Machine Learning Methods for Personalized Email Prioritization

Shinjae Yoo

CMU-LTI-10-011

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213

June 10, 2010

Thesis Committee:
Yiming Yang, *Chair*
Jaime Carbonell
Jamie Callan
Micahel Freed, *SRI*

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Language and Information Technologies

Abstract

Email is one of the most prevalent communication tools today, and solving the email overload problem is pressingly urgent. A good way to alleviate email overload is to automatically prioritize received messages f1ording to the priorities of each user. However, research on statistical learning methods for fully personalized email prioritization has been sparse due to privacy issues, since people are reluctant to share personal messages and priority judgments with the research community. It is therefore important to develop and evaluate personalized email prioritization methods under the assumption that only limited training examples can be available, and that the system can only have the personal email data of each user during the training and testing of the model for that user.

We focus on three aspects: 1) we investigate how to express the ordinal relations among the priority levels through classification and regression. 2) we analyze personal social networks to capture user groups and to obtain rich features that represent the social roles from the viewpoint of a particular user. 3) We also developed a semi-supervised (transductive) learning algorithm that propagates importance labels from training examples to test examples through messages and user nodes in a personal email network. These methods together enable us to obtain both a better modeling priority and an enriched vector representation of each new email message.

Our contribution is as follows. First, we have successfully collected multiple users' private email data with their fine grained personal priority labels. Second, we apply and propose learning approaches from multi-type information such as text, and sender / recipients information. Third, to supplement additional information to sparse training data, we identify the importance of a contact and similar contacts from social networks. Fourth, we exploit a semi-supervised learning on the personal email networks. Finally, we conducted and completed systematic evaluations with respect to email prioritization, targeting the discovery of better modeling of email priorities. Through our suggested approaches, email prioritization alleviates email glut and should help our daily productivity.

This thesis is dedicated to my wife, *Hayan Lee* for her love and support.

Acknowledgements

First and foremost, I would like to thank God for giving me wisdom and guidance throughout my life. There are so many people I would like to thank - this thesis completed based on their help. My advisor, Yiming Yang, continuously gave me support and encouragement on this thesis work. I would also like to thank my committee, Jaime Carbonell, Jamie Callan, and Michael Freed, for all the generous comments and support. I would like to special thanks Il-Chul Moon, Frank Lin, Richard Wang on discussion, code, and encouragement. I thank Eunice Kim, Kevin Gimpel, Borah Lee, and Frank Lin for thesis proof reading. I thank group members, Jian Zhang, Fan Li, Chang Yi, Monica Rogati, Bryan Kisiel, Bryan Klimt, Abhimanyu Lad, Sachin Agawal, Henry Shu, Abhay Harpale, Konstantin Salomatin and Siddharth Gopal. I thank LTI staffs, Mary Jo Bensasi, Brooke Hyatt, Linda Hager, Radha Rao, Donna Gates, Stacey Young, Dana Houston. I thank LTI Students, Jean Oh, Jungwoo Ko, Jaedong Kim, Chanwoo Kim, Moonyoung Kang, Amr Ahmed, Hassan Al-Haj, Justin Betteridge, Ming-yu Chen, Yee Man Cheng, Elsas Jonathan, Kenneth Heafield, Sanjika Hewavitharana, Ni Lao, Yung-hui Li, Henry Lin, Udhyakumar Nallasamy, Thuy Linh Nguyen, Paul Ogilvie, Nico Schlaefer, Hideki Shima, Kishore Sunkeswari Prahallad, Yi-Chia Wang, Grace Yang, Le Zhao, Pinar Donmez, Einat Minkov, Wen Wu, Vitor Carvalho, Bing Zhao, and Yi Zhang. I also thank, my heavenly family members, Pastor Eunsoo Lee, Won Young Rhee, Subyoung Lim, Thomas Song, Brother Hunjae Jung, Taehee Jeong, Mason Kim, Taehoon Kim, Jun Park, Jong-do Park, Jeongheon Park, Jongho Yoon, In-ho Song, JongHyup Lee, Myung Roh, Namsuk Bae, Chongho Lee Hyuunjin Lee, Dong Hyun Ku, Donghun Lee, Minwoo Yun, Dongsu Han, Mu Kyum Kim, Jaewook Kim, James Park, Ildoo Kim, Seungjun Kim, Taewon Seo, Seungmin Roh, Sister Eun-Ryeong Hahm, Minjung Kim, Hyung-Jeong Yang, Ji Eun Kim, Hayeon Lee, Aelee Kim, Jinkyung Kim, Jiyoung Song, Somin Lee, Heejin Park, Kyungin Oh, Hyungjoo Kang, Sunhee Kim, Grace Huh, and Gahgene Gweon.

Contents

List of Figures

List of Tables

1 Introduction

Email Prioritization aims at sorting or filtering incoming unread emails with respect to each user's criteria. This chapter introduces email overload problems and our approaches. Then it differentiates our work with others. This chapter also presents thesis statement and contributions.

1.1 Motivation and Challenges

Email is one of the most prevalent personal and business communication tools today; however, it is not without significant drawbacks. In contrast to telephone conversations or face-to-face meetings, communication through email is asynchronous in the sense that we receive all messages (after some spam filtering) in the same way regardless of our level of interest, and a single sender can flood multiple receivers (unlike telephone or instant messaging). Users are left with the burden of having to process a large volume of email messages of differing importance. This tedious task has been shown to cause significant negative effects on both personal and organization performance [16, 42]. There is an urgent need to solve this information overload problem, i.e., we need to develop systems that can automatically learn personal priorities for each user, and that can identify personally interesting and important messages among others for user's attention. To alleviate this *email overload* problem, this thesis targets to identify the priorities of unread emails through machine learning approaches.

The first obstacle in email prioritization is *privacy* issue. Since email overload problem has been raised in 1982 [17], few researches have been done on email prioritization except spam filtering. Especially email prioritization researches using machine learning is very rare. One of the critical reasons for this phenomenon is the privacy issue. Unlike news corpus or web documents, in case of email messages people need to share personal email contents although they do not mind to share spams. Anonymization can be one of the solutions for this problem [4, 30] but after anonymization, many important information could not be extracted such as speech acts or temporal expression anchoring. As a result, we must carefully design experiments before doing any email related experiments.

Personalization is also a tough problem. By personalization, we mean that the same email may have different priority levels to different recipients so that we need each person's priority labels for their own emails. Suppose that a grant proposal email sent to multiple recipients. Depending on each user, the importance of the same email could differ dramatically. If the user is irrelevant, the message would be classified as spam. But for principal investigator or a key contributor of the proposed work, it will be very important. Recently there are some publicly available datasets such as Enron [27]. However, these datasets do not have the recipient's personal labels.

Sparse training data for each user makes personalized prioritization of emails particularly challenging. It is a crucial problem not only for building prioritization models but also for actual applications. If a deployed email prioritization system requires lots of training labels, users refuse to use the system. Especially, busy users used to hesitate spending time on labeling or learning new tools. Therefore we must find an effective way to overcome sparse training data.

Given these privacy, personalization and sparse training data challenges, we have to build appropriate machine learning models for email prioritization and evaluate them systematically. Due to limited research foundings for email prioritization, it is not clear what is the right direction for email prioritization and what are the right evaluation metrics. For instance, we may model the

multiple priority levels through ordinal regression which encodes the relations among the different priority levels. However, the ordinal regression including support vector ordinal regression and logistic ordinal regression are worse than classification approach including SVM classification and logistic regression classifier.

1.2 Our Approach

This thesis models priority in terms of *intrinsic importance*, although we collected the importance and *the urgency* of an email, known as *Eisenhower* priority matrix [13]. The importance stands for how important the email is to the recipient and urgency stands for how urgent the email is to the recipient with respect to the recipient's reaction. For instance, if the email is related to a grant proposal and the recipient is actively engaged, then the importance of emails belongs to this grant proposal is very high. However, if an email has no specific deadline, the urgency of the email is not very urgent. Horvitz et al. [25] modeled the criticality as their priority. They defined the criticality of a notification as the expected cost of delayed action associated with reviewing the message, which modeled in terms of only the urgency. Denning [17], Cadiz et al. [9] and Dabbish et al. [16, 15, 14] modeled only the importance of an email as a priority. The reason people used the same terminology, the priority, for these two different factors, the urgency and the importance, is that both factors contribute to the priority.

Priority is modeled with five levels in terms of importance. Horvitz et al. [25] and Johansen et al. [30] modeled priority into two levels, high and low priority. In that case, it is basically similar to spam filtering. So we do not set just two levels. To make prioritization system realistic, at least three levels or more are required, low, medium, and high. During user study of this thesis, it was observed that the most dominant priority level is medium. Furthermore, depending on the amount of email receiving, many people made distinction between highest priority and higher priority as well as between lowest priority and lower priority on top of medium priority level. Therefore we defined five levels for the priority. The other extreme is only a rank based priority which sorts all unread emails. It could be natural to sort unread emails but Hasegawa and Ohara [23] requested label all ranks. Horvitz et al. [25] modeled 100 levels from 1 to 100 during evaluation. Even this 100 levels are quite fuzzy to the users too because a user may have difficulty in distinguishing between 32 and 33 priority levels. Instead of requesting every regression levels, we may learn a partial rank based preference function to alleviate heavy load labeling burden. But it may not be able to associate the predicted rank with certain actions. For instance, depending on the priority levels, we may provide email coloring to show importance level or send SMS message to one's cell phone. Moreover Cadiz et al. [9] used five priority levels on their survey questions to identify the importance relations.

We proposed a fully personalized methodology for technical development and evaluation. By fully personalized we mean that only the personal email data (textual or social network information) of each user is available for the system during the training and testing of the user-specific model. This is an important assumption for the generality of personalized email prioritization methods, i.e., we cannot rely on the availability of centralized access to customer private data, neither in the development circle nor in the evaluation phase, and we cannot take the liberty to use a particular user's private data to build models for other users because the potential leak of private information across users. This assumption makes our work in this paper fundamentally different from those in spam filtering and other previous work on email-based prediction tasks.

We investigate various machine learning methods to model priorities including classification and (ordinal) regression. How to model ordinal priority level is not studied well. The classification model uses multiple models for each priority levels but (ordinal) regression model uses a single model with multiple thresholds to determine multiple levels. Based on our pilot study, we observed that separate models for each priority level such as classification is better than a single model with multiple thresholds such as ordinal regression. However, the multiple models can not take advantages of the adjacency priority relations natively. So we propose to use multiple models with the considerations of adjacent priority relations. It is also interesting that the priority models are consistent among the users.

To cope with the lack of training data, we would like to explore additional information which requires any or partial prior priority labels. Since email is an interactive communication media, we may find the interactions among the users by analyzing the relations of senders and receivers, from which we can find social networks. We may identify who is the important person from my email social network by analyzing social importance metric or who are similar to a priori known person through social clustering. We also investigate the effects of email specific meta information such as attachments, the length of email, the number of recipients, etc.

1.3 Related Work

1.3.1 Spam Filtering

Spam filtering [37, 38, 31] is a kind of email prioritization but the spam filtering only focuses on filtering unwanted emails or two level prioritization systems. Sahami et al [37] reported surprisingly good results in Spam filtering using Naive Bayes classifiers. After Sahami, lots of duplicated experimental results confirm Sahami's finding. Zhang et al [47] reported similar results on several different spam collections with various machine learning algorithms. They also reported both header and body information were important in identifying spam. However, spam filtering was identified more difficult problems than what Sahami discovered because of the attacks of statistical classifiers [43]. One attack out of four identified attacks by Wittel [43] is tokenization attack, which is working against the feature selection (tokenization) of a message by splitting or modifying key message features such as splitting up words with spaces and using HTML layout tricks. To overcome these attacks, Boykin and Roychowdhury [6] utilized social networks to fight spam. Gray and Haahr [22] proposed collaborative spam filtering methods. Goodman et al [21] summarized other advancements except machine learning in Spam filtering and they reported that Spam filtering was under control to the user but the battle between spammer and spam resarcher was on going. However, these spam filtering alleviate the overload of the recipients to certain degree but it can not be solution for email overload because the recipients still need to read all incoming legitimate emails and spam filters have not discriminated the difference among important emails.

1.3.2 Prior Email Prioritization

Among the early efforts in email prioritization, Horvitz et al. [25] built an email alerting system which used Support Vector Machines to classify newly arrived email messages into two categories, i.e., high or low in terms of utility. Probabilistic scores were also provided along with the system-made predictions. Personalization, however, was not considered in their method, and priority

modeling and social network analysis were not their technical focus.

Hasegawa and Ohara [23] proposed to use Linear Regression [28] and used two levels for evaluation. They used about one thousand rules to extract features. Even though they mentioned the priority should be personalized, they again evaluated their model on only one user. No systematic evaluation of different priority modeling approaches and social network analysis were addressed.

Not much work has been done on email prioritization research and none of the prior works evaluated their models on multiple users considering the personalization issues. Therefore, it is difficult to draw meaningful observations from the prior works.

1.3.3 Social Clustering

Tyler et al. [39] utilized Newman clustering algorithm to discover social structures automatically from email messages. They found that the automatically discovered social structures are quite similar, or consistent, with human interpretation of organizational structures. They also used email social networks to identify social leaders. However, they did not use the social network analysis (clusters or leadership scores) to prioritize email messages.

Gomes et al. [20] used email messages to automatically group users in two ways, i.e., by sender clusters and by recipient clusters, respectively. The senders were clustered based on similarity of their recipient lists, and the recipients were clustered based on similarity of their sender lists as well; email contents were not used. They examined the use of those clusters in spam detection, i.e., to separate spam messages from non-spam messages. Prioritization among non-spam messages, however, was not addressed.

McCallum et al. [33] modeled the links between sender and recipients along with direction-sensitive topic distribution built on Latent Dirichlet Allocation (LDA) [5], called Author-Recipient-Topic (ART) model. With ART model, we could discover the probabilistic topic distribution according to the relationships between people. Then they extended ART model to include social roles, called Role-ART (RART) model. ART model encompassed text with social network and it could be good features for email prioritization but we did not utilize it mainly because the slow speed of LDA style algorithms keep us from using it on email prioritization.

Johansen et al. [30] proposed a social clustering approach to importance prediction of email messages. They collected email data from multiple users and induced social clusters of users. For each user, some clusters are treated as ”important” and the others are not. The importance of each test instance of email message is predicted based on the cluster membership of its sender: if the sender belongs to an important cluster, then the messages is considered important; otherwise, it is predicted as not important. The fundamental difference in their method from ours is that their clusters were induced from a community social network, not based on personal social networks. In addition, they only focused on social associations, not taking any textual features into account in the modeling and the prediction of importance.

1.3.4 Social Importance Metrics

Various social metrics has been used in email research. Neustaedter et al. [34] defined metrics for measuring the social importance of individuals based on the observations in the email fields: from, to and cc, and in the recorded actions of replying and reading. They used these metrics for retrieving old email messages rather than prioritizing incoming email messages.

Boykin and Roychowdhury [7] used clustering coefficients as enriched features to represent email messages and a Bayesian classifier to detect spam messages. Martin et al. [32] used the out-degree (the number of unique recipients) and in-degree (the number of unique senders) of each person in an email social network to detect worms which propagated through the email messages. Prioritization among non-spam messages was again not addressed by those methods.

1.4 Thesis Statement

Email prioritization can be done effectively by learning individual preferences and priorities of each user. The most dramatic improvement comes from the proper modeling of personalized email priority, our proposed ensemble learning. Further improvement can be achieved by combining the textual content of the email(e.g. subject, body) and the induced social relations between the email recipient and the various senders. With proper modeling and text with enriched social relations, we can effectively categorize email by importance for each user who provides sufficient importance labels for supervised training.

1.5 Contributions

This thesis presents the first study with several statistical classification and clustering methods addressing the personalized email prioritization problem based on personal importance judgments by multiple users. We constructed a new dataset, email messages from each user, and systematically evaluate several hypothesis models. More specifically, our contribution is as follows:

1. We created a new collection of personal email data with fine-grained importance levels. Previous work used datasets with only two priority levels, i.e., spam vs. non-spam [30], which are not sufficient for discriminating personal importance levels on non-spam email messages. On the other hand, past research with human subjects indicates that users would have difficulties in producing consistent labels if too many levels were required [29, 3]. Hence, we took a middle ground with 5 levels. To our knowledge, this is the first multi-user email prioritization dataset with fine-grained importance labels.

2. We proposed a fully personalized methodology for technical development and evaluation. By fully personalized we mean that only the personal email data (textual or social network information) of each user is available for the system during the training and testing of the user-specific model. This is an important assumption for the generality of personalized email prioritization methods, i.e., we cannot rely on the availability of centralized access to customer private data, neither in the development circle nor in the evaluation phase, and we cannot take the liberty to use a particular user's private data to build models for other users because the potential leak of private information across users. This assumption makes our work in this thesis fundamentally different from those in spam filtering and other previous work on email-based prediction tasks.

3. We developed a supervised classification framework for modeling personal email message priorities, and for predicting importance levels for new messages. Especially, we explored and proposed the best model for a fully personalized email prioritization. The personalized email prioritization can be modeled by several different approaches and among them,

we identified two main stream of approaches, classification based and (ordinal) regression based approach. We compared these two approaches in terms of the model assumptions and identified the best working conditions for each approach. Further, we proposed models taking advantages of both approaches.

4. We proposed to use enriched representation of each input email message, especially in the part that represent the contact persons (sender or recipients in the CC list) in the message. We explored four different types of enriched features that are automatically induced based on personal social networks and meta information from email headers as follows:

   - **Clustering contact persons based on personal social networks** We want to capture social groups among senders and recipients, which can be learned from personal email messages without importance labels (unsupervised learning). For example, email messages from two different senders who are members of the same team may carry similar importance. A personal social network is constructed for each user using his or her own data. Finding closely-associated user groups from the personal perspective enables us to estimate the expected importance level per group, as a strategy for improving the robustness of importance prediction when training data are relative sparse.
   - **Measuring social importance of contacts** We want to capture leadership levels of individual contacts, and we define eight centrality measures that can be automatically computed using the graph structure of each personalized social network. Most of those metrics have been commonly used in Social Network Analyses (SNA) research for spam filtering; however, their use in personalized email prioritization has not been studied in depth. As personal social networks are different from user to user, using multi-dimensional leadership metrics to jointly characterize different users would lead to more robust predictions than using any single metric alone.
   - **Semi-supervised importance propagation** When importance labels are available for some email messages (e.g. older messages) but not available for other messages (e.g. newer ones), we can use the personal social network of each user to propagate the importance scores from messages to contacts, then from contacts to messages, and repeat the propagation until all the scores are stabilized. By doing so, we make another use of personal social networks, i.e., leveraging the transitivity of importance scores through personal social connections.
   - **Meta information** Given an email message, we may extract message size, the number of attachments, whether the email is a reply to the recipient's previously sent message, whether the recipient's email address is listed in To or CC list, etc. The meta information extracted from the email header could be meaningful. We investigated the effects of such meta features on the personalized email prioritization.

5. We present an empirical evaluation of both (1) identifying the best personalized prioritization models and (2) the usefulness of the enriched representation using social network and meta information. First, we validated each modeling approach including our proposed models with realistic personalized email prioritization data, ordinal regression benchmark datasets and our synthetic dataset to test the controlled environment. We confirmed that our proposed

approaches are more effective than ordinal regression in personalized email prioritization dataset although the later has been the natural choice for predicting ordinal output in general. The synthetic dataset experiments confirmed which approach would work best given different data distributions. Second, with the enriched representation using social network and meta information, we achieved further error-rate reduction. Our experiments also show that for different users we need to rely on very different social network features for accurate email priority predictions and that our system can automatically discover and utilize those features.

2 Data Collection and Evaluation

Although this thesis is not the first one in email prioritization, the previous works have not evaluated their algorithms or systems on an multiple users because of the privacy issue and the difficulty of personalization. In this chapter, we introduce what are available information from email and how we collected data from our email client program and explain the user study that allows us to collect email data. After that, we explore several evaluation metrics for email prioritization.

2.1 Features in Email

We can capture six types of information from email: text, social link (sender or recipients), threading, meta information, attachment itself, and user feedbacks.

The text is available like any news articles. It also has title and body text. In other words, we may apply text mining techniques such as classification or clustering on email data. However, email has much rich representation than news article or other format of documents.

Email explicitly shows who are the recipients except bcc. News articles tend to write to general public but email has a specific recipient list. Also we may induce social networks from this sending and receiving relations [39]. We may draw a contact network, which has edges between senders and recipients or an email network, which includes email itself as a node and has edges between email and sender or between email and recipients.

Email contains the discussion context information through email threads. The thread is a series of email communication about a topic but practically, we define email thread as a series of email messages that share the same title within a limited time period.

Email also contains meta information such as time stamp, the length of email, the number of attachments, the number of recipients, and email body text type such as HTML or plain text.

The attached file itself can be served as additional information but we need to convert it to text or extract meaningful information from the attached file. For instance, an image file is difficult to be used except filename but if the attached file is a PDF or Word file, then it may be easy to extract additional information.

Finally, we may collect user interaction with email client, also called implicit feedback such as reading time, writing time, re-reading frequency of an email, and whether the email is replied, forwarded, or replied to all. These user interaction features can be extracted from email client directly. Note that these information is not available when we predict the priority of a new email.

This thesis use text features, sender and recipient list as our base features and the induced social networks are considered in Chapter 4.1, 4.2 and 4.3. In Chapter 4.4, we discuss the effects of meta features. This thesis does not consider email threading or user interaction features as candidate features.

2.2 Data Collection

Although email prioritization is very important and urgent research, it is not an easy research due to the difficulty of collecting email messages with labels. As our target is personalized email prioritization, we could not use publicly available email corpus such as Enron [27]. If somebody labels whole emails of users or corpus, then it is no longer personalized and it could not correctly repre-

sent the recipients' interest and thus we could not verify our proposed models correctly. Therefore we have to collect email and its labels by ourselves.

The first obstacle was going through IRB (Institutional Review Board). Because the information we collected from the subject had serious concerns on human subject matters and potential to have social impact, it was not an easy process. Therefore, we offered selectively Opt-In / Opt-Out message functions, keyword based anonymization, encrypted storage of dumped email messages, delayed submission to allow change the subject mind, cancellation of submitted email messages even after submission, and anytime cancellation of participation of research.

The second obstacle was actually implementing data collection tools and recruiting the subjects. We did the first data collection process but due to lack of the amount of collected messages, we went through the second data collection process in addition. The following is detailed descriptions of the design goal of such process and functionality of implemented tools and the collected results.

2.2.1 The First Data Collection

During the first data collection period, our highest concern is how to protect the subject privacy. In our study, although we provided anonymization functionality, we asked the subject to release their textual data not to be anonymized as much as they can because we need to understand why a certain algorithm fails and how we can improve our algorithm in response.

Due to its popularity among staff members and some students and faculty, we choose Microsoft Outlook as our email data collection platform, shown in Figure 2.1. All the user interaction functions are listed on toolbar from SUBMIT to STATUS button.

First of all, we allow the subject to selectively submit their emails. We provide the manual Opt-In / Opt-Out function for each email and the subject may choose which one is default. If the email is private to the subject, the subject may Opt-Out in case of default Opt-In mode or may not select Opt-In in case of default Opt-Out mode. But we advice the subjects to submit email messages which are similar to email messages in the one's inbox. However, we cannot control the distribution of collected emails not to be different from the distribution of one's inbox.

The second function to protect the subject privacy is that we allow user to redact the sensitive keywords [2]. The subject may put any keyword to be anonymized in textbox of toolbar of Figure 2.1 and then the words in all the email that the subject decided to submit are converted to MD5 hash values when messages are submitted. To see the masking effect, user may click MASK button, then it showed masked email messages. It is useful if a subject has a concern releasing a certain person's name or an organization. However, most users did not use this functions.

The third feature is email encryption. The email client stores local copies of labeled emails not to loose the labeled emails before deleting email messages. The email client stores encrypted version of those labeled emails in user's hard disk drive until it actually submitted.

The fourth feature is delayed submission. Even though the subject rates email priority labels, we did not collect those messages immediately. We wait the user have time to consider whether it's fine or not and alway users may select Opt-Out button for that matters. Once the user click SUBMIT button, the collected messages were transfered to the server. However, to make sure we did not loose any information, we manually collected the stored encrypted messages with logs at the end of study and removed the email client Add-In program.

Also we provide STATUS button to show the status of message such as whether the message

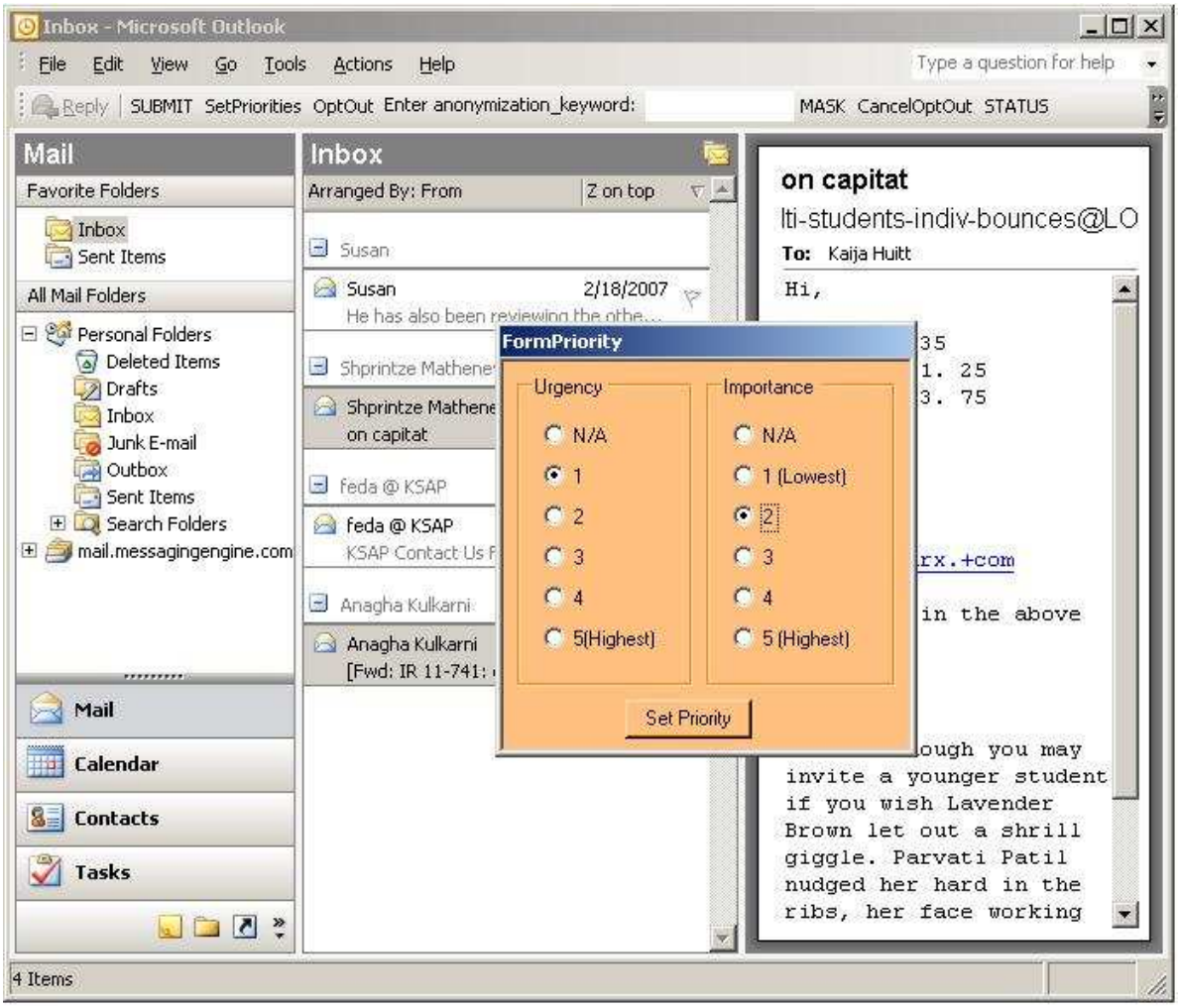


Figure 2.1: Outlook Add-In Snapshot of setting priorities in two selected email messages

was submitted or not, Opt-In / Opt-Out status, and current priority ratings. Such information automatically was displayed on STATUS button when the user select only one message. When the user clicks STATUS button, then it shows pop-up window for more detailed information with explanation.

Finally the collected information from the users is the email messages and user interaction feedback. The email message includes a header, subject, body text, attachment information, and folder information. The user feedback information is basically all user interaction events between users and email client program. Each event is time stamped with the event names. Based on these events, we may construct the reading orders, reading time, foldering, etc.

We recruited 25 experimental subjects mainly from the LTI department of Carnegie Mellon University. We recruited eight faculty member, five staff member, and twelve students. We asked the subject to label at least 400 non-spam emails during one month period and suggested labeling 800 non-spam emails (or equivalently labeling 40 emails per day). The importance and urgency level specified in 5 levels (importance levels – not important at all, not important, neutral, impor-

tant, and very important). During data collection, 15 subjects gave up to submit email data or labels due to personal reasons. Table 2.1 shows the summary statistics of finally collected emails with labels. Among them we tested seven users who actually submitted more than 200 importance labels for the first data collection.

2.2.2 The Second Data Collection

During the second data collection period, our highest concern is how to recruit more experimental subjects because we faced the extreme difficulty in recruiting additional experimental subjects. Therefore, we support Thunderbird email client program because some users want to use Thunderbird email client and we want to support Hotmail or Yahoo! Mail through Thunderbird Add-On programs.

