To appear in SIAM data mining 2005

Model-based Clustering With Probabilistic Constraints

Martin H. C. Law* Alexander Topchy* Anil K. Jain*

Abstract

The problem of clustering with constraints is receiving increasing attention. Many existing algorithms assume the specified constraints are correct and consistent. We take a new approach and model the uncertainty of constraints in a principled manner by treating the constraints as random variables. The effect of specified constraints on a subset of points is propagated to other data points by biasing the search for cluster boundaries. By combining the a posteriori enforcement of constraints with the log-likelihood, we obtain a new objective function. An EM-type algorithm derived by variational method is used for efficient parameter estimation. Experimental results demonstrate the usefulness of the proposed algorithm. In particular, our approach can identify the desired clusters even when only a small portion of data participates in constraints.

1 Introduction

The goal of (partitional) clustering [8] is to discover the "intrinsic" grouping of a data set without any class labels. Clustering is an ill-posed problem because the absence of class labels obfuscates the goal of analysis: what is the proper definition of "intrinsic"? In some applications, however, there is a preference for certain clustering solutions. This preference or extrinsic information is often referred to as *side-information*. Examples include alternative metrics between objects, orthogonality to a known partition, additional labels or attributes, relevance of different features and ranks of the objects.

Perhaps the most natural type of side-information in clustering is a set of *constraints*, which specify the relationship between cluster labels of different objects. Constraints are naturally available in many clustering applications. For instance, in image segmentation one can have partial grouping cues for some regions of the image to assist in the overall clustering [20]. 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 [2]; such groupings may be used for semi-supervised clustering of the entire database.

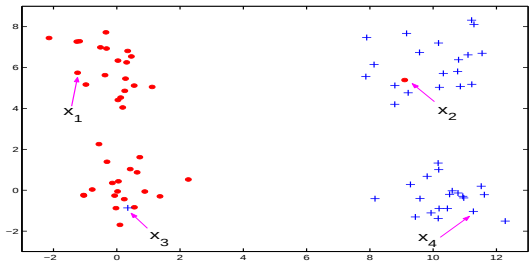


Figure 1: A counter-intuitive clustering solution with pairs (x_1, x_2) and (x_3, x_4) in must-link constraints. The cluster labels of the neighbors of x_2 and x_3 are different from those of x_2 and x_3. This is the consequence of computing cluster labels instead of cluster boundaries.

A pairwise *must-link/positive* constraint corresponds to the requirement that two objects should be placed in the same cluster. A pairwise *must-not-link/negative* constraint, on the contrary, means that two objects should be placed in different clusters. Positive constraints tend to be more informative, and the experimental results in [17] suggest that negative constraints only help the clustering results marginally, at the expense of increased computation. Therefore, in this paper we shall focus on positive constraints, though negative constraints can also be incorporated in our model [14]. Note that clustering with constraints is different from learning with unlabelled data, because constraints only specify the relative relationship between labels.

It is important that the effect of constraints be propagated: not only the labels of points involved with constraints should be affected, but also their neighbors [12]. Without this, one can obtain a weird clustering solution, as shown in Figure 1. This intuitive requirement of constraint propagation, unfortunately, is not satisfied by many existing approaches, which estimate the *cluster labels* directly. Our algorithm instead searches for *cluster boundaries* that are most consistent with the constraints and the data.

Different algorithms have been proposed for clustering with constraints. COBWEB and k-means with constraints were proposed in [18] and [19], respectively. Spectral clustering has also been modified to work with

*Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48823, USA. This research was supported by ONR contract # N00014-01-1-0266.

constraints [11, 20]. Metric-learning and clustering with constraints in k-means were considered simultaneously in [4], and was extended to a Hidden Markov random field formulation in [3]. Correlation clustering [1] uses only constraints for clustering. Coordinated conditional information bottleneck [6] discovers novel cluster structure in a data set.

