Discriminative Identification of Duplicates

Peter Haider, Ulf Brefeld, and Tobias Scheffer

Humboldt Universität zu Berlin
Unter den Linden 6, 10099 Berlin, Germany
{haider,brefeld,scheffer}@informatik.hu-berlin.de

Abstract. The problem of finding duplicates in data is ubiquitous in data mining. We cast the problem of finding duplicates in sequential data into a poly-cut problem on a fully connected graph. The edge weights can be identified with parameterized pairwise similarities between objects that are optimized by structural support vector machines on labeled training sets. Our approach adapts the similarity measure to the data and is independent of the number of clusters. We present three large margin approximations of learning the pairwise similarities: an integrated QP-formulation, a sequential multi-class approach and a pairwise classifier. We report on experimental results.

1 Introduction

The problem of identifying duplicates has applications ranging from recognizing objects from different perspectives and angles to the identification of objects that are intentionally altered to obfuscate their true identity, origin, or purpose. This occurs, for instance, in the context of email spam and virus detection.

Spam and virus senders avoid mailing identical copies of their messages because it would be an easy giveaway. Identifying a batch of messages would allow email service providers to hold back the entire batch, and to identify hijacked servers that are being used to disseminate spam or viruses. Therefore, spam senders generate messages according to templates. Table 1 shows an example of two spam messages that have been generated with a spamming tool. Slots of a common template are filled according to a grammar; the tool also applies obfuscation techniques such as random insertions of spaces.

In the database community, the "database deduping problem" is another popular instance of the duplicate identification problem. Other occurrences of the problem include named entity resolution, and the grouping of images that show, for instance, the same person.

A natural approach to identifying duplicates is to group similar objects together by a cluster algorithm. However, prominent algorithms like k-means or Expectation Maximization require the number of clusters beforehand. Moreover, given a problem at hand, it is often ambiguous to decide whether two objects are similar or not.

Correlation clustering [3] meets our requirements by accounting for potentially infinitely many clusters. Its solution is equivalent to a maximum poly-cut

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

Table 1. Two spam mails from the same batch.

in a fully connected graph spanned by the objects and their pairwise similarities [11].

We address the problem of learning a duplicate detection hypothesis from labeled data. That is, we start from data in which all elements that are duplicates of one another have been tagged as such. This allows us to learn the similarity function that parameterizes the clustering model such that it correctly groups the duplicates in the training data. The similarity measure can be learned by structural SVMs in a discriminative way.

We firstly derive a loss augmented optimization problem that can be solved directly. Due to a cubic number of variables, solving this initial problem is hardly tractable for large data sets. Secondly, we present an approach that makes use of the sequential nature of the objects and thirdly, we approximate the optimal solution by a pairwise classifier. Experiments detail characteristics of all three methods.

The rest of our paper is structured as follows. We report on related work in Section 2 and introduce our problem setting together with the decoding strategy in Section 3. We present support vector algorithms for identifying duplicates in Section 4 and report on experimental results in Section 5. Section 6 concludes.

2 Related Work

The identification of duplicates has been studied with fixed similarity measures, such as the fraction of matching words [9, 8] and sentences [6]. Other applications include the identification of duplicates in data bases [5], and in centralized [14] and decentralized networks [23].

Correlation clustering on fully connected graphs is introduced in [2, 3]. A generalization to arbitrary graphs is presented in [7] and [11] shows the equivalence to a poly-cut problem. Approximation strategies to the NP-complete decoding are presented in [10, 17]. Finley and Joachims [13] investigated supervised clustering with structural support vector machines.

Prior information about the cluster structure of a data set allows for enhancements to classical clustering algorithms such as k-means. E.g., Wagstaff et al. [21] incorporate the background knowledge as must-link and cannot-link constraints into the clustering process, while [4, 22] learn a metric over the data space that incorporates the prior knowledge.

