Inferring Ongoing Activities of Workstation Users by Clustering Email

Yifen Huang, Dinesh Govindaraju, Tom Mitchell
School of Computer Science
Carnegie Mellon University

1. The Problem

We are interested in automatically discovering the key ongoing activities of a workstation user, such as committees to which she belongs, writing projects in which she is involved, etc., based on the contents of her workstation. The thesis underlying our research is that this collection of user activities can be automatically inferred from the variety of data available on most users' workstations, including their emails, files, online calendar, and history of web page accesses. If such activities could be inferred, this knowledge about the user's activities could be used in a variety of ways to support the user. For example, it could be used to cross-index email, calendar events, files, and web accesses according to activity, or to produce a 'briefing folder' for each meeting on the user's calendar (i.e., a folder containing emails, files, etc., relevant to the activity associated with this meeting).

In this short paper we describe our initial research on inferring such activities by examining only the user's email. In particular, we describe here a variety of unsupervised clustering methods designed specifically for emails, and present experimental results showing that these algorithms often produce clusters of emails aligned with distinct user activities. Also, we present a variety of representations to summarize a cluster, which is targeted to interpret the meanings of a given cluster to the user.

2. Clustering Approach

Designing an algorithm to cluster user emails involves several design choices. First, we must represent emails in some way compatible with the clustering algorithm. In our case, we choose to represent each email in terms of both header and body features. The header features (we will refer to these as the H features) include the subject line, email addresses other than the recipient, domains of these email addresses, and words judged to be proper nouns. The subject line in H is a single string produced after removing modifiers (e.g., 'Re:', 'Fw:', 'Fwd:', numerals) and ensuring the entire string is of some minimum length. The body features, which we refer to as B, include the bag of words found in the email body.

Second, we must choose a clustering algorithm to operate on the emails represented in this form. There are several dimensions along which the clustering algorithm can vary, including (a) whether it is an agglomerative, bottom-up algorithm that joins documents, or a top-down algorithm that clusters all documents simultaneously, (b) the initialization of clusters for the iterative clustering algorithm, and (c) the way in which the algorithm combines the header and body features. Here we explore two specific classes of algorithms:

1. *EM-based mixture of multinomials algorithms*. (EM) This clustering algorithm represents each email as a feature vector potentially including both header and body features. Clustering is viewed as a problem of identifying the mixture components in a mixture distribution assumed to generate the data, as is (Nigam et al., 2000). Each mixture component corresponds to a Naïve Bayes model in which the feature values are assumed to be conditionally independent given the mixture component. An EM algorithm is applied, beginning with an initial assignment of emails to clusters, then iteratively solving for a locally maximum likelihood assignment of emails to clusters. The number of

clusters, *k*, is provided as an input to the program. We consider two approaches to generating the initial assignment of emails to clusters. Random Initialization (RI) involves randomly assigning each email a probability of belonging to each of the *k* clusters. This is done by drawn *k* numbers at random from a uniform distribution, then normalizing so that these numbers sum to one. Distant Initialization (DI) generates *k* distinct initial groups of emails, where each group consists of five similar emails (according to the cosine similarity over all features). These *k* groups are chosen using a heuristic sampling method designed to find groups with maximum inter-group distance (again measured by cosine similarity).

In order to ensure a minimum size of a cluster, we adjust tiny clusters (AT) by the following steps: first, eliminate a cluster if the summation of instance weights of the cluster is less than two; second, create a new cluster with an instance of the least document likelihood given the current model.

We also want to use thread information, such like replies with the same subject line, which is specific in the email domain. We call it blending identification views (BV). To incorporate identification views into the Naïve Bayes model, we modify the formula of posterior probability by blending in the average posterior probability of the identification group.

$$P(c_i|d_k) = \frac{P(d_k|c_i)}{\sum_{j=1}^{m} P(d_k|c_j)}$$

$$\text{Given } D_L = \{d_{l1}, d_{l2}, ...\} \text{ has the same subject line (L feature). } |D_L| = Q$$

$$P(c_i|D_L) = \sum_{q=1}^{Q} P(c_i|d_{lq})/Q$$

$$P^{new}(c_i|d_k) = \lambda * P(c_i|D_L) + (1-\lambda) * P(c_i|d_k)$$

2. Bottom-up agglomerative clustering. (BU) In contrast to the above algorithm, this algorithm finds clusters bottom-up, by iteratively merging email documents into trees that represent clusters. The process begins by considering only the subject lines of emails, grouping those emails containing identical subject lines (obtained in the same manner as subject lines from H features) into nodes of a cluster subtree. This is done with the goal of discerning email groups that formed a discussion thread, as it was found that approximately 99% of emails that were from the same thread belonged to the same user activity. Once groups of emails belonging to the same thread are identified and made into subtrees, the algorithm continues by finding the pair of subtrees with the largest cosine distance (indicating greatest similarity). This pair of subtrees is merged, and the process iterated until all emails are merged into a single tree that forms a hierarchical clustering of the emails. The target number of clusters, *k*, is provided as an input to the program and *k* pairs of adjacent leaves in the tree which have the smallest cosine distances are found. These are then taken to be the boundaries between activity clusters with leaves in between boundaries considered to be emails from the same activity.

3. Cluster Representation

1. Keywords. We choose chi-squared measurement for keyword selection on vocabularies based on counts of occurrence and absence within and not within a given cluster. The top N salient words are considered to be the keywords of the cluster.

2. Participating email addresses and their fraction of participation.

3. *The percentage of all user's emails that fall into this cluster.*

4. *Intensity of user involvement.* The fraction of emails within this cluster was authored by the user.

