Constrained Spectral Clustering through Affinity Propagation

Zhengdong Lu
CSEE, OGI, Oregon Health & Science University
zhengdon@csee.ogi.edu

Miguel Á. Carreira-Perpiñán
EECS, University of California, Merced
mcarreira-perpinan@ucmerced.edu

Abstract

Pairwise constraints specify whether or not two samples should be in one cluster. Although it has been successful to incorporate them into traditional clustering methods, such as K-means, little progress has been made in combining them with spectral clustering. The major challenge in designing an effective constrained spectral clustering is a sensible combination of the scarce pairwise constraints with the original affinity matrix. We propose to combine the two sources of affinity by propagating the pairwise constraints information over the original affinity matrix. Our method has a Gaussian process interpretation and results in a closed-form expression for the new affinity matrix. Experiments show it outperforms state-of-the-art constrained clustering methods in getting good clusterings with fewer constraints, and yields good image segmentation with user-specified pairwise constraints.

There is an emerging interest in incorporating pairwise constraints into clustering algorithms in the machine learning and data mining communities. In addition to the data values, we assume there are a number of instance-level constraints on cluster assignment. More specifically, we consider the following two types of *pairwise constraints*: **must-link** constraints, which specify that two samples should be assigned to the same cluster; and **cannot-link** constraints, which specify that two samples should be assigned to different clusters. Pairwise constraints may arise from knowledge of domain experts [15], perceived similarity (or dissimilarity) [11], or even common sense [12]. There are generally two categories of methods using pairwise constraints in clustering. The first category adapts the traditional centroid-based clustering methods, such as k-means [15, 1] or Gaussian mixtures [12, 11] to follow the pairwise constraints. The second category tries to learn a Mahalanobis distance that minimizes the distance between must-linked samples and maximizes the distance between cannot-linked samples; this can be done in the original vector space [16] or in kernel feature space [2]. After the metric learning, a clustering method such as k-means is used to get the final clustering result.

In this paper, we try to adapt to using pairwise constraints another popular clustering method, spectral clustering, on which only some preliminary effort [10] is known to us. The major difficulty is that pairwise constraints specify a highly informative affinity measure, but that is only available for a small number of pairs. For the rest, we have to rely on the abundant but less informative affinity measure derived from the feature vectors (or provided for each pair). Prior to this paper, there is no natural way to blend the two affinity measures. The method of [10] simply uses the Gaussian kernel as the affinity but replacing entries for must-linked pairs with 1 and for cannot-linked pairs with 0. Not surprisingly, this method generally does not work very well since the effect of the pairwise constraints is limited to a small number of entries in the affinity matrix. To deal with this difficulty we propose a way to propagate, in a way consistent with the given affinities, the pairwise constraints from a few specified sample pairs to the rest of the entries in the affinity matrix, thus increasing the effect of the pairwise constraints. The paper is organized as follows: section 1 introduces the basic idea of affinity propagation (with appendix A giving an alternative interpretation), section 2 gives clustering algorithms, section 3 gives experimental results and section 4 discusses related work.

1. Affinity Propagation

Let us interpret the original affinity matrix $\mathbf{K} \succ 0$ as the covariance matrix of a zero-mean Gaussian process f:

$$P(\mathbf{f}) = |2\pi\mathbf{K}|^{-N/2}\, e^{-\frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f}} \qquad (1)$$

where $\mathbf{f} = (f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_N))^T$ at the data points. Viewing $f(\mathbf{x}_i) \in \mathbb{R}$ as a continuous label of $\mathbf{x}_i$ ($f > 0$: label 1, $f < 0$: label 2), we find that $\mathrm{E}\left\{f(\mathbf{x}_i)f(\mathbf{x}_j)\right\} = \mathrm{cov}\left\{f(\mathbf{x}_i), f(\mathbf{x}_j)\right\} = \mathbf{K}_{ij}$ provides a natural measurement of the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ (namely, how often $\mathbf{x}_i$ and $\mathbf{x}_j$ are co-labelled). We now treat the given pairwise constraints as a kind of observation:

(M) If we know $\mathbf{x}_i$ and $\mathbf{x}_j$ are must-linked, we assume it is observed that $f(\mathbf{x}_i) - f(\mathbf{x}_j) \sim \mathcal{N}(0, \epsilon_m^2)$

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

(C) If we know $\mathbf{x}_i$ and $\mathbf{x}_j$ are cannot-linked, we assume it is observed that $f(\mathbf{x}_i) + f(\mathbf{x}_j) \sim \mathcal{N}(0, \epsilon_c^2)$

where ϵ_m and ϵ_c soften the constraints. Call Ω the observation described above and $\mathcal{M}$ and $\mathcal{C}$ the set of all must-links and cannot-links, respectively. The likelihood $P(\Omega|\mathbf{f})$ of Ω given $\mathbf{f}$ is then proportional to

$$\exp\Big(- \sum_{ij\in\mathcal{M}} \frac{(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2}{2\epsilon_m^2} - \sum_{ij\in\mathcal{C}} \frac{(f(\mathbf{x}_i) + f(\mathbf{x}_j))^2}{2\epsilon_c^2} \Big).$$

From Bayes' rule, the posterior probability of $\mathbf{f}$ given Ω is:

$$P(\mathbf{f}|\Omega) \propto \exp\Big(- \frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f} \Big) \times$$
$$\exp\Big(- \sum_{ij\in\mathcal{M}} \frac{(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2}{2\epsilon_m^2} - \sum_{ij\in\mathcal{C}} \frac{(f(\mathbf{x}_i) + f(\mathbf{x}_j))^2}{2\epsilon_c^2} \Big).$$