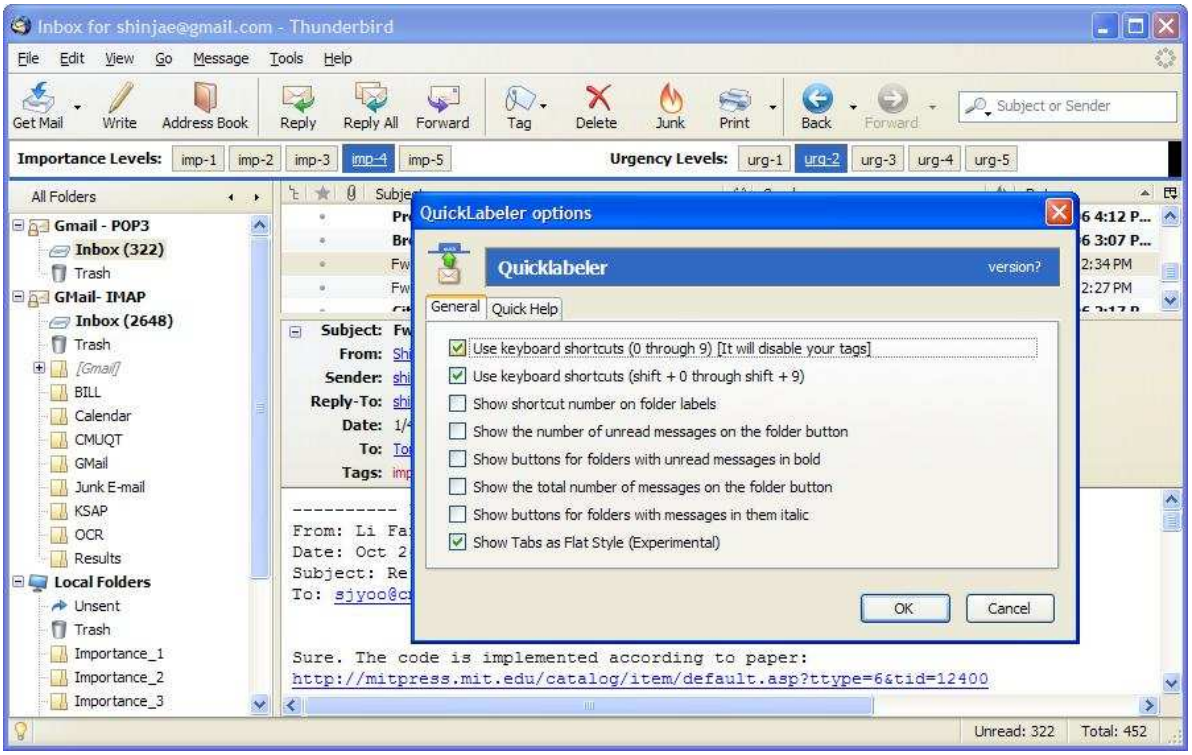


Figure 2.2: Thunderbird Add-On Snapshot of setting options. It also shows the importance level and urgency level setting tool bar

We removed some of features that were supported from Outlook such as redaction, email encryption and user feedback collection functionalities from Thunderbird client Add-On program. Redaction was not used because we observe that people do not submit emails if it contains sensitive keywords. Email encryption is also meaningless because Thunderbird stores emails in unencrypted format. Since we are not using any user feedbacks in our study, the function collecting user feedback was removed from Thunderbird email client program. Finally we also removed SUBMIT button as well because we noticed that we had to visit the subject machine anyway to uninstall our Add-On program.

However, we changed the design and added new functionalities. First, we changed the layout of setting priority from pop-up window to fixed button on toolbar as shown in Figure 2.2, which enable the users to easily set priority. Second, to further speed up labeling process, we supported keyboard short-cut based labeling. The subject can label email messages without using mouse, which improved labeling speed. Third, additional information on priority labeling button such as short-cut keys or the number of labeled messages were added due to the demand of the participants. The new design and functionality made the labeling process to be faster and collect more users.

We recruited a few experimental subjects from the LTI but mainly recruited subjects from the church, KCCP (Korean Central Church of Pittsburgh). Finally we collected emails from two pastors, six employees of institutions from Pittsburgh and Korea, two graduate students, one faculty and one undergraduate student who had a job. Table 2.1 shows the final collection statistics.

Collection	User	# of emails
First	1	1750
	2	503
	3	519
	4	989
	5	275
	6	279
	7	234
	*	153
	*	167
Second	8	408
	9	404
	10	899
	11	282
	12	863
	13	758
	14	476
	15	2989
	16	569
	17	816
	18	582
	19	1126
	Avg	658.8

Table 2.1: The number of collected Emails with labels

2.3 Evaluation Metric

To evaluate the performance of email prioritization, we consider several different metrics in terms of classification or regression point of views and discuss what would be better for email prioritization.

2.3.1 Classification Metrics

We may apply Recall, Precision, F-measure and Accuracy (or Error Rate) as the classification performance measures, which have been conventional in benchmark evaluations for text classification. Let A, B, C and D be, respectively, the number of true positives, false alarms, misses and true negatives for a specific priority level, and $N = A + B + C + D$ be the total number of test emails. We used four different metrics defined as:

$$Precision = A/(A + B) \tag{2.1}$$

$$Recall = A/(A + C) \tag{2.2}$$

$$F_\beta = \frac{(1 + \beta^2)A}{A + B + \beta^2(A + C)} \tag{2.3}$$

$$Accuracy = (A + D)/N \tag{2.4}$$

$$ErrorRate = (B + C)/N = 1 - Accuracy \tag{2.5}$$

Parameter β of F_β was set to 1.0 to balance Recall and Precision.

There are two conventional ways to compute the performance average over multiple users. One way is pooling the test instances from all users to obtain a joint test set, and computing the metrics on the pool. This way has been called micro-average. The other way is to compute the metrics on the test instances of each user and then take the average of the per-user metric values. This way has been called the macro-average. The former gives each instance an equal weight, and tends to be dominated by the system's performance on the data of users who have the largest test sets. The latter gives each user an equal weight instead. Both methods can be informative; therefore we present the evaluation results in both variants of the metric.

The advantage of classification metrics is that Precision, Recall, F1 and Accuracy are very intuitive and effectively measure the classification performance. However they ignore the ordinal priority relations. In other words, the error between priority level 1 and priority level 5 is the same as the error between priority level 1 and priority level 2, which is unfair.

2.3.2 Regression Metrics

The above disadvantage can be resolved by adopting regression metrics such as MAE (Mean Absolute Error) or MSE (Mean Square Error).

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{2.6}$$

or

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.7}$$

where y_i is true priority level and $\hat{y}_i$ is predicted priority level. If there are only two priority levels, then MAE and MSE is the same as Accuracy. Otherwise, MAE and MSE may distinguish different error levels. For instance, since we have five levels of importance, the MAE scores range from zero (the best possible) to four (the worst possible). MAE can be interpreted as the error distance on average but MSE is not.

Although MAE_β can tell the level of errors, it is a symmetric error metric. In other words, the prediction error to priority level 5 when the truth is 1 is the same to the prediction error to priority level 1 when truth is 5. The latter case [5(truth) to 1(predicted)] is more of a serious error than the former [1(truth) to 5(predicted)] because the later error misses a very import message and the former error just annoyed a user. For this reason, Sakkis et al. [38] used asymmetric metrics in spam filtering tasks. So we propose Asymmetric MAE (AMAE) to the extension of Weighted Accuracy.

$$AMAE_\alpha = \frac{1}{N}\sum_{i=1}^{N} c \cdot |y_i - \hat{y}_i| \text{ where } c = \begin{cases} 1 & \text{if } y_i > \hat{y}_i \\ \alpha & \text{otherwise} \end{cases} \tag{2.8}$$

where α is the relative directional cost to MAE. If α is 1, then $AMAE_1$ is reduced to MAE_β. Otherwise, it will give more or less penalty. If we replace N with $\sum_{i=1}^{N} c$ and there are only two levels, then $AMAE_\alpha$ is reduced to Weighted Accuracy of Sakkis et al. [38].

However, $AMAE$ can still perform unfairly because the error rate between 1 (not important at all) and that of 2 (not important) are treated as the same error rate between 3 (neutral) and 4 (important). The error rate between 3 and 4 should be more heavily penalized than the error between 1 and 2. Therefore we propose Weighted AMAE (WAMAE).

$$WAMAE_{\alpha,\beta} = \frac{1}{N}\sum_{i=1}^{N} c \cdot {y_i}^\beta \cdot |y_i - \hat{y}_i| \text{ where } c = \begin{cases} 1 & \text{if } y_i > \hat{y}_i \\ \alpha & \text{otherwise} \end{cases} \tag{2.9}$$

If β is 0, then $WAMAE_{\alpha,0}$ is reduced to $AMAE_\alpha$. But if β is not 0, then it differentiates the error according to y_i. For instance, if $\beta = 1$, $y_i = 5$ and $\hat{y}_i = 4$, then it will give 5 as error weight but if $\beta = 1$, $y_i = 1$ and $\hat{y}_i = 2$, then we give only 1 as an error weight. In summary, $WAMAE_{\alpha,\beta}$ gives more freedom to us to choose what a user wants but it is not clear how to choose α and β values. In case of α, Sakkis et al. [38] tried just 1, 9, and 99 for α but the choices of α and β should be further studied. Therefore, we only propose $AMAE$ and $WAMAE$ but we use $Accuracy$ and MAE as our main evaluation metric.

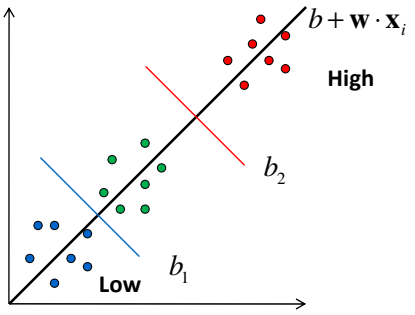


Figure 3.1: Three ordinal levels with a regression model and two separating thresholds

3 Priority Modeling

3.1 Motivation

Personalized email prioritization (PEP) is an ordinal regression problem [46], which is different from conventional text classification where for each category, there are only two levels, true or false. Users may rate their importance from one to five or from *not important at all* to *very important*, resulted in ordinal regression problem. Given limited amount of time, users may want to selectively read important emails or may associate actions to certain importance levels.

The personalized email prioritization entails two main research challenges: (1) the sparse training data and (2) one's own priority definition. First of all, unlike spam filtering, we could not share training data among different users because of privacy issues and different interests. People hesitate to share their very personal labeling information except spam emails. Even though there are users who are willing to share the very personal labeling information, the personal labeling information could not be shared. For instance, the importance of a grant proposal email could be extremely important to the principal investigator but it could be marginally important or not important to the person who is not actively working on the proposal.

Second, one's own priority definition could lead to diverse way of defining priority. In that case, the assumption of the current state-of-the-art ordinal regression such as Support Vector Ordinal Regression (SVOR) [12] might not be good enough. For instance, regression-based approaches assume one weight vector to model all levels of email priorities from the lowest priority level to the highest priority level, resulted in all decision boundaries to be parallel. Since the email text is very high dimensional space, it is not easy to visualize and check whether regression-based approach assumption will be held or not. Therefore, we have to do any form of empirical evaluation to conform what kinds of approaches are the best.

We present the first thorough study with both regression-based approach and classification-based approach (including our new approaches) addressing the PEP problem based on personal importance judgments of multiple users and further analyzing on ordinal regression benchmark dataset for general performance and synthetic dataset for controlled study. Our primary research question is: *How can we effectively learn robust user-specific models for accurate prediction of personalized importance using only small amount of labeled training data?*

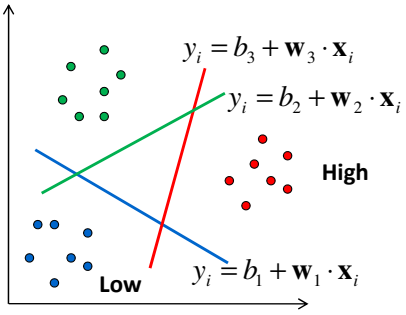


Figure 3.2: Three ordinal classes with three hyperplanes (OVA)

3.2 Regression-based Approaches

3.2.1 Pure Regression

The natural choice to handle ordinal response variables such as priority levels, survey answers or movie preference ratings is regression models. We may map r ordinal level response variable y_i to any certain real numbers, i.e. $y_i \in \{1, 2, \ldots, r\}$. We may apply standard regression such as linear regression [28] or support vector regression [19].

For instance, SVR (Support Vector Regression) optimizes the following conditions:

$$\min_{\mathbf{w},b,\xi,\xi^*} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{3.1}$$

subject to

$$\begin{aligned}(\mathbf{w}\cdot\mathbf{x_i} - b) - y_i \leq \varepsilon + \xi_i, \xi_i \geq 0, \forall i \\ (\mathbf{w}\cdot\mathbf{x_i} - b) - y_i \geq -\varepsilon - \xi_i^*, \xi_i^* \geq 0, \forall i\end{aligned} \tag{3.2}$$

where $\mathbf{w} \in R^d$ is a row weight vector and $\mathbf{x_i} \in R^d$ is a column vector for the input, ε is the margin for regression, ξ_i and ξ_i^* are slack variables, C is a regularization parameter and b is the intercept of a regression model. In case of prediction, we pick the closest level l from the predicted score of $\mathbf{w}\cdot\mathbf{x_i} - b$.

There are two important assumptions we need to address when we model ordinal regression problems by using pure regression model. The first assumption is that one weight vector $\mathbf{w}$ defines the whole ordinal relations among different levels from Equation 3.1. As shown in Figure 3.1, the decision hyperplanes are parallel to each other and orthogonal to the weight vector $\mathbf{w}$. We call it *one model assumption* because there is only one weight vector $\mathbf{w}$ compared to multiple weight vectors of classification-based approach. Since it is biased to have only one model or parallel decision hyperplanes, it is economical and it could be less sensitive to the noisy data than multiple models as shown in Figure 3.2 where we have three hyperplanes and they are not parallel. Since PEP (Personalized Email Prioritization) has to handle limited amount of training data, it would be attractive to have only one model to represent whole priority relations. However, if the assumption does not hold, the performance of regression model may not be guaranteed. In other words, the decision hyperplanes may not be parallel. In practice, PEP has to handle personalized priorities

and the user defined priority is not necessarily satisfying this assumption. If a priority is based on a task or topic, then it could be more close to classification than regression.

The second underlying assumption is that it assumes *the fixed equal distance* between adjacent ordinal levels. This assumption could be less critical than *one model assumption* but it is still affecting the accuracy of prediction because regression model predicts to the closest level. For instance, the difference between *important* and *very important* could be smaller than the difference between *neutral* and *important*.

3.2.2 Ordinal Regression

Rather than modeling ordinal regression problem through pure regression, we may explicitly model ordinal regression. Ordinal regression models drop the second assumption, *the fixed equal distance* between adjacent levels. Therefore, it provides multiple thresholds which tell us the predicted priority levels as shown in Figure 3.1, although it still learns one regression weight vector $\mathbf{w}$. These thresholds allow us to have different distances among different levels. For example, Support Vector Ordinal Regression (SVOR) [12] learns a model weight vector $\mathbf{w}$ and $r-1$ thresholds when we have r priority levels.

More specifically, SVOR optimizes the following conditions:

$$\min_{\mathbf{w},b,\xi,\xi^*} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{j=1}^{r-1}\sum_{i=1}^{n_j}\left(\xi_i^j + \xi_i^{*j}\right) \tag{3.3}$$

subject to

$$\begin{aligned} (\mathbf{w}\cdot\mathbf{x_i^j} - b_j) &\leq -1+\xi_i^j, \xi_i^j \geq 0, \forall i,j \\ (\mathbf{w}\cdot\mathbf{x_i^j} - b_{j-1}) &\geq 1-\xi_i^{*j}, \xi_i^{*j} \geq 0, \forall i,j \\ b_{j-1} &\leq b_j, \text{for} j = 2,\cdots,r-1. \end{aligned} \tag{3.4}$$

where n_j is the number of training emails which belong to priority level j, b_j is the threshold for j or lower level threshold, and $\mathbf{x_i^j}$ is j^{th} priority level email. The formulation of SVOR is quite similar to SVR but SVOR has $r-1$ thresholds, b_j, compared to only one intercept b of SVR.

3.3 Classification-based Models

3.3.1 Multi-class Classification

We can even drop *one model assumption* by treating ordinal regression problem as multi-class classification problems and thus we may have multiple models for each priority level. Multi-class classification provides the most flexible model but there are no relations among different priority levels. Although there are numerous ways to build multi-class classifiers from binary classifiers, we focus on three popular approaches: OVA (One vs. All), OVO (One vs. One), and DAGSVM [36].

One vs. All (OVA), also known as One vs. Rest (OVR), is the most common way to handle multi-class classification problem, Figure 3.2. OVA treats remaining classes as negatives and thus we need r models if we have r priority levels. When testing, we choose the most confident priority level as our prediction.

One vs. One (OVO), also known as all pairs, build all possible pairs of binary classifiers [26] such as (1 vs. 2), (1 vs. 3), ..., ($r-1$ vs. r). When testing, each classifier votes and the

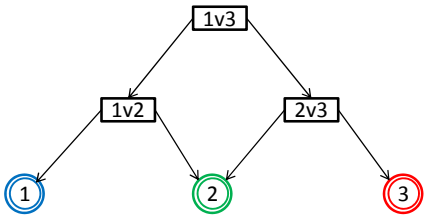


Figure 3.3: Decision DAG (Directed Acyclic Graph) for One vs. One multi-class classification. The rectangular represents a OVO classifier and the double circle shows the final decision. When testing a decision node, take the left child if the left-hand class is more probable than the right-hand class.

majority class will be the predicted class. Although One vs. One (OVO) classification requires $r \cdot (r-1)/2$ classifiers, each classifier has less amount of training examples than OVA classifiers and thus overall training time is reduced [26].

Instead of majority voting, we may use decision DAG (Directed Acyclic Graph) during testing as shown in Figure 3.3. We call it DAG instead of DAGSVM [36] because we may apply it to different classifiers too instead of SVM. DAG is faster than OVO during prediction because it requires only $r-1$ test. Although Plattet et al. [36] reported the order of classes from DAG did not affect final results, we sorted the order of priority levels as shown in Figure 3.3.

3.3.2 Order Based DAG

Although regression model makes use of priority relations, their models are not flexible due to *one model assumption*. It could be critical for personalized email prioritization because each person might have different assumption about the priority levels. Multi-class classification provides flexibility because they allow multiple models among the different priority levels. However, they ignore the ordinal relations among the priority levels. Therefore, we propose models which have both the flexibility of multi-class classification models and the ordinal relations of regression model.

Rather than directly predicting each priority level, we may use the order information for guiding better specific cases. Figure 3.4 shows the decision directed acyclic graph (DAG) for Order-Based (OB) classification models. When there are multiple paths available from top nodes to leaf nodes, any path may guide to the correct decision as long as each node's decision is correct. Since there are multiple choices available, we can always choose the **most confident** decision node among candidate decision nodes, OB-MC or we may do **majority voting**, OB-MV. For instance, when we have three priority levels, we can start from both "12 vs 3" and "1 vs 23" of Figure 3.4. For a testing email x_i, suppose that an SVM classifier trained "12" as positive and "3" as negative training classes (12 vs 3) and the classifier predicted 0.7 but SVM trained with "1" as positive and "23" as negative training labels (1 vs 23) and predicted -0.9. In case of OB-MC, we follow "1 vs 23" decision path because -0.9 is more confident than 0.7 and the next decision node is "2v3" instead of "1" due to the negative prediction score. OB-MV test all possible paths and then majority voting will determine which one is our final decision. If there are even votes, we may test even votes results using one vs remaining even vote node classification. For instance, "12 vs 3"

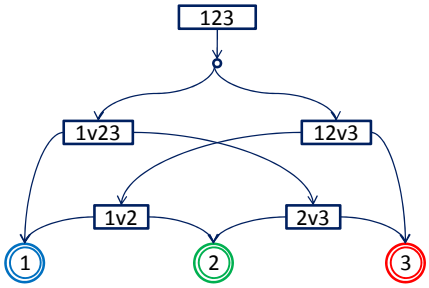


Figure 3.4: Decision DAG (Directed Acyclic Graph) for three level Order-Based (OB) classification. The rectangular represents a OB classifier and the double circle shows the final decision. When testing a decision node, take the left child if the lefthand class is more probable than the righthand class.

predicted "1" for final decision but "1 vs 23" ended up with "3". Then we choose the better one out of "1 vs 3".

Through Order-Based approaches, we have multiple flexible models as classification-based models but we also have model bias to the order of priority levels as regression-based model, resulted in robust modeling to the noisy data. If the priority levels have no relations (perfect for classification) or satisfy ordinal regression assumption (perfect for regression), our proposed order-based approach may not be able to outperform than two approaches. However, if users have set any form of partial ordinal relations, then our proposed models have a potential to improve the prediction accuracy.

When we apply r level prioritizer, the total number of basic classifier is $\sum_{k=1}^{r}(r-k+1)\cdot(k-1)$. The classification models listed above can be paired with any kinds of classification algorithm and we tested SVMs and Regularized Logistic Regression depending on dataset.

3.4 Experiments and Analysis

We evaluated regression-based approach and classification-based approach on three different dataset.

3.4.1 Personalized Email Prioritization

Dataset and Preprocessing We used the dataset described in Section 2.2. Table 3.1 shows the training and testing split statistics of finally collected emails. We split the first 150 email messages as training and the rest as testing based on the timestamp of email messages. If we did not reserve the first 150 email messages as training, then we could build prioritization models from future data and it would not be realistic.

We preprocessed email messages by tokenization but we did not remove stop words or apply stemming. The basic features were the tokens in the sections of from, to, and cc address, title, and body text of email messages.

Classifiers and Parameter Tuning For classification-based approaches, we used linear SVM classifiers as our base classifiers. Each classifier took the vector representation of each message

User	# of emails	# of train	# of test
1	1750	150	1600
2	503	150	353
3	519	150	469
4	989	150	839
5	275	150	125
6	279	150	129
7	234	150	84
8	408	150	258
9	404	150	254
10	899	150	749
11	282	150	132
12	863	150	713
13	758	150	608
14	476	150	326
15	2989	150	2839
16	569	150	419
17	816	150	666
18	582	150	432
19	1126	150	1076
Avg	658.8	150	555.62

Table 3.1: Training and testing split of collected emails for prioritization model experiments

as its input, and produced a score with respect to a specific importance level. In case of OVA, the importance level with the highest score is taken as the predicted importance level by our system for the corresponding input message. We used the SVM^{light} software package and tuned the margin parameter C in SVM which controls the balance between training-set errors and model complexity. We split the training set of each user into 10 subsets and repeated a 10-fold cross validation procedure: using one subset for validation and the union of the remaining subsets for training the SVM with a specific value of C. We repeated this procedure on 10 validation subsets, with the C values in the range from 10^{-3} to 10^{3}. The value of each parameter which yielded the best average performance on the 10 validation sets was selected for evaluation on the test set of each user. We found the system's performance relatively stable (with small variance) with the settings of $C \in [1, 1000]$.

Regressiors For regression-based approach, we tested only SVOR with implicit constraints [12] with linear kernel. We tested explicit constraints SVOR and other non-linear kernels but they showed worse results than implicit constraints SVOR with linear kernels in terms of MAE. Again we tuned only regularization parameter with the same ranges of SVM classifiers.

Estimation and Baseline Since we want to show improvement on limited amount of training data through learning curves, we randomly shuffled 150 training examples ten times and choose every 30 training email increments from 30 emails to 150 emails. Our baseline is predicting to

always priority level 3 out of 5 levels, which is the most common priority level on our data collection.

Significance Testing We also conducted four types of significance test, pairwise t-tests for macro level MAE and Accuracy, Wilcoxon signed-rank test for micro level MAE, proportional test (p-test) for micro level Accuracy to assess the statistical significance of performance difference among baseline, SVORs and SVMs.

In case of pairwise t-test, we calculated per-user performance difference in terms of MAE and Accuracy between two approaches and used the mean of the per-user differences to estimate the p-value under the null hypothesis (which assumes the zero mean). This test is most popular and strong test but it requires normality assumption of score distribution.

For Wilcoxon signed-rank test, we calculated the difference in the absolute error of between two approaches on each test message, and throw away no difference instances. We computed the ranks of absolute values of two score difference. Then we multiply the sign of two score difference to the rank, called signed rank. The test statistics is the minimum of the sum of positive ranks and the sum of negative ranks, which is used to estimate the p-value under the alternative hypothesis (which assumes one is better than the other). Wilcoxon signed-rank test is non-parametric test, resulting that it does not require normality assumption. Our micro-level MAE is ordinal outcome and we could not assume the normality assumption.

Last, p-test (proportional test) [44], also known as proportional z-test, was conducted for micro level accuracy test because Accuracy is proportional metric. We can calculated z score, based on two proportional metric scores under the alternative hypothesis (which assumes one is better than the other). It is naturally micro-level test along with Wilcoxon signed-rank test.

Results and Analysis First of all, surprisingly, the state-of-the-art regression-based approach, SVOR, showed significantly worse performance than the performance of classification based approach, OB-MV, shown in Figure 3.5 and 3.6 and Table 3.2. The performance gap is not only significant regardless of evaluation metric but also it is statistically significant regardless of the types of significance test. It is evident that SVOR performance among machine learning models suggested that *one model assumption* did not hold on personalized email prioritization.

Second, we could validate the machine learning approaches significantly improve over baseline. In other words, we could make use of machine learning approach to improve the prediction performance of personal importance.

Third, among the classification methods, the evaluation results show that there are not much distinctions among classification based methods on Figure A.5. However, OVA showed the worst performance except 30 trainings and others did notably better. Also our proposed order based approaches, especially OB-MV, showed the overall best performances in terms of MAE among the classification approaches and the difference was statistically significant. We conjecture that order based approaches could take advantages of the partial order relations. Between DAG and OVO, DAG showed statistically significantly better but it was on limited ranges.

Suppose that we might have very limited amount of training data (less than 30 messages) and we might not be sure about *one model assumption*, we might use OVA. However, we may want to try order-based DAGs when we have more emails available. If we have to choose it from popular classification-based approaches, then DAGs are good choice given enough amount

of training email messages.

3.4.2 Benchmark Experiments

Dataset and Experimental Setups Our next research question was whether our proposed order-based approaches would work well or not on benchmark dataset. Therefore, we tested order-based approaches along with other approaches on ordinal regression benchmark dataset generated from UCI dataset [11] [1]. [11] used two collections of dataset but we tested only one of them because the size of the other collection was too small to test different training set size. The dataset was normalized to be zero mean and unit variance for each feature. The response variable was split into 10 ordinal levels using equal-size binning. Note that this procedure will satisfy *one model assumption* but does not guarantee *fixed equal distance assumption*. In other words, they are good for ordinal regression approach but not for pure regression approach such as linear regression or support vector regression. We randomly selected training data from 25 instances to 300 instances by 25 increments and then tested on the remaining. The training and testing splits were repeated 100 times independently. Table 3.3 summarizes datasets and their statistics.

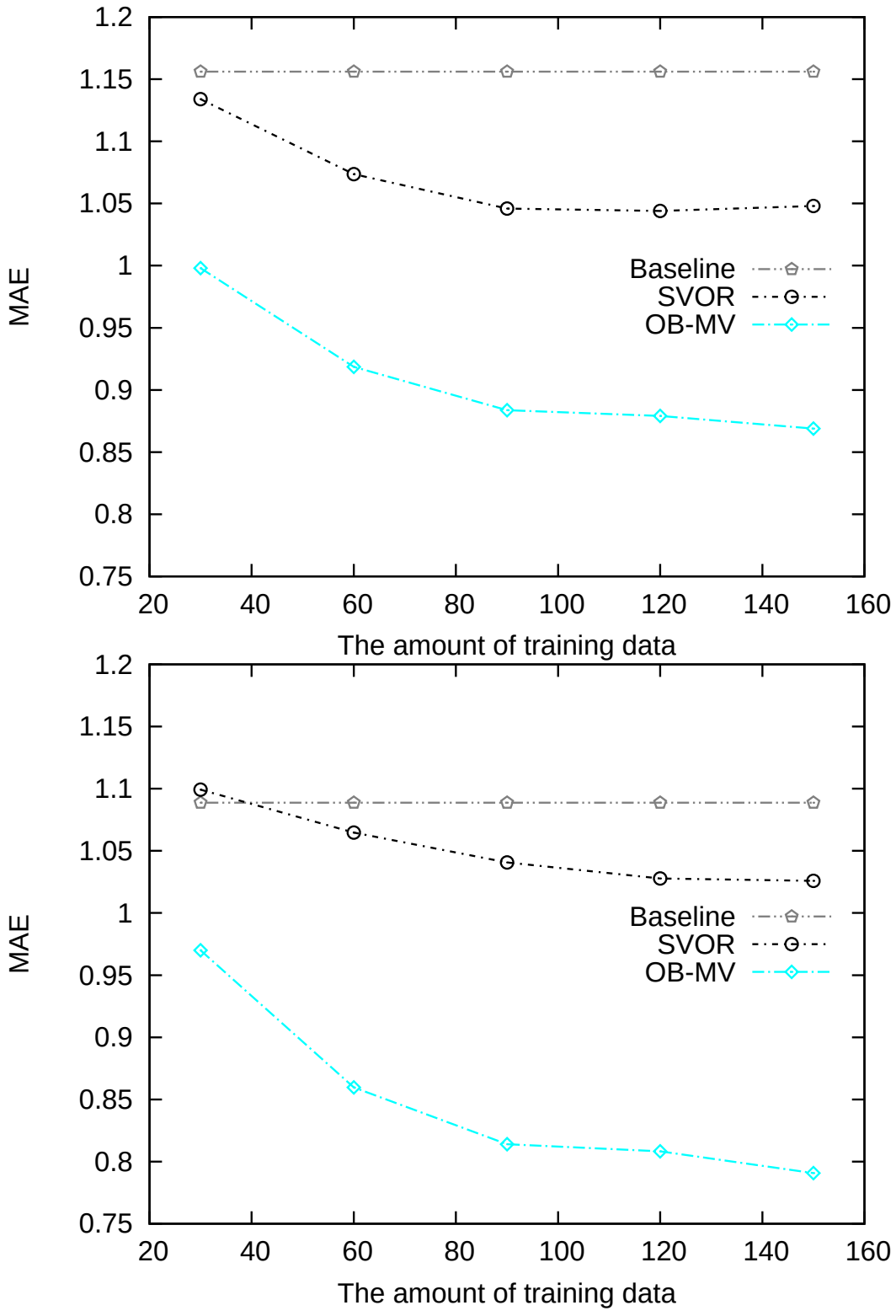
For classification-based approaches, we could not use SVM classifiers as our base classifiers due to the slow speed of SVM classifiers and thus we used Regularized Logistic Regression [45] due to its convergence properties and comparable accuracies. We got similar performance with regularized logistic regression performance compared to SVM classifier on this benchmark dataset and [28] reported both of them showed similar performance. We tuned regularization parameter λ from 10^{-8} to 10^{-1}. We applied the same SVOR settings as in personalized email prioritization.