We earlier had proposed a graphical model to represent constraints in model-based clustering [13]. In this paper, we extend that model by (i) incorporating a posterior term in the objective function that corresponds to the enforcement of constraints, (ii) introducing tradeoff parameters of such terms as the strengths of constraints, and (iii) deriving an EM-like algorithm for parameter estimation based on variational method.

2 Method

Let $\mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ be the set of d-dimensional data to be clustered by mixture model-based clustering [5] with K clusters. Let $z_i \in \{1, 2, \ldots, K\}$ be the iid (hidden) cluster label of $\mathbf{y}_i$, and let $q_j(.|\theta_j)$ be the probability distribution of the j-th component with parameter θ_j, which is assumed to be Gaussian. Extensions to other type of component distributions are straightforward. Let α_j be the prior probability of the j-th cluster. Consider *group constraints*, a generalization of pairwise constraints, where multiple data points (possibly more than two) are constrained to be in the same cluster. Let w_l be the cluster label of the l-th constraint group, with L as the total number of groups. The random variable z_i takes the value of w_l when the constraint on $\mathbf{y}_i$ is enforced. Introduce the random variable v_i, which corresponds to the constraint on $\mathbf{y}_i$. When it is "on" (nonzero), the constraint is enforced, i.e., $z_i = w_l$. When it is "off" (zero), the constraint is disabled, and z_i is distributed independently according to its prior probabilities. The probability that the constraint is "on" corresponds to the certainty of the constraint. For example, to represent the constraints for the data in Figure 3, we should assign $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4$ to the first group and $\mathbf{y}_5, \mathbf{y}_6$ to the second group. Since there are two groups, $L = 2$. If we assume the confidence of all the constraints to be 0.5, the first group constraint is represented by setting the parameters $\gamma_{i2} = 0$ and $\gamma_{i1} = 0.5$ for $i = 1, 2, 3, 4$, whereas the second group constraint is represented by $\gamma_{i1} = 0$ and $\gamma_{i2} = 0.5$ for $i = 5, 6$. The meaning of γ_{il} will be defined shortly after.

The presence of constraints introduces dependence only among z_i. Different $\mathbf{y}_i$ are still independent given z_i. Therefore, our model can be factorized as

$$P(\mathcal{Y}) = \sum_{\mathbf{z},\mathbf{v},\mathbf{w}} \Big(\prod_i P(\mathbf{y}_i|z_i) P(z_i|v_i, \mathbf{w}) P(v_i) \Big) \prod_{l=1}^{L} P(w_l).$$

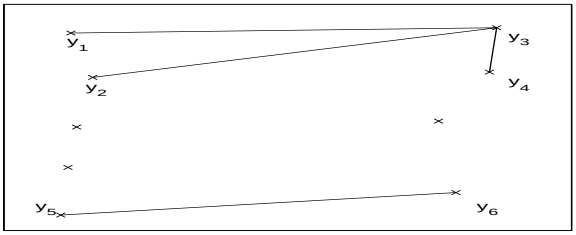


Figure 3: An example set of constraints. Points that should be put in the same cluster are joined by lines.

The rest of the model is specified as follows:

$$\begin{aligned} P(w_l = j) &= \alpha_j, \quad 1 \le l \le L, 1 \le j \le K, \\ P(v_i = l) &= \gamma_{il}, \quad 1 \le i \le N, 1 \le l \le L, \\ P(z_i = j|v_i, \mathbf{w}) &= \begin{cases} \alpha_j & \text{if } v_i = 0 \\ \delta_{w_l,j} & \text{if } v_i = l \end{cases}, \\ P(\mathbf{y}_i|z_i = j) &= q_j(\mathbf{y}_i), \end{aligned} \tag{2.1}$$

where $\mathbf{z} = (z_1, \ldots, z_n)$, $\mathbf{v} = (v_1, \ldots, v_n)$ and $\mathbf{w} = (w_1, \ldots, w_L)$ are the hidden variables. Here, γ_{il} denotes the probability that the constraint of tying $\mathbf{y}_i$ to the l-th group is "on". The values of γ_{il} are either specified by the user to represent the confidence of different constraints, or they can be set to 0.5 when the certainties of the constraints are unknown. An example of such a model with seven data points and three group labels is shown in Figure 2. The model in [17] is a special case of this model when all γ_{il} are binary.

