Personalized Email Prioritization Based on Content and Social Network Analysis

Yiming Yang, Shinjae Yoo, and Frank Lin, *Carnegie Mellon University*

Il-Chul Moon, *Korea Advanced Institute of Science and Technology*

The proposed system combines unsupervised clustering, social network analysis, semisupervised feature induction, and supervised classification to model user priorities among incoming email messages.

Email is one of the most prevalent personal and business communication tools today, but it exhibits some significant drawbacks. Unlike telephone conversations or face-to-face meetings, email messages are received (after some spam filtering) in the same way regardless of a user's level of interest, and a single sender can flood multiple receivers. As a result, users must process a large volume of email messages of different importance levels.[1] Research recently estimated that businesses lose US$650 billion annually in productivity due to unnecessary email interruptions (http://www.forbes.com/2008/10/15/cio-email-manage-tech-cio-cx_rm_1015email.html). There is an urgent need to solve this information overload problem by developing systems that can learn personal priorities from data and identify important messages for each user.

Personalized email prioritization (PEP) has been underexplored. Unlike spam filtering, where people are less concerned with sharing individually labeled spam messages, PEP research looks at collecting nonspam email messages with personally assigned importance labels. Few people are willing to share their personal messages due to privacy concerns, however, and companies such as Google, Yahoo, and Microsoft, that have access to customers' email messages, cannot share private data with academic institutions for the same reason. Publicly available email data, such as the Enron corpus, are insufficient for training and testing of PEP systems because they lack personal importance judgments. This leaves researchers no choice but to collect private data under strict Institutional Review Board (IRB) guidelines. Such data-collection processes are costly, time consuming, and tedious, making it difficult to acquire a large number of users with diverse criteria in judging the importance of email messages.

This article presents the first study on PEP with a fully personalized methodology,[2] where only each user's personal email data (textual content and social network information) is available for the system during the system's training and testing. This is an important assumption for the generality of PEP methods—that is, we cannot rely on the availability of

1541-1672/10/$26.00 © 2010 IEEE
Published by the IEEE Computer Society

Related Work in Personalized Email Prioritization

Eric Horvitz and his colleagues built an email alerting system that used support vector machines to classify newly arrived email messages into two categories—that is, high or low in terms of utility.[1] However, their task did not consider personalization or investigate social network analysis.

Joshua Tyler and his colleagues used the Newman Clustering algorithm to discover social structures from email messages.[2] They found that the automatically discovered social structures (such as social leaders) are consistent with human interpretation of organizational structures. However, they did not focus on the email prioritization problem.

Carman Neustaedter and her colleagues defined metrics for measuring the social importance of individuals based on the From, To, and CC fields in email messages and recorded user actions in replying and reading email.[3] They used these metrics for retrieving old email messages rather than prioritization of new messages.

Lisa Johansen and her colleagues used social clustering to predict the importance of email messages.[4] The major difference between their method and ours is that their clusters were induced from a community social network, not based on personal social networks or the content information in email messages.

Lastly, Fei-Yue Wang and his colleagues discussed the theoretical, methodological, and technological underpinnings of social computing in general and reviewed the major application areas.[5]

With this article, we leverage the good ideas in these previous works and develop new techniques for personalized email prioritization.

References

1. E. Horvitz, A. Jacobs, and D. Hovel, "Attention-Sensitive Alerting," *Proc. Conf. Uncertainty and Artificial Intelligence*, Morgan Kaufmann, 1999, pp. 305–313.
2. J R. Tyler, D.M. Wilkinson, and B.A. Huberman, "Email as Spectroscopy: Automated Discovery of Community Structure within Organizations," *Communities and Technologies*, M. Huysman, E. Wenger, and V. Wulf, eds., Kluwer, 2003, pp. 81–96.
3. C. Neustaedter et al., "The Social Network and Relationship Finder: Social Sorting for Email Triage," *Proc. Conf. E-mail and Anti-Spam*, 2005; http://www.ceas.cc/2005/papers/149.pdf.
4. L. Johansen, M. Rowell, and P. McDaniel, "Email Communities of Interest," *Proc. 4th Conf. E-mail and Anti-Spam*, 2007; http://www.ceas.cc/2007/papers/paper-59.pdf.
5. F.Y. Wang et al., "Social Computing: From Social Informatics to Social Intelligence," *IEEE Intelligent Systems*, vol. 22, no. 2, 2007, pp. 79–83.

centralized access to customer private data in the development cycle or evaluation phase, and we cannot take the liberty of using a particular user's private data to build models for other users because of the potential leak of private information. Such strictly separate data makes our work fundamentally different from research in spam filtering and other previous work on email-based prediction. (See the "Related Work in Personalized Email Prioritization" sidebar for other approaches.)

