When is Constrained Clustering Beneficial, and Why? *

Kiri L. Wagstaff
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive, Pasadena CA 91109
kiri.wagstaff@jpl.nasa.gov

Sugato Basu
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
basu@ai.sri.com

Ian Davidson
Department of Computer Science
State University of New York, Albany
1400 Washington Ave., Albany, NY 12222
davidson@cs.albany.edu

Abstract

Several researchers have illustrated that constraints can improve the results of a variety of clustering algorithms. However, there can be a large variation in this improvement, even for a fixed number of constraints for a given data set. We present the first attempt to provide insight into this phenomenon by characterizing two constraint set properties: inconsistency and incoherence. We show that these measures are strongly anti-correlated with clustering algorithm performance. Since they can be computed prior to clustering, these measures can aid in deciding which constraints to use in practice.

Introduction and Motivation

The last five years have seen extensive work on incorporating instance-level constraints into clustering methods (Wagstaff *et al.* 2001; Klein, Kamvar, & Manning 2002; Xing *et al.* 2003; Bilenko, Basu, & Mooney 2004; Bar-Hillel *et al.* 2005). 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 on several real-world applications, such as noun phrase coreference resolution and GPS-based map refinement (Wagstaff *et al.* 2001), person identification from surveillance camera clips (Bar-Hillel *et al.* 2005) and landscape detection from hyperspectral data (Lu & Leen 2005).

However, the common practice of presenting results using learning curves, which average performance over multiple constraint sets of the same size, obscures important details. For example, we took four UCI data sets (Blake & Merz 1998) and generated 100 randomly selected constraint sets, each containing 50 constraints. We then clustered them with COP-KMeans (Wagstaff *et al.* 2001), using the same initialization for each clustering run. We observed that, even when the number of constraints is fixed, the accuracy of the output partition measured using the Rand Index (Rand 1971) varies greatly, by 4 to 11% (Table 1). Since the starting point for each run was held fixed, the only source of variation is the constraint set. We have identified two constraint set properties that help explain these variations: *inconsistency* and *incoherence*.

	Accuracy		
Data set	Min	Mean	Max
Glass	67.6	69.9	72.3
Iris	82.2	88.4	93.4
Ionosphere	58.2	60.1	62.3
Wine	68.0	71.3	74.3

Table 1: Variation in accuracy (Rand Index) obtained by COP-KMeans using 50 randomly generated constraints and a fixed starting point, over 100 trials.

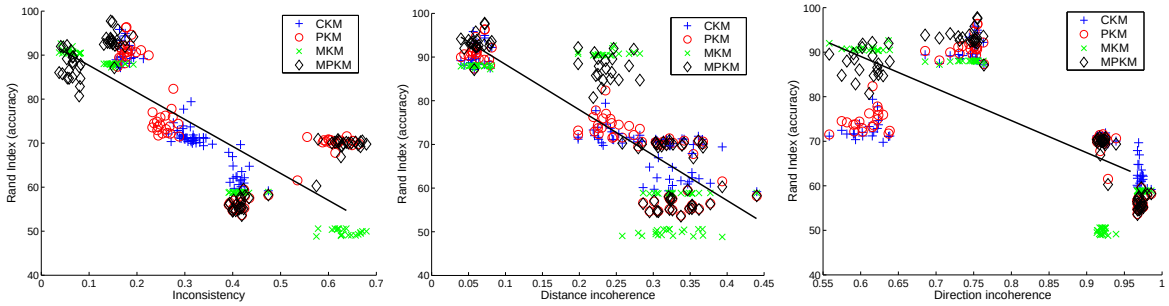
Quantifying Inconsistency/Incoherence

Inconsistency is the amount of conflict between the constraints and the underlying objective function and search bias of an algorithm. Our measure of inconsistency quantifies the degree to which the algorithm cannot "figure out" the constraints on its own. Given an algorithm $\mathcal{A}$, we generate partition $P_{\mathcal{A}}$ by running $\mathcal{A}$ on the data set without any constraints. We then calculate the fraction of constraints in constraint set $\mathcal{C}$ that are unsatisfied by $P_{\mathcal{A}}$, and average this measure over multiple unconstrained clustering runs.

Incoherence is the amount of internal conflict between the constraints in set $\mathcal{C}$, given a distance metric $\mathcal{D}$. It is not algorithm dependent. Constraints can be incoherent with respect to *distance incoherence* and/or *directional incoherence*. Intuitively, the points involved in an ML constraint should be close together, while points involved in a CL constraint should be far apart, as viewed by the ideal distance metric. In fact, this line of reasoning is what motivates some of the metric-learning constrained clustering algorithms (Klein, Kamvar, & Manning 2002; Bilenko, Basu, & Mooney 2004). We define distance incoherence as the fraction of constraint pairs (each pair consisting of a ML and a CL constraint) for which the separation of the points in the ML constraint is more than that of the CL constraint.

