Clustering with Soft and Group Constraints*

Martin H.C. Law, Alexander Topchy, and Anil K. Jain

Department of Computer Science and Engineering
Michigan State University
East Lansing, MI, 48824, USA
{lawhiu,topchyal,jain}@cse.msu.edu

Abstract. Several clustering algorithms equipped with pairwise hard constraints between data points are known to improve the accuracy of clustering solutions. We develop a new clustering algorithm that extends mixture clustering in the presence of (i) soft constraints, and (ii) group-level constraints. Soft constraints can reflect the uncertainty associated with a priori knowledge about pairs of points that should or should not belong to the same cluster, while group-level constraints can capture larger building blocks of the target partition when afforded by the side information. Assuming that the data points are generated by a mixture of Gaussians, we derive the EM algorithm to estimate the parameters of different clusters. Empirical study demonstrates that the use of soft constraints results in superior data partitions normally unattainable without constraints. Further, the solutions are more robust when the hard constraints may be incorrect.

1 Introduction

Modern cluster analysis [1] is largely driven by the quest for scalable and more robust clustering algorithms capable of detecting clusters with diverse shapes and densities. Data clustering is an ill-posed problem when the associated objective function is not well defined, leading to fundamental limitations of generic clustering algorithms. Multiple clustering solutions may seem to be equally plausible due to an inherent arbitrariness in the notion of a cluster. Any side (auxiliary) information must be used in order to reduce this degeneracy of possible solutions and improve the quality of clustering.

Unlike supervised classification, only recently some attention has been given to the role of prior information in data clustering. Prior information can be available in several forms: labelled data, known data groupings or associations, additional inter-pattern similarity estimates, feature relevance, object ranks, etc. We are primarily interested in various inter-point constraints that can complement already known pattern or proximity matrix. For example, pairwise constraints on the data points tell us which pairs of points must be placed in the same cluster (positive constraint) or different clusters (negative constraint). Ideally,

* This work was supported by the U.S. ONR grant no. N000140410183.

A. Fred et al. (Eds.): SSPR&SPR 2004, LNCS 3138, pp. 662–670, 2004.

the target partition must satisfy all the given data constraints. Hence, a clustering algorithm should be driven by attribute (feature) values as well as the constraint information, such that the clustering solution is biased in favor of the constraints.

Constraints are naturally available in many clustering applications. For instance, in image segmentation one can have partial grouping cues for several regions to assist in the overall clustering [2]. Clustering of customers in market-basket database can have multiple records pertaining to the same person. In video retrieval tasks different users may provide alternative annotations of images in small subsets of a large database [3]. Such groupings may be used for semi-supervised clustering of the entire database.

The prior knowledge was provided at the instance level in the form of positive (must-link) and negative (cannot-link) pairwise constraints in [4, 5]. A constrained k-means algorithm is proposed in [4]: must-link data points are replaced by their centroid, and a data point is assigned to the closest cluster center that does not violate any constraints. "Soft" constraints were introduced in the dissertation of Wagstaff [6], where a heuristic is employed to assign a point to a cluster that gets the lowest penalty for constraint violations. Similarly, the constrained version of COBWEB algorithm is considered in [5]. Constrained modification of the complete-link algorithm was proposed in [7]. Spectral clustering is modified in [8] to work with constraints. Again, a heuristic procedure augments the affinity matrix derived from feature space by the constraints. The EM algorithm for mixture model clustering with hard data constraints was developed in [9] and was shown to be superior to the constrained k-means algorithm [4]. Constraints were also incorporated into image segmentation algorithms using graph-based clustering in [10, 2]. Recently, Xing et al. [11] proposed a way to perform clustering by metric learning using side-information: metric can be learned from the constraints and then applied globally in the feature space to obtain the final clustering. Correlation clustering [12] uses only the positive and negative constraints to partition the vertices in a graph. It has been extended to cope with soft constraints [13, 14].