We propose a novel approach that combines unsupervised clustering, social network analysis, semisupervised feature induction, and supervised classification to model user priorities among incoming email messages. We treat the priority prediction task as a supervised classification problem and use standard support vector machines (SVMs) as the classifiers. The novel part of our approach is the enriched representations of email messages and users, with automatically extracted features.

We constructed a data set of anonymized email messages with user-annotated importance levels (from 1 to 5) for this study. We use personal email data to induce such enriched features. A personal social network (PSN) is automatically constructed for each user based on the messages he or she receives. The PSN is a graph with nodes that represent email contacts (senders plus recipients in the CC lists) and links that indicate pairwise email interactions among the contacts. We constructed a PSN for two reasons:

- We do not want our method to rely on the unrealistic assumption that multiuser private data are always available for system development and model optimization.
- A PSN better represents a user's social activity than a global social network, which might include noisy features and de-emphasize personalization in the inductive learning of important features through the network.

By analyzing each user's PSN graph structure, our system can capture social groups of senders and recipients who have similar email interaction patterns or similar social roles and possibly share similar priority judgments over email messages. Our system can also propagate priority scores through a personal email network, from user-labeled messages (training instances) to other messages that do not have user-assigned importance scores.

Social Clustering

To predict the importance of email messages, the sender information would be highly informative. For example, we might have multiple project teams or social activity groups, and members in each group might naturally share corecipient lists and have similar judgments on message priority levels. Thus, capturing such groups would help us predict the importance of email message senders or recipients.

When we have a limited amount of training data, we will likely encounter

senders who have no labeled messages in the training set during the testing phase. If we can identify such users as members of groups based on unsupervised clustering, we can infer each user's priorities for messages from other group members. That is, we can cluster users based on their interaction patterns in a personal email data set. The cluster membership of the sender of each email message can be treated as the message's features (in addition to a standard bag-of-word representation) when inferring its importance. The importance of each sender group can be automatically learned by SVM classifiers.

We chose the Newman Clustering (NC) algorithm, which researchers have used to successfully find social structures in large organizations.[3] It defines the edge-betweenness (which we discuss in detail later) as a measure of the shortest path(s) going through a specific link among all-pairs shortest paths. A link with a high edge-betweenness score is crucial for connecting two highly connected component clusters. By deleting links with high edge-betweenness scores and removing those edges from the graph, we obtain disconnected component clusters.

One way to control the granularity level of clusters is to prespecify the number of desired clusters, which might be based on domain knowledge about the social networks in email or automatically determined by algorithms with a certain optimization criterion or heuristic measure. For example, the NC method can pick the number that yields the largest decrease in the sum of edge-betweenness per cluster.[4] We use this method in our work.

Unsupervised Learning of Social Importance Features

We measure the social importance levels of contacts without relying on the availability of labeled training data. We examine multiple graph-based metrics to characterize the social centrality of each contact in a PSN. Most of these metrics have been used in social network analysis (SNA) or link structure analysis but have not been studied in any depth with respect to PEP.

Let us define graph $G = (V, E)$ for a PSN, where vertices V correspond to the contacts and edges E reflect the email interactions: $E_{ij} = 1$ if there is (at least) one message from contact i to contact j; otherwise $E_{ij} = 0$.

We have defined seven metrics to describe email message features:

- in-degree centrality,
- out-degree centrality,
- total-degree centrality,
- clustering coefficient,
- clique count,
- betweenness centrality, and
- PageRank score.

In-degree centrality is a normalized measure for the in-degree of each contact (i):

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

where $|V|$ is the total number of contacts in the PSN. A high score indicates a popular receiver in the PSN.

Out-degree centrality is a normalized measure for the out-degree of each contact (i). It might imply some degree of importance, for example, as an announcement sender or a mailing-list organizer.

