Measuring Constraint-Set Utility for Partitional Clustering Algorithms

Ian Davidson[1], Kiri L. Wagstaff[2], and Sugato Basu[3]

[1] State University of New York, Albany, NY 12222
davidson@cs.albany.edu
[2] Jet Propulsion Laboratory, Pasadena, CA 91109
kiri.wagstaff@jpl.nasa.gov
[3] SRI International, Menlo Park, CA 94025
basu@ai.sri.com

Abstract. Clustering with constraints is an active area of machine learning and data mining research. Previous empirical work has convincingly shown that adding constraints to clustering improves performance, with respect to the true data labels. However, in most of these experiments, results are averaged over different randomly chosen constraint sets, thereby masking interesting properties of individual sets. We demonstrate that constraint sets vary significantly in how useful they are for constrained clustering; some constraint sets can actually decrease algorithm performance. We create two quantitative measures, informativeness and coherence, that can be used to identify useful constraint sets. We show that these measures can also help explain differences in performance for four particular constrained clustering algorithms.

1 Introduction

The last five years have seen extensive work on incorporating instance-level constraints into clustering methods [1,2,3,4,5]. Constraints provide guidance about the desired partition and make it possible for clustering algorithms to increase their performance, sometimes dramatically. Instance-level constraints specify that two items must be placed into the same cluster (must-link, ML) or different clusters (cannot-link, CL). This semi-supervised approach has led to improved performance for several UCI data sets as well as for real-world applications, such as person identification from surveillance camera clips [5], noun phrase coreference resolution and GPS-based map refinement [6], and landscape detection from hyperspectral data [7].

Constraints can be generated from background knowledge about the data set [6,8] or from a subset of the data with known labels [1,2,3,4,5]. Based on the strong positive empirical results that have been reported, the opinion of the community is that constraints help improve clustering performance with respect to accuracy, as measured on the set of extrinsic labels used to generate the constraints. While we might expect that different constraint sets would contribute more or less to improving clustering accuracy, we have found that, surprisingly,

J. Fürnkranz, T. Scheffer, and M. Spiliopoulou (Eds.): PKDD 2006, LNAI 4213, pp. 115–126, 2006.

some constraint sets actually *decrease* clustering performance. We present experimental evidence of this phenomenon in Section 2. We observe that constraints can have ill effects even when they are generated directly from the data labels that are used to evaluate accuracy, so this behavior is not caused by noise or errors in the constraints. Instead, it is a result of the interaction between a given set of constraints and the algorithm being used.

The two major contributions of this work are:

1. The first explicit identification of the adverse effects constraints can have on the clustering process, and
2. The first attempt to characterize constraint set utility to explain clustering performance.

The key question that this work addresses is: *Why do some constraint sets increase clustering accuracy while others have no effect or even decrease accuracy?* We propose two measures, *informativeness* and *coherence*, that capture relevant properties of constraint sets (Section 3). These measures provide insight into the effect a given constraint set has for a specific constrained clustering algorithm. In experiments on several data sets, we find that in general, constraint sets with high informativeness and coherence are most beneficial, and that this trend holds for four different algorithms (Section 4). Finally, we use the CMU Face Images data set [9] to show visual examples of informative and coherent constraint sets.

2 Motivation: Constraints Can Decrease Performance

The operating assumption behind all constrained clustering methods is that the constraints provide information about the true (desired) partition, and that more information will increase the agreement between the output partition and the true partition. Therefore, if the constraints originate from the true partition labels, and they are noise-free, then it should not be possible for them to decrease clustering accuracy. However, as we show in this section, this assumption does not always hold.

The experimental methodology adopted by most previous work in constrained clustering involves generating constraints by repeatedly drawing pairs of data points at random from the labeled subset (which may be the entire data set). If the labels of the points in a pair agree, then an ML constraint is generated; otherwise, a CL constraint is generated. Once the set of constraints has been generated, the constrained clustering algorithm is run several times and the average clustering accuracy is reported. Learning curves are produced by repeating this process for different constraint set sizes, and the typical result is that, on average, when more constraints are provided, clustering accuracy increases [1,2,3,4,5,6,7,8]. However, the focus on characterizing *average* behavior has obscured some interesting and exceptional behavior that results from specific constraint sets. In this work, we will empirically demonstrate such cases and provide insight into the reasons for this behavior.

