{X}}_T, \bar{\mathbf{y}}_{kM})\}$ from the labeled set $\mathcal{D}_L$.
4: for $m = 1$ to $\mathcal{M}$ do
5: Train an SVM model $\mathcal{M}_{km}$ on $\bar{X}_m$ and $\bar{\mathbf{y}}_{km}$.
6: Derive $\tilde{\mathbf{y}}_{km}$ by predicting on the unlabeled data X_U using $\mathcal{M}_{km}$.
7: Add $\tilde{\mathbf{y}}_{km}$ into the working set $\mathcal{Y}_{kM}$
8: end for
9: Compute $\{\mathbf{w}_{k1}, \mathbf{w}_{k2}, ..., \mathbf{w}_{kM}\}$ and μ_k by solving Problem (3.8).
10: Calculate prediction $p_{jk} = \sum_{m=1}^{M} \mu_{km} \mathbf{w}_{km}^\top \mathbf{x}_j$ for a test data $\mathbf{x}_j$.
11: end for
12: Choose the class label for $\mathbf{x}_j$ by $\arg\max_k \{p_{jk}\}_{k=1}^K$.

cheap learner. In our experiments, I use LIBLINEAR to train linear SVMs as the weak annotators. Hence, this leads to a complexity of $O(\mathcal{M}nd)$ where n and d stand for the number and the dimension of data respectively. In the label aggregation step, MLKL can be solved according to Problem (3.9) by LIBLINEAR [43], and μ, the coefficient of base label kernels, can be simply updated by closed-form solution, which results in the complexity of $O(\mathcal{M}nd)$. Compared with many other SSL methods that require a number iterations for label switching and model training, the proposed ROSSEL only requires *one* iteration. Therefore, the overall complexity of ROSSEL is $O(\mathcal{M}nd)$, which does not rely on T, the number iterations.

In Table 3.1, I list the complexity of various SSL algorithms, including LapSVM [9], LapRLS [9], meanS3VM [76], CS3VM [75] and ASL [128]. LapSVM and LapRLS have high complexity w.r.t. the number of instances n due to the inverse of a dense Gram matrix. Note that meanS3VM,

Table 3.2: Data statistics.

Datasets	# train	# test	# features	# classes
CNAE9	800	280	856	9
dna	2,559	627	180	3
connect4-10k	8,000	2,000	126	3
protein	19,200	5,187	357	3
rcv1-train	12,384	3,114	47,236	38
rcv1-all	420,000	111,920	47,236	40

CS4VM and ASL require to update their models iteratively. Consequently, their complexity contains T. It can be expensive if a large number of iterations is required.

Table 3.3: Average accuracy (±Standard Deviation(%)) over 10 runs.

Methods	CNAE9	dna	connect4-10k	protein	rcv1-train	rcv1-all
LIBLINEAR	82.86(±2.56)	84.78(±1.52)	64.40(±1.73)	60.54(±0.64)	74.45(±1.89)	87.56(±0.09)
LIBSVM	83.04(±2.94)	85.96(±1.42)	63.43(±2.43)	61.84(±1.31)	74.93(±1.88)	87.57(±0.12)
ensemble-10SVM	79.75(±2.41)	83.32(±1.38)	65.26(±2.54)	60.78(±1.40)	72.39(±1.53)	87.46(±0.09)
ensemble-50SVM	81.56(±2.42)	84.63(±1.84)	65.70(±1.99)	60.91(±0.85)	73.17(±1.90)	87.60(±0.08)
LapSVM	85.33(±3.13)	85.63(±1.28)	64.39(±1.82)	60.46(±0.85)	74.91(±1.90)	*
LapRLS	85.47(±2.72)	85.84(±1.23)	63.41(±1.63)	60.72(±0.61)	74.55(±1.92)	*
meanS3VM	83.12(±3.57)	85.04(±1.17)	–	–	–	–
CS4VM	84.93(±2.98)	88.04(±1.12)	62.04(±2.14)	–	–	–
ASL	82.61(±2.15)	90.03(±0.98)	60.83(±1.41)	58.94(±1.19)	*	*
ROSSEL10	85.11(±2.42)	88.50(±1.91)	67.89(±1.16)	61.88(±1.34)	79.22(±2.00)	89.20(±0.15)
ROSSEL50	85.04(±3.14)	88.52(±1.54)	68.20(±0.98)	62.33(±0.90)	78.77(±2.25)	89.18(±0.11)

I report the results of ensemble-SVM and ROSSEL with both 10 and 50 weak annotators. Semi-supervised methods with maximum accuracy are in bold. Some of the compared algorithms either require much memory (indicated by "*" in the above table) or very expensive in computation (*e.g.*more than a day, indicated by "–" in the above table). Therefore, these algorithms can not be applied to the large datasets such as the rcv1-all dataset.

Table 3.4: Average training time (in seconds) over 10 runs.

Methods	CNAE9	dna	connect4-10k	protein	rcv1-train	rcv1-all
LIBLINEAR	0.0008	0.0009	0.0909	0.0126	0.3405	1.6855
LIBSVM	0.0052	0.0385	0.1408	0.2387	1.3338	672.4409
ensemble-10SVM	0.0060	0.0136	0.0487	0.2329	2.1081	33.2070
ensemble-50SVM	0.0224	0.0405	0.3919	0.8019	14.5482	119.0243
LapSVM	0.1596	7.0668	14.3528	152.9257	494.4695	*
LapRLS	0.1715	7.0214	13.0248	152.8537	420.5253	*
meanS3VM	2.8588	13.8941	–	–	–	–
CS4VM	1.3219	9.5178	539.8876	–	–	–
ASL	3.4355	16.3261	115.6894	1748.2612	*	*
ROSSEL10	0.2123	0.2271	0.7955	3.4457	45.5584	815.0660
ROSSEL50	0.5481	1.4133	3.2811	16.5558	336.4487	6024.5965

I report the results of ensemble-SVM and ROSSEL with both 10 and 50 weak annotators. Semi-supervised methods with minimum training time are in bold. Some of the compared algorithms either require much memory (indicated by "*" in the above table) or very expensive in computation (*e.g.*more than a day, indicated by "–" in the above table). Therefore, these algorithms can not be applied to the large datasets such as the rcv1-all dataset.

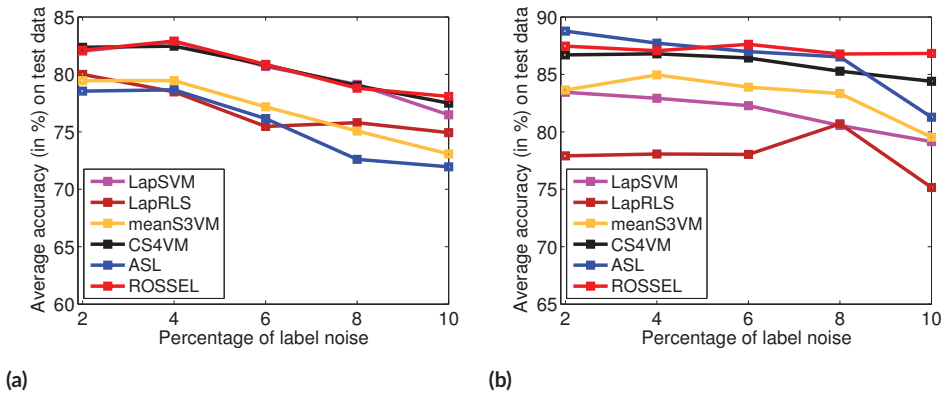


Figure 3.2: Average accuracy on the CNAE9 and dna datasts over 10 runs when label noise is present.

3.4 Experiments

In this section, I demonstrate the robustness and performance of the proposed algorithm by comparing with eight baselines. These baselines include three supervised learning methods, namely LIBLINEAR [43], LIBSVM [24], ensemble LIBSVM, and five SSL algorithms, namely LapSVM [9], LapRLS [9], meanS3VM [76], CS4VM [75] and ASL [128]. In total six datasets are used, namely CNAE9, dna, connect4, protein and rcv1-train and rcv1-all. Three experiments are performed, which respectively investigate the resistance to label noise, performance on various scale datasets and the impact of different numbers of weak annotators in ROSSEL. All experiments are conducted on a workstation with an Intel(R) CPU (Xeon(R) E5-2687W v2 @ 3.40GHz) and 32 GB memory.

3.4.1 Datasets

Five UCI datasets including CNAE9, dna, connect4, protein and rcv1 are used in the experiments. Among them, CNAE9 and dna are two small scale datasets that every competing method is able to handle. Protein, connect4 and rcv1 are large scale datasets which are used to investigate both the accuracy and scalability of competing methods. The size of connect4 and rcv1, which contain 67,557 and 534,135 samples respectively, is very large for SSL algorithms. Consequently, for the convenience of comparison on connect4, I generate a new dataset called connect4-10k by sampling 10,000 instances from connect4 at random. I report results of the rcv1 dataset on both

the standard training set and the full set.

In all experiments, to simulate the SSL scenario, I randomly sample three disjointed subsets from each dataset as the labeled set (5% samples), unlabeled set (75% samples) and test set (20%). More information about the six datasets is listed in Table 3.2. I report accuracy as the evaluation metric for comparison in all tables and figures.

3.4.2 Resistance to Label Noise

In this experiment, I investigate the resistance of SSL algorithms to label noise on the CNAE9 and dna datasets. I randomly select 2%, 4%, ..., 10% labels from the labeled set and switch them to wrong labels as label noise. The resultant accuracy reported in Figure 3.2 demonstrates that our algorithm can be more resistant to label noise than other baselines used in the experiment.

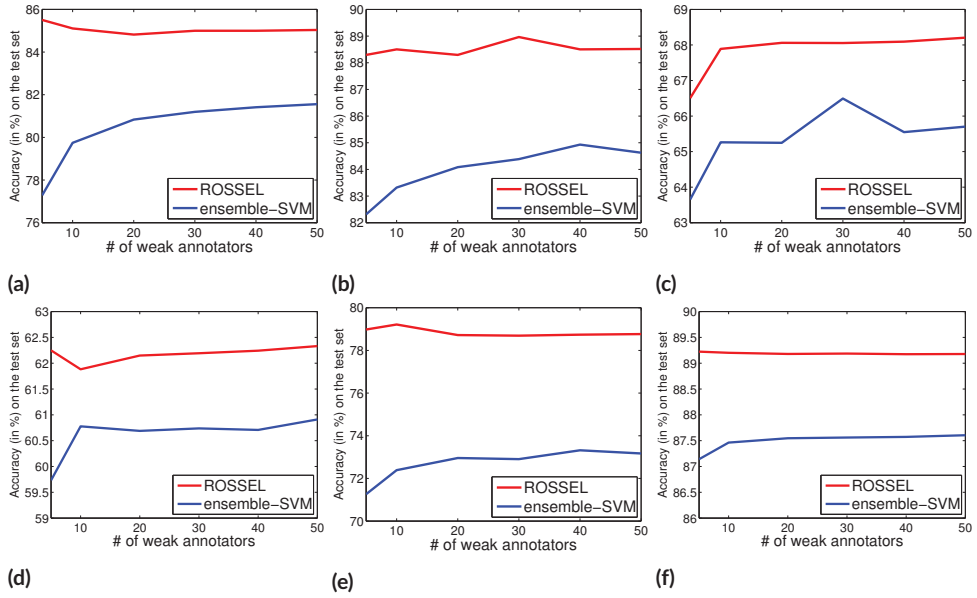


Figure 3.3: Average accuracy over 10 runs on various datasets with different number of weak annotators.

3.4.3 Comparison of Accuracy and Scalability

In this experiment, I investigate the accuracy and scalability of SSL algorithms. I compare the proposed algorithm with eight other methods, including three supervised learning algorithms and five SSL methods. The three supervised learning baselines are listed as below:

- LIBLINEAR [43] is a supervised linear SVM baseline, efficient for large scale data. In the experiment, I tune two types of SVM including L2-regularized L2-loss and L2-regularized L1-loss and report the best results. I apply the one-vs-all strategy for all experiments.

- LIBSVM [24] is a supervised non-linear SVM baseline, which is usually slower than LIBLINEAR when kernels are present. In the experiment, I tune various kernels, including the linear kernel, polynomial kernel, Gaussian kernel and sigmoid kernel. I apply the one-vs-all strategy for all experiments.

- Ensemble-SVM is an ensemble supervised learning baseline, by which I demonstrate the effectiveness of the proposed SSL method. Each of the base classifier is trained by LIBLINEAR on a bootstrap replicate. The predicted label on a test instance is computed by plurality voting of all base classifier.

The five SSL competing methods are listed as follows:

- LapSVM [9] is a graph-based SSL algorithm. The objective function of SVMs is regularized by graph Laplacian.

- LapRLS [9], similar to LapSVM, is regularized by graph Laplacian. The objective function is based on the least squared loss.

- meanS3VM [76], instead of estimating the label of each unlabeled data, exploits the label means of unlabeled data, and maximizes the margin between the label means.

- CS4VM [75] is a cost-sensitive semi-supervised SVM algorithm, which treats various misclassification errors with different costs.

- ASL [128] is a recently proposed SSL method that avoids expensive graph construction and adaptively adjusts the weights of data, which can be robust to boundary points.

In this experiment, I use Gaussian kernel $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2})$ to compute the kernel matrix. The kernel parameter σ is fixed as 1, and all feature matrices are normalized before

the experiment. I select from the range of $\{10^{-5}, 10^{-3}, 10^{-1}, 10^{0}, 10^{1}, 10^{3}, 10^{5}\}$ for the parameters to be tuned in all methods. I empirically set the parameter k-nearest neighbour as 5 for the graph-based methods, LapSVM and LapRLS.

Ensemble-SVM and ROSSEL are two ensemble based methods. In this experiment, I report the results of these two methods with both 10 and 50 weak annotators. When sampling, I bootstrap 50% labeled data into a bootstrap replicate.

For comparison, I perform the experiment 10 times on various splits of the labeled, unlabeled and test sets. Average accuracy on the test set and average training time of all competing methods over 10 runs are reported in Table 3.3 and Table 3.4 respectively. Results in the two tables demonstrate that the proposed method is very competitive in terms of accuracy and scalability. SSL algorithms usually suffer from poor scalability. As can be seen from the results, even on the full rcv1 dataset that contains more than 400,000 training examples with 47,236 features, ROSSEL provides promising accuracy within much less training time.

3.4.4 Impact of Number of Weak Annotators

In this experiment, I study the effect of various numbers of weak annotators used in the two ensemble based methods, ROSSEL and ensemble-SVM. I perform this experiment on all the six datasets.To investigate the influence of different numbers of weak annotators, 5, 10, 20, 30, 40, 50 weak annotators are used in these two methods. I run the experiment over 10 different splits of labeled, unlabeled and the test sets. The accuracy on test data of different numbers of weak annotators of the two algorithms is reported in Figure 3.3.

As observed, ensemble-SVM usually performs better with more weak annotators. However, our method with different numbers of weak annotators gives very close performance. This observation demonstrates that our algorithm is stable and will provide competitive performance even when there are a small number of weak annotators involved.

3.5 Conclusions

SSL is proposed to improve the performance by exploiting both labeled data and unlabeled data. It plays an increasingly crucial role in practical applications due to the rapid boosting of the volume of data. However, conventional SSL algorithms usually suffer from the poor efficiency and may degenerate remarkably when label noise is present. To address these two challenges, I propose ROSSEL to approximate ground-truth labels for unlabeled data though the weighted aggregation of pseudo-labels generated by low-cost weak annotators. Meanwhile ROSSEL trains an inductive SSL model. I formulate the label aggregation problem as a multiple label kernel learning (MLKL) problem which can be solved very efficiently. The complexity of ROSSEL is much lower than related SSL methods. Extensive experiments are performed on five benchmark datasets to investigate the robustness, accuracy and efficiency of SSL methods.

4

Active Learning for Image Classification by Privileged Information

4.1 Introduction

This chapter [*] focuses on active learning for image classification by privileged information. Active learning aims to enable the interaction between the algorithms and users, by which the human labor for labeling unlabeled data can be significantly reduced. To ensure the unlabeled

[*] The main results of this chapter were previously published in Yan Yan, Feiping Nie, Wen Li, Chenqiang Gao, Yi Yang, Dong Xu. "Image Classification by Cross-Media Active Learning with Privileged Information." *IEEE Transactions on Multimedia* 18, no. 12 (2016): 2494-2502.

samples selected by the active learning algorithm to be representative, I exploit the diversity measurement, such that the selected samples are less similar to each other. I formulate a ratio objective function to maximize cross-media uncertainty and minimize the similarity of selected data. Then I propose to measure uncertainty and diversity for training sample selection [152]. A new optimization method is proposed to solve the proposed model, which automatically learns the optimal ratio of uncertainty to similarity. In this way, I avoid introducing the trade-off parameter between the two types of measurements. I summarize the main contributions of this chapter as follows:

- By exploiting privileged information, I propose a new notion of cross-media uncertainty measurement, which measures the uncertainty of unlabeled images by jointly considering visual features as the main information and text features as the privileged information.
- I propose a new method to optimize the objective without using the trade-off parameter between diversity and uncertainty.

4.2 The Proposed Model

In our task, the training data consists of an active seed set containing a few labeled samples, and a pool set containing unlabeled samples. I aim to select the most useful unlabeled samples, and query the Oracle to label them. Thus, I have a new training set for training.