We propose to use $\overline{\mathbf{K}}_{ij} \equiv \mathrm{E}\left\{f(\mathbf{x}_i)f(\mathbf{x}_j)|\Omega\right\}$ as the new affinity between $\mathbf{x}_i$ and $\mathbf{x}_j$. Since $\mathbf{f}|\Omega$ is still a Gaussian and $\mathrm{E}\left\{\mathbf{f}|\Omega\right\} = \mathbf{0}$, we have the following key result:

$$\overline{\mathbf{K}} = (\mathbf{K}^{-1} + \mathbf{M})^{-1} = \mathbf{K} - \mathbf{K}(\mathbf{I} + \mathbf{M}\mathbf{K})^{-1}\mathbf{M}\mathbf{K} \qquad (2)$$

$$\mathbf{M}_{ij} = \begin{cases} \frac{m_i}{\epsilon_m^2} + \frac{c_i}{\epsilon_c^2} & \text{if } i = j \\ -\frac{1}{\epsilon_m^2} & (i,j) \in \mathcal{M} \\ \frac{1}{\epsilon_c^2} & (i,j) \in \mathcal{C} \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

where we assume $\mathbf{x}_i$ is must-linked to m_i samples and cannot-linked to c_i samples, and the $N \times N$ matrix $\mathbf{M}$ encapsulates the constraint information. When ϵ_m (ϵ_c) $\to 0$, we get hard must-links (cannot-links) and when ϵ_m (ϵ_c) $\to \infty$ we get no constraint. Clearly $\overline{\mathbf{K}} = (\mathbf{K}^{-1} + \mathbf{M})^{-1} \succ 0$, but unlike an affinity matrix from the Gaussian kernel, it may contain negative entries. Computing $\overline{\mathbf{K}}$ adds almost no overhead since it requires inverting a small matrix of dimension $\mathcal{O}(m_i + c_i)$ (see below). $P(\mathbf{f}|\Omega)$ can also be seen as the distribution that minimises the divergence to $P(\mathbf{f})$ (eq. (1)) while satisfying certain constraints (see appendix).

Analysis: equivalent kernels Here we work out in closed form the new affinity $\overline{\mathbf{K}}$ when there is only one constraint (must-link or cannot-link). We show that $\overline{\mathbf{K}}$ can be represented by an *equivalent kernel* which gives an intuition about the propagation of affinity. Assume a given symmetric affinity matrix $\mathbf{K}$ of $N \times N$ satisfying $K_{ii} = 1$ and $0 \le K_{ij} < 1 \; \forall i \neq j$ (e.g. Gaussian affinities). We place a single link between points 1 and 2. Write

$$\mathbf{K} = \begin{pmatrix} K_{11} & K_{12} & K_{13} & \cdots \\ K_{21} & K_{22} & K_{23} & \cdots \\ K_{31} & K_{32} & \mathbf{K}_3 & \\ \cdots & \cdots & & \end{pmatrix} \qquad \begin{aligned} \mathbf{u}^- &= \mathbf{K}_{\bullet 1} - \mathbf{K}_{\bullet 2} \\ \mathbf{u}^+ &= \mathbf{K}_{\bullet 1} + \mathbf{K}_{\bullet 2} \end{aligned} \qquad (4)$$

where $\mathbf{K}_3$ is a block of $(N-2) \times (N-2)$ and $\mathbf{K}_{\bullet 1}$ denotes column 1 of $\mathbf{K}$, so $\mathbf{u}^+$ and $\mathbf{u}^-$ are sum and difference column vectors of $N \times 1$. Using the same block structure, the $\mathbf{M}$ matrix is

$$\mathbf{M}_m = \frac{1}{2\epsilon_m^2} \begin{pmatrix} 1 & -1 & \mathbf{0} \\ -1 & 1 & \mathbf{0} \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{0} \end{pmatrix} \qquad \mathbf{M}_c = \frac{1}{2\epsilon_c^2} \begin{pmatrix} 1 & 1 & \mathbf{0} \\ 1 & 1 & \mathbf{0} \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{0} \end{pmatrix} \qquad (5)$$

for must-links and cannot-links, respectively. From (2) we need the inverse of $(\mathbf{I} + \mathbf{M}\mathbf{K})$. This exists if $K_{ii} > K_{ij}$ $\forall i,j$ and can be computed in closed form using $\left(\begin{smallmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{smallmatrix}\right)^{-1} = \left(\begin{smallmatrix} \mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{smallmatrix}\right)^{-1}$. (When there are $L > 1$ links, only a small matrix $\mathbf{A}$ of $2L \times 2L$ need be inverted, so computing $\overline{\mathbf{K}}$ in practice is fast since $L \ll N$.)

Applying (2) we find that the new affinity matrix $\overline{\mathbf{K}}$ is the weighted average of the original affinity matrix $\mathbf{K}$ and a link affinity $\mathbf{K}'$ matrix caused by the link:

$$\overline{\mathbf{K}}_m = \frac{1}{\epsilon_m^2 + 1 - K_{12}} (\epsilon_m^2 \mathbf{K} + (1 - K_{12})\mathbf{K}'_m) \qquad (6)$$

$$\overline{\mathbf{K}}_c = \frac{1}{\epsilon_c^2 + 1 + K_{12}} (\epsilon_c^2 \mathbf{K} + (1 + K_{12})\mathbf{K}'_c). \qquad (7)$$

The weight is controlled by ϵ, so that for a hard constraint we obtain $\lim_{\epsilon\to 0} \overline{\mathbf{K}} = \mathbf{K}'$ (i.e., the effect is given purely by $\mathbf{K}'$) and for no constraint $\lim_{\epsilon\to\infty} \overline{\mathbf{K}} = \mathbf{K}$ (i.e., no effect). When $K_{12} \to 1$ we have $\overline{\mathbf{K}}_c \to \mathbf{K}$ but $\overline{\mathbf{K}}_m \nrightarrow \mathbf{K}$; thus, a must-link between identical points has no effect, but a cannot-link between identical points does have an effect. Let us now focus on the hard constraint case ($\epsilon = 0$, $\overline{\mathbf{K}} = \mathbf{K}'$). The link affinity $\mathbf{K}'_m$ (resp. $\mathbf{K}'_c$) is symmetric and independent of ϵ_m (resp. ϵ_c). It has columns 1 and 2 both equal to $\frac{1}{2}\mathbf{u}^+$ for must-link (i.e., an average affinity) and equal to $\frac{1}{2}\mathbf{u}^-$ and $-\frac{1}{2}\mathbf{u}^-$, respectively, for cannot-link (i.e., an average affinity difference). Columns 1, 2 give the affinity between the link sites $(1,2)$ and the remaining points $(3,4,\dots)$. We can see that the must-link equalises points $1,2$ while the cannot-link polarises them with opposite sign. Now consider the $3{:}N \times 3{:}N$ block of $\mathbf{K}'$, which gives the affinities for the remaining points $3,4,\dots$ It is given by:

$$\mathbf{K}_{m,3} - \frac{\mathbf{u}^-_{3:N}(\mathbf{u}^-_{3:N})^T}{2(1 - K_{12})} \qquad \mathbf{K}_{c,3} - \frac{\mathbf{u}^+_{3:N}(\mathbf{u}^+_{3:N})^T}{2(1 + K_{12})} \qquad (8)$$

so the affinities undergo a negative rank–1 update. We can interpret them by means of an **equivalent kernel** K'_{ij}:

$$\text{must-link: } K'_{ij} = K_{ij} - \frac{(K_{i1} - K_{i2})(K_{j1} - K_{j2})}{2(1 - K_{12})}$$

$$\text{cannot-link: } K'_{ij} = K_{ij} - \frac{(K_{i1} + K_{i2})(K_{j1} + K_{j2})}{2(1 + K_{12})}.$$