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

Total-degree centrality is a normalized measure for the number of unique senders and recipients who had links with node i. That is, it is the simple average of the node's in-degree and out-degree:

$$\text{TotalDegreeCent}(i) = \frac{1}{|V|}\sum_{j=1}^{|V|}\left[\frac{E_{ij} + E_{ji}}{2}\right]$$

The *clustering coefficient* measures the connectivity among the neighbors of node i:

$$\text{ClusterCoef}(i) = \frac{1}{Z}\sum_{j\in Nbr(i)} \sum_{k\in Nbr(i)\, j\neq k} E_{jk}$$

where $Nbr(i) - \{x : (E_{xi} \neq 0) \vee (E_{ix} \neq 0)\}$ is the node's neighborhood and $Z = |Nbr(i)| \cdot (|Nbr(i)| - 1)$ is the normalization denominator. Previous research used this metric to discriminate spam from nonspam email messages.[5]

A *clique* is generally defined as a fully connected subgraph in an undirected graph. The *clique count* of node i in our case is defined as

$$\text{ClqCnt}(i) = \sum_{c\in G} I(c, i) \times I\left(|c| \geq 3\right)$$

where G is a PSN graph, $c \in G$ is a clique, $I(c, i) \in \{0, 1\}$ is the binary indicator of whether clique c contains node i, and $I(|c| \geq 3) \in \{0, 1\}$ is a binary indicator of whether the size of clique c is at least three. This metric reflects the node's centrality in its local neighborhood, taking all the related nontrivial cliques (including the nested ones) into account. We follow the convention in clique-based social network analyses of ignoring cliques of size one or two.

The *betweenness centrality* is the percentage of shortest paths going through node i out of all possible paths. A high score in this measure means that the corresponding person

is a contact point between different social groups.

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

where σ_{jk} is the number of all-pairs shortest paths going through j and k (from j to k), and $\sigma_{jk}(i)$ is the number of all-pairs shortest paths going through j and k via i. This metric has been used in social network analysis.[3]

PageRank is a popular method in link-analysis research. We use it to induce a global measure of importance for email contacts. It is recursively defined, taking the transitivity of popularity into account. Let us use an N-by-N matrix $\boldsymbol{X}$ to represent email connections among N contacts in a personal email data set and define the matrix elements as

$$\mathrm{X}_{ij} = \frac{n_{ij}}{\sum_{j'=1}^{n} n_{ij'}}$$

where n_{ij} is the count of messages from i to j. Let $\boldsymbol{U}$ be a matrix with elements that have an identical score of $1/N$ and define a linear combination of $\boldsymbol{X}$ and $\boldsymbol{U}$ as $\boldsymbol{E} = (1 - a)\ \boldsymbol{X} + a\boldsymbol{U})^T$ $0 < a < 1$.

Use an $N \times 1$ vector $\mathbf{r}$ (the PageRank vector) to store the importance scores of the N contacts, and set the initial values of its elements to be $1/N$. Then update this vector iteratively: $\mathbf{r}^{(k+1)} = \boldsymbol{E}\mathbf{r}^{(k)}$. The vector converges to the principal eigenvector of matrix $\boldsymbol{E}$ when k is sufficiently large. The stationary vector contains one PageRank score per contact in a personal email data set.

We call all these metrics the social importance (SI) features of email messages. That is, we represent the sender of each message in a personal email data set using the automatically extracted SI features, in addition to the sender ID. The enriched sender representation is a part of the message representation. These features (together with other message features) are weighted by SVM classifiers, based on how informative they are in making priority predictions.

Semisupervised Learning of Social Importance Features

Semisupervised SI features are those we induce based on both the user-assigned importance labels (in five levels) of training instances (messages) and the graphical structure of email interactions in a personal email data set. Typically, only a small subset of the messages has importance labels. We propose the Level-Sensitive PageRank (LSPR) approach to propagate labeled importance of the training examples to other messages and connected users.

> Our method leverages the frequencies and importance of messages, while conventional link-analysis methods use only one type of directed link.

We define $\boldsymbol{V}$ as an N-by-5 matrix, where rows represent users (indexed by $i = 1, 2, \ldots, N$), columns are for importance levels (labeled as $k = 1, 2, 3, 4, 5$), and each cell is the number of messages received by a user and at the corresponding level. We further normalize each column vector of the matrix using the sum of all elements in each column as the denominator to normalize each column element. The normalized column vector $\mathbf{v}_k$ shows the proportions of the labels at level k over users. Vector $\mathbf{v}_k$ is sparse when the user only labels a few instances at level k in the training set.