Distance aside, the directional position of a constraint in the feature space is also important. An ML (or CL) constraint can be viewed as imposing an attractive (or repulsive) force in the feature space within its vicinity. An ML/CL constraint pair is directionally incoherent if they exert contradictory forces in the same vicinity. To determine if two constraints, c_1 and c_2, are incoherent, we treat the constraints as line segments and compute their *projected overlap*, or how much the projection of c_1 along the direction of c_2 overlaps with (interferes with) c_2. We define directional incoherence as the fraction of constraint pairs that have a non-zero projected overlap.

*Abstract type: original, unpublished research. This work was supported by NSF ITR #0325329.



(a) Inconsistency ($r = -0.80$) (b) Distance Incoherence ($r = -0.82$) (c) Direction Incoherence ($r = -0.74$)

Figure 1: Mean accuracy of algorithms, as a function of problem inconsistency and incoherence, over four UCI data sets. The line is a linear fit to the data that shows strong negative correlations (r values, 99% confidence).

Experimental Results

In these experiments, we examined what happens when inconsistent or incoherent constraint sets are provided to different constrained clustering methods. Because our difficulty measures can be computed *a priori*, they can aid in deciding which constraints to use.

The two most common approaches to constrained clustering involve either satisfying the constraints directly or learning a distance metric that accommodates the constraints. We compared a representative of each approach and a hybrid method that performs both functions: (1) COP-KMeans (CKM): hard constraint satisfaction in KMeans (Wagstaff *et al.* 2001); (2) PC-KMeans (PKM): soft constraint satisfaction (Bilenko, Basu, & Mooney 2004); (3) M-KMeans (MKM): metric learning from constraints (Bilenko, Basu, & Mooney 2004); and (4) MPC-KMeans (MPKM): hybrid approach, performing both soft constraint satisfaction and metric learning (Bilenko, Basu, & Mooney 2004).

For each data set in Table 1, we generated a constraint set, $\mathcal{C}$, by randomly selecting item pairs and examining their true labels. We computed the inconsistency and incoherence of $\mathcal{C}$, then clustered the data set using each algorithm 10 times from different random initializations and reported the average Rand Index obtained. We repeated this process 10 times for each number of constraints, $|\mathcal{C}| = \{10, 20, \ldots, 200\}$.

Figure 1 collects results for all four data sets, permitting us to make general conclusions for a variety of inconsistency and incoherence values. We observe a strong negative correlation between all three measures and accuracy. That is, constraint sets that are more consistent with the algorithm's bias, or more internally coherent, tend to yield the largest gains in accuracy. This helps explain why constraint sets of the same size can provide such different performance results, in contrast to traditional learning curves, which average the results for a given constraint set size.

We also assessed each algorithm's individual sensitivity to each measure. We found that CKM and MKM are the most sensitive to inconsistency ($r = -0.95, -0.98$), while PKM and MPKM are more robust ($r = -0.60, -0.73$). This analysis provides insights into the expected future performance on a problem with specific inconsistency and incoherence. Learning curves only characterize behavior on an isolated problem.

Conclusions and Future Work

Identifying meaningful constraint set properties is of benefit to practitioners and researchers. For scenarios in which the user can generate multiple constraint sets, these results strongly recommend selecting the one with the lowest inconsistency and incoherence. When multiple algorithms are available, choosing the algorithm that has the lowest inconsistency should help produce the best quality clustering with minimal computational effort. Our measures can also be used to prune noisy constraints or to actively choose constraints. We intend to explore these options in the future. We also plan to generalize our definition of incoherence to non-metric distance measures. Further, these measures can provide insight into the black-box computation of different metric-learning algorithms. Since metric-learning methods modify the distance metric, they effectively reduce incoherence as they iterate, and we can now quantify how much improvement is achieved at each iteration.

References

Bar-Hillel, A.; Hertz, T.; Shental, N.; and Weinshall, D. 2005. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research* 6:937–965.

Bilenko, M.; Basu, S.; and Mooney, R. J. 2004. Integrating constraints and metric learning in semi-supervised clustering. In *21st Int'l Conf. on Machine Learning*, 11–18.

Blake, C. L., and Merz, C. J. 1998. UCI Repository of Machine Learning Databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

Klein, D.; Kamvar, S. D.; and Manning, C. D. 2002. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *19th Int'l Conf. on Machine Learning*, 307–313.

Lu, Z., and Leen, T. K. 2005. Semi-supervised learning with penalized probabilistic clustering. In *NIPS 17*.

Rand, W. M. 1971. Objective criteria for the evaluation of clustering methods. *JASA* 66(366):846–850.

Wagstaff, K.; Cardie, C.; Rogers, S.; and Schroedl, S. 2001. Constrained k-means clustering with background knowledge. In *18th Int'l Conf. on Machine Learning*, 577–584.

Xing, E. P.; Ng, A. Y.; Jordan, M. I.; and Russell, S. 2003. Distance metric learning, with application to clustering with side-information. In *NIPS 15*, 505–512.