Supervised Clustering of Streaming Data for Email Batch Detection

Peter Haider HAIDER@MPI-INF.MPG.DE
Ulf Brefeld BREFELD@MPI-INF.MPG.DE
Tobias Scheffer SCHEFFER@MPI-INF.MPG.DE
Max Planck Institute for Computer Science, Saarbrücken, Germany

Abstract

We address the problem of detecting batches of emails that have been created according to the same template. This problem is motivated by the desire to filter spam more effectively by exploiting collective information about entire batches of jointly generated messages. The application matches the problem setting of supervised clustering, because examples of correct clusterings can be collected. Known decoding procedures for supervised clustering are cubic in the number of instances. When decisions cannot be reconsidered once they have been made – owing to the streaming nature of the data – then the decoding problem can be solved in linear time. We devise a sequential decoding procedure and derive the corresponding optimization problem of supervised clustering. We study the impact of collective attributes of email batches on the effectiveness of recognizing spam emails.

1. Introduction

Senders of spam, phishing, and virus emails avoid mailing multiple identical copies of their messages. Once a message is known to be malicious, all subsequent identical copies of the message could be blocked easily, and without any risk of erroneously blocking regular emails. Collective features of jointly generated batches of messages could provide additional hints for automatic classification, if batches could be recognized as such. Tools for spam, phishing, and virus dissemination employ templates and stochastic grammars, for text messages as well as for images and the source code of viruses. The templates are instantiated for each message. Table 1 shows two illustrative spam messages, generated from the same template.

A natural approach to identifying batches in incoming messages is to cluster groups of similar instances. But unlike for exploratory data analysis, a *ground truth* of *correct clusterings* exists. In order to decide which technique to use, one has to consider the characteristics of electronic messaging.

The overall amount of spam in electronic messages is estimated to be approximately 80 percent. Currently, 80 to 90 percent of these messages are generated by only a few spam senders, each of them maintaining a small number of templates at a time, but exchanging them rapidly. Thus, examining the total email traffic of a short time window, the bulk of incoming messages has been generated by a small number of templates while the remaining 20 percent cover newsletters, personal, and business communications. In a clustering solution, the latter would result in a large number of singleton clusters while newsletters and spam batches congregate in many large and some very large groups. An appropriate clustering algorithm needs to allow for arbitrarily many clusters and an adjustable similarity measure that can be adapted to yield the ground truth of correct clusterings.

At first blush, correlation clustering meets all these requirements. Finley and Joachims (2005) adapt the similarity measure of correlation clustering by structural support vector machines. The solution is equivalent to a poly-cut in a fully connected graph spanned by the messages and their pairwise similarities. However, this solution ignores the temporal structure of the data. And although training can be performed offline, the correlation clustering procedure has to make a decision for each incoming message in real time as to whether it is part of a batch. Larger email service providers have to deal with an amount of emails in the order of 10^8 emails each day. Being cubic in the

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

Table 1. Two spam mails from the same batch.

Hello, This is Terry Hagan.We are accepting your mo rtgage application. Our company confirms you are legible for a $250.000 loan for a $380.00/month. Approval process will take 1 minute, so please fill out the form on our website: http://www.competentagent.com/application/ Best Regards, Terry Hagan; Senior Account Director Trades/Fin ance Department North Office
Dear Mr/Mrs, This is Brenda Dunn.We are accepting your mortga ge application. Our office confirms you can get a $228.000 lo an for a $371.00 per month payment. Follow the link to our website and submit your contact information. Easy as 1,2,3. http://www.competentagent.com/application/ Best Regards, Brenda Dunn; Accounts Manager Trades/Fin ance Department East Office

number of instances, this solution leads to intractable problems in practice.

We devise a sequential clustering technique that overcomes these drawbacks. Exploiting the temporal nature of the data, it is linear in the number of instances. Sequential clustering can easily be integrated in structural SVMs, allowing for the similarity measure to be adapted on a labeled training set.

Our paper is structured as follows. We discuss related work in Section 2 and introduce the problem setting in Section 3. In Section 4, we derive a learning method starting from a relaxed clustering variant. In Section 5, we exploit the temporal nature of the data and devise a sequential clustering algorithm with an appropriate learning variant. We report on experimental results in Section 6. Section 7 concludes.

2. Related Work

Prior work on clustering of streaming data mainly focused on finding single-pass approximations to k-Center algorithms. Guha et al. (2003) develop a constant-factor approximation to k-Median clustering, whereas Ordonez (2003) use an incremental version of k-Means for clustering streams of binary data.

Prior information about the clustering structure of a data set allows for enhancements to clustering algorithms such as k-Means. For instance, Wagstaff et al. (2001) incorporate the background knowledge as must-link and cannot-link constraints into the clustering process, while Bar-Hillel et al. (2003) and Xing et al. (2002) learn a metric over the data space that incorporates the prior knowledge.