Let us denote

$$\{(\mathbf{x}_1, \tilde{\mathbf{x}}_1, \mathbf{y}_1), (\mathbf{x}_2, \tilde{\mathbf{x}}_2, \mathbf{y}_2), ..., (\mathbf{x}_{n_s}, \tilde{\mathbf{x}}_{n_s}, \mathbf{y}_{n_s})\}$$

as the seed set and

$$\{(\mathbf{x}_{n_s+1}, \tilde{\mathbf{x}}_{n_s+1}), (\mathbf{x}_{n_s+2}, \tilde{\mathbf{x}}_{n_s+2}), ..., (\mathbf{x}_{n_s+n_p}, \tilde{\mathbf{x}}_{n_s+n_p})\}$$

as the pool set, where n_s and n_p represent the numbers of data in the active seed set and pool set respectively. For the i-th instance, I denote $\mathbf{x}_i \in \mathbb{R}^d$ as the main feature and $\tilde{\mathbf{x}}_i \in \mathbb{R}^{\tilde{d}}$ as the

privileged feature, and also denote $\mathbf{y}_i \in \{-1, +1\}^t$ as the label, where t is the number of tasks or classes.

A typical active learning procedure consists of two main phases. The first step is to train an initial model (*e.g.*, a classifier or a regression predictor) based on the labeled data, which can be used to measure the importance of the unlabeled data. In the second step, I generate the ranking scores for all unlabeled samples based on the importance measure. Given the ranking scores, the unlabeled samples that would be incorporated to the training set can thus be simply determined. I follow this process, and propose a new active learning algorithm with privileged information.

Besides the uncertainty, the proposed model additionally consider the diversity of the queried unlabeled data. While uncertain samples would benefit the performance, the underlying distribution of unlabeled data could not always be presented correctly by a few labeled data. I would miss some important information if the attention is only paid to the most uncertain samples. Therefore, I propose to find the samples that best explain the distribution of the data at the same time with considering the most uncertain ones. This can be intuitively achieved by making the unlabeled query samples as dissimilar as possible. In the second phase, I propose to consider the diversity based on the similarity matrix of the unlabeled data when computing the ranking scores. The similarity measurement can be obtained from the kernel matrix. I propose an efficient optimization method to solve the objective function with uncertainty and similarity jointly.

4.2.1 Uncertainty Measurement with Privileged Information

Privileged Information and Uncertainty

In the real world data, particularly Web images, auxiliary information such as text information is often approachable. LUPI is proposed in [126] as a new learning paradigm. LUPI assumes that additional features, namely privileged information, are contained in the training phase, but not in the test data. It is similar to the teacher in a class who offers extra explanations to students. On the Internet, people also tend to write some additional texts to facilitate the management of

their multimedia repository, which usually includes images and videos. The text could provide more detailed descriptions for understanding the visual content in their repository. Several computer vision tasks like image classification can be benefited from the surrounding texts of web images. In this work, I show the learned model can benefit from the additional text information associated with web images during the learning procedure.

However, the auxiliary text data usually cannot be applied in the image training procedure directly since it is in another feature space. To involve such privileged information when learning, in [126], the authors introduce slack functions into the formulation of a non-separable support vector machine (SVM) as follow

$$\begin{aligned} &\min_{\mathbf{w},\tilde{\mathbf{w}},b,\tilde{b}} \frac{1}{2}(||\mathbf{w}|| + \gamma||\tilde{\mathbf{w}}||) + C\sum_{i=1}^{n_s}(\tilde{\mathbf{w}}^\top\tilde{\varphi}(\tilde{\mathbf{x}}_i) + \tilde{b}) \\ &s.t. \quad y_i(\mathbf{w}^\top\varphi(\mathbf{x}_i) + b) \geq 1 - (\tilde{\mathbf{w}}^\top\tilde{\varphi}(\tilde{\mathbf{x}}_i) + \tilde{b}), \quad i = 1, ..., n_s, \\ &\qquad (\tilde{\mathbf{w}}^\top\tilde{\varphi}(\tilde{\mathbf{x}}_i) + \tilde{b}) \geq 0, \quad i = 1, ..., n_s, \end{aligned} \tag{4.1}$$

where $\varphi(\mathbf{x}_i)$ is the feature mapping function for main information, and $\tilde{\varphi}(\tilde{\mathbf{x}}_i)$ is the feature mapping function for privileged information. C is the trade-off parameter between data loss and model regularization, and γ is the trade-off parameter between the influence of main information and privileged information. $\mathbf{w}$ and $\tilde{\mathbf{w}}$ are the weight vectors for main features and privileged features respectively. The above problem is rather similar to a non-separable SVM problem, which can be formulated as

$$\begin{aligned} &\min_{\mathbf{w},b} = \frac{1}{2}||\mathbf{w}|| + C\sum_{i=1}^{n_s}\xi_i \\ &s.t. \quad y_i(\mathbf{w}^\top\mathbf{z}_i + b) \geq 1 - \xi_i, \quad i = 1, ..., n_s, \\ &\qquad \xi_i \geq 0, \quad i = 1, ..., n_s. \end{aligned} \tag{4.2}$$

The difference between LUPI and the conventional SVM comes with the slack variable ξ_i. In SVM, the slack variables can be optimized by the quadratic solver. In LUPI, the slack variables are replaced by the slack function $\hat{\xi}_i = (\tilde{\mathbf{w}}^\top\tilde{\varphi}(\tilde{\mathbf{x}}_i) + \tilde{b})$. This slack function is defined for the

correcting (text) feature instead of the main (visual) feature.

LUPI aims to determine the value of the slack function by leveraging the privileged information, rather than directly learning slack variables for the main feature. Although privileged information is not in the same space with the principal feature, it can still assist to obtain slack variables. If the learned slack function yields a larger value for the sample $\mathbf{x}_i$, then $\mathbf{x}_i$ is allowed to maintain a larger distance to the decision boundary. In other words, this implies that it would be more difficult for the learner to classify this sample correctly. Hence it is natural to measure the uncertainty of instances by the learned slack function. For example, suppose there are two samples m_1 and m_2 and $\hat{\xi}_1$ and $\hat{\xi}_2$ are their corresponding values returned by the learned slack function. If $\hat{\xi}_1 > \hat{\xi}_2$ then m_1 is likely more uncertain to classify correctly.

Cross-media Uncertainty Measurement

Our method measures cross-media uncertainty by simultaneously learning from images and their surrounding texts. Next, I detail the strategies of cross-media uncertainty measurement.

It is natural to obtain two simple ways to measure the uncertainty. The first one is based on the prediction confidence associated with the predictor according to the visual feature, which is a traditional measurement, while the other one exploits the correcting function of text features.

- Prediction confidence uncertainty measurement is based on the predictions of the visual feature, which is commonly used in many active sample selection algorithms. In a multi-class scenario, as mentioned, several strategies can be applied to generate the uncertainty. I focus on the margin sampling, which is a simple and effective sampling approach, and can be written as

$$p_i = 1/(\hat{y}_{i,k_1} - \hat{y}_{i,k_2} + \varepsilon), \tag{4.3}$$

  where $\hat{y}_{i,k_j} = \mathbf{w}_{k_j}^\top \varphi(\mathbf{x}_i) + b_{k_j}$ is the prediction of the i-th unlabeled sample, p_i presents the corresponding uncertainty measurement, $\hat{y}_{i,k_1} \geq \hat{y}_{i,k_2} \geq \ldots \geq \hat{y}_{i,k_K}$ are the predicted values of the pool data $\mathbf{x}_i$ for K classes which are sorted in descending order, k_i presents the sorted index of classes, and $1 \leq i \leq n_p$. ε is a small constant that avoids division

by zero. There are also other uncertainty sampling approaches. For instance, *least confidence* [107] strategy simply samples the data with least prediction confidence. However, this approach only considers the least confident class and ignores other classes. To exploit the remaining label distribution, margin sampling method considers more information about other class labels.

- Correcting function uncertainty measurement is based on the optimized slack function of text features. As mentioned, the value of the slack function implies how difficult a sample may be correctly classified by the model. Hence, after training the LUPI model, U simply use the correcting function $p_i = \tilde{\xi}_i = (\tilde{\mathbf{w}}^\top \tilde{\varphi} \tilde{\mathbf{x}}_i + \tilde{b})$ to measure the uncertainty of unlabeled data $\tilde{\mathbf{x}}_i$ in the privileged feature.

Suppose $\mathbf{p}_{pred} \in \mathbb{R}^{n_p}$ and $\mathbf{p}_{corr} \in \mathbb{R}^{n_p}$ denote the prediction confidence uncertainty measurement and correcting function uncertainty measurement on pool data. Based on these two simple uncertainty measurements I present five strategies to combine them together, which are listed as follows.

- LUPI-sum: element-wise sum of two measurements. I use the sum of them as the uncertainty measurement: $\mathbf{p}_{sum} = \mathbf{p}_{corr} + \mathbf{p}_{pred}$.
- LUPI-max: element-wise maximum of two measurements. I use the max of them as the uncertainty measurement: $\mathbf{p}_{max} = \max(\mathbf{p}_{corr}, \mathbf{p}_{pred})$.
- LUPI-min: element-wise minimum of two measurements. I use the min of them as the uncertainty measurement: $\mathbf{p}_{min} = \min(\mathbf{p}_{corr}, \mathbf{p}_{pred})$.
- LUPI-pro: Hadamard product of two measurements. I use the Hadamard product of them as the uncertainty measurement: $\mathbf{p}_{had} = \mathbf{p}_{corr} \odot \mathbf{p}_{pred}$, where $\odot$ denotes the Hadamard product.
- LUPI-dis: element-wise distance of two measurements. I use the distance of them as the uncertainty measurement, namely $\mathbf{p}_{sub} = |\mathbf{p}_{corr} - \mathbf{p}_{pred}|$, where $|\cdot|$ computes the element-wise absolute values.

In this chapter, I study all of the above five strategies.

4.2.2 Active Sampling with Uncertainty and Similarity Measurement

Typically, active sample selection aims to sample the most uncertain instances from the active pool set which most confuse the trained classifiers. In this way, the decision boundary can be refined and hopefully closer to optimal. However, as the number of initial labeled data only accounts for a small proportion of the entire data, there could be sample bias when selecting unlabeled samples from active pool set for labeling [62]. Thus, to achieve promising performance, researchers propose to combine more strategies together. For example, in [62], the authors consider both uncertainty and representativeness. In [138], multiple criteria are taken into consideration.

Similar to previous works, to obtain better performance, I consider to combine the uncertainty component and diversity component together in this chapter. A very simple and straightforward way is to optimize an objective function that combines various components by trade-off parameters, which is similar to other works [62, 138, 152]. This objective function may be written as

$$\min \mathit{Uncert} + \lambda_1 \mathit{Similar},$$

where *Uncert* and *Similar* denote some measurement of uncertainty and similarity respectively. However, in the real world, it is not practical to tune the trade-off parameter λ_1 very well for all datasets. Therefore, in this chapter, I propose a ratio objective function which computes ranking scores for all unlabeled data that maximizes the uncertainty, and meanwhile minimizes the similarity. This hence avoids tuning the trade-off parameter. Our proposed objective function is as follow

$$\begin{aligned} &\max_{\mathbf{r}} \frac{\mathbf{r}^\top \mathbf{p}}{\mathbf{r}^\top \mathbf{A}\mathbf{r}} \\ &s.t. \quad \sum_{i=1}^{n_p} r_i = 1, r_i \geq 0, \end{aligned} \tag{4.4}$$

where $\mathbf{r} \in \mathbb{R}^{n_p}$ is the ranking score vector for samples in active pool set, $\mathbf{p} \in \mathbb{R}^{n_p}$ is the uncer-

tainty measurement computed by one of the strategies in 4.2.1 and $\mathbf{A} \in \mathbb{R}^{n_p \times n_p}$ is the kernel matrix of the samples in active pool set, which represents the similarity among the unlabeled samples. Let $\lambda = \frac{\mathbf{r}^\top \mathbf{p}}{\mathbf{r}^\top \mathbf{A}\mathbf{r}}$ denotes the ratio of the uncertainty component to the similarity component. Here I use the radial basis function kernel. By this objective function, a higher ranking score tends to be assigned to a sample with higher uncertainty and less similarity with other data. In the next section, I propose an optimization scheme to solve the objective function efficiently.

When solving this objective function, it is not necessary to consider the entire pool set. In other words, after obtaining the uncertainty vector $\mathbf{p}$, I can find a specific portion of unlabeled data and input them into Problem 4.4. For instance, there are 1,000 unlabeled data in the pool set, while the target number of queries is 100. I may only use the most uncertain 200 instances from the pool set, instead of using the full uncertainty vector $\mathbf{p}$ and the entire kernel matrix $\mathbf{A}$. Since the size of the pool set is often huge in real world, this method is natural to speed up the optimization.

4.2.3 A Brief Overview of Augmented Lagrangian Method

I present an efficient optimization method based on Augmented Lagrangian method (ALM) [12] in the next subsection. ALM is to solve the following problem:

$$\begin{aligned} \min_{T} \quad & g(T) \\ \text{s.t.} \quad & h(T) = 0, \end{aligned} \tag{4.5}$$

where $g : \mathbb{R}^t \to \mathbb{R}$, $h : \mathbb{R}^t \to \mathbb{R}^s$ and $T \in \mathbb{R}^t$ is the optimization variable. To solve the above constrained problem, one can construct an augmented Lagrangian function as

$$L(T, Z, \mu) = g(T) + \langle Z, h(T) \rangle + \frac{\mu}{2} ||h(T)||_F^2, \tag{4.6}$$

where Z is the Lagrangian coefficient and μ is a scalar. The general approach to update T, Z and μ is briefed in Algorithm 5.

Algorithm 5 An overview of ALM algorithm [12]

Initialize $\varrho > 1$, $t = 0$, $Z_t = 0$, and $\mu_t > 0$.
1: $T_t = \arg\min_{T, Z_t, \ \ t}$.
2: $Z_{t+1} = Z_t + \mu_t h(T_{t+1})$.
3: $\mu_{t+1} = \varrho \mu_t$.
4: t = t + 1.
5: Go to step 1 until convergence.

4.2.4 Optimization of Problem (4.4)

The objective function is the ratio of the uncertainty to the similarity and it is not feasible to directly maximize the ratio. In this section, I propose an optimization approach to update this ratio, λ, iteratively, which is summarized in Algorithm 6. By this optimization approach, I can obtain ranking scores of unlabeled data that maximizes the uncertainty and minimizes the similarity meantime.

As illustrated in Algorithm 6, to start with, I exploit the ratio variable λ to rewrite Problem 4.4 as follow, which is a subproblem of our objective function,

$$\begin{aligned} &\min_{\mathbf{r}} \lambda \mathbf{r}^\top \mathbf{A}\mathbf{r} - \mathbf{r}^\top \mathbf{p} \\ &s.t. \quad \sum_{i=1}^{n_p} r_i = 1, r_i \geq 0. \end{aligned} \tag{4.7}$$

The constraints on $\mathbf{r}$ aims to limit the scale of the ranking scores. This objective function is now a general form that combines the uncertainty component and the diversity component by the variable λ. Problem 4.7 is a quadratic programming problem, but for the efficiency, I propose to use a faster optimization method based on augmented Lagrange multiplier (ALM) [12, 36],

which is analogous to [152]. Introducing a new variable v, Problem 4.7 can be rewritten as

$$\begin{aligned} &\min_{\mathbf{r}} \frac{1}{2}\mathbf{r}^\top \hat{\mathbf{A}}\mathbf{r} + \mathbf{r}^\top \hat{\mathbf{p}} \\ &s.t. \quad \mathbf{r}^\top 1_{n_p} = 1, \mathbf{r} = \mathbf{v}, \mathbf{v} \succeq 0, \end{aligned} \tag{4.8}$$

where $1_{n_p} \in \mathbb{R}^{n_p}$ is a vector with all elements of 1, $\hat{\mathbf{A}} = 2\lambda\mathbf{A}$ and $\hat{\mathbf{p}} = -\mathbf{p}$.

Then I can obtain the augmented Lagrangian function of Problem 4.8 as below

$$\begin{aligned} &L(\mathbf{r}, \mathbf{v}, \mu, \delta, \gamma) \\ =&\langle \delta, \mathbf{r}^\top 1_{n_p} - 1\rangle + \langle \gamma, \mathbf{r} - \mathbf{v}\rangle \\ &+ \frac{\mu}{2}||\mathbf{r}^\top 1_{n_p} - 1||_F^2 + \frac{\mu}{2}||\mathbf{r} - \mathbf{v}||_F^2 + \frac{1}{2}\mathbf{r}^\top \hat{\mathbf{A}}\mathbf{r} + \mathbf{r}^\top \hat{\mathbf{p}} \\ =&\frac{\mu}{2}[||\mathbf{r} - \mathbf{v}||_F^2 + \frac{2}{\mu}\langle \gamma, \mathbf{r} - \mathbf{v}\rangle + \frac{\gamma^2}{\mu^2}] \\ &\frac{\mu}{2}[||\mathbf{r}^\top 1_{n_p}||_F^2 + \frac{2}{\mu}\langle \delta, \mathbf{r}^\top 1_{n_p} - 1\rangle + \frac{\delta^2}{\mu^2}] \\ &+ \frac{1}{2}\mathbf{r}^\top \hat{\mathbf{A}}\mathbf{r} + \mathbf{r}^\top \hat{\mathbf{p}} + \frac{\gamma^2}{2\mu} + \frac{\delta^2}{2\mu} \\ =&\frac{\mu}{2}(\mathbf{r}^\top 1_{n_p} - 1 + \frac{1}{\mu}\delta)^2 + \frac{\mu}{2}||\mathbf{r} - \mathbf{v} + \frac{1}{\mu}\gamma||_F^2 \\ &+ \frac{1}{2}\mathbf{r}^\top \hat{\mathbf{A}}\mathbf{r} + \mathbf{r}^\top \hat{\mathbf{p}} + \frac{\gamma^2}{2\mu} + \frac{\delta^2}{2\mu} \end{aligned} \tag{4.9}$$

where $\mathbf{v} \succeq 0$, δ and γ are the Lagrangian coefficients, and μ is a scalar. According to [12], the Lagrangian function and the original problem have the same local minimization solution. Let

$$\tilde{\mathbf{A}} = \hat{\mathbf{A}} + \mu 1_{n_p} 1_{n_p}^\top + \mu \mathbf{I}_{n_p}$$

and

$$\tilde{\mathbf{p}} = \mu\mathbf{v} + \mu 1_{n_p} - \hat{\mathbf{p}} - \delta 1_{n_p} - \gamma,$$

and then the Lagrangian function can be

$$L(\mathbf{r}, \mathbf{v}, \mu, \delta, \gamma) = \frac{1}{2}\mathbf{r}^\top \tilde{\mathbf{A}}\mathbf{r} - \mathbf{r}^\top \tilde{\mathbf{p}} + \frac{\mu}{2}(-1 + \frac{\delta}{\mu})^2 + \frac{\mu}{2}|| - \mathbf{v} + \frac{\gamma}{\mu}||_F^2. \tag{4.10}$$

By setting the derivative of $L(\mathbf{r}, \mathbf{v}, \mu, \delta, \gamma)$ w.r.t. $\mathbf{r}$ as zero, the objective function can be solved as follow

$$\mathbf{r}^* = \tilde{\mathbf{A}}^{-1}\tilde{\mathbf{p}}. \tag{4.11}$$

After updating the ranking score vector $\mathbf{r}$, I need to update the auxiliary variable $\mathbf{v}$ by

$$\min_{\mathbf{v}\geq 0} L(\mathbf{r}, \mathbf{v}, \mu, \delta, \gamma) \Rightarrow \min_{\mathbf{v}\geq 0} ||\mathbf{v} - (\mathbf{r} + \frac{1}{\mu}\gamma)||^2. \tag{4.12}$$

Specifically, I can solve this problem by

$$v_i = \max(0, r_i + \frac{1}{\mu}\gamma_i), \quad 1 \leq i \leq n_p. \tag{4.13}$$

Once completing the ALM subproblem and obtaining the resultant ranking scores $\mathbf{r}^*$, I consider to update the ratio variable λ as follow

$$\lambda = \frac{\mathbf{r}^{*\top}\mathbf{p}}{\mathbf{r}^{*\top}\mathbf{A}\mathbf{r}^*}. \tag{4.14}$$

I repeat the process for optimizing $\mathbf{r}$ and updating λ until convergence.