By substituting the original affinity kernel (e.g. Gaussian K_{ij}) we obtain the form of K'_{ij}. The effect of the affinity propagation in the problem with constraints is the same as

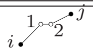
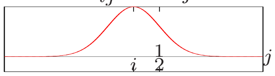
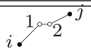
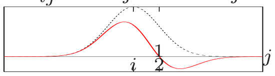
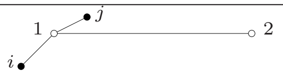
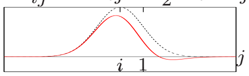
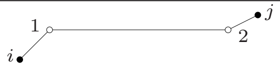
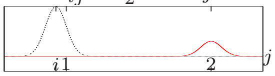
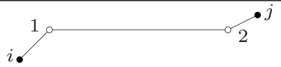
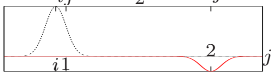
	must-link		cannot-link	
	i, j close	i, j far	i, j close	i, j far
$1, 2$ close $(K_{12} \approx 1)$	 $K'_{ij} \approx K_{ij}$ 	same as must-link $(1, 2$ close; i, j close)	 $K'_{ij} \approx K_{ij} - K_{i1}K_{j1}$ 	same as must-link $(1, 2$ close; i, j close)
$1, 2$ far $(K_{12} \approx 0)$	 $K'_{ij} \approx K_{ij} - \frac{1}{2}K_{i1}K_{j1}$ 	 $K'_{ij} \approx \frac{1}{2}K_{i1}K_{j2}$ 	same as must-link $(1, 2$ far; i, j close)	 $K'_{ij} \approx -\frac{1}{2}K_{i1}K_{j2}$ 

Table 1. Equivalent kernels for a single must-link or cannot-link for relevant arrangements (illustrated by a diagram) of the link sites $(1, 2)$ and the point pair (i, j) under consideration. The plot shows the original affinity kernel $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ (dashed black, assumed Gaussian) and the new, equivalent one $k'(\mathbf{x}_i, \mathbf{x}_j)$ (red), as a function of j (X axis) for fixed i. The point locations are labelled as $1, 2, i, j$.

having a problem without constraints with affinities given by the equivalent kernel. It is instructive to consider the equivalent kernel in specific cases for i, j. Firstly note that for point pairs (i, j) where both i, j are far (in the sense of $\mathbf{K}$) from $1, 2$, the affinities are practically unchanged $(K'_{ij} \approx K_{ij})$ since $K_{i1}, K_{i2}, K_{j1}, K_{j2} \approx 0$. For pairs i, j where both i and j are near at least one of $1, 2$, table 1 summarises the results. When must-link links faraway points (row "1, 2 far"), e.g. if $1, 2$ are in different clusters, then the affinity of a pair i, j: (a) increases if i, j are each near a different link site (column "i, j far"), i.e., affinity propagates through the link to nearby points, creating a focus of positive affinity across clusters; and (b) decreases if i, j are both near the same link site (column "i, j close", within-cluster), possibly becoming slightly negative, note the negative lobe of the equivalent kernel. Thus, we get an affinity decrease around a must-link site and an affinity increase across must-linked sites. For cannot-links, the affinity always decreases. When i, j are close to a link site, the effect is similar to a must-link around a site; when i, j are close to different link sites, a focus of negative affinity arises that is propagated through the link. Fig. 1 shows the 3 types of equivalent kernels we can obtain: asymmetric with a negative lobe k'_{a}, symmetric positive k'_{s} and symmetric negative $-k'_{\mathrm{s}}$.

In summary, the effect of eq. (2) is that each link creates a wormhole between the link sites through which positive or negative affinity propagates to points near the sites, diffusing over a distance according to the manifold structure of the data. Importantly, a single link can have an effect of a *large* magnitude on *many* affinities. Must-links make points more similar even across clusters, while cannot-links make points more dissimilar even within a cluster. The new affinity can be described with an equivalent kernel derived from the original affinity kernel and the constraint type.

A limitation of our model Our way of calculating the affinity matrix $\overline{\mathbf{K}}$ has a limitation deeply related to its meaning as the covariance matrix of a Gaussian process. Generally, $\overline{\mathbf{K}}$ is a sensible measure of affinity only when there are two classes, as illustrated next. Suppose both $\mathbf{x}_i$ and $\mathbf{x}_j$ are cannot-linked to $\mathbf{x}_k$, then we have from **(C)** that $f(\mathbf{x}_i) \approx -f(\mathbf{x}_k) \approx f(\mathbf{x}_j)$, which is equivalent to putting a must-link between $\mathbf{x}_i$ and $\mathbf{x}_j$; hence, we will get $\overline{\mathbf{K}}_{ij}$ significantly greater than 0 even if $\mathbf{K}_{ij} \approx 0$. This interaction between cannot-links creates a **false affinity** between $\mathbf{x}_i$ and $\mathbf{x}_j$ when we have more than two classes. On the other hand, equation (2) is still conceptually correct for multiclass situations when there are no more than one cannot-link.

