



Published in final edited form as:

IEEE Int Workshop Mach Learn Signal Process. 2012 December 31; 2013: 1–6. doi:10.1109/MLSP.

2013.04.07.65

CONSTRAINED SPECTRAL CLUSTERING FOR IMAGE SEGMENTATION

Jamshid Sourati, Dana H. Brooks, Jennifer G. Dy, and Deniz Erdogmus

Electrical and Computer Engineering Department, Northeastern University, Boston, MA

Abstract

Constrained spectral clustering with affinity propagation in its original form is not practical for large scale problems like image segmentation. In this paper we employ novelty selection sub-sampling strategy, besides using efficient numerical eigen-decomposition methods to make this algorithm work efficiently for images. In addition, entropy-based active learning is also employed to select the queries posed to the user more wisely in an interactive image segmentation framework. We evaluate the algorithm on general and medical images to show that the segmentation results will improve using constrained clustering even if one works with a subset of pixels. Furthermore, this happens more efficiently when pixels to be labeled are selected actively.

Index Terms

Constrained spectral clustering; active learning; image segmentation

1. INTRODUCTION

Data clustering and image segmentation are two challenging problems in machine learning and computer vision. In many image segmentation applications creating supervisory labels is very time-consuming and labels might not even be accurate. At the same time unsupervised algorithms may not yield sufficient accuracy. In the last decade, many semi-supervised algorithms have been proposed where limited amount of supervisory information is used to efficiently guess the full label set.

In image segmentation, there are two different ways of interacting with clustering algorithms: either by specifying labels of a small subset of pixels [1, 2], or by putting equivalence constraints [3, 4, 5] in which instead of determining class of pixels explicitly, relationships between them are given. The way that these information are taken into account also varies from using them to formulate constrained optimization problems [6, 4] to make posterior distributions in probabilistic frameworks [5]. In the latter scenario, there should be defined a model to be used as the likelihood of the data given the constraints.

In interactive image segmentation, besides developing semi-supervised versions of computer vision methods, such as contour-based algorithms [7], machine learning strategies have been also used widely. One of the most popular family of methods in this area is graph-cut based clustering [6, 4]. The main difficulty with such methods is that creating graphs and processing them is very time- and memory-consuming for large datasets (like images). To cope with this problem, the number of available data points (pixels) is reduced either by means of over-segmentation to super-pixels using strategies like Mean-shift [4], or by down-sampling the image [8].

In this paper we use a semi-supervised type of *spectral clustering* method which uses *probabilistic* models to involve the user's *pairwise* constraints to guide a *binary*

segmentation. As Kumar and Kummamuru discussed [3], pairwise constraints are not suitable for multi-class clustering.

This paper is mainly based on the work done by Lu and Carreira-Perpiñán [5]; applying their graph-based algorithm to image segmentation is not straightforward because of the aforementioned issues for large datasets. Besides there are some heuristics used in their work which make the algorithm fail at some points. Here, we use sub-sampling strategy for data reduction in addition to a sophisticated algebraic tool for speeding up the eigen