Results and Analysis On the contrary to personalized email prioritization dataset, we got quite different results from UCI benchmark dataset, shown in Figure 3.7 and each dataset results in Figure A.7. First of all, SVOR showed the best performance regardless of training size and dataset and OVA showed the worst performance in most cases. As personalized email prioritization dataset, DAG is better than OVO in four out of seven dataset, Bank Domains (1), Bank Domains (2), Census Domains (1), and California Housing dataset and showed similar performances on the rest of dataset. Order-Based DAGs showed better performance than DAG on Bank Domains (1), Bank Domains (2), and California Housing but the improvement is limited to the limited training size. With the limited amount of training data, order information was more helpful but with enough training data, DAG performance is similar to OB-DAG. The main difference between personalized email prioritization dataset and UCI dataset is whether the dataset satisfies *one model assumption* or not.

3.4.3 Principle Component Analysis

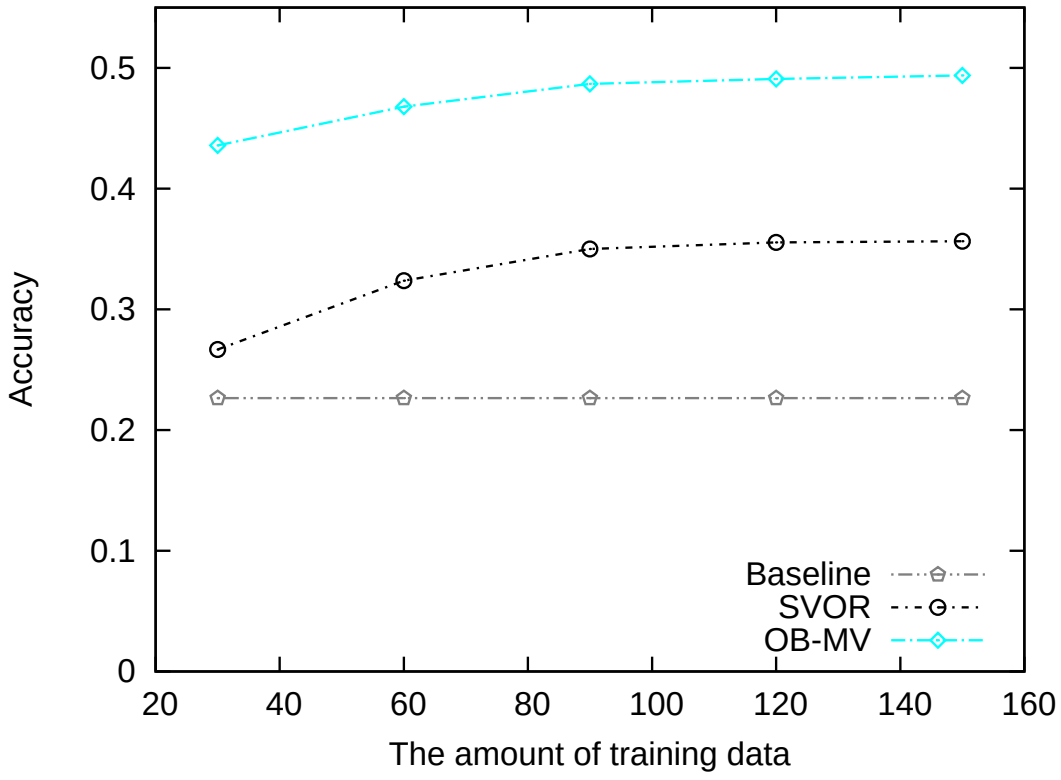
However, it was not clear why SVOR outperformed on certain datasets but it did not outperform on the other dataset. To answer this question, we applied Principal Component Analysis (PCA), which is one of most popular dimensionality reduction approach. We projected Email Prioritization and UCI dataset onto two most correlated reduced dimensions with the ordinal response variable by using Pearson Correlation Coefficients. Note that, this projection should be the best projection for

[1] http://www.gatsby.ucl.ac.uk/ chuwei/ordinalregression.html

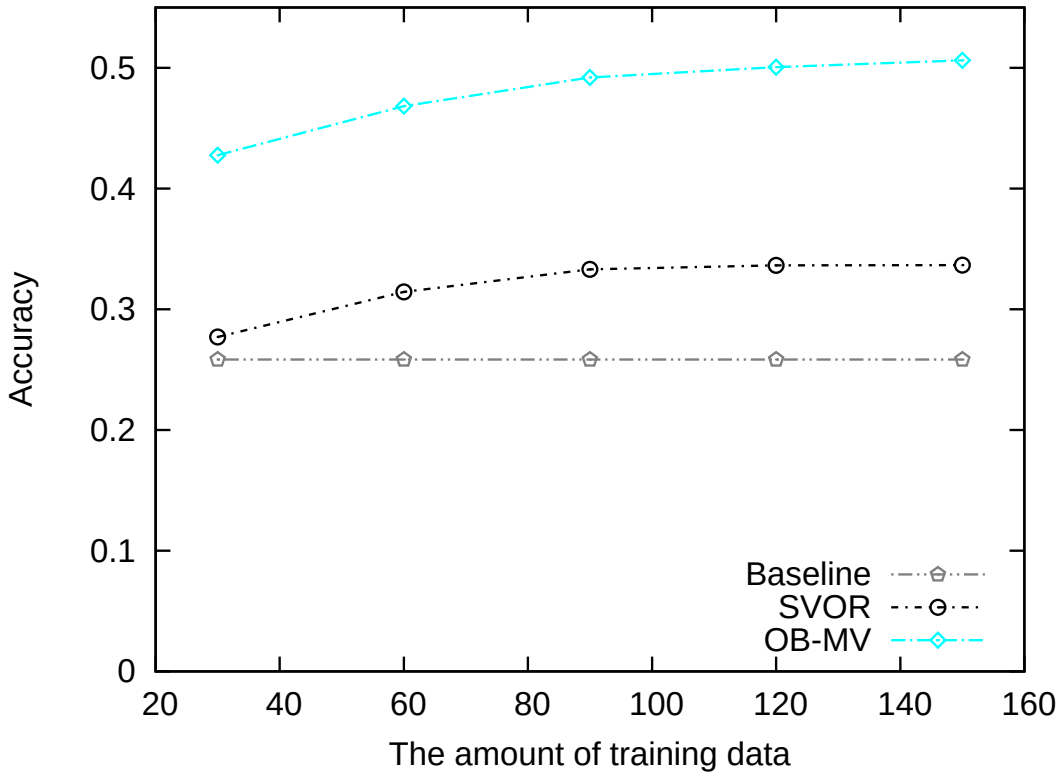


(a) Micro Average

Figure 3.5: Macro and Micro Average MAE Learning Curves with Baseline, SVOR and OB-MV



(a) Macro Average



(b) Micro Average

Figure 3.6: Macro and Micro Average Accuracy Learning Curves with Baseline, SVOR and OB-MV

# of tr	Baseline(b)	SVOR(o)		OB-MV		
	MAE	MAE	p-value(b)	MAE	p-value(b)	p-value(o)
30	1.1560	1.1340	0.3576	**0.9980**	* 0.0148	* 0.0288
60	1.1560	1.0736	0.1362	**0.9185**	* 0.0010	* 0.0197
90	1.1560	1.0459	0.0844	**0.8837**	* 0.0004	* 0.0189
120	1.1560	1.0441	0.0746	**0.8791**	* 0.0003	* 0.0141
150	1.1560	1.0480	0.0902	**0.8689**	* 0.0002	* 0.0143

(a) Macro MAE Results

# of tr	Baseline(b)	SVOR(o)		OB-MV		
	MAE	MAE	p-value(b)	MAE	p-value(b)	p-value(o)
30	1.0887	1.0992	* 0.0000	**0.9700**	* 0.0000	* 0.0000
60	1.0887	1.0647	* 0.0000	**0.8597**	* 0.0000	* 0.0000
90	1.0887	1.0406	* 0.0000	**0.8140**	* 0.0000	* 0.0000
120	1.0887	1.0278	* 0.0000	**0.8083**	* 0.0000	* 0.0000
150	1.0887	1.0259	* 0.0000	**0.7907**	* 0.0000	* 0.0164

(b) Micro MAE Results

# of tr	Baseline(b)	SVOR(o)		OB-MV		
	ACC	ACC	p-value(b)	ACC	p-value(b)	p-value(o)
30	0.2265	0.2668	* 0.0210	**0.4358**	* 0.0000	* 0.0000
60	0.2265	0.3237	* 0.0039	**0.4679**	* 0.0000	* 0.0000
90	0.2265	0.3499	* 0.0020	**0.4868**	* 0.0000	* 0.0002
120	0.2265	0.3554	* 0.0018	**0.4908**	* 0.0000	* 0.0006
150	0.2265	0.3565	* 0.0024	**0.4938**	* 0.0000	* 0.0010

(c) Macro Accuracy Results

# of tr	Baseline(b)	SVOR(o)		OB-MV		
	ACC	ACC	p-value(b)	ACC	p-value(b)	p-value(o)
30	0.2584	0.2771	* 0.0000	**0.4276**	* 0.0000	* 0.0000
60	0.2584	0.3144	* 0.0000	**0.4682**	* 0.0000	* 0.0000
90	0.2584	0.3330	* 0.0000	**0.4919**	* 0.0000	* 0.0000
120	0.2584	0.3365	* 0.0000	**0.5006**	* 0.0000	* 0.0000
150	0.2584	0.3365	* 0.0000	**0.5061**	* 0.0000	* 0.0000

(d) Micro Accuracy Results

Table 3.2: Evaluation results of varying training set size. It shows MAE with p-value (macro: paired t-test, micro: signed rank test) and Accuracy (macro: paired t-test, micro: proportional test), indicating the statistical significances of better performance compared to the baseline(b) or SVOR(o). Numbers in bold font indicating the best approach for each fixed training-set size. The star indicates the p-values equal or less than 5%.

Data Sets	Features	Instances
Bank Domains(1)	8	8192
Bank Domains(2)	32	8192
Computer Activities(1)	12	8192
Computer Activities(2)	21	8192
California Housing	8	15640
Census Domains(1)	8	16784
Census Domains(2)	16	16784

Table 3.3: UCI Ordinal Regression Benchmark Dataset Statistics

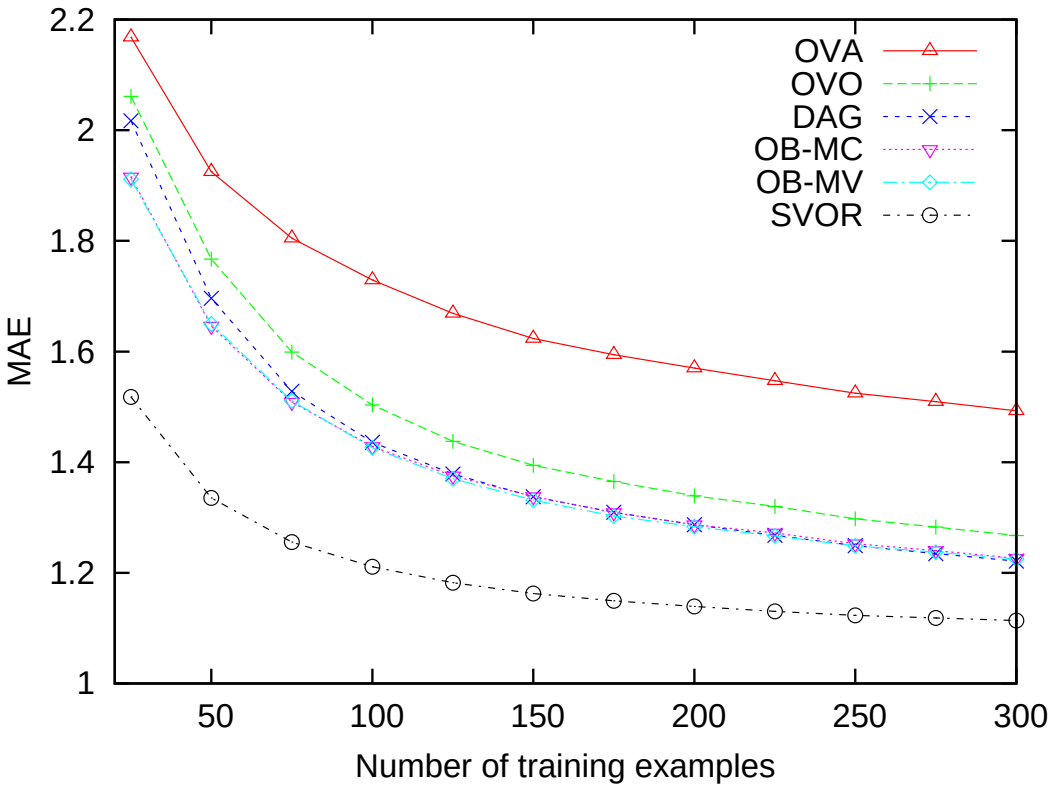
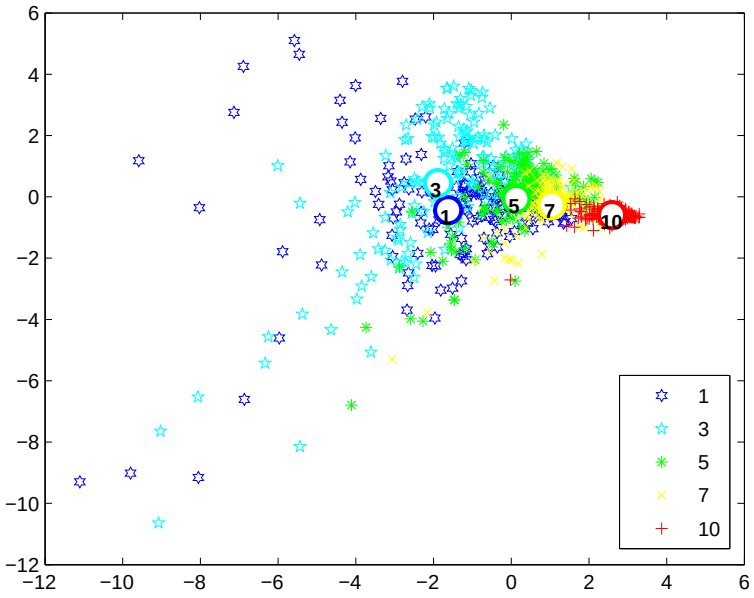
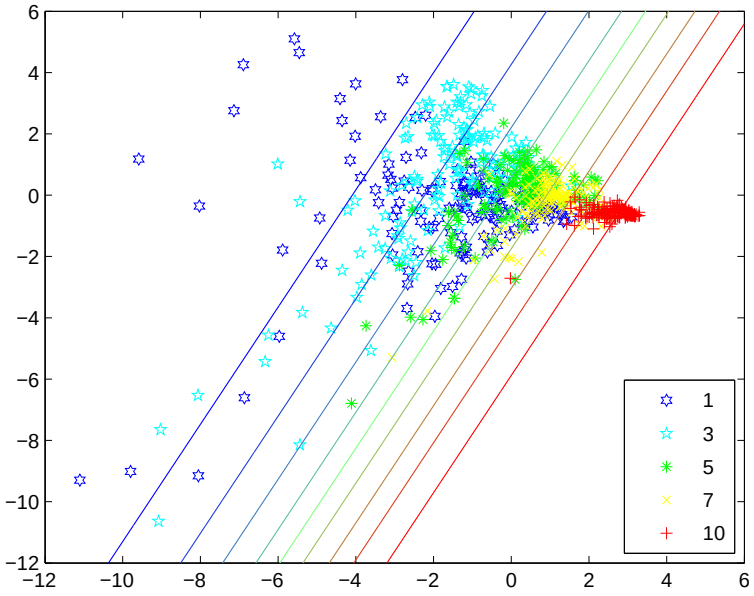


Figure 3.7: UCI 7 Dataset Average MAE Results

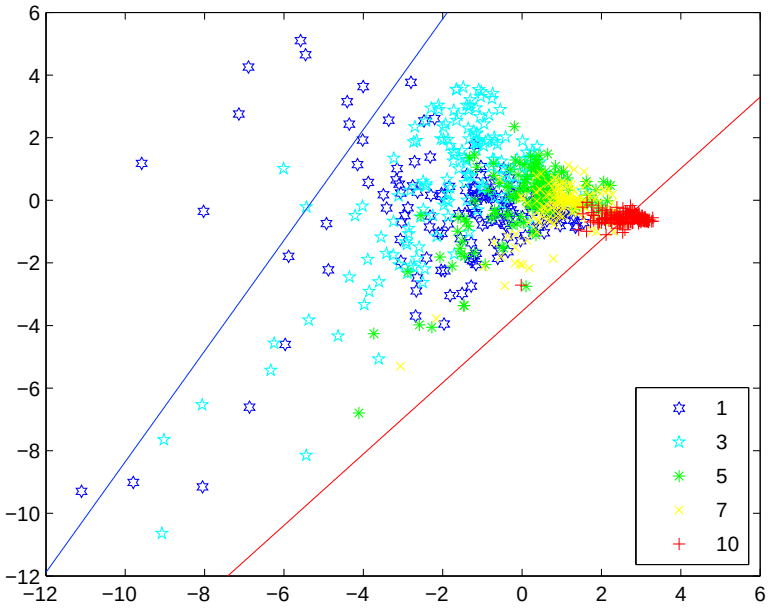


(a) PCA projection with Centroids

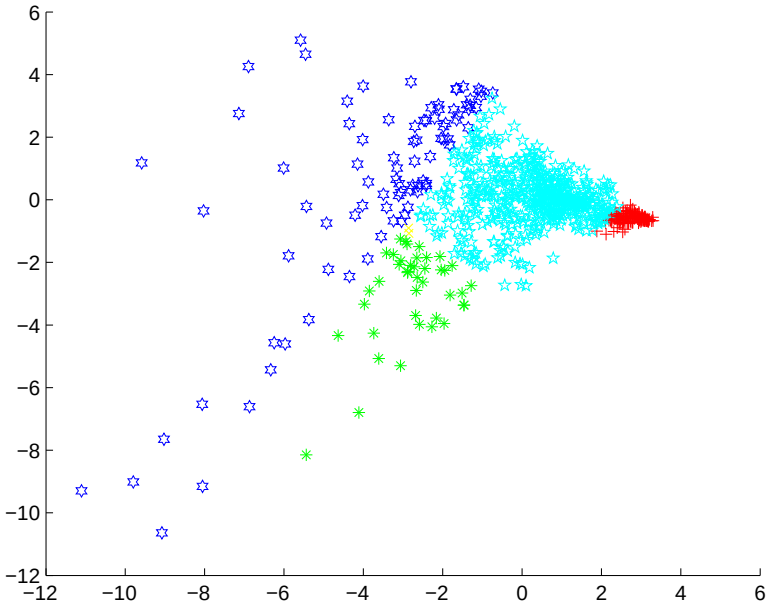


(b) PCA projection with Ordinal Regression Decision Hyperplanes

Figure 3.8: Computer Activities (2) on two the most correlated reduced dimensions with the response levels. The drawn lines are threshold for each ordinal levels and the fixed equal distance assumption do not hold here. Ordinal regression thresholds well captured different levels except level 1.

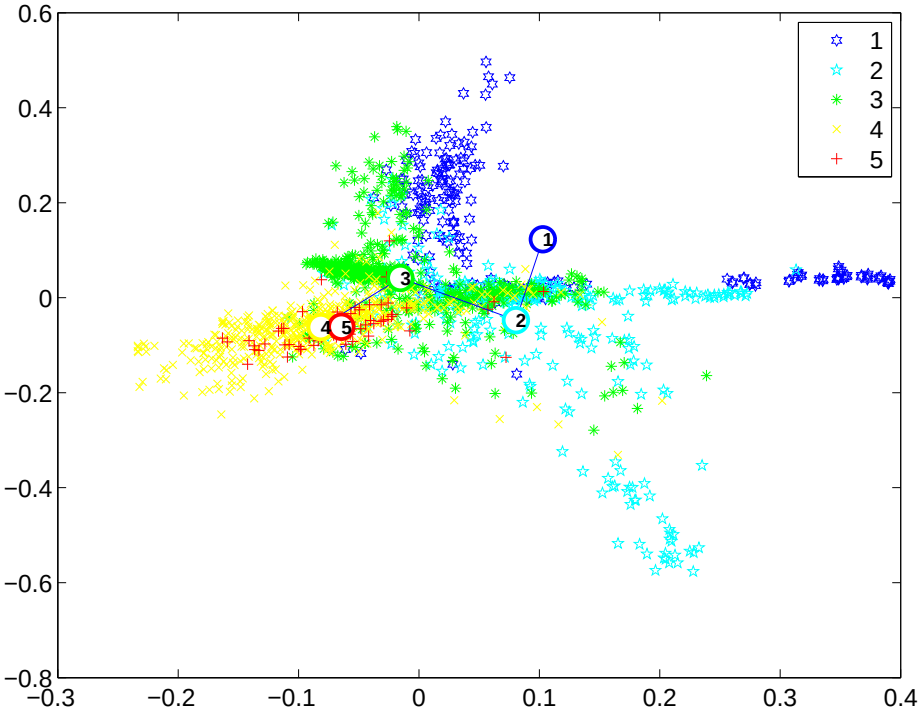


(a) PCA projection with Classification Decision Hyperplanes

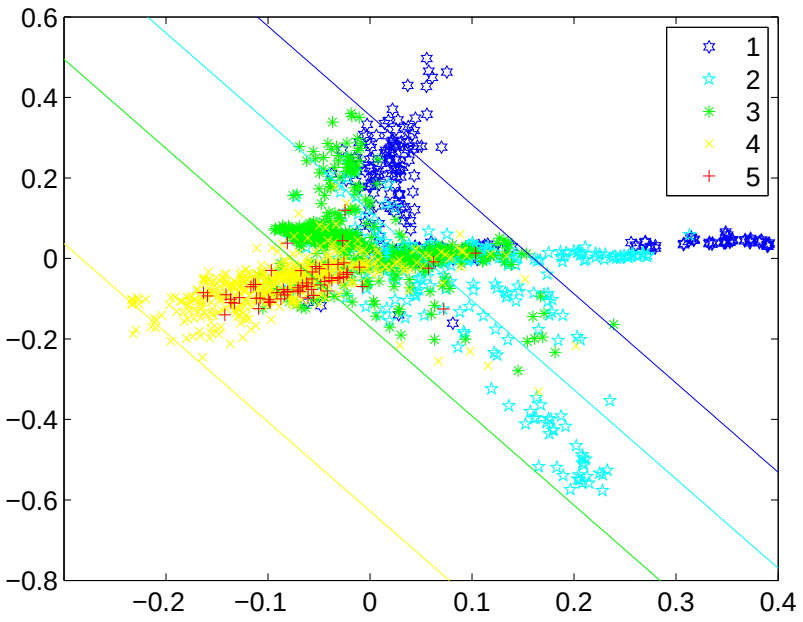


(b) PCA projection of predicted labels

Figure 3.9: Computer Activities (2) on two the most correlated reduced dimensions with the response levels. The drawn lines are threshold for each classification decision hyperplanes and some of hyperplanes are not shown here because the remaining hyperplanes are too high or low.

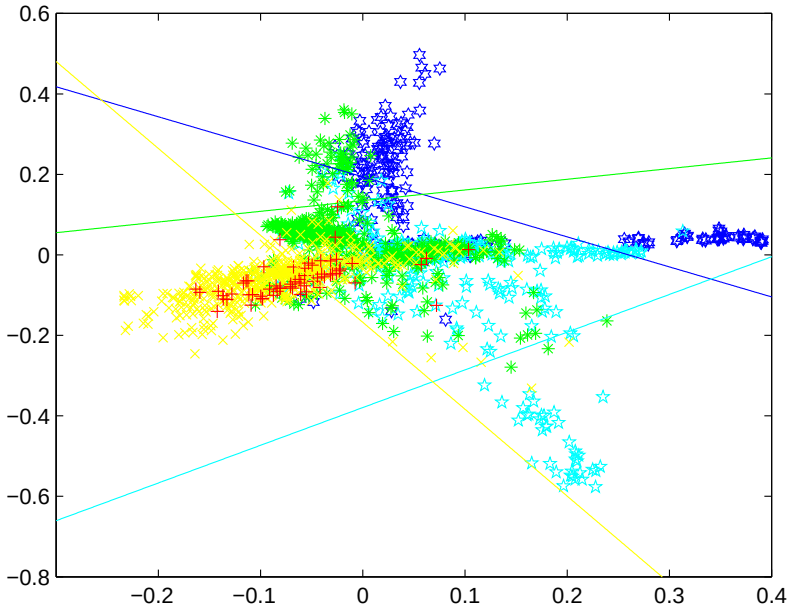


(a) PCA projection with Classification Decision Hyperplanes

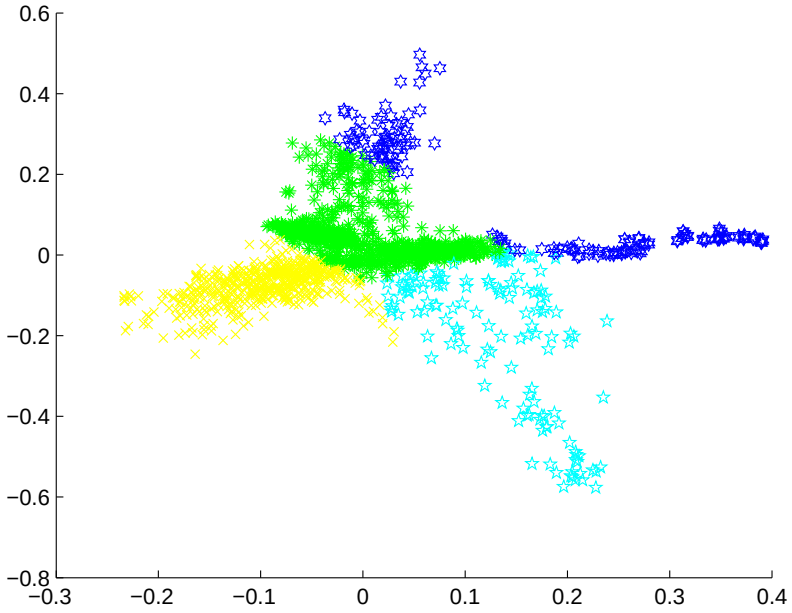


(b) PCA projection with Ordinal Regression Decision Hyperplanes

Figure 3.10: One user of email prioritization dataset was projected on two most correlated reduced direction with the response levels. The drawn lines are threshold for each ordinal levels. Ordinal regression thresholds captured different levels to some degree but it was not as good as CPU Activity (2).



(a) PCA projection with Classification Decision Hyperplanes



(b) PCA projection with Ordinal Regression Decision Hyperplanes

Figure 3.11: One user of email prioritization dataset was projected on two most correlated reduced direction with the response levels. The drawn lines are threshold for each classification decision hyperplanes. Classification did show better accuracy than the accuracy of regression approach on the plotted data.

regression based approach. We also learned OVA and SVOR models for benchmark dataset from the projected two dimensional dataset and drew decision hyperplanes from Figure 3.8 $\sim$ 3.11.

Among seven ordinal regression benchmark datasets, we focus on Computer Activities (2) datasets because the datasets well characterized ordinal regression conditions and with the same reason we chose one user from email prioritization dataset. We observe the data distribution looks quite different. First, the centroids of Computer Activities (2) on Figure 8(a) were well aligned as a linear line according to the ordinal levels (except level 1), resulted in good alignment with SVOR decision hyperplanes compared to email prioritization dataset where the centroids are not well aligned to the line, so that we have better distribution for classification hyperplanes.

In summary, this analysis tells us whether the dataset follows *one model assumption* or not. Computer Activities (2) follows *one model assumption* pretty well, so that regression-based approach outperformed classification based approaches. However email prioritization dataset seemed not well fitted with *one model assumption*, resulted in better classification performance.

Note that we projected data onto two most correlated directions and thus there were other dimensions which were better suited for classification approaches. Also we could observe that there were partial ordinal relations from email prioritization dataset, which confirmed why our proposed order-based approaches worked better than other classification approaches.

3.4.4 Synthetic Experiments

Dataset and Experimental Setups Although we reflected the correlations to the response variable on PCA, our two dimensional analysis may not be perfect. Through our synthetic analysis experiments, we could confirm that what we discovered is still valid on the controlled study.

We generated two dimensional Gaussian data distribution with the centroids on (1,1), (2,2), (3,3), (4,4) and (5,5) as shown in Figure 12(a). Note that it satisfies *one model assumption* and *fixed equal distance assumption*. To control the linearity of the centroid distribution, we shifted centroids from (2,2) to (0,4), from (4,4) to (2,6) and from (3,3) to (5,1), shown in Figure 12(b). We repeated the above procedures 100 times independently and reported the average results along with t-test. We apply the same evaluation strategy of UCI ordinal regression benchmark dataset to this synthetic dataset.

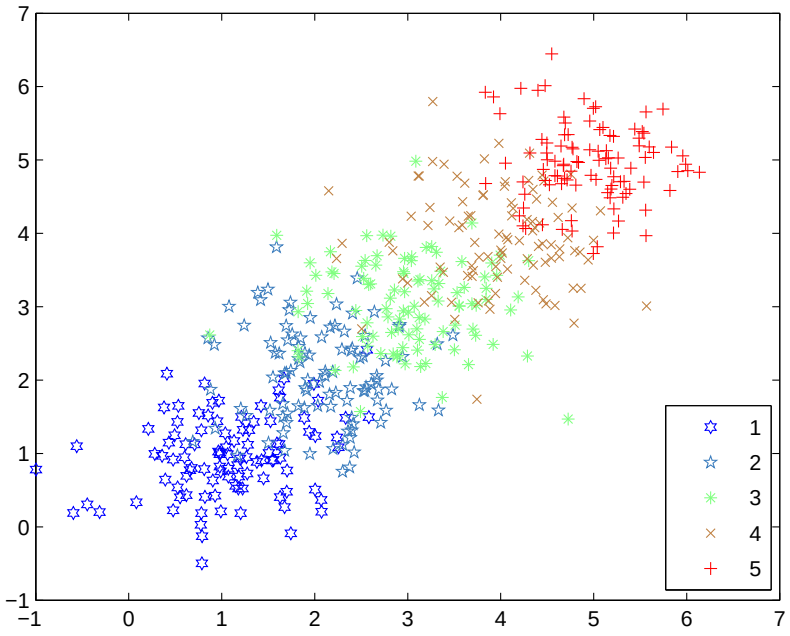
Results and Analysis First of all, with linearly aligned centroids, SVOR did not show the better performance. However, SVOR showed better performance than OVA approaches. All classification approaches except OVA they showed better performance than SVOR. But with more difficult cases (high signal-to-noise ratio), we could observe SVOR showed better results than any other classification based approaches.