The simple one-versus-the-others strategy does not work

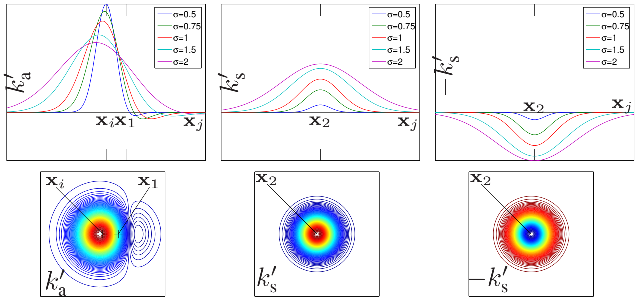


Figure 1. Graph of the equivalent kernel K'_{ij} assuming an originally Gaussian affinity kernel $K_{ij} = \exp\left(-\frac{1}{2}\left\|(\mathbf{x}_i - \mathbf{x}_j)/\sigma\right\|^2\right)$ with scale σ. *Above*: in 1D ($\mathbf{x}_i \in \mathbb{R}$) for several scales σ. The asymmetric kernel k'_{a} is a difference of offset Gaussians, while the symmetric $\pm k'_{\mathrm{s}}$ are positive/negative Gaussians. Compare k'_{a} with must-link $(1, 2$ far; i, j close) in table 1, k'_{s} with must-link $(1, 2$ far; i, j far) and $-k'_{\mathrm{s}}$ with cannot-link $(1, 2$ far; i, j far). *Below*: in 2D ($\mathbf{x}_i \in \mathbb{R}^2$) for $\sigma = 1$. All σ values are relative to $\|\mathbf{x}_i - \mathbf{x}_1\|$. Note for the 2D k'_{a} the right contours are negative.

here. One possible way to generalize the model to the multiclass situation is to consider more than one latent Gaussian process. However, we cannot capture the posterior distribution of all latent processes with only one $\overline{\mathbf{K}}$ as the posterior covariance. The Gaussian process classifier-based method of [4], after proper modification, is a potential candidate, although rather complicated. In this paper, we propose a simple modification of equation (2) to avoid the interaction between cannot-links, which we discuss below.

2. Constrained Clustering Algorithms

To do constrained clustering, we can run a standard clustering algorithm using the new, propagated affinities instead of the original ones $\mathbf{K}$ (Gaussian, heat kernel, etc.). In this paper we use spectral clustering, specifically the normalised cut [13]. However, we cannot use $\overline{\mathbf{K}}$ from (2) as affinity matrix directly, since some affinities are now negative (mostly due to cannot-links, since cannot-linked samples are forced to have opposite-sign affinities), and some samples can have a negative degree $\sum_j \overline{\mathbf{K}}_{ij}$. (A negative degree prevents computing a normalised graph Laplacian and means that splitting any node with negative degree from others has a negative cost in a graph cut and thus would be highly favoured by the algorithm.) We found that adding a constant bias to all $\overline{\mathbf{K}}_{ij}$ did not work well in practice. An approach that we found much more effective is to set to 0 all negative entries in $\overline{\mathbf{K}}$. Note we still take advantage of cannot-links since our algorithm will greatly decrease (to near-zero) some originally strong affinities.

Algorithm A (for two classes) Given $\mathbf{K} \succ 0$:

1. Compose the matrix $\mathbf{M}$ according to equation (3) based on all constraints and let $\overline{\mathbf{K}} = (\mathbf{K}^{-1} + \mathbf{M})^{-1}$.
2. Let $\mathbf{A}_{ij} = \max\,(0, \overline{\mathbf{K}}_{ij})\ \forall i, j$.
3. Do spectral clustering with $\mathbf{A}$ as the affinity matrix.

Algorithm B described below is a generalization of Algorithm A for multiclass situations. The basic idea is to avoid the interaction (false affinity) between cannot-links by enforcing them separately and getting many versions of $\overline{\mathbf{K}}$. Since the main effect of cannot-links is to weaken the affinities between some samples, for any entry between $\mathbf{x}_i$ and $\mathbf{x}_j$ we always use the smallest (most weakened) among all the different versions.

Algorithm B (for more than two classes) Given $\mathbf{K} \succ 0$:

1a. Compose the matrix $\mathbf{M}^m$ according to equation (3) based on only must-links and let $\mathbf{K}^m = (\mathbf{K}^{-1} + \mathbf{M}^m)^{-1}$.

1b. Suppose we have n_c cannot-links. Compose the matrix $\mathbf{M}^{c,k}, k = \{1, 2, \dots, n_c\}$ according to equation (3) based on the i^{th} cannot-link and let $\mathbf{K}^{c,k} = ((\mathbf{K}^m)^{-1} + \mathbf{M}^{c,k})^{-1}$.

2. Let $\mathbf{A}_{ij} = \max\left(0, \min\left(\mathbf{K}^{c,1}_{ij}, \dots, \mathbf{K}^{c,n_c}_{ij}\right)\right)\ \forall i, j$.

3. Do spectral clustering with $\mathbf{A}$ as the affinity matrix.

Illustrative examples We provide three 1D examples to demonstrate the mechanics of our algorithm. In all them we use the Gaussian kernel for the original affinity $\mathbf{K}$. Figure 2 shows how one must-link merges two clusters into one. Row **A** shows a 3–cluster dataset and one must-link across two clusters. As shown in row **B**, the affinity matrix $\mathbf{K}$ and the eigenvalues and eigenvectors of the normalised graph Laplacian with $\mathbf{K}$ as the affinity matrix suggest three clusters. After enforcing the must-link with Algorithm A, we get the affinity matrix $\overline{\mathbf{K}}$ in row **C** that clearly shows many new affinities between the first two clusters. Those affinities are centred at the entry corresponding to the must-linked pair and diffuse to other entries isotropically, as predicted by the equivalent kernel. Equally salient are the changes of the eigenvalues and eigenvectors in row **C**. The second eigenvector suggests to group the first two clusters into one and keep the third cluster separate (contrary to the distribution of the data but consistent with the must-link constraint).

Figure 3 illustrates how one cannot-link breaks a continuum into two clusters. As one can see by comparing the affinity matrix before and after the constraint, the cannot-link (enforced with Algorithm A) greatly weakens the affinities between data around the cannot-linked samples. As a result, the eigenvalues and eigenvectors suggest a two-cluster structure with the boundary between the two cannot-linked samples. Fig. 4 shows how algorithm B avoids false affinities created by multiple cannot-links.

