Bounding and Comparing Methods for Correlation Clustering Beyond ILP

Micha Elsner and Warren Schudy
Department of Computer Science
Brown University
Providence, RI 02912
{melsner,ws}@cs.brown.edu

Abstract

We evaluate several heuristic solvers for correlation clustering, the NP-hard problem of partitioning a dataset given pairwise affinities between all points. We experiment on two practical tasks, document clustering and chat disentanglement, to which ILP does not scale. On these datasets, we show that the clustering objective often, but not always, correlates with external metrics, and that local search always improves over greedy solutions. We use semi-definite programming (SDP) to provide a tighter bound, showing that simple algorithms are already close to optimality.

1 Introduction

Correlation clustering is a powerful technique for discovering structure in data. It operates on the pairwise relationships between datapoints, partitioning the graph to minimize the number of unrelated pairs that are clustered together, plus the number of related pairs that are separated. Unfortunately, this minimization problem is NP-hard (Ailon et al., 2008). Practical work has adopted one of three strategies for solving it. For a few specific tasks, one can restrict the problem so that it is efficiently solvable. In most cases, however, this is impossible. Integer linear programming (ILP) can be used to solve the general problem optimally, but only when the number of data points is small. Beyond a few hundred points, the only available solutions are heuristic or approximate.

In this paper, we evaluate a variety of solutions for correlation clustering on two realistic NLP tasks, text topic clustering and chat disentanglement, where typical datasets are too large for ILP to find a solution. We show, as in previous work on consensus clustering (Goder and Filkov, 2008), that local search can improve the solutions found by commonly-used methods. We investigate the relationship between the clustering objective and external evaluation metrics such as F-score and one-to-one overlap, showing that optimizing the objective is usually a reasonable aim, but that other measurements like number of clusters found should sometimes be used to reject pathological solutions. We prove that the best heuristics are quite close to optimal, using the first implementation of the semi-definite programming (SDP) relaxation to provide tighter bounds.

The specific algorithms we investigate are, of course, only a subset of the large number of possible solutions, or even of those proposed in the literature. We chose to test a few common, efficient algorithms that are easily implemented. Our use of a good bounding strategy means that we do not need to perform an exhaustive comparison; we will show that, though the methods we describe are not perfect, the remaining improvements possible with any algorithm are relatively small.

2 Previous Work

Correlation clustering was first introduced by Ben-Dor et al. (1999) to cluster gene expression patterns. The correlation clustering approach has several strengths. It does not require users to specify a parametric form for the clusters, nor to pick the number of clusters. Unlike fully unsupervised clustering methods, it can use training data to optimize the pairwise classifier, but unlike classification, it does not require samples from the specific clusters found in the test data. For instance, it can use messages about cars to learn a similarity function that can then be applied to messages about atheism.

Correlation clustering is a standard method for coreference resolution. It was introduced to the area by Soon et al. (2001), who describe the first-link heuristic method for solving it. Ng and Cardie (2002) extend this work with better features, and develop the best-link heuristic, which finds better solutions. McCallum and Wellner (2004) explicitly describe the problem as correlation clustering and use an approximate technique (Bansal et al., 2004) to enforce transitivity. Recently Finkel and Manning (2008) show that the optimal ILP solution outperforms the first and best-link methods. Cohen and Richman (2002) experiment with various heuristic solutions for the cross-document coreference task of grouping references to named entities.

Finally, correlation clustering has proven useful in several discourse tasks. Barzilay and Lapata (2006) use it for content aggregation in a generation system. In Malioutov and Barzilay (2006), it is used for topic segmentation—since segments must be contiguous, the problem can be solved in polynomial time. Elsner and Charniak (2008) address the related problem of disentanglement (which we explore in Section 5.3), doing inference with the voting greedy algorithm.