When the centroids are not linearly aligned, classification based approaches showed significantly better results than SVOR. Therefore, to be the best condition for SVOR, noisy and linearly aligned centroids are required, which is favorable for *one model assumption*.

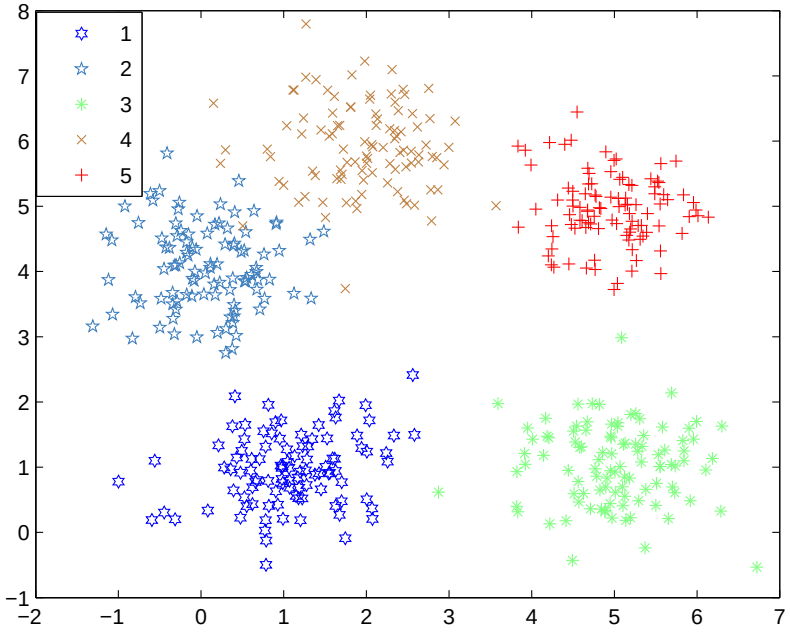
3.5 Summary

Personalized email prioritization requires effective mapping from a high-dimensional input feature space to ordinal output variables. We presented a comparative study of two types of supervised

learning approaches: ordinal regression-based and classification-based. Our conceptual analyses and empirical evaluations show that the effectiveness of ordinal-regression based method crucially depends on the separability of priority classes by parallel hyperplanes, which may be too restrictive for personalized email prioritization based on our collected personalized email prioritization dataset. Classification-based methods, on the other hand, offer more general and robust solutions when complex decision boundaries are needed because they allow multiple non-parallel hyperplanes as decision functions. With the proposed OB-MV and OB-MC schemes, we effectively combine the outputs of different binary classifiers into email priority predictions, yielding significant improvements over the results of SVOR, a state-of-the-art method among ordinal-regression based approach on our collected personalized email prioritization dataset. Our experiments with synthetic datasets and ordinal-regression benchmark datasets further support our conclusions, and provide additional insights regarding when regression-based method work better and when classification-based methods work better.

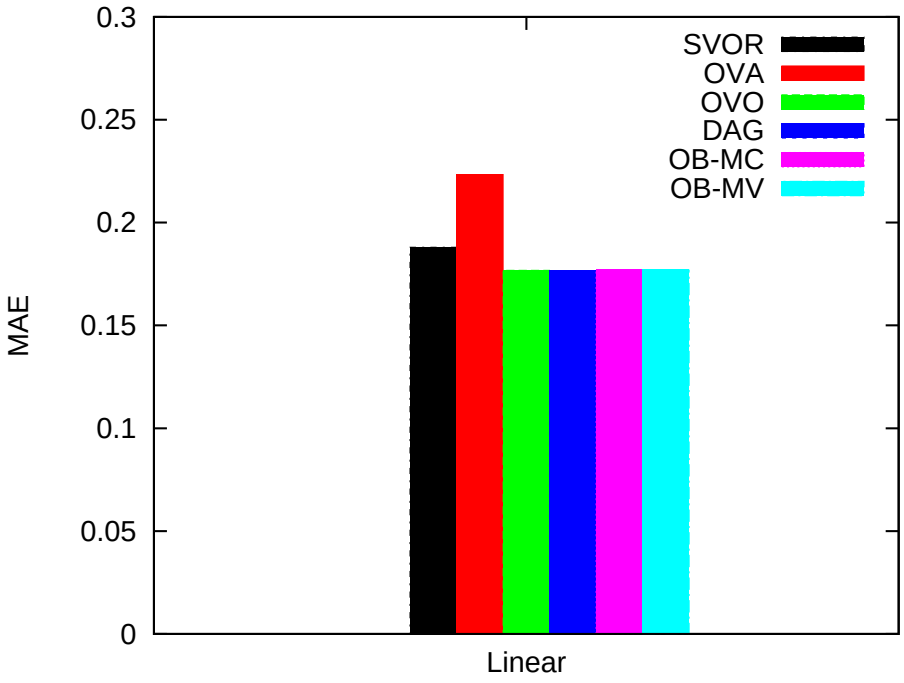


(a) Linearly Aligned Centroids on $y = x$

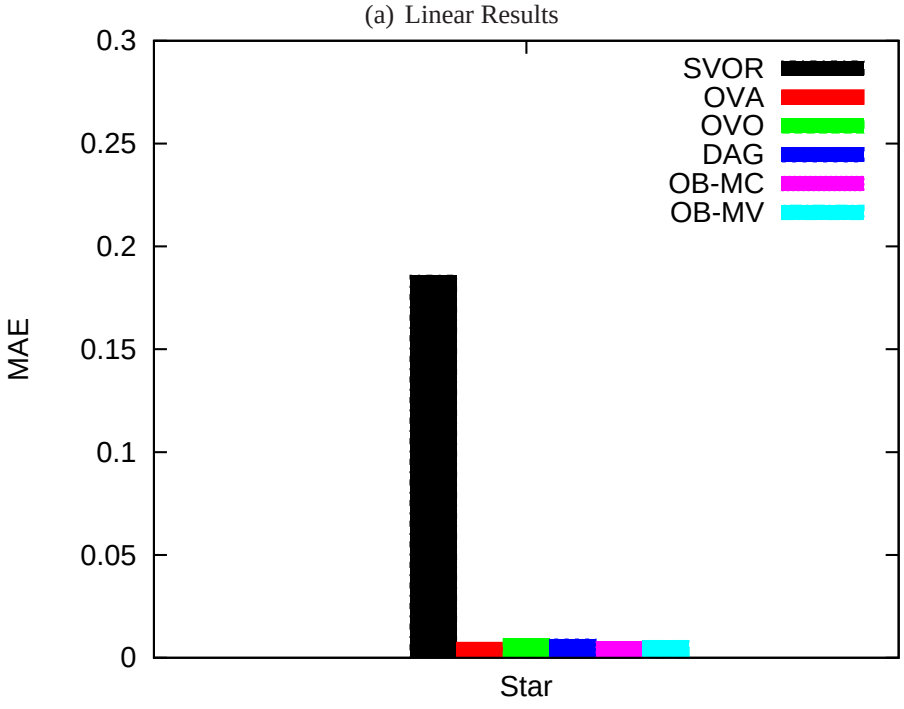


(b) Star-shaped Centroids

Figure 3.12: Two synthetic data generation conditions (Linear and Star)



(a) Linear Results



(b) Star Results

Figure 3.13: Experiment results of two synthetic data conditions

4 Learning from Social Network and User Interactions

Due to privacy and personalization, we do not have publicly available email data and enough labels to investigate. However, in email inbox, there are lots of unlabeled email data that has no privacy concerns and also there is meta information of email headers that can be extracted. This chapter investigates how we can improve email priority learning curves with the limited amount of labels. Especially we focus on the social networks induced from email communication network and meta information of messages.

4.1 Social Clustering

For predicting the importance of email messages, the sender information would be highly informative. For example, we may have multiple project teams or social activity groups, and membership in such social groups may be naturally reflected through co-recipient lists of email messages. The group members who share similar sender/recipient patterns may have similar judgments on priority levels of messages. Thus, capturing such groups would be informative for predicting the importance of contact persons (senders or recipients) of email messages.

When we have a limited amount of training data, it is very likely that in the testing phase we encounter a sender who does not have any labeled instances in the training set. If we can identify this user as a member of a group based on unsupervised clustering, then we can infer that user's importance from that of other group members. That is, we can cluster users based on their communication patterns in a personal social network, and infer the importance of users in each group. Further, the cluster membership of the sender of each email message can be treated as features (in addition to a standard bag-of-word representation) of the message when making inference about its importance. As a result, senders without labeled messages could also receive non-zero weight through their clusters, effectively addressing the data sparsity problem.

We first discuss how to construct a social network from a user's personal email INBOX and how to extract the group information.

4.1.1 Personalized Social Networks

We construct a *personalized* social network for each particular user using only the email data of that user. There are two reasons for this: **Practicality**-we want our method to not rely on the unrealistic assumption that multi-user private data are always available for system development and model optimization. **Personalization**-we want the social network best representing the user's own social activity; a global social network may include noisy features and de-emphasize personalization in the inductive learning of important features through the network.

Let us use a graph $G = (V, E)$ to represent the email contact network where vertices V correspond to the email contacts (users) in the network, and edges E correspond to the messages sending events among users. The edges are binary, i.e., $E_{ij} = 1$ if there is (at least) a message from user i to user j, and $E_{ij} = 0$ otherwise. We ignore the direction of edges if it is not explicitly mentioned. By default, a graph G is un-weighted symmetric graph.

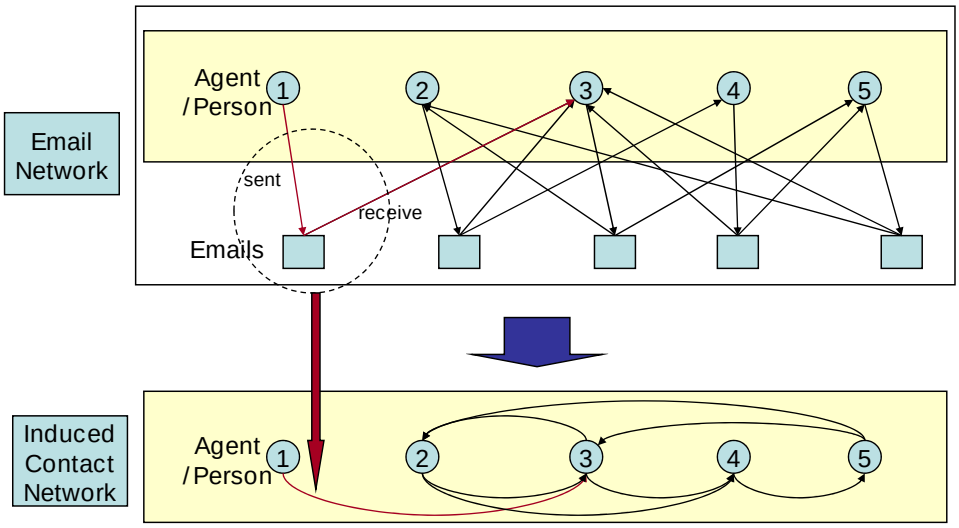


Figure 4.1: An example email contact network induced from email messages. Circles represent nodes in the network. An edge between a node i and a node j implies i sent email to j.

4.1.2 Social Clustering Algorithms

To select an appropriate clustering algorithm, our main criterion is an algorithm that finds social clusters that represent real world social groups. We choose Newman, CONCOR (CONvergence of iterated CORrelations), K-means and Spectral clustering algorithms [18] on contact networks.

Newman Clustering We choose the Newman clustering algorithm, which has been reported to successfully find social structures in large organizations [35, 39]. It defines the *edge-betweenness* as a normalized number of shortest paths going through a specific link from all-pairs shortest paths. If a link has a high edge-betweenness score, it means that the link is crucial between two boundary nodes of two different highly-connected clusters. The algorithm assumes that members in a highly-connected cluster have many communication passages within the cluster, but not many links outside the cluster. Based on this assumption, it deletes links with high edge-betweenness scores, which results in disconnect components as clusters.

To find more than two clusters, we need to specify the number of clusters that the network may have embedded. For this, users may use either their own knowledge about the network or they can use an automatic selection algorithm, described in [35]. This automatic selection algorithm is implemented in Organization Risk Analyzer (ORA) [10], and that is the implementation we use in this work. Figure 4.2 shows embedded clusters inn a network where ORA selects 27 as the number of clusters.

CONCOR Clustering CONCOR [41] is known for finding a structural equivalence in a social network and has been one of the earliest approaches. CONCOR hinges on a procedure based on the convergence of iterated correlations. Basically it repeatedly calculates Pearson Correlation Coefficients (PCC) between rows (or columns) of a matrix where the matrix has the Pearson Correlation

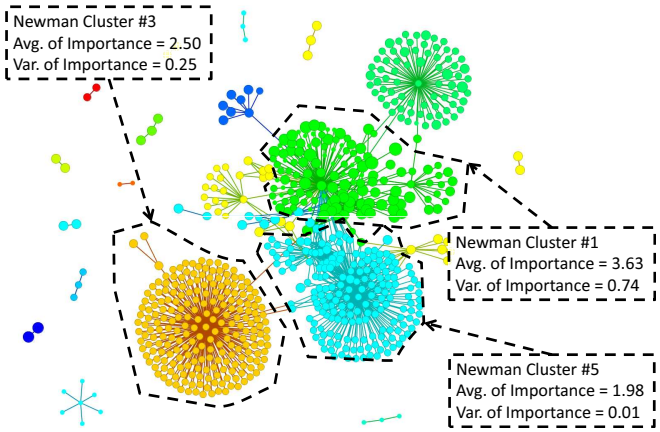


Figure 4.2: The analyzed user's contact network from email exchanges, node colors represent the Newman cluster affiliation of email contacts, node sizes are adjusted to the average importance of the contacts' email importance values. The average importance values of contacts within specific clusters are similar, which means that members in a cohesive cluster shares similar importance. As an example, we add average and variance of importance only from big three clusters only

Coefficient matrix of previous iteration.

$$X_{ij}^{t+1} = PCC(X_i^t, X_j^t) \tag{4.1}$$

where $X_{ij}^0 = E_{ij}$, X^0 is an adjacency matrix, X_i^t is the i^{th} row (or column) after the t^{th} iteration. When $t = 0$, X is an adjacency matrix but if the iteration t continues until it converges, $X_{ij} \in \{-1, 1\}$. This procedure finds only two clusters of '-1' and '1'.

To find more than two clusters, we need to repeatedly apply CONCOR to sub-clusters and it should formulate binary tree structures. We regard the number of clusters as parameter k of kNN algorithm. We determine the best number of clusters through cross validation.

K-Means K-Means clustering algorithm is one of the most popular clustering algorithm due to its simplicity. Since we will run K-Means on adjacency matrix, X, it will find structurally similar persons.

K-Means algorithm tries to minimize the following objective function [18].

$$\sum_{i=1}^{K} \sum_{x_j \in C_i} (x_j - \mu_i)^2 \tag{4.2}$$

where C_i is the i^{th} cluster and μ_i is the centroid of the i^{th} cluster. In other words, it tries to minimize intra-cluster variance in the inner summation and find the sum of each cluster variance (inter-cluster variance) to be small in the outer summation. To solve Equation 4.2, the following greedy iterative procedure can be used.

1. Randomly select K seed nodes as centroids.

2. Assign each node to the closest centroid.

3. Recompute the centroids.

4. Repeat the second and third steps until it converges.

We use Euclidean distance as our distance metric. Since the above procedures will converge to the local optimum, we repeat the above procedures 100 times and select the best cluster assignments based on Equation 4.2. We again consider the number of cluster as our parameter and use the best number K determined by the cross validation.

Spectral Clustering Along with K-Means, spectral clustering algorithm is also widely used in various domains [40]. We first define graph Laplacian matrix L:

$$L = D - X \tag{4.3}$$

where D is diagonal matrix and it contains the sum of its row elements, $D_{i,i} = \sum_{j=1}^{n} X_{ij}$. One of interesting properties is that if G has k connected components, then the first k eigenvalues are 0 and the first k eigenvectors will be indicator for each connected components [40].

To find k clusters, the normalized spectral clustering algorithms compute the first k eigenvectors, $Lx = \lambda Dx$ and then apply K-Means clustering algorithm on those k eigenvectors.

For K-Means, we use Euclidean distance but only 10 times to find the best K-Means cluster assignments according to Equation 4.2. We also consider the number of clusters as our parameter and used the best number K determined by the cross validation.

4.2 Measuring Social Importance

4.2.1 Motivation

We want to measure the social importance levels of contacts, and this can be done without labeled training data. Instead, the personal contact network induced from senders and recipients link relations provides useful information about the importance of each contact in the network. For instance, the Newman Cluster #1 in Figure 4.2 is highly connected with others and the person in the center of the cluster may be an important person in the network. We examine multiple graph-based metrics to characterize the social importance of each node, which have been commonly used in social network analysis (SNA) or link structure analysis.

4.2.2 Node Degree Metrics

In-degree centrality We define $InDegreeCent(i)$ as the normalized measure for the in-degree of each contact (i):

$$InDegreeCent(i) = \frac{1}{|V|} \sum_{j=1}^{|V|} E_{ji} \tag{4.4}$$

where $|V|$ is the total number of contacts in the personal email social network and $E_{ji} \in \{0, 1\}$. A high in-degree may indicate that the recipient is a popular person.

Out-degree centrality We define $OutDegreeCent(i)$ as the normalized measure for the out-degree of each contact (i). Having a high out-degree may also imply some degree of importance, e.g., as an announcement sender or a mailing-list organizer.

$$OutDegreeCent(i) = \frac{1}{|V|} \sum_{j=1}^{|V|} E_{ij} \tag{4.5}$$

Total-degree centrality $TotalDegreeCent(i)$is defined as the normalized number of unique senders and recipients who had email communication with node i. That is, it is a simple or operation of the in-degree and out-degree of the node:

$$TotalDegreeCent(i) = \frac{1}{|V|} \sum_{j=1}^{|V|} \lceil \frac{E_{ij} + E_{ji}}{2} \rceil \tag{4.6}$$

4.2.3 Neighborhood Metrics

Clustering Coefficient Clustering Coefficient of node v, denoted as $ClustCoef(v)$, measures the connectivity among the neighborhood of the node.

$$ClustCoef(v) = \frac{1}{Z} \sum_{i \in Nbr(v)} \sum_{j \in Nbr(v), j \neq i} E_{ij} \tag{4.7}$$

where $Nbr(v) = \{x : E_{v,x} \neq 0, E_{x,v} \neq 0\}$ is the neighborhood and $Z = |Nbr(v)| \cdot (|Nbr(v)| - 1)$ is the normalization denominator. Boykin and Roychowdhury [7] used this metric to discriminate spam from non-spam email messages based on the neighborhood connectivity of the recipients of messages.

Clique Count A clique is generally defined as a fully connected sub-graph in an undirected graph. The clique count of a node v in our case is defined as:

$$ClqCnt(v) = \sum_{c \in G} I(c \in v) \times I(|c| \geq 3) \tag{4.8}$$

where $c \in G$ is a clique c in the personalized social network G, $I(c \in v) \in 0, 1$ is the binary indicator of whether or not clique c contains node v, and $I(|c| \geq 3) \in 0, 1$ is a binary indicator of whether or not the size of clique c is at least three. This metric reflects the centrality of the node in its local neighborhood, taking all the related non-trivial cliques (including the nested ones) into account.

4.2.4 Global Metrics

Betweenness centrality Betweenness centrality of a node v, $BetCent(v)$, is the percentage of existing shortest paths out of all possible paths that goes through the node v. A node with high

betweenness centrality means that the corresponding person is a contact point between different social groups.

$$BetCent(i) = \frac{1}{(n-1)(n-2)} \sum_{j=1, j\neq i}^{|V|} \sum_{k=1, k\neq j, k\neq i}^{|V|} \frac{\sigma_{jk}(i)}{\sigma_{jk}} \tag{4.9}$$

where σ_{jk} is the number of shortest paths contain j and k and $\sigma_{jk}(i)$ is the number of shortest paths contain j and k that goes through i. This metric has been used in social network analysis [35].

PageRank We use the popular PageRank method in link analysis research [8] to induce a global importance measure for email contacts. The difference between the PageRank importance from the other metrics discussed so far is that it is recursively defined, taking the transitivity of popularity into account. Let us use matrix X to represent email connections among N contacts in a personal network, and define the elements as:

$$X_{ij} = \frac{n_{ij}}{\sum_{j'=1}^{N} n_{ij'}} \tag{4.10}$$

where n_{ij} is the count of messages from i to j. Matrix X is further combined with a teleportation matrix U defined as:

$$E = ((1-\alpha)X + \alpha U)^T \tag{4.11}$$

$$\text{where } U = \left[\frac{1}{N}\right]_{N\times N}, \quad \text{and} \quad 0 \leq \alpha \leq 1$$

Using an N-dimensional vector $\vec{r}$ to store the PageRank scores of the N contacts, the vector is initially set with equally valued elements of $1/N$, and then iteratively updated as:

$$\vec{r}^{(k+1)} = E\vec{r}^{(k)} \tag{4.12}$$

The vector converges to the principal eigenvector of matrix E when k is sufficiently large.

4.2.5 Social Importance Analysis

We call the above metrics the *Social Importance* (SI) features of email messages. To illustrate that the SI features would be informative for a personalized email prioritization system, we computed the PCC (Pearson Correlation Coefficient, which ranges from -1 to +1). Figure 4.3 shows the absolute values of the correlation coefficient scores: larger absolute values mean stronger dependencies among the SI features and the importance levels. It can be observed that the multi-metric PCC values differ from user to user, which is not surprising. For user 1, as an example, Clustering Coefficient, Clique Count and HITS Hub scores are highly informative, but In-degree, Out-degree and Total-degree are less informative. In contrast, for User 5, HITS Authority score is not a good indicator but in-degree is highly informative. This observation suggest it is important for the system to learn user-specific SI feature weights. We accomplish this goal by training user-specific SVM classifiers. This is, we train five SVMs for each user based on his or her personal email dataset; each SVM is responsible for learning the weights of features (including SI features and other types of features) conditioned on a specific importance level and for the specific user. Our system does not use the PCC's because they do not take the interactions among features into account and hence

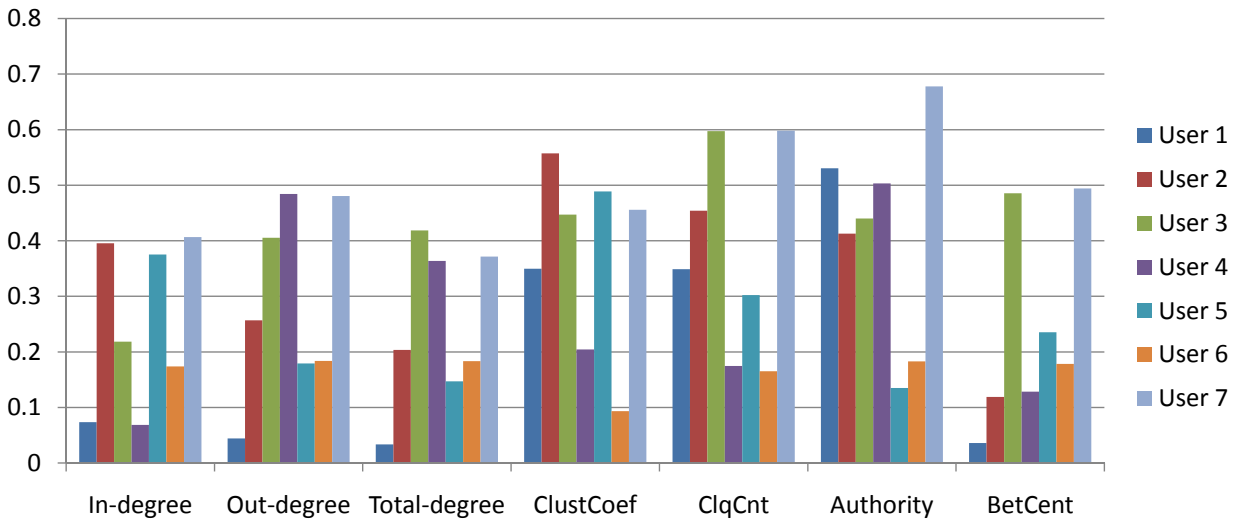


Figure 4.3: The Pearson Correlation Scores (vertical axis) of social importance metrics (horizontal axis) for different users

would be suboptimal compared to SVM-learned weights of SI features. We show the PCC sores in Figure 4.3 just for illustrative purposes: they intuitively indicate dependencies among SI features and importance levels.

4.3 Semi-Supervised Measure of Social Importance

4.3.1 Motivation

The social importance features are all induced from personal social networks without leveraging human-assigned importance labels of email messages. Therefore, we call them unsupervised SI features. Now we focus on how to induce semi-supervised SI features. Here semi-supervised means that the features are induced from personal email data where only a subset of the messages have human-assigned importance labels (in 5 levels), and the rest of messages do not have such labels. We propose a new approach, namely the Level-Sensitive PageRank (LSPR) approach which can be viewed as a new important variant of existing personalized PageRank or topic sensitive PageRank methods [24].

4.3.2 LSPR Algorithm

First, we use a matrix to encode the information about how human-assigned importance labels of messages are related to the users in a personal email collection. The rows of the matrix are the users (i = 1,2,$\cdots$,N), the columns are the importance levels (k =1,2,3,4,5), and each cell is the count of labeled messages received by a user at the corresponding level. We further normalize the elements of each column using the sum of all elements in the column as the denominator, i.e., making the normalized elements in each column sum to one. Let us denote the matrix (N-by-5) as $V = \vec{v_1}, \vec{v_2}, \cdots, \vec{v_5}$ where the column vectors show the distributions of labeled messages over all users at each level, and the row vector $\vec{v_i} = (v_{i1}, v_{i2}, v_{i3}, v_{i4}, v_{i5})$ can be viewed as the initial LSPR

profile of user i based on the labeled messages he or she received. Notice that v_{jk} = 0 if user i does not have any labeled message in the personal email collection. Generally speaking, matrix V is very sparse when only a few messages are labeled.

Next, we construct a different transition matrix for each importance level as:

$$E_k = (1-\alpha)X + \alpha U_k \tag{4.13}$$

Maxtrix X is the same as we defined in Chapter 4.2.4 whose cells are the estimated transition probabilities from each node (email contact) based on unlabeled email interactions. In the second term we have $U_k = \vec{v}_k \cdot \vec{1}^T$, which depends on the labeled data at level k and differs from the teleportation matrix in standard PageRank. The balance between the two transition matrixes is controlled using constant mixture weight $\alpha \in [0,1]$. Matrix E_k is used to calculate the Level-Sensitive PageRank (LSPR) vector iteratively as:

$$\begin{aligned} \vec{p_k}^{(t+1)} &= E_k \vec{p_k}^{(t)} \\ &= (1-\alpha)X\vec{p_k}^{(t)} + \alpha U_k \vec{p_k}^{(t)} \\ &= (1-\alpha)X\vec{p_k}^{(t)} + \alpha \vec{p_k}^{(1)} \end{aligned} \tag{4.14}$$

where $U_k\vec{p_k}^{(t)} = \vec{p_k}^{(1)}\vec{1}^T\vec{p_k}^{(t)} = \vec{p_k}^{(1)}$ and $\vec{p_k}^{(1)} = \vec{v_k}$ is the initial vector. The LSPR vector converges when t is sufficiently large, to the principal eigenvector of matrix E_k. The stationary LSPR vector is denoted as $\vec{p_k}$, whose elements sum to one, representing the expected proportion for each node to receive the importance values from others through a biased transition network, i.e., the messages at the same level (k) make their receivers more connected.

Applying this calculation to each importance level, we obtain five stationary vectors in matrix $P = (\vec{p_1}, \vec{p_2}, \vec{p_3}, \vec{p_4}, \vec{p_5})$. The row vectors of matrix P provide a 5-dimensional representation for each user based on both partially available message labels, and the level-sensitive transition networks. The row vectors of P are much denser than the initial user profiles, i.e., the row vectors in matrix V. We use the LSPR row vectors as additional features in an enriched representation of each message, i.e., as the semi-supervised social importance features of its sender. Those enriched vector representations are used both in the training phrase of our system (Support Vector Machines), and in the testing phase as the input vector of each new message for the system to make a prediction.

Notice that the elements in matrix P are typically small when the number of users (N) in the personal email network is large. To make the values of LSPR features in a range comparable with those of other features (e.g., term weights and the values of unsupervised SI features) in the enriched vector representation of email messages, we renormalize each LSPR sub-vector (5-dimensional) into a unit vector as follows:

$$p_{ki} = \frac{p_{ki} + s}{\sum_{j=1}^{5} p_{kj} + 5 \cdot s} \tag{4.15}$$

where s is smoothing constant for normalization. We added smoothing constant here because we do not want to give too much weight for p_{ki} when p_{ki} is too small value. These vectors provide 5 additional features (with the corresponding weights) in the enriched representation of the contact person of each email message, in the input vector for importance prediction using a SVM.

4.3.3 Connections between SIP and Topic Sensitive PageRank

Our formulae for LSPR are quite similar to those in Topic Sensitive PageRank (TSPR) and Personalized PageRank (PPR) methods where a topic distribution is used to represent the interest of each user [24]. In fact the LSPR method is intrigued by the TSPR and PPR work. The main differences in our problem and the LSPR solution are:

- Our graph structure is constructed using two types of objects (i.e., persons and messages) while the graph structures in TSPR and PPR (and in PageRank) has nodes of only one type (i.e., web pages). And, our method leverages both frequencies of messages and importance of messages while there is only one type of linkage (directed) in conventional link analysis methods.