Theorem 1. *Algorithm 6 increases monotonously in each iteration until the algorithm converge.*

Proof. Let $\mathbf{r}_{k+1}$ be the updated $\mathbf{r}$ in the iteration $k+1$, $\mathbf{r}_k$ be the $\mathbf{r}$ computed in the iteration k and λ_k be the λ computed in the iteration k. Firstly, according to the step 2 in Algorithm 6, I have

$$\lambda_k \mathbf{r}_k^\top \mathbf{A}\mathbf{r}_k - \mathbf{r}_k^\top \mathbf{p} = 0.$$

Since $\mathbf{r}^*$ is the solution of Problem (4.7), it clear that $\lambda_k \mathbf{r}_{k+1}^\top \mathbf{A}\mathbf{r}_{k+1}^\top - \mathbf{r}_{k+1}^\top \mathbf{p} \leq \lambda_k \mathbf{r}_k^\top \mathbf{A}\mathbf{r}_k^\top -$

$\mathbf{r}_k^\top \mathbf{p} = 0$. Therefore, I can easily obtain

$$\frac{\mathbf{r}_{k+1}^\top \mathbf{p}}{\mathbf{r}_{k+1}^\top \mathbf{A}\mathbf{r}_{k+1}} \geq \lambda_k = \frac{\mathbf{r}^\top \mathbf{p}}{\mathbf{r}^\top \mathbf{A}\mathbf{r}}.$$

□

Theorem 2. *Let*

$$g(\lambda) = \min_{\mathbf{r}^\top \mathbf{1}_{np}, \mathbf{r} \geq 0} \lambda \mathbf{r}^\top \mathbf{A}\mathbf{r} - \mathbf{r}^\top \mathbf{p}.$$

If $g(\lambda^*) = 0$*, then* λ^* *is the global solution of problem 4.4.*

Proof. From $g(\lambda^*) = 0$, I can easily obtain

$$\min_{\mathbf{r}^\top \mathbf{1}_{np}, \mathbf{r} \geq 0} \lambda^* \mathbf{r}^\top \mathbf{A}\mathbf{r} - \mathbf{r}^\top \mathbf{p} = 0.$$

Hence, for all $\mathbf{r}$,

$$\begin{aligned} &\lambda^* \mathbf{r}^\top \mathbf{A}\mathbf{r} - \mathbf{r}^\top \mathbf{p} \geq 0 \\ \Rightarrow &\frac{\mathbf{r}^\top \mathbf{p}}{\mathbf{r}^\top \mathbf{A}\mathbf{r}} \leq \lambda^*. \end{aligned} \tag{4.15}$$

Thus λ^* is the global solution. □

Theorem 3. *Algorithm 6 can converge to the global solution.*

Proof. Let λ^* and $\mathbf{r}^*$ are the values for λ and $\mathbf{r}$ when Algorithm 6 converges. According to the step 2 of Algorithm 6, I have

$$\lambda^* = \frac{\mathbf{r}^\top \mathbf{p}^*}{\mathbf{r}^* \mathbf{A}\mathbf{r}^*}.$$

Based on the step 1 of Algorithm 6, I have

$$\mathbf{r}^* = \arg \min_{\mathbf{r}^\top \mathbf{1}_{np}, \mathbf{r} \geq 0} \lambda^* \mathbf{r}^\top \mathbf{A}\mathbf{r} - \mathbf{r}^\top \mathbf{p}.$$

Thereby $g(\lambda^*) = 0$. According to Theorem 2, λ^* is the global solution. □

Algorithm 6 ALM based optimization to solve the problem in (4.4).

Initialize r with random. Set $\lambda = \frac{\mathbf{r}^\top \mathbf{p}}{\mathbf{r}^\top \mathbf{A}\mathbf{r}}$. It is obvious that $0 \leq \lambda \leq \lambda^*$ where λ^* is the optima of λ.
1: Update $\mathbf{r}$ by solving problem 4.7
1.1 Set $\hat{\mathbf{A}} = 2\lambda\mathbf{A}$ and $\hat{\mathbf{p}} = -\mathbf{p}$.
1.2 Set $\tilde{\mathbf{A}} = \hat{\mathbf{A}} + \mu 1_{n_p} 1_{n_p}^\top + \mu \mathbf{I}_{n_p}$.
1.3 Set $\tilde{\mathbf{p}} = \mu \mathbf{v} + \mu 1_{n_p} - \hat{\mathbf{p}} - \delta 1_{n_p} - \gamma$.
1.4 Set $\mathbf{r}^* = \tilde{\mathbf{A}}^{-1}\tilde{\mathbf{p}}$.
2: Update λ by $\lambda = \frac{\mathbf{r}^\top \mathbf{p}}{\mathbf{r}^\top \mathbf{A}\mathbf{r}^\top}$.
3: Go to step 1 until convergence.

4.2.5 Application to Large Datasets

I present a simple strategy to apply our proposed method to large scale datasets. Note that our goal is to query a small number of data from the large pool set (when n_p is large). Those unlabeled instances with high certainty would not be useful in active selection. As a result, I propose to only select a small portion of the unlabeled data as candidates for query. For example, if I aim to query n unlabeled data, I can only consider the top $6n$ uncertain instances in Problem (4.4) ($6n \ll n_p$). Therefore, our proposed method can be efficient enough to handle large scale datasets.

4.3 Experiments

4.3.1 Datasets and Setups

In this section, I study the performance of the proposed active sample selection algorithm on the following four datasets which contains both image data and text data. I randomly sample 70% instances as the training set and use remaining 30% instances as the test set. In the training set, I again randomly select 10% data as the seed set and 90% data as the pool set. Four public datasets are used in experiments.

WebQueries [71] contains 71,478 images with metadata of 353 queries collected from the Internet. The text data in the metadata files are captured up to 10 words before and after the corresponding image on the HTML web page. I remove the instances that do not contain any

text data, select the 19 queries that contains more than 150 positive instances, and extract all the positive instances from these 19 queries as the new dataset, namely WebQueries19. Eventually I obtain a dataset with 19 classes and 3,323 instances and name it WebQueries19. The seed set of WebQueries19 consists of 12 instances for each categories. There are 2,099 and 996 instances in the pool set and test set, respectively.

Wikipedia articles [99] contains 2,866 Wikipedia articles including images and texts. I randomly sample 20 instances from each class into the seed set. Then I random select 1,807 instances as the pool set. The rest 860 instances are in the test set.

Pascal Sentences [98] contains 1,000 images with captions for 20 classes. There are 50 images for each class and 5 caption sentences for each image. I randomly select 8 images from every class into the seed set. The pool set consists of 620 images by random sampling. The rest 300 images are used as the test set.

MSCOCO[†] is a multi-label dataset and includes 82,783 samples in the training set, 40,504 samples in the validation set and 40,775 samples in the test set. Each instance contains an image and several caption sentences. Since the labels on the test set is not available, I use the training set and the validation set. I randomly select 15 classes that contains 3,000 to 5,000 instances from the entire dataset, and randomly sample 3,000 data from the selected subset, which is called as MSCOCO15. On MSCOCO15, there are around 240 positive labels for each class on average and the average number of labels for each instance is about 1.2. In addition, to investigate the efficiency of the proposed method and the competing baselines, I construct a larger binary dataset from MSCOCO by randomly selecting two disjoint classes, which contains 15,056 instances in total.

For the three multiclass datasets, namely WebQueries19, Wikipedia Articles and Pascal Sentences, accuracy is naturally used as the evaluation measurement. On the multi-label dataset, MSCOCO15, accuracy is not an appropriate evaluation. Hence I use average Macro F1 score and Micro F1 score over 10 runs as the evaluation.

As for the visual features, I extract CNNs features using the VGG16 model [112] and the Caffe

[†] http://mscoco.org/dataset/

package [64] to obtain the fc7 layer feature. As for the texts, I convert all the words into word vector using the public dictionary[‡], and then obtain the document-level features by the Bag-Of-Words (BoW). The dictionary size of the BoW model is set to 300.

I compare the proposed algorithm with other five baseline methods below.

- pKNN [63] is a probabilistic variant of the K-Nearest-Neighbor method. I use radial basis function kernel (RBF kernel) as its input distance measurement.
- LOD [56] is an unsupervised active sample selection method. I only compare it in Pascal Sentences dataset due to its high computational complexity.
- Quire [62] queries the unlabeled data with combining uncertainty and representativeness. I use RBF kernel as the input for Quire. I turn it to a batch mode method by querying according to ranking scores for a fair comparison.
- HSE [88] is a hierarchical subquery evaluation algorithm. The number of nearest neighbor is 10.
- Aggressive Co-testing [93] is a Co-testing active learning algorithm which adopts an aggressive strategy.
- Conservative Co-testing [93] is a Co-testing active learning algorithm which adopts a conservative strategy.
- Random strategy selects unlabeled data randomly.
- Initial results are calculated by the model trained on the initial labeled data.

For all methods, I randomly sample the seed set, pool set and test set for 10 times and report the average results. For each run, all methods query 10,20,...,100 instances into the training set and linear SVMs are trained on the selected training set. The parameters in all methods are tuned from 10^{-5} to 10^{+5}. I select LUPI-max as the uncertainty measurement (denoted as AL-LUPI-max).

[‡] https://code.google.com/p/word2vec/

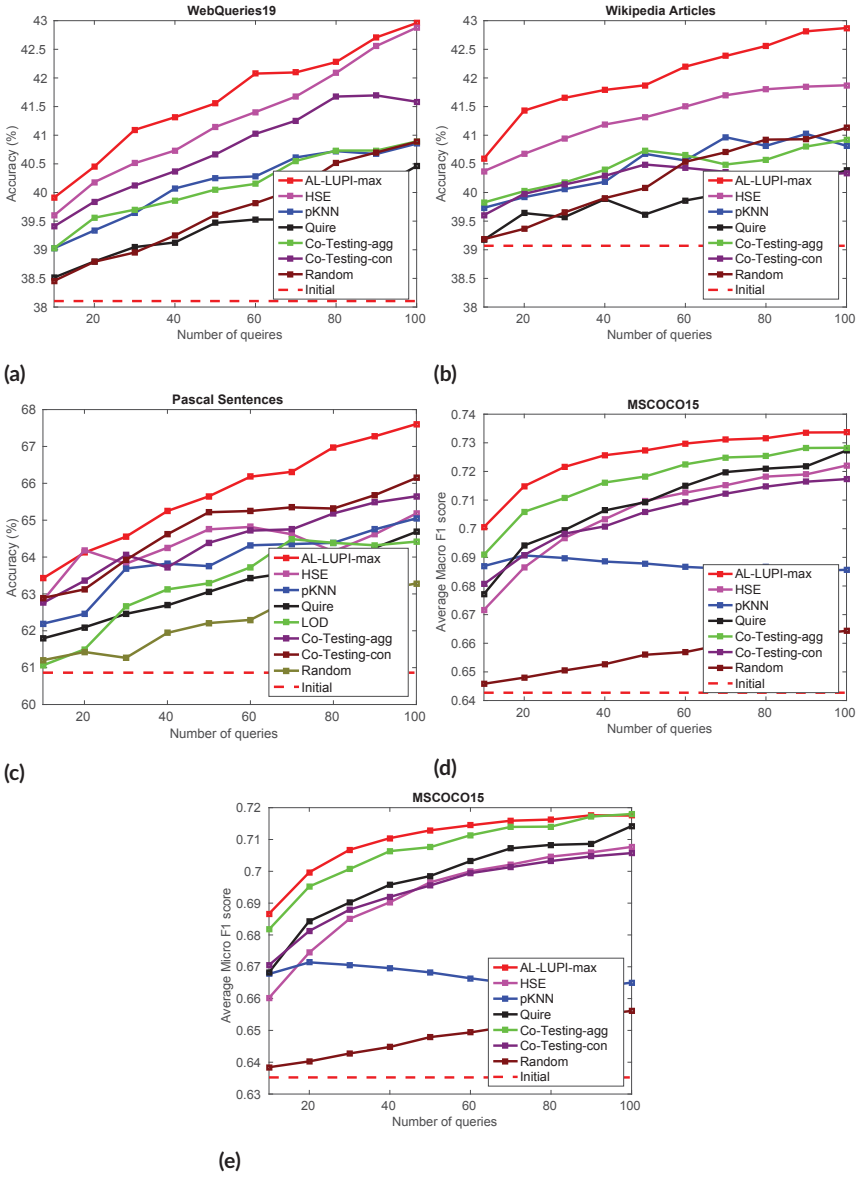


Figure 4.1: Average results on various datasets.

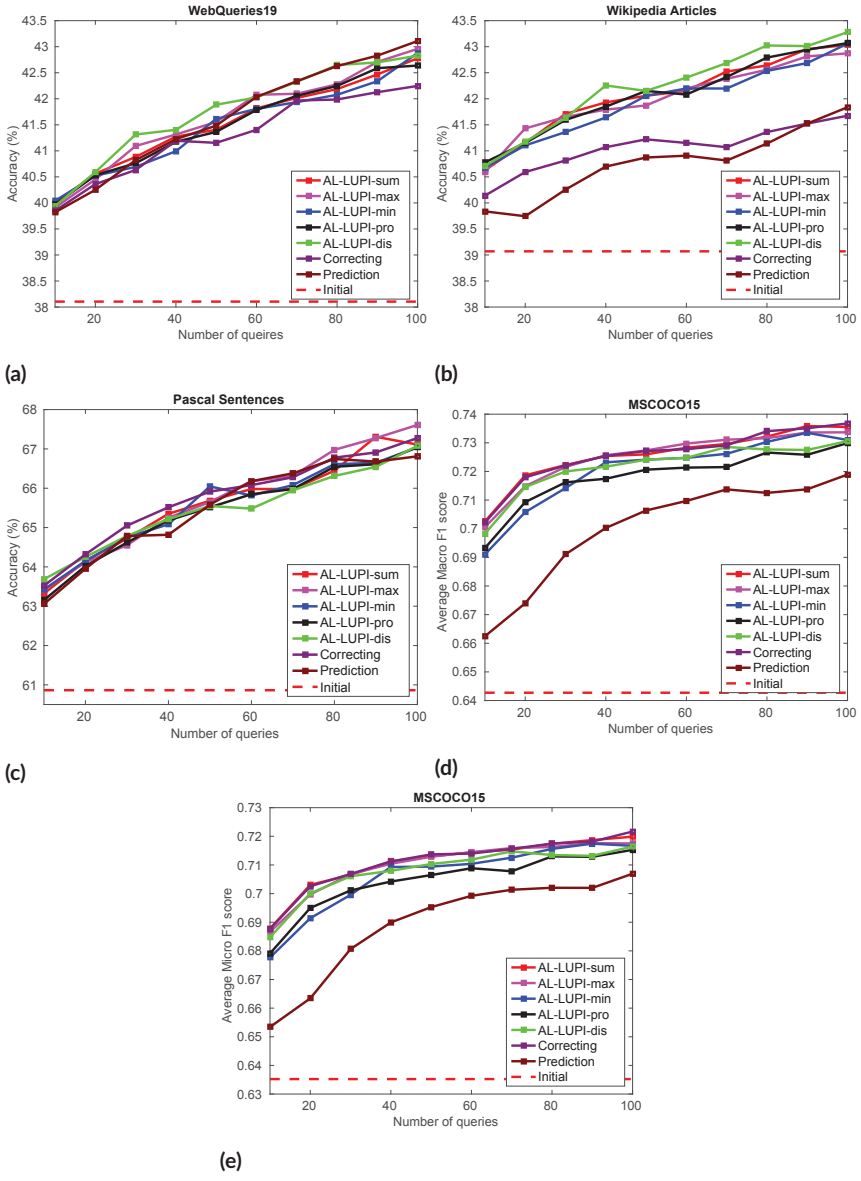


Figure 4.2: Average results on various datasets.