We begin by examining the behavior of four different constrained clustering algorithms on several standard clustering problems. The two major types of

Table 1. Average performance (Rand Index) of four constrained clustering algorithms, for 1000 trials with 25 randomly selected constraints. The best result for each algorithm/data set combination is in bold.

	Algorithm							
	CKM		**PKM**		**MKM**		**MPKM**	
Data Set	Unconst.	Const.	Unconst.	Const.	Unconst.	Const.	Unconst.	Const.
Glass	69.0	**69.4**	43.4	**68.8**	39.5	**56.6**	39.5	**67.8**
Ionosphere	58.6	**58.7**	58.8	**58.9**	**58.9**	**58.9**	**58.9**	**58.9**
Iris	84.7	**87.8**	84.3	**88.3**	88.0	**93.6**	88.0	**91.8**
Wine	70.2	**70.9**	71.7	**72.0**	**93.3**	91.3	**93.3**	90.6

constrained clustering techniques are (a) direct constraint satisfaction and (b) metric learning. The techniques of the first category try to satisfy the constraints during the clustering algorithm; the latter techniques treat an ML (or CL) constraint as specifying that the two points in the constraint and their surrounding points should be nearby (or well separated) and tries to learn a distance metric to achieve this purpose. We evaluated an example of each kind of algorithm as well as a hybrid approach that uses both techniques:

- COP-KMeans (CKM) performs hard constraint satisfaction [1].
- PC-KMeans (PKM) performs soft constraint satisfaction (permits some constraints to be violated) [4].
- M-KMeans (MKM) performs metric learning from constraints, but does not require that the constraints be satisfied [4].
- MPC-KMeans (MPKM) is a hybrid approach, performing both soft constraint satisfaction and metric learning [4].

Table 1 compares the results (averaged over 1000 trials) for each algorithm in terms of its unconstrained and constrained performance, when provided with 25 randomly selected constraints. We evaluated these algorithms on four UCI data sets [10]: Glass ($n = 214$), Ionosphere ($n = 351$), Iris ($n = 150$), and Wine ($n = 178$). Clustering performance was measured in terms of the Rand Index [11]. The Rand Indices of the unconstrained algorithms differ (e.g., 69.0% for CKM vs. 43.4% for PKM on the Glass data set) because of variations such as different cluster centroid initialization strategies and data pre-processing. In general, as expected and previously reported [1,3,4], average constrained clustering accuracy was equal to or greater than average unconstrained accuracy. The exception is MKM and MPKM's performance on the Wine data set, for which the constraints resulted in a *reduction* in average accuracy.

A careful examination of individual trials reveals that several constraint sets adversely affect clustering performance. Table 2 shows the fraction of these 1000 trials that suffered a drop in clustering accuracy when using constraints, compared to not using constraints. Note that each trial involved the same initialization of the centroids for both the unconstrained and constraint experiments so any change in performance is due to the constraints. We see that, for CKM on all data sets, at least 25% of the constraint sets resulted in a decrease in

Table 2. Fraction of 1000 randomly selected 25-constraint sets that caused a drop in accuracy, compared to an unconstrained run with the same centroid intialization

	Algorithm			
Data Set	**CKM**	**PKM**	**MKM**	**MPKM**
Glass	28%	1%	11%	0%
Ionosphere	26%	77%	0%	77%
Iris	29%	19%	36%	36%
Wine	38%	34%	87%	74%

performance. For the other algorithms, the fraction of negative results ranges up to 77% (for PKM and MPKM) and 87% (for MKM). The high proportion of negative results for MKM and MPKM on the Wine data set help explain why the average results show a decrease in performance (Table 1). These negative results occur frequently for all data sets and algorithms. In fact, only two of the 16 cases presented in Table 2 are completely free of the negative effect (MKM and MPKM with the Ionosphere and Glass data sets respectively). The average performance results tend to mask this effect, since positive gains are often of more magnitude than negative losses. However, for most real applications, we are more interested in performance for the (single) set of available constraints than "average" performance over many sets of constraints.