Using batch information for spam classification has been studied for settings where multiple users receive spam emails from the same batch. Gray and Haahr (2004) as well as Damiani et al. (2004) discuss difficulties concerning the distribution of batch information and trust between users, while mostly heuristics are used to identify duplicate emails from the same batch. More sophisticated exploration of robust identification of duplicates has been done in other domains. Learning adaptive similarity measures from data has previously been studied by Ristad and Yianilos (1997).

Correlation clustering on fully connected graphs is introduced in (Bansal et al., 2002). A generalization to arbitrary graphs is presented in (Charikar et al., 2005), and Emanuel and Fiat (2003) show the equivalence to a poly-cut problem. Approximation strategies to the NP-complete decoding are presented in (Demaine & Immorlica, 2003; Swamy, 2004). Finley and Joachims (2005) investigated supervised clustering with structural support vector machines.

Several discriminative algorithms have been studied that use joint spaces of input and output variables; these include max-margin Markov models (Taskar et al., 2004) and structural support vector machines (Tsochantaridis et al., 2005). These methods use kernels to compute the inner product in input output space. This approach allows to capture arbitrary dependencies between inputs and outputs. An application-specific learning method is constructed by defining appropriate features, and choosing a decoding procedure that efficiently calculates the argmax, exploiting the dependency structure of the features.

3. Problem Setting

In this section, we abstract the problem of detecting batches in an email stream into a well-defined problem setting. We decompose the problem into *decoding* and *parameter estimation* and derive an appropriate loss function for the parameter estimation step.

A mail transfer agent processes a continuous stream of messages; for each message, it needs to decide which action to take. Possible actions are to accept the message from the connecting agent and to deliver it to the recipient; to reject the message within the SMTP session; or to accept the message and file it into the recipient's spam folder. We focus on the decision on which messages are part of the same batch. The policy on a final action to take can depend on whether this batch

is already blacklisted as being malicious, and possibly on the output of a classifier that uses information in the email as well as in the entire batch.

The agent can take only a fixed number of messages into account when making decisions, for obvious memory constraints. We model the problem such that at each time, a window of messages $\mathbf{x}$ is visible. The output is an adjacency matrix $\mathbf{y}$, where $y_{jk} = 1$ if x_j and x_k are elements of the same batch, and 0 otherwise.

Training data consists of n *sets* of training emails $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$ with $T^{(1)}, \ldots T^{(n)}$ elements. Each set $\mathbf{x}^{(i)}$ represents a snapshot of the window of observable messages. For each training set we are given the correct partitioning into batches and singleton emails by means of adjacency matrices $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n)}$.

A set of pairwise feature functions $\phi_d : (x_j, x_k) \mapsto r \in \mathbb{R}$ with $d = 1, \ldots, D$ is available. The feature functions implement aspects of the correspondence between x_j and x_k. Examples of such functions are the TFIDF similarity of the message bodies, the edit distance of the subject lines, or the similarity of color histograms of images included in the messages. All feature functions are stacked into a similarity vector $\Phi(x_j, x_k)$.

The desired solution is a procedure that produces an adjacency matrix minimizing the number of incorrect assignments of emails to batches, where *incorrect* refers to the ground truth that is reflected in the training data. The number of incorrect assignments is measured by the following loss function $\Delta : (\mathbf{y}, \hat{\mathbf{y}}) \mapsto r \in \mathbb{R}_0^+$. Mis-assigning an element x_j to a batch corrupts a number of matrix elements y_{jk} equal to the size of the batch. Intuitively, mis-assigning a message to a small batch is as bad as mis-assigning it to a large batch. Therefore, in order to quantify the total number of incorrect assignments, the number of bad links for each x_j is divided by the size of the batch that x_i is assigned to:

$$\Delta_N(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{j,k:k<j} \frac{|y_{jk} - \hat{y}_{jk}|}{\sum_{k' \neq j} y_{k'k}}.$$

We will now introduce the *model parameters* and decompose the problem into *decoding* and *parameter estimation*. It is natural to find a similarity value $\mathrm{sim}_{\mathbf{w}}(x_j, x_k)$ by linearly combining the pairwise feature functions with a weight vector $\mathbf{w}$, forging the parameterized similarity measure of Equation 1.

$$\mathrm{sim}_{\mathbf{w}}(x_j, x_k) = \sum_{d=1}^{D} w_d \phi_d(x_j, x_k) = \mathbf{w}^\top \Phi(x_j, x_k) \quad (1)$$

Applying the similarity function to all pairs of emails in a set yields a similarity matrix. The problem of creating a consistent clustering of instances from a similarity matrix is equivalent to the problem of *correlation clustering* (Bansal et al., 2002).