The main contribution of this paper is to adopt soft constraints in mixture (model-based) clustering. Each constraint becomes a real valued variable in between 0 and 1. The value of the constraint reflects the certainty of the prior knowledge that a pair of objects comes from the same cluster. Variable strength of the constraint allows for better control of clustering bias introduced by the constraints. Our main clustering algorithm is based on a generative model, where constraint variables are explicitly identified with the nodes in the corresponding Bayesian network. In this sense, we extend the work by Shental et al. [9] whose method included only the hard constraints into the mixture model clustering. To account for soft constraints, we use a more sophisticated graphical model, yet preserve linear complexity of the inference process (clustering) in the model. Coupled with mixture clustering, soft constraints are not strictly enforced but rather serve as prior values that can be changed by the observed data. Mutually conflicting "soft"constraints are allowed. Moreover, our model can operate with

group constraints, namely we can specify the certainty of each of several points belonging to the same cluster (not in a pairwise manner).

2 Clustering with Constraints

Consider a data set $\mathcal{D} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ of size N. Let $z_i \in \{1, \ldots, K\}$ be the (hidden) cluster label of point $\mathbf{y}_i$, where K is the number of clusters. Suppose we want to incorporate the constraint that the first three points $\mathbf{y}_1$, $\mathbf{y}_2$ and $\mathbf{y}_3$ are in the same group and should belong to the same cluster. This can be done by setting z_1, z_2 and z_3 to a common value w_1, $z_1 = w_1$, $z_2 = w_1$ and $z_3 = w_1$. Here, w_1 is an auxiliary random variable that serves as a "group-label"[1] – cluster label of the group $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\}$. The likelihood function for the observed data $\mathcal{D}$ can be derived based on the group membership assumptions, and the EM algorithm is used for parameter estimation [9].

Note that the equalities $z_1 = w_1$ and $z_2 = w_1$ mean that this pairwise constraint is "hard", namely, the points $\mathbf{y}_1$ and $\mathbf{y}_2$ are certain to have the same cluster label. In general, this may not be true. Alternatively, we require $z_i = w_l$ to be true only with a probability γ_{il}. The value of $\gamma_{il} \in [0, 1]$ can be interpreted as the strength of the constraint that $\mathbf{y}_i$ belongs to the l-th group. To have a logically consistent framework, any $\mathbf{y}_i$ without any associated constraint information should be equivalent to $\gamma_{il} = 0$, $\forall l$. This ensures uniform treatment for data points with and without constraints. If $\gamma_{il} = 0$, z_i is chosen independently of the other group and cluster labels.

Formally, let α_j be the prior probability for the j-th mixture component $q_j(\mathbf{y}; \theta_j)$, which is parameterized by θ_j. For simplicity, we write $q_j(\mathbf{y}) = q_j(\mathbf{y}; \theta_j)$. Let w_l $(l = 1, \ldots, L)$ be the set of (hidden) group-labels. Each group label can take a value from 1 to K. Let v_i be a discrete random variable that takes value in $\{0, \ldots, L\}$ and determines how z_i is generated. If $v_i = l$, the point $\mathbf{y}_i$ participates in the l-th group and thus $z_i = w_l$ and $P(v_i = l) = \gamma_{il}$. When $v_i = 0$, label z_i is generated independently according to the prior probabilities $\{\alpha_j\}$. Hence, the model for $\mathbf{y}_i$ is specified as follows:

$$P(w_l = j) = \alpha_j, \quad l \in \{1, \ldots, L\}, j \in \{1, \ldots, K\} \tag{1}$$

$$P(v_i = l) = \gamma_{il}, \quad i \in \{1, \ldots, N\}, l \in \{1, \ldots, L\} \tag{2}$$

$$P(z_i = j | v_i, w_1, \ldots, w_L) = \begin{cases} \alpha_j & \text{if } v_i = 0 \\ \delta_{w_l, j} & \text{if } v_i = l \end{cases} \tag{3}$$

$$p(\mathbf{y} | z = j) = q_j(\mathbf{y}), \tag{4}$$

where δ_{ij} is the Kronecker delta. An example of such model with seven data points and three group labels is shown in Figure 1. Typically, when a user specifies γ_{il}, many of them are set to zero. It means that the label z_i is only tied to a small number of group-labels. The case when z_i has more than one group-label corresponds to the existence of possibly incompatible constraint, because z_i can

[1] It corresponds to the term "chunklet" defined in [9].