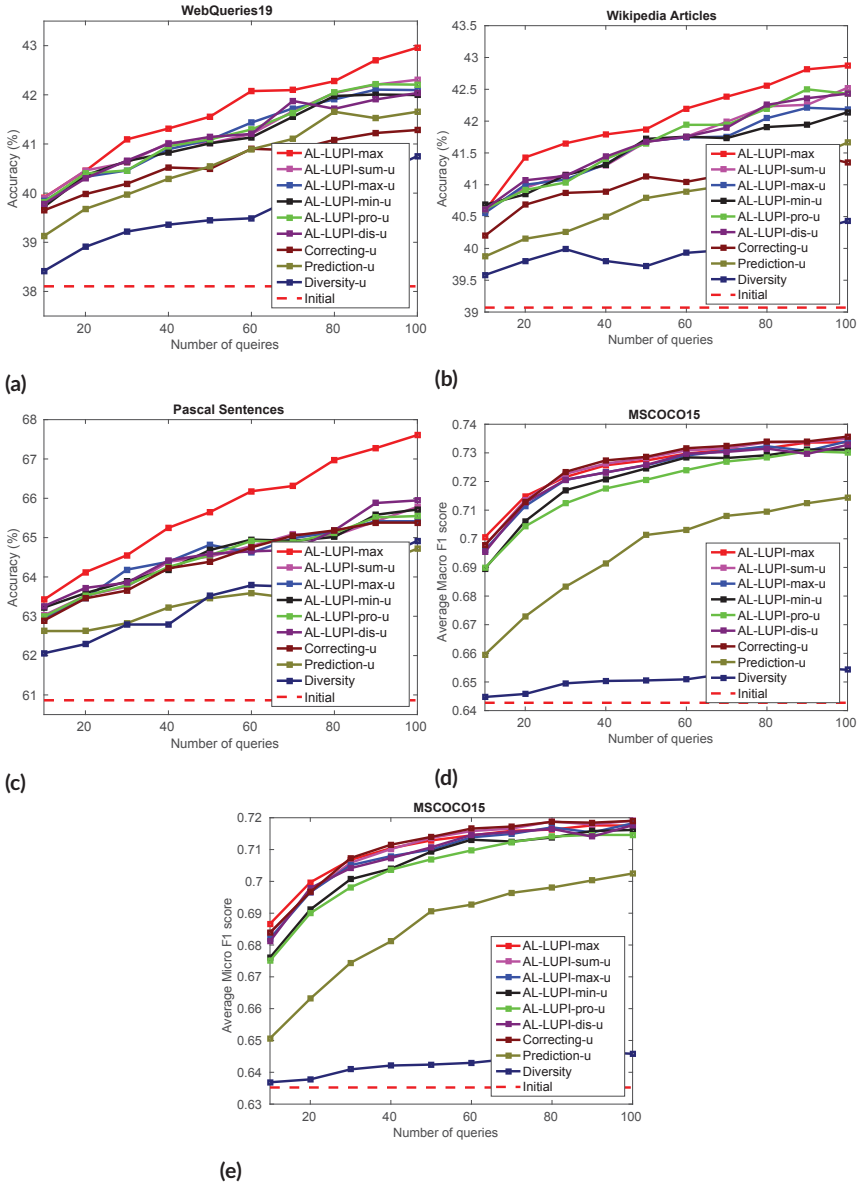


Figure 4.3: Comparison of the contribution of uncertainty and diversity.

4.3.2 Comparison with Other Active Sampling Methods

Resulting accuracy of AL-LUPI-max and competing methods on WebQueries19, Pascal Sentences and Wikipedia Articles are illustrated in Fig. 4.1a, Fig. 4.1c and Fig. 4.1b. Resulting Macro F1 score and Micro F1 score of of AL-LUPI-max and competing methods on MSCOCO15 are reported in Fig. 4.1d and Fig. 4.1e respectively.

On the three multiclass datasets, namely WebQueries19, Pascal Sentences and Wikipedia Articles, I can observe that our proposed algorithm outperforms the competing methods from the results in Fig. 4.1. Note that pKNN, LOD, Quire and HSE are the state-of-the-art active sampling algorithms in the single-view cases. Unlike these single-view methods, our proposed method, aggressive Co-testing and conservative Co-testing are based on multiple features, and often obtain better performance. This performance improvement is likely from the benefit of additional textual features.

On the MSCOCO15, which is a multi-label dataset, our method also obtain the best performance compared to all other methods in terms of both macro-F1 score and micro F1 score. Aggressive Co-testing performs second best in this experiment. These results further demonstrate that it would be beneficial to exploit multiple categories of information for multimedia tasks. Note that conservative Co-testing performs much worse than aggressive Co-testing in MSCOCO15. This would imply that it is difficult to find a strategy that always outperforms other competitors on all datasets, which is similar to the summary on different multi-view active sample selection strategies in [93] However, as can be observed, AL-LUPI-max is more stable than other algorithms on the four datasets, and achieves the best performer on all the four datasets.

4.3.3 Comparison of Various Uncertainty Measurements

To investigate the performance of the proposed five strategies of the uncertainty measurement, I compare them on the four datasets. These results can be found in Fig. 4.2. I denote other four strategies, namely LUPI-sum, LUPI-min, LUPI-pro and LUPI-dis by AL-LUPI-sum, AL-

LUPI-min, AL-LUPI-pro and AL-LUPI-dis. In general, I found that LUPI-max is stable on various datasets and achieves relatively better performance.

In addition, to investigate the contribution of privileged information in the active sampling task, I also compare two simple strategies of uncertainty measurement. The first strategy measures the uncertainty only according to the correcting function, while the other one measures the uncertainty only by the prediction confidence. I denote these two methods by "correcting" and "prediction" in Fig. 4.2. As observed, correcting function method achieves satisfactory performance on Pascal Sentences and MSCOCO15 and sometimes is competitive compared to LUPI-max. This demonstrates the effectiveness of privileged information in uncertainty measurement. However, on WebQueries and Wikipedia Articles, I observe that correcting function method performs much worse than LUPI-max. Therefore, it would be unstable if uncertainty measurement is only dependent on the privileged information. As for the prediction confidence strategy, it performs much worse than correcting function, particularly on Wikipedia Articles and MSCOCO15. Thus, I demonstrate that privileged information in the training procedure is very useful for the active sample selection task.

4.3.4 Contribution of Uncertainty and Diversity

In this section, I perform an experiment to investigate the contribution of uncertainty and diversity measurements. The proposed active sample selection algorithm samples unlabeled data based on two major measurements, namely uncertainty and diversity. To show the contribution of these two components respectively, I perform two baselines. For the first one, I drop the diversity measurement and sample unlabeled data only replying on the uncertainty measurement. Since I propose five various strategies, I denote those variants as "AL-LUPI-sum-u", "AL-LUPI-max-u", "AL-LUPI-min-u", "AL-LUPI-pro-u" and "AL-LUPI-dis-u". In addition, I examine two simple strategies which solely exploit the visual feature or text features respectively to measure uncertainty of unlabeled data. I denote them as "Prediction-u" and "Correction-u" respectively. As for the second baseline, I ignore the uncertainty measurement and only preserve diversity measurement. Then I sample unlabeled data directly according to the diversity

ranking. I show these results in Fig. 4.3.

From the results, I observe that uncertainty measurement is more effective and beneficial for improving the performance of active sample selection, compared to diversity measurement. Even compared to AL-LUPI-max, the full version of our proposed method, uncertainty based baseline achieves relatively competitive performance. For example, on MSCOCO15, these baselines outperform AL-LUPI-max slightly in some cases. However, I note that it would be unreliable to depend on only one type of measurement in the active sampling procedure. On other three datasets, AL-LUPI-max outperforms all baselines significantly. Although diversity sampling is less effective than uncertainty sampling, our results demonstrate that by combining these two strategies, I achieve significant improvement. Therefore, I could conclude that the improvement of our proposed method comes from the combination of the uncertainty information and diversity measurements, which makes our algorithm robust and effective.

4.3.5 Comparison of Efficiency

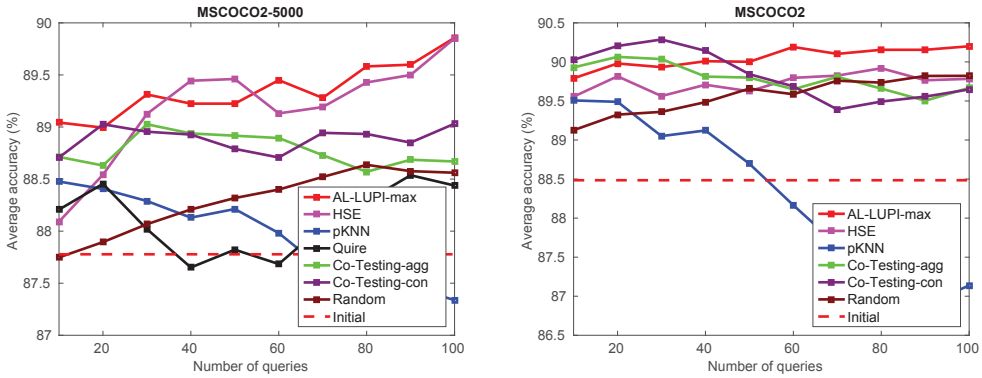
I compare our method with other active learning baselines on MSCOCO2-5000 and MSCOCO2. I report the average accuracies of different number of actively selected samples over 10 runs in Fig. 4.4. I did not compare our method with LOD since it cannot finish a single run in 12 hours on both datasets. The corresponding average training time of active learning methods is listed in Table 4.1. Note that active learning algorithm often relies on some auxiliary information, such as kernel matrices, graph or clustering. For instance, our method and pKNN requires kernel matrices on training data. Graph and cluster data are required in HSE. For the ease of comparison, Table 4.1 only contains the time spent on generating queries, instead of the time spent on the auxiliary information acquisition.

From Fig. 4.4, it is easy to observe that our method outperform other baselines in terms of average accuracy, even compared to the two multi-view baselines, namely Co-Testing-agg and Co-Testing-con. Table 4.1 shows that our algorithm is more efficient than HSE and Quire. Although pKNN is faster than our method, it cannot achieve satisfactory accuracy. I demonstrate our algorithm is efficient and can be applied to large dataset. The reasons of the low computa-

Table 4.1: Average training time (in seconds) over 10 runs.

Dataset	AL-LUPI-max	HSE	pKNN	Quire
MSCOCO2-5000	17.5666	54.4269	0.2605	159.7137
MSCOCO2	18.3564	278.6959	0.2800	*

* Quire requires expensive computational cost, and does not finish a single run in 8 hours. Therefore, I did not include the training time in this table.



(a) (b)

Figure 4.4: Comparison on MSCOCO2-5000 and MSCOCO2.

tional cost may be twofold. The first reason lies on the simple strategy presented in Section 4.2.5 for large-scale datasets. Specifically, among those large number of unlabeled data, only the ones with high uncertainty are considered as the candidates, while the rest ones are directly dropped and are not the input of the active learning methods. The second reason may be the efficiency of the proposed optimization algorithm, which is based on the efficient ALM method. The convergence is analyzed but the convergent speed is not studied, but this subsection shows empirical efficiency on large-scale datasets.

5

Online Learning for Imbalanced Data

5.1 Introduction

To deal with the issues in online learning for imbalanced data (analyzed in Section 0.3.4), in this chapter [*] , I present a unified framework for learning with imbalanced streaming data that is easily adapted to different performance measures. The proposed framework simultaneously learns multiple classifiers with various cost vectors. In particular, at each iteration, the prediction is made by a classifier which is selected randomly according to a sampling distribution, which is updated based on the current performance measures of classifiers, similarly to the well-know

[*]The main results of this chapter were previously published in Yan Yan, Tianbao Yang, Yi Yang, Jianhui Chen. A Framework of Online Learning with Imbalanced Streaming Data. In *Thirty-First AAAI Conference on Artificial Intelligence (AAAI)* 2017.

exponential weighted average algorithm [83]. The selection of the optimal classifier is adaptive and evolving according to the streaming data. I emphasize that the proposed approach is different from the cross-validation approach, which replies on a separate validation set. Furthermore, the proposed framework enjoys a rigorous theoretical justification for the F-measure maximization. Empirical studies demonstrate that the proposed algorithm is more effective than previous online learning algorithms for imbalanced streaming data.

Now we specifically explain why the method is memory-efficient. It is designed for imbalanced data, where the evaluation metric is those imbalance measurements, e.g., F-measure, AUROC and AUPRC, rather than the error rate. However, this algorithm is also a framework for in the online or stochastic setting, where large-scale datasets can be dealt with.

Specifically, the online/stochastic setting is to train the model iteratively. In each iteration, a (subset of) training instance(s) is used to update the model. After that, the instance is dropped and never used. The difficulty to apply these imbalance measurements in such online/stochastic setting is their non-decomposability. Traditionally, one has to scan the entire dataset to compute the measurement, so there are totally $O(N \times T)$ times of scan on instances, assuming N and T are the number of instances and iterations, respectively. The framework is designed to deal with the online/stochastic setting and approximate the optimal solution to the non-decomposable imbalance measurements. In contrast to the traditional methods, in each iteration, the proposed framework scans the instance once only. The extra expense is required to maintain several auxiliary variables which do not depend on how many instances the framework scans. As a result, the framework requires $O(T)$ times of scan on instances, removing the dependence on N. What should be emphasized is that, no matter how large T is in the online/stochastic setting, the memory cost only includes the storage for the single instance scanned in the current iteration and the storage for the auxiliary variables, which is not dependent on N.

5.2 Online Multiple Cost-Sensitive Learning

I first present some notations. Let $\mathbf{x}_t \in \mathbb{R}^d$ denote the feature vector of the example received at the t-th iteration, and $y_t \in \{1, -1\}$ denote its true class label. I denote by $f_t(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ a

Table 5.1: Notations (subindex t refers to the t-th round in online learning.)

Notations	Meaning	Notations	Meaning
$\mathbf{x}_t \in \mathbb{R}^d$	feature vector	$\ell(yf(\mathbf{x}))$	loss
$y_t \in \{1, -1\}$	class label	$\hat{y}_t$	$\mathbf{I}(f_t > 0)$
$f_t(\cdot)$	prediction func.	$\bar{y}_t$	$\frac{y_t+1}{2}$
$f_t = f_t(\mathbf{x}_t)$	prediction	$\sigma(f)$	$\frac{1}{1+\exp(-f)}$
$\mathbf{I}(b)$	indicator func.	$p_t \in \mathbb{R}^K$	sampl. probs.
M_t: performance measure based on $\{\hat{y}$ $, y$ $,$ $= 1, \ldots, t-1\}$			

prediction function at the t-th iteration and by $f_t = f_t(\mathbf{x}_t)$ the prediction on the t-th example. Let $\mathbf{I}(b)$ denote an indicator function, where $\mathbf{I}(b) = 1$ if b is true and 0 otherwise. Commonly used notations in this chapter are summarized in Table 5.1.

In traditional online learning, the performance of $f_t(\cdot)$ on the example $\mathbf{x}_t$ is usually measured by a loss function $\ell(y_t f_t(\mathbf{x}_t))$, *e.g.*, hinge loss $\ell(z) = \max(0, 1 - z)$ and logistic loss $\ell(z) = \log(1 + \exp(-z))$, which are considered to be a surrogate loss of 0-1 error $\mathbf{I}(sign(f_t(\mathbf{x}_t)) \neq y_t)$. Previous studies cast the problem into learning a sequence of classifiers $f_1(\cdot), \ldots, f_T(\cdot)$ such that the regret defined below is minimized, $R_T = \sum_{t=1}^{T} \ell(y_t f_t(\mathbf{x}_t)) - \min_f \sum_{t=1}^{T} \ell(y_t f(\mathbf{x}_t))$. Many online learning algorithms have been proposed to minimize the regret such as online gradient descent [161]). However, a critique over the standard surrogate loss functions is that they ignore the cost asymmetry between the majority class and the minority one. To resolve this issue, cost-sensitive loss functions have been proposed, which give different costs to different classes: $\ell_c(f(\mathbf{x}), y) = c_+ \mathbf{I}(y = 1)\ell(f(\mathbf{x})) + c_- \mathbf{I}(y = -1)\ell(-f(\mathbf{x}))$, where $\mathbf{c} = (c_+, c_-)$ is the cost vector that controls the balance between the two loss terms. How to decide the value of c_+ and c_- remains an issue. Previous works use ad-hoc approaches to set up these parameters [129]. However, there is no guarantee that these ad-hoc approaches use appropriate values for c_+ and c_-. In addition, if c_+ and c_- are changing during the training, it is difficult to analyze the performance of the learned classifier. Another commonly used practice in batch learning is by a cross-validation approach that tunes the values of c_+ and c_- based on the offline performance on a separate validation set. Nevertheless, in online learning a separate validation set is usually not available and even if it is available there is no guarantee that the distribution of the examples in the validation set is the same as the received examples in online learning.

To address these issues, I propose an online learning framework of multiple cost-sensitive learning. The motivation is that if multiple classifiers with a number of $\mathbf{c}$ are learned simultaneously, there must exist one setting that is most appropriate to the data. Without loss of generality, I assume $c_+ + c_- = 1$ and as a result one parameter $c_+ \in (0,1)$ is needed to be set. To construct the pool of multiple values of c_+, I discretize $(0,1)$ into K evenly distributed values $\vartheta_1, \ldots, \vartheta_K$, i.e., $\vartheta_j = j/(K+1)$. With the value of $c_+ = 1 - \vartheta_j/2$, the corresponding cost sensitive loss is denoted by

$$\begin{aligned} \ell_c^j(f(\mathbf{x}), y) =& (1 - \vartheta_j/2)\mathbf{I}(y = 1)\ell(f(\mathbf{x})) \\ &+ (\vartheta_j/2)\mathbf{I}(y = -1)\ell(-f(\mathbf{x})) \end{aligned} \tag{5.1}$$

The reason that I divide ϑ_j by 2 will be clear when I present the theoretical justification. Then I learn K sequences of classifiers $f_t^1(\cdot), f_t^2(\cdot), \ldots, f_t^K(\cdot)$ simultaneously in online learning, with each sequence of $f_t^j(\cdot), t = 1, \ldots, T$ to minimize the associated regret $R_T^j = \sum_{t=1}^T \ell_c^j(f_t^j(\mathbf{x}_t), y_t) - \min_f \sum_{t=1}^T \ell_c^j(f(\mathbf{x}_t), y_t)$.