- We focus on effective use of a partially labeled personal network, and we assume the transitivity of importance among users is sensitive to the importance levels of messages exchanged among these users. The assumption is conceptually different from conventional use of topics or user profiles in TSPR and PPR methods. This is the fundamental difference between LSPR from TSPR and PPR. Specifically, the stationary solution in TSPR and PPR (and standard PageRank) is the vector of the expected probabilities of web pages being visited by users in random browsing based on hyperlink connections; on the other hand, the stationary solution in LSPR is the vector of importance scores of email messages assuming their importance levels are transitive with respect to each other through the interactions in a personal email network.

Other than the above, our formulae are indeed quite similar to those in TSPR, PPR and PageRank. The convergence analyses for those methods and the formulae of the closed-form solution (i.e., the principal eigenvector) of the transition matrix also apply here; we omit those details (see [24][8]).

4.4 Meta Features

On top of email text and social network information, there is meta information of email message such as message size, the existence of attachment files, assigned folder, etc. They can be correlated with different priority levels. Table 4.1 summarizes considered meta-level features.

4.5 Incorporating Additional Features into Prioritization Models

In case of extended feature vector space, each email's extended feature vector is $\mathbf{e_i^{st}} = \langle t_1, t_2, \cdots, t_k, s_1, s_2, \cdots, s_m \rangle$ where $\mathbf{e_i^t} = \langle t_1, t_2, \cdots, t_k \rangle$ are textual feature vector and $\mathbf{e_i^s} = \langle s_1, s_2, \cdots, s_m \rangle$ are social network feature vector. $\mathbf{e_i^t} = \langle t_1, t_2, \cdots, t_k \rangle$ is the feature vector of the baseline. These email feature vectors then can be used as input to a learning algorithm. The basic features are full text features such as *from*, *to*, *cc*, *title*, and *body text* from the email.

The social-network based features are represented as follows: We use a m-dimensional sub-vector to represent the Newman (NM), K-Means (KM), Spectral (SC), or CONCOR clustering (CC) where m be the number of clusters produced by the clustering algorithm: each element of the sub-vector is 1 if the user belongs to the corresponding cluster, or 0 otherwise; each user can

Feature	Description
ReplyToMine	Reply to my message
MyAddrInFrom	Whether my address is listed in FROM field
MyAddrInTo	Whether my address is listed in TO field
MyAddrInCC	Whether my address is listed in CC field
NumRecipients	the number of recipients in TO and CC field
NumCC	the number of recipients in CC field
Folder	Folder that the email belongs to
Size	$log($ size of email $)$
Attachment	Whether the email has attachments

Table 4.1: The Meta-Level features

belong to one and only one cluster. We also use another sub-vector (7-dimensional) to represent the social importance (SI) features per user, whose elements are real-valued. In addition, we use a 5-dimensional sub-vector to represent the five LSPR scores per sender, i.e., the mixture weights of the user at the five importance levels. The concatenation of those sub-vectors together with the full text (FT) vector yields a synthetic vector per email message as its full representation.

4.6 Experiments

Basically we tested two conditions, online condition and batch condition. Online condition does not allow us to look at test instances at all as we can not see future data. However it does not mean that our learning framework is online adaptive where we continuously re-train or update our model whenever getting user feedback. Online condition is more close to real world settings but it could not utilize the structure of test data. Especially our dataset size is considerably smaller than actual users' INBOX size and thus our experimental analysis could be biased to the small sample messages.

In contrast to online condition, batch condition allows us to take advantage of test data social network structure during training and may produce better estimations. Therefore, we may have more stable and close to one's INBOX social network structures but we utilize the test dataset. Note that we do not use any test label information. We first evaluate strict online evaluation condition and then report batch evaluation condition experiments.

4.6.1 Online Condition

Data For this condition, we evaluated on the first data collection which consists of seven users who actually submitted more than 200 messages with importance labels. Specifically, we again sort the email messages in a temporal order for each personal collection, and split the sorted list into 70% and 30% portions. The 70% portion was used for training and parameter tuning, and the remaining 30% was used for testing. Table 4.2 summarizes the dataset statistics (message counts). The full set of training examples in each personal data collection was used to induce the Newman-cluster (NC) features and the Social Importance (SI) features. For LSPR features, we

used all the messages in the training set to propagate 30, 60, 90, 120 and 150 labels in the training set, respectively.

Note that all the test-set sizes are even smaller than the dataset in Chapter 3 due to 30% testing and smaller dataset size. Here, the average number of test message is 169.4 among seven users but we had 514.1 average test instances, which means we have less confidence on micro level significance test.

User	# of emails	# of train	# of labels	# of test
1	1750	1225	$30 \sim 150$	525
2	376	263	$30 \sim 150$	113
3	484	339	$30 \sim 150$	145
4	596	417	$30 \sim 150$	179
5	233	163	$30 \sim 150$	70
6	279	195	$30 \sim 150$	84
7	234	164	$30 \sim 150$	70
Average	564.6	395.2	$30 \sim 150$	169.4

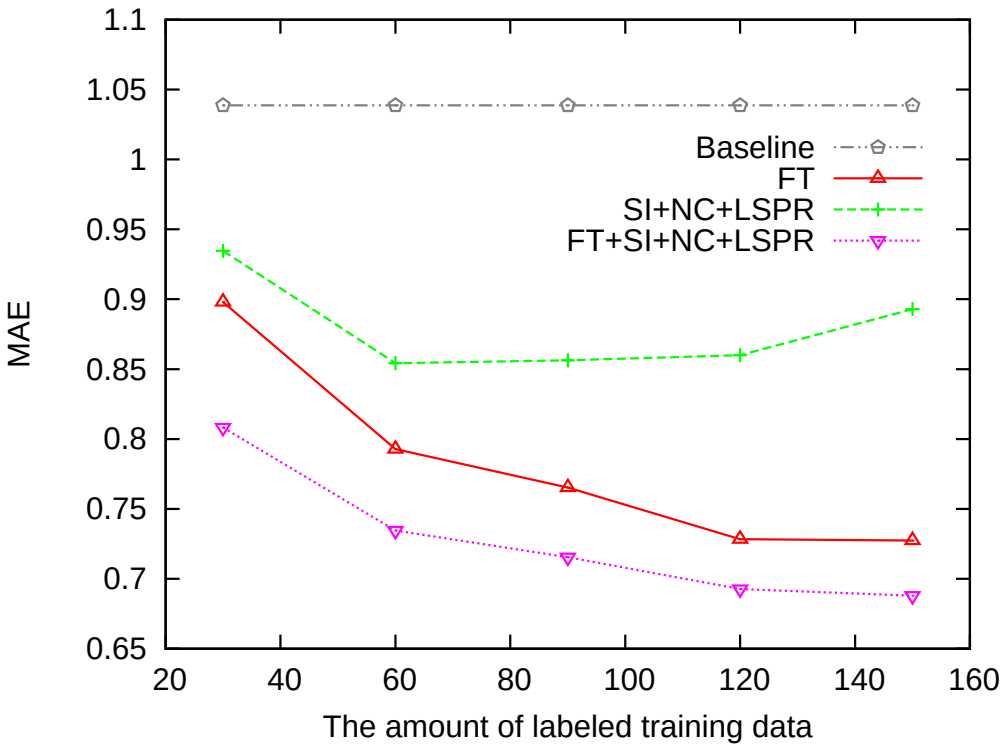
Table 4.2: 70% train and 30% test split on our early first data collection

Preprocessing We applied a multi-pass preprocessing to email messages. First, we applied email address canonicalization. Since each person may have multiple email accounts, it is necessary to unify them before applying social network analysis. For instance, ”John Smith” john.smith+@cs.cmu.edu, ”John” smith@cs.cmu.edu and ”John Smith” john747@gmail.com might be the email addresses of the same person. We used regular expression patterns and longest string matching algorithms to identify email addresses which may belong to the same user. We then manually checked all the groups and corrected the errors in the process. We also applied word tokenization and stemming using the Porter stemmer; we did not remove stop words from the title and body text.

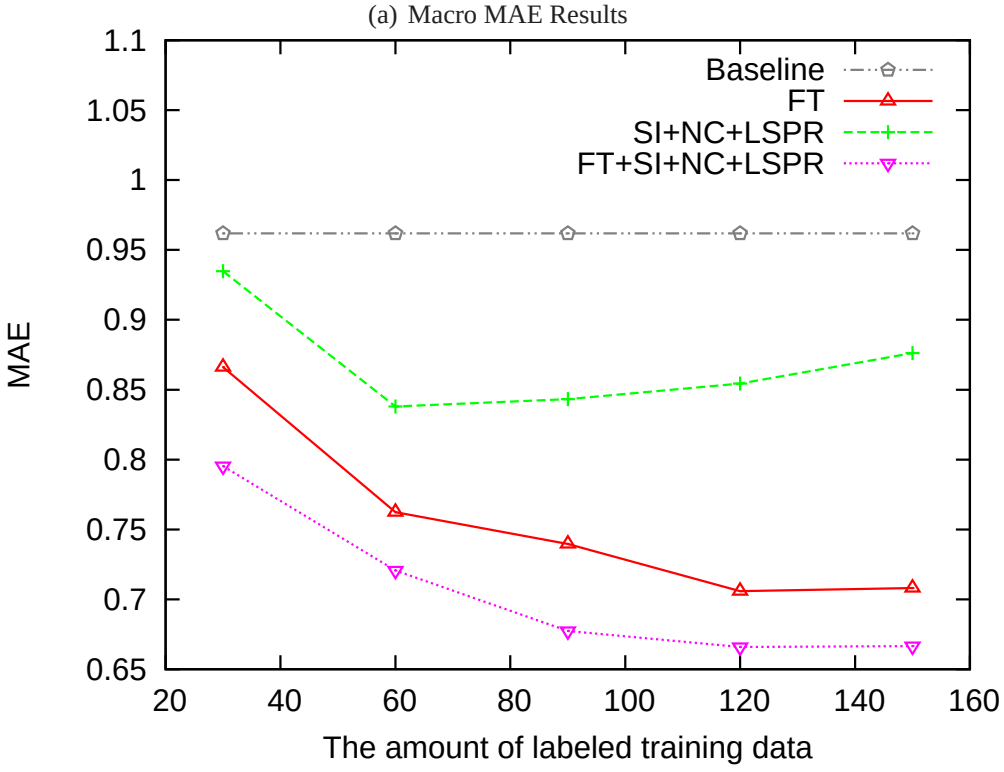
Classifiers We use five linear SVM classifiers for the prediction of importance level per email message (OVA). Each classifier takes the vector representation of each message (as described in Chapter 4.5) as its input, and produces a score with respect to a specific importance level.

Our baseline is again predicting to always priority level 3 out of 5 levels, which is the most common priority level on our data collection. We ran the SVM classifiers with the full text (FT) or all social network features (SI+NC+LSPR) for machine learning approach basis where all social network features are combining FT with Newman Clustering (NC), seven unsupervised social importance (SI) features and five semi-supervised LSPR features (SI+NC+LSPR). We also tested with FT with social network features, namely (FT+SI+NC+LSPR). We varied the number of the training labels per user from 30 to 150 labeled email messages.

Results and Analysis First of all, It can be observed that Baseline shows again the worst performance and the most results are statistically significant, shown in Figure 4.4, 4.5 and Table 4.3. Second, social network only (SI+NC+LSPR) or full text only (FT) showed significant improvement over baseline but full text (FT) showed better results than social network only features

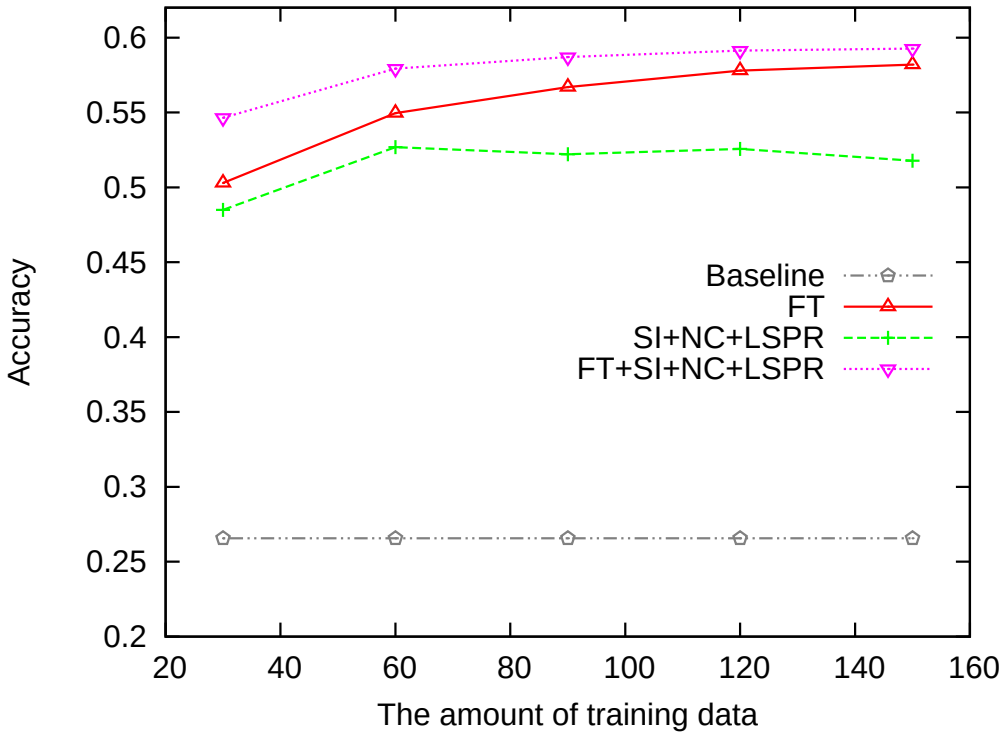


(a) Macro MAE Results

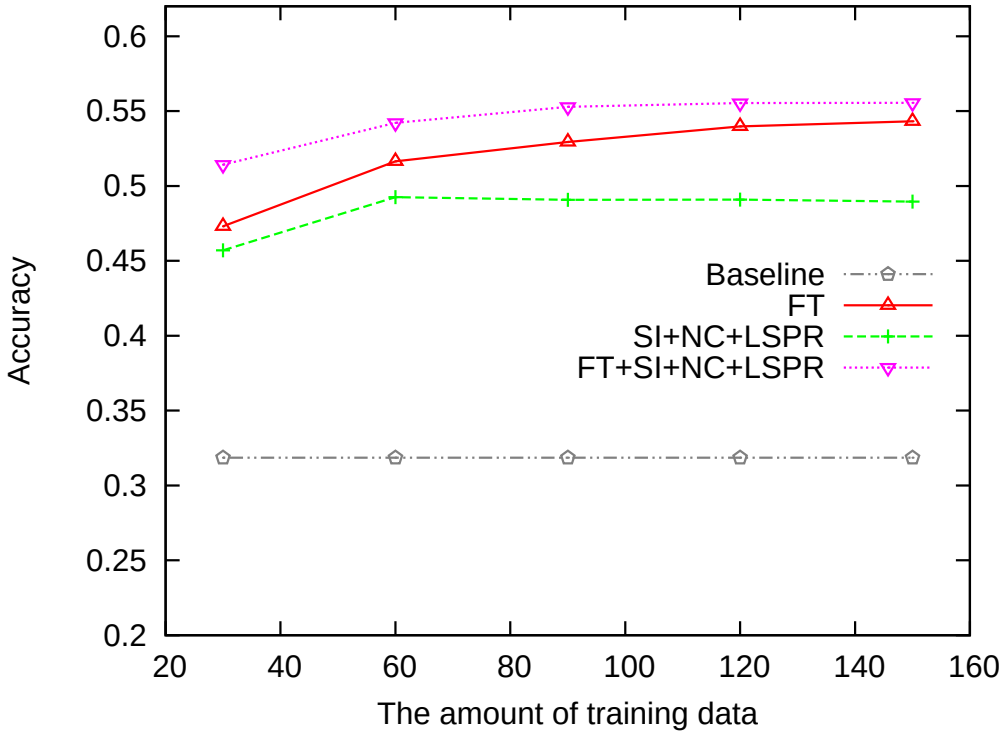


(b) Micro MAE Results

Figure 4.4: Overall MAE Results



(a) Macro Accuracy Results



(b) Micro Accuracy Results

Figure 4.5: Overall Accuracy Results

# of tr	Baseline(b)	FT(f)		SI+NC+LSPR(s)		FT+SI+NC+LSPR			
	MAE	MAE	p-value(b)	MAE	p-value(b)	MAE	p-value(b)	p-value(f)	p-value(s)
30	1.0387	0.8980	0.1382	0.9346	0.2127	**0.8081**	0.0755	* 0.0170	* 0.0239
60	1.0387	0.7928	* 0.0472	0.8543	0.0946	**0.7345**	* 0.0332	0.0642	* 0.0297
90	1.0387	0.7652	* 0.0419	0.8563	0.0908	**0.7154**	* 0.0248	* 0.0053	* 0.0197
120	1.0387	0.7282	* 0.0227	0.8599	0.0855	**0.6927**	* 0.0161	* 0.0012	* 0.0238
150	1.0387	0.7274	* 0.0233	0.8930	0.1429	**0.6879**	* 0.0143	* 0.0011	* 0.0029

(a) Macro MAE Results

# of tr	Baseline(b)	FT(f)		SI+NC+LSPR(s)		FT+SI+NC+LSPR			
	MAE	MAE	p-value(b)	MAE	p-value(b)	MAE	p-value(b)	p-value(f)	p-value(s)
30	0.9619	0.8661	* 0.0022	0.9348	* 0.2931	**0.7953**	* 0.0000	* 0.0000	* 0.0000
60	0.9619	0.7624	* 0.0000	0.8381	* 0.0000	**0.7207**	* 0.0000	* 0.0099	* 0.0000
90	0.9619	0.7397	* 0.0000	0.8433	* 0.0014	**0.6775**	* 0.0000	* 0.0000	* 0.0000
120	0.9619	0.7058	* 0.0000	0.8544	* 0.0002	**0.6658**	* 0.0000	* 0.0011	* 0.0000
150	0.9619	0.7081	* 0.0000	0.8763	* 0.0053	**0.6665**	* 0.0000	* 0.0025	* 0.0000

(b) Micro MAE Results

# of tr	Baseline(b)	FT(f)		SI+NC+LSPR(s)		FT+SI+NC+LSPR			
	ACC	ACC	p-value(b)	ACC	p-value(b)	ACC	p-value(b)	p-value(f)	p-value(s)
30	0.2657	0.5029	* 0.0095	0.4850	* 0.0162	**0.5464**	* 0.0041	* 0.0149	* 0.0069
60	0.2657	0.5496	* 0.0031	0.5269	* 0.0071	**0.5793**	* 0.0021	* 0.0131	* 0.0292
90	0.2657	0.5670	* 0.0024	0.5220	* 0.0061	**0.5870**	* 0.0015	* 0.0142	* 0.0121
120	0.2657	0.5779	* 0.0017	0.5257	* 0.0061	**0.5913**	* 0.0014	0.0531	* 0.0172
150	0.2657	0.5820	* 0.0018	0.5178	* 0.0056	**0.5927**	* 0.0015	0.0553	* 0.0020

(c) Macro Accuracy Results

# of tr	Baseline(b)	FT(f)		SI+NC+LSPR(s)		FT+SI+NC+LSPR			
	ACC	ACC	p-value(b)	ACC	p-value(b)	ACC	p-value(b)	p-value(f)	p-value(s)
30	0.3186	0.4731	* 0.0000	0.4570	* 0.0000	**0.5142**	* 0.0000	* 0.0014	* 0.0000
60	0.3186	0.5164	* 0.0000	0.4925	* 0.0000	**0.5422**	* 0.0000	* 0.0197	* 0.0001
90	0.3186	0.5294	* 0.0000	0.4907	* 0.0000	**0.5528**	* 0.0000	0.0827	* 0.0000
120	0.3186	0.5397	* 0.0000	0.4908	* 0.0000	**0.5554**	* 0.0000	0.1748	* 0.0000
150	0.3186	0.5431	* 0.0000	0.4895	* 0.0000	**0.5556**	* 0.0000	0.2280	* 0.0000

(d) Micro Accuracy Results

Table 4.3: Evaluation results of varying training set size. It shows MAE with p-value (macro: paired t-test, micro: signed rank test) and Accuracy (macro: paired t-test, micro: proportional test), indicating the statistical significances of better performance compared to the baseline(b), FT(f) or SI+NC+LSPR(s). Numbers in bold font indicating the best approach for each fixed training-set size. The star indicates the p-values equal or less than 5%.

(SI+NC+LSPR). When we combined text with social network features (FT+SI+NC+LSPR), we could get further improvement and most of them are statistically significantly better than full text (FT) or social network (SI+NC+LSPR) except 120 and 150 Accuracy over FT. Therefore, we could verify that social network induced features are informative and we should consider both text and social network induced features together.

4.6.2 Batch Condition

Data and Classifiers As a batch condition, we used the same split with Chapter 3, which is the first 150 as training and the remaining as testing and the email messages were also sorted in a temporal order for each personal collection. Table 3.1 summarizes the dataset statistics (message counts). Note that this dataset has not only more number of users but also much large number of test messages. We also ran the additional social clustering features such as CONCOR Clustering (CC), KMeans Clustering (KM), and Spectral Clustering (SC).

Social Clustering Results First of all, the performance of baseline and FT is worse than the performance of online conditioned baseline and FT, which tells us that without considering social network structure, it is more difficult to predict with batch condition.

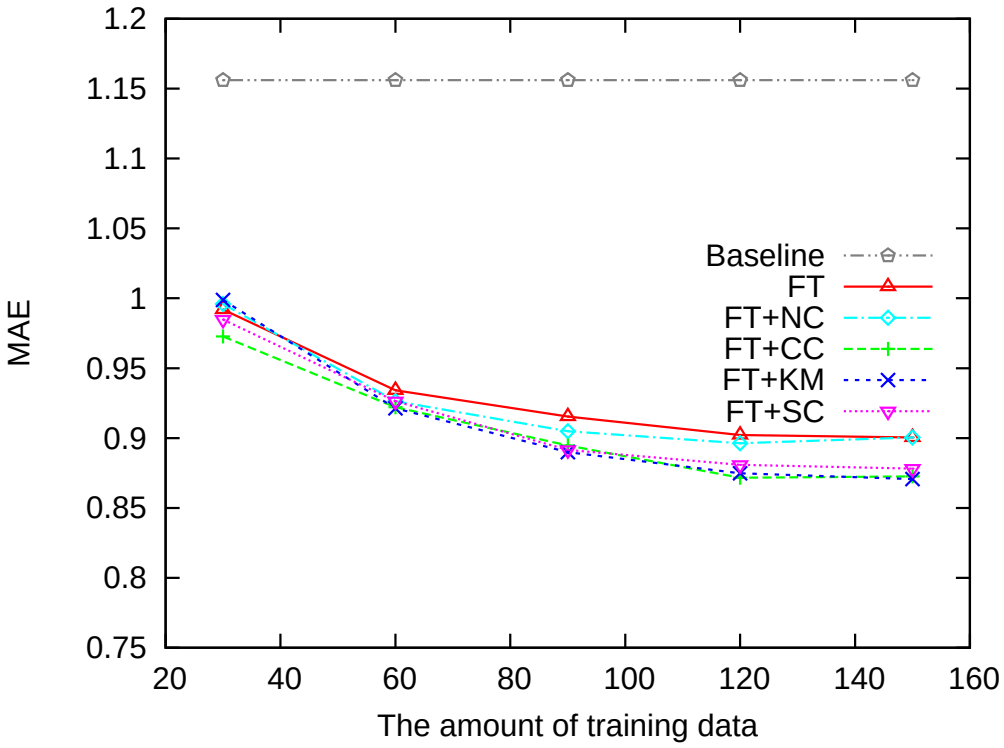
Second, We could observe that the social context captured by unsupervised social clustering is useful in predicting the personal importance of email messages, shown in Figure 4.6, 4.7 and Table 4.4. So it can be candidate features for handling the paucity of training label. Most clustering algorithm performed similarly in terms of Accuracy but Newman clustering (NC) showed the little improvement over FT with MAE. For our additional analysis, we will use NC as further consideration of social feature combinations due to consistency of our previous online experiments.

Social Importance and LSPR Results Social Importance (SI) features show consistent improvements and the improvement is significant, which means the social importance also can be captures through social network analysis and it can leverage the burden of the lack of training label in personalized importance prediction problem.

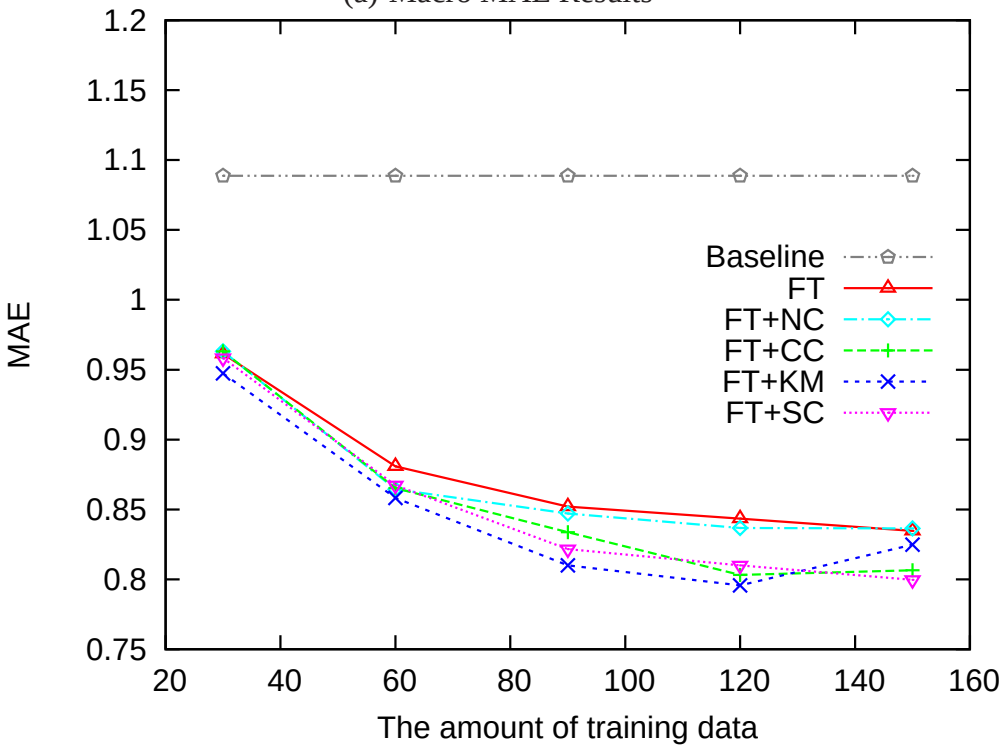
In case of LSPR, it did show improvement in terms of MAE but LSPR did not show significant improvement on Accuracy. Most p-values of SI is statistically significant and LSPR showed statistically significantly better than baseline. Semi-supervised LSPR, at least, showed the potential of improvements and it will be further investigated on the combining social features.

Combining Diverse Social Features The results we got are similar to the results of our online condition. Social features only (SI+NC+LSPR) show significant improvements over baseline and the results are statistically significant but the social features only can not outperform full text (FT) features, shown in Figure 4.10, 4.11 and Table 4.6.

Second, full combination of text and social features (FT+SI+NC+LSPR) showed significant improvements over FT, SI+NC+LSPR, or baseline and most results are statistically significant especially with micro level tests, which support our main claim that social network induced features can leverage the paucity of training data and produce robust prediction.