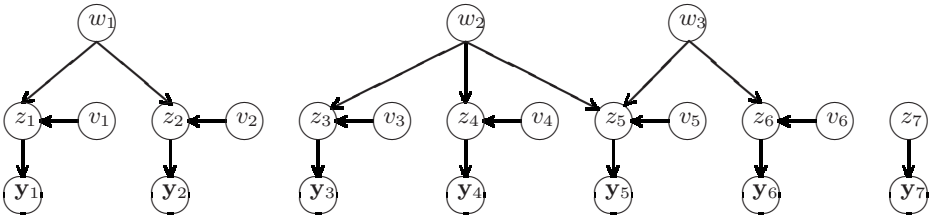


Fig. 1. An example of the graphical model for the proposed soft constraint. There are seven data points $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_7\}$ with three group-labels $\{w_1, w_2, w_3\}$. There are competing constraints for z_5. Note that each connected component in the graph is a polytree and hence the belief propagation algorithm can be used. The number of clusters, K, determine the possible values that w_l and z_i can assume.

belong to more than one group. This probabilistic model can be given generative interpretation (Figure 1). First, the L group-labels $\{w_l\}$ are generated according to the component prior probabilities $\{\alpha_j\}$. For each $i \in \{1, \ldots, N\}$, we generate v_i with the probabilities $\{\gamma_{il}\}$. The outcome determines how z_i gets its value: if v_i is between 1 and L, z_i is set to w_l; otherwise, z_i is generated independently according to $\{\alpha_j\}$. Based on the value of z_i, the point $\mathbf{y}_i$ is generated from $q_{z_i}(\mathbf{y})$.

2.1 Constraints Specification

One important advantage of adopting soft constraints is its robustness. It is usually difficult to obtain definitive statements on the properties of patterns in real world applications. Thus a practical clustering algorithm using constraint information should tolerate noisy constraints. However, a single erroneous "cannot-link" constraint can break down the constrained k-means algorithm in [4]. The dissimilarity values between multiple items can be drastically altered by a single bad constraint. In our proposed approach, $z_i = w_l$ is required to be true only with a certain probability. This flexibility protects us from disastrous clustering solution when some constraints may be wrong. Although the algorithm in [9] also does not break down in view of erroneous constraints, later in section 3 we shall demonstrate that the use of soft constraints in the proposed algorithm can lead to superior results when the constraints are noisy.

Different constraints are specified by assigning different values to γ_{il}, which in turn specifies the topology of the graphical model by the sparsity of the matrix $\{\gamma_{il}\}$. Note that we have made the abstraction that the group-label may not correspond to the label of any particular data point. However, it is easy to enforce the group-label w_l to be the same as the cluster label z_i by setting $\gamma_{il} = 1$. Equivalence constraint information between $\mathbf{y}_i$ and $\mathbf{y}_j$ can then be incorporated by setting γ_{jl} to be the confidence that they are in the same cluster. The abstraction of group-label is useful in the distributed learning scenario described in [9]. Different teachers are asked to assign group labels to different subsets of the data. Further, suppose the teachers also provide confidence values in their assignment. Let w_l correspond to a group labelled by a certain teacher. The confidence that $\mathbf{y}_i$ belongs to the l-th group is represented by γ_{il}.