3. Experiments

Artificial and real-world data (fig. 5–6) We compared our algorithm with constrained K-means (CKmeans) [15], constrained Gaussian mixture model [11], and a preliminary implementation of constrained spectral clustering (KKM) [10] on a variety of data sets with varying numbers of pairwise constraints. For all clustering algorithms, the number of clusters is always set as the number of classes, and we use hard constraints ($\epsilon = 10^{-5}$). We used Algorithm A when there are only two classes and Algorithm B otherwise. For the four artificial data sets (fig. 5) and three UCI data sets (fig. 6 left) we used Gaussian affinities $\mathbf{K}$ of suitable width, while for the 20–newsgroups (fig. 6 right) we used a heat diffusion kernel $\mathbf{K} = \exp(-20\Delta)$, where Δ is the normalized graph Laplacian on a 10–nearest-neighbour graph (to cope with the highly sparse distribution of the feature vectors). The three clustering tasks on the 20–newsgroups are chosen to represent different levels of difficulty. For all 10 clustering tasks, the clustering accuracy is measured with the Rand index and the reported clustering accuracy is averaged over 100 random realisations of pairwise constraints. Our algorithm is superior to competing methods in that it can achieve a considerable improvement of clustering accuracy with a relatively small number of pairwise constraints.

Semi-supervised image segmentation (fig. 7) The user is

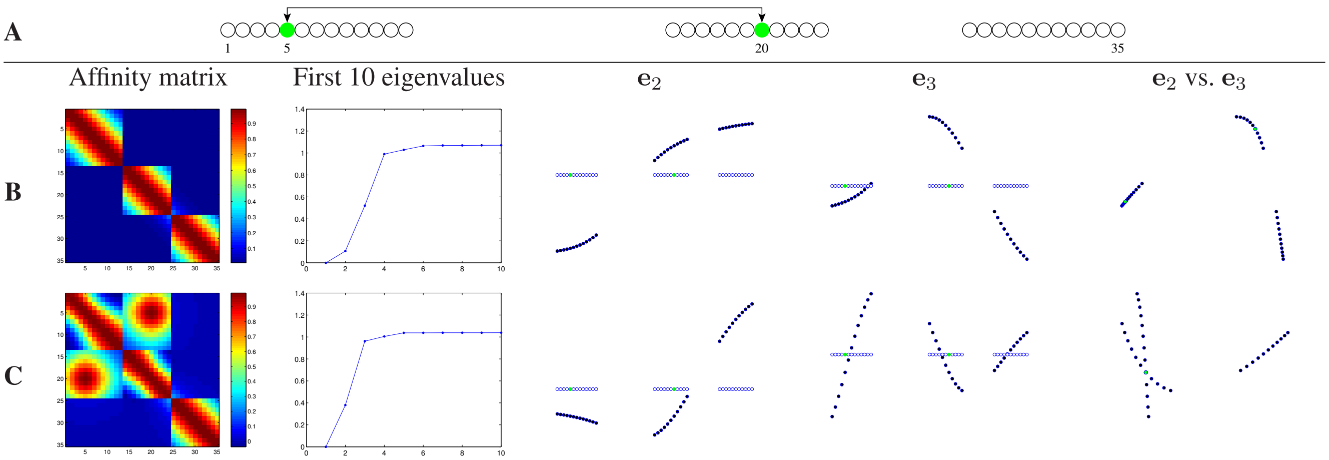


Figure 2. One must-link merges two clusters. **A**: 1D data plotted as circles (the small numbers represent indices, not coordinates), and one must-link between the 5th and 20th samples. **B**: affinity matrix $\mathbf{K}$ using a Gaussian kernel, associated eigenvalues and eigenvectors $\mathbf{e}_2$, $\mathbf{e}_3$ of the normalised graph Laplacian (nearly piecewise constant over the clusters; $\mathbf{e}_1 = \mathbf{1}$ not shown); the eigenspace $(\mathbf{e}_2, \mathbf{e}_3)$ shows 3 clusters. **C**: the new affinity matrix after incorporating the must-link using Algorithm A; the eigenspace $(\mathbf{e}_2, \mathbf{e}_3)$ shows 2 clusters.

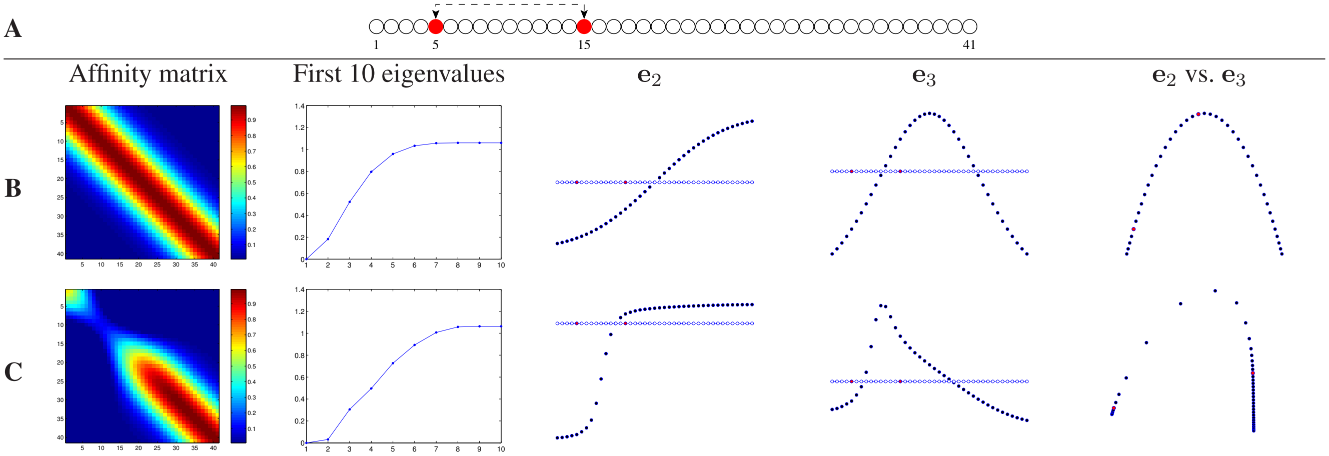


Figure 3. One cannot-link splits data into two clusters. **A**: 1D continuous dataset plotted as circles, and one cannot-link between the 5th and 15th samples. **B**, **C**: as in fig. 2, before and after incorporating the cannot-link. $\mathbf{e}_2$ changes from continuous to almost piecewise constant with a split around the 10th sample; the eigenspace $(\mathbf{e}_2, \mathbf{e}_3)$ changes from one continuous cluster to two clusters.

shown an image and asked to specify several pairwise constraints to guide the segmentation. As original affinities we use the Gaussian kernel with one feature vector $\mathbf{x}_i \in \mathbb{R}^3$ per pixel consisting of location and intensity. Fig. 7**A**–**C** consist of a 16×16 image of an occluder to be segmented from an irregular background [3]; fig. 7**D** is a 43×43 noisy image with 3 objects on a background [7]. All cases are difficult for unsupervised spectral clustering because the object boundaries are ill defined and contain smooth intensity gradients. The results show the segmented image and the leading eigenvalues and eigenvectors $(\mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4)$ of the normalised graph Laplacian, without constraints (upper row) and with constraints (lower row). Adding a few constraints (e.g. across an ill-defined boundary) helps to separate the objects; note the improvement in the eigenvectors.