Given the parameters $\mathbf{w}$, the decoding problem is to produce an adjacency matrix $\hat{\mathbf{y}} = \mathrm{argmax}_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})$ that maximizes a decision function f, subject to the constraint that $\hat{\mathbf{y}}$ be a consistent clustering. In standard correlation clustering, the objective is the intra-cluster similarity:

$$f(\mathbf{x}, \mathbf{y}) = \sum_{j,k} y_{jk} \mathrm{sim}_{\mathbf{w}}(x_j, x_k). \quad (2)$$

The parameter learning problem is to obtain weights $\mathbf{w}$ such that, for a new stream of messages, the $\mathbf{w}$-parameterized decoding procedure produces clusterings that minimize *risk; i.e.,* the expected loss

$$R(f) = \int \Delta(\mathbf{y}, \mathrm{argmax}_{\bar{\mathbf{y}}} f(\mathbf{x}, \bar{\mathbf{y}})) p(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}, \quad (3)$$

where $p(\mathbf{x}, \mathbf{y})$ is the (unknown) distribution of sets of objects and their correct clusterings.

4. Learning to Cluster

Supervised clustering elegantly fits into the framework of learning support vector machines with structured output spaces (Tsochantaridis et al., 2005). Finley and Joachims (2005) use an iterative algorithm for learning the weight vector; it starts with an empty set of constraints and adds the most strongly violated constraint in each iteration. We briefly review the model and decoding problem and derive the parameter optimization problem for our loss function. We arrive at a compact optimization problem that can be solved using standard tools instead of an iterative procedure.

In standard correlation clustering, the decision function to be maximized by the clustering is the intra-cluster similarity. Substituting Equation 1 into Equation 4 shows that the decision function is an inner product of parameters and a vector $\Psi(\mathbf{x}, \mathbf{y})$ that jointly represents input $\mathbf{x}$ and output $\mathbf{y}$ (Equation 5).

$$\begin{aligned} f(\mathbf{x}, \mathbf{y}) &= \sum_{t=1}^{T} \sum_{k=1}^{t-1} y_{tk} \mathrm{sim}_{\mathbf{w}}(x_t, x_k) && (4) \\ &= \sum_{t=1}^{T} \sum_{k=1}^{t-1} y_{tk} \mathbf{w}^\top \Phi(x_t, x_k) \\ &= \mathbf{w}^\top \left(\sum_{t=1}^{T} \sum_{k=1}^{t-1} y_{tk} \Phi(x_t, x_k) \right) \\ &= \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y}). && (5) \end{aligned}$$

Given parameters $\mathbf{w}$ and a set of instances $\mathbf{x}$, the decoding problem is to find the highest-scoring clustering

$$\begin{aligned}\hat{\mathbf{y}} &= \text{argmax}_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}) \\ \text{s.t. } & \forall_{jkl} : (1 - y_{jk}) + (1 - y_{kl}) \geq (1 - y_{jl}) \\ & \forall_{jk} : y_{jk} \in \{0, 1\}.\end{aligned} \tag{6}$$

Equation 6 requires $\hat{\mathbf{y}}$ to be a consistent clustering: if x_j and x_k are elements of the same cluster and x_k and x_l are in the same cluster, then x_j and x_l have to be in the same cluster as well. Unfortunately, maximizing $f(\mathbf{x}, \mathbf{y})$ over integer assignments of matrix elements y_{jk} is NP-complete. A common approach is to approximate it by relaxing the binary edge labels y_{jk} to continuous variables $z_{jk} \in [0, 1]$.

$$\begin{aligned}\hat{\mathbf{z}} &= \text{argmax}_{\mathbf{z}} f(\mathbf{x}, \mathbf{z}) \\ \text{s.t. } & \forall_{jkl} : (1 - z_{jk}) + (1 - z_{kl}) \geq (1 - z_{jl}) \\ & \forall_{jk} : z_{jk} \in [0, 1]\end{aligned} \tag{7}$$

We refer to this decoding strategy as the *LP decoding*; it is cubic in the size of the window $\mathbf{x}$. Parameter $\mathbf{w}$ is chosen as to minimize the regularized empirical counterpart of the risk in Equation 3 (Tsochantaridis et al., 2005):

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi^{(i)} \tag{8}$$

$$\text{s.t. } \forall_i \ \mathbf{w}^\top \Psi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) + \xi^{(i)} \geq \max_{\bar{\mathbf{y}}} \mathbf{w}^\top \Psi(\mathbf{x}^{(i)}, \bar{\mathbf{y}}) + \Delta(\mathbf{y}^{(i)}, \bar{\mathbf{y}}) \tag{9}$$

$$\forall_i \ \xi^{(i)} \geq 0. \tag{10}$$

Replacing the right-hand side of constraint 9 with their continuous approximations and substituting the normalized loss function Δ_N, we can write it as