2.2 Parameter Estimation

The model parameters $\{\alpha_j\}$ and $\{\theta_j\}$ can be estimated by maximizing the data log-likelihood function. Note that γ_{il} values are provided by the user and do not need to be estimated. Since $\{z_i\}$, $\{v_i\}$ and $\{w_l\}$ are hidden variables, this is a missing data problem and the EM algorithm can be used. We refer the readers to texts like [15] for more details on the EM algorithm. The complete data log-likelihood can be written as

$$\begin{aligned}\mathcal{L} &= \log p(\{\mathbf{y}_i\}, \{z_i\}, \{w_l\}, \{v_i\}) \\ &= \begin{cases} -\infty & \exists v_i \neq 0 : z_i \neq w_{v_i} \\ \sum_{i=1}^{N} \left(\log q_{z_i}(\mathbf{y}_i) + \log \gamma_{v_i,i} + \delta_{v_i,0} \log \alpha_{z_i}\right) + \sum_{l=1}^{L} \log \alpha_{w_l} & \text{otherwise} \end{cases}\end{aligned} \tag{5}$$

The data is said to be inconsistent (have zero probability) if there exists $v_i \neq 0$ such that $z_i \neq w_{v_i}$. Let $\boldsymbol{\theta}$ denote the current parameter estimate. Taking expectation of $\mathcal{L}$ with respect to the missing data, given $\boldsymbol{\theta}$ and $\mathcal{D}$, we obtain

$$\begin{aligned}&E[\log p(\{\mathbf{y}_i\}, \{z_i\}, \{w_l\}, \{v_i\})] \\ =&\sum_{i=1}^{N}\sum_{j=1}^{K} P(z_i = j|\{\mathbf{y}_i\}) \log q_j(\mathbf{y}_i) + \sum_{l=1}^{L}\sum_{j=1}^{K} P(w_l = j|\{\mathbf{y}_i\}) \log \alpha_j + \\ &\sum_{i=i}^{N}\sum_{l=0}^{L} P(v_i = l|\{\mathbf{y}_i\}) \log \gamma_{li} + \sum_{i=1}^{N}\sum_{j=1}^{K} P(v_i = 0, z_i = j|\{\mathbf{y}_i\}) \log \alpha_j\end{aligned} \tag{6}$$

Note that different $\mathbf{y}_i$'s may not be independent because they can be related indirectly by a common w_l. Also, the inconsistency of hidden data does not depend on the parameter values. The expected value of $\mathcal{L}$ is computed over only the set of consistent values of hidden variables and hence no infinite values are encountered. The expected complete data log-likelihood can be maximized with respect to the parameters $\{\alpha_j, \theta_j\}$ by

$$\hat{\alpha_j} = \frac{\sum_{l=1}^{L} P(w_l = j|\{\mathbf{y}_i\}) + \sum_{i=1}^{N} P(v_i = 0, z_i = j|\{\mathbf{y}_i\})}{\sum_{j=1}^{K}\left(\sum_{l=1}^{L} P(w_l = j|\{\mathbf{y}_i\}) + \sum_{i=1}^{N} P(v_i = 0, z_i = j|\{\mathbf{y}_i\})\right)} \tag{7}$$

$$\hat{\boldsymbol{\mu}}_j = \frac{\sum_{i=1}^{N} P(z_i = j|\{\mathbf{y}_i\})\mathbf{y}_i}{\sum_{i=1}^{N} P(z_i = j|\{\mathbf{y}_i\})} \tag{8}$$

$$\hat{\mathbf{C}}_j = \frac{\sum_{i=1}^{N} P(z_i = j|\{\mathbf{y}_i\})(\mathbf{y}_i - \hat{\boldsymbol{\mu}}_j)(\mathbf{y}_i - \hat{\boldsymbol{\mu}}_j)^T}{\sum_{i=1}^{N} P(z_i = j|\{\mathbf{y}_i\})}, \tag{9}$$