A remaining issue is how to choose a classifier from K candidates to predict $\mathbf{x}_t$ at the t-th iteration. Based on our motivation, a greedy approach is to track the "performance" of K classifiers and select the best performer on historical examples. However, it may lead to overfitting problems. I thus propose a theoretically sound randomized method that selects a classifier for prediction according to a distribution $p_t = (p_t^1, \ldots, p_t^K)^\top$ such that $\sum_j p_t^j = 1$ and $p_j^t \geq 0$. To compute the sampling probabilities, I use the following formula

$$p_t^j = \frac{\exp(\gamma \mathcal{M}_t^j)}{\sum_{j=1}^K \exp(\gamma \mathcal{M}_t^j)}, \quad j = 1, \ldots, K, \tag{5.2}$$

where $\gamma > 0$ is a learning rate hyper-parameter, and $\mathcal{M}_t^j$ is some favorite performance measure (the higher the better, *e.g.*, F-measure, AUROC, AUPRC, *etc.*) on historical examples $(\mathbf{x}_\tau, y_\tau), \tau = 1, \ldots, t-1$ using the predictions $f_1^j, \ldots, f_{t-1}^j$ of the j-th sequence of classifiers. From the Equation (5.2), classifier with higher performance will have a higher probability to be selected for making the prediction. Note that when $\gamma \to \infty$, the above approach reduces to the greedy approach. I would like to emphasize that the sampling probabilities defined above

Algorithm 7 A Framework of Online Multiple Cost-sensitive Learning

1: Input: the number of classifiers K
2: Initialize $p_1 = (1/K, \ldots, 1/K), f_1^j(\mathbf{x}) = 0, j = 1, \ldots, K$
3: for $t = 1, \ldots, T$ do
4: Receive an example $\mathbf{x}_t$
5: Sampling a classifier f_t^j by choosing j_t according to $\Pr(j) = p_t^j$
6: Compute a predicted label $\tilde{y}_t = sign(f_{j_t}^*(\mathbf{x}_t))$
7: Receive the true label y_t
8: for $j = 1, \ldots, K$ do
9: Update the classifier $f_{t+1}^j(\cdot) = \mathcal{A}(f_t^j(\cdot), \mathbf{x}_t, y_t)$
10: Update the performance $\mathcal{M}_{t+1}^j = \mathcal{M}(y_{1:t}, f_{1:t}^j)$
11: end for
12: Update the sampling probabilities p_{t+1} according to (5.2)
13: end for

are similar to that in exponentially weighted average algorithm [83] for selecting the best expert advice but with a key difference. In the learning with expert advice problem, the sampling probabilities are computed based on the cumulative loss $\sum_{\tau=1}^{t-1} \ell_\tau^j$ of different experts indexed by j, while our sampling probabilities are computed based on interesting performance measure that is suited for imbalanced data.

Now I can summarize our online multiple cost-sensitive learning in Algorithm 7. In the remainder of this section, I discuss how to update the classifier in step 9 and present a theoretical analysis of the proposed framework of online multiple cost-sensitive learning (OMCSL). In the next two sections, I discuss the step 10 that updates the performance for different measures.

For updating the classifier, I can use any online learning algorithms as long as they are designed to minimize the regret, such as online gradient descent (OGD) [161], online dual averaging [140], follow the regularized leader [67], and some specialized algorithms for online classification including online passive aggressive learning [31], perceptron [109], *etc.*. Due to the popularity and simplicity of OGD, I present the online gradient descent update. For the easy presentation, I here consider $f_t^j(\cdot)$ as a linear function, namely $f_t^j(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}_t^j$. I thus update $\mathbf{w}_t^j$ by:

$$\mathbf{w}_{t+1}^j = \mathbf{w}_t^j - \eta_t \nabla_{\mathbf{w}} \ell_c^j(\mathbf{x}_t^\top \mathbf{w}_t^j, y_t), j = 1, \ldots, K. \tag{5.3}$$

where η_t is a step size hyper-parameter, which can be set to a small value or to be decreasing

depending on the property of the loss function [161]. It is worth noting that (i) the loss function could include a regularizer on $\mathbf{w}$, *e.g.*, $\frac{}{2}\|\mathbf{w}\|_2^2$; (ii) a bias term can be incorporated by adding an extra constant feature to $\mathbf{x}$. The proposition below provides the regret guarantee for the j-th sequence of classifiers. For ease of presentation, I specialize to the linear function. One can easily generalize it to a non-linear function from a RKHS.

Proposition 2. *(Theorem 3.1 [54]) Let the linear prediction function $f_t^j(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}_t^j$ be updated based on (5.3) and $\mathbf{w}_*^j$ be the optimal prediction function that minimizes the cumulative cost-sensitive loss $\sum_{t=1}^T \ell_c^j(\mathbf{w}^\top \mathbf{x}_t, y_t)$. Assume that $\left\|\nabla_{\mathbf{w}} \ell_c^j(y\mathbf{x}^\top \mathbf{w})\right\|_2 \leq G$ and $\left\|\mathbf{w}_*^j\right\|_2 \leq D$. By setting $\eta_t = \frac{D}{G\sqrt{t}}$, then I have*

$$R_T^j = \sum_{t=1}^T \ell_c^j(f_t^j(\mathbf{x}_t), y_t) - \sum_{t=1}^T \ell_c^j(\mathbf{x}_t^\top \mathbf{w}_*^j, y_t) \leq 3GD\sqrt{T},$$

which implies that the averaged regret converges to zero at a rate of $1/\sqrt{T}$, i.e, $\frac{R_T^j}{T} \leq \frac{3GD}{\sqrt{T}}$.

Next, I analyze the updating rule of sampling probabilities in (5.2). I first present a proposition below and then provide an explanation of it.

Proposition 3. *Let $\mathbf{M}_t = (\mathcal{M}_t^1, \ldots, \mathcal{M}_t^K)^\top$ and p_t updated according to (5.2). Then there exists a $\gamma > 0$ such that $p_T^\top \mathbf{M}_T \geq \max_{1\leq j\leq K} \mathcal{M}_T^j - (V_T + \sqrt{V_T \log K})$, where $V_T = 2\sum_{t=1}^T \|\mathbf{M}_t - \mathbf{M}_{t-1}\|_\infty$ is the scaled sum of consecutive variation of the performance measure.*

From the proposition above, I can see that when the variation of the performance measure is small, the expected performance of the selected classifier (the L.H.S of the inequality) is close to the best performance measure (the R.H.S). I emphasize that the lower bound of $p_T^\top \mathbf{M}_T$ in Proposition 3 is by no means tight. It only explains to some degree why the employed sampling probabilities make sense. Bounding the online performance of an non-decomposable measure (*e.g.*, F-measure, AUROC and AUPRC) is still very challenging. In next section, I provide a theoretical analysis of the proposed framework for the F-measure optimization.

5.3 OMCSL for F-measure

In this section, I first present how to update the online F-measure, and then show that OMCSL has a solid theoretical foundation for F-measure maximization, in which the best classifier among the K classifiers will eventually yield a close-to-optimal F-measure provided that K is sufficiently large.

Given a sequence of labels $y_1, \ldots, y_t$ and a sequence of predictions $f_1, \ldots, f_t$, I can calculate the F-measure by $F_{t+1} = \frac{2\sum_{i=1}^{t} \bar{y}_i \hat{y}_i}{\sum_{i=1}^{t} \bar{y}_i + \sum_{i=1}^{t} \hat{y}_i}$ where $\bar{y}_t = (y_t + 1)/2 \in \{1, 0\}$ and $\hat{y}_t = \mathbf{I}(f_t > 0)$. However, directly calculating the online F-measure by going through all examples is expensive, which requires to store all predictions $f_t(\mathbf{x}_t)$ and y_t. Indeed, the online F-measure can be calculated incrementally. To this end, I let $a_t = \sum_{i=1}^{t} \bar{y}_i \hat{y}_i$ and $c_t = \sum_{i=1}^{t} \bar{y}_i + \sum_{i=1}^{t} \hat{y}_i$. Then I can calculate $F_{t+1} = \frac{2a_t}{c_t}$ and update a_t and c_t incrementally by

$$a_{t+1} = \begin{cases} a_t + 1, \text{ if } y_{t+1} = 1 \text{ and } f_{t+1} > 0, \\ a_t, \text{ otherwise;} \end{cases}$$

$$c_{t+1} = \begin{cases} c_t + 2, \text{ if } y_{t+1} = 1 \text{ and } f_{t+1} > 0, \\ c_t + 1, \text{ if } y_{t+1} = 1 \text{ or } f_{t+1} > 0, \\ c_t, \text{ if } y_{t+1} = -1 \text{ and } f_{t+1} \leq 0. \end{cases} \tag{5.4}$$

5.3.1 A Theoretical Justification

I show that when K is sufficiently large, there exists a sequence of classifiers among the K sequences that will eventually converge to a classifier that has a close-to-optimal F-measure. To this end, I assume the data is i.i.d. The analysis is built on several previous works on the F-measure maximization [97] and the theory of consistency for cost-sensitive surrogate loss minimization [105]. To present the results, I first give some notations. Let $h(\mathbf{x}) \in \mathcal{H} : \mathbb{R}^d \to \{1, -1\}$ denote a classifier and $\mathbf{e}(h) = (e_1(h), e_2(h))^\top$ denote the false negative (FN) error and false positive (FP) error of $h(\mathbf{x})$, respectively, i.e., $e_1(h) = \Pr(y = 1, h(\mathbf{x}) = -1)$, $e_2(h) =$

$\Pr(y = -1, h(\mathbf{x}) = 1)$ where $\Pr(\cdot)$ denotes the probability over $(\mathbf{x}, y)$. When it is clear from the context, I write $\mathbf{e} = \mathbf{e}(h)$ for short. Let P_1 denote the marginal probability of the positive class, i.e., $P_1 = \Pr(y = 1)$. Then the F-measure of $h(\cdot)$ on the population level can be computed by [97] $F(h) \triangleq F(\mathbf{e}) = \frac{2(P_1 - e_1)}{2P_1 - e_1 + e_2}$. Let $\mathbf{c}(\tau) = (1 - \frac{\tau}{2}, \frac{\tau}{2})^\top$. The following proposition exhibits that maximizing F-measure is equivalent to minimizing a cost-sensitive error.

Proposition 4. *(Proposition 4 in [97]) Let* $F_* = \max_{\mathbf{e}} F(\mathbf{e})$. *Then I have* $\mathbf{e}_* = \arg\min_{\mathbf{e}} \mathbf{c}(F_*)^\top \mathbf{e} \Leftrightarrow F(\mathbf{e}_*) = F_*$.

The above proposition indicates that one can optimize the following cost-sensitive error

$$\mathbf{c}(F_*)^\top \mathbf{e} = \left(1 - \frac{F_*}{2}\right) e_1 + \frac{F_*}{2} e_2, \tag{5.5}$$

to obtain an optimal classifier $h^*(\mathbf{x})$, which will give the optimal F-measure, i.e., $F(h^*) = F_*$. However, the cost-sensitive error in (5.5) requires knowing the exact value of the optimal F-measure. To address this issue, I discretize $(0, 1)$ to have a set of evenly distributed values $\{\vartheta_1, \ldots, \vartheta_K\}$ such that $\vartheta_{j+1} - \vartheta_j = \varepsilon_0/2$, which serve as the candidate values of F_*. Then I can solve for a series of K classifiers to minimize the cost-senstive error

$$h_j^* = \arg\min_{h \in \mathcal{H}} \left(1 - \frac{\vartheta_j}{2}\right) e_1 + \frac{\vartheta_j}{2} e_2 = \mathbf{c}(\vartheta_j)^\top \mathbf{e}, j = 1, \ldots, K. \tag{5.6}$$

This explains our choice of $\vartheta_j/2$ in (5.1). The following proposition shows that there exists one classifier among $\{h_j^*, \cdots, h_K^*\}$ that can achieve a close-to-optimal F-measure as long as ε_0 is small enough.

Proposition 5. *Let* $\{\vartheta_1, \ldots, \vartheta_K\}$ *be a set of values evenly distributed in* $(0, 1)$ *such that* $\vartheta_{j+1} - \vartheta_j = \varepsilon_0/2$. *Then there exists* $h_j^* \in \{h_j^*, \cdots, h_K^*\}$ *such that* $F(h_j^*) \geq F_* - \frac{2\varepsilon_0 B}{P_1}$, *where* $B = \max_{\mathbf{e}} \|\mathbf{e}\|_2$.

Remark: The above proposition also implies an interesting result that the smaller P_1 (i.e., more imbalanced of the data), the larger gap between $F(h_j^*)$ and F_* (i.e., more difficult to optimize the F-measure).

Proposition 5 only provides the guarantee on the optimal classifiers $\{h_j^*, \cdots, h_K^*\}$. In practice, one cannot obtain these optimal classifiers because the distribution of the data is unknown. The following proposition shows that as long as the obtained classifiers achieve a cost-sensitive error close to the optimal classifiers, a similar guarantee to that in Proposition 5 holds.

Proposition 6. *Let* $\{\vartheta_1, \ldots, \vartheta_K\}$ *be a set of values evenly distributed in* $(0, 1)$ *such that* $\vartheta_{j+1} - \vartheta_j = \varepsilon_0/2$. *Let* $\{\hat{h}_1, \ldots, \hat{h}_K\}$ *be a set of classifiers that minimize the cost-sensitive errors in (5.6) to a certain degree such that* $\mathbf{c}(\vartheta_j)^\top \mathbf{e}(\hat{h}_j) \leq \mathbf{c}(\vartheta_j)^\top \mathbf{e}(h_j^*) + \varepsilon_1$. *Then there exists* $\hat{h}_j$ *such that* $F(\hat{h}_j^*) \geq F_* - \frac{(2\varepsilon_0 B + \varepsilon_1)}{P_1}$, *where* $B = \max_{\mathbf{e}} \|\mathbf{e}\|_2$.

Remark: The result in Proposition 5 is a special case of Proposition 6 when $\varepsilon_1 = 0$. Proposition 6 is a corollary of Proposition 5 in [97].

Finally, I are ready to present the theoretical guarantee on the presented OMCSL algorithm for the F-measure maximization.

Theorem 1. *Let* $\{\vartheta_1, \ldots, \vartheta_K\}$ *be a set of values evenly distributed in* $(0, 1)$ *such that* $\vartheta_{j+1} - \vartheta_j = \varepsilon_0/2$, $\mathbf{w}_t^j, t = 1, \ldots, T$ *be a sequence updated according to (5.3) based on the j-th cost-sensitive loss in (5.1) such that* $\|\mathbf{w}_t^j\|_2 \leq D$, *and* $\widehat{\mathbf{w}}_T^j = \sum_{t=1}^T \mathbf{w}_t^j/T$. *Assume* $(\mathbf{x}_t, y_t), t = 1, \ldots, T$ *are i.i.d. samples such that* $\|\mathbf{x}_t\|_2 \leq R$ *and the loss function* $\ell(z) = \max(0, 1 - z)$ *is the hinge loss. There exists a* $j \in \{1, \ldots, K\}$ *with a probability* $1 - \delta$ *such that*

$$F(\hat{h}_T^j) \geq F_* - \frac{2\varepsilon_0 B + 3RD(1 + \ln(2/\delta))/\sqrt{T}}{P_1}$$

where $\hat{h}_T^j(\mathbf{x}) = sign(\mathbf{x}^\top \widehat{\mathbf{w}}_T^j)$.

Remark: The theorem implies that when $T \to \infty$, there exists a classifier $\hat{h}_T^j = sign(\mathbf{x}^\top \widehat{\mathbf{w}}_T^j)$ achieves a close-to-optimal F-measure as long as ε_0 is small enough. The proof is presented in the supplement.

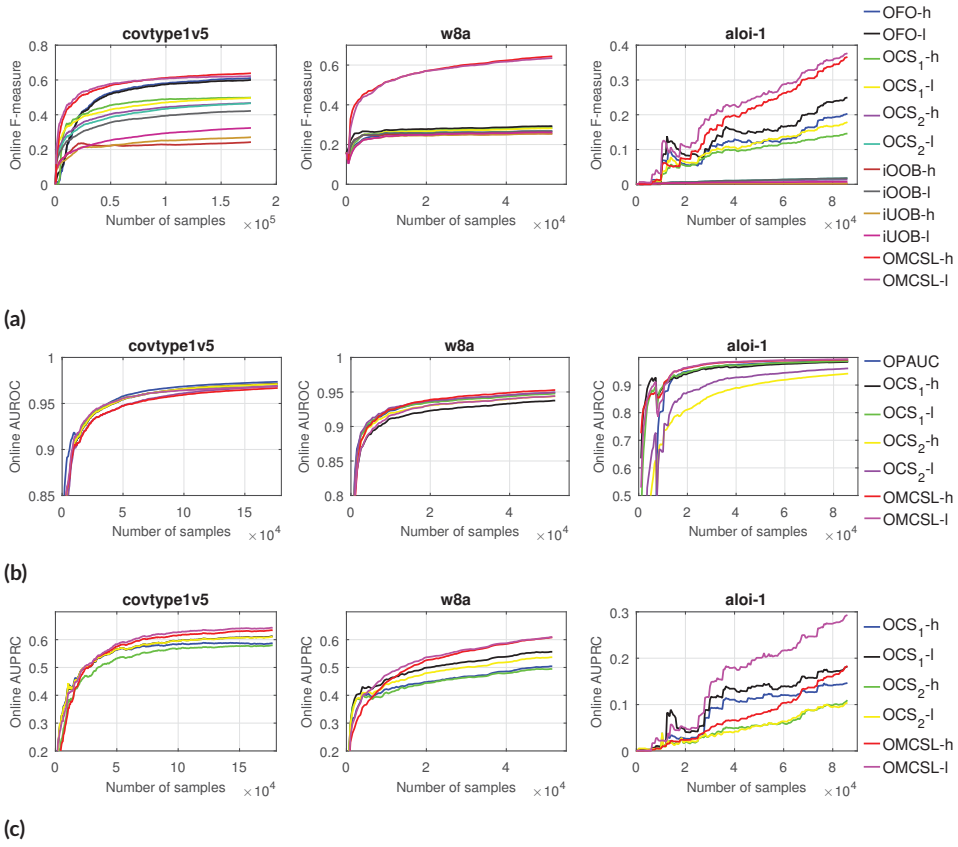


Figure 5.1: Online performance.