$$\begin{aligned}&\max_{\bar{\mathbf{z}}} \mathbf{w}^\top \Psi(\mathbf{x}^{(i)}, \bar{\mathbf{z}}) + \Delta_N(\mathbf{y}^{(i)}, \bar{\mathbf{z}}) \\ &= \max_{\bar{\mathbf{z}}} \mathbf{w}^\top \Psi(\mathbf{x}^{(i)}, \bar{\mathbf{z}}) + \sum_{k<j} \frac{|y_{jk}^{(i)} - \bar{z}_{jk}|}{\sum_{k' \neq j} y_{k'k}^{(i)}} \\ &= \max_{\bar{\mathbf{z}}} d^{(i)} + \sum_{j,k<j} z_{jk}^{(i)} (\mathbf{w}^\top \Phi(x_j^{(i)}, x_k^{(i)}) - e_{jk}^{(i)}),\end{aligned}$$

where $d^{(i)} = \sum_{j,k<j} \frac{y_{jk}^{(i)}}{\sum_{k' \neq j} y_{k'k}^{(i)}}$ and $e_{jk}^{(i)} = \frac{2y_{jk}^{(i)} - 1}{\sum_{k' \neq j} y_{k'k}^{(i)}}$, and $\bar{\mathbf{z}}$ ranges over all relaxed adjacency matrices which satisfy the triangle inequality (Equation 7). Integrating these constraints into the objective function leads to the corresponding Lagrangian

$$\begin{aligned}L(\mathbf{z}^{(i)}, \boldsymbol{\lambda}^{(i)}, \boldsymbol{\nu}^{(i)}, \boldsymbol{\kappa}^{(i)}) &= d^{(i)} + \boldsymbol{\nu}^{(i)\top} \mathbf{1} + \boldsymbol{\lambda}^{(i)\top} \mathbf{1} \\ &+ \left[\Phi(\mathbf{x}^{(i)})\mathbf{w} - \mathbf{e}^{(i)} - A^{(i)\top} \boldsymbol{\lambda}^{(i)} - \boldsymbol{\nu}^{(i)} + \boldsymbol{\kappa}^{(i)}\right]^\top \mathbf{z}^{(i)},\end{aligned}$$

where the coefficient matrix $A^{(i)}$ is defined as

$$A_{jkl,j'k'}^{(i)} = \begin{cases} +1 & : \text{ if } (j' = j \wedge k' = k) \\ & : \ \vee (j' = k \wedge k' = l) \\ -1 & : \text{ if } j' = j \wedge k' = l \\ 0 & : \text{ otherwise.} \end{cases}$$

The substitution of the derivatives with respect to $\mathbf{z}^{(i)}$ into the Lagrangian and elimination of $\boldsymbol{\kappa}^{(i)}$ removes its dependence on the primal variables and we resolve the corresponding dual that is given by

$$\begin{aligned}\min_{\boldsymbol{\lambda}^{(i)}, \boldsymbol{\nu}^{(i)}} & \ d^{(i)} + \boldsymbol{\nu}^{(i)\top} \mathbf{1} + \boldsymbol{\lambda}^{(i)\top} \mathbf{1} \\ \text{s.t. } & \Phi(\mathbf{x}^{(i)})\mathbf{w} - \mathbf{e}^{(i)} - A^{(i)\top} \boldsymbol{\lambda}^{(i)} - \boldsymbol{\nu}^{(i)} \leq \mathbf{0} \\ & \boldsymbol{\lambda}^{(i)}, \boldsymbol{\nu}^{(i)} \geq \mathbf{0}.\end{aligned}$$

Strong duality holds and the minimization over $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$ can be combined with the minimization over $\mathbf{w}$. The reintegration into Equations 8-10 finally leads to the integrated Optimization Problem 1.

Optimization Problem 1 *Given n labeled clusterings, $C > 0$; over all $\mathbf{w}$, $\xi^{(i)}$, $\boldsymbol{\lambda}^{(i)}$, and $\boldsymbol{\nu}^{(i)}$, minimize $\frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{n} \xi^{(i)}$ subject to the constraints*

$$\begin{aligned}\forall_{i=1}^{n} \ & \mathbf{w}^\top \Psi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) + \xi^{(i)} \geq d^{(i)} + \boldsymbol{\nu}^{(i)\top} \mathbf{1} + \boldsymbol{\lambda}^{(i)\top} \mathbf{1}, \\ \forall_{i=1}^{n} \ & \mathbf{w}^\top \Phi(\mathbf{x}^{(i)}) - \mathbf{e}^{(i)} \leq A^{(i)} \boldsymbol{\lambda}^{(i)} + \boldsymbol{\nu}^{(i)}, \\ \forall_{i=1}^{n} \ & \boldsymbol{\lambda}^{(i)}, \boldsymbol{\nu}^{(i)} \geq \mathbf{0}.\end{aligned}$$

Optimization Problem 1 can be solved directly using standard QP-solvers. Because of the cubic number of triangle inequalities, the number of Lagrange multipliers $\boldsymbol{\lambda}^{(i)}$ in Optimization Problem 1 is cubic in the number of emails $T^{(i)}$ per set. Finley and Joachims (2005) chose a similar approach but arrive at an iterative algorithm to learn the weight vector. The iterative algorithm represents only a subset of the constraints and therefore achieves a speedup at training time. In our case, the training samples are modestly sized whereas, at application time, a high-speed stream has to be processed. Therefore, we will develop a linear decoder in the next section. The linear decoder will also reduce the complexity of the parameter optimization problem from cubic to quadratic.