4. Related Models

Besides the connection to the metric learning model of [5], our model is related to several other semi-supervised learning models. We have compared our model empirically with the preliminary constrained clustering method in [10], a variant of which has been proposed in [17] with a slightly different treatment of eigenvectors. Hoi et al. [9] learn a nonparametric kernel matrix from the pairwise constraints by incorporating the original affinity through a graph-Laplacian regulariser; consequently the learned kernel does not naturally degenerate to the original one even

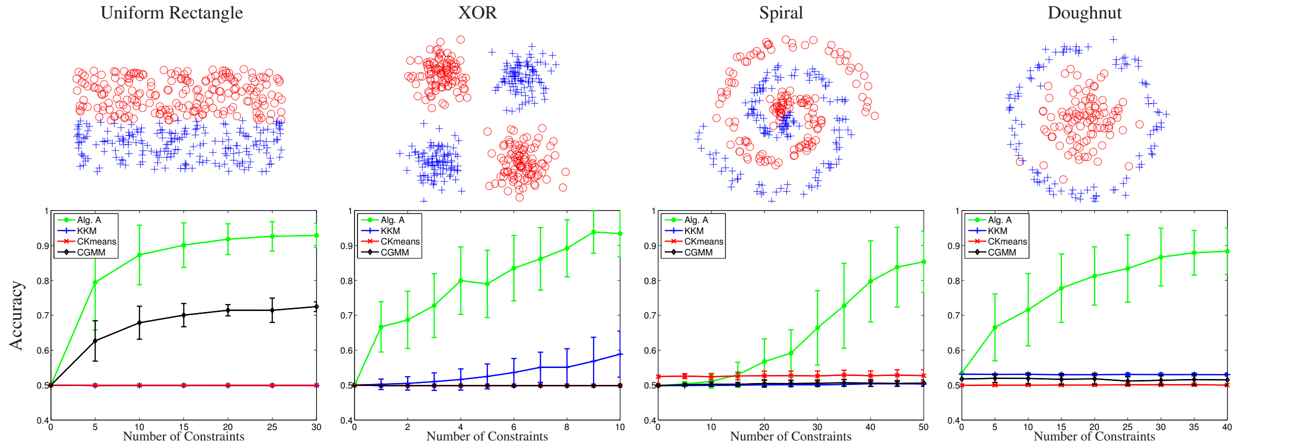


Figure 5. Synthetic datasets (each one has 2 classes and 200 points in each class): with our method, the clustering accuracy increases very quickly with the number of pairwise constraints. Errorbars over 100 random realisations of pairwise constraints.

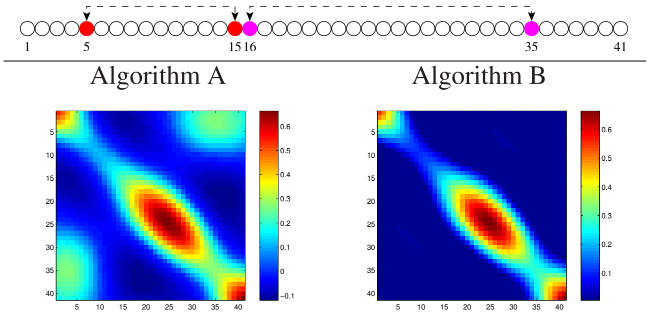


Figure 4. False affinity generated by multiple cannot-links. The data and the first cannot-link is the same as in figure 3. The second cannot-link is between the 16th and 35th samples. Note the false affinity generated between the 5th and 35th samples when using Algorithm A to enforce the two cannot-links, and compare it to the affinity matrix generated using Algorithm B.

if no constraints are specified. In [8] and [14], the idea of distorting the RKHS space is close to our way of modifying the affinity matrix. However, their modification of kernel entries is mainly based on incorporating a graph prior, and in [8] the pairwise relations are used as an aid to the labeled samples in a conventional semi-supervised learning scenario. Most constrained image segmentation approaches [11, 7, 6] enforce the constraints on a instance level, and thus often make inefficient use of the constraints. The constrained image segmentation algorithm of [18] can also be viewed as a kind of affinity propagation that is implemented by forcing a constrained pixel to be in the same cluster as its neighbors in a vicinity specified by user; thus, the affinity propagation is controlled by the user in a rather ad hoc way and does not naturally generalize to non-image data.

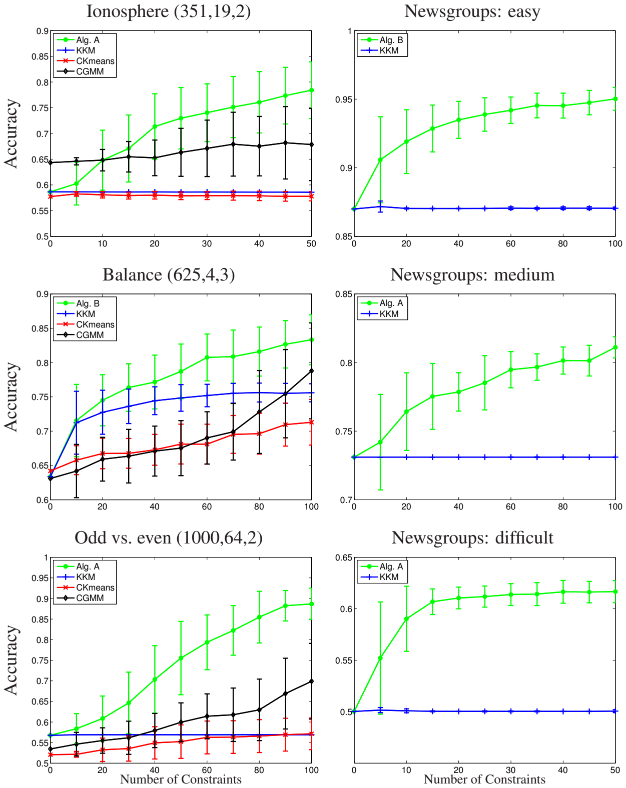


Figure 6. Results on three UCI datasets (left column) (points, features, classes) and three 20–newsgroup tasks (right column), all datasets with 1 000 samples evenly distributed in the classes. Easy: `alt.atheism, rec.sport.baseball,sci.space`; Medium: `Windows, Macintosh`; Difficult: `talk.politics.misc, talk.politics.guns, talk.politics.mideast`.