The possibility of a negative impact from constraints has significant implications for the practice of constrained clustering. First, the assumption that constraints are always helpful (or at least, do no harm) for clustering has been disproven by this empirical evidence. The adverse effects we observe are not restricted to a single data set or constrained clustering algorithm. This underscores the need for a means of characterizing relevant properties of a given constraint set, so that we can understand why it has a positive or negative effect on clustering. Such a characterization can also aid in future studies, so that useful constraints can be selected preferentially and constraints with adverse effects can be avoided. In the next section, we offer two constraint set measures that provide the first steps toward this goal.

3 Characterizing the Utility of Constraint Sets

A major contribution of this work is the introduction of two measures, informativeness and coherence, that quantify important constraint set properties.

- **Informativeness** refers to the amount of information in the constraint set that the algorithm cannot determine on its own. It is determined by the clustering algorithm's objective function (bias) and search preference. For example, in Figure 1(a), an algorithm such as CKM would be biased towards grouping nearby points together and separating distant points, but the specified constraints contradict this bias.
- **Coherence** measures the amount of agreement within the constraints themselves, with respect to a given distance metric. Figure 1(b) shows two

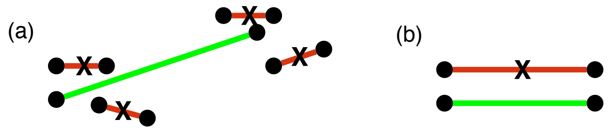

Fig. 1. Simple illustrative examples of (a) constraints with high informativeness for CKM and (b) highly incoherent constraints, given a Euclidean distance metric. Must-link constraints are depicted as solid line segments; cannot-link constraints have an 'X' through them.

constraints (ML and CL) that are very close and parallel. The ML constraint indicates that the distance between the points (and surrounding points) should be small, while the CL constraint implies the opposite. With respect to a Euclidean distance metric, these two constraints are incoherent.

The hypothesis that we investigate in this paper is that *constraint sets with high informativeness and coherence are most likely to provide performance gains.* We also expect that the negative performance effects are caused by highly incoherent constraint sets. First, incoherent constraints (as in Figure 1(b)) can cause metric learning methods (MKM, MPKM) to learn suboptimal global metrics. Also, since these algorithms use the constraints to initialize the cluster centroids, incoherent constraint sets are more likely to lead to a bad cluster initialization, increasing the chance of the clustering algorithm getting stuck in a poor local minimum.

3.1 Quantifying Informativeness

We begin this section with some straightforward but necessary definitions.

Definition 1. ***Partition Specification.*** *For any partition P of a data set D containing n items, a set of constraints C completely* **specifies** *P if it is a set of at most $\binom{n}{2}$ must-link and cannot-link constraints that uniquely defines P.*

Definition 2. ***Incomplete Constraint Set.*** *A set of constraints $\hat{C}$ is* ***incomplete*** *with respect to a data set D if it does not specify a unique partition P of D.*

In practice, most interesting problems will have an incomplete set of constraints, so that there exist multiple partitions that satisfy all constraints. We first introduce an idealized definition of constraint set informativeness.

Definition 3. ***Idealized Informativeness.*** *Let P^* be the partition that globally minimizes the objective function of some algorithm $\mathcal{A}$, in the absence of any constraints. Let C^* specify P^* in the sense given in Definition 1. The informativeness in a given constraint set C is the fraction of constraints in C that are* ***violated*** *by C^*.*

That is, $\{C^* - C\}$ is the set of constraint relationships that $\mathcal{A}$ is unable to correctly determine using its default bias. These constraints are therefore informative with respect to maximizing clustering accuracy. For illustration, consider a data set $\{a, b, c, d, e\}$ with $P^* = \{[a, b], [c, d, e]\}$. Using definition 1, we obtain C^*, which can be compactly represented as $\{ML(a, b), ML(c, d), ML(d, e), CL(a, c)\}$ due to the transitive and entailment properties of ML and CL constraints respectively [1]. If we are given a set of constraints $C_1 = \{ML(a, b), ML(c, d)\}$, then C_1 has an informativeness of 0; each of the constraints was already satisfied by the algorithm's default output P^*. In contrast, $C_2 = \{ML(a, b), ML(b, c)\}$ has an informativeness of 0.5 because $ML(b, c)$ is not in C^* and is therefore new information.

This definition of informativeness cannot be realized in practice, since we do not know P^* prior to clustering. We next present an efficiently computable approximation.