Bertolacci and Wirth (2007), Goder and Filkov (2008) and Gionis et al. (2007) conduct experiments on the closely related problem of *consensus clustering*, often solved by reduction to correlation clustering. The input to this problem is a set of clusterings; the output is a "median" clustering which minimizes the sum of (Rand) distance to the inputs. Although these papers investigate some of the same algorithms we use, they use an unrealistic lower bound, and so cannot convincingly evaluate absolute performance. Gionis et al. (2007) give an external evaluation on some UCI datasets, but this is somewhat unconvincing since their metric, the *impurity index*, which is essentially precision ignoring recall, gives a perfect score to the all-singletons clustering. The other two papers are based on objective values, not external metrics.[1]

A variety of approximation algorithms for correlation clustering with worst-case theoretical guarantees have been proposed: (Bansal et al., 2004; Ailon et al., 2008; Demaine et al., 2006; Charikar et al., 2005; Giotis and Guruswami, 2006). Researchers including (Ben-Dor et al., 1999; Joachims and Hopcroft, 2005; Mathieu and Schudy, 2008) study correlation clustering theoretically when the input is generated by randomly perturbing an unknown ground truth clustering.

3 Algorithms

We begin with some notation and a formal definition of the problem. Our input is a complete, undirected graph G with n nodes; each edge in the graph has a probability p_{ij} reflecting our belief as to whether nodes i and j come from the same cluster. Our goal is to find a clustering, defined as a new graph G' with edges $x_{ij} \in \{0, 1\}$, where if $x_{ij} = 1$, nodes i and j are assigned to the same cluster. To make this consistent, the edges must define an equivalence relationship: $x_{ii} = 1$ and $x_{ij} = x_{jk} = 1$ implies $x_{ij} = x_{ik}$.

Our objective is to find a clustering as consistent as possible with our beliefs—edges with high probability should not cross cluster boundaries, and edges with low probability should. We define w^+_{ij} as the cost of cutting an edge whose probability is p_{ij} and w^-_{ij} as the cost of keeping it. Mathematically, this objective can be written (Ailon et al., 2008; Finkel and Manning, 2008) as:

$$\min \sum_{ij:i<j} x_{ij} w^-_{ij} + (1 - x_{ij}) w^+_{ij}. \tag{1}$$

There are two plausible definitions for the costs w^+ and w^-, both of which have gained some support in the literature. We can take $w^+_{ij} = p_{ij}$ and $w^-_{ij} = 1 - p_{ij}$ (*additive* weights) as in (Ailon et al., 2008) and others, or $w^+_{ij} = \log(p_{ij}), w^-_{ij} = \log(1 - p_{ij})$ (*logarithmic* weights) as in (Finkel and Manning, 2008). The logarithmic scheme has a tenuous mathematical justification, since it selects a maximum-likelihood clustering under the assumption that the

[1] Bertolacci and Wirth (2007) gave normalized mutual information for one algorithm and data set, but almost all of their results study objective value only.

p_{ij} are independent and identically distributed given the status of the edge ij in the true clustering. If we obtain the p_{ij} using a classifier, however, this assumption is obviously untrue—some nodes will be easy to link, while others will be hard—so we evaluate the different weighting schemes empirically.

3.1 Greedy Methods

We use four greedy methods drawn from the literature; they are both fast and easy to implement. All of them make decisions based on the *net weight* $w^{\pm}_{ij} = w^{+}_{ij} - w^{-}_{ij}$.

These algorithms step through the nodes of the graph according to a permutation π. We try 100 random permutations for each algorithm and report the run which attains the best objective value (typically this is slightly better than the average run; we discuss this more in the experimental sections). To simplify the pseudocode we label the vertices $1, 2, \ldots n$ in the order specified by π. After this relabeling $\pi(i) = i$ so π need not appear explicitly in the algorithms.

Three of the algorithms are given in Figure 1. All three algorithms start with the empty clustering and add the vertices one by one. The BEST algorithm adds each vertex i to the cluster with the strongest $w^{\pm}$ connecting to i, or to a new singleton if none of the $w^{\pm}$ are positive. The FIRST algorithm adds each vertex i to the cluster containing the most recently considered vertex j with $w^{\pm}_{ij} > 0$. The VOTE algorithm adds each vertex to the cluster that minimizes the correlation clustering objective, i.e. to the cluster maximizing the total net weight or to a singleton if no total is positive.

Ailon et al. (2008) introduced the PIVOT algorithm, given in Figure 2, and proved that it is a 5-approximation if $w^{+}_{ij} + w^{-}_{ij} = 1$ for all i, j and π is chosen randomly. Unlike BEST, VOTE and FIRST, which build clusters vertex by vertex, the PIVOT algorithm creates each new cluster in its final form. This algorithm repeatedly takes an unclustered pivot vertex and creates a new cluster containing that vertex and all unclustered neighbors with positive weight.