where the j-th component is assumed to be a Gaussian with mean $\boldsymbol{\mu}_j$ and covariance $\mathbf{C}_j$. The parameter update in Equations (7) to (9) corresponds to the M-step of the EM algorithm. The E-step consists of the computation of the probabilities $P(w_l{=}j|\{\mathbf{y}_i\})$, $P(z_i{=}j|\{\mathbf{y}_i\})$ and $P(v_i{=}0, z_i{=}j|\{\mathbf{y}_i\})$. Unlike the standard Gaussian mixture, it is not easy to express these probabilities by simple

equations because of the interdependence of $\{\mathbf{y}_i\}$ via w_l. Instead, these probabilities can be computed by standard Bayesian network inference algorithms like belief propagation or junction tree. The two-variable query $P(v_i{=}0, z_i{=}j|\{\mathbf{y}_i\})$ can be easily handled since the node v_i is a parent of z_i. Because of the simplicity of the structure of the graphical model, inference can be carried out efficiently. In particular, the complexity is virtually the same as the standard EM algorithm when there are no competing constraints for all the data points. This is the most usual scenario in constraint clustering.

3 Experiments

3.1 Synthetic Data

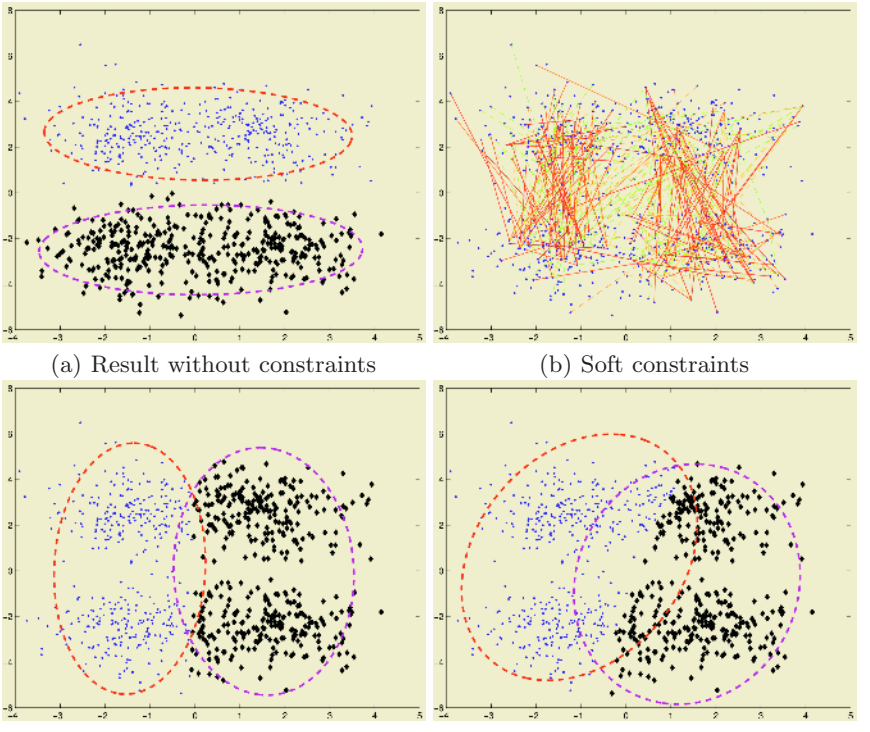
In the first experiment, we investigate how constraint information can be used to bias the search for the appropriate clusters. Four 2D Gaussian distributions with mean vectors $\left[\begin{smallmatrix}1.5\\2.5\end{smallmatrix}\right]$, $\left[\begin{smallmatrix}-1.5\\2.5\end{smallmatrix}\right]$, $\left[\begin{smallmatrix}-1.5\\-2.5\end{smallmatrix}\right]$, $\left[\begin{smallmatrix}1.5\\-2.5\end{smallmatrix}\right]$, and identity covariance matrix are considered (Figure 2). 200 data points are generated from each of the four Gaussians. The number of target clusters (K) is two. The two natural clusters are recovered by the EM algorithm without any constraint (Figure 2(a)). Ten multiple random restarts are used to avoid poor local minima.