Approximate Measure of Informativeness. Our approximation is based on measuring the number of constraints that the clustering algorithm cannot predict using its default bias. Given a possibly incomplete set of constraints C and an algorithm $\mathcal{A}$, we generate the partition $P_\mathcal{A}$ by running $\mathcal{A}$ on the data set without any constraints. We then calculate the fraction of constraints in C that are unsatisfied by $P_\mathcal{A}$:

$$\mathcal{I}_\mathcal{A}(C) = \frac{1}{|C|}\left[\sum_{c \in C} unsat(c, P_\mathcal{A})\right] \tag{1}$$

where $unsat(c, P_\mathcal{A})$ is 1 if P does not satisfy c and 0 otherwise. This approach effectively uses the constraints as a hold-out set to test how accurately the algorithm predicts them. Given this equation, we can quantify the informativeness of the constraint sets in Figure 1 for the CKM algorithm as $\mathcal{I}_{CKM}(C_a) = 1.0$ and $\mathcal{I}_{CKM}(C_b) = 0.5$.

3.2 Quantifying Coherence

Coherence is the amount of agreement between the constraints themselves, given a metric $\mathcal{D}$ that specifies the distance between points. It does not require knowledge of the optimal partition P^* and can be computed directly. The coherence of a constraint set is independent of the algorithm used to perform constrained clustering.

One view of an $ML(x, y)$ (or $CL(x, y)$) constraint is that it imposes an attractive (or repulsive) force within the feature space along the direction of a line formed by (x, y), within the vicinity of x and y. Two constraints are incoherent if they exert contradictory forces in the same vicinity. We consider all constraint pairs composed of an ML and a CL constraint (pairs composed of the same constraint type cannot be contradictory). To determine the coherence of two constraints, a and b, we compute the *projected overlap* of each constraint on the other as follows (see Figure 2 for examples).

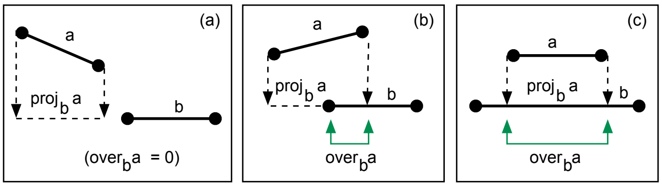

Fig. 2. Three cases of computing the projected overlap between constraints a and b

Let $\overrightarrow{a}$ and $\overrightarrow{b}$ be vectors connecting the points constrained by a and b respectively. Without loss of generality, we use the convention (x_1, x_2) to refer to the points connected by a vector $\overrightarrow{x}$. In the context of Figure 2, x_1 appears to the left of x_2 for all vectors shown. We first project $\overrightarrow{a}$ onto $\overrightarrow{b}$:

$$\overrightarrow{p} = proj_{\overrightarrow{b}}\overrightarrow{a} = \left(|\overrightarrow{a}|\cos\theta\right)\frac{\overrightarrow{b}}{|\overrightarrow{b}|},$$

where θ is the angle between the two vectors. Next, we calculate how much of this projection overlaps with $\overrightarrow{b}$. Since $\overrightarrow{p}$ and $\overrightarrow{b}$ are colinear ($\theta = 0$), we simply compute the distance from b_2 to each of b_1, p_1, and p_2. There are three cases, corresponding to the three examples in Figure 2:

$$over_b a = \begin{cases} 0 & \text{if } d_{b_2,b_1} \leq d_{b_2,p_2}, d_{b_2,b_1} \leq d_{b_2,p_1} \\ d_{b_1,p_2} & \text{if } d_{b_2,p_2} < d_{b_2,b_1}, d_{b_2,p_1} \geq d_{b_2,b_1} \\ d_{p_1,p_2} & \text{if } d_{b_2,p_2} < d_{b_2,b_1}, d_{b_2,p_1} < d_{b_2,b_1} \end{cases} \quad (2)$$

Given this background, we now define coherence, $\mathcal{COH}$, as the fraction of constraint pairs that have zero projected overlap:

$$\mathcal{COH}_{\mathcal{D}}(C) = \frac{\sum_{m \in C_{ML}, c \in C_{CL}} \delta(over_c m = 0 \text{ and } over_m c = 0)}{|C_{ML}||C_{CL}|} \quad (3)$$

We quantify the coherence of the constraint sets in Figure 1 as $\mathcal{COH}(C_a) = 0.0$ (all ML/CL pairs have some overlap) and $\mathcal{COH}(C_b) = 0.0$ (the single constraint pair completely overlaps).