Treating $\mathbf{v}_k$ as the initial label distribution at level k over all users and assuming labels are transitive from user to user through their email connections, we define the iterative update of an LSPR vector as

$$\mathbf{p}_k^{(t+1)} = (1-a)\mathbf{X}^T\mathbf{p}_k^{(t)} + a\mathbf{p}_k^{(1)} \quad (1)$$

In the first term in the formula, matrix $\boldsymbol{X}$ is the same as we defined earlier for PageRank. It represents the transitional probabilities among users based on unlabeled email interactions. The second term in the formula represents the supervised label bias over users. Constant $a \in [1, 0]$ controls the balance between the two terms in the iterative updating of the LSPR vector. The vector converges to the principal eigenvector of matrix $E_k = (1 - \alpha)\boldsymbol{X}^T + \alpha\mathbf{v}_k 1^T$ when t is sufficiently large.[6] The stationary LSPR vector is denoted as $\mathbf{p}_k$, with elements that sum to one, representing the expected proportion for each node to have the importance labels at level k.

Applying this calculation to importance level $k = 1, 2, 3, 4$, and 5, we obtain five stationary vectors in matrix $\boldsymbol{P} = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5)$. The row vectors of matrix $\boldsymbol{P}$ provide a 5D representation. We use the LSPR row vectors as additional features to represent each message, as the semisupervised LSPR features of its sender. The elements in matrix $\boldsymbol{P}$ 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 (such as term weights and the values of unsupervised SI features) in the enriched vector representation of email messages, we renormalize each LSPR subvector (5D) into a unit vector. That is, we use the sum of the five elements as the denominator of each element in the normalization.

Our formulae for LSPR are algorithmically 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 over webpages. In fact, the LSPR method is inspired by the TSPR and PPR work. However, in our method, the graph structure is constructed using two types of objects (people and messages), whereas the graph structures in TSPR and PPR (and in PageRank) have only one type of node (webpages). Our method also leverages both the frequencies and importance of messages, while conventional link-analysis methods use only one type of directed link. More importantly, we focus on effectively using a partially labeled personal email network and assume the transitivity of importance among users is sensitive to the importance levels of messages exchanged among these users.

Experiments

We recruited a set of subjects, mostly from the Language Technologies Institute at Carnegie Mellon University, including faculty members, staff, and graduate students. Each subject was asked to label at least 400 nonspam messages during a one-month period using a five-level scale. Only seven users actually labeled more than 200 messages, which we used as the collected data for our experiments.

In each personal data collection, we sorted the email messages temporally and split the sorted list into 70 and 30 percent portions. We used the 70 percent portion for training and parameter tuning and the remaining 30 percent for testing. The full set of training examples was used to induce the NC and SI features. For LSPR, we used all the messages in the training set to propagate 30, 60, 90, 120, and 150 labels in the training set, respectively. The average number of training messages per user was 395 (with the maximum of 1,225 and the minimum of 164); the average number of test messages per user was 169 (with the maximum of 525 and the minimum of 70).

Preprocessing

We applied a multipass preprocessing to the email messages. First, we applied email address canonicalization. Because each person might 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 that might 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.

Features

The basic features (BF) are the tokens in the From, To, CC, Title, and Body Text sections in email messages. We used a vector to represent those features for each email message with a dimension $\mathbf{v}$, the vocabulary size, which we call the BF subvector.

We used an m-dimensional subvector to represent the NC features for each email message's sender, where m is the number of clusters produced by the clustering algorithm based on the user's personal social network. An element of the subvector is 1 if the user belongs to the corresponding cluster and 0 otherwise; each user can belong to only one cluster. If the sender of a message in the test set is not in the training set, he or she is assigned to a default cluster. We calculated the sum of the importance values of messages in each cluster and used it as the cluster's importance value. The cluster with the median importance value is the default cluster.

We also used another 7D subvector to represent the SI features per user, with real-valued elements, and a 5D subvector to represent each user's LSPR features, with elements that are the mixture weights of the user at the five importance levels. If the sender of a message in the test set was not in the training set, the LSPR subvector of this message was assigned to the mean of LSPR vectors by default.

The concatenation of all these subvectors yields a synthetic vector per email message as its full representation.

Classifiers

We used five linear SVM classifiers to predict the importance level per email message. Each classifier takes each message's vector representation as its input and produces a score with respect to a specific importance level. 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 standard SVM^{light} software package (http://svmlight.joachims.org).