3.2 Local Search

We use the straightforward local search previously used by Gionis et al. (2007) and Goder and Filkov

```
$k \leftarrow 0$ // number of clusters created so far
for $i = 1 \ldots n$ do
    for $c = 1 \ldots k$ do
        if BEST then
            $Quality_c \leftarrow \max_{j \in C[c]} w^{\pm}_{ij}$
        else if FIRST then
            $Quality_c \leftarrow \max_{j \in C[c] : w^{\pm}_{ij} > 0} j$
        else if VOTE then
            $Quality_c \leftarrow \sum_{j \in C[c]} w^{\pm}_{ij}$
    $c^* \leftarrow \arg\max_{1 \le c \le k} Quality_c$
    if $Quality_{c^*} > 0$ then
        $C[c^*] \leftarrow C[c^*] \cup \{i\}$
    else
        $C[k{+}{+}] \leftarrow \{i\}$ // form a new cluster
```

Figure 1: BEST/FIRST/VOTE algorithms

```
$k \leftarrow 0$ // number of clusters created so far
for $i = 1 \ldots n$ do
    $P \leftarrow \bigcup_{1 \le c \le k} C[c]$ // Vertices already placed
    if $i \notin P$ then
        $C[k{+}{+}] \leftarrow \{i\} \cup \{ i < j \le n :$
        $j \notin P$ and $w^{\pm}_{ij} > 0 \}$
```

Figure 2: PIVOT algorithm by Ailon et al. (2008)

(2008). The allowed *one element moves* consist of removing one vertex from a cluster and either moving it to another cluster or to a new singleton cluster. The best one element move (BOEM) algorithm repeatedly makes the most profitable best one element move until a local optimum is reached. *Simulated Annealing* (SA) makes a random single-element move, with probability related to the difference in objective it causes and the current temperature. Our annealing schedule is exponential and designed to attempt $2000n$ moves for n nodes. We initialize the local search either with all nodes clustered together, or at the clustering produced by one of our greedy algorithms (in our tables, the latter is written, eg. PIVOT/BOEM, if the greedy algorithm is PIVOT).

4 Bounding with SDP

Although comparing different algorithms to one another gives a good picture of relative performance, it is natural to wonder how well they do in an absolute sense—how they compare to the optimal solution.

For very small instances, we can actually find the optimum using ILP, but since this does not scale beyond a few hundred points (see Section 5.1), for realistic instances we must instead bound the optimal value. Bounds are usually obtained by solving a *relaxation* of the original problem: a simpler problem with the same objective but fewer constraints.

The bound used in previous work (Goder and Filkov, 2008; Gionis et al., 2007; Bertolacci and Wirth, 2007), which we call the *trivial bound*, is obtained by ignoring the transitivity constraints entirely. To optimize, we link ($x_{ij} = 1$) all the pairs where w^+_{ij} is larger than w^-_{ij}; since this solution is quite far from being a clustering, the bound tends not to be very tight.

To get a better idea of how good a real clustering can be, we use a semi-definite programming (SDP) relaxation to provide a better bound. Here we motivate and define this relaxation.

One can picture a clustering geometrically by associating cluster c with the standard basis vector $e_c = (\underbrace{0, 0, \dots, 0,}_{c-1} 1, \underbrace{0, \dots, 0}_{n-c}) \in \mathbb{R}^n$. If object i is in cluster c then it is natural to associate i with the vector $r_i = e_c$. This gives a nice geometric picture of a clustering, with objects i and j in the same cluster if and only if $r_i = r_j$. Note that the dot product $r_i \bullet r_j$ is 1 if i and j are in the same cluster and 0 otherwise. These ideas yield a simple reformulation of the correlation clustering problem:

$$\begin{gathered}\min_r \textstyle\sum_{i,j:i<j} (r_i \bullet r_j) w^-_{ij} + (1 - r_j \bullet r_j) w^+_{ij} \\ \text{s.t. } \forall i \;\; \exists c : r_i = e_c\end{gathered}$$

To get an efficiently computable lower-bound we relax the constraints that the r_is are standard basis vectors, replacing them with two sets of constraints: $r_i \bullet r_i = 1$ for all i and $r_i \bullet r_j \geq 0$ for all i, j.