5.4 OMCSL for AUROC and AUPRC

In this section, I briefly present how to update AUROC and AUPRC in an online fashion. The challenge of updating AUROC and AUPRC in the online setting lies at that I need to compare the present example to historically received examples in terms of predictions. A naive way to achieve this is to store the labels and predictions of all classifiers for historically received examples. However, this would increase the memory requirements, which is usually not allowed in online learning. To avoid storing the labels and predictions of all examples, I introduce two hash tables L_t^+ and L_-^t with a fixed length of m that partitions $(0, 1)$ into m ranges $(0, 1/m), (1/m, 2/m), \ldots, ((m-1)/m, 1)$. For $i \in \{1, \ldots, m\}$, $L_t^+[i]$ stores the number of positive examples before the t-th iteration (including the t-th iteration) whose predictions f are such that $\sigma(f) \in [(i-1)/m, i/m)$ [†], and $L_t^-[i]$ stores the number of negative examples before the t-th iteration (including t-th iteration) whose predictions f are such that $\sigma(f) \in [(i-1)/m, i/m)$.

Given L_t^+ and L_t^-, I can show that AUROC_{t+1} can be updated approximately using the two hash tables. In particular, if $y_{t+1} = 1$, I have

$$\text{AUROC}_{t+1} = \frac{N_t^+}{N_t^+ + 1}\text{AUROC}_t + \frac{1}{(N_t^+ + 1)N_t^-}\left(\sum_{j=1}^{i} L_t^-[j] + L_t^-[i+1]/2\right),$$

where i is the largest index such that $i/m \leq \sigma(f_{t+1})$, and if $y_{t+1} = -1$, I update it by

$$\text{AUROC}_{t+1} = \frac{N_t^-}{N_t^- + 1}\text{AUROC}_t + \frac{1}{N_t^+(N_t^- + 1)}\left(\sum_{j=i+1}^{m-1} L_t^+[j] + L_t^+[i]/2\right),$$

where i is the smallest index such that $i/m \geq \sigma(f_{t+1})$.

Similarly, by using L_t^+ and L_t^-, I derive the online update of AUPRC_t as below: $\text{AUPRC}_{t+1} = \frac{1}{2}\sum_{i=0}^{m-1}(\text{R}(i) - \text{R}(i+1))(\text{P}(i) + \text{P}(i+1))$. where $\text{R}(i) = \frac{\sum_{j=i+1}^{m} L_t^+[j]}{N_t^+}$, and $\text{P}(i) = \frac{\sum_{j=i+1}^{m} L_t^+[j]}{\sum_{j=i+1}^{m} L_t^+[j] + \sum_{j=i+1}^{m} L_t^-[j]}$. The overall time complexity of computing AUROC_{t+1} and AUPRC_{t+1} is $O(m)$. Detailed development of the online update of AUROC and AUPRC can be found in

[†] $\sigma(f)$ is the sigmoid function defined in Table 5.1.

Table 5.2: Data statistics.

Datasets	#Examples	#Features	#Pos:#Neg
covtype1v5	211,840	54	1:22.3
w8a	64,700	300	1:32.5
aloi-1	108,000	128	1:999

Table 5.3: Average prediction performance on the testing set over 25 trials.

Methods	covtype1v5			w8a			aloi-1		
	Fmeasure	AUROC	AUPRC	Fmeasure	AUROC	AUPRC	Fmeasure	AUROC	AUPRC
OPAUC	–	0.9813	–	–	0.9602	–	–	0.9993	–
OFO-h	0.7071	–	–	0.6616	–	–	0.2596	–	–
OCS_1-h	0.5204	0.5000	0.4999	0.4948	0.4761	0.4726	0.3148	0.4285	0.3148
OCS_2-h	0.5035	0.5035	0.4820	0.4478	0.4478	0.4478	0.1062	0.1204	0.0311
iOOB-h	0.1180	–	–	0.0837	–	–	0.0021	–	–
iUOB-h	0.1174	–	–	0.0839	–	–	0.0021	–	–
OMCSL-h	0.6449	0.9809	0.7042	0.7147	0.9598	0.7087	0.4560	0.9996	0.7732
OFO-l	0.6600	–	–	0.6325	–	–	0.1407	–	–
OCS_1-l	0.5230	0.5627	0.5230	0.5156	0.5156	0.6381	0.4473	0.4966	0.6176
OCS_2-l	0.5044	0.5044	0.5044	0.4405	0.4511	0.6241	0.1429	0.0237	0.4760
iOOB-l	0.1356	–	–	0.0907	–	–	0.0038	–	–
iUOB-l	0.1256	–	–	0.0903	–	–	0.0026	–	–
OMCSL-l	0.6597	0.9823	0.7187	0.6891	0.9551	0.7086	0.5197	0.9998	0.8208

† Suffixes "-h" and "-l" stand for the algorithms with hinge loss and logistic loss respectively. The top results are in bold.

* For OFO and OPAUC which directly optimize a specific measure, I omit their results in other measures indicated by "–". iOOB and iUOB are resampling based ensemble algorithms which predict a new instance by voting, rather than decision values. Therefore, AUROC and AUPRC are unavailable, indicated by "–".

Appendix C.

5.5 Experiments

In this section, I evaluate OMCSL for optimizing three measures, F-measure, AUROC and AUPRC, and compare with competing online learning algorithms on three public imbalanced datasets. Table 5.2 lists the statistics of used three datasets. To construct imbalanced data from multiclass datasets `covtype`, I sample instances of the fifth class as positive and instances of the first class as negative, denoted by `covtype1v5`. Similarly, for `aloi`, I sample instances of the first class as positive, and the rest as negative, denoted by `aloi-1`. For each dataset, I randomly sample 4/5 instances as the training set and the rest 1/5 as the testing set. I repeat the experiment on 25 various random splits and report the average results.

I compare the proposed OMCSL method with several state of the art online learning algorithms, namely OCS_1, OCS_2 [129], OFO [19], OPAUC [47], and iOOB, iUOB [131]. Among

Table 5.4: Average absolute error between $\hat{c}_+$ predicted by OMCSL and c_+^*.

Methods	covtype1v5			w8a			aloi-1		
	Fmeasure	AUROC	AUPRC	Fmeasure	AUROC	AUPRC	Fmeasure	AUROC	AUPRC
OMCSL-h	0.010	0.004	0.052	0.059	0.052	0.137	0.084	0.016	0.154
OMCSL-l	0.005	0	0	0.078	0.023	0.127	0.144	0.003	0.165

them, OFO and OPAUC directly optimize the target measures (i.e., F-measure and AUROC, respectively), OCS_1 and OCS_2 are both cost-sensitive online methods, iOOB and iUOB are resampling based ensemble methods (oversampling and undersampling). Since the latter two algorithms apply voting to predict a new instance, rather than decision values, I only compute F-measure for them. To examine the performance of using different loss functions, I investigate both the hinge loss and the logistic loss in the experiment and denote these two loss functions by suffixing "-h" and "-l" to the corresponding methods respectively. Note that OPAUC is designed only for square loss, thus I only report one result for OPAUC. The details of hyperparameters of these methods can be found in Appendix D.

5.5.1 Results

I evaluate and compare both online performance on training data and testing performance on testing data. Note that the testing performance is to evaluate the returned models on the testing data in batch, which demonstrates the generalization ability of different online learning algorithms. Table 5.3 lists the prediction performance on testing data of various algorithms. Fig. 5.1a, Fig. 5.1b and Fig. 5.1c demonstrate the averaged online performance (i.e., F-measure, AUROC and AUPRC) of various algorithms on three datasets over 25 trials. As can be observed from both online performance and testing performance, OMCSL achieves better performance than cost-sensitive online algorithms and resampling based online algorithms. The three figures also exhibit a clear trend that when the ratio of positive examples to the negative examples increases, the advantage of OMCSL becomes more striking. Compared to the methods that directly optimize target measure, i.e., OFO and OPAUC, OMCSL achieves competitive if not better performance.

An important reason that OMCSL achieves satisfactory performance is the capability to se-

lect a close-to-optimal cost vector $[c_+, c_-]$, which is also exhibited by our theoretical analysis for F-measure optimization. To investigate this property, I perform online cost-sensitive learning with the same costs used in OMCSL, i.e., $c_+ = \{0.55, 0.60, 0.65, ..., 0.95\}$, respectively to find the best cost according to the overall online performance, which I denote by c_+^*. To compare the selected best cost (corresponding to the largest selection probability) by the proposed OMCSL, I average c_+ selected by OMCSL in the last 5,000 iterations as an estimate of the best cost, which I denote by $\hat{c}_+$. Then I compute the absolute error between c_+^* and $\hat{c}_+$, i.e., $err = |c_+^* - \hat{c}_+|$, and report the average error over 25 trials in Table 5.4. Our observation is that $\hat{c}_+$ predicted by OMCSL is often close to c_+^* given that the step length for search of c_+ is set to 0.05. This property is particularly crucial in the online scenario due to the requirement of going through the training data only once. The proposed OMCSL provides an accurate estimation of the optimal cost.

5.6 Conclusion

This work presents a unified online learning framework for imbalanced data. The proposed algorithm simultaneously trains multiple classifiers with various costs, and predicts by randomly selecting a classifier based on a distribution determined by online performance of individual learners. A rigorous theoretical justification for the F-measure maximization is provided. Empirical studies show the superior performance of OMCSL and its capability to select satisfactory costs.

5.7 Appendix A: Proof of Proposition 3

Let $\mathbf{r}_t = \mathbf{M}_t - \mathbf{M}_{t-1}$ and $\mathbf{M}_0 = 0$. Then $\mathbf{M}_t = \sum_{\tau=1}^{t} \mathbf{r}_\tau$. Then by induction, I can show that

$$p_t^j = \frac{p_{t-1}^j \exp(\gamma r_t^j)}{\sum_{j=1}^{K} p_{t-1}^j \exp(\gamma r_t^j)}$$

As a result, the update of p_t can be considered as the mirror descent update for the linear loss $\ell_t(p) = -p^\top \mathbf{r}_t$ using the negative entropy function $\omega(p) = \sum_i p_i \ln p_i$ as the potential function. In particular, let $V(p, \mathbf{q}) = \omega(p) - \omega(\mathbf{q}) - \nabla\omega(\mathbf{q})^\top (p - \mathbf{q}) = \sum_{i=1}^{d} p_i \ln \frac{p_i}{q_i}$ denote the

Bregman distance induced by $\omega(p)$, and $\Delta = \{p \in \mathbb{R}^K; p \geq 0, \sum_{i=1}^K p_i = 1\}$. Then the update of p_t is equivalent to

$$\begin{aligned} \nabla\omega(\mathbf{q}_t) &= \nabla\omega(p_{t-1}) + \gamma\mathbf{r}_t \\ p_t &= \min_{p\in\Delta} V(p, \mathbf{q}_t) \end{aligned} \tag{5.7}$$

The following lemma establishes the regret bound the above update.

Lemma 1. *[54] Let p_t be updated according to (5.7). Then for any $p \in \Delta$ I have*

$$-\sum_{t=1}^T p_t^\top \mathbf{r}_t + \sum_{t=1}^T p^\top \mathbf{r}_t \leq \frac{V(p, p_1)}{\gamma} + \frac{\gamma}{2}\sum_{t=1}^T \|\mathbf{r}_t\|_\infty^2$$

Since $p_1 = (1/K, \ldots, 1/K)^\top$, then

$$V(p, p_1) = \sum_{i=1}^K p_i \ln(Kp_i) \leq \ln K$$

Thus,

$$-\sum_{t=1}^T p_t^\top \mathbf{r}_t + \sum_{t=1}^T p^\top \mathbf{r}_t \leq \frac{\ln K}{\gamma} + \frac{\gamma}{2}\sum_{t=1}^T \|\mathbf{M}_t - \mathbf{M}_{t-1}\|_\infty^2$$

Then

$$\begin{aligned} &-\sum_{t=1}^T p_T^\top \mathbf{r}_t + \sum_{t=1}^T (p_T - p_t)^\top \mathbf{r}_t + \sum_{t=1}^T p^\top \mathbf{r}_t \\ &\leq \frac{\ln K}{\gamma} + \frac{\gamma}{2}\sum_{t=1}^T \|\mathbf{M}_t - \mathbf{M}_{t-1}\|_\infty^2 \end{aligned}$$

By noting that $\sum_{t=1}^T \mathbf{r}_t = \mathbf{M}_T$ and $\mathbf{r}_t = \mathbf{M}_t - \mathbf{M}_{t-1}$, I have,

$$\begin{aligned} p_T^\top \mathbf{M}_T \geq p^\top \mathbf{M}_T + \sum_{t=1}^T (p_T - p_t)^\top (\mathbf{M}_t - \mathbf{M}_{t-1}) + \\ - \left(\frac{\ln K}{\gamma} + \frac{\gamma}{2}\sum_{t=1}^T \|\mathbf{M}_t - \mathbf{M}_{t-1}\|_\infty^2 \right) \end{aligned}$$

By the definition of $V_T = 2\sum_{t=1}^{T} \|\mathbf{M}_t - \mathbf{M}_{t-1}\|_\infty^2$. Let $\gamma = 2\sqrt{\ln K / V_T}$, then

$$\begin{aligned}
&\frac{\ln K}{\gamma} + \frac{\gamma}{2}\sum_{t=1}^{T} \|\mathbf{M}_t - \mathbf{M}_{t-1}\|_\infty^2 \leq \sqrt{V_T \ln K} \\
&\sum_{t=1}^{T} (p_t - p_T)^\top (\mathbf{M}_t - \mathbf{M}_{t-1}) \\
&\leq \sum_{t=1}^{T} \|(p_t - p_T)\|_1 \|\mathbf{M}_t - \mathbf{M}_{t-1}\|_\infty \\
&\leq \sum_{t=1}^{T} 2\|\mathbf{M}_t - \mathbf{M}_{t-1}\|_\infty = V_T
\end{aligned}$$

As a result

$$p_T^\top \mathbf{M}_T \geq p^\top \mathbf{M}_T - V_T - \sqrt{V_T \ln K}$$

Since the above inequality holds for any $p \in \Delta$, then

$$\begin{aligned}
p_T^\top \mathbf{M}_T &\geq \max_{p \in \Delta} p^\top \mathbf{M}_T - (V_T + \sqrt{V_T \ln K}) \\
&= \max_{1 \leq j \leq K} M_T^j - (V_T + \sqrt{V_T \ln K})
\end{aligned}$$

5.8 Appendix B: Proof of Theorem 1

From Proposition 1, I have

$$\frac{1}{T}\sum_{t=1}^{T} \ell_c^j(\mathbf{x}_t^\top \mathbf{w}_t^j, y_t) - \frac{1}{T}\sum_{t=1}^{T} \ell_c^j(\mathbf{x}_t^\top \mathbf{w}_*^j, y_t) \leq \frac{3DR}{\sqrt{T}}$$

Lemma 2. *Let $\mathbf{w}_t^j, t = 1, \ldots, T$ be a sequence updated according to (5.3) based on the j-th cost-sensitive loss in (5.1) such that $\|\mathbf{w}_t^j\|_2 \leq D$, and $\widehat{\mathbf{w}}_T^j = \sum_{t=1}^{T} \mathbf{w}_t^j / T$. Assume $(\mathbf{x}_t, y_t), t = 1, \ldots, T$ are i.i.d. samples such that $\|\mathbf{x}\|_2 \leq R$ and the loss function $\ell(z) = \max(0, 1 - z)$ is the*

hinge loss. With a high probability $1-\delta$*, I have*

$$\mathbf{E}_{\mathbf{x},y}[\ell_c^j(\mathbf{x}^\top \widehat{\mathbf{w}}_T^j, y)] \leq \frac{1}{T}\sum_{t=1}^{T} \ell_c^j(\mathbf{x}_t^\top \mathbf{w}_t^j, y_t) + DR\sqrt{\frac{2}{T}\ln\left(\frac{1}{\delta}\right)}$$

The above lemma is a result of the Corollary 2 in [22] by noting that $\ell_c^j(\mathbf{x}_t^\top \mathbf{w}_t^j, y_t) \in [0, DR]$ due to the non-negativity and the Lipschitz continuity of $\ell_c^j(\mathbf{w}^\top \mathbf{x}, y)$ and $|\mathbf{x}_t^\top \mathbf{w}_t^j| \leq DR$.