Now suppose that prior information favors two vertical clusters instead of the more natural horizontal clusters. This prior information can be incorporated by constraining a data point in the leftmost (rightmost) top cluster to belong to the same cluster as a data point on the leftmost (rightmost) bottom cluster. We select 50 points randomly ($L = 50$) and link them to seven different points. To create more realistic constraints, a link can be absent with a probability of 0.05. The strength of the constraint is randomly drawn from the interval [0.6,1]. To demonstrate the importance of soft constraints, the constraints are corrupted with some noise: a data point is connected to a randomly chosen point with probability one minus the constraint strength. An example of the constraints is shown in Figure 2(b).

The proposed algorithm is run using the specified soft constraints and the obtained clustering solution is shown in Figure 2(c). The constraint information indeed helps to detect the preferred cluster structure, instead of natural clusters in Figure 2(a) when the constraints are absent. The soft constraints can be converted to hard constraints by changing all nonzero γ_{li} to 1. In this case, the proposed algorithm becomes equivalent to the algorithm in [9] for positive constraints. The result of using hard constraints is shown in Figure 2(d). While the estimated cluster structure is close to what we seek, the noise in the constraints notably distorts the detected clusters. This confirms that the use of soft constraints can significantly improve the robustness of mixture model clustering with hard constraints.

3.2 Real World Data Set

In the second experiment, we investigate how constraints can assist in obtaining superior cluster boundaries. Two data sets from the UCI machine learning



(a) Result without constraints (b) Soft constraints

(c) Result of soft constraints (d) Result of hard constraints

Fig. 2. The results of soft constraint clustering on a synthetic data set of 800 points. The ellipses represent the estimated Gaussian components. The solid lines in (b) correspond to the "strong" constraints, while the dotted lines correspond to the "weak" constraints.

repository are considered. The Iris data set (`iris`) has 150 points in 4D from 3 classes. The wine recognition data set[2] (`wine`) has 178 points with 13 features from 3 classes. For each data set, half of the points are used for training (learning the clusters), and the rest for testing (comparing the clusters obtained with the ground truth). A Gaussian is fit to each of the classes and the ambiguous data points (5% of the total number of data) are identified by examining the class posterior probabilities and the true class labels. The ambiguous data points are then constrained to be in the same cluster as the points near the center of the class. Examples of ambiguous data points are shown in Figure 3. For each data set, we randomly split them into two parts. As in the previous experiment, the strength of the constraint is drawn randomly from [0.6,1], and the constraint is additionally corrupted by noise. The clusters obtained (with soft constraints) are used to "classify" the other half of the data points. The experiment is re-

[2] The variables of `wine` are standardized to have means equal to zero and variances equal to one.

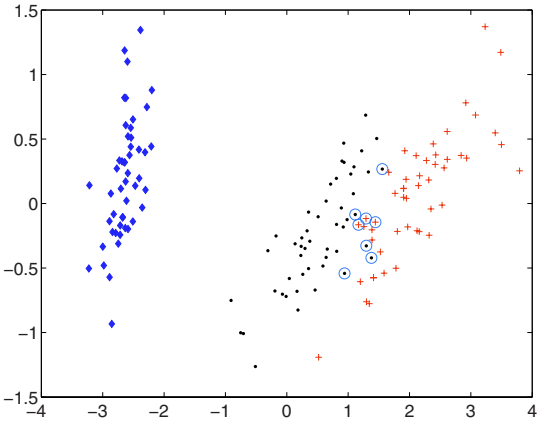


Fig. 3. The iris data set projected to the first two principal components. The eight ambiguous data points are circled.

peated 20 times. The error rates of "no constraint', "soft constraint" and "hard constraint" are 6.7%, 2.7% and 8% for `iris`, and 5.6%, 3.4% and 3.4% for `wine`. For both the data sets, soft constraints yield clusters that are at least as good as clusters obtained by hard constraints 19 times out of 20, with 8 ties for `iris` and 6 ties for `wine`. Soft constraints also give better clusters than no constraints (18 out of 20 for `iris` and 19 out of 20 for `wine`), with 6 ties for `iris` and 7 ties for `wine`. Soft constraint information indeed helps to identify the target clusters more accurately and tolerates potentially erroneous constraints.