We ran the SVM classifiers with messages represented using the BFs

only as the performance baseline. We also ran the SVM classifiers with additional features, including the unsupervised SI features, the NC features, and the semisupervised LSPR features. We named the baseline system SVM.BF and the system using the combination of all the feature types SVM.BF+. We varied the number of labeled messages used in training the SVM classifiers from 30 to 150 labeled messages per user and measured the system performance under these conditions. All the training-set sizes are relatively small, compared to large data collections used in benchmark evaluations for text categorization—for example, the RCV1 news story collection has 780,000 training examples for 103 categories. This is part of the difficulty we must deal with for PEP.

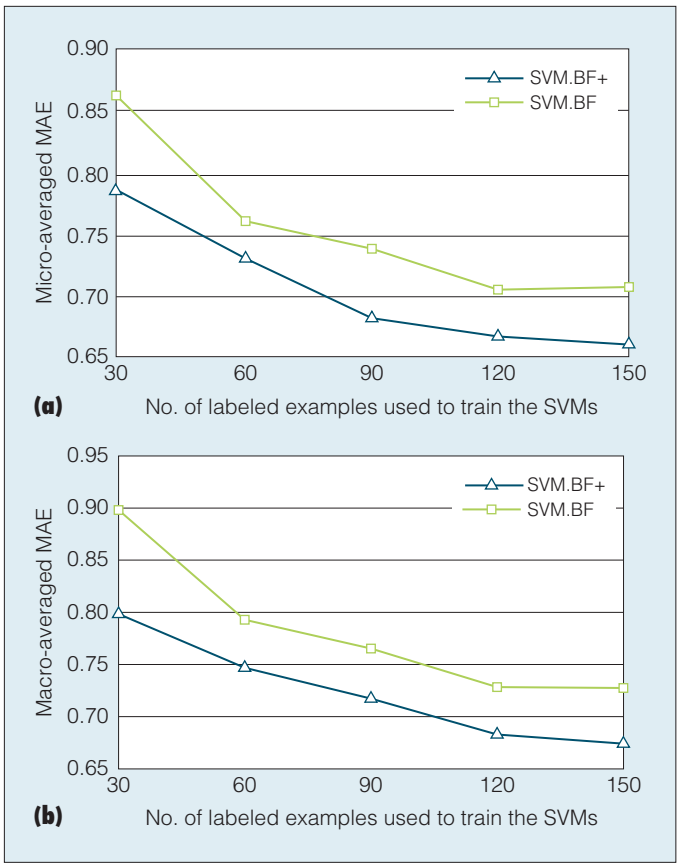

Figure 1. Performance of support vector machines (SVMs) in (a) micro-averaged mean absolute error (MAE) and (b) macro-averaged MAE. The MAE ranges from 0 to 4, where a lower value means better performance. Results from the baseline system (SVM.BF) and the system using the combination of all the feature types (SVM.BF+) strongly support the advantage of leveraging the social-network features in combination with content-based features over the baseline approach.

Metrics

We used mean absolute error (MAE) as the main evaluation metric, which is standard in evaluating systems that produce multilevel discrete predictions. MAE is defined as

$$\text{MAE} = 1/N \sum_{i=1}^{N} |y_i - \hat{y}_i|$$

where N is the number of messages in the test set, y_i is the true importance level of message i, and $\hat{y}_i$ is the predicted importance level for that message. Because we have five levels of importance, the MAE scores range from 0 (best) to 4 (worst).

There are two conventional ways to compute the performance average over multiple users. The first, *micro-averaged MAE*, involves pooling the test instances from all users to obtain a joint test set and computing the MAE on the pool. The other way, *macro-averaged MAE*, is to compute the MAE on the test instances of each user and then take the average of the per-user MAE values. 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. Both methods can be informative, so we present the evaluation results in both variants of the metric.

Results

Figure 1 shows the performance of SVM.BF and SVM.BF+ conditioned on varying training-set sizes of 30 to 150 labeled messages. Adding the social-network based features (SI, NC, and LSPR) significantly reduced the importance prediction errors in both micro- and macro-averaged MAE. We conducted Wilcoxon signed-rank tests to compare the results of SVMs using only BF features versus using the additional features. The p-values in these conditions are below 1 percent except in one case, when the training-set size is 60 and the p-value is 5 percent. These results strongly support the advantage of leveraging the social-network features in combination with content-based features over the baseline approach.

Parameter Tuning

We tuned two parameters per user on held-out validation data: the margin parameter C in SVM, which controls the balance between training-set errors and model complexity, and the parameter a in LSPR, which balances the two terms in Equation 1. We split each user's training set 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, or running LSPR with a specific value of a.