Lemma 3. *Assume* $(\mathbf{x}_t, y_t), t = 1, \ldots, T$ *are i.i.d. samples such that* $\|\mathbf{x}_t\|_2 \leq R$ *and the loss function* $\ell(z) = \max(0, 1-z)$ *is the hinge loss. For any* $\mathbf{w}_*$ *such that* $\|\mathbf{w}_*\|_2 \leq D$*. With a high probability* $1-\delta$*, I have*

$$\frac{1}{T}\sum_{t=1}^{T} \ell_c^j(\mathbf{x}_t^\top \mathbf{w}_*^j, y_t) \leq \mathbf{E}_{(\mathbf{x},y)}[\ell_c^j(\mathbf{x}^\top \mathbf{w}_*^j, y)] + DR\sqrt{\frac{1}{2T}\ln\left(\frac{1}{\delta}\right)}$$

The above lemma is a result of Hoeffding bound [16] by noting that $\ell(\mathbf{w}_*^\top \mathbf{x}_t, y_t) \in [0, DR]$. Combining the above two lemmas and Proposition 1, I have with a probability $1-2\delta$

$$\mathbf{E}_{\mathbf{x},y}[\ell_c^j(\mathbf{x}^\top \mathbf{w}_t^j, y)] - \mathbf{E}_{(\mathbf{x},y)}[\ell(\mathbf{x}^\top \mathbf{w}_*^j, y)] \leq \frac{3DR}{\sqrt{T}} + \frac{3DR\sqrt{\ln(1/\delta)}}{\sqrt{T}}$$

This proves the first inequality in Theorem 1. To prove the second inequality, I leverage the calibrated result in [105]. In particular, since $\ell_c^j(z, y) = (1-\vartheta_j/2)\ell(z) + \vartheta_j/2\ell(-z)$ is $\vartheta_j/2$ classification calibrated, therefore the excess risk for the surrogate loss indicates the excess risk for the cost-sensitive error. Let $\mathbf{w}_*^j$ be the solution to $\min_{\|\mathbf{w}\|_2 \leq D} \mathbf{E}_{(\mathbf{x},y)}[\ell_c^j(\mathbf{x}^\top \mathbf{w}, y)]$. By using the consistency result for the weighted hinge loss, I have

$$\begin{aligned}
&\mathbf{c}(\vartheta_j)^\top \mathbf{e}(\hat{p}_T^j) - \mathbf{c}(\vartheta_j)^\top \mathbf{e}(p_*^j) \\
&\qquad \leq \mathbf{E}_{\mathbf{x},y}[\ell_c^j(\mathbf{x}^\top \mathbf{w}_t^j, y)] - \mathbf{E}_{(\mathbf{x},y)}[\ell_c^j(\mathbf{x}^\top \mathbf{w}_*^j, y)]
\end{aligned}$$

which then implies the second inequality. The the third inequality is proved by leveraging the result in Proposition 5.

5.9 Appendix C: detailed development of online AUROC and AUPRC

AUROC is defined as the area under the receiver operating characteristic (ROC) curve (i.e., true positive rate versus false positive rate), and AUPRC is defined as the area under the precision-recall curve. A traditional approach to calculate both measures is based on empirical curve using trapezoidal estimation. However, this approach needs to go through all examples at every iteration and needs to store the labels and predictions of all examples, which is expensive for big data. Below, I develop online updates for approximating the two measures. The key to our development is to use an efficient data structure.

For AUROC, I use the following analytical definition [156]:

$$\begin{aligned}\text{AUROC}_t =& \text{AUROC}(y_{1:t}, f_{1:t}) \\ =& \frac{\sum_{i=1}^{N_t^+}\sum_{i=1}^{N_t^-}\mathbf{I}(f(\mathbf{x}_i^+) > f(\mathbf{x}_j^-))}{N_t^+ N_t^-},\end{aligned} \tag{5.8}$$

where $\{\mathbf{x}_1^+, \ldots, \mathbf{x}_{N_t^+}^+\}$ are positive examples and $\{\mathbf{x}_1^-, \ldots, \mathbf{x}_{N_t^-}^+\}$ are negative examples. Now, I present an online update for computing AUROC_{t+1} based on AUROC_t with a new example $\mathbf{x}_{t+1}$ whose true label is y_{t+1} and prediction is given by $f(\mathbf{x}_{t+1})$. I consider two scenarios, $y_{t+1} = 1$ and $y_{t+1} = -1$.

If $y_{t+1} = 1$, $N_{t+1}^+ = N_t^+ + 1$ and $N_{t+1}^- = N_t^-$. Then

$$\begin{aligned}&\text{AUROC}_{t+1} \\ =& \frac{\sum_{i=1}^{N_t^+}\sum_{j=1}^{N_t^-}\mathbf{I}(f(\mathbf{x}_i^+) > f(\mathbf{x}_j^-)) + \sum_{j=1}^{N_t^-}\mathbf{I}(f(\mathbf{x}_{t+1}) > f(\mathbf{x}_j^-))}{(N_t^+ + 1)N_t^-} \\ =& \frac{N_t^+}{N_t^+ + 1}\text{AUROC}_t + \frac{1}{(N_t^+ + 1)N_t^-}\sum_{j=1}^{N_t^-}\mathbf{I}(f(\mathbf{x}_{t+1}) > f(\mathbf{x}_j^-))\end{aligned} \tag{5.9}$$

If $y_{t+1} = -1$, $N_{t+1}^+ = N_t^+$ and $N_{t+1}^- = N_t^- + 1$. Then

$$\begin{aligned}
&\text{AUROC}_{t+1} \\
=&\frac{\sum_{i=1}^{N_t^+}\sum_{i=1}^{N_t^-}\mathbf{I}(f(\mathbf{x}_i^+) > f(\mathbf{x}_j^-)) + \sum_{i=1}^{N_t^+}\mathbf{I}(f(\mathbf{x}_i^+ > f(\mathbf{x}_{t+1}))}{N_t^+(N_t^- + 1)} \\
=&\frac{N_t^-}{N_t^- + 1}\text{AUROC}_t + \frac{1}{N_t^+(N_t^- + 1)}\sum_{i=1}^{N_t^+}\mathbf{I}(f(\mathbf{x}_i^+) > f(\mathbf{x}_{t+1})).
\end{aligned} \tag{5.10}$$

From above I can see that given AUROC_t, the additional computation of AUROC_{t+1} is to count how many negative examples before $t+1$ have smaller prediction than $\mathbf{x}_{t+1}$ if it has a positive label or count how many positive examples before $t+1$ have larger predictions than $\mathbf{x}_{t+1}$ if it has a negative label. To avoid storing the labels and predictions of all examples, I introduce two hash tables L_t^+ and L_-^t with a length of m that partitions $(0,1)$ into m ranges $(0,1/m), (1/m, 2/m), \ldots, ((m-1)/m, 1)$. For $i \in \{1, \ldots, m\}$, $L_t^+[i]$ stores the number of positive examples before the t-th iteration (including t-th iteration) whose predictions f are such that $\sigma(f) \in [(i-1)/m, i/m)$ [‡], and $L_t^-[i]$ stores the number of negative examples before the t-th iteration (including t-th iteration) whose predictions f are such that $\sigma(f) \in [(i-1)/m, i/m)$.

Given a new example $\mathbf{x}_{t+1}$, if $y_{t+1} = 1$ I first find the largest i such that $i/m \leq \sigma(f_{t+1})$ then estimate the number of negative examples whose predictions are less than f_{t+1} by $\sum_{j=1}^{i} L_t^-[j] + L_t^-[i+1]/2$; if $y_{t+1} = -1$ I first find the smallest i such that $i/m \geq \sigma(f_{t+1})$ then estimate the number of positive examples that are larger than f_{t+1} by $\sum_{j=i+1}^{m-1} L_t^+[j] + L_t^+[i]/2$. Here, the half terms $L_t^-[i+1]/2$ and $L_t^+[i]/2$ are added assuming that the predictions in $(i/m, (i+1)/m]$ are uniformly distributed. To summarize, if $y_{t+1} = 1$ the AUROC_{t+1} is updated by

$$\begin{aligned}
\text{AUROC}_{t+1} =&\frac{N_t^+}{N_t^+ + 1}\text{AUROC}_t \\
&+ \frac{1}{(N_t^+ + 1)N_t^-}\left(\sum_{j=1}^{i} L_t^-[j] + L_t^-[i+1]/2\right),
\end{aligned} \tag{5.11}$$

[‡] $\sigma(f)$ is the sigmoid function defined in Table 5.1.

where i is the largest index such that $i/m \leq \sigma(f_{t+1})$ and if $y_{t+1} - 1$ it will be updated by

$$\begin{aligned}\mathrm{AUROC}_{t+1} =& \frac{N_t^-}{N_t^- + 1}\mathrm{AUROC}_t \\ &+ \frac{1}{N_t^+(N_t^- + 1)}\left(\sum_{j=i+1}^{m-1} L_t^+[j] + L_t^+[i]/2\right),\end{aligned} \tag{5.12}$$

where i is the smallest index such that $i/m \geq \sigma(f_{t+1})$. After this, I need to update L_t^+ or L_t^- accordingly. The overall time complexity for updating AUROC_{t+1} is $O(m)$ independent of the size of data.

Next, I consider how to update AUPRC incrementally. Our method is based on a number m of recall and precision values of predictions. In particular, at the t-th iteration given $y_1, \ldots, y_t$ and predictions $f_1, \ldots, f_t$, I let $\mathrm{R}(i)$ and $\mathrm{P}(i)$ denote the recall and precision, respectively, when the threshold of sigmoid prediction is given by i/m, i.e.,

$$\begin{aligned}\mathrm{R}(i) &= \frac{\sum_{\quad=1}^{t} \mathbf{I}(\sigma(f_{\quad}) \geq i/m \wedge y_{\quad} = 1)}{N_t^+}, \\ \mathrm{P}(i) &= \frac{\sum_{\quad=1}^{t} \mathbf{I}(\sigma(f_{\quad}) \geq i/m \wedge y_{\quad} = 1)}{\sum_{\quad=1}^{t} \mathbf{I}(\sigma(f_{\quad}) \geq i/m)}.\end{aligned} \tag{5.13}$$

Then AUPRC_{t+1} can be estimated by

$$\mathrm{AUPRC}_{t+1} = \sum_{i=0}^{m-1}(\mathrm{R}(i) - \mathrm{R}(i+1))(\mathrm{P}(i) + \mathrm{P}(i+1))/2. \tag{5.14}$$

Using the two hash tables L_t^+ and L_t^-, I can compute $\mathrm{R}(i)$ and $\mathrm{P}(i)$ by

$$\mathrm{R}(i) = \frac{\sum_{j=i+1}^{m} L_t^+[j]}{N_t^+}, \quad \mathrm{P}(i) = \frac{\sum_{j=i+1}^{m} L_t^+[j]}{\sum_{j=i+1}^{m} L_t^+[j] + \sum_{j=i+1}^{m} L_t^-[j]}. \tag{5.15}$$

6

Conclusion

In this dissertation, I investigate the solutions of machine learning algorithms to handle the large-scale data. I mainly focus on two aspects of the scalability of machine learning algorithms, i.e., the computational cost and the memory cost, and the thesis investigates three potential approaches to deal with the large-scale problem, i.e., improving the computational efficiency of the algorithms (Section 2), reducing the scale of the used training data (Section 3 and 4) and reducing the memory cost (Section 5). By these sections, such three angles provide some insights and techniques that can be used in the future large-scale problems.

Particularly, I analyze four realistic machine learning tasks, i.e., matrix completion by maximum margin matrix factorization, semi-supervised learning by label aggregation, active learning

for image classification by privileged information and online learning for imbalanced data. I investigate the computational cost in the first three tasks, and focus on the memory cost in the last task. The contributions of this dissertation are summarized in the subsequent paragraphs.

In Chapter 2 [147], I present a new maximum margin matrix factorization algorithm for matrix completion. To cope with the scalability issue and the latent factor detection issue of existing methods for maximum margin matrix factorization, I propose an active Riemannian subspace search for M^{3}F (ARSS-M^{3}F). The main contributions of this chapter are as follows:

- Leveraging the nonlinear Riemannian conjugate gradient, I propose an efficient blockwise nonlinear Riemannian conjugate gradient (BNRCG) algorithm, which reconstructs $\mathbf{X}$ and learns multiple thresholds ϑ in M^{3}F in a joint framework. Compared to existing M3F algorithms, the proposed algorithm is much more efficient.

- Based on BNRCG, I proposed the ARSS-M^{3}F method which applies a simple and efficient pursuit scheme to automatically compute the number of latent factors, which avoids expensive model selections.

- Extensive experiments on both synthetic data sets and real-world data sets demonstrate the superior efficiency and effectiveness of the proposed methods.

In Chapter 3 [148], I present a semi-supervised learning algorithm which significantly decreases the computational complexity compared with those requiring matrix Laplacian. This chapter focuses on the two challenges of semi-supervised learning, i.e. scalability and robustness. Inspired by crowdsourcing [111, 116], I propose an efficient RObust Semi-Supervised Ensemble Learning (ROSSEL) method to approximate ground-truth labels of unlabeled data through aggregating a number of pseudo-labels generated by low-cost *weak annotators*, such as linear SVM classifiers. Meanwhile, based on the aggregated labels, ROSSEL learns an inductive SSL classifier by Multiple Label Kernel Learning (MLKL) [77]. Unlike most existing SSL algorithms, the proposed ROSSEL requires neither expensive graph Laplacian nor iterative label switching. Instead, it only needs *one* iteration for label aggregation and can be solved by an SVM solver very efficiently. The major contributions are listed as follows,

- Leveraging an ensemble of low-cost supervised weak annotators, I propose ROSSEL to efficiently obtain a weighted combination of pseudo-labels of unlabeled data to approximate ground-truth labels to assist semi-supervised learning.
- Instead of simple label aggregation strategies used in crowdsourcing (*e.g.*majority voting), ROSSEL performs a weighted label aggregation using MLKL. Meanwhile it learns an inductive SSL classifier, which only requires *one* iteration and linear time complexity w.r.t. number of data and features.
- Complexity analysis of several competing SSL methods and the proposed method is provided.

In Chapter 4 [146], I present an active learning algorithm for image classification by privileged information. To ensure the samples selected by the active learning algorithm to be representative, I exploit the diversity measurement, such that the selected samples are less similar to each other. I formulate a ratio objective function to maximize cross-media uncertainty and minimize the similarity of selected data. Then I propose to measure uncertainty and diversity for training sample selection [152]. A new optimization method is proposed to solve the proposed model, which automatically learns the optimal ratio of uncertainty to similarity. In this way, I avoid introducing the trade-off parameter between the two types of measurements. Compared to the general SDP solver, the proposed optimization algorithm can be computationally affordable for large-scale data. The main contributions of this chapter are summarized as follows:

- By exploiting privileged information, I propose a new notion of cross-media uncertainty measurement, which measures the uncertainty of unlabeled images by jointly considering visual features as the main information and text features as the privileged information.
- I propose a new method to optimize the objective without using the trade-off parameter between diversity and uncertainty.

In Chapter 5 [149], I present an online learning for imbalanced data. In this chapter, I present a unified framework for learning with imbalanced streaming data that is easily adapted to different performance measures. The proposed framework simultaneously learns multiple classifiers with various cost vectors. In particular, at each iteration, the prediction is made by a classifier which is selected randomly according to a sampling distribution, which is updated based on the current performance measures of classifiers, similarly to the well-know exponential weighted average algorithm [83]. The selection of the optimal classifier is adaptive and evolving according to the streaming data. I would like to emphasize that the proposed approach is different from the cross-validation approach, which replies on a separate validation set. Furthermore, the proposed framework enjoys a rigorous theoretical justification for the F-measure maximization. Empirical studies demonstrate that the proposed algorithm is more effective than previous online learning algorithms for imbalanced streaming data.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, USA, 2008.

[2] Charu C Aggarwal. *Data classification: algorithms and applications*. CRC Press, 2014.

[3] Mohammad Akbari, Xia Huc, Nie Liqianga, and Tat-Seng Chua. From tweets to wellness: Wellness event detection from twitter streams. In *AAAI*, 2016.

[4] Phil Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *NIPS*, 2014.

[5] Maria-Florina Balcan, Andrei Broder, and Tong Zhang. Margin based active learning. In *Learning Theory*, pages 35–50. Springer, 2007.

[6] L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. In *Proceedings of Allerton*, 2010.

[7] Mikhail Belkin, Irina Matveeva, and Partha Niyogi. Regularization and semi-supervised learning on large graphs. In *COLT*, 2004.

[8] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. On manifold regularization. In *AISTATS*, 2005.

[9] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 7:2399–2434, 2006.

[10] J. Bennett and S. Lanning. The netflix prize. In *Proceedings of the KDD Cup Workshop 2007*. ACM, 2007.

[11] Kristin Bennett, Ayhan Demiriz, et al. Semi-supervised support vector machines. In *NIPS*, 1999.

[12] Dimitri P Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[13] Prakhar Biyani, Kostas Tsioutsiouliklis, and John Blackmer. "8 amazing secrets for getting more clicks": Detecting clickbaits in news streams using article informality. In *AAAI*, 2016.

[14] Avrim Blum and Shuchi Chawla. Learning from labeled and unlabeled data using graph mincuts. In *ICML*, 2001.

[15] N. Boumal and P.-A. Absil. Rtrmc: A riemannian trust-region method for low-rank matrix completion. In *NIPS*, 2011.

[16] Olivier Bousquet, St□hane Boucheron, and G□or Lugosi. Introduction to statistical learning theory. In *Advanced Lectures on Machine Learning*, volume 3176, pages 169–207. Springer, 2003.

[17] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[18] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, October 2001.

[19] Róbert Busa-Fekete, Balázs Szörényi, Krzysztof Dembczynski, and Eyke Hüllermeier. Online f-measure optimization. In *NIPS*, pages 595–603, 2015.

[20] E. J. Candés and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[21] E. J. Candés and B. Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9:717–772, 2009.

[22] Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

[23] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[24] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *TIST*, 2(3):27, 2011.

[25] X. Chang, Z. Ma, Y. Yang, Z. Zeng, and A. G. Hauptmann. Bi-level semantic representation analysis for multimedia event detection. *IEEE Transactions on Cybernetics*, PP(99):1–18, 2016.

[26] Xiaojun Chang, Feiping Nie, Zhigang Ma, Yi Yang, and Xiaofang Zhou. A convex formulation for spectral shrunk clustering. In *AAAI*, 2015.