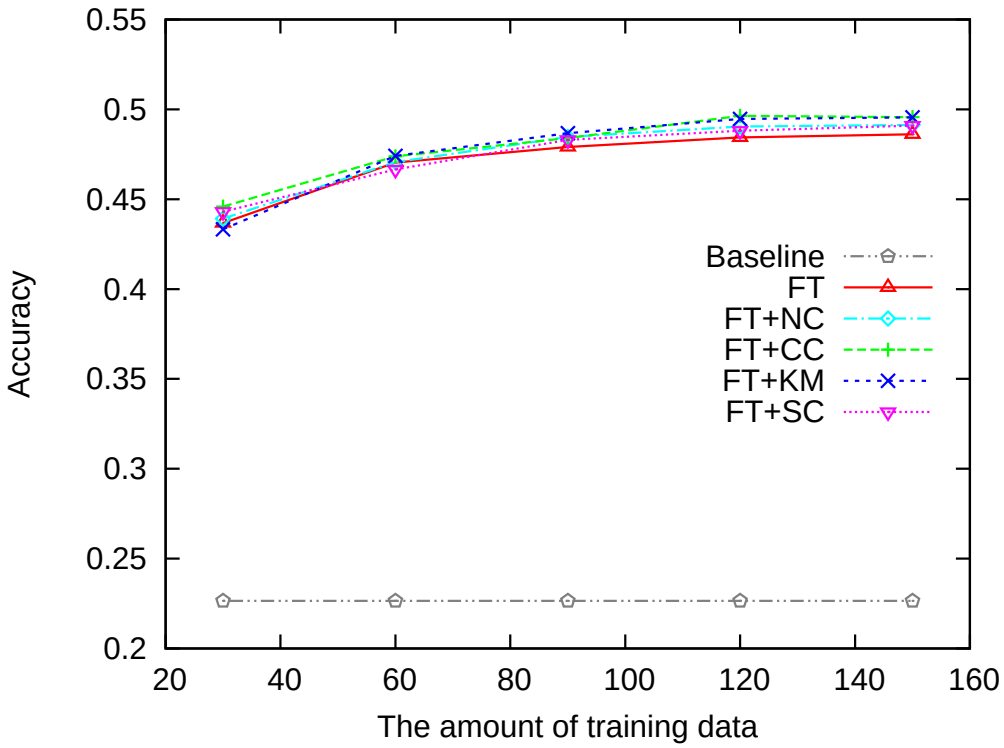


(a) Macro MAE Results

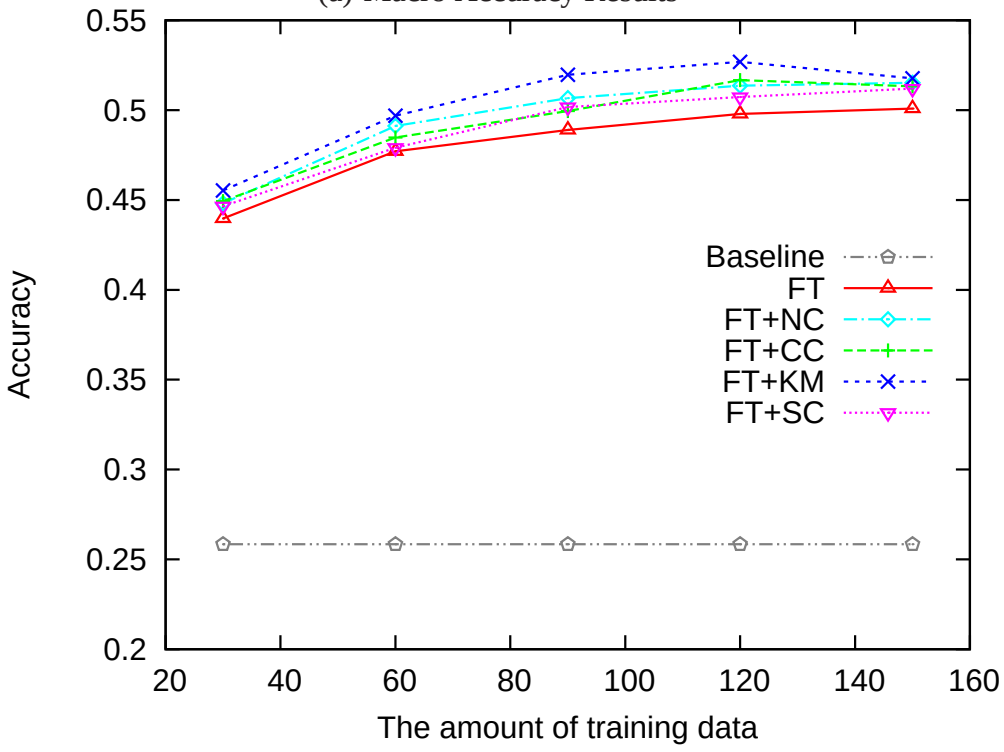


(b) Micro MAE Results

Figure 4.6: Social clustering algorithm comparison results (MAE)



(a) Macro Accuracy Results



(b) Micro Accuracy Results

Figure 4.7: Social clustering algorithm comparison results (Accuracy)

# of tr	Baseline(b)	FT(f)		FT+NC		
	MAE	MAE	p-value(b)	MAE	p-value(b)	p-value(f)
30	1.1560	0.9920	* 0.0132	**0.9960**	* 0.0237	0.5523
60	1.1560	0.9342	* 0.0026	**0.9264**	* 0.0024	0.3540
90	1.1560	0.9153	* 0.0015	**0.9049**	* 0.0012	0.3148
120	1.1560	0.9022	* 0.0009	**0.8963**	* 0.0011	0.3931
150	1.1560	0.9004	* 0.0010	**0.9005**	* 0.0018	0.5007

(a) Macro MAE Results

# of tr	Baseline(b)	FT(f)		FT+NC		
	MAE	MAE	p-value(b)	MAE	p-value(b)	p-value(f)
30	1.0887	0.9614	* 0.0000	**0.9632**	* 0.0000	0.0615
60	1.0887	0.8809	* 0.0000	**0.8645**	* 0.0000	0.1338
90	1.0887	0.8520	* 0.0000	**0.8470**	* 0.0000	0.3188
120	1.0887	0.8435	* 0.0000	**0.8368**	* 0.0000	0.3290
150	1.0887	0.8347	* 0.0000	**0.8365**	* 0.0000	0.1799

(b) Micro MAE Results

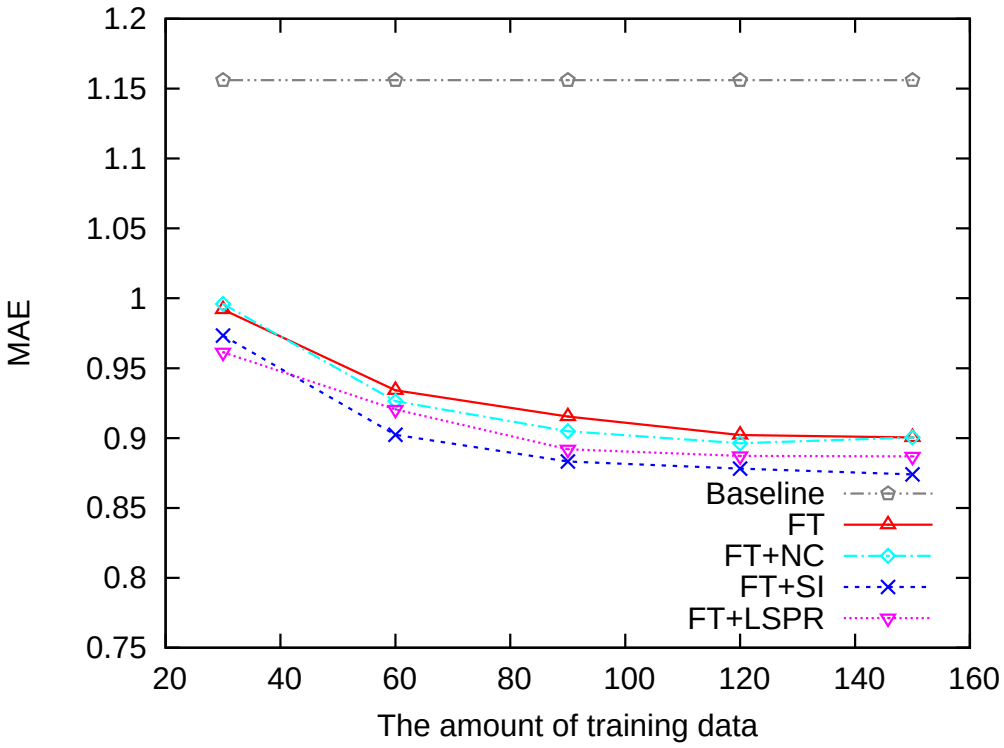
# of tr	Baseline(b)	FT(f)		FT+NC		
	ACC	ACC	p-value(b)	ACC	p-value(b)	p-value(f)
30	0.2265	0.4367	* 0.0000	**0.4391**	* 0.0000	0.4189
60	0.2265	0.4704	* 0.0000	**0.4706**	* 0.0000	0.4896
90	0.2265	0.4791	* 0.0000	**0.4847**	* 0.0000	0.2455
120	0.2265	0.4844	* 0.0000	**0.4905**	* 0.0000	0.2391
150	0.2265	0.4861	* 0.0000	**0.4913**	* 0.0000	0.2747

(c) Macro Accuracy Results

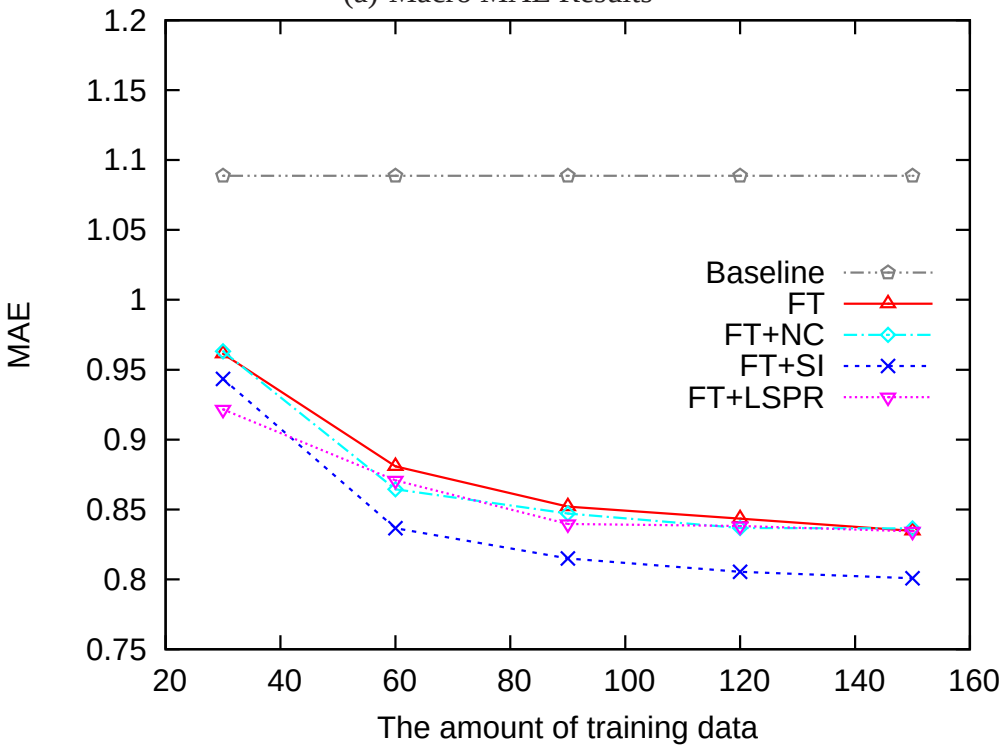
# of tr	Baseline(b)	FT(f)		FT+NC		
	ACC	ACC	p-value(b)	ACC	p-value(b)	p-value(f)
30	0.2584	0.4397	* 0.0000	**0.4482**	* 0.0000	0.2980
60	0.2584	0.4771	* 0.0000	**0.4912**	* 0.0000	0.4777
90	0.2584	0.4889	* 0.0000	**0.5067**	* 0.0000	0.1112
120	0.2584	0.4978	* 0.0000	**0.5135**	* 0.0000	0.0925
150	0.2584	0.5008	* 0.0000	**0.5152**	* 0.0000	0.1307

(d) Micro Accuracy Results

Table 4.4: Evaluation results of varying training set size. It shows MAE with p-value (macro: paired t-test, micro: signed rank test) and Accuracy (macro: paired t-test, micro: proportional test), indicating the statistical significances of better performance compared to the baseline(b) or FT(f). Numbers in bold font indicating the best approach for each fixed training-set size. The star indicates the p-values equal or less than 5%.

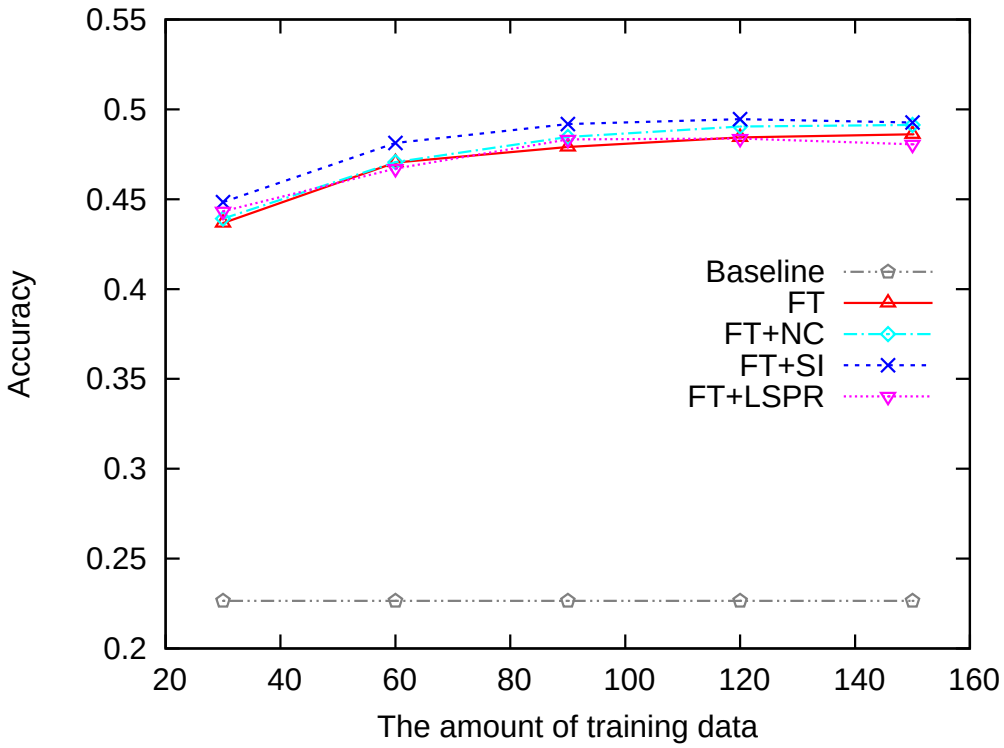


(a) Macro MAE Results

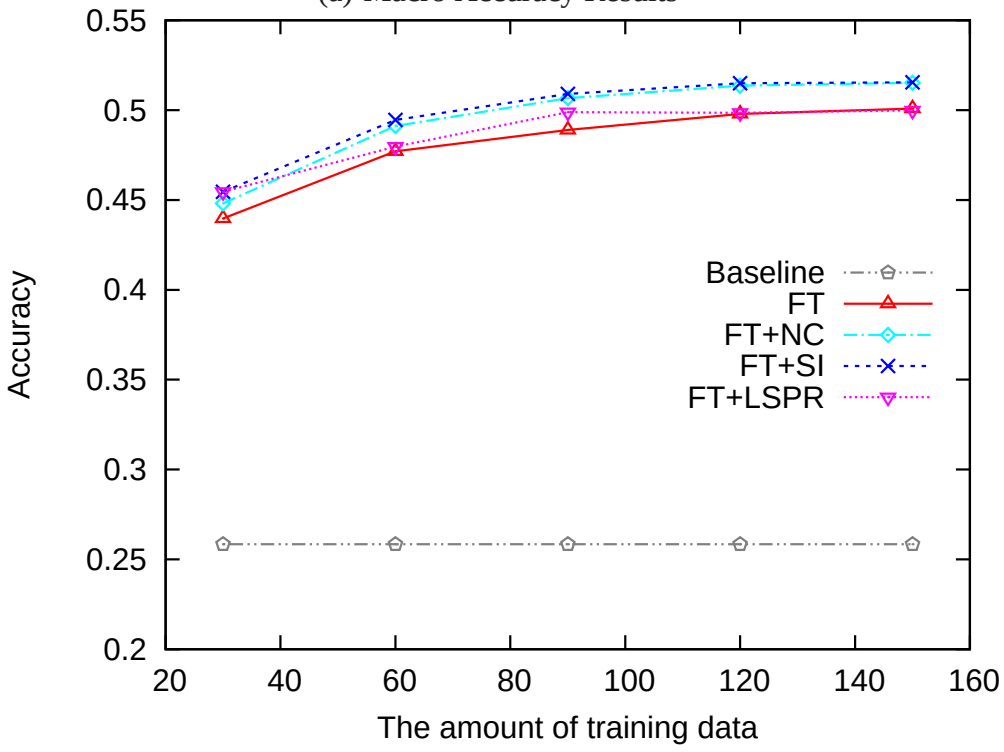


(b) Micro MAE Results

Figure 4.8: Social feature comparison results (MAE)



(a) Macro Accuracy Results



(b) Micro Accuracy Results

Figure 4.9: Social feature comparison results (Accuracy)

# of tr	Baseline(b)	FT(f)	FT+SI			FT+LSPR		
	MAE	MAE	MAE	p-value(b)	p-value(f)	MAE	p-value(b)	p-value(f)
30	1.1560	0.9920	0.9734	* 0.0110	0.1365	**0.9614**	* 0.0084	0.0757
60	1.1560	0.9342	**0.9024**	* 0.0010	* 0.0030	0.9205	* 0.0022	0.2341
90	1.1560	0.9153	**0.8832**	* 0.0005	* 0.0004	0.8919	* 0.0010	0.1376
120	1.1560	0.9022	**0.8781**	* 0.0005	* 0.0384	0.8873	* 0.0008	0.2369
150	1.1560	0.9004	**0.8739**	* 0.0005	* 0.0226	0.8869	* 0.0009	0.2575

(a) Macro MAE Results

# of tr	Baseline(b)	FT(f)	FT+SI			FT+LSPR		
	MAE	MAE	MAE	p-value(b)	p-value(f)	MAE	p-value(b)	p-value(f)
30	1.0887	0.9614	**0.9434**	* 0.0000	* 0.0000	0.9216	* 0.0000	* 0.0000
60	1.0887	0.8809	**0.8365**	* 0.0000	* 0.0000	0.8708	* 0.0000	0.4104
90	1.0887	0.8520	**0.8149**	* 0.0000	* 0.0000	0.8396	* 0.0000	0.2405
120	1.0887	0.8435	**0.8053**	* 0.0000	* 0.0000	0.8382	* 0.0000	* 0.0052
150	1.0887	0.8347	**0.8008**	* 0.0000	* 0.0000	0.8344	* 0.0000	* 0.0252

(b) Micro MAE Results

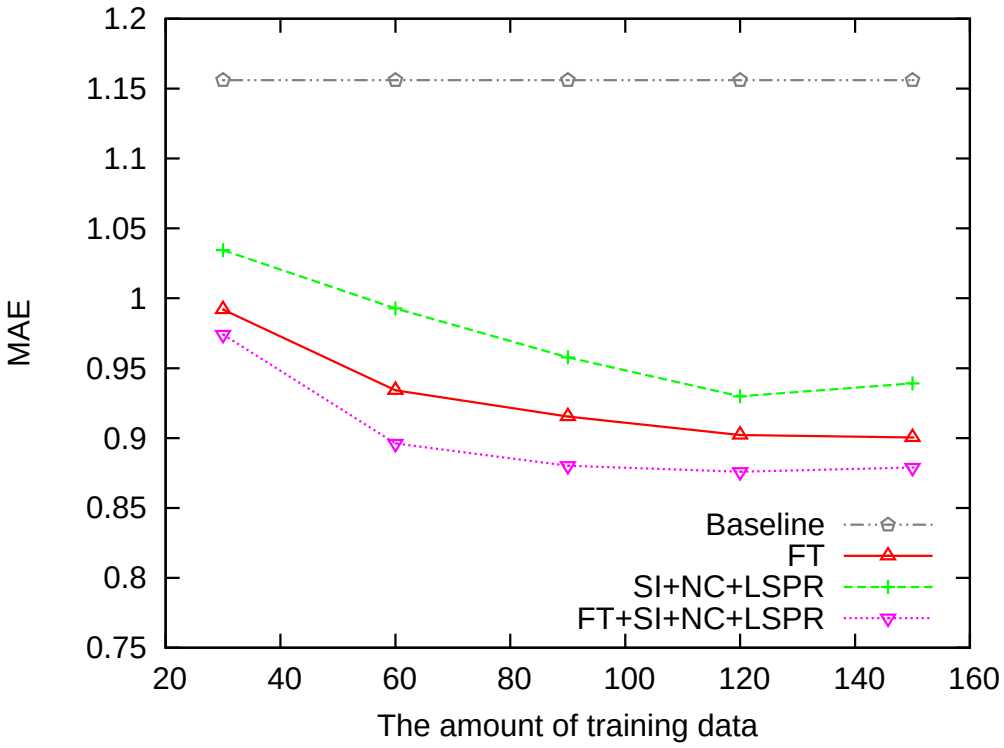
# of tr	Baseline(b)	FT(f)	FT+SI			FT+LSPR		
	ACC	ACC	ACC	p-value(b)	p-value(f)	ACC	p-value(b)	p-value(f)
30	0.2265	0.4367	**0.4484**	* 0.0000	* 0.0336	0.4433	* 0.0000	0.1729
60	0.2265	0.4704	**0.4813**	* 0.0000	* 0.0047	0.4670	* 0.0000	0.6730
90	0.2265	0.4791	**0.4918**	* 0.0000	* 0.0018	0.4833	* 0.0000	0.2778
120	0.2265	0.4844	**0.4945**	* 0.0000	* 0.0363	0.4837	* 0.0000	0.5340
150	0.2265	0.4861	**0.4926**	* 0.0000	0.0819	0.4805	* 0.0000	0.7258

(c) Macro Accuracy Results

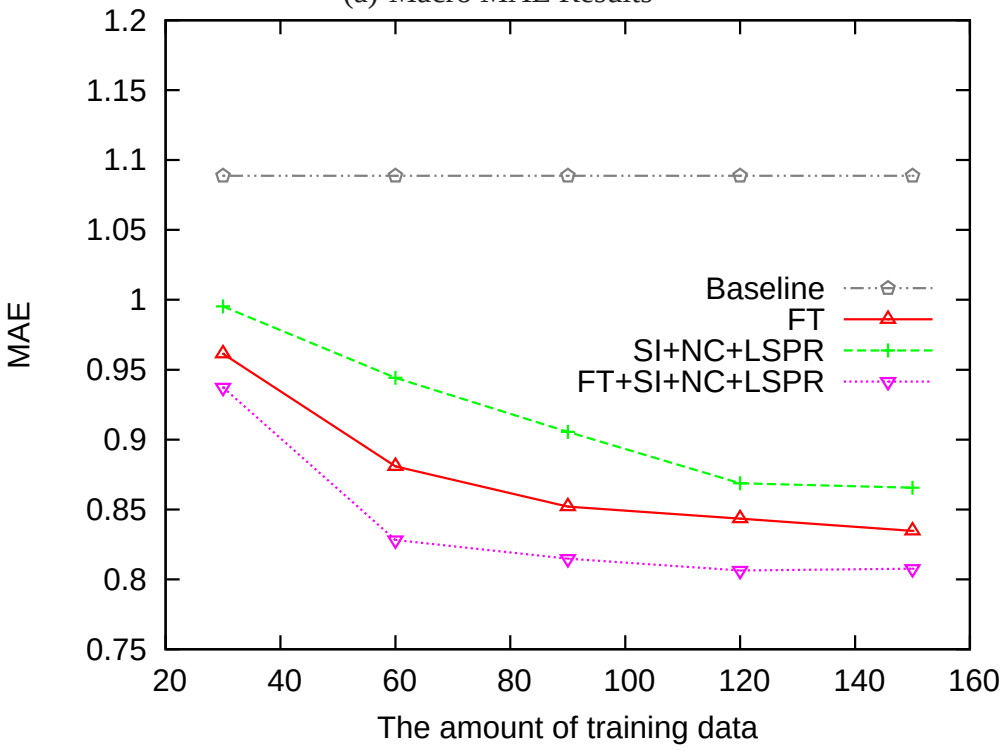
# of tr	Baseline(b)	FT(f)	FT+SI			FT+LSPR		
	ACC	ACC	ACC	p-value(b)	p-value(f)	ACC	p-value(b)	p-value(f)
30	0.2584	0.4397	**0.4546**	* 0.0000	* 0.0053	**0.4546**	* 0.0000	0.0748
60	0.2584	0.4771	**0.4946**	* 0.0000	* 0.0086	0.4796	* 0.0000	0.7690
90	0.2584	0.4889	**0.5090**	* 0.0000	* 0.0027	0.4988	* 0.0000	0.1803
120	0.2584	0.4978	**0.5149**	* 0.0000	* 0.0138	0.4985	* 0.0000	0.5583
150	0.2584	0.5008	**0.5154**	* 0.0000	0.0785	0.4999	* 0.0000	0.8885

(d) Macro Accuracy Results

Table 4.5: Evaluation results of varying training set size. It shows MAE with p-value (macro: paired t-test, micro: signed rank test) and Accuracy (macro: paired t-test, micro: proportional test), indicating the statistical significances of better performance compared to the baseline(b) or FT(f). Numbers in bold font indicating the best approach for each fixed training-set size. The star indicates the p-values equal or less than 5%.

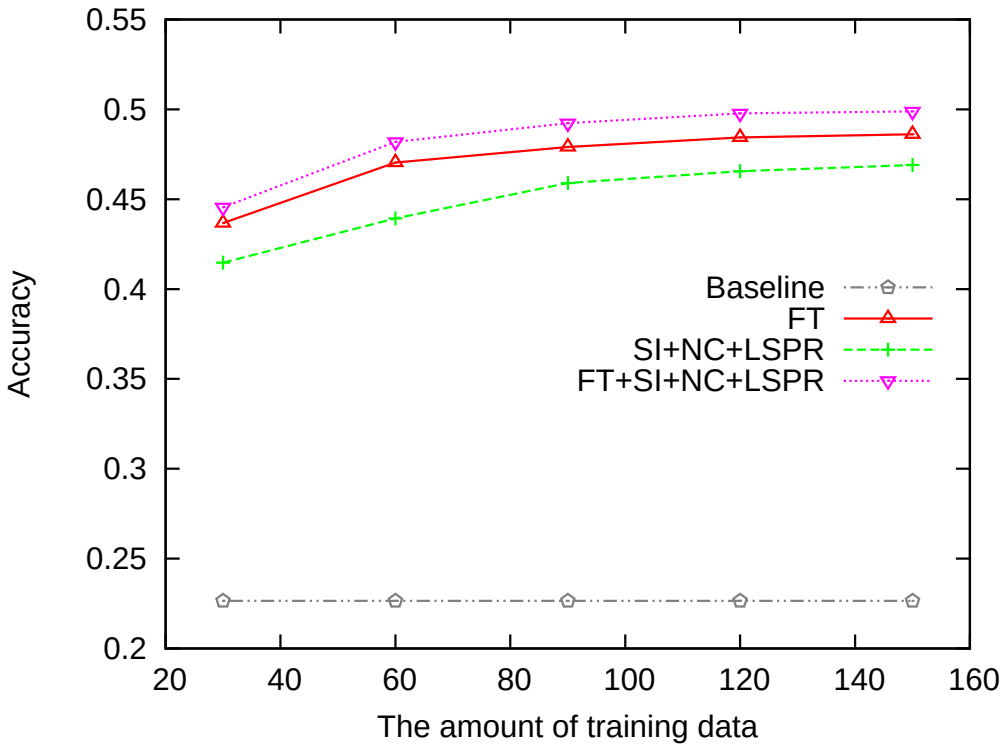


(a) Macro MAE Results

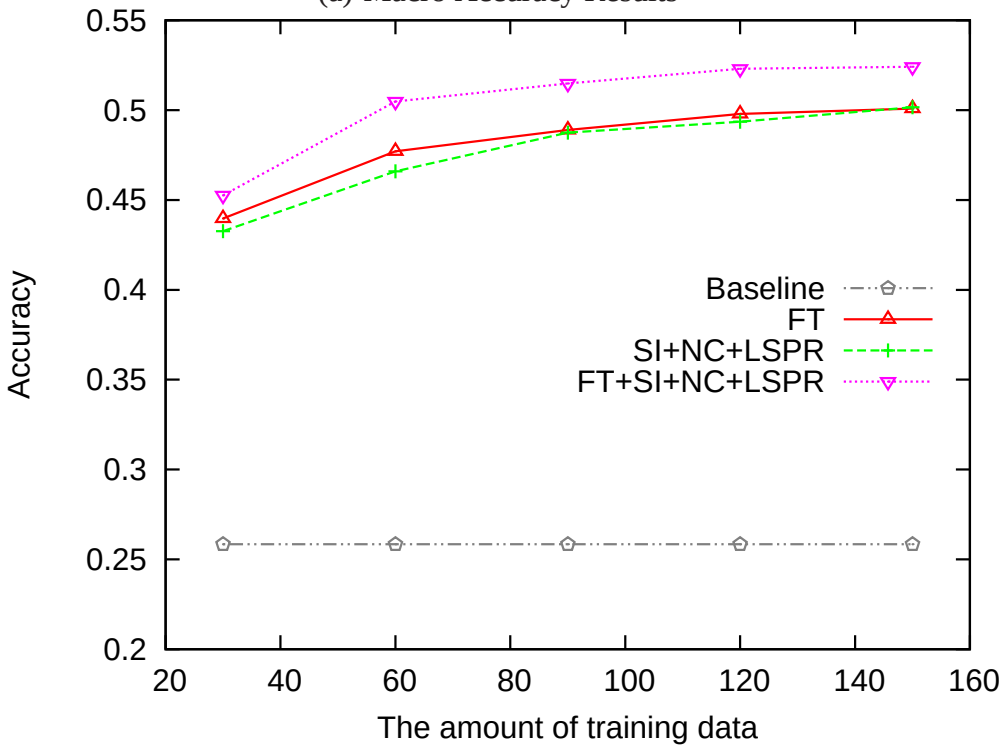


(b) Micro MAE Results

Figure 4.10: Combining social feature results (MAE)



(a) Macro Accuracy Results



(b) Micro Accuracy Results

Figure 4.11: Combining social feature results (Accuracy)

# of tr	Baseline(b)	FT(f)		SI+NC+LSPR(s)		FT+SI+NC+LSPR			
	MAE	MAE	p-value(b)	MAE	p-value(b)	MAE	p-value(b)	p-value(f)	p-value(s)
30	1.1560	0.9920	* 0.0132	1.0345	0.0522	**0.9740**	* 0.0120	0.2612	* 0.0015
60	1.1560	0.9342	* 0.0026	0.9928	* 0.0248	**0.8962**	* 0.0009	* 0.0245	* 0.0010
90	1.1560	0.9153	* 0.0015	0.9577	* 0.0097	**0.8802**	* 0.0006	* 0.0414	* 0.0030
120	1.1560	0.9022	* 0.0009	0.9298	* 0.0070	**0.8759**	* 0.0007	0.1056	0.0551
150	1.1560	0.9004	* 0.0010	0.9391	* 0.0107	**0.8790**	* 0.0008	0.1557	* 0.0311

(a) Macro MAE Results

# of tr	Baseline(b)	FT(f)		SI+NC+LSPR(s)		FT+SI+NC+LSPR			
	MAE	MAE	p-value(b)	MAE	p-value(b)	MAE	p-value(b)	p-value(f)	p-value(s)
30	1.0887	0.9614	* 0.0000	0.9953	* 0.0000	**0.9374**	* 0.0000	0.2509	* 0.0000
60	1.0887	0.8809	* 0.0000	0.9443	* 0.0000	**0.8281**	* 0.0000	* 0.0000	* 0.0000
90	1.0887	0.8520	* 0.0000	0.9056	* 0.0000	**0.8147**	* 0.0000	* 0.0000	* 0.0000
120	1.0887	0.8435	* 0.0000	0.8688	* 0.0000	**0.8064**	* 0.0000	* 0.0000	* 0.0000
150	1.0887	0.8347	* 0.0000	0.8656	* 0.0000	**0.8077**	* 0.0000	* 0.0000	* 0.0000