Our measure of coherence is applicable to any space where vector projection is defined. The preceding examples and the experimental results presented later all make use of a Euclidean distance metric, since the four algorithms we evaluate use either Euclidean distance or a close variant, such as a generalized Mahalanobis (weighted Euclidean) distance metric.

4 Experimental Results

In this section, we present three important results. First, we analyze the relationship between the proposed measures (informativeness and coherence) and

Table 3. Average measures of informativeness ($\mathcal{I}$) and coherence ($\mathcal{COH}$) of 5000 randomly generated 3-constraint sets. Compare with Table 1.

	Algorithm				
Data Set	$\mathcal{I}_{CKM}$	$\mathcal{I}_{PKM}$	$\mathcal{I}_{MKM}$	$\mathcal{I}_{MPKM}$	$\mathcal{COH}$
Glass	0.34	0.43	0.50	0.50	0.45
Ionosphere	0.41	0.41	0.42	0.42	0.27
Iris	0.12	0.12	0.11	0.11	0.51
Wine	0.28	0.28	0.06	0.06	0.60

constrained clustering performance. Next, we show the benefits that can be obtained when using these measures to filter constraint sets. Finally, we analyze constraint sets from an image data set and show how informativeness and coherence can provide insights into why clustering performance increases or decreases.

4.1 Impact of Informativeness and Coherence on Clustering Performance

To understand how these constraint set properties affect various algorithms, we performed the following experiment. We randomly generated constraint sets of just three constraints 5000 times. With such a small number of constraints, the possible combinations of informativeness and coherence values is limited, permitting a detailed study. For each data set, we can compare the performance of each algorithm for each possible informativeness/coherence situation.

First, we report the average informativeness and coherence we observed for each algorithm and data set (Table 3). Although Tables 1 and 3 are not directly comparable due to the difference in constraint set sizes, we see an interesting trend. In Table 1, the Glass data set exhibited the largest increases in accuracy when using constraints; we find in Table 3 that the average informativeness for these constraints is also high. However, high informativeness is not sufficient for predicting accuracy improvement: the Ionosphere constraints, although informative, also tend to have very low coherence. Incoherent sets are difficult to completely satisfy, and we see this reflected in the lack of significant improvement when using constraints with this data set. Conversely, the Iris constraints have relatively high coherence but low informativeness, leading to the modest (but positive) average effect on performance for all algorithms. The Wine constraints have a remarkable lack of informativeness for MKM and MPKM, so the incoherence of the data set dominates performance and explains the small decrease in average accuracy.

We have shown that average results can obscure individual behavior. Therefore, we conducted a detailed analysis to better understand the relationships between each measure and performance. Table 4 focuses on constraint sets that are fully coherent, comparing performance between sets with high vs. low informativeness. We find that high informativeness almost always leads to an increase in performance, for all algorithms. The exception is MKM and MPKM on the Wine data set. Table 5 explores the opposite situation, focusing on constraint

Table 4. Average accuracy for fully coherent constraint sets, comparing performance of sets with high ("Inform.") and low ("Uninf.") informativeness

	Algorithm							
	CKM		**PKM**		**MKM**		**MPKM**	
Data Set	Inform.	Uninf.	Inform.	Uninf.	Inform.	Uninf.	Inform.	Uninf.
Glass	**68.9**	67.8	**57.8**	57.1	**58.1**	49.6	**54.9**	54.3
Ionosphere	58.9	58.9	58.8	58.7	58.9	58.9	**93.9**	93.5
Iris	**89.2**	88.1	**88.1**	86.7	**92.9**	89.2	**93.9**	93.5
Wine	71.8	71.8	**72.1**	71.8	92.2	**93.9**	93.5	**93.9**

Table 5. Average accuracy for non-informative constraint sets, comparing performance of coherent ("Coh.") and incoherent ("Incoh.") sets

	Algorithm							
	CKM		**PKM**		**MKM**		**MPKM**	
Data Set	Coh.	Incoh.	Coh.	Incoh.	Coh.	Incoh.	Coh.	Incoh.
Glass	**67.9**	67.4	**57.1**	54.3	**49.6**	49.4	**54.8**	50.2
Ionosphere	58.9	58.9	58.7	58.7	58.9	58.9	58.8	58.8
Iris	**86.7**	85.2	**86.7**	85.2	89.2	89.2	**89.3**	88.8
Wine	71.8	71.8	71.9	71.8	94.0	93.9	**93.5**	93.2

sets that have low informativeness but a variety of coherence values. Incoherence tends to adversely affect performance, particularly for the Glass and Iris data sets. It has less impact on the Ionosphere and Wine data sets.