Since the r_i only appear as dot products, we can rewrite in terms of $x_{ij} = r_i \bullet r_j$. However, we must now constrain the x_{ij} to be the dot products of some set of vectors in $\mathbb{R}^n$. This is true if and only if the symmetric matrix $X = \{x_{ij}\}_{ij}$ is *positive semi-definite*. We now have the standard semi-definite programming (SDP) relaxation of correlation clustering (e.g. (Charikar et al., 2005; Mathieu and Schudy, 2008)):

$$\begin{gathered}\min_x \textstyle\sum_{i,j:i<j} x_{ij} w^-_{ij} + (1 - x_{ij}) w^+_{ij} \\ \text{s.t.} \begin{cases} x_{ii} = 1 & \forall i \\ x_{ij} \geq 0 & \forall i, j \\ X = \{x_{ij}\}_{ij} \text{ PSD} \end{cases} .\end{gathered}$$

This SDP has been studied theoretically by a number of authors; we mention just two here. Charikar et al. (2005) give an approximation algorithm based on rounding the SDP which is a 0.7664 approximation for the problem of maximizing agreements. Mathieu and Schudy (2008) show that if the input is generated by corrupting the edges of a ground truth clustering B independently, then the SDP relaxation value is within an additive $O(n\sqrt{n})$ of the optimum clustering. They further show that using the PIVOT algorithm to round the SDP yields a clustering with value at most $O(n\sqrt{n})$ more than optimal.

5 Experiments

5.1 Scalability

Using synthetic data, we investigate the scalability of the linear programming solver and SDP bound. To find optimal solutions, we pass the complete ILP[2] to CPLEX. This is reasonable for 100 points and solvable for 200; beyond this point it cannot be solved due to memory exhaustion. As noted below, despite our inability to compute the LP bound on large instances, we can sometimes prove that they must be worse than SDP bounds, so we do not investigate LP-solving techniques further.

The SDP has fewer constraints than the ILP ($O(n^2)$ vs $O(n^3)$), but this is still more than many SDP solvers can handle. For our experiments we used one of the few SDP solvers that can handle such a large number of constraints: Christoph Helmberg's ConicBundle library (Helmberg, 2009; Helmberg, 2000). This solver can handle several thousand datapoints. It produces loose lower-bounds (off by a few percent) quickly but converges to optimality quite slowly; we err on the side of inefficiency by running for up to 60 hours. Of course, the SDP solver is only necessary to bound algorithm performance; our solvers themselves scale much better.

[2]Consisting of the objective plus constraints $0 \leq x_{ij} \leq 1$ and triangle inequality (Ailon et al., 2008).

5.2 Twenty Newsgroups

In this section, we test our approach on a typical benchmark clustering dataset, 20 Newsgroups, which contains posts from a variety of Usenet newsgroups such as `rec.motorcycles` and `alt.atheism`. Since our bounding technique does not scale to the full dataset, we restrict our attention to a subsample of 100 messages[3] from each newsgroup for a total of 2000—still a realistically large-scale problem. Our goal is to cluster messages by their newsgroup of origin. We conduct experiments by holding out four newsgroups as a training set, learning a pairwise classifier, and applying it to the remaining 16 newsgroups to form our affinity matrix.[4]

Our pairwise classifier uses three types of features previously found useful in document clustering. First, we bucket all words[5] by their log document frequency (for an overview of TF-IDF see (Joachims, 1997)). For a pair of messages, we create a feature for each bucket whose value is the proportion of shared words in that bucket. Secondly, we run LSA (Deerwester et al., 1990) on the TF-IDF matrix for the dataset, and use the cosine distance between each message pair as a feature. Finally, we use the same type of shared words features for terms in message subjects. We make a training instance for each pair of documents in the training set and learn via logistic regression.

The classifier has an average F-score of 29% and an accuracy of 88%—not particularly good. We should emphasize that the clustering task for 20 newsgroups is much harder than the more common classification task—since our training set is entirely disjoint with the testing set, we can only learn weights on feature categories, not term weights. Our aim is to create realistic-looking data on which to test our clustering methods, not to motivate correlation clustering as a solution to this specific problem. In fact, Zhong and Ghosh (2003) report better results using generative models.