An EM algorithm can be derived to learn the parameters of this model by maximizing the data log-likelihood [13]. The M-step is described by

$$a_j = \sum_{l=1}^{L} P(w_l = j|\mathcal{Y}) + \sum_{i=1}^{N} P(v_i = 0, z_i = j|\mathcal{Y}), \tag{2.2}$$

$$\hat{\alpha_j} = \frac{a_j}{\sum_{j'=1}^{K} a_{j'}}, \tag{2.3}$$

$$\hat{\boldsymbol{\mu}}_j = \frac{\sum_{i=1}^{N} P(z_i = j|\mathcal{Y}) \mathbf{y}_i}{\sum_{i=1}^{N} P(z_i = j|\mathcal{Y})}, \tag{2.4}$$

$$\hat{\mathbf{C}}_j = \frac{\sum_{i=1}^{N} P(z_i = j|\{\mathbf{y}_i\})(\mathbf{y}_i - \hat{\boldsymbol{\mu}}_j)(\mathbf{y}_i - \hat{\boldsymbol{\mu}}_j)^T}{\sum_{i=1}^{N} P(z_i = j|\mathcal{Y})}. \tag{2.5}$$

Here, the j-th component is assumed to be a Gaussian with mean $\boldsymbol{\mu}_j$ and covariance $\mathbf{C}_j$, $\theta_j = (\boldsymbol{\mu}_j, \mathbf{C}_j)$. The E-step consists of computing the probabilities $P(w_l = j|\mathcal{Y})$, $P(z_i = j|\mathcal{Y})$ and $P(v_i = 0, z_i = j|\mathcal{Y})$, which can be done by standard Bayesian network inference algorithms like belief propagation or junction tree [10]. 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

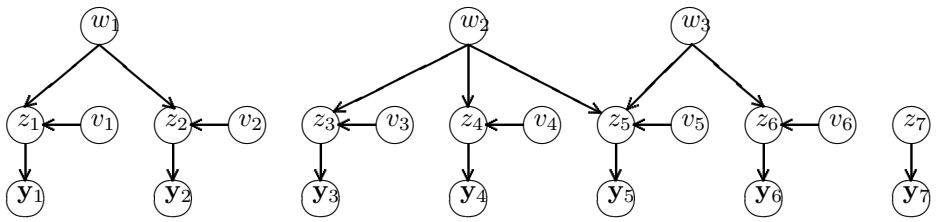


Figure 2: An example of the graphical model with constraint uncertainties for 9 points in 3 groups. Note that each connected component in the graph is a polytree and hence the belief propagation algorithm can be used to calculate the probabilities exactly.

EM algorithm when there are only positive constraints that tie each of the z_i to one group label.

2.1 Modification of the Objective Function The proposed graphical model can handle the uncertainties of constraints elegantly. However, the tradeoff between the constraint information and the data information cannot be controlled explicitly. To cope with this, an additional term that represents the *a posteriori* enforcement of constraints is included in the objective function. This is a distinct characteristic of the proposed model: since each constraint is represented as a random variable, we can consider its posterior probability. The posterior probability that a constraint is "on" reflects how strongly a constraint is enforced by the current parameter estimate. Instead of the binary statement that a constraint is satisfied or violated, we can now consider the *partial* degree of satisfaction of a constraint. This way, the violation of constraints is measured more accurately. Formally, the new objective function is

$$E = \log P(\mathcal{Y}|\Theta) + \sum_{i,l} \beta_{il} \log P(v_i = l|, \mathcal{Y}, \Theta), \tag{2.6}$$