[27] Xiaojun Chang, Feiping Nie, Yi Yang, and Heng Huang. A convex formulation for semi-supervised multi-label feature selection. In *AAAI*, 2014.

[28] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-supervised learning*. MIT press Cambridge, 2006.

[29] Olivier Chapelle, Jason Weston, and Bernhard Schölkopf. Cluster kernels for semi-supervised learning. In *NIPS*, 2002.

[30] Lin Chen, Dong Xu, Ivor W Tsang, and Jiebo Luo. Tag-based image retrieval improved by augmented features and group-based refinement. *Multimedia, IEEE Transactions on*, 14(4):1057–1067, 2012.

[31] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7, 2006.

[32] Koby Crammer and Yoram Singer. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991, 2003.

[33] Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*, pages 208–215. ACM, 2008.

[34] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *ICML*, pages 233–240, 2006.

[35] Dennis DeCoste. Collaborative prediction using ensembles of maximum margin matrix factorizations. In *ICML*. ACM, 2006.

[36] Frédéric Delbos and Jean Charles Gilbert. Global linear convergence of an augmented lagrangian algorithm for solving convex quadratic optimization problems. *Journal of Convex Analysis*, 12:45–69, 2005.

[37] Cheng Deng, Rongrong Ji, Wei Liu, Dacheng Tao, and Xinbo Gao. Visual reranking through weakly supervised multi-graph learning. In *ICCV*, 2013.

[38] Thomas G Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer, 2000.

[39] Mark Dredze, Koby Crammer, and Fernando Pereira. Confidence-weighted linear classification. In *ICML*, 2008.

[40] G. Dror, N. Koenigstein, Y. Koren, and M. Weimer. The Yahoo! Music Dataset and KDD-Cup'11. *JMLR Workshop and Conference Proceedings*, 18:3–18, 2012.

[41] Lixin Duan, Wen Li, Ivor Wai-Hung Tsang, and Dong Xu. Improving web image search by bag-based reranking. *IEEE Transactions on Image Processing (T-IP)*, 20(11):3280–3290, 2011.

[42] Charles Elkan. The foundations of cost-sensitive learning. In *IJCAI*, 2001.

[43] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *JMLR*, 9:1871–1874, 2008.

[44] M. Fazel. *Matrix rank minimization with applications*. PhD thesis, Stanford University, 2002.

[45] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 28(2-3):133–168, 1997.

[46] Chuang Gan, Ming Lin, Yi Yang, Yueting Zhuang, and Alexander G Hauptmann. Exploring semantic inter-class relationships (sir) for zero-shot action recognition. In *AAAI*, 2015.

[47] Wei Gao, Rong Jin, Shenghuo Zhu, and Zhi-Hua Zhou. One-pass AUC optimization. In *ICML*, pages 906–914, 2013.

[48] Claudio Gentile. A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research*, 2:213–242, March 2002.

[49] Zoubin Ghahramani. Unsupervised learning. In *Advanced lectures on machine learning*, pages 72–112. Springer, 2004.

[50] Mark Goadrich, Louis Oliphant, and Jude W. Shavlik. Gleaner: Creating ensembles of first-order clauses to improve recall-precision curves. *Machine Learning*, 64(1-3):231–261, 2006.

[51] Yahong Han, Zhongwen Xu, Zhigang Ma, and Zi Huang. Image classification with manifold learning for out-of-sample data. *Signal Processing*, 93(8):2169–2177, 2013.

[52] Yahong Han, Yi Yang, Zhigang Ma, Haoquan Shen, N. Sebe, and Xiaofang Zhou. Image attribute adaptation. *Multimedia, IEEE Transactions on*, 16(4):1115–1126, June 2014.

[53] Yahong Han, Yi Yang, Yan Yan, Zhigang Ma, Nicu Sebe, and Xiaofang Zhou. Semisupervised feature selection via spline regression for video semantic recognition. *TNNLS*, 26(2):252–264, 2015.

[54] Elad Hazan. *Introduction to Online Convex Optimization*. now Publishers Inc., 2015.

[55] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[56] Xiaofei He, Wanli Min, Deng Cai, and Kun Zhou. Laplacian optimal design for image retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 119–126. ACM, 2007.

[57] Mark Herbster and Manfred K. Warmuth. Tracking the best expert. *Machine Learning*, 32(2):151–178, 1998.

[58] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR*. ACM, 1999.

[59] Cho-Jui Hsieh and Peder Olsen. Nuclear norm minimization via active subspace selection. In *ICML*, 2014.

[60] Junjie Hu, Haiqin Yang, Irwin King, Michael R Lyu, and Anthony Man-Cho So. Kernelized online imbalanced learning with fixed budgets. In *AAAI*, pages 2666–2672, 2015.

[61] Jin Huang, Feiping Nie, and Heng Huang. Robust discrete matrix completion. In *AAAI*, 2013.

[62] S.J. Huang, R. Jin, and Z.-H. Zhou. Active learning by querying informative and representative examples. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(10):1936–1949, 2014.

[63] Prateek Jain and Ashish Kapoor. Active learning for large multi-class problems. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 762–769. IEEE, 2009.

[64] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[65] Xiao-Yuan Jing, Qian Liu, Fei Wu, Baowen Xu, Yangping Zhu, and Songcan Chen. Web page classification based on uncorrelated semi-supervised intra-view and inter-view manifold discriminant feature extraction. In *IJCAI*, 2015.

[66] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *ICML*, 1999.

[67] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, pages 291–307, 2005.

[68] Purushottam Kar, Harikrishna Narasimhan, and Prateek Jain. Online and stochastic gradient methods for non-decomposable loss functions. In *NIPS*, pages 694–702, 2014.

[69] Alexandros Karatzoglou, Markus Weimer, and Alex J Smola. Collaborative filtering on a budget. In *AISTATS*, 2010.

[70] Ron Kohavi and Foster Provost. Glossary of terms. *Machine Learning*, 30(2/3):271–274, 1998.

[71] J. Krapac, M. Allan, J. Verbeek, and F. Jurie. Improving web image search results using query-relative classifiers. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1094–1101, June 2010.

[72] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1106–1114, 2012.

[73] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, 2013.

[74] Wen Li, Li Niu, and Dong Xu. Exploiting privileged information from web data for image categorization. In *Computer Vision–ECCV 2014*, pages 437–452. Springer, 2014.

[75] Y-F Li, James T Kwok, and Z-H Zhou. Cost-sensitive semi-supervised support vector machine. In *AAAI*, 2010.

[76] Yu-Feng Li, James T Kwok, and Zhi-Hua Zhou. Semi-supervised learning using label mean. In *ICML*, 2009.

[77] Yu-Feng Li, Ivor W Tsang, James T Kwok, and Zhi-Hua Zhou. Tighter and convex maximum margin clustering. In *AISTATS*, pages 344–351, 2009.

[78] Yu-Feng Li, Ivor W. Tsang, James T. Kwok, and Zhi-Hua Zhou. Convex and scalable weakly labeled svms. *JMLR*, 14(1):2151–2188, 2013.

[79] Guohua Liang and Anthony G Cohn. An effective approach for imbalanced classification: Unevenly balanced bagging. In *AAAI*, pages 1633–1634. AAAI Press, 2013.

[80] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical report, UIUC, 2010.

[81] A. Lindner and S. Susstrunk. Semantic-improved color imaging applications: It is all about context. *Multimedia, IEEE Transactions on*, 17(5):700–710, May 2015.

[82] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

[83] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, pages 212–261, 1994.

[84] Yang Liu and Mingyan Liu. Finding one's best crowd: Online learning by exploiting source similarity. In *AAAI*, 2016.

[85] Xinyan Lu, Fei Wu, Siliang Tang, Zhongfei Zhang, Xiaofei He, and Yueting Zhuang. A low rank structural large margin method for cross-modal ranking. In *SIGIR*, 2013.

[86] Zhiwu Lu, Xin Gao, Liwei Wang, Ji-Rong Wen, and Songfang Huang. Noise-robust semi-supervised learning by large-scale sparse coding. In *AAAI*, 2015.

[87] Michal Lukasik and Trevor Cohn. Convolution kernels for discriminative learning from streaming text. In *AAAI*, 2016.

[88] Oisin Mac Aodha, Neill DF Campbell, Jan Kautz, and Gabriel J Brostow. Hierarchical subquery evaluation for active learning on a graph. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 564–571. IEEE, 2014.

[89] Hamed Masnadi-Shirazi and Nuno Vasconcelos. Risk minimization, probability elicitation, and cost-sensitive svms. In *ICML*, 2010.

[90] G. Meyer, S. Bonnabel, and R. Sepulchre. Linear regression under fixed-rank constraints: A riemannian approach. In *ICML*, 2011.

[91] Bamdev Mishra, K Adithya Apuroop, and Rodolphe Sepulchre. A riemannian geometry for low-rank matrix completion. *arXiv preprint arXiv:1211.1550*, 2012.

[92] Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *NIPS*, 2007.

[93] Ion Muslea, Steven Minton, and Craig A Knoblock. Active learning with multiple views. *Journal of Artificial Intelligence Research*, pages 203–233, 2006.

[94] T. T. Ngo and Y. Saad. Scaled gradients on grassmann manifolds for matrix completion. In *NIPS*, 2012.

[95] Hieu T Nguyen and Arnold Smeulders. Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 79. ACM, 2004.

[96] Feiping Nie, Heng Huang, and Chris Ding. Low-rank matrix recovery via efficient schatten p-norm minimization. In *AAAI*, 2012.

[97] Shameem Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. Optimizing f-measures by cost-sensitive classification. In *NIPS*, 2014.

[98] Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. Collecting image annotations using amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 139–147. Association for Computational Linguistics, 2010.

[99] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos. A new approach to cross-modal multimedia retrieval. In *Proceedings of the international conference on Multimedia*, pages 251–260. ACM, 2010.

[100] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum rank solutions to linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52(3), 2010.

[101] Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *NIPS*, 2011.

[102] Jasson D. M. Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *ICML*, 2005.

[103] Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 2012.

[104] Clayton Scott. Surrogate losses and regret bounds for cost-sensitive classification with example-dependent costs. In *ICML*, 2011.

[105] Clayton Scott. Calibrated asymmetric surrogate losses. *Electronic Journal of Statistics*, 6:958–992, 2012.

[106] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[107] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin, Madison, 2010.

[108] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[109] Shai Shalev-Shwartz and Yoram Singer. A new perspective on an old perceptron algorithm. In *COLT*, 2005.

[110] Jude W Shavlik and Thomas Glen Dietterich. *Readings in machine learning*. Morgan Kaufmann, 1990.

[111] Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *KDD*, 2008.

[112] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[113] Vikas Sindhwani, Wei Chu, and S Sathiya Keerthi. Semi-supervised gaussian process classifiers. In *IJCAI*, 2007.

[114] Vikas Sindhwani, Partha Niyogi, Mikhail Belkin, and Sathiya Keerthi. Linear manifold regularization for large scale semi-supervised learning. In *Proc. of the 22nd ICML Workshop on Learning with Partially Classified Training Data*, 2005.

[115] Alexander J Smola and Risi Kondor. Kernels and regularization on graphs. In *Annual Conference on Computational Learning Theory*, 2003.

[116] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *EMNLP*, 2008.

[117] Nathan Srebro, Jason D. M. Rennie, and Tommi S. Jaakola. Maximum-margin matrix factorization. In *NIPS*. MIT Press, 2005.

[118] Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009:4, 2009.

[119] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

[120] Mingkui Tan, Ivor W. Tsang, Li Wang, Bart Vandereycken, and Sinno Jialin Pan. Riemannian pursuit for big matrix recovery. *ICML*, 2014.

[121] Kim-Chuan Toh and Sangwoon Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, 6:615––640, 2010.

[122] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66, 2002.

[123] Ivor W. Tsang and James T Kwok. Large-scale sparsified manifold regularization. In *NIPS*, 2006.

[124] Bart Vandereycken. Low-rank matrix completion by riemannian optimization. *SIAM J. Optim.*, 23(2):1214––1236, 2013.

[125] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[126] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5):544–557, 2009.

[127] V Vovk. Aggregating algorithms. In *COLT*, 1990.

[128] De Wang, Feiping Nie, and Heng Huang. Large-scale adaptive semi-supervised learning via unified inductive and transductive model. In *KDD*, 2014.

[129] Jialei Wang, Peilin Zhao, and Steven C. H. Hoi. Cost-sensitive online classification. In *ICDM*, pages 1140–1145, 2012.

[130] Sen Wang, Zhigang Ma, Yi Yang, Xue Li, Chaoyi Pang, and A.G. Hauptmann. Semi-supervised multiple feature analysis for action recognition. *Multimedia, IEEE Transactions on*, 16(2):289–298, Feb 2014.

[131] Shuo Wang, Leandro L Minku, and Xin Yao. Resampling-based ensemble methods for online class imbalance learning. *IEEE Transactions on Knowledge and Data Engineering*, 27(5):1356–1368, 2015.

[132] Wei Wang and Zhi-Hua Zhou. On multi-view active learning and the combination with semi-supervised learning. In *Proceedings of the 25th international conference on Machine learning*, pages 1152–1159. ACM, 2008.

[133] Markus Weimer, Alexandros Karatzoglou, Quoc Viet Le, and Alex Smola. Maximum margin matrix factorization for collaborative ranking. *NIPS*, 2007.

[134] Markus Weimer, Alexandros Karatzoglou, and Alex Smola. Improving maximum margin matrix factorization. *Machine Learning*, 72(3):263–276, 2008.

[135] Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a non-linear successive over-relaxation algorithm. *Math. Program. Comput.*, 4(4):333–361, 2012.

[136] Fei Wu, Zhou Yu, Yi Yang, Siliang Tang, Yin Zhang, and Yueting Zhuang. Sparse multi-modal hashing. *Multimedia, IEEE Transactions on*, 16(2):427–439, 2014.

[137] Mingrui Wu. Collaborative filtering via ensembles of matrix factorizations. In *Proceedings of KDD Cup and Workshop*, 2007.

[138] Yi Wu, I. Kozintsev, J.-Y. Bouguet, and C. Dulong. Sampling strategies for active learning in personal photo retrieval. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 529–532, July 2006.

[139] Yu Wu, Yutian Lin, Xuanyi Dong, Yan Yan, Wanli Ouyang, and Yi Yang. Exploit the unknown gradually: One-shot video-based person re-identification by stepwise learning. In *CVPR*, 2018.

[140] Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *J. Mach. Learn. Res.*, 11:2543–2596, 2010.

[141] Shijie Xiao, Wen Li, Dong Xu, and Dacheng Tao. FaLRR: A Fast Low Rank Representation Solver. In *CVPR*, 2015.

[142] Shijie Xiao, Mingkui Tan, and Dong Xu. Weighted block-sparse low rank representation for face clustering in videos. In *ECCV*, 2014.

[143] Minjie Xu, Jun Zhu, and Bo Zhang. Nonparametric max-margin matrix factorization for collaborative prediction. In *NIPS*, 2012.

[144] Minjie Xu, Jun Zhu, and Bo Zhang. Fast max-margin matrix factorization with data augmentation. In *ICML*, 2013.

[145] Zenglin Xu, Irwin King, Michael Rung-Tsong Lyu, and Rong Jin. Discriminative semi-supervised feature selection via manifold regularization. *TNN*, 21(7):1033–1047, 2010.

[146] Yan Yan, Feiping Nie, Wen Li, Chenqiang Gao, Yi Yang, and Dong Xu. Image classification by cross-media active learning with privileged information. *IEEE Transactions on Multimedia*, 18(12):2494–2502, 2016.

[147] Yan Yan, Mingkui Tan, Ivor W Tsang, Yi Yang, Chengqi Zhang, and Qinfeng Shi. Scalable maximum margin matrix factorization by active riemannian subspace search. In *IJCAI*, pages 3988–3994, 2015.

[148] Yan Yan, Zhongwen Xu, Ivor W Tsang, Guodong Long, and Yi Yang. Robust semi-supervised learning through label aggregation. In *AAAI*, pages 2244–2250, 2016.

[149] Yan Yan, Tianbao Yang, Yi Yang, and Jianhui Chen. A framework of online learning with imbalanced streaming data. In *AAAI*, pages 2817–2823, 2017.

[150] Tianbao Yang, Mehrdad Mahdavi, Rong Jin, Jinfeng Yi, and Steven C. H. Hoi. Online kernel selection: Algorithms and evaluations. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*, 2012.

[151] Yi Yang, Zhigang Ma, Alexander G. Hauptmann, and Nicu Sebe. Feature selection for multimedia analysis by sharing information among multiple tasks. *IEEE Transactions on Multimedia*, 15(3):661–669, 2013.

[152] Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G Hauptmann. Multi-class active learning by uncertainty sampling with diversity maximization. *International Journal of Computer Vision*, pages 1–15, 2014.

[153] Yi Yang, Feiping Nie, Dong Xu, Jiebo Luo, Yueting Zhuang, and Yunhe Pan. A multimedia retrieval framework based on semi-supervised ranking and relevance feedback. *TPAMI*, 34(4):723–742, 2012.

[154] Yi Yang, Yue-Ting Zhuang, Fei Wu, and Yun-He Pan. Harmonizing hierarchical manifolds for multimedia document semantics understanding and cross-media retrieval. *Multimedia, IEEE Transactions on*, 10(3):437–446, 2008.

[155] Kai Zhang, Liang Lan, J.T. Kwok, S. Vucetic, and B. Parvin. Scaling up graph-based semisupervised learning via prototype vector machines. *TNNLS*, 26(3):444–457, March 2015.

[156] Peilin Zhao, Steven C. H. Hoi, Rong Jin, and Tianbao Yang. Online auc maximization. In *ICML*, pages 233–240, 2011.

[157] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. *NIPS*, 16(16):321–328, 2004.

[158] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[159] Xiaojin Zhu. *Semi-Supervised Learning with Graphs*. PhD thesis, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 5 2005.

[160] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.

[161] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.