5. Conclusion

Our method proposes a natural way (based on a Gaussian process formulation) to propagate affinity information through pairwise constraints; the latter act as wormholes that connect space regions that are faraway (low affinity) for must-links, or disconnect nearby regions for cannot-links. The new affinity matrix has a closed-form expression (eq. 2–3) that can be obtained by inverting a small matrix, at a neglibible overhead over spectral clustering. This new affinity can be represented by a new kernel function derived from the original one. Experimentally, our method needs very few constraints to achieve good clusterings as compared with other methods. Two areas of further research are: (1) how to profit more effectively from the negative affinities generated by the method, and (2) a more natural extension to the multiclass case.

A. Minimum divergence formulation

We give an interpretation of our affinity propagation model based on the min-divergence principle. Again, suppose the original affinity matrix $\mathbf{K} \succ 0$ is the covariance matrix of a zero-mean Gaussian distribution $P(\mathbf{f})$, eq. (1). Now we want to find a probability distribution $\overline{P}(\mathbf{f})$ that is closest to $P(\mathbf{f})$ while satisfying must-link (α) and cannot-link (β) constraints. This is the variational problem:

$$\begin{aligned} \min_{\overline{P}} \quad & \mathrm{D}\left(\overline{P}(\mathbf{f}) \| P(\mathbf{f})\right) \\ \text{s.t.} \quad & \mathrm{E}_{\overline{P}}\left\{(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2\right\} \le \alpha_{ij}, \quad (i,j) \in \mathcal{M} \\ & \mathrm{E}_{\overline{P}}\left\{(f(\mathbf{x}_i) + f(\mathbf{x}_j))^2\right\} \le \beta_{ij}, \quad (i,j) \in \mathcal{C} \end{aligned}$$

where $\mathrm{E}_{\overline{P}}\{\cdot\} = \iint (\cdot) \overline{P}(f(\mathbf{x}_i), f(\mathbf{x}_j))\, df(\mathbf{x}_i)\, df(\mathbf{x}_j)$ and $\mathrm{D}\left(p \| q\right) = \int p \log\left(p/q\right)$. Applying calculus of variations we find that the optimal $\overline{P}(\mathbf{f})$ takes the following form:

$$\overline{P}(\mathbf{f}) \propto P(\mathbf{f}) \times \exp\Big(- \sum_{(i,j)\in\mathcal{M}} \lambda_{ij} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 - \sum_{(i,j)\in\mathcal{C}} \lambda_{ij} (f(\mathbf{x}_i) + f(\mathbf{x}_j))^2 \Big), \quad (9)$$

with Lagrange multipliers $\lambda_{ij} \ge 0$ (Karush-Kuhn-Tucker conditions). Thus $\overline{P}(\mathbf{f})$ is also Gaussian, with covariance matrix $\overline{\mathbf{K}} = (\mathbf{K}^{-1} + \mathbf{M})^{-1}$, where $\mathbf{M} \in \mathbb{R}^{N \times N}$

$$\mathbf{M}_{ij} = \begin{cases} 2 \sum_{k:\ (i,k)\in\mathcal{M}\cup\mathcal{C}} \lambda_{ik}, & i = j \\ -2\lambda_{ij}, & (i,j) \in \mathcal{M} \\ 2\lambda_{ij}, & (i,j) \in \mathcal{C} \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

has the same form as (3). This means that, given the constraint sets $\mathcal{M}$ and $\mathcal{C}$, we can derive the new affinity matrix $\overline{\mathbf{K}}$ either by defining the constraints softly in terms of “scales” ϵ_m, ϵ_c and applying eqs. (2)–(3) (which is easy); or by defining the constraints as hard bounds α, β and solving the optimisation problem for λ_{ij} and so for $\mathbf{M}$ and $\overline{\mathbf{K}}$ (which is computationally difficult).

Since the previous variational problem is optimised by a Gaussian, we could write an optimisation problem directly over the covariance matrix $\overline{\mathbf{K}}$ of $\overline{P}$ as follows:

$$\begin{aligned} \min_{\overline{\mathbf{K}}} \quad & \log\left(|\mathbf{K}| \,/\, |\overline{\mathbf{K}}|\right) + \mathrm{tr}\left(\mathbf{K}^{-1}\overline{\mathbf{K}}\right) \\ \text{s.t.} \quad & \overline{\mathbf{K}} \succ 0 \\ & \overline{\mathbf{K}}_{ii} + \overline{\mathbf{K}}_{jj} - 2\overline{\mathbf{K}}_{ij} \le \alpha_{ij}, \quad (i,j) \in \mathcal{M} \\ & \overline{\mathbf{K}}_{ii} + \overline{\mathbf{K}}_{jj} + 2\overline{\mathbf{K}}_{ij} \le \beta_{ij}, \quad (i,j) \in \mathcal{C}. \end{aligned}$$

We lately learned (I. Dhillon, pers. comm.) that this is similar to the information-theoretic metric learning of [5], which tries to find a Mahalanobis distance that is closest to the Euclidean distance while satisfying all the pairwise constraints expressed as a distance between constrained pairs. This shows an interesting connection between our Bayesian perspective (using Gaussian processes) and the perspective of metric learning by divergence minimisation in a Hilbert space [5]. Also, the approach in [5] expresses cannot-links differently from us and lacks a closed-form solution (unlike our (2)), so numerical optimisation is required.

References

[1] S. Basu, M. Bilenko, and R. Mooney. A probabilistic framework for semi-supervised clustering. In *KDD*, 2004.

