Mining E-Mail Data

Steffen Bickel
Humboldt-Universität zu Berlin, Germany

Tobias Scheffer
Humboldt-Universität zu Berlin, Germany

INTRODUCTION

E-mail has become one of the most important communication media for business and private purposes. Large amounts of past e-mail records reside on corporate servers and desktop clients. There is a huge potential for mining this data. E-mail filing and spam filtering are well-established e-mail mining tasks. E-mail filing addresses the assignment of incoming e-mails to predefined categories to support selective reading and organize large e-mail collections. First research on e-mail filing was conducted by Green and Edwards (1996) and Cohen (1996). Pantel and Lin (1998) and Sahami, Dumais, Heckerman, and Horvitz (1998) first published work on spam filtering. Here, the goal is to filter unsolicited messages. Recent research on e-mail mining addresses automatic e-mail answering (Bickel & Scheffer, 2004) and mining social networks from e-mail logs (Tyler, Wilkinson, & Huberman, 2004).

In Section *Background* we will categorize common e-mail mining tasks according to their objective, and give an overview of the research literature. Our *Main Thrust* Section addresses e-mail mining with the objective of supporting the message creation process. Finally, we discuss *Future Trends* and conclude.

BACKGROUND

There are two objectives for mining e-mail data: supporting communication and discovering hidden properties of communication networks.

Support of Communication

The problems of filing e-mails and filtering spam are text classification problems. Text classification is a well studied research area; a wide range of different methods is available. Most of the common text classification algorithms have been applied to the problem of e-mail classification and their performance has been compared in several studies. Because publishing an e-mail data set involves disclosure of private e-mails, there are only a small number of standard e-mail classification data sets. Since there is no study that compares large numbers of data sets, different classifiers and different types of extracted features, it is difficult to judge which text classifier performs best specifically for e-mail classification.

Against this background we try to draw some conclusions on the question which is the best text classifier for e-mail. Cohen (1996) applies rule induction to the e-mail classification problem and Provost (1999) finds that Naïve Bayes outperforms rule induction for e-mail filing. Naïve Bayes classifiers are widely used for e-mail classification because of their simple implementation and low computation time (Pantel & Lin, 1998; Rennie, 2000; Sahami, Dumais, Heckerman, & Horvitz, 1998). Joachims (1997, 1998) shows that Support Vector Machines (SVMs) are superior to the Rocchio classifier and Naïve Bayes for many text classification problems. Drucker, Wu, and Vapnik (1999) compares SVM with boosting on decision trees. SVM and boosting show similar performance but SVM proves to be much faster and has a preferable distribution of errors.

The performance of an e-mail classifier is dependent on the extraction of appropriate features. Joachims (1998) shows that applying feature selection for text classification with SVM does not improve performance. Hence, using SVM one can bypass the expensive feature selection process and simply include all available features. Features that are typically used for e-mail classification include all tokens in the e-mail body and header in bag-of-words representation using TF- or TFIDF-weighting. HTML tags and single URL elements also provide useful information (Graham, 2003).

Boykin and Roychowdhury (2004) propose a spam filtering method that is not based on text classification but on graph properties of message sub-graphs. All addresses that appear in the headers of the inbound mails are graph nodes; an edge is added between all pairs of addresses that jointly appear in at least one header. The resulting sub-graphs exhibit graph properties that differ significantly for spam and non-spam sub-graphs. Based on this finding "black-" and "whitelists" can be constructed for spam and non-spam addresses. While this idea is appealing, it should be noted that the approach is not immediately practical since most headers of spam e-mails do not

contain other spam recipients' addresses, and most senders' addresses are used only once.

Additionally, the "*semantic e-mail*" approach (McDowell, Etzioni, Halevy, & Levy, 2004) aims at supporting communication by allowing automatic e-mail processing and facilitating e-mail mining; it is the equivalent of *semantic web* for e-mail. The goal is to make e-mails human- and machine-understandable with a standardized set of e-mail processes. Each e-mail has to follow a standardized process definition that includes specific process relevant information. An example for a *semantic e-mail* process is meeting coordination. Here, the individual process tasks (corresponding to single e-mails) are issuing invitations and collecting responses. In order to work, *semantic e-mail* would require a global agreement on standardized semantic processes, special e-mail clients and training for all users. Additional mining tasks for support of communication are automatic e-mail answering and sentence completion. They are described in Section *Main Thrust*.

Discovering Hidden Properties of Communication Networks

E-mail communication patterns reveal much information about hidden social relationships within organizations. Conclusions about informal communities and informal leadership can be drawn from e-mail graphs. Differences between informal and formal structures in business organizations can provide clues for improvement of formal structures which may lead to enhanced productivity. In the case of terrorist networks, the identification of communities and potential leaders is obviously helpful as well. Additional potential applications lie in marketing, where companies – especially communication providers – can target communities as a whole.

In social science, it is common practice for studies on electronic communication within organizations to derive the network structure by means of personal interviews or surveys (Garton Garton, Haythornthwaite, & Wellman, 1997; Hinds & Kiesler, 1995). For large organizations, this is not feasible. Building communication graphs from e-mail logs is a very simple and accurate alternative provided that the data is available. Tyler, Wilkinson, and Huberman (2004) derive a network structure from e-mail logs and apply a divisive clustering algorithm that decomposes the graph into communities. Tyler, Wilkinson, and Huberman verify the resulting communities by interviewing the communication participants; they find that the derived communities correspond to informal communities.

Tyler et al. also apply a force-directed spring algorithm (Fruchterman & Rheingold, 1991) to identify leadership hierarchies. They find that with increasing distance of vertices from the "spring" (center) there is a tendency of decreasing real hierarchy depth.

E-mail graphs can also be used for controlling virus attacks. Ebel, Mielsch, and Bornholdt (2002) show that vertex degrees of e-mail graphs are governed by power laws. By equipping the small number of highly connected nodes with anti-virus software the spreading of viruses can be prevented easily.

MAIN THRUST

In the last section we categorized e-mail mining tasks regarding their objective and gave a short explanation on the single tasks. We will now focus on the ones that we consider to be most interesting and potentially most beneficial for users and describe them in greater detail. These tasks aim at supporting the message creation process. Many e-mail management systems allow the definition of message templates that simplify the message creation for recurring topics. This is a first step towards supporting the message creation process, but past e-mails that are available for mining are disregarded. We describe two approaches for supporting the message creation process by mining historic data: mining question-answer pairs and mining sentences.

Mining Question-Answer Pairs

We consider the problem of learning to answer incoming e-mails from records of past communication. We focus on environments in which large amounts of similar answers to frequently asked questions are sent – such as call centers or customer support departments. In these environments, it is possible to *manually* identify equivalence classes of answers in the records of *outbound* communication. Each class then corresponds to a set of semantically equivalent answers sent in the past; it depends strongly on the application context which fraction of the outbound communication falls into such classes. Mapping *inbound* messages to one of the equivalence classes of answers is now a multi-class text classification problem that can be solved with text classifiers.

This procedure requires a user to manually group previously sent answers into equivalence classes which can then serve as class labels for training a classifier. This substantial manual labeling effort reduces the benefit of the approach. Even though it can be reduced by employing semi-supervised learning (Nigam, McCallum, Thrun, & Mitchell, 2000; Scheffer, 2004), it would still be much preferable to learn from only the available data: stored inbound and outbound messages. Bickel and Scheffer (2004) discuss an algorithm that learns to answer ques-