Several discriminative algorithms have been studied that utilize joint spaces of input and output variables; these include max-margin Markov models [18], kernel conditional random fields [15], hidden Markov support vector machines [1], and support vector machines for structured output spaces [20]. These methods utilize 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 Preliminaries

The task is to find a model f such that given a set of instances $\mathbf{x}$ the true partitioning $\mathbf{y}$ given as an adjacency matrix yields the highest score

$$\mathbf{y} = \operatorname*{argmax}_{\bar{\mathbf{y}} \in \mathcal{Y}} f(\mathbf{x}, \bar{\mathbf{y}}). \tag{1}$$

We measure the quality of f by an appropriate, symmetric, nonnegative loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_0^+$ that details the distance between the true partition $\mathbf{y}$ and the prediction $\hat{\mathbf{y}} = \operatorname{argmax}_{\bar{\mathbf{y}}} f(\mathbf{x}, \bar{\mathbf{y}})$. A natural measure for two clusterings is the Rand index [16]. The corresponding loss function Δ_{Rand} is given by

$$\begin{aligned}\Delta_{Rand}(\mathbf{y}, \hat{\mathbf{y}}) &= 1 - Q_{Rand}(\mathbf{y}, \hat{\mathbf{y}}) \\ &= 1 - \frac{\sum_{j,k<j} [[y_{jk} = \hat{y}_{jk}]]}{|\mathbf{y}|} \\ &= \sum_{j,k<j} \frac{[[y_{jk} \neq \hat{y}_{jk}]]}{|\mathbf{y}|},\end{aligned}$$

where $[[\sigma]]$ is the indicator function which yields 1 if the proposition σ is true and 0 otherwise. We can restate the optimization problem as finding a function f that minimizes the expected risk

$$R(f) = \int_{\mathcal{X} \times \mathcal{Y}} \Delta_{Rand}(\mathbf{y}, \operatorname{argmax}_{\bar{\mathbf{y}}} f(\mathbf{x}, \bar{\mathbf{y}})) dP(\mathbf{x}, \mathbf{y}) \tag{2}$$

where $P(\mathbf{x}, \mathbf{y})$ is the (unknown) distribution of sets of objects and their clusterings. As in the classical setting we address this problem by searching a minimizer of the empirical risk given by

$$R_S(f) = \frac{1}{n} \sum_{i=1}^{n} \Delta_{Rand}(\mathbf{y}^{(i)}, \text{argmax}_{\bar{\mathbf{y}}}\, f(\mathbf{x}, \bar{\mathbf{y}})), \tag{3}$$

regularized by $\|f\|^2$.

Correlation clustering [3] maintains a symmetric similarity matrix whose elements denote pairwise similarities between objects. This representation allows to recast the problem as a poly-cut problem in a fully connected graph, where objects are identified with nodes and edges are weighted with the respective pairwise similarities. The optimal partitioning can either be found by minimizing the edge weights between clusters of objects or by maximizing the edge weights within clusters of objects. Following the latter leads to the integer optimization problem

$$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \mathcal{Y}}{\text{argmax}}\ f(\mathbf{x}, \mathbf{y}) \ = \ \underset{\mathbf{y} \in \mathcal{Y}}{\text{argmax}} \sum_{j=1}^{|\mathbf{x}|} \sum_{k=1}^{j-1} y_{jk} sim(x_j, x_k) \tag{4}$$

where y_{jk} indicates wether x_j and x_k belong to the same cluster. The set $\mathcal{Y}$ contains all equivalence relations over $\mathbf{x}$ given as an adjacency matrix, that is, all $\mathbf{y}$ which satisfy the triangle inequality $(1 - y_{jk}) + (1 - y_{kl}) \geq (1 - y_{jl})$ where $y_{jk} \in \{0, 1\}$. The maximum is attained by the partitioning $\mathbf{y}$ that maximizes the within-cluster similarities. We follow [13] and use a parameterized similarity measure between two objects x_j and x_k given by

$$sim(x_j, x_k) = \sum_{t=1}^{T} w_t \phi_t(x_j, x_k) = \mathbf{w}^\top \Phi(x_j, x_k), \tag{5}$$