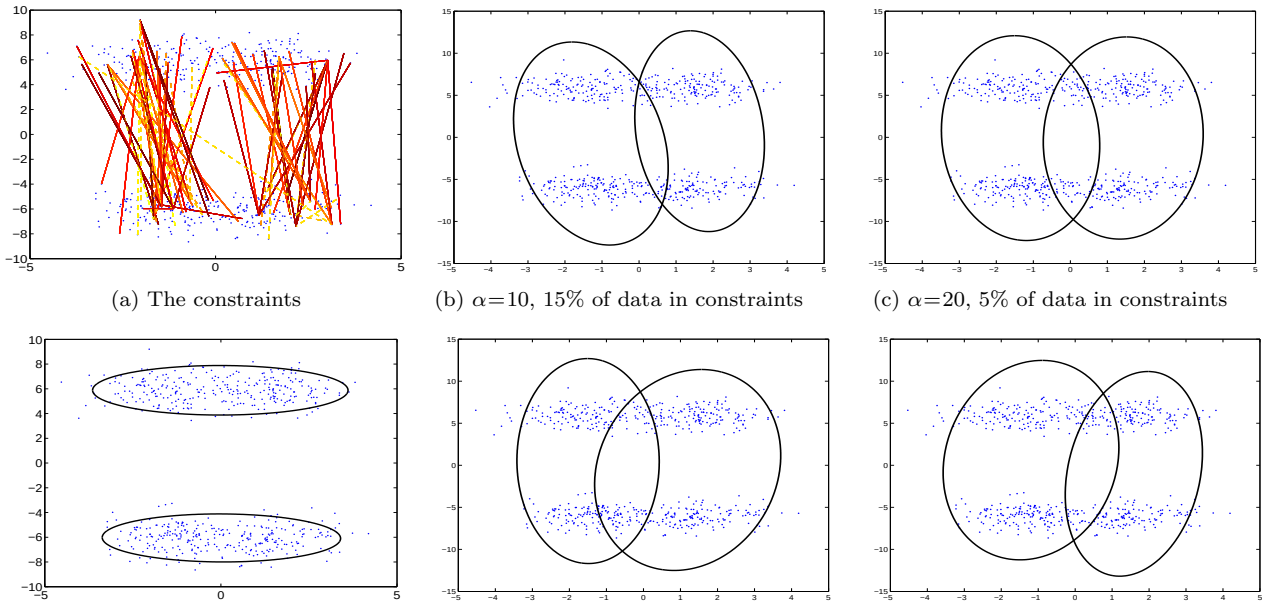
with the convention that β_{il} is zero when $P(v_i = l) = 0$. The posterior enforcement of the constraint on $\mathbf{y}_i$ is represented by $\log P(v_i = l|, \mathcal{Y}, \Theta)$, because the event $v_i = l$ corresponds to the constraint that z_i is tied to w_l. The strengths of the constraints are represented by β_{il}, which are the user-specified tradeoff parameters between the influence of the posterior probability and the data log-likelihood. In this paper, we set $\beta_{il} = \alpha\tau_{il}$, where α is a global constraint strength parameter and τ_{il} represents the goodness of the constraint tying $\mathbf{y}_i$ to the l-th group. If τ_{il} is unknown, we can assume that all constraints are equally important and set τ_{il} to one. The only parameter that needs to be specified is α.

Direct optimization of E is difficult and we resort to variational method. Due to limitation of space, we defer the derivation and the update formulae in the long version of the paper [14]. In brief, there is no change in the E-step, whereas the M-step (Equations (2.3) to (2.5)) is modified by replacing the cluster label probabilities with a weighted sum of constraint satisfaction and cluster label probabilities.