We evaluate our clusterings using three different metrics (see Meila (2007) for an overview of clustering metrics). The *Rand* measure counts the number of pairs of points for which the proposed clustering agrees with ground truth. This is the metric which is mathematically closest to the objective. However, since most points are in different clusters, any solution with small clusters tends to get a high score. Therefore we also report the more sensitive *F-score* with respect to the minority ("same cluster") class. We also report the *one-to-one* score, which measures accuracy over single points. For this metric, we calculate a maximum-weight matching between proposed clusters and ground-truth clusters, then report the overlap between the two.

When presenting objective values, we locate them within the range between the trivial lower bound dis-

Logarithmic Weights

	Obj	Rand	F	1-1
SDP bound	51.1%	-	-	-
VOTE/BOEM	55.8%	93.80	33	41
SA	56.3%	93.56	31	36
PIVOT/BOEM	56.6%	93.63	32	39
BEST/BOEM	57.6%	93.57	31	38
FIRST/BOEM	57.9%	93.65	30	36
VOTE	59.0%	93.41	29	35
BOEM	60.1%	93.51	30	35
PIVOT	100%	90.85	17	27
BEST	138%	87.11	20	29
FIRST	619%	40.97	11	8

Additive Weights

	Obj	Rand	F	1-1
SDP bound	59.0%	-	-	-
SA	63.5%	93.75	32	39
VOTE/BOEM	63.5%	93.75	32	39
PIVOT/BOEM	63.7%	93.70	32	39
BEST/BOEM	63.8%	93.73	31	39
FIRST/BOEM	63.9%	93.58	31	37
BOEM	64.6%	93.65	31	37
VOTE	67.3%	93.35	28	34
PIVOT	109%	90.63	17	26
BEST	165%	87.06	20	29
FIRST	761%	40.46	11	8

Table 1: Score of the solution with best objective for each solver, averaged over newsgroups training sets, sorted by objective.

[3]Available as `mini_newsgroups.tar.gz` from the UCI machine learning repository.

[4]The experiments below are averaged over four disjoint training sets.

[5]We omit the message header, except the subject line, and also discard word types with fewer than 3 occurrences.

cussed in Section 4 and the objective value of the singletons clustering ($x_{ij} = 0, i \neq j$). On this scale, lower is better; 0% corresponds to the trivial bound and 100% corresponds to the singletons clustering. It is possible to find values greater than 100%, since some particularly bad clusterings have objectives worse than the singletons clustering. Plainly, however, real clusterings will not have values as low as 0%, since the trivial bound is so unrealistic.

Our results are shown in Table 1. The best results are obtained using logarithmic weights with VOTE followed by BOEM; reasonable results are also found using additive weights, and annealing, VOTE or PIVOT followed by BOEM. On its own, the best greedy scheme is VOTE, but all of them are substantially improved by BOEM. First-link is by far the worst. Our use of the SDP lower bound rather than the trivial lower-bound of 0% reduces the gap between the best clustering and the lower bound by over a factor of ten. It is easy to show that the LP relaxation can obtain a bound of at most 50%[6]—the SDP beats the LP in both runtime and quality!

We analyze the correlation between objective values and metric values, averaging Kendall's tau[7] over the four datasets (Table 2). Over the entire dataset, correlations are generally good (large and negative), showing that optimizing the objective is indeed a useful way to find good results. We also examine correlations for the solutions with objective values within the top 10%. Here the correlation is much poorer; selecting the solution with the best objective value will not necessarily optimize the metric, although the correspondence is slightly better for the log-weights scheme. The correlations do exist, however, and so the solution with the best objective value is typically slightly better than the median.

In Figure 3, we show the distribution of one-to-one scores obtained (for one specific dataset) by the best solvers. From this diagram, it is clear that log-weights and VOTE/BOEM usually obtain the best scores for this metric, since the median is higher than other solvers' upper quartile scores. All solvers have quite high variance, with a range of about 2% between quartiles and 4% overall. We omit the F-score plot, which is similar, for space reasons.

[6]The solution $x_{ij} = \frac{1}{2}\mathbb{1}\left(w_{ij}^{-} > w_{ij}^{+}\right)$ for $i < j$ is feasible in the LP.

[7]The standard Pearson correlation coefficient is less robust to outliers, which causes problems for this data.