where $\Phi(x_j, x_k) = (..., \phi_t(x_j, x_k), ...)^\top$ is the similarity vector of x_j and x_k; e.g., in our running example $\phi_{234}(x_j, x_k)$ might be an indicator function that equals 1 if both mails are of the same mime-type. Substituting 5 into 4 shows that we can rewrite f as a generalized linear model in joint input output space

$$\begin{aligned} f(\mathbf{x}, \mathbf{y}) &= \sum_{j=1}^{|\mathbf{x}|} \sum_{k=1}^{j-1} y_{jk} sim(x_j, x_k) && (6) \\ &= \sum_{j=1}^{|\mathbf{x}|} \sum_{k=1}^{j-1} y_{jk} \mathbf{w}^\top \Phi(x_j, x_k) && (7) \\ &= \mathbf{w}^\top \underbrace{\left(\sum_{j=1}^{|\mathbf{x}|} \sum_{k=1}^{j-1} y_{jk} \Phi(x_j, x_k) \right)}_{=:\Psi(\mathbf{x}, \mathbf{y})} && (8) \\ &= \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y}). && (9) \end{aligned}$$

In the following we will refer to a sample S of n input output pairs $(\mathbf{x}^{(1)},\mathbf{y}^{(1)}),\ldots,(\mathbf{x}^{(n)},\mathbf{y}^{(n)})$, drawn *i.i.d.* according to $P(\mathbf{x},\mathbf{y})$. The i-th pair contains $|\mathbf{x}^{(i)}| = m_i$ instances $x_1^{(i)},\ldots,x_{m_i}^{(i)}$ with adjacency matrix $\mathbf{y}^{(i)}$ such that $y_{jk}^{(i)} = 1$ if $x_j^{(i)}$ and $x_k^{(i)}$ are in the same partition. We denote the set of all adjacency matrices of possible partitionings of the i-th set by $\mathcal{Y}^{(i)}$.

4 Discriminative Identification of Duplicates

In this section we present three discriminative approaches to the identification of duplicates: an integrated QP-formulation, a sequential multi-class approach, and a pairwise classifier.

4.1 Integrated Optimization Problem

Bansal et al. [3] show that exact inference is NP-complete. However, the optimal solution can be approximated by substituting real valued edge weights $z_{jk} \in [0,1]$ for the integer valued edge weights $y_{jk} \in \{0,1\}$. The decoding problem in Equation 4 can be solved approximately by the following decoding strategy.

Decoding Strategy 1 *Given m instances $x_1,\ldots,x_m \in \mathcal{X}$ and a similarity measure $sim_{\mathbf{w}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$. Over all values $\mathbf{z} \in \mathbb{R}^m$ maximize $\sum_{j=1}^{m}\sum_{k=1}^{j-1} z_{jk} sim(x_j,x_k)$ subject to the constraints $\forall_{j,k,l}\,(1-z_{jk})+(1-z_{kl}) \geq (1-z_{jl})$ and $\forall_{j,k}\, 0 \leq z_{jk} \leq 1$.*

The substitution of the approximate labels, gives rise to the loss function $\Delta_{Rand}(\mathbf{y},\mathbf{z}) = \sum_{j,k<j}(|y_{jk}-z_{jk}|)/|\mathbf{y}|$. The optimization problem of the structural support vector machine in terms of approximate labels $\mathbf{z}$ can be stated as follows.