4.2 Constraint Selection Based on Coherence

We posit that informativeness and coherence can provide guidance in selecting the most useful constraint sets. Returning to the 25-constraint experiments from Section 2, we applied a coarse constraint set selection strategy by removing the 500 least coherent constraint sets and calculating average performance on the remaining 500 sets (Table 6). We find a small but consistent increase in the average accuracy with those sets removed, suggesting that generating or selecting constraint sets with high coherence can provide gains in future constrained clustering experiments. The Iris data set, when analyzed by MPKM, is an exception to this rule. The MPKM results suggest that there are some less-coherent constraint sets that yield very good performance, when both metric learning and constraint satisfaction are used. We plan to investigate this exception more thoroughly in future work.

4.3 Visualizing Informative and Coherent Constraint Sets

We have demonstrated empirically that highly informative and coherent constraint sets lead to improved clustering performance, while incoherent sets can have an adverse effect. In this section, we show examples of constraint sets from an image data set that permit us to visualize informative and coherent constraint sets.

For these experiments, we used the CMU Face Images data set [9]. We used a subset containing 271 images of human faces with a variety of orientations and expressions. Each image is labeled as Male or Female, and the goal is to identify clusters that correspond with these categories (215 Male and 56 Female images). Each image is approximately 120×120 pixels, yielding a total of 14402 features. We conducted 100 trials, each time generating two randomly selected constraints. Without constraints, the algorithms achieved Rand Indices of: CKM (53.2%), PKM (53.9%), MKM (51.9%) and MPKM (51.9%). When using two randomly selected constraints, the performance ranges were: CKM [53.6%,54.5%], PKM [53.6%, 55.3%], MKM [49.8%,53.7%], MPKM [49.9%, 66.3%]. For this problem, just two constraints can significantly improve the performance of the MKM and MPKM algorithms, suggesting that the constraints are very useful for metric learning.

Since the items in this data set are images, we can directly visualize the constraint sets. Figure 3 shows two constraint sets (one per line) that improved the performance of MPKM from 51.9% to over 65%; both sets have an informativeness and coherence of 1.0. Figure 4 shows two constraint sets (one per line) that either provided no improvement (CKM) or adversely affected performance (PKM, MKM, and MPKM) with respect to the unconstrained performance; both sets have an informativeness and coherence of 0.0.

We see that the beneficial constraint sets have an intuitive interpretation: the must-linked images connect examples with different facial orientations, while the cannot-link constraints are between images with very similar orientations. Because "orientation" is not a feature provided to the algorithm, these constraints are very informative. They encourage the algorithm to create clusters that avoid grouping images simply based on where the bright "face" pixels are located. In contrast, in the constraint sets that have negative effects, the must-linked instances are of different faces in the similar orientation, and the cannot-link constrained instances have different orientation. This biases the constrained clustering algorithms towards clustering faces with the same orientation, which is

Table 6. Clustering performance (Rand Index) when using constraint sets selectively. We report average accuracy over all 1000 25-constraint sets (results copied from Table 1) compared to average accuracy over the 500 most coherent sets. Statistically significant increases at the 95% confidence interval are shown in bold.

	Algorithm							
	CKM		**PKM**		**MKM**		**MPKM**	
Data Set	All	Top 500	All	Top 500	All	Top 500	All	Top 500
Glass	69.4	**70.4**	68.8	**70.6**	56.6	56.6	67.8	**68.4**
Ionosphere	58.6	**59.3**	58.9	58.9	58.8	**59.3**	58.9	58.9
Iris	87.8	**88.3**	88.3	88.3	93.6	**94.5**	**91.8**	91.4
Wine	70.9	**71.5**	72.0	**72.5**	91.3	**93.3**	90.6	**91.1**