4 Conclusion and Future Work

We have proposed a new EM algorithm for clustering in the presence of soft constraints. Experimental results demonstrate that the proposed approach is promising and can be superior to hard constraints in the presence of noise. One notable property of the proposed approach is its efficiency. Despite the apparent increase in the complexity of the model, no additional parameters need to be estimated when compared with a standard mixture of Gaussians. Also, the inference procedure is of similar complexity as the standard EM algorithm when each data point is associated with few group-labels. One limitation of the proposed algorithm is that it does not deal with the negative constraints. In principle, the graphical model can be extended in a manner similar to [9] to include the negative constraints. We choose not to do so, however, for two reasons. First, the addition of negative constraints results in only a slight improvement as reported in [9]. Secondly, the presence of negative constraints can increase the complexity of the graphical model and hence increase the inference complexity.

There are several directions for future work. The total strength of constraint information is currently determined by the number of constrained data points.

This can be undesirable as a large amount of data can dilute the constraint information. This relates to the fundamental issue of how to appropriately weight the information contained in the data and the constraints. One possibility is to include an additional penalty term in the likelihood function that balances the posterior probabilities of cluster labels with the constraints. The number of cluster, K, is assumed to be given. Since we are using a mixture model, the idea of minimum message length described in [16] can be adopted to the current algorithm to estimate K.

References

1. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall (1988)
2. Yu, S.X., Shi, J.: Segmentation given partial grouping constraints. IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004) 173–183
3. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning via equivalence constraints, with applications to the enhancement of image and video retrieval. In: Proc. IEEE Confernce on Computer Vision and Pattern Recognition. (2003)
4. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained k-means clustering with background knowledge. In: Proc. International Conference on Machine Learning. (2001) 577–584
5. Wagstaff, K., Cardie, C.: Clustering with instance-level constraints. In: Proc. International Conference on Machine Learning. (2000) 1103–1110
6. Wagstaff, K.: Intelligent Clustering with Instance-Level Constraints. PhD thesis, Department of Computer Science, Cornell University (2002)
7. 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: Proc. International Conference on Machine Learning. (2002) 307–314
8. Kamvar, S., Klein, D., Manning, C.D.: Spectral learning. In: Proc. of the Eighteenth International Joint Conference on Artificial Intelligence, MIT Press (2003)
9. Shental, N., Bar-Hillel, A., Hertz, T., Weinshall, D.: Computing gaussian mixture models with EM using equivalence constraints. In: Advances in Neural Information Processing Systems 16, MIT Press (2004)
10. Yu, S.X., Shi, J.: Grouping with bias. In: Advances in Neural Information Processing Systems 13, MIT Press (2001)
11. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning, with application to clustering with side-information. In: Advances in Neural Information Processing Systems 15, Cambridge, MA, MIT Press (2003)
12. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. In: Proc. of the 43d Annual IEEE Symp. on Foundations of Computer Science. (2002)
13. Charikar, M., Guruswami, V., Wirth, A.: Clustering with qualitative information. In: Proc. of the 44th Annual IEEE Symposium on Foundations of Computer Science. (2003)
14. Demaine, E.D., Immorlica, N.: Correlation clustering with partial information. In: Proc. of the 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, Princeton, New Jersey (2003)
15. McLachlan, G., Peel, D.: Finite Mixture Models. John Wiley & Sons, New York (2000)
16. Figueiredo, M.A.T., Jain, A.K.: Unsupervised learning of finite mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 381–396