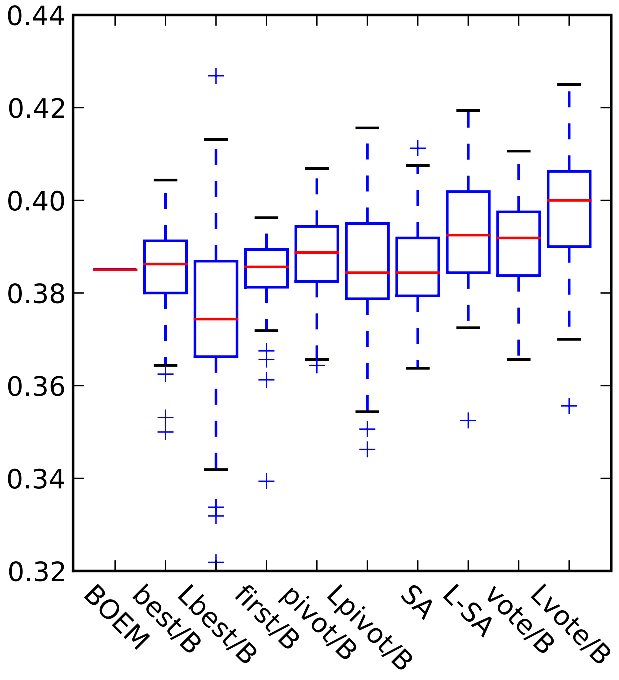

Figure 3: Box-and-whisker diagram (outliers as $+$) for one-to-one scores obtained by the best few solvers on a particular newsgroup dataset. L means using log weights. B means improved with BOEM.

	Rand	F	1-1
Log-wt	-.60	-.73	-.71
Top 10 %	-.14	-.22	-.24
Add-wt	-.60	-.67	-.65
Top 10 %	-.13	-.15	-.14

Table 2: Kendall's tau correlation between objective and metric values, averaged over newsgroup datasets, for all solutions and top 10% of solutions.

5.3 Chat Disentanglement

In the disentanglement task, we examine data from a shared discussion group where many conversations are occurring simultaneously. The task is to partition the utterances into a set of conversations. This task differs from newsgroup clustering in that data points (utterances) have an inherent linear order. Ordering is typical in discourse tasks including topic segmentation and coreference resolution.

We use the annotated dataset and pairwise classifier made available by Elsner and Charniak (2008);[8] this study represents a competitive baseline, although more recently Wang and Oard (2009) have improved it. Since this classifier is ineffective at linking utterances more than 129 seconds apart, we treat all decisions for such utterances as abstentions, $p = .5$. For utterance pairs on which it does make a decision, the classifier has a reported accuracy of 75% with an F-score of 71%.

As in previous work, we run experiments on the 800-utterance test set and average metrics over 6 test annotations. We evaluate using the three metrics reported by previous work. Two node-counting metrics measure global accuracy: *one-to-one match* as explained above, and *Shen's F* (Shen et al., 2006): $F = \sum_i \frac{n_i}{n} \max_j(F(i,j))$. Here i is a gold conversation with size n_i and j is a proposed conversation with size n_j, sharing n_{ij} utterances; $F(i,j)$ is the harmonic mean of precision ($\frac{n_{ij}}{n_j}$) and recall ($\frac{n_{ij}}{n_i}$). A third metric, the *local agreement*, counts edgewise agreement for pairs of nearby utterances, where nearby means "within three utterances."

In this dataset, the SDP is a more moderate improvement over the trivial lower bound, reducing the gap between the best clustering and best lower bound by a factor of about 3 (Table 3).

Optimization of the objective does not correspond to improvements in the global metrics (Table 3); for instance, the best objectives are attained with FIRST/BOEM, but VOTE/BOEM yields better one-to-one and F scores. Correlation between the objective and these global metrics is extremely weak (Table 5). The local metric is somewhat correlated.

Local search does improve metric results for each particular greedy algorithm. For instance, when BOEM is added to VOTE (with log weights), one-to-one increases from 44% to 46%, local from 72% to 73% and F from 48% to 50%. This represents a moderate improvement on the inference scheme described in Elsner and Charniak (2008). They use voting with additive weights, but rather than performing multiple runs over random permutations, they process utterances in the order they occur. (We experimented with processing in order; the results are unclear, but there is a slight trend toward worse performance, as in this case.) Their results (also shown in the table) are 41% one-to-one, 73% local and .44% F-score.[9] Our improvement on the global metrics (12% relative improvement in one-to-one, 13% in F-score) is modest, but was achieved with better inference on exactly the same input.