(b) Micro MAE Results

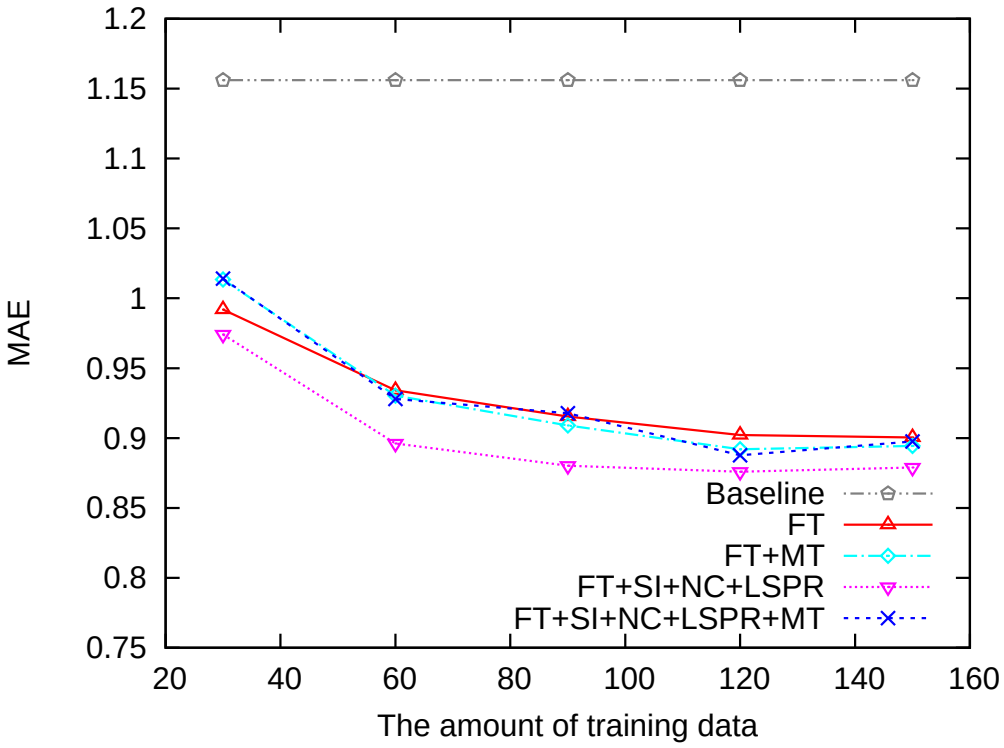
# of tr	Baseline(b)	FT(f)		SI+NC+LSPR(s)		FT+SI+NC+LSPR			
	ACC	ACC	p-value(b)	ACC	p-value(b)	ACC	p-value(b)	p-value(f)	p-value(s)
30	0.2265	0.4367	* 0.0000	0.4147	* 0.0000	**0.4455**	* 0.0000	0.1850	* 0.0003
60	0.2265	0.4704	* 0.0000	0.4393	* 0.0000	**0.4819**	* 0.0000	0.0724	* 0.0001
90	0.2265	0.4791	* 0.0000	0.4589	* 0.0000	**0.4923**	* 0.0000	* 0.0369	* 0.0002
120	0.2265	0.4844	* 0.0000	0.4656	* 0.0000	**0.4977**	* 0.0000	0.0591	* 0.0035
150	0.2265	0.4861	* 0.0000	0.4690	* 0.0000	**0.4988**	* 0.0000	0.0640	* 0.0066

(c) Macro Accuracy Results

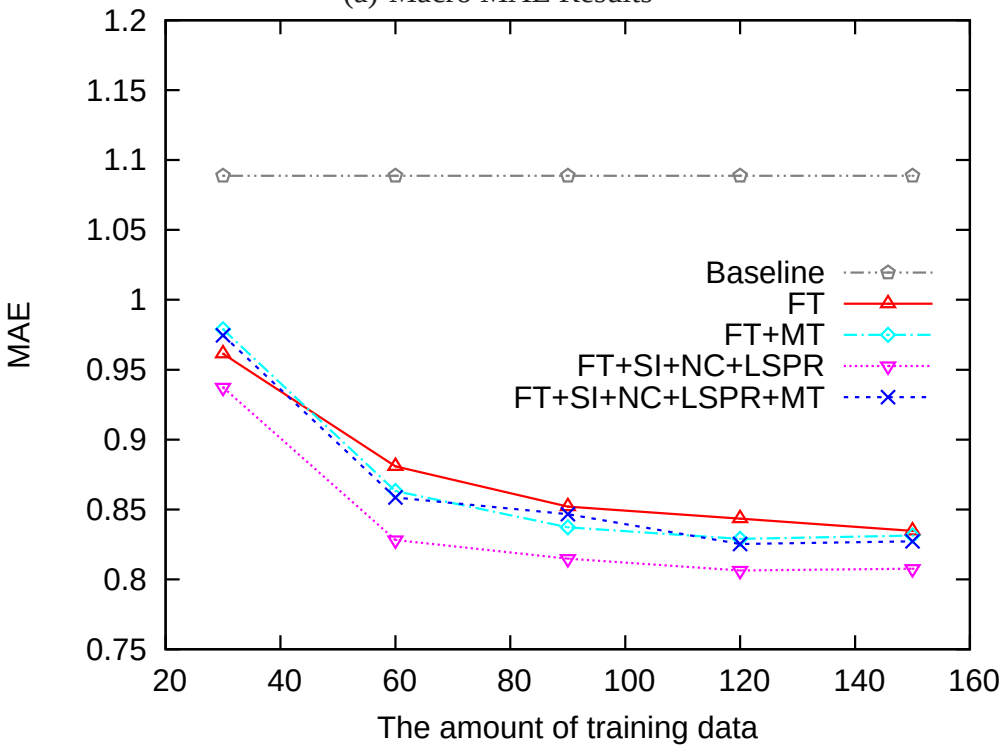
# of tr	Baseline(b)	FT(f)		SI+NC+LSPR(s)		FT+SI+NC+LSPR			
	ACC	ACC	p-value(b)	ACC	p-value(b)	ACC	p-value(b)	p-value(f)	p-value(s)
30	0.2584	0.4397	* 0.0000	0.4326	* 0.0000	**0.4526**	* 0.0000	* 0.0275	* 0.0000
60	0.2584	0.4771	* 0.0000	0.4659	* 0.0000	**0.5048**	* 0.0000	* 0.0058	* 0.0000
90	0.2584	0.4889	* 0.0000	0.4874	* 0.0000	**0.5149**	* 0.0000	* 0.0019	* 0.0000
120	0.2584	0.4978	* 0.0000	0.4936	* 0.0000	**0.5230**	* 0.0000	* 0.0018	* 0.0000
150	0.2584	0.5008	* 0.0000	0.5017	* 0.0000	**0.5241**	* 0.0000	* 0.0029	* 0.0000

(d) Micro Accuracy Results

Table 4.6: Evaluation results of varying training set size. It shows MAE with p-value (macro: paired t-test, micro: signed rank test) and Accuracy (macro: paired t-test, micro: proportional test), indicating the statistical significances of better performance compared to the baseline(b), FT(f) or SI+NC+LSPR(s). Numbers in bold font indicating the best approach for each fixed training-set size. The star indicates the p-values equal or less than 5%.

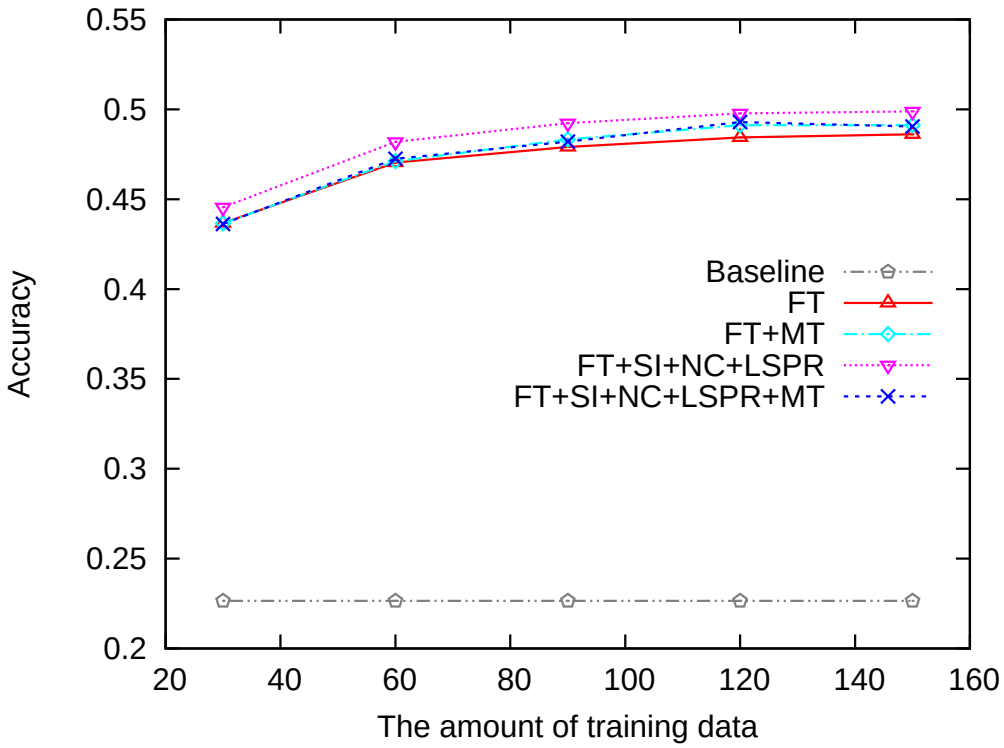


(a) Macro MAE Results

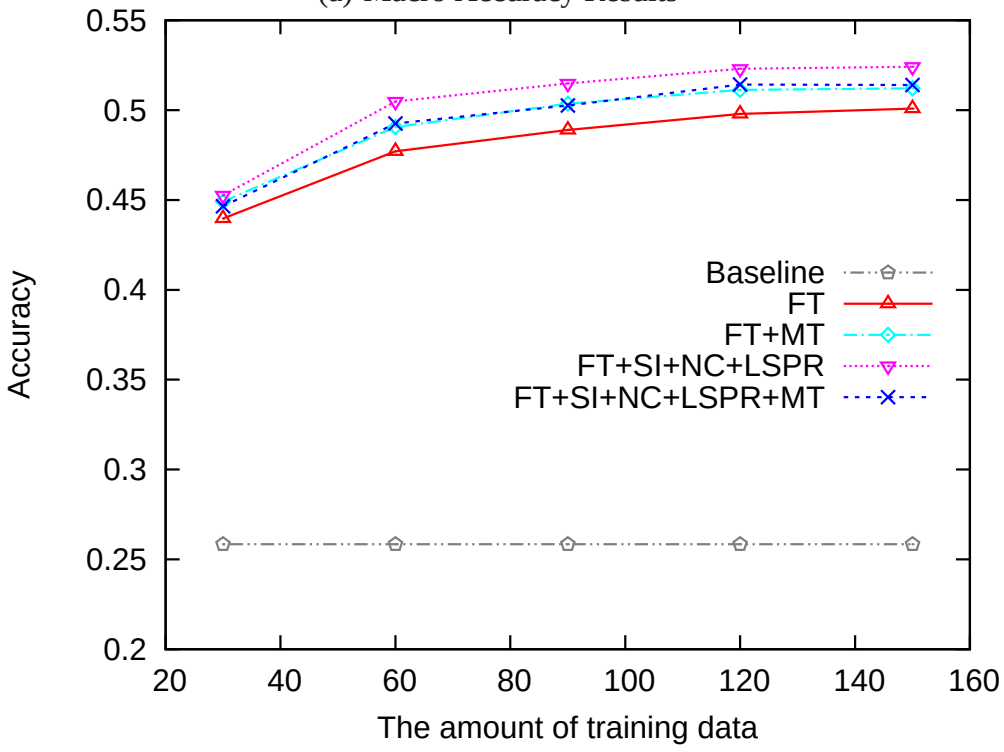


(b) Micro MAE Results

Figure 4.12: Meta feature results (MAE)



(a) Macro Accuracy Results



(b) Micro Accuracy Results

Figure 4.13: Meta feature results (Accuracy)

Meta Information Results Although meta information of email (MT) helped on certain ranges of training data (FT+MT) of Figure 4.12 and 4.13, the combined meta level features with social features, FT+SI+NC+LSPR+MT, showed similar performance with FT+MT. However, if we could incorporate additional information such as whether the user read the message or not, then meta information might be more useful.

4.7 Summary

We focus on social network analysis to capture user groups in each personal social network, and un-supervised and semi-supervised learning of rich features for representing user-centric social importance. These methods enable us to obtain an enriched vector representation of each new email message, as the basis of accurate modeling of individual users and for generating robust predictions for individual users in email prioritization. The effectiveness of the proposed approach is proved in our experiments on personal email data from multiple users. Gathering data to infer social networks of individual users requires only access to their email messages, no explicit labeling required, and thus in a real deployment, the social networks would be richer and perhaps even more useful. In case of meta level features, we could not observe the usefulness of meta level features when combined with other social network induced features but it could be the limitation of our user study.

5 Conclusions and Future Directions

5.1 Conclusions

To overcome email overloading, we proposed to prioritize email messages using machine learning methods. We face three major challenges: the lack of publicly available datasets, building personalized prioritization models, and sparse training data.

- **No Publicly Available Datasets** The most difficult challenge was the lack of publicly available email prioritization datasets. Due to privacy issues, no one wants to share email messages. Unlike spam filtering, where people do not mind sharing spam messages, we need fine grained priority labels with personal email messages. We had to build a new email prioritization dataset. We went through the IRB (Institutional Review Board) process and developed Microsoft Outlook and Mozilla Thunderbird plug-in programs. We recruited 39 subjects and tested our approaches on 19 subjects who actually submitted more than 200 messages.

- **Modeling Personal Email Priority** No one had addressed modeling personal email priority due to the lack of publicly available datasets. We analyzed the characteristics of email prioritization datasets by empirical evaluation and visualization of personal email data and observed that ordinal regression, generally believed to be the best and natural choice, showed worse results than classification based approaches on our email prioritization dataset. We further improved the prediction accuracy by utilizing partial ordinal relations among the priority levels through our proposed order based ensemble approaches.

- **Sparse Training Data** Training data is sparse because of personalization meaning that the same message might have different priority levels depending on the recipients. We enriched the representation of email messages through social network analysis and meta level features with no or little prior label information. Specifically, we captured social contexts through social clustering, social importance through social metrics and semi-supervised social weight through importance propagation on the personal social network. These personalized social network induced features did not outperform full text features but when we combined full text features with these induced social network features we further reduced the error rate of priority prediction.

Through our proposed modeling and enriched features, we verified that personalized email prioritization can be addressed by machine learning methods and we can alleviate the email overloading problem.

5.2 Future Directions

For future investigation, we would like to consider two main directions: deployment and new research in personalized email prioritization. Especially, we are eager to deploy what we learned in real-world applications and the following is our considerations:

- **User Interface** Email prioritization may not be useful without a proper user interface. One of the most important concerns is how to present predicted priorities of messages. It includes the layout of the reading pane of the email client, highlighting, fonts, colors, etc. How to

get feedback from the user is another important concern because proper user feedback is essential to adaptive personal priority learning. How and when to alert user are important as well. We may alert users through SMS (Short Messaging Service), IM (Instant Messaging), or a modal dialog box if the system detects a really important message. The timing of alert can be a critical issue for the productivity of users. If a system interrupts a user too frequently, then the productivity of the user might be decreased. However, if the user is not alerted, then the user may miss very important messages and loose one of reasons to use email prioritization.

- **Scalability** If email prioritization is deployed in Web services such as GMail, Hotmail or Yahoo! Mail, then our proposed approaches should be scalable, and thus we might seriously consider more efficient learning models or alternative social network induced features. For instance, we might consider triad count, the number of triangle, instead of clique counts because triad count can be efficiently calculated.

- **Benefits of Deployed Email Client** After an email client is deployed, the client program may access all of personal email messages and collect implicit feedbacks whereas we collected selectively submitted email messages and we did not be able to collect implicit feedback features. There are two notable benefits. First, given whole email messages of a user, we may build richer personal social network and we may improve the prediction accuracy further. Second, we may use implicit feedback features such as reading time, print, reply, forward, etc and may improve priority prediction accuracy. Such implicit feedback features can serve as the evaluation of the effectiveness such as the number of message selections or reading time changes.

As our future research direction toward personalized email prioritization, we are considering the following topics:

- **Urgency Prediction** Although our investigation of importance is indispensable to email prioritization, investigation into urgency prediction is also crucial. Because we already have collected urgency labels, we are ready to investigate similarities and dissimilarities between importance and urgency.

- **Topic Drifting** Due to limited amount of collected email messages, we assumed static priority models in this thesis. However, if we have user activities from a long span of time, we may also investigate the temporal nature of personal email priority such as topic or interest drifting, which requires email prioritization to be online and adaptive.

- **Dialog Structure Analysis** In email messages, we not only have social relations through the sending and receiving of messages but also thread structures. We may reconstruct dialog structures through email threads and such dialog structures may have correlation to priority, especially urgency prediction, because urgency is sensitive to the stage of discussion.

- **Temporal Expressions** Urgency can heavily depend on the remaining time to deadline. With the help of temporal expression analysis, we may compute the amount of remaining time to the event and it could be a critical feature for urgency prediction.

- **Joint Prediction of Importance and Urgency** Depending on users, importance and urgency might have correlation. For instance, if a message is not important at all, then it tends to be not urgent. The joint prediction of importance and urgency may provide better prioritization than if they were done separately.

References

[1] *CEAS 2005 - Second Conference on Email and Anti-Spam, July 21-22, 2005, Stanford University, California, USA*, 2005.

[2] RFC 1321. The MD5 algorithm. *www.ietf.org/rfc/rfc1321.txt*.

[3] Duane F. Alwin and Jon A. Krosnick. The reliability of survey attitude measurement: The influence of question and respondent attributes. *Sociological Methods Research*, 20(1):139–181, August 1991.

[4] Paul N. Bennett and Jaime G. Carbonell. Detecting action-items in e-mail. In Ricardo A. Baeza-Yates, Nivio Ziviani, Gary Marchionini, Alistair Moffat, and John Tait, editors, *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005*, pages 585–586. ACM, 2005.

[5] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[6] Boykin and Roychowdhury. Leveraging social networks to fight spam. *COMPUTER: IEEE Computer*, 38, 2005.

[7] P. Oscar Boykin and Vwani P. Roychowdhury. Leveraging social networks to fight spam. *Computer*, 38(4):61–68, 2005.

[8] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks ISDN Systems*, 30(1-7):107–117, 1998.

[9] JJ Cadiz, Laura Dabbish, Anoop Gupta, and Gina D. Venolia. Supporting email workflow. Technical Report MSR-TR-2001-88, Microsoft Research (MSR), September 2001.

[10] K. M. Carley, D. Columbus, M. DeReno, J. Reminga, and I. Moon. Ora user's guide 2007. *Carnegie Mellon University, SCS ISRI, Technical Report*, (07-115), 2007.

[11] Wei Chu and Zoubin Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6:1019–1041, 2005.

[12] Wei Chu and S. Sathiya Keerthi. New approaches to support vector ordinal regression. In Luc De Raedt and Stefan Wrobel, editors, *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, volume 119 of *ACM International Conference Proceeding Series*, pages 145–152. ACM, 2005.

[13] Stephen R. Covey. *The 7 Habits of Highly Effective People*. Free Press, 1990.

[14] Dabbish, Laura A., Kraut, Robert E., Susan Fussell, and Sara Kiesler. Understanding email use: Predicting action on a message. In *Proceedings of ACM CHI 2005 Conference on Human Factors in Computing Systems*, volume 1 of *Email and security*, pages 691–700, 2005.

[15] Laura A. Dabbish and Robert E. Kraut. Controlling interruptions: Awareness displays and social motivation for coordination. In James D. Herbsleb and Gary M. Olson, editors, *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work, CSCW 2004, Chicago, Illinois, USA, November 6-10, 2004*, pages 182–191. ACM, 2004.

[16] Laura A. Dabbish and Robert E. Kraut. Email overload at work: An analysis of factors associated with email strain. In Pamela J. Hinds and David Martin, editors, *Proceedings of the 2006 ACM Conference on Computer Supported Cooperative Work, CSCW 2006, Banff, Alberta, Canada, November 4-8, 2006*, pages 431–440. ACM, 2006.

[17] Peter J. Denning. ACM President's letter: Electronic junk. *Communications of the ACM*, 25(3):163–165, 1982.

[18] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 29, New York, NY, USA, 2004. ACM.

[19] Harris Drucker, Christopher J. C. Burges, Linda Kaufman, Alex J. Smola, and Vladimir Vapnik. Support vector regression machines. In Michael Mozer, Michael I. Jordan, and Thomas Petsche, editors, *NIPS*, pages 155–161. MIT Press, 1996.

[20] Luiz H. Gomes, Fernando D. O. Castro, Virgílio A. F. Almeida, Jussara M. Almeida, Rodrigo B. Almeida, and Luis M. A. Bettencourt. Improving spam detection based on structural similarity. In *SRUTI'05: Proceedings of the Steps to Reducing Unwanted Traffic on the Internet on Steps to Reducing Unwanted Traffic on the Internet Workshop*, pages 12–12, Berkeley, CA, USA, 2005. USENIX Association.

[21] Joshua Goodman, Gordon V. Cormack, and David Heckerman. Spam and the ongoing battle for the inbox. *Commununications of the ACM*, 50(2):24–33, 2007.

[22] A. Gray and M. Haahr. Personalised, collaborative spam filtering. In *Proceedings of the 1st Conference on Email and Anti-Spam (CEAS)*. CEAS, 2004.

[23] Takaaki Hasegawa and Hisashi Ohara. Automatic priority assignment to E-mail messages based on information extraction and user's action history. In Rasiah Loganantharaj and Günther Palm, editors, *Intelligent Problem Solving, Methodologies and Approaches, 13th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE 2000, New Orleans, Louisiana, USA, June 19-22, 2000, Proceedings*, volume 1821 of *Lecture Notes in Computer Science*, pages 573–582. Springer, 2000.

[24] Taher Haveliwala, Sepandar Kamvar, and Glen Jeh. An analytical comparison of approaches to personalizing pagerank. Technical report, Stanford University, 2003.

[25] Eric Horvitz, Andy Jacobs, and David Hovel. Attention-sensitive alerting. In Kathryn B. Laskey and Henri Prade, editors, *UAI '99: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Stockholm, Sweden, July 30-August 1, 1999*, pages 305–313. Morgan Kaufmann, 1999.

[26] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13:415–425, 2002.

[27] Bryan Klimt and Yiming Yang. The Enron corpus: A new dataset for email classification research. In Jean-François Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi, editors, *Machine Learning: ECML 2004, 15th European Conference on Machine Learning, Pisa, Italy, September 20-24, 2004, Proceedings*, volume 3201 of *Lecture Notes in Computer Science*, pages 217–226. Springer, 2004.

[28] Fan Li and Yiming Yang. A loss function analysis for classification methods in text categorization. In *Proceedings of ICML-03, 20th International Conference on Machine Learning*, Washington, DC, 2003. Morgan Kaufmann Publishers, San Francisco, US.

[29] Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 140:1–55, 1932.

[30] Kevin Butler Lisa Johansen, Michael Rowell and Patrick McDaniel. Email communities of interest. In *Proceedings of the 4th Conference on Email and Anti-Spam (CEAS)*. CEAS, 2007.

[31] Lynam and Cormack. On-line spam filter fusion. In *SIGIR: Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006.

[32] Steve Martin, Blaine Nelson, Anil Sewani, Karl Chen, and Anthony D. Joseph. Analyzing behavioral features for email classification. In *CEAS* [1].

[33] Andrew McCallum, Xuerui Wang, and Andrés Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *J. Artif. Int. Res.*, 30(1):249–272, 2007.

[34] Carman Neustaedter, A. J. Bernheim Brush, Marc A. Smith, and Danyel Fisher. The social network and relationship finder: Social sorting for email triage. In *CEAS* [1].

[35] M. E. J. Newman. Modularity and community structure in networks. *Physical Sciences*, 2006.

[36] John C. Platt, Nello Cristianini, and Shawe J. Taylor. Large margin DAGs for multiclass classification. In Sara A. Solla, T. K. Leen, and K. R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 2000.

[37] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A Bayesian approach to filtering junk e-mail. In *AAAI-98 Workshop on Learning for Text Categorization*, pages 55–62, 1998.

[38] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. D. Spyropoulos, and P. Stamatopoulos. Stacking classifiers for anti-spam filtering of e-mail. In *Proceedings of "Empirical Methods in Natural Language Processing" (EMNLP 2001), L. Lee and D. Harman (Eds.), pp. 44-50, Carnegie Mellon University, Pittsburgh, PA, 2001*, pages 44–50, june 2001.

[39] Joshua R. Tyler, Dennis M. Wilkinson, and Bernardo A. Huberman. Email as spectroscopy: automated discovery of community structure within organizations. pages 81–96, 2003.

[40] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[41] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, 1994.

[42] Martin Wattenberg, Rohall, Steven L., Daniel Gruen, and Bernard Kerr. E-mail research: Targeting the enterprise. *Human-Computer Interaction*, 20(1/2):139–162, 2005.

[43] Gregory L. Wittel and S. Felix Wu. On attacking statistical spam filters. In *Proceedings of the First Conference on Email and Anti-Spam (CEAS)*, 2004.

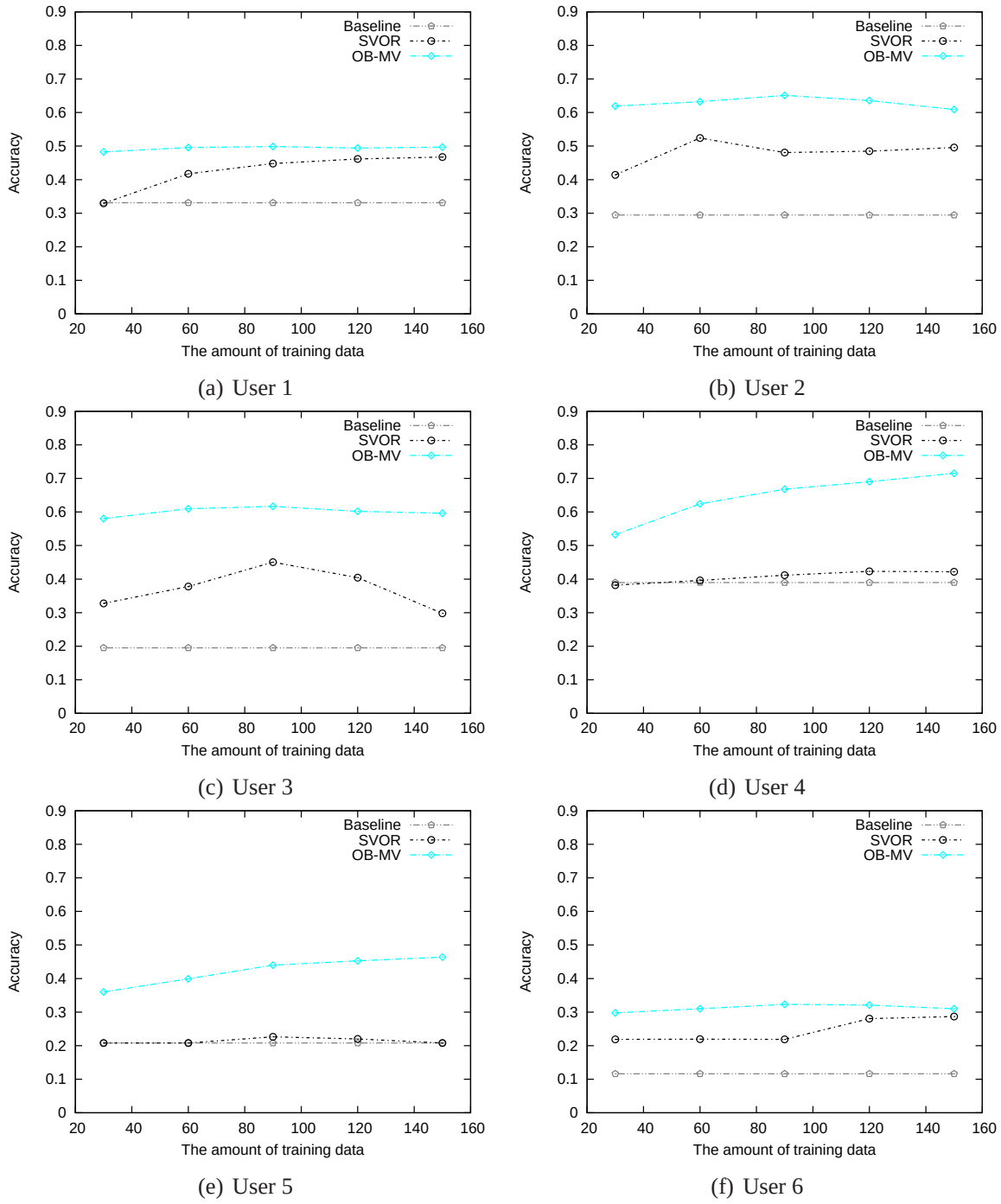
[44] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49, New York, NY, USA, 1999. ACM.

[45] Yiming Yang, Shinjae Yoo, Jian Zhang, and Bryan Kisiel. Robustness of adaptive filtering methods in a cross-benchmark evaluation. In Ricardo A. Baeza-Yates, Nivio Ziviani, Gary Marchionini, Alistair Moffat, and John Tait, editors, *SIGIR*, pages 98–105. ACM, 2005.

[46] Shinjae Yoo, Yiming Yang, Frank Lin, and Il-Chul Moon. Mining social networks for personalized email prioritization. In John F. Elder IV, Françoise Fogelman-Soulié, Peter A. Flach, and Mohammed Javeed Zaki, editors, *KDD*, pages 967–976. ACM, 2009.

[47] Le Zhang, Jingbo Zhu, and Tianshun Yao. An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing*, 3(4):243–269, December 2004.