Optimization Problem 1 *Given n labeled clusterings, loss function Δ_{Rand}, $C > 0$; over all $\mathbf{w}$ and ξ_i minimize $||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$ subject to the constraints $\forall_{i=1}^{n}\mathbf{w}^\top\Psi(\mathbf{x}^{(i)},\mathbf{y}^{(i)}) + \xi_i \geq \max_{\mathbf{z}\in\mathcal{Z}^{(i)}}\left[\mathbf{w}^\top\Psi(\mathbf{x}^{(i)},\mathbf{z}) + \Delta_{Rand}(\mathbf{y}^{(i)},\mathbf{z})\right]$ and $\forall_{i=1}^{n}\xi_i \geq 0$, where $\mathcal{Z}^{(i)}$ consists of all possible approximate labelings of $\mathbf{x}^{(i)}$ which satisfy the triangle inequality.*

Similar to [19] the loss can be integrated into the decoding of the top scoring clustering. This gives us

$$\begin{aligned}
\max_{\mathbf{z}_i} \quad & d^{(i)} + \sum_{j,k<j} \mathbf{z}_{i,jk}(\mathbf{w}^\top\Phi(x_j^{(i)},x_k^{(i)}) - e_{jk}^{(i)}) \\
\text{s.t.} \quad & \forall_{j,k,l} \quad (1-z_{i,jk}) + (1-z_{i,kl}) \geq (1-z_{i,jl}), \\
& \forall j,k \quad 0 \leq z_{i,jk} \leq 1,
\end{aligned}$$

where $d^{(i)} = \frac{\sum_{j,k<j} y_{jk}^{(i)}}{|y^{(i)}|}$ and $e_{jk}^{(i)} = \frac{2y_{jk}^{(i)}-1}{|y^{(i)}|}$. Integrating the constraint into the objective function leads to the corresponding Lagrangian

$$L(\mathbf{z}_i,\lambda_i,\nu_i,\kappa_i) = d^{(i)} + \nu_i^\top\mathbf{1} + \lambda_i^\top\mathbf{1} + \left[\mathbf{w}^\top\Phi(\mathbf{x}^{(i)}) - \mathbf{e}^{(i)} - A^{(i)}\lambda_i^\top - \nu_i + \kappa_i\right]^\top \mathbf{z}_i$$

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

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

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

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

Strong duality holds and the minimization over λ and ν can be combined with the minimization over $\mathbf{w}$. The reintegration into optimization problem 1 leads to the integrated Optimization Problem 2 that can be solved directly.

Optimization Problem 2 *Given n labeled clusterings, $C > 0$; over all $\mathbf{w}$, ξ_i, λ_i, and ν_i, minimize $||\mathbf{w}||^2 + C\sum_{i=1}^n \xi_i$ subject to the constraints 10-12.*

$$\begin{aligned} \forall_{i=1}^n \; \mathbf{w}^\top \Psi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) + \xi_i &\geq d^{(i)} + \nu_i^\top \mathbf{1} + \lambda_i^\top \mathbf{1}, && (10) \\ \forall_{i=1}^n \quad \mathbf{w}^\top \Phi(\mathbf{x}^{(i)}) - \mathbf{e}^{(i)} &\leq A^{(i)}\lambda_i + \nu_i, && (11) \\ \forall_{i=1}^n \qquad \lambda_i, \nu_i &\geq \mathbf{0}, && (12) \end{aligned}$$

The number of Lagrange multipliers λ_i in Optimization Problem 2 is cubic in the number of instances m_i; i.e., its solution becomes intractable for large data sets. In the following two sections we present two approaches that overcome this drawback.

4.2 Sequential Clustering

Our second approach accounts for the sequential nature of the data. In the server-sided batch detection scenario incoming mails have to be classified immediately upon arrival. In our running example each incoming email is either grouped to an existing batch or it becomes its own singelton batch.

Therefore, it suffices to maintain a window that contains the last m incoming mails. As soon as a new mail arrives it is substituted for the oldest mail in the window and a new clustering is computed. The latter step can be approximated by finding a cluster or opening a new batch only for the latest mail, respectively. Algorithm 1 details this approach.