5. Clustering of Streaming Data

In our batch detection application, incoming emails are processed sequentially. The decision on the cluster assignment has to be made immediately, within an SMTP session, and cannot be altered thereafter. Because of the high volume of the email stream, any

Algorithm 1 Sequential Clustering

$\mathcal{C} \leftarrow \{\}$
for $t = 1 \dots T$ **do**
  $c_j \leftarrow \text{argmax}_{c \in \mathcal{C}} \sum_{x_k \in c} \mathbf{w}^\top \Phi(x_k, x_t)$
  if $\sum_{x_k \in c_j} \mathbf{w}^\top \Phi(x_k, x_t) < 0$ **then**
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{\{x_t\}\}$
  else
    $\mathcal{C} \leftarrow \mathcal{C} \setminus \{c_j\} \cup \{c_j \cup \{x_t\}\}$
  end if
end for
return $\mathcal{C}$

decoding algorithm requiring more than linear execution time in the number of emails processed and the number of emails in the window would be prohibitive.

We therefore impose the constraint that cluster membership cannot be reconsidered once a decision has been made in the decoding procedure. When the partitioning of all previous emails in the window is fixed, a new mail is processed by either assigning it to one of the existing clusters, or creating a new singleton batch. Algorithm 1 details this approach; the initially empty partitioning $\mathcal{C}$ becomes a singelton cluster when the first message arrives. Every new message then either groups to an existing cluster c_j or extends $\mathcal{C}$ by forming its own singelton cluster $\{x_t\}$, respectively.

In general, given a fixed clustering of $x_1, \dots, x_{T-1}$, the decoding problem of finding the $\mathbf{y}$ that maximizes Equation 5 reduces to

$$\max_{\mathbf{y}} \sum_{t=1}^{T} \sum_{k=1}^{t-1} y_{tk} \text{sim}_{\mathbf{w}}(x_t, x_k) \tag{11}$$

$$= \max_{\mathbf{y}} \sum_{t=1}^{T-1} \sum_{k=1}^{t-1} y_{tk} \text{sim}_{\mathbf{w}}(x_t, x_k) + \sum_{k=1}^{T-1} y_{Tk} \text{sim}_{\mathbf{w}}(x_T, x_k). \tag{12}$$

The first summand is constant. Finding the maximum in Equation 11 therefore amounts to assigning it to the cluster which is most similar to x_T or, if no existing cluster has positive total similarity, establishing a new singleton cluster.

In terms of the adjacency matrix $\mathbf{y}^{(i)}$ of the i-th input, the task is to find entries for the T-th row and column, realizing the optimal clustering of x_T. We denote the set of matrices that are consistent clusterings and are equal to the i-th example, $\mathbf{y}^{(i)}$, in all rows/columns except for the T-th row/column, by $\mathcal{Y}_T^{(i)}$. If we denote the potential new cluster (which is empty before inserting x_T) with $\bar{c}$, $\mathcal{Y}_T^{(i)}$ is of the size $|\mathcal{C} \cup \{\bar{c}\}| \leq T^{(i)}$.

Finding the new optimal clustering can be expressed as the following maximization problem.

Decoding Strategy 1 *Given* $T^{(i)}$ *instances* $x_1, \dots, x_{T^{(i)}}$, *similarity measure* $\text{sim}_{\mathbf{w}} : (x_j, x_k) \mapsto r \in \mathbb{R}$, *and a clustering of instances* $x_1, \dots, x_{T^{(i)}-1}$; *the sequential decoding problem is defined as*

$$\hat{\mathbf{y}} = \max_{\bar{\mathbf{y}} \in \mathcal{Y}_T^{(i)}} \sum_{k=1}^{T^{(i)}-1} \bar{y}_{T^{(i)}k} \text{sim}_{\mathbf{w}}(x_{T^{(i)}}, x_k). \tag{13}$$

Now, we derive an optimization problem that requires the sequential clustering to produce the correct output for all training data. Optimization Problem 2 constitutes a compact formulation for finding the desired optimal weight vector by treating every message as the most recent message once, in order to exploit the available training data as effectively as possible.