A Additional Result Graphs and Tables

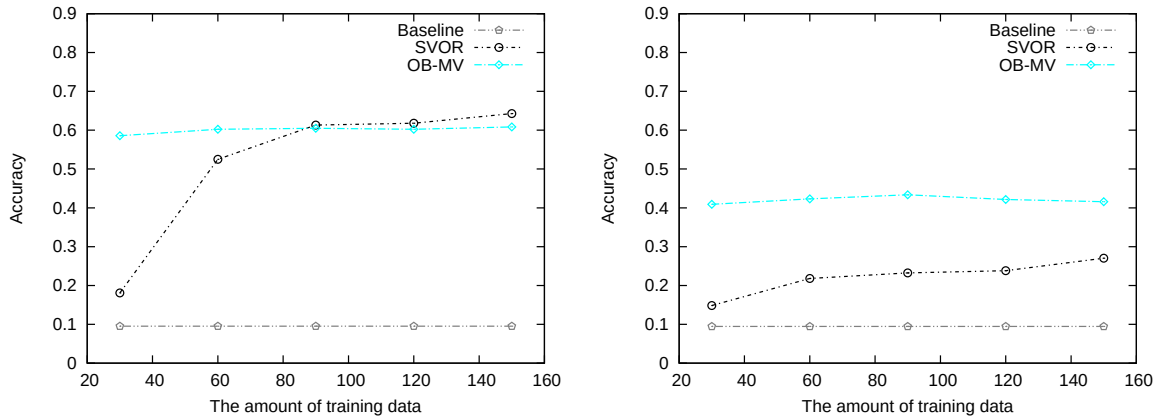


(a) User 1 (b) User 2

(c) User 3 (d) User 4

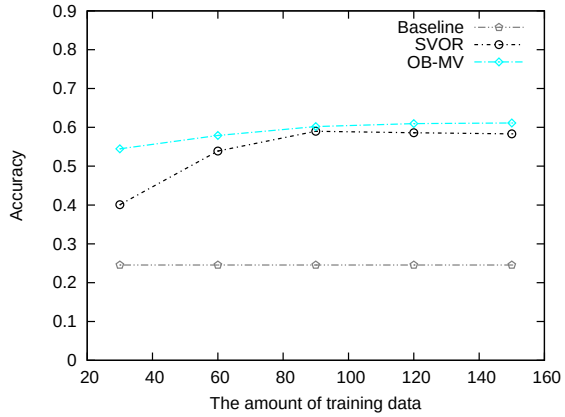
(e) User 5 (f) User 6

Figure A.1: Per-User Accuracy Learning Curves with Baseline, SVOR and OVA SVM (User 1-6)

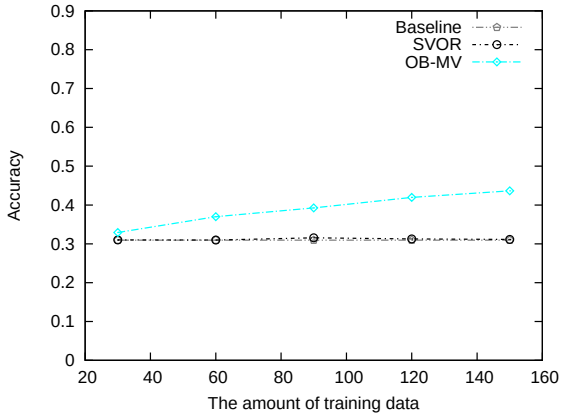


(a) User 7

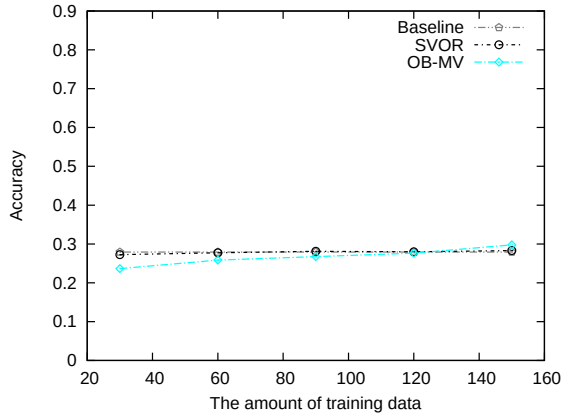
(b) User 8



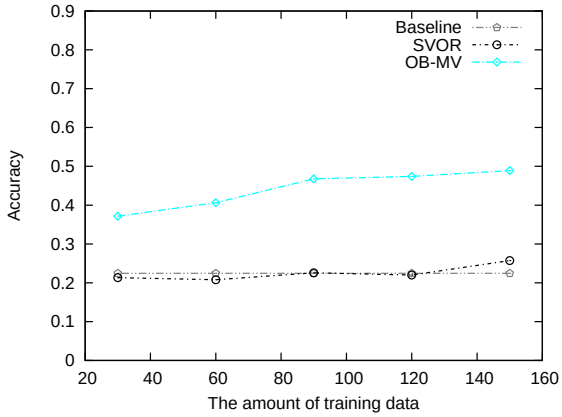
(c) User 9



(d) User 10

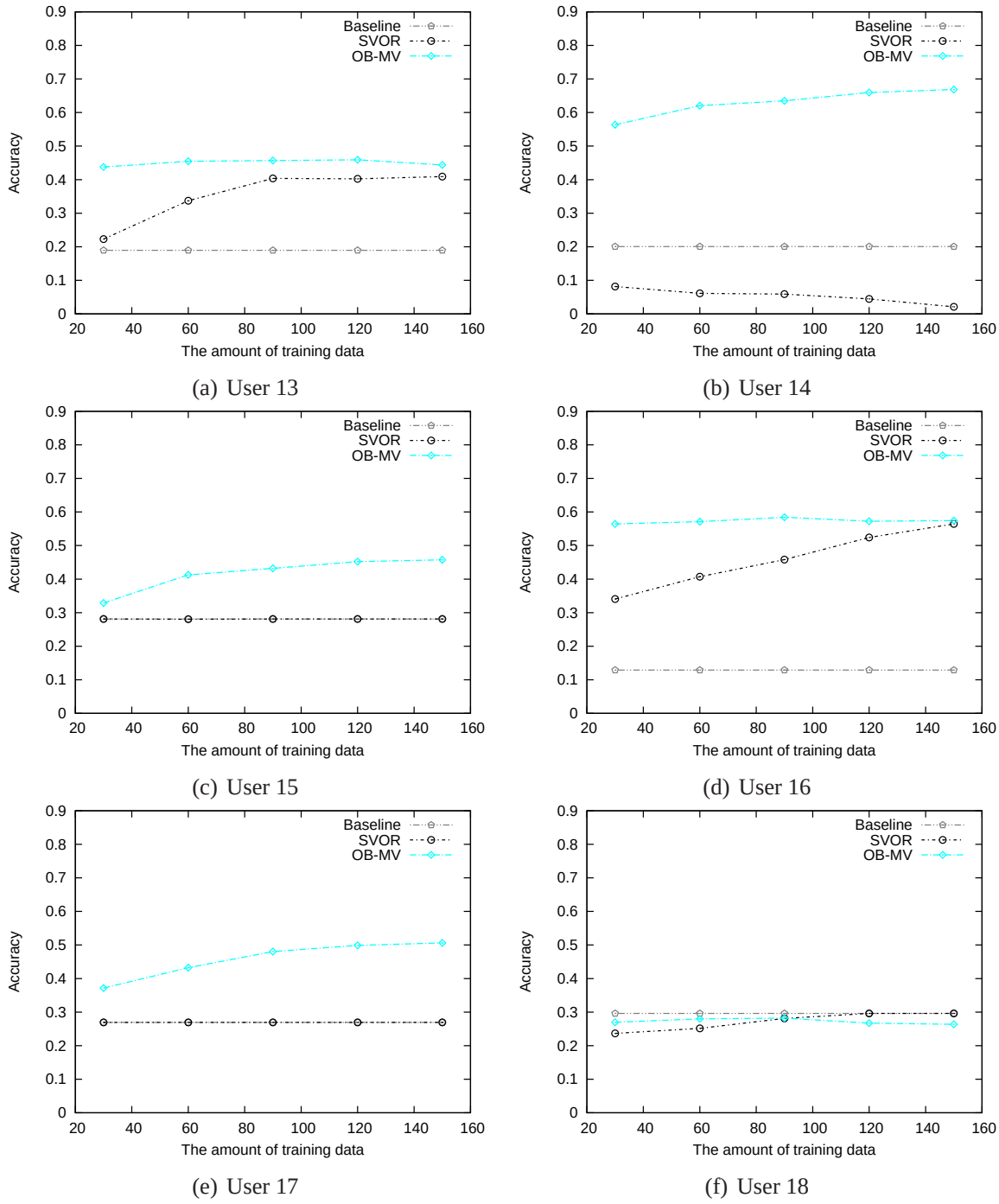


(e) User 11



(f) User 12

Figure A.2: Per-User Accuracy Learning Curves with Baseline, SVOR and OB-MV (User 7-12)

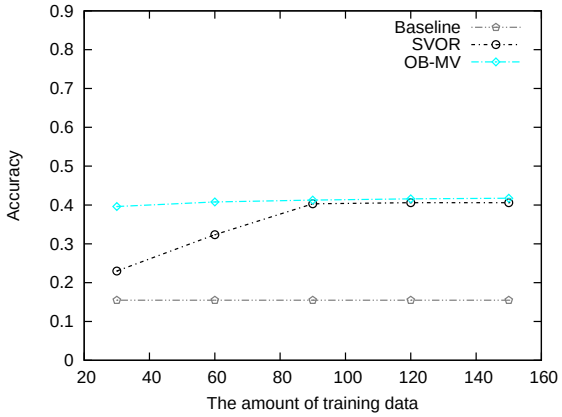


(a) User 13 (b) User 14

(c) User 15 (d) User 16

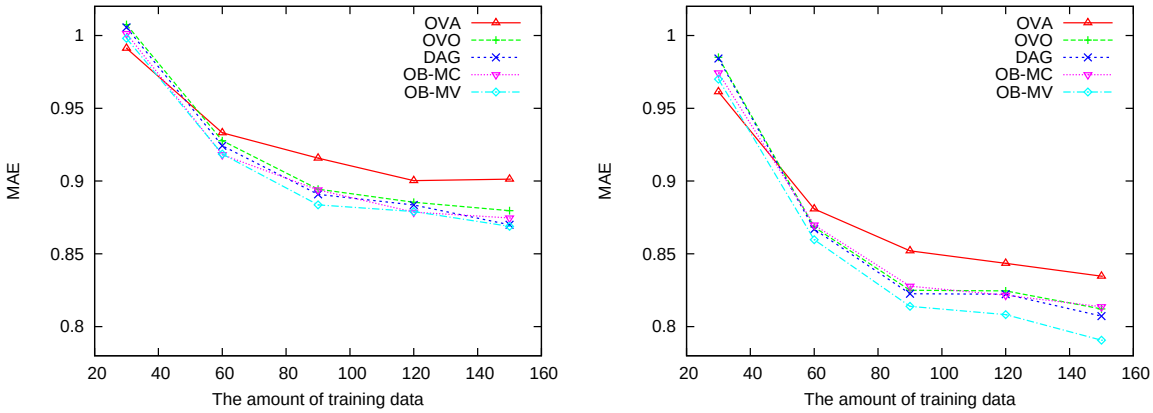
(e) User 17 (f) User 18

Figure A.3: Per-User Accuracy Learning Curves with Baseline, SVOR and OB-MV (User 13-18)



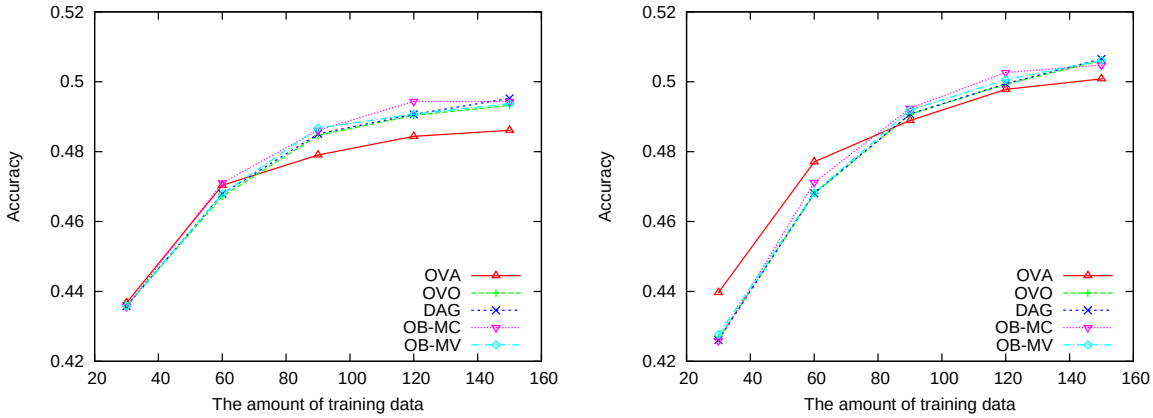
(a) User 19

Figure A.4: Per-User Accuracy Learning Curves with Baseline, SVOR and OB-MV (User 19)



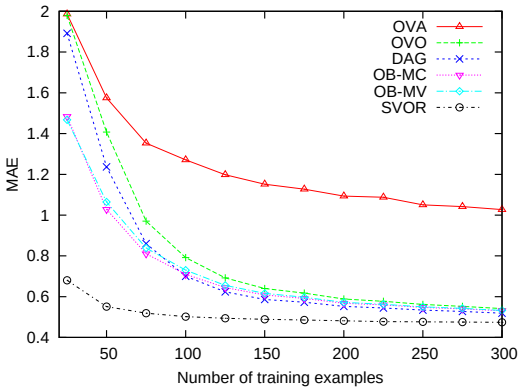
(a) Macro MAE (b) Micro MAE

Figure A.5: Comparisons among classification based approaches using MAE

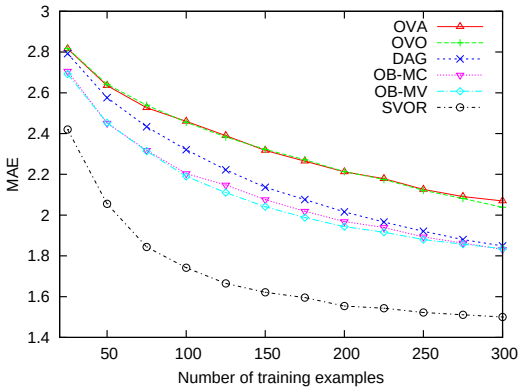


(a) Macro Accuracy (b) Micro Accuracy

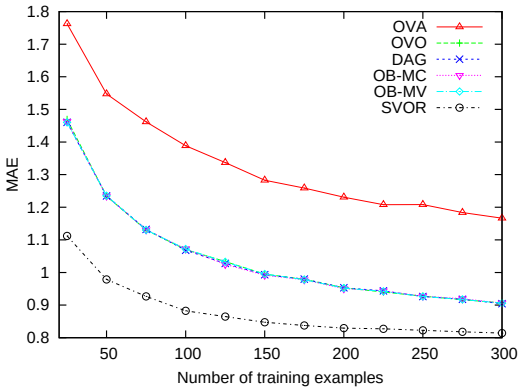
Figure A.6: Comparisons among classification based approaches using Accuracy



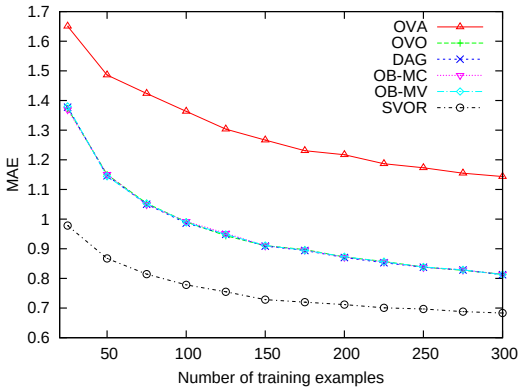
(a) Bank Domains(1)



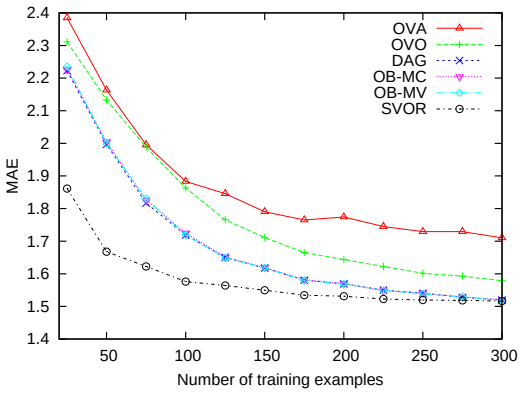
(b) Bank Domains(2)



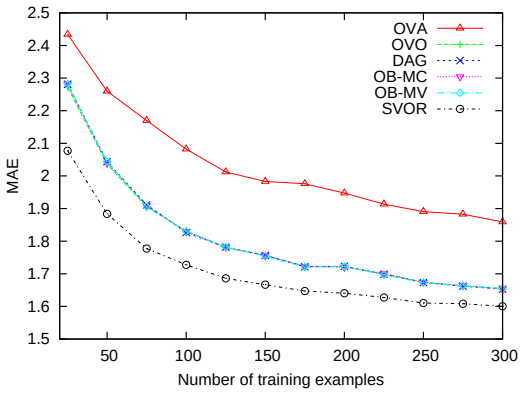
(c) Computer Activities(1)



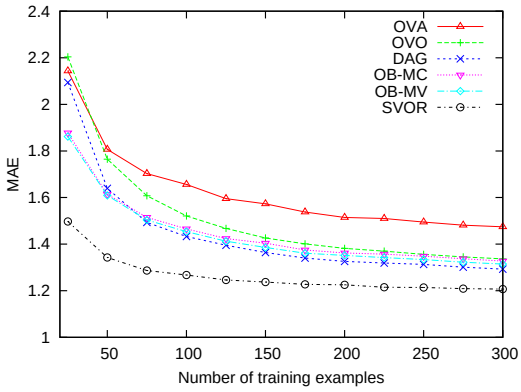
(d) Computer Activities(2)



(e) Census Domains(1)

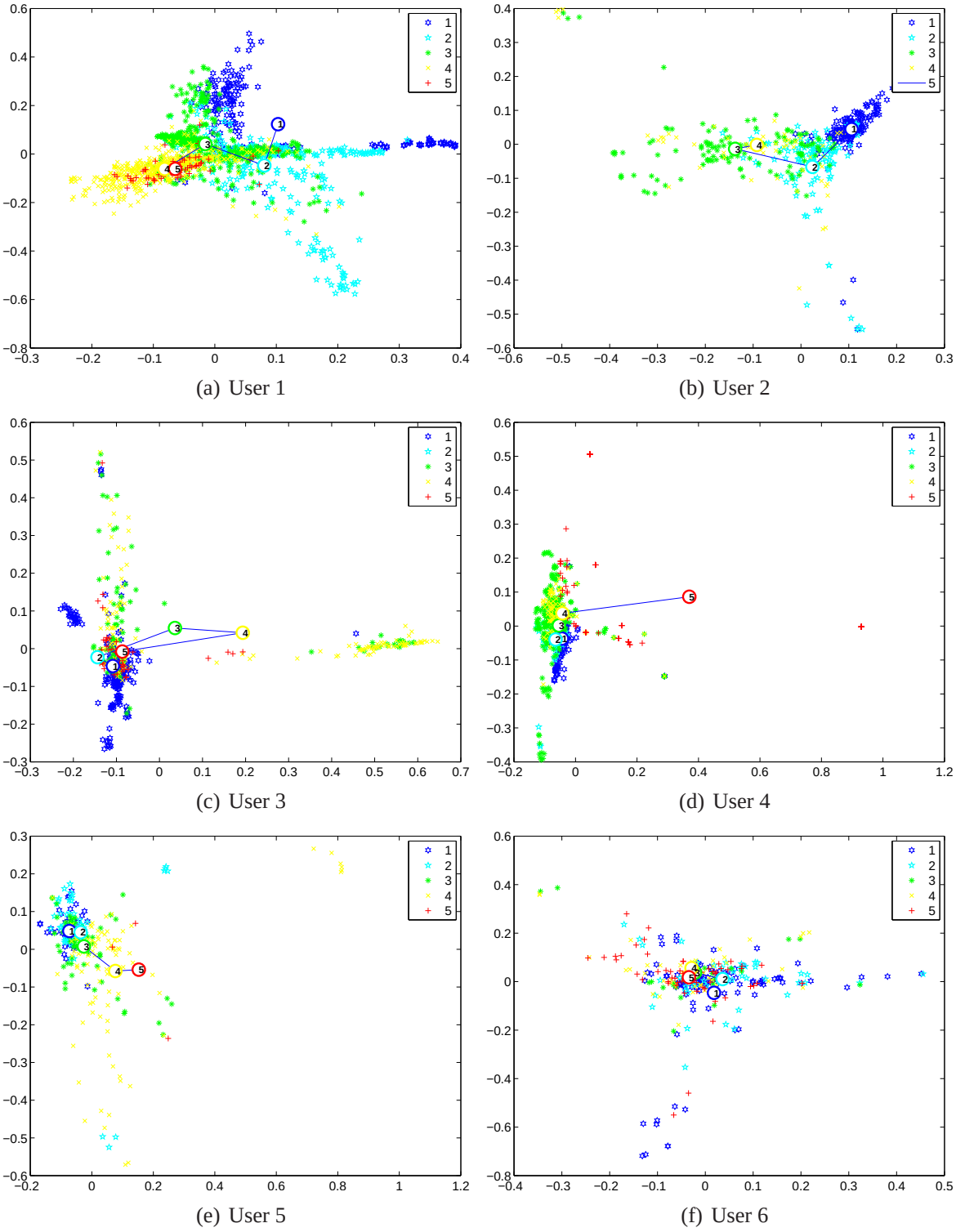


(f) Census Domains(2)



(g) California Housing

Figure A.7: UCI Dataset Results

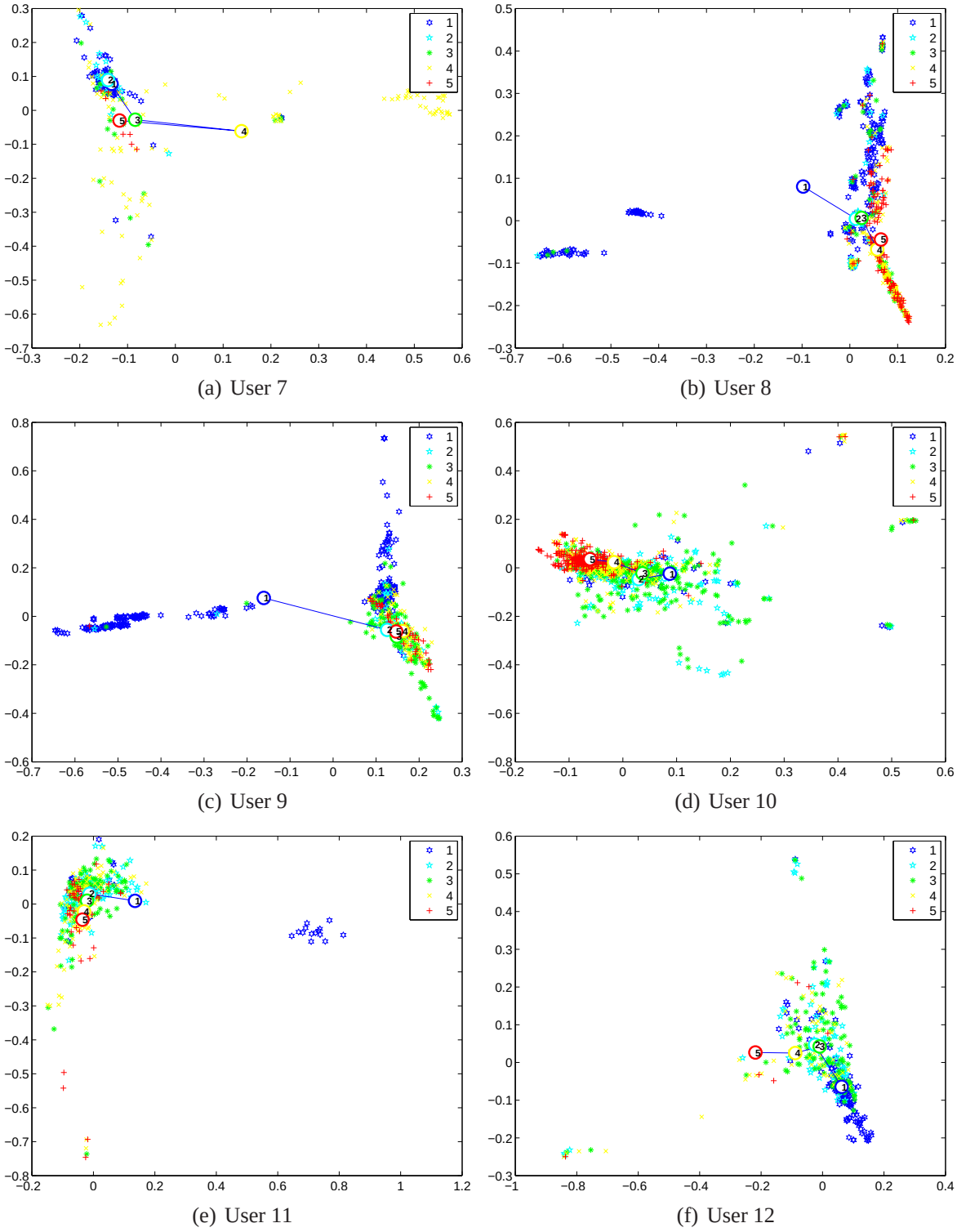


(a) User 1 (b) User 2

(c) User 3 (d) User 4

(e) User 5 (f) User 6

Figure A.8: Email Prioritization PCA Analysis (User 1 - 6)



(a) User 7

(b) User 8

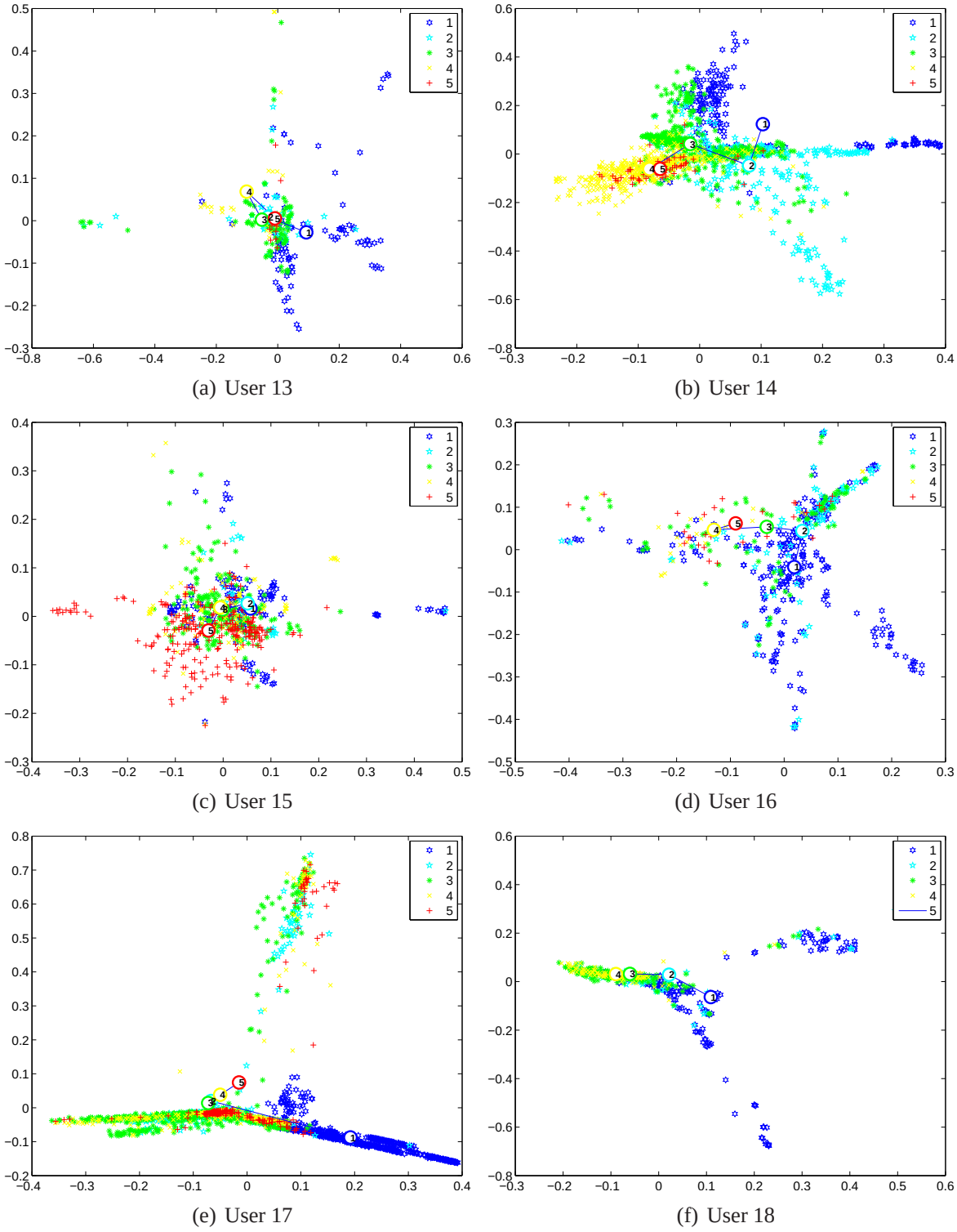
(c) User 9

(d) User 10

(e) User 11

(f) User 12

Figure A.9: Email Prioritization PCA Analysis (User 7 - 12)

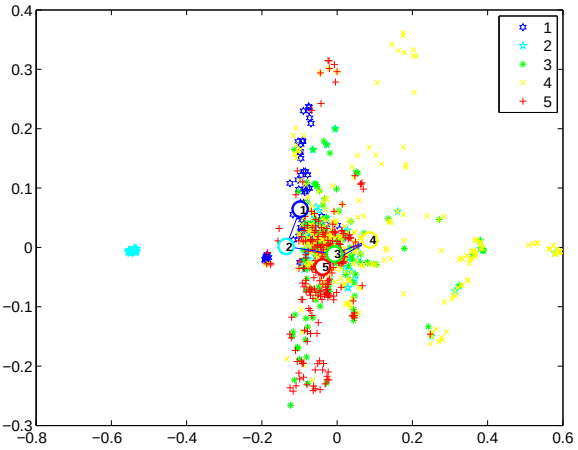


(a) User 13 (b) User 14

(c) User 15 (d) User 16

(e) User 17 (f) User 18

Figure A.10: Email Prioritization PCA Analysis (User 13 - 18)



(a) User 19

Figure A.11: Email Prioritization PCA Analysis (User 19)