We repeated this procedure on 10 validation subsets, with the C values

THE AUTHORS

Yiming Yang is a professor in the Language Technologies Institute and the Machine Learning Department in the School of Computer Science at Carnegie Mellon University (CMU). Her research centers on statistical learning methods for a range of problems, including large-scale text categorization, relevance- and novelty-based retrieval and adaptive filtering, personalization and active learning for recommendation systems, and personalized email prioritization. Yang has a PhD in computer science from Kyoto University. Contact her at yiming@cs.cmu.edu.

Shinjae Yoo is a research associate at the Brookhaven National Laboratory. His current research interests include statistical learning approaches to personalized email prioritization, text mining, and heterogeneous network analysis. Yoo has a PhD in language technologies from the School of Computer Science at Carnegie Mellon University. Contact him at shinjae@gmail.com.

Frank Lin is a PhD student in the Language Technologies Institute at CMU. His current research interests include graph-based clustering and semisupervised learning and how these methods can be efficiently applied to general large-scale data. Lin has an MS in language technologies from the School of Computer Science at Carnegie Mellon University. Contact him at frank@cs.cmu.edu.

Il-Chul Moon is a postdoctoral researcher in the Department of Electrical Engineering at the Korea Advanced Institute of Science and Technology. His research interests include social-network analysis, agent-based simulation and counterterrorism, defense modeling, and simulation. Moon has a PhD in computation, organization, and society from Carnegie Mellon University. Contact him at icmoon@smslab.kaist.ac.kr.

in the range from 10^{-3} to 10^{3}, and the values in the range from 0.05 to 0.25. The value of each parameter that 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 $a \in [0.05, 0.25]$ and $C \in [1, 1{,}000]$.

Computational Efficiency

The computational cost consists of several parts:

1. unsupervised NC clustering and SI-feature induction,
2. semi-supervised induction of LSPR features,
3. supervised training of SVM classifiers (5 per user), and
4. online construction of NC, SI, and LSPR features for new senders in the test set but not in the training set, and priority prediction on test messages.

Parts 1 through 3 belong to the offline training and validation phase, and part 4 belongs to the online testing phase performed for each instance. We measured the CPU time on an Intel Xeon 3.16-GHz processor in training and testing over the data set of one user (who has the largest data set). Part 1 took 12 seconds, part 2 took 6.7 seconds, and parts 3 and 4 took under a second each.

Because the data sets were relatively small, computational cost was not an issue in our experiments. In future applications of our method, the training data from some users could grow much larger; in that case, sampling from the available training data is a potential solution for efficient computation. For example, we could use the most recent few hundred (or thousands) of messages for updating the features and classifiers periodically offline (once a day or once a week as needed).

Our experiments demonstrate the effectiveness of our proposed approach on personal email data from multiple users. Future work would include collecting more data and comparative studies on different clustering, graph mining, and classification algorithms with respect to PEP.

Acknowledgments

This work is supported, in part, by DARPA under contract NBCHD030010; the US National Science Foundation (NSF) under grant IIS_0704689; and the Brain Korea 21 Project, the School of Information Technology, KAIST. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors. This article is an extended version of an earlier report published in ACM SIGKDD 2009.[2]

References

1. L.A. Dabbish and R.E. Kraut, "Email Overload at Work: An Analysis of Factors Associated with Email Strain," *Proc. 20th Anniversary Conf. Computer Supported Cooperative Work*, ACM Press, 2006, pp. 431–440.
2. S. Yoo et al., "Mining Social Networks for Personalized Email Prioritization," *Proc. 15th ACM SIGKDD Conf. Knowledge Discovery and Data Mining*, ACM Press, 2009, pp. 967–976.
3. J.R. Tyler, D.M. Wilkinson, and B.A. Huberman, "Email as Spectroscopy: Automated Discovery of Community Structure within Organizations," *Communities and Technologies*, M. Huysman, E. Wenger, and V. Wulf, eds., Kluwer, 2003, pp. 81–96.
4. A. Clauset, M.E.J. Newman, and C. Moore, "Finding Community Structure in Very Large Networks," *Physical Rev. E*, vol. 70, no. 6, 2004, pp. 066111-1–066111-6.
5. P.O. Boykin and V.P. Roychowdhury, "Leveraging Social Networks to Fight Spam," *Computer*, vol. 38, no. 4, 2005, pp. 61–68.
6. T. Haveliwala, S. Kamvar, and G. Jeh, *An Analytical Comparison of Approaches to Personalizing Pagerank*, tech. report, Stanford Univ., 2003.