Optimization Problem 2 *Given* n *labeled clusterings,* $C > 0$; *over all* $\mathbf{w}$ *and* $\boldsymbol{\xi}$, *minimize* $\frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i,j} \xi_j^{(i)}$ *subject to the constraints*

$$\mathbf{w}^\top \Psi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) + \xi_t^{(i)} \geq \mathbf{w}^\top \Psi(\mathbf{x}^{(i)}, \bar{\mathbf{y}}) + \Delta_N(\mathbf{y}^{(i)}, \bar{\mathbf{y}})$$

for all $1 \leq i \leq n$, $1 \leq t \leq T^{(i)}$, *and* $\bar{\mathbf{y}} \in \mathcal{Y}_t^{(i)}$.

Note that Optimization Problem 2 has at most $\sum_{i=1}^{n} (T^{(i)})^2$ constraints and can efficiently be solved with standard QP-solving techniques.

6. Experimental Results

In this section we evaluate the performance and benefit of batch detection on a collection of emails. We compare our learning methods with the iterative learning procedure for supervised clustering by Finley and Joachims (2005) and perform an error analysis. We evaluate how the identification of email batches can actually support the classification of emails as spam or non-spam. Furthermore, we assess the execution time of the presented decoding methods. Quadratic programs are solved with CPLEX.

6.1. Email Batch Data

Email batch detection is performed at a mail transfer agent that processes a dense stream of messages. Standard email collections such as the Enron corpus or the TREC spam collection are collected from final recipients and therefore exhibit different characteristics. A mail transfer agent experiences many large batches over a short period of time. Existing spam corpora were harvested over a longer period from clients and

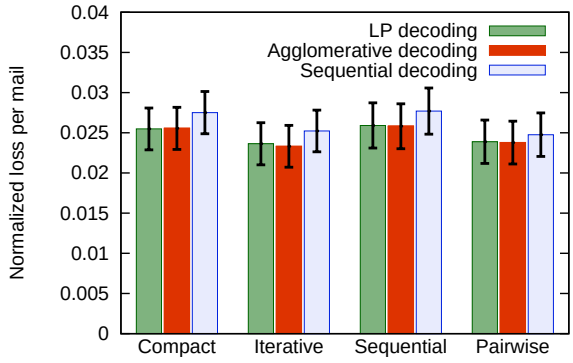

Figure 1. Average loss for window size m = 100.

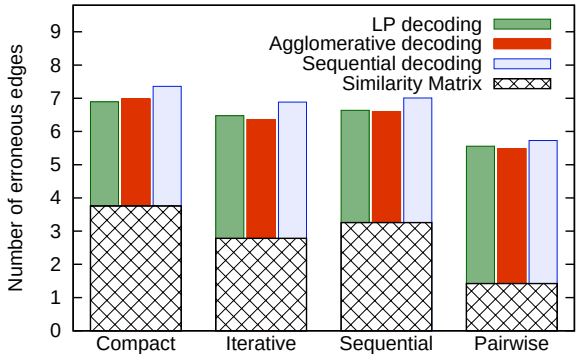

Figure 2. Fraction of the loss induced by the learning algorithm (similarity matrix) and the decoding.

contain fewer and more scattered copies of each batch. We therefore create an email corpus that reflects the characteristics of an email stream, but remedies the obvious privacy concerns that would arise from simply recording an email stream at a mail transfer agent. We do record the email stream for a short period of time, but only extract spam messages from this record. We randomly insert non-spam messages from the Enron collection and batches of newsletters. We remove the headers except for the sender address, MIME part information, and the header size.

The final corpus contains 2,000 spam messages, 500 Enron messages, and 500 newsletters (copies of 50 distinct newsletters). We manually group these emails into 136 batches with an average of 17.7 emails, and 598 remaining singleton mails. We implement 47 feature functions. They include the TFIDF similarity, equality of sender, equality of the MIME type, and differences in letter-bigram-counts.

We design a cross validation procedure such that no elements of the same newsletter or spam batch occur in both the training and test set at any time. To this end, we construct each test set by using one non-singular batch, and filling the test sample with singletons and emails of other batches to a total size of 100. Batches with more than 50 emails are divided over several test sets, to ensure a reasonable mixture of emails from the test batch and other emails. Overall, there are 153 test sets. For each of these test sets, nine training sets $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(9)}$ are generated by sampling randomly from the remaining emails, excluding emails from the test batch in case of split test batches. All reported results are averaged over the results from each of the 153 training/test combinations.

6.2. Batch Identification

We compare the parameter vectors obtained by four strategies. Parameters are estimated by solving Optimization Problem 1 (compact), solving Optimization Problem 2 (sequential), and by using the iterative training algorithm of Finley and Joachims (2005) (iterative). As an additional baseline, we train a pairwise classifier (pairwise) classifier: each pair of emails within a set constitutes a training example, with label $+1$ if they belong to the same cluster, and -1 otherwise. On these pairs, a linear SVM is trained, and the weight vector is directly used as parameter of the similarity measure. The final clustering is then obtained by one of the decoding strategies, using the similarity matrix obtained from pairwise learning.

