Clustering with Constraints

Ian Davidson
Department of Computer Science
University of California
Davis CA
davidson@cs.ucdavis.edu

Semi-supervised clustering

The area of clustering with constraints makes use of hints or advice in the form of constraints to aid or bias the clustering process. The most prevalent form of advice are conjunctions of pair-wise instance level constraints of the form must-link (ML) and cannot-link (CL) which state that pairs of instances should be in the same or different clusters respectively. Given a set of points P to cluster and a set of constraints C, the aim of clustering with constraints is to use the constraints to improve the clustering results. Constraints have so far being used in two main ways: a) Writing algorithms that use a standard distance metric but attempt to satisfy all or as many constraints as possible and b) Using the constraints to learn a distance function that is then used in the clustering algorithm.

The idea of using constraints to guide clustering was first introduced by Wagstaff and Cardie in their seminal paper ICML 2000 [13] with a modified COBWEB-style algorithm that attempts to satisfy all constraints. Later [14] they introduced constraints to the k-means algorithms. Their algorithms (as most algorithms now do) look at satisfying a conjunction of must-link and cannot-link constraints. Independently, Cohn, Caruana and McCallum [4, 3] introduced constraints as a user feedback mechanism to guide the clustering algorithm to a more useful result.

In 2002 Xing and collaborators [15] (NIPS 2002) and Klein and collaborators (ICML 2002) [12] explored making use of constraints by learning a distance function for non-hierarchical clustering and a distance matrix for hierarchical clustering respectively.

Basu and collaborators more recently have looked at key issues such as which are the most informative sets of constraints [2] and seeding algorithms using constraints [1]. Gondek has explored using constraints to find orthogonal/alternative clusterings of data [11, 3]. Davidson and Ravi explored the intractability issues of clustering under constraints for non-hierarchical clustering [5], hierarchical clustering [6] and non-hierarchical clustering with feedback [9].

Clustering has many successful applications in a variety of domains where the objective function of the clustering algorithm finds a novel and useful clustering. However, in some application domains the typical objective functions may lead to well-known or non-actionable clusterings of the data. This could be overcome by an ad-hoc approach such as manipulating the data. The introduction of constraints into clustering allows a principled approach to incorporate user preferences or domain expertise into the clustering process so as to guide the algorithm to a desirable solution or away from an undesirable solution. The typical semi-supervised learning situations involves having a label associated with a subset of the available instances. However in many domains, knowledge of the relevant categories is incomplete and it is easier to obtain pairwise constraints either automatically or from domain experts.

Types of Constraints. Must-link and cannot-link constraints are typically used since they can be easily generated from small amounts of labeled data (generate a must-link between two instances if the labels agree, cannot-link if they disagree) or from domain experts. They can be used to represent geometric properties [14, 5] by noting that for instance, making the maximum cluster diameter be α is equivalent to enforcing a conjunction of cannot-link constraints between all points whose distance is greater than α. Similarly, clusters can be separated by distance at at least δ by enforcing a conjunction of must-link constraints between all points whose distance is less than δ. Both types of instance-level constraints have interesting properties that can be used to effectively generate many additional constraints. Must-link constraints are transitive: $ML(x, y), ML(y, z) \rightarrow ML(x, z)$ and cannot link constraints have an entailment property: $ML(a, b), ML(x, y), CL(a, x) \rightarrow CL(a, y), CL(b, x), CL(b, y)$.

How Constraints Are Used. Constraints have typically been used in clustering algorithms in two ways. Constraints can be used to modify the cluster assignment stage of the cluster algorithm [14, 4], to enforce satisfaction of the constraints or as many as possible [2, 5]. These approaches typically use a standard distance or likelihood function. Alternatively, the distance function of the clustering algorithm can also be trained either before or after the clustering actually occurs using the constraints [12] [15]. The former are called constraint-based approaches and the later distance based approaches.

Constraint-based methods. In constraint-based approaches, the clustering algorithm itself (typically the assignment step) is modified so that the available constraints are used to bias the search for an appropriate clustering of the data. Figure 2 shows how though two clusterings exist (a horizontal and vertical clustering) just three constraints can rule out the former.

Constraint-based clustering is typically achieved using one of the following approaches:

-
- Enforcing constraints to be satisfied during the cluster assignment in the

clustering algorithm [13][6].

- Modifying the clustering objective function so that it includes a term for satisfying specified constraints. Penalties for violating constraints have been explored in the maximum likelihood framework [2] and distance framework [5].
- Initializing clusters and inferring clustering constraints based on neighborhoods derived from labeled examples [1].

Each of the above approaches provides a simple method of modifying existing partitional and agglomerative style hierarchical algorithms to incorporate constraints. For more recent advances in algorithm design such as the use of variational techniques for constrained clustering see [3].