3 Experiments

3.1 Synthetic Data Set Four 2D Gaussian distributions with mean vectors $\left[\begin{smallmatrix}1.5\\6\end{smallmatrix}\right]$, $\left[\begin{smallmatrix}-1.5\\6\end{smallmatrix}\right]$, $\left[\begin{smallmatrix}-1.5\\-6\end{smallmatrix}\right]$, $\left[\begin{smallmatrix}1.5\\-6\end{smallmatrix}\right]$, and identity covariance matrix are considered. 150 data points are generated from each of the four Gaussians. The number of target clusters (K) is two. In the absence of any constraints, two horizontal clusters are successfully discovered by the EM algorithm (Figure 4(d)). Ten multiple random restarts were 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 in the leftmost (rightmost) bottom cluster. To determine the strength of a constraint, τ_{il} is randomly drawn from the interval [0.6,1], and we set $\beta_{il} = \alpha\tau_{il}$, where α is the global constraint strength specified by the user. To demonstrate the importance of constraint uncertainty, the constraints are corrupted with noise: a data point is connected to a randomly chosen point with probability $1 - \tau_{il}$. An example set of constraints with 15% of data points involved in the constraints is shown in Figure 4(a). Different portions of points participating in constraints are studied in the experiment. In all cases, the proposed algorithm can recover the desired two "vertical" clusters, whereas other algorithms (such as [17]) fail. It is worthy of note that our algorithm can recover the target structure with as few as 2.5% of the data points participating in constraints. If a clustering with constraint algorithm that deduces cluster labels directly is used, the anomaly illustrated in Figure 1 can happen, because of the small number of constraints.

3.2 Real World Data Sets Experiments are also performed on three data sets in the UCI machine learning repository (Table 1). For each data set, K is set



(a) The constraints (b) α=10, 15% of data in constraints (c) α=20, 5% of data in constraints

(d) Result without constraints (e) α=20, 10% of data in constraints (f) α=40, 2.5% of data in constraints

Figure 4: Results of the proposed algorithm when different number of data points participate in constraints.

	Full name	N	D	K	d	m
`wdbc`	Wisconsin breast cancer	569	30	2	10	6
`derm`	Dermatology	366	33	6	5	12
`image`	Image Segmentation	2410	18	7	10	14

Table 1: Data sets used in the experiments. N: size of data. D: no. of features. K: no. of classes. d: PCA dimension. m: no. of points labelled by a teacher.

to the true number of classes. The distributed learning scenario described in [17], where different teachers label a small subset of the data, is used to generate the constraints. Each teacher labels $2K$ or $3K$ data points, depending on the size of the data. The labels assigned by the teachers are corrupted with noise with probability based on the confidence of those labels. The confidence is used as constraint strengths as in the case for synthetic data. The number of teachers is determined by the percentage of points in constraints. PCA is used to reduce the dimensionality of the data sets to d to ensure there are a sufficient number of data points to estimate the covariance matrix, with d determined by the size of the data. For each data set, half of the data is used for clustering, while the other half is used to evaluate the clusters based on the ground truth labels. We compare the performance of soft constraints [13], hard constraints (equivalent to [17]) and the proposed method in Table 2. The proposed algorithm leads to superior clusters when compared with the results of hard and soft constraints. The improvement due to constraints is not very significant for the Dermatology data set, because the standard EM algorithm is able to find a good quadratic cluster boundary. The degradation of performance in `image` for hard constraints is due to the existence of erroneous constraints.

4 Discussion

The proposed algorithm can be viewed from alternative perspectives. It can be regarded as training a mixture of Gaussians in a discriminative manner [9], with the constraints serving as relaxed label information. If different Gaussians share the same covariance matrix, EM algorithm is related to performing discriminant analysis with posterior probabilities as weights [7]. This provides an alternative justification of our approach even when the Gaussian assumption is not satisfied, because the EM algorithm finds the clusters that are best separated.

The global constraint strength α is the only parameter that requires tuning. In practice, α is chosen automatically by setting apart a set of "validation constraints" or "validation teachers", which are not used to estimate the clusters. The smallest value of α that leads to clusters that violate the validation information the least is chosen. Note that we do not observe significant overfitting in our experiments. So, one may as well use the value of α that leads to the smallest violation of the training constraints.