Though three of the four optimization problems refer to a specific decoding strategy, we evaluate each of them with every decoder for comparison. We study three decoders: LP decoding (exact solution of Equation 8), the sequential decoder (Decoding Strategy 1), and the greedy agglomerative clustering described in (Finley & Joachims, 2005). Figure 1 shows the average normalized loss per mail of these combinations with standard error. For this problem, there are no significant differences between either of these training and decoding methods. The sequential decoder operates under the constraint of linearity, and it would be plausible to assume that it incurs a higher loss than the LP decoding on average. The data suggests that this *might* be the case, but the difference is at most slight and by no means significant.

Figure 2 gives more insight into the characteristics of the compared methods. On the y-axis, the number of disagreeing edges with respect to the true clustering is depicted. The hatched areas indicate the number of disagreements between the true clustering and the signs of the similarity matrix induced by the weight vector and the pairwise features. The similarity matrix serves as input to the decoder; the decoder transforms it into a consistent partitioning. The colored bars in-

dicate the numbers of wrong edges after clustering.

It is apparent that the simplest learning method, pairwise learning, leads to the fewest wrong edges before clustering, but the induced similarity matrix is furthest away from being a consistent partitioning. This corresponds to the intuition that the training constraints of pairwise learning refer to individual links instead of the entire partitioning. The iterative algorithm leads to similarity matrices which are significantly nearer to a consistent clustering (the colored bars are shorter). The similarity measures learned by the compact optimization problems lead to a similarity matrix with still more disagreeing edges, while yielding comparable error rates after decoding. This indicates that the decoding step has to resolve fewer inconsistencies, making it more robust to approximations.

6.3. Classification Using Batch Information

We evaluate how the classification of emails as spam or non-spam benefits from identification of batches. As a baseline, we train a linear support vector machine with the word-counts of the training emails as features. We remove all email header information except for the subject line in order to eliminate artefacts from the data collection procedure.

We construct a collective filter that sums up the word counts of all emails in a batch, and includes four additional features: the size of the batch, a binary feature indicating whether the batch is larger than one, a binary feature indicating whether the subject of all emails in the batch is identical, and a binary feature indicating whether the sender address of all emails in the batch is identical. This results in all emails within a batch having the same feature representation.

We examine how the classification performance is affected by the batch detection. As an upper bound, we investigate the performance of the collective classifier given perfect clustering information, based on the manual clustering. In addition to that, we assess how sensitive the benefit of collective classification is with respect to the accuracy of the clustering. In the setting of clustering with noise, each email is collectively classified in a cluster that contains increasingly many wrongly clustered emails.

Figure 3 shows the area under the ROC curve (AUC) for the classifiers under investigation. The performance of the collective classifier based on a perfect clustering can be seen on the right hand side of the graph (ideal clustering at 0% noise). The difference between the collective classification based on a perfect clustering and based on the inferred clusterings

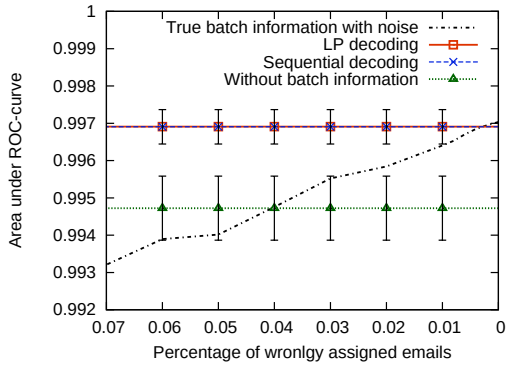

Figure 3. Classification accuracy with batch information.

is not significant. The collective classifiers perform indistinguishably well; sequential and LP decoder perform alike. We can see that using ideal batch information, the risk of misclassification (1 - AUC) is reduced by 43.8%, while with non-ideal batch information obtained through approximate clustering still 41.4% reduction are achieved. Even though the AUC of the baseline appears high already, in spam filtering a 40% reduction of the risk is a substantial improvement!

6.4. Clustering Runtime

An important aspect in clustering on streams and especially in identifying spam batches is efficiency. The window size has to be sufficiently large to contain at least one representative of each currently active batch. The time required to cluster one additional email depending on the window size is therefore a crucial criterion for selecting an appropriate clustering method.

Figure 4 illustrates the observed time required for processing an email by LP-decoding and sequential decoding with respect to the window size. While the computation time of the LP approximation grows at least cubicly, the time for an incremental update for a single email with sequential decoding grows only linearly. Due to the different time-scales of the two methods (note that the center graph shows micro-seconds instead of seconds), we use a logarithmic time-scale to plot the curves in a single diagram (right-hand graph).