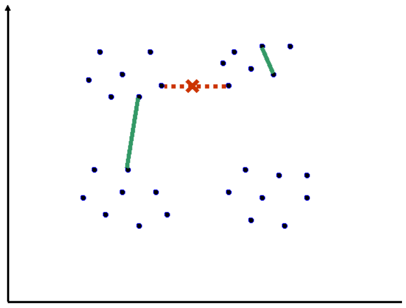

Figure 1: Input instances and constraints

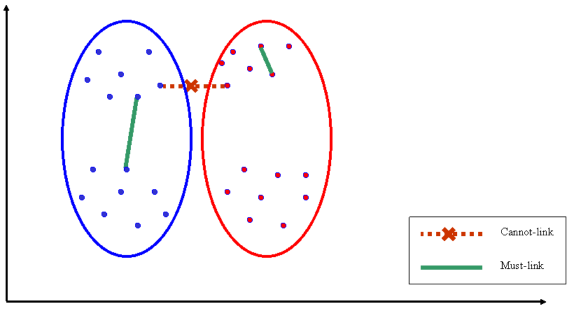

Figure 2: A Clustering That Satisfies All Constraints

Distance-based methods In distance-based approaches, an existing clustering algorithm that uses a distance measure is employed. However, rather than use the Euclidean distance metric, the distance measure is first trained to "satisfy" the given constraints. The approach of Xing and collaborators [15] casts the problem of learning a distance *metric* from the constraints so that the points (and surrounding points) that are part of the must-link (cannot-link) constraints are close together (far apart). They consider two formulations: firstly learning a generalized Mahanabolis distance metric which essentially stretches or compresses each axis as appropriate. Figure 4 gives an example where the constraints can be satisfied by stretching the x-axis and compressing the y-axis and then applying a clustering algorithm to the new data space. The second formulation allows a more complex transformation on the space of points.

Klein and collaborators [12] explore learning a distance *matrix* from constraints for agglomerative clustering. Only points that are directly involved in the constraints are brought closer together or far apart using a multi-step approach of making must-linked points have a distance of 0 and cannot-linked points having the greatest distance.

There have been some algorithms that try to both enforce constraints and learn distance functions from constraints [2].

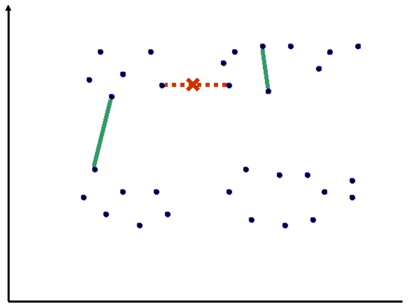

Figure 3: Input instances and Constraints

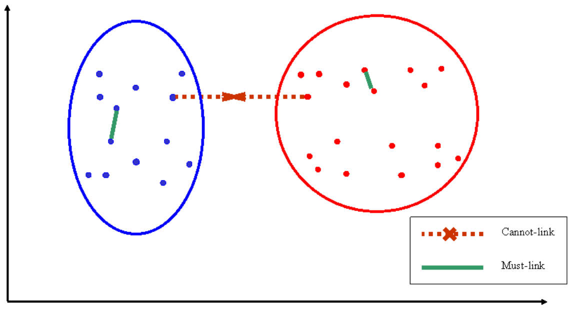

Figure 4: A Learnt distance space respective of the constraints.

Key application areas include images, video , biology, text, web pages, audio (speaker identification) [3] and GPS trace information [14].

www.constrained-clustering.org

Clustering, semi-supervised learning

References

[1] S. Basu, A. Banerjee, R. Mooney, Semi-supervised Clustering by Seeding, 19^{th} *International Conference on Machine Learning*, 2002.

[2] S. Basu, M. Bilenko and R. J. Mooney, Active Semi-Supervision for Pairwise Constrained Clustering, 4^{th} *SIAM Data Mining Conference*. 2004.

[3] S. Basu, I. Davidson and K. Wagstaff (editors), *Constrained Clustering: Advances in Algorithms, Theory and Applications*, Chapman Hall, CRC Press, 2008.

[4] D. Cohn, R. Caruana, R. and A. McCallum, Semi-supervised clustering with user feedback, Cornell University, 2003, Technical Report 2003-1892.

[5] I. Davidson and S. S. Ravi, Clustering with Constraints: Feasibility Issues and the k-Means Algorithm, 5^{th} *SIAM International Conference on Data Mining*, 2005.

[6] I. Davidson and S. S. Ravi, Hierarchical Clustering With Constraints: Theoretical and Empirical Results, 9^th *European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2005.

[7] I. Davidson and S. S. Ravi, The Complexity of Non-Hierarchical Clustering with Instance and Cluster Level Constraints, *Data Mining and Knowledge Discovery*, Vol. 14, No. 1, February 2007, pp. 25–61.

[8] I. Davidson and S. S. Ravi, Identifying and Generating Easy Sets of Constraints For Clustering, 21^{st} *AAAI Conference*, 2006.

[9] I. Davidson and S. S. Ravi, Intractability and Clustering with Constraints, 24^th *International Conference on Machine Learning* (ICML 2007), 2007.

[10] I. Davidson, M. Ester and S. S. Ravi, Efficient Incremental Clustering with Constraints, *Conference of Knowledge Discovery and Data Mining*, 2007.

[11] D. Gondek, T. Hofmann, Non-Redundant Data Clustering, 4^th *IEEE International Conference on Data Mining*, 2004.

[12] D. Klein, S. D. Kamvar and C. D. Manning, From Instance-Level Constraints to Space-Level Constraints: Making the Most of Prior Knowledge in Data Clustering, 19^th *International Conference on Machine Learning* 2002.

[13] K. Wagstaff and C. Cardie, Clustering with Instance-Level Constraints, 17^{th} *International Conference on Machine Learning* 2000.

[14] K. Wagstaff, C. Cardie, S. Rogers and S. Schroedl, Constrained K-means Clustering with Background Knowledge, 18^{th} *International Conference on Machine Learning*, 2001.

[15] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. *Advances in Neural Information Processing Systems 15* 2002.