	20% of data in constraints					10% of data in constraints					5% of data in constraints				
	H	S	P	$P \geq H$	$P \geq S$	H	S	P	$P \geq H$	$P \geq S$	H	S	P	$P \geq H$	$P \geq S$
`wdbc`	6.5	1.9	16.7	9	10	2.8	1.5	13.3	9	9	3.4	-1.1	9.4	9	10
`derm`	0.5	1.0	3.5	5	6	2.9	2.5	5.2	5	6	1.4	2.3	4.5	6	9
`image`	-2.6	2.6	6.1	8	10	-3.1	0.5	9.0	9	8	-5.8	2.2	5.0	9	6

Table 2: Results on real world data sets. Average improvements in accuracy (in %) with respect to no constraints for soft constraints (S), hard constraints (H), posterior constraints, i.e., the proposed algorithm, (P), are shown. The number of times that the proposed algorithm outperforms the other two constraint algorithms in 10 runs is also shown.

5 Conclusion

We have proposed a graphical model with the constraints as random variables. This principled approach enables us to state the prior certainty and posterior enforcement of a constraint. The model is more robust towards noisy constraints, and it provides a more general approach to estimate constraint violation. Metric learning is automatic because covariance matrices are estimated. The use of variational method provides an efficient approach for parameter estimation. Experimental results show the utility of the proposed method. For future work, we plan to estimate the number of clusters automatically. Can traditional criteria like AIC, BIC or MDL be modified to work in the presence of constraints? A kernel version of this algorithm can be developed for clusters with general shapes.

References

[1] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. In *Proc. Annual IEEE Symp. on Foundations of Computer Science*, 2002.

[2] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning via equivalence constraints, with applications to the enhancement of image and video retrieval. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[3] S. Basu, M. Bilenko, and R.J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proc. ACM SIGKDD, Intl. Conf. on Knowledge Discovery and Data Mining*, 2004.

[4] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proc. Intl. Conf. on Machine Learning*, 2004.

[5] M. A. T. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.

[6] D. Gondek and T. Hofmann. Non-redundant data clustering. In *Proc. Intl. Conf. on Data Mining*, 2004.

[7] T. Hastie and R. Tibshirani. Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society series B*, 58:158–176, 1996.

[8] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, September 1999.

[9] T. Jebara and A. Pentland. Maximum conditional likelihood via bound maximization and the CEM algorithm. In *Advances in Neural Information Processing Systems 11*, pages 494–500. MIT Press, 1998.

[10] M. I. Jordan. *Learning in Graphical Models*. Institute of Mathematical Statistics, 1999.

[11] S. Kamvar, D. Klein, and C. D. Manning. Spectral learning. In *Proc. Intl. Joint Conf. on Artificial Intelligence*, pages 561–566, 2003.

[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. In *Proc. Intl. Conf. on Machine Learning*, pages 307–314, 2002.

[13] M. H. Law, A. Topchy, and A. K. Jain. Clustering with soft and group constraints. In *Proc. Joint IAPR International Workshops on Structural, Syntactic, And Statistical Pattern Recognition*, 2004.

[14] M. H. C. Law, A. P. Topchy, and A. K. Jain. Model-based clustering with soft and probabilistic constraints. Technical report, Michigan State University, 2004.

[15] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: an empirical study. In *Proc. Conf. on Uncertainty in Artificial Intelligence*, San Francisco, 1999.

[16] S. T. Roweis, L. K. Saul, and G. E. Hinton. Global coordination of local linear models. In *Advances in Neural Information Processing Systems 14*, pages 889–896. MIT Press, 2002.

[17] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing gaussian mixture models with EM using equivalence constraints. In *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.

[18] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *Proc. Intl. Conf. on Machine Learning*, pages 1103–1110, 2000.

[19] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proc. Intl. Conf. on Machine Learning*, pages 577–584, 2001.

[20] S. X. Yu and J. Shi. Segmentation given partial grouping constraints. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(2):173–183, 2004.