5. Name entity and temporal information. We used the stand-alone java toolkit, Minorthird (Cohen et al.), for information extraction purposes. We extract three specific types of entities, names, times and dates, from the body of emails within the same cluster.

6. *Request emails.* We used a previously-trained email classifier (Cohen et al. 2004) to identify emails within the cluster that constitute requests (e.g., requests for a meeting, requests for information).

4. Experiments and Results

We evaluate the results of our clustering algorithms over three email corpora from three authors. Two corpora (DG and YH) were sorted before the experiments by its user into distinct email folders that typically reflect distinct user activities and the other corpus (TM) was a flat collection of the user's emails. The clustering algorithms were given the union of emails from these folders, but not the folder labels. For DG and YH, the algorithms were evaluated by their accuracy in reconstructing the original folder partitioning of the emails. The DG corpus included a total of 486 emails from 15 folders and the YH corpus included a total of 623 emails from 11 folders and 5 of them are hierarchical. Table 1 and 2 summarizes the accuracies of several algorithms and feature sets on DG and YH corpora. Algorithm names beginning with EM and BU indicate EM and Bottom-Up respectively, and RI and DI refer to the initialization method discussed above. The representation used by the algorithm is indicated by H, and B, which refer to the Header, and Body tokens as described above. The variance of accuracy reported in the table is based on 10 trials on the same EM configuration with different randomly chosen starting points. As shown in these tables, each of these algorithms results in accuracy substantial greater than randomly assigning emails to folders, and the best performing algorithm is EM-DI on DG corpus and EM+BV-DI on YH corpus.

TABLE I. RESULTS ON DG CORPUS

BU		
BU(B)		0.55
BU(HB)		0.72
EM(HB)		
Initialization	Adjustment	Accuracy
Random-ini	-	0.60 +- 0.18
Distant-ini(HB)	-	**0.79 +- 0.08**
Distant-ini(HB)	BV(λ=0.5)	0.75 +- 0.14
Distant-ini(HB)	AT	0.76 +- 0.13

TABLE II. RESULTS ON YH CORPUS

BU		
BU(B)		0.41
BU(HB)		0.49
EM(HB)		
Initialization	Adjustment	Results
Random-ini	-	0.41 +- 0.14
Distant-ini(HB)	-	0.48 +- 0.13
Distant-ini(HB)	BV(λ=0.5)	**0.50 +- 0.07**
Distant-ini(HB)	AT	0.48 +- 0.10

We used the TM corpus to explore the feasibility of automatically constructing structured descriptions of the dominant activity associated with each discovered cluster. In a qualitative evaluation, user TM found each cluster was related to one or more of his ongoing activities (e.g., a committee, family email, etc.), but that each cluster also contained 20-50% of extraneous emails not affiliated with the main activity of the cluster. Despite this non-homogeneity in the clusters, the post-processed descriptions of each cluster often produced quite reasonable descriptions of its dominant activity. Figure 1 shows the automatically constructed description for one cluster of TM emails related to a research project named CALO, involving research on intelligent workstation assistants. The keywords, emails, and extracted names were found by the user to be highly related to this CALO activity.

ActivityCluster5 (105 emails)

- *Keywords*: CALO, TFC, SRI, examples, heads, labeled, Leslie, HMM, contacts, email, task, estimates, zero, reschedule, baseline, Rebecca
- *Primary Senders:* mitchell@cs.cmu.edu(39), lpk@ai.mit.edu(7), mccallum@cs.umass.edu(6)
- *User Activity Fraction*: 105/1448=.072 of total email
- *Intensity Of User Involvement:* created 37% of traffic; (default 31%)
- *Extracted Names:* Leslie(23), Rebecca(21), Carlos(12), Ray(10), Stuart(9), William(9), April(9), …
- *Extracted Dates:* Wed(39), Tues(33), Fri(25), Mon(23), Thurs(20),⋯ Feb 18 (16)
- *Extracted Times:* 5pm(24), noon(14), morning(8), 8am(7), before 5pm(7),...

Figure 1: Automatically constructed activity description

4. Conclusions and Future Work

In summary, our clustering algorithms produced email clusters that reproduced email folder assignments for DG and YH at accuracies of 40-80%, and produced clusters that TM found qualitatively aligned with recognizable activities. Despite the non-homogeneity of these clusters, they often produced reasonable structured representations of the dominant user activity associated with the cluster, such as the one shown in Figure 1. In future work we intend to explore the use of social networks of senders and receivers to refine clusters. In addition, we plan to explore a co-clustering algorithm that is the unsupervised analog to the co-training algorithm (Blum & Mitchell, 1998), and to explore algorithms that jointly cluster calendar entries, files, and web accesses that are related to emails.

This work was supported by Darpa under the CALO and RADAR project contracts.

Reference

Text Classification from Labeled and Unlabeled Documents using EM. Kamal Nigam, Andrew McCallum, Sebastian Thrun and Tom Mitchell. *Machine Learning,* 2000.

Combining Labeled and Unlabeled Data with Co-Training . Avrim Blum and Tom Mitchell. *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT-98).*

Minorthird project webpage. William W. Cohen, Edoardo Airoldi, Vitor Rocha de Carvalho, Einat Minkov, Sunita Sarawagi, Kevin Steppe, and Richard Wang. *http://minorthird.sourceforge.net/*

Learning to Classify Email into "Speech Acts". William W. Cohen, Vitor R. Carvalho and Tom Mitchell. *EMNLP-2004 (to appear)*