7. Conclusion

We devised a sequential clusering algorithm and two integrated formulations for learning a similarity measure to be used with correlation clustering. First, we derived a compact optimization problem based on the LP approximation to correlation clustering to learn the weights of the similarity measure. Starting from the assumption that decisions for already processed emails cannot be reconsidered, we devised an efficient cluster-

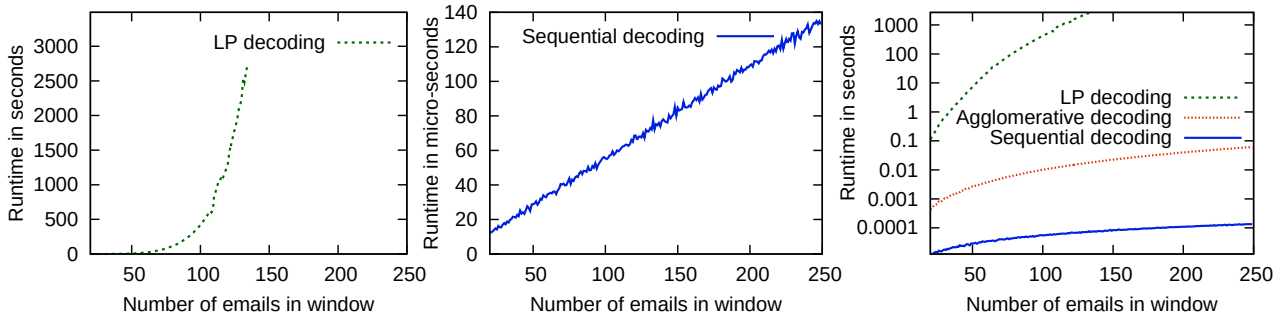

Figure 4. Computation time for adding one email depending on window size.

ing algorithm with computational complexity linear in the number of emails in the window. From this algorithm we derived a second integrated method for learning the weight vector.

Our empirical results indicate that there are no significant differences between the learning or decoding methods in terms of accuracy. Yet the integrated learning formulations optimize the weight vector more directly to yield consistent partitionings. Using the batch information obtained from decoding with the learned models, email spam classification performance increases substantially over the baseline with no batch information. The efficiency of the sequential clustering algorithm makes supervised batch detection in enterprise-level scales, with millions of emails per hour and thousands of recent emails as reference, feasible.

Acknowledgments

We gratefully acknowledge support from STRATO AG and from the German Science Foundation DFG.

References

Bansal, N., Blum, A., & Chawla, S. (2002). Correlation clustering. *Proceedings of the Symposium on Foundations of Computer Science*.

Bar-Hillel, A., Hertz, T., Shental, N., & Weinshall, D. (2003). Learning distance functions using equivalence relations. *Proceedings of the International Conference on Machine Learning*.

Charikar, M., Guruswami, V., & Wirth, A. (2005). Clustering with qualitative information. *Journal of Computer and System Sciences*, *71*, 360–383.

Damiani, E., di Vimercati, S. D. C., Paraboschi, S., & Samarati, P. (2004). P2P-based collaborative spam detection and filtering. *Proceedings of the International Conference on Peer-to-Peer Computing*.

Demaine, E. D., & Immorlica, N. (2003). Correlation clustering with partial information. *Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*.

Emanuel, D., & Fiat, A. (2003). Correlation clustering – minimizing disagreements on arbitrary weighted graphs. *Proceedings of the European Symposium on Algorithms*.

Finley, T., & Joachims, T. (2005). Supervised clustering with support vector machines. *Proceedings of the International Conference on Machine Learning*.

Gray, A., & Haahr, M. (2004). Personalised, collaborative spam filtering. *Proceedings of the Conference on Email and Anti-Spam*.

Guha, S., Meyerson, A., Mishra, N., Motwani, R., & O'Callaghan, L. (2003). Clustering data streams: Theory and practice. *IEEE Transactions on Knowledge and Data Engineering*, *15*, 515–528.

Ordonez, C. (2003). Clustering binary data streams with k-means. *Proceedings of the Workshop on Research Issues in Data Mining and Knowledge Discovery*.

Ristad, E. S., & Yianilos, P. N. (1997). Learning string edit distance. *Proceedings of the International Conference on Machine Learning*.

Swamy, C. (2004). Correlation clustering: maximizing agreements via semidefinite programming. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*.

Taskar, B., Guestrin, C., & Koller, D. (2004). Max-margin Markov networks. *Advances in Neural Information Processing Systems*.

Tsochantaridis, I., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, *6*, 1453–1484.

Wagstaff, K., Cardie, C., Rogers, S., & Schrödl, S. (2001). Constrained k-means clustering with background knowledge. *Proceedings of the International Conference on Machine Learning*.

Xing, E. P., Ng, A. Y., Jordan, M. I., & Russell, S. (2002). Distance metric learning, with application to clustering with side-information. *Advances in Neural Information Processing Systems*.