Since the objective function fails to distinguish good solutions from bad ones, we examine the types of solutions found by different methods in the hope of explaining why some perform better than others. In this setting, some methods (notably local search run on its own or from a poor starting point) find far fewer clusters than others (Table 4; log weights not shown but similar to additive). Since the classifier abstains for utterances more than 129 seconds apart, the objective is unaffected if very distant utterances are linked on the basis of little or no evidence; this is presumably how such large clusters form. (This raises the question of whether abstentions should be given weaker links with $p < .5$. We leave this for future work.) Algorithms which find reasonable numbers of clusters (VOTE, PIVOT, BEST and local searches based on these) all achieve good metric scores, although there is still no reliable way to find the best solution among this set of methods.

6 Conclusions

It is clear from these results that heuristic methods can provide good correlation clustering solutions on datasets far too large for ILP to scale. The particular solver chosen[10] has a substantial impact on the quality of results obtained, in terms of external metrics as well as objective value.

For general problems, our recommendation is to use log weights and run VOTE/BOEM. This algorithm is fast, achieves good objective values, and yields good metric scores on our datasets. Although objective values are usually only weakly correlated with metrics, our results suggest that slightly better scores can be obtained by running the algorithm many times and returning the solution with the best objective. This may be worth trying even when the datapoints are inherently ordered, as in chat.

[8]Downloaded from `cs.brown.edu/~melsner`

[9]The F-score metric is not used in Elsner and Charniak (2008); we compute it ourselves on the result produced by their software.

[10]Our C++ correlation clustering software and SDP bounding package are available for download from `cs.brown.edu/~melsner`.

Log Weights

	Obj	1-1	Loc_3	Shen F
SDP bound	13.0%	-	-	-
FIRST/BOEM	19.3%	41	**74**	44
VOTE/BOEM	20.0%	**46**	**73**	**50**
SA	20.3%	42	**73**	45
BEST/BOEM	21.3%	43	**73**	47
BOEM	21.5%	22	72	21
PIVOT/BOEM	22.0%	**45**	72	**50**
VOTE	26.3%	44	72	48
BEST	37.1%	40	67	44
PIVOT	44.4%	39	66	44
FIRST	58.3%	39	62	41

Additive Weights

	Obj	1-1	Loc_3	Shen F
SDP bound	16.2%	-	-	-
FIRST/BOEM	21.7%	40	**73**	44
BOEM	22.3%	22	**73**	20
BEST/BOEM	22.7%	44	**74**	**49**
VOTE/BOEM	23.3%	**46**	**73**	**50**
SA	23.8%	41	72	46
PIVOT/BOEM	24.8%	**46**	73	**50**
VOTE	30.5%	44	71	**49**
EC '08	-	41	**73**	44
BEST	42.1%	43	69	47
PIVOT	48.4%	38	67	44
FIRST	69.0%	40	59	41

Table 3: Score of the solution with best objective found by each solver on the chat test dataset, averaged over 6 annotations, sorted by objective.

Whatever algorithm is used to provide an initial solution, we advise the use of local search as a post-process. BOEM always improves both objective and metric values over its starting point.

The objective value is not always sufficient to select a good solution (as in the chat dataset). If possible, experimenters should check statistics like the number of clusters found to make sure they conform roughly to expectations. Algorithms that find far too many or too few clusters, regardless of objective, are unlikely to be useful. This type of problem can be especially dangerous if the pairwise classifier abstains for many pairs of points.

SDP provides much tighter bounds than the trivial bound used in previous work, although how much

	Num clusters
Max human annotator	*128*
PIVOT	122
VOTE	99
PIVOT/BOEM	89
VOTE/BOEM	86
Mean human annotator	*81*
BEST	70
FIRST	70
Elsner and Charniak (2008)	*63*
BEST/BOEM	62
SA	57
FIRST/BOEM	54
Min human annotator	*50*
BOEM	7

Table 4: Average number of clusters found (using additive weights) for chat test data.