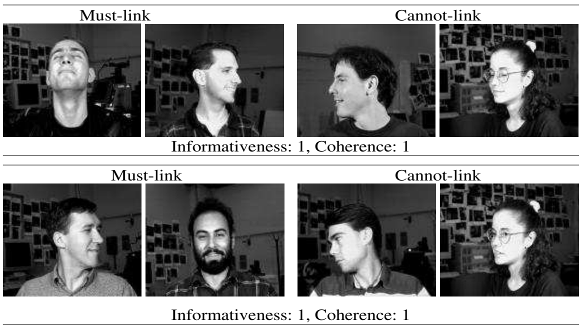
Must-link
Cannot-link
Informativeness: 1, Coherence: 1
Must-link
Cannot-link
Informativeness: 1, Coherence: 1

Fig. 3. Examples of beneficial constraint sets (one per line) that significantly improved the performance of MPKM

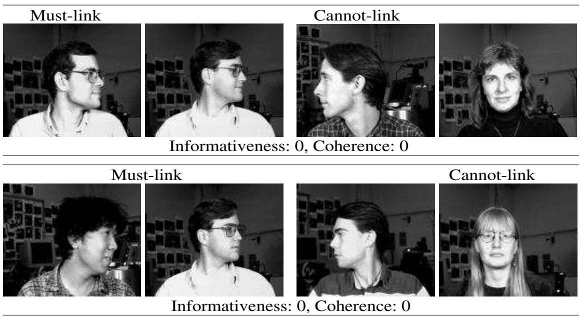
Must-link
Cannot-link
Informativeness: 0, Coherence: 0
Must-link
Cannot-link
Informativeness: 0, Coherence: 0

Fig. 4. Examples of constraint sets (one per line) that had no effect or an adverse effect on algorithm performance

not a useful strategy when trying to separate images by gender. Our measures of informativeness and coherence correctly capture this concept by characterizing the likely utility of each set.

5 Conclusions and Future Work

The contributions of this paper are two-fold. First, we have shown the first evidence that constraints can result in a decrease in clustering accuracy. This occurs even with constraints that are completely accurate and noise-free. In experiments with four UCI data sets and four constrained clustering algorithms, we found that the fraction of randomly generated constraint sets that result in a performance drop can range well above 50%. Second, we proposed two constraint set properties, informativeness and coherence, that provide a quantitative basis for explaining why a given constraint set increases or decreases performance. We demonstrated that performance gains are largely attributable to constraint sets with high informativeness and coherence, while drops in performance are associated with incoherent data sets.

Our experiments with selectively filtering randomly generated constraints to remove sets with low coherence suggest a promising avenue for future work with constrained clustering algorithms. We plan to more fully explore the use of informativeness and coherence to select the most useful constraints for clustering. Ultimately, this research direction could lead to reduced computational effort (since fewer constraint sets are needed to assess performance) and higher average performance on a variety of data sets.

Acknowledgments. We thank the anonymous reviewers for their thoughtful comments. We thank the National Science Foundation (ITR grant #0325329) and DARPA (CALO Project: Contract #NBCHD030010, Order #T310) for partially supporting this work financially. The research described in this paper was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

1. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained k-means clustering with background knowledge. In: Proceedings of the Eighteenth International Conference on Machine Learning. (2001)
2. Klein, D., Kamvar, S.D., Manning, C.D.: From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In: Proceedings of the Nineteenth International Conference on Machine Learning. (2002)
3. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning, with application to clustering with side-information. In: NIPS 15. (2003)
4. Basu, S., Bilenko, M., Mooney, R.J.: A probabilistic framework for semi-supervised clustering. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA (2004)
5. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning a Mahalanobis metric from equivalence constraints. Journal of Machine Learning Research **6** (2005)
6. Wagstaff, K.L.: Intelligent Clustering with Instance-Level Constraints. PhD thesis, Cornell University (2002)
7. Lu, Z., Leen, T.K.: Semi-supervised learning with penalized probabilistic clustering. In: Advances in Neural Information Processing Systems 17. (2005)
8. Davidson, I., Ravi, S.S.: Clustering with constraints: Feasibility issues and the k-means algorithm. In: Proceedings of the 2005 SIAM International Conference on Data Mining. (2005)
9. Mitchell, T.: Machine Learning. McGraw Hill, New York, NY (1997)
10. Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html (1998)
11. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association **66**(366) (1971)