[2] H. Chang and D. Yeung. Semi-supervised metric learning by kernel matrix adaptation. In *Int. Conf. Machine Learning and Cybernetics (ICMLC'05)*, 2005.

[3] C. Chennubhotla and A. Jepson. EigenCuts: Half lives of eigenflows for spectral clustering. In *NIPS*, 2003.

[4] W. Chu, V. Sindhwani, Z. Ghahramani, and S. Keerthi. Relational learning with Gaussian processes. In *NIPS*, 2007.

[5] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. In *ICML*, 2006.

[6] A. Eriksson, C. Olsson, and F. Kahl. Normalized cuts revisited: A reformulation for segmentation with linear grouping constraints. In *ICCV*, 2007.

[7] M. Figueiredo, D. Seon, and V. Murino. Clustering under prior knowledge with application to image segmentation. In *NIPS*, 2007.

[8] A. Goldberg, X. Zhu, and S. Wright. Dissimilarity in graph-based semi-supervised classification. In *AISTATS*, 2007.

[9] S. Hoi, R. Jin, and M. Lyu. Learning nonparametric kernel matrices from pairwise constraints. In *ICML*, 2007.

[10] S. Kamvar, D. Klein, and C. Manning. Spectral learning. In *IJCAI*, 2003.

[11] Z. Lu and T. Leen. Penalized probabilistic clustering. *Neural Computation*, 19, 2007.

[12] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing Gaussian mixture models with EM using equivalence constraints. In *NIPS*, 2003.

[13] J. Shi and J. Malik. Normalized cut and image segmentation. *IEEE Trans. PAMI*, 22(8), 2000.

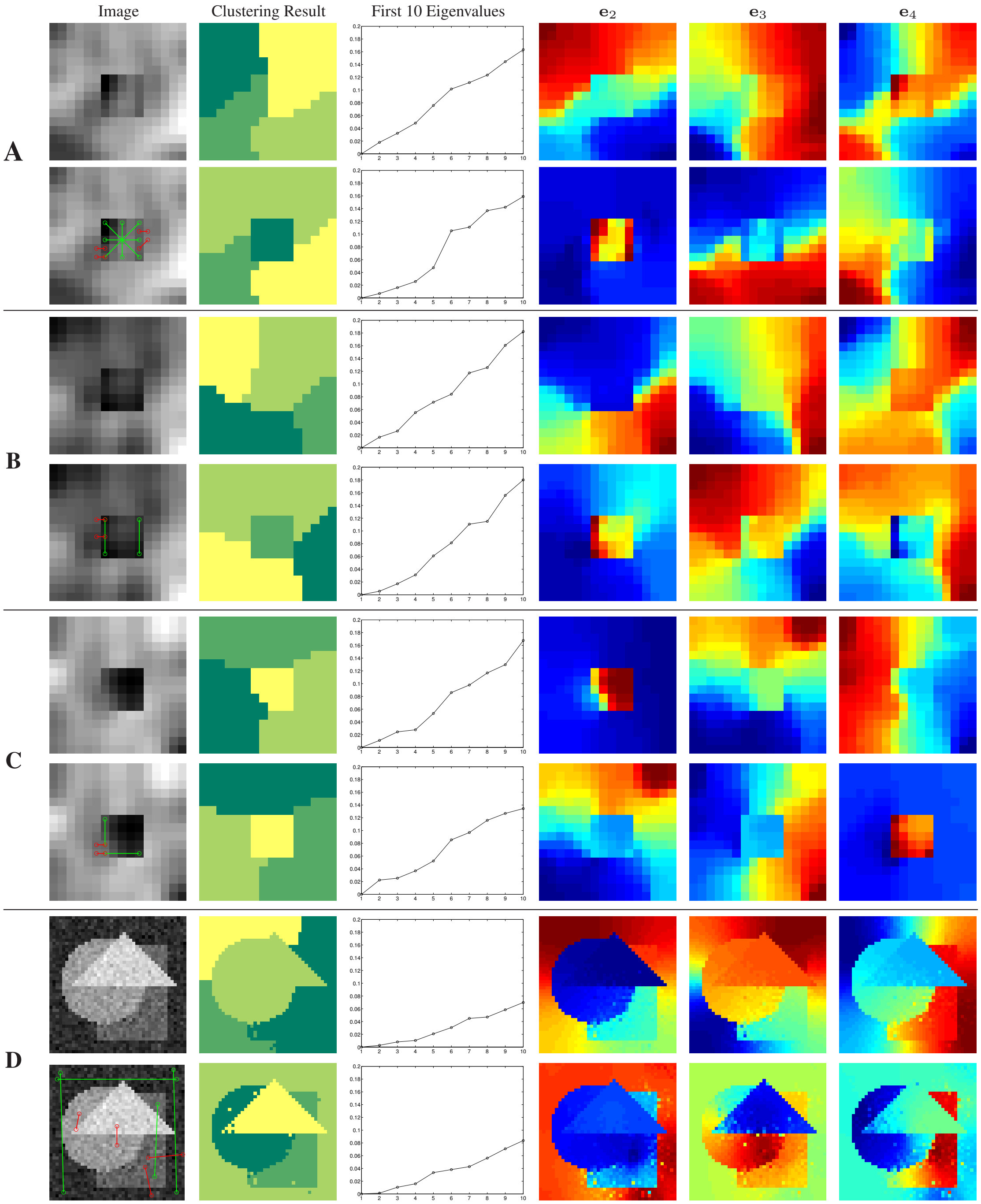


Figure 7. Image segmentation (in all 4 images **A**–**B**, the number of clusters in spectral clustering is set to 4). The must-links (cannot-links) are visualised as the green (red) lines connecting pixel pairs. Upper/lower row: results without/with constraints.

[14] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: From transductive to semi-supervised learning. In *ICML*, 2005.

[15] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained K-means clustering with background knowledge. In *ICML*, 2001.

[16] E. Xing, A. Ng, M. Jordan, and S. Russe. Distance metric learning with applications to clustering with side information. In *NIPS*, 2003.

[17] Q. Xu, M. desJardins, and K. Wagstaff. Active constrained clustering by examining spectral eigenvectors. In *Proc. 8th Int. Conf. on Discovery Science*, 2005.

[18] X. Yu and J. Shi. Grouping with bias. In *NIPS*, 2001.