	1-1	Loc_3	Shen F
Log-wt	-.40	-.68	-.35
Top 10 %	.14	-.15	.15
Add-wt	-.31	-.67	-.25
Top 10 %	-.07	-.22	.13

Table 5: Kendall's tau correlation between objective and metric values for the chat test set, for all solutions and top 10% of solutions.

tighter varies with dataset (about 12 times smaller for newsgroups, 3 times for chat). This bound can be used to evaluate the absolute performance of our solvers; the VOTE/BOEM solver whose use we recommend is within about 5% of optimality. Some of this 5% represents the difference between the bound and optimality; the rest is the difference between the optimum and the solution found. If the bound were exactly optimal, we could expect a significant improvement on our best results, but not a very large one—especially since correlation between objective and metric values grows weaker for the best solutions. While it might be useful to investigate more sophisticated local searches in an attempt to close the gap, we do not view this as a priority.

Acknowledgements

We thank Christoph Helmberg, Claire Mathieu and three reviewers.

References

Nir Ailon, Moses Charikar, and Alantha Newman. 2008. Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM*, 55(5):Article No. 23.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. 2004. Correlation clustering. *Machine Learning*, 56(1-3):89–113.

Regina Barzilay and Mirella Lapata. 2006. Aggregation via set partitioning for natural language generation. In *HLT-NAACL*.

Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. 1999. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3-4):281–297.

Michael Bertolacci and Anthony Wirth. 2007. Are approximation algorithms for consensus clustering worthwhile? In *SDM '07: Procs. 7^{th} SIAM International Conference on Data Mining*.

Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. 2005. Clustering with qualitative information. *J. Comput. Syst. Sci.*, 71(3):360–383.

William W. Cohen and Jacob Richman. 2002. Learning to match and cluster large high-dimensional data sets for data integration. In *KDD '02*, pages 475–480. ACM.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

Erik D. Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. 2006. Correlation clustering in general weighted graphs. *Theor. Comput. Sci.*, 361(2):172–187.

Micha Elsner and Eugene Charniak. 2008. You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of ACL-08: HLT*, pages 834–842, Columbus, Ohio, June. Association for Computational Linguistics.

Jenny Rose Finkel and Christopher D. Manning. 2008. Enforcing transitivity in coreference resolution. In *Proceedings of ACL-08: HLT, Short Papers*, pages 45–48, Columbus, Ohio, June. Association for Computational Linguistics.

Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas. 2007. Clustering aggregation. *ACM Trans. on Knowledge Discovery from Data*, 1(1):Article 4.

Ioannis Giotis and Venkatesan Guruswami. 2006. Correlation clustering with a fixed number of clusters. *Theory of Computing*, 2(1):249–266.

Andrey Goder and Vladimir Filkov. 2008. Consensus clustering algorithms: Comparison and refinement. In *ALENEX '08: Procs. 10^{th} Workshop on Algorithm Enginering and Experiments*, pages 109–117. SIAM.

Cristoph Helmberg. 2000. Semidefinite programming for combinatorial optimization. Technical Report ZR-00-34, Konrad-Zuse-Zentrum für Informationstechnik Berlin.

Cristoph Helmberg, 2009. *The ConicBundle Library for Convex Optimization*. Ver. 0.2i from `http://www-user.tu-chemnitz.de/~helmberg/ConicBundle/`.

Thorsten Joachims and John Hopcroft. 2005. Error bounds for correlation clustering. In *ICML '05*, pages 385–392, New York, NY, USA. ACM.

Thorsten Joachims. 1997. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *International Conference on Machine Learning (ICML)*, pages 143–151.

Igor Malioutov and Regina Barzilay. 2006. Minimum cut model for spoken lecture segmentation. In *ACL*. The Association for Computer Linguistics.

Claire Mathieu and Warren Schudy. 2008. Correlation clustering with noisy input. Unpublished manuscript available from http://www.cs.brown.edu/∼ws/papers/clustering.pdf.

Andrew McCallum and Ben Wellner. 2004. Conditional models of identity uncertainty with application to noun coreference. In *Proceedings of the 18th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 905–912. MIT Press.

Marina Meila. 2007. Comparing clusterings–an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, May.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111.

Dou Shen, Qiang Yang, Jian-Tao Sun, and Zheng Chen. 2006. Thread detection in dynamic text message streams. In *SIGIR '06*, pages 35–42, New York, NY, USA. ACM.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Lidan Wang and Douglas W. Oard. 2009. Context-based message expansion for disentanglement of interleaved text conversations. In *Proceedings of NAACL-09 (to appear)*.

Shi Zhong and Joydeep Ghosh. 2003. Model-based clustering with soft balancing. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, page 459, Washington, DC, USA. IEEE Computer Society.