The adjacency matrix $\mathbf{y}$ can be obtained from the clustering $\mathcal{C}$ by $y_{jk}(\mathcal{C}) = [[\exists c \in \mathcal{C} : x_j \in c \wedge x_k \in c]]$. Given a fixed clustering of $x_1, \ldots, x_{m-1}$, the decoding

Algorithm 1 Sequential Clustering

```
$\mathcal{C} \leftarrow \{\}$
for $j = 1 \ldots |\mathbf{x}|$
    $c_j = \mathrm{argmax}_{c \in \mathcal{C}} \sum_{x_k \in c} \mathbf{w}^\top \Phi(x_k, x_j)$
    if $\sum_{x_k \in c_j} \mathbf{w}^\top \Phi(x_k, x_j) < 0$
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{\{x_j\}\}$
    else
        $\mathcal{C} \leftarrow \mathcal{C} \setminus \{c_j\} \cup \{c_j \cup \{x_j\}\}$
    endif
endfor
return $\mathcal{C}$
```

problem in 4 reduces to

$$\max_{\mathbf{y} \in \mathcal{Y}} \sum_{j=1}^{m} \sum_{k=1}^{j-1} y_{jk} sim_{\mathbf{w}}(x_j, x_k) = \tag{13}$$

$$\max_{\mathbf{y} \in \mathcal{Y}} \sum_{j=1}^{m-1} \sum_{k=1}^{j-1} y_{jk} sim_{\mathbf{w}}(x_j, x_k) + \sum_{k=1}^{m-1} y_{mk} sim_{\mathbf{w}}(x_m, x_k). \tag{14}$$

The first summand of Equation 14 is constant; thus finding a cluster for x_m reduces to the Decoding Strategy 2, where the additional cluster $\bar{c}$ accounts for x_m being dissimilar to its predecessors in the window.

Decoding Strategy 2 *Given m instances $x_1, \ldots, x_m \in \mathcal{X}$, similarity measure $sim_{\mathbf{w}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, and a clustering $\mathcal{C}$ of instances $x_1, \ldots, x_{m-1}$; over all values $c \in \{\mathcal{C} \bigcup \bar{c}\}$ maximize $\sum_{x_k \in c} sim_{\mathbf{w}}(x_m, x_k)$.*

If we denote the set of all possible clusterings in which x_j is reassigned to any cluster by $\mathcal{C}_j$ we derive the following minimization problem.

Optimization Problem 3 *Given n labeled clusterings, $C > 0$; over all $\mathbf{w}$ and ξ_{ij}, minimize $\frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i,j} \xi_{ij}$ subject to the constraints $\forall_{i=1}^{N}, \forall_{j=1}^{m_i}, \forall \hat{\mathcal{C}} \in \mathcal{C}_j^{(i)} \mathbf{w}^\top \Psi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) + \xi_{ij} \geq [\mathbf{w}^\top \Psi(\mathbf{x}^{(i)}, \mathbf{y}(\hat{\mathcal{C}})) + \Delta(\mathbf{y}^{(i)}, \mathbf{y}(\hat{\mathcal{C}}))]$*

Since the number of clusters is upper bounded by the window size, $|\mathcal{C}| \leq m$, Optimization Problem 3 has at most $n \cdot \sum_{i=1}^{n} m_i^2$ constraints and can be solved by standard techniques. This approach is equivalent to single-vector multi-class classification [12]. Also note that the obtained solution for the weight vector $\mathbf{w}$ is independent of the used decoding strategy, and can thus be used with every other approximation of correlation clustering as well.

4.3 Pairwise Classification

The multi-class approach can be further approximated by a binary classifier that outputs class +1 if two instances are similar and class −1 otherwise. Therefore, we use all pairs of instances $(x_j^{(i)}, x_k^{(i)})$ within the training tuple $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ as inputs and define the labels $\upsilon_{jk}^{(i)} = +1$ if $y_{jk}^{(i)} = 1$, and $\upsilon_{jk}^{(i)} = -1$ if $y_{jk}^{(i)} = 0$. This leads us to the standard formulation of a binary support vector machine in Optimization Problem 4.

Optimization Problem 4 *Given n labeled clusterings, $C > 0$; over all $\mathbf{w}$ and ξ_{ijk}, minimize $\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i,j,k}\xi_{ijk}$ subject to the constraints $\forall_{i=1}^{N}, \forall_{j=1}^{m_i}, \forall_{k=1}^{j-1} \upsilon_{jk}^{(i)}(\mathbf{w}^\top \Phi(x_j^{(i)}, x_k^{(i)}) + b) \geq 1 - \xi_{ijk}$.*

The weight vector $\mathbf{w}$ can directly be used as parameter of the similarity measure, i.e. the decision function of the binary classifier is equivalent to the pairwise similarity function. Analogously to the sequential clustering, the pairwise classification allows the use of any decoding strategy.

However, this approach suffers several drawbacks compared to the two previously devised solutions. Firstly, an application-specific loss function cannot be incorporated into the learning problem that implicitly minimizes the 0/1 error. Secondly, transitive dependencies within the training tuples are ignored, that is the training instances are not i.i.d.

5 Empirical Evaluation

We investigate our approaches by applying them to an email batch identification task. We compare the presented training methods with the iterative learning procedure for support vector machines with structured outputs by Finley and Joachims [13]. We explore the benefit of each approach and perform an error analysis.

In our experiments we use a slightly modified variant of the loss function based on the Rand index. Instead of normalizing over the number of all mails as in Equation 2 we use the number of emails in the current batch as normalization. That is, each wrong edge is weighted by the inverse size of its batch. The loss function 15 is linear in $\mathbf{z}$ and independent of the size of the batches, and thus better reflects the intuition about the quality of a batch detection method.

$$\Delta_N(\mathbf{y}^*, \mathbf{y}, j) = \sum_{k \neq j} \frac{[[\,[[y_j^* = y_k^*]] \neq [[y_j = y_k]]\,]]}{\sum_{k' \neq j}[[y_{k'}^* = y_k^*]]}. \tag{15}$$

The feature functions are simple pairwise indicators or measures, such as equality of sender or mimetype, difference of message length, edit-distance of the subject lines, cosine distance of TFIDF-vectors, or differences in letter-bigram-counts. Each wrong edge gets weighted by the inverse of the number of members of its corresponding batch, to even out the influences of large and small batches.

We evaluate our proposed methods on a set of 3000 emails, consisting of 2000 spam mails collected by an email service provider, 1000 non-spam mails from the public Enron corpus, and 500 newsletters. These mails were manually grouped into batches, resulting in 136 batches with at average 17.7 emails and 598 remaining single mails. Our results are obtained through a cross validation procedure, where each test set contains a non-singular batch and is filled up with randomly drawn emails to a total size of 100 emails. The training data consist of nine sets of 100 emails each, sampled randomly from the remaining emails.

Each of the obtained models is applied to the test sets, using either the approximative clustering based on the linear program, the sequential clustering algorithm, or the greedy clustering algorithm by [13]. Figure 1 shows the experimental results of three of the training methods. The integrated learning problem is not tractable for this amount of training data.

In a second experiment, we split each training and each test set in two halves, resulting in 18 sets of 50 emails each for training. That is, the total number of training emails remains the same but the integrated learning problem becomes tractable. Figure 2 shows the results for this setting.

In both experiments, the LP-decoding strategy and the greedy clustering algorithm perform equally well. By contrast, the sequential clustering performs significantly worse in most of the cases according to a paired t-Test on a 5% confidence level. However, this loss in performance comes with a gain in execution time that is linear in the number of examples (see Table 2).

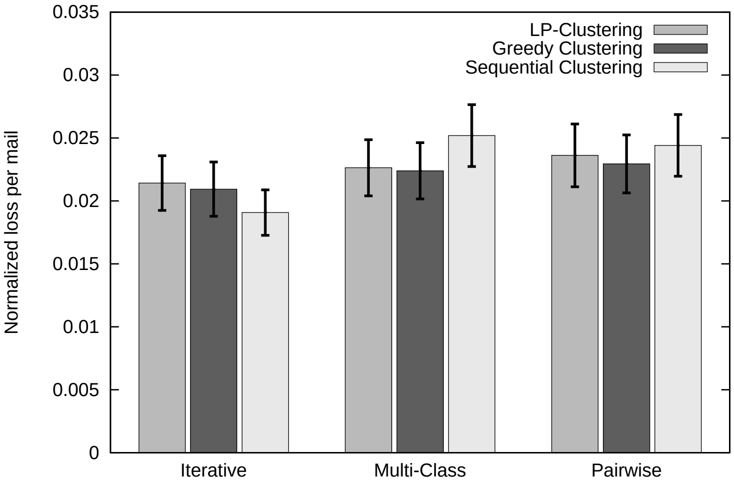

Fig. 1. Average loss and standard errors for $m = 100$.

Figure 3 details which fraction of the error is caused by the decoding and which by the learning algorithm. The dashed area indicates the error caused by the training method. We quantify this error by counting the number of different edges in the true and the predicted similarity matrix, respectively. The additional error of the subsequent decoding is indicated for all three decoding strategies.

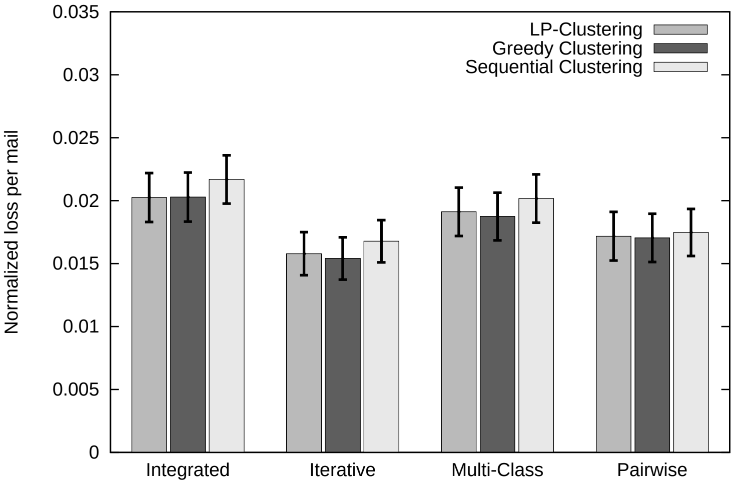

Fig. 2. Average loss and standard errors for $m = 50$.

Except for the sequential decoding, the multi-class optimization leads to correct clusterings that fulfill the transitivity constraints between triples of nodes. On the contrary, the solution of the pairwise optimization has the lowest error but fails to satisfy these transitivity constraints. Neither decoding strategy can compensate the errors.

Table 2. Execution time of the decoding strategies in seconds.

Window size m	25	50	100	200
LP-Clustering	$2.3 \cdot 10^{-1}$	$6.4 \cdot 10^{0}$	$4.0 \cdot 10^{2}$	
Greedy Clustering	$6.4 \cdot 10^{-4}$	$2.5 \cdot 10^{-3}$	$9.9 \cdot 10^{-3}$	$4.0 \cdot 10^{-2}$
Sequential Clustering	$1.5 \cdot 10^{-5}$	$2.9 \cdot 10^{-5}$	$5.6 \cdot 10^{-5}$	$1.1 \cdot 10^{-4}$

6 Conclusion

We devised three large margin approaches to supervised clustering of sequential data. The integrated approach has at least cubic execution time and can be solved directly for small training sets. Treating the problem as multi-class classification allowed us to use larger data sets. The pairwise classification approach is a rough but fast approximation of the original problem.

Experimental results were carried out on all combinations of learning algorithms and decoding strategies in our discourse area. The results showed that the LP-decoding performs equally well as the greedy algorithm presented in [13].

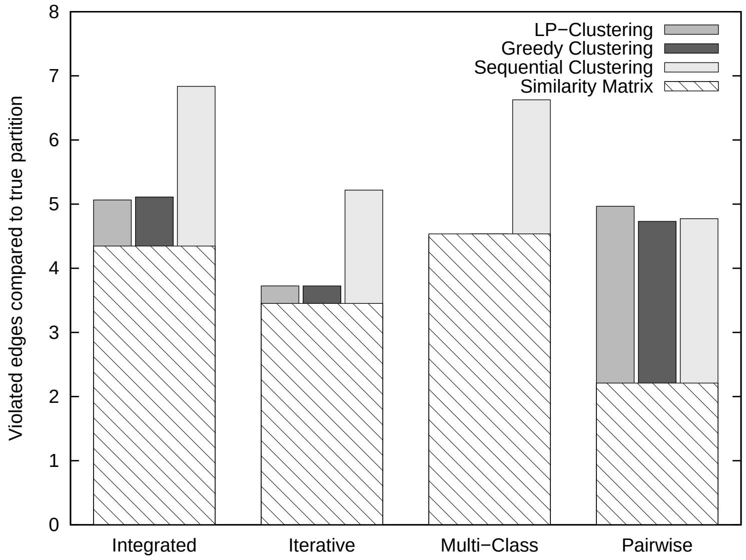

Fig. 3. Fraction of the loss induced by the learning algorithm (similarity matrix) and the decoding.

However, both methods are computationally expensive. The sequential decoding makes use of the sequential nature of the data and leads to slightly increased losses.

References

1. Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden Markov support vector machines. In *Proceedings of the International Conference on Machine Learning*, 2003.
2. Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*, 2002.
3. Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.
4. Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. Learning distance functions using equivalence relations. In *ICML '03: Proceedings of the Twentieth International Conference on Machine Learning*, 2003.
5. Mikhail Bilenko and Raymond J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 2003.
6. Sergey Brin, James Davis, and Hector García-Molina. Copy detection mechanisms for digital documents. In *Proceedings of the International Conference on Management of Data*, 1995.

7. Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, 2003.
8. J. Cooper, A. Coden, and E. Brown. Detecting similar documents using salient terms. In *Proceedings of the International Conference on Information and Knowledge Management*, 2002.
9. J. Cooper, A. Coden, and E. Brown. A novel method for detecting similar documents. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, 2002.
10. Erik D. Demaine and Nicole Immorlica. Correlation clustering with partial information. In *Proceedings of the 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems and 7th International Workshop on Randomization and Approximation Techniques in Computer Science*, 2003.
11. Dotan Emanuel and Amos Fiat. Correlation clustering – minimizing disagreements on arbitrary weighted graphs. *Lecture Notes in Computer Science*, 2832:208–220, 2003.
12. Michael Fink, Shai Shalev-Shwartz, Yoram Singer, and Shimon Ullman. Online multiclass learning by interclass hypothesis sharing. In *ICML '06: Proceedings of the 23nd international conference on Machine learning*, 2006.
13. Thomas Finley and Thorsten Joachims. Supervised clustering with support vector machines. In *Proceedings of the International Conference on Machine Learning*, 2005.
14. Aleksander Kolcz, Abdur Chowdhury, and Joshua Alspector. The impact of feature selection on signature-driven spam detection. In *Proceedings of the First Conference on Email and Anti-Spam*, 2004.
15. J. Lafferty, X. Zhu, and Y. Liu. Kernel conditional random fields: representation and clique selection. In *Proc. of the International Conference on Machine Learning*, 2004.
16. W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:622–626, 1971.
17. Chaitanya Swamy. Correlation clustering: maximizing agreements via semidefinite programming. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, 2004.
18. B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *Advances in Neural Information Processing Systems*, 2004.
19. Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin. Learning structured prediction models: a large margin approach. In *Proceedings of the International Conference on Machine Learning*, 2005.
20. I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.
21. Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schr&#246;dl. Constrained k-means clustering with background knowledge. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, San Francisco, CA, USA, 2001.
22. Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems.* The MIT Press, 2002.
23. Feng Zhou, Li Zhuang, Ben Y. Zhao, Ling Huang, Anthony D. Joseph, and John Kubiatowicz. Approximate object location and spam filtering on peer-to-peer systems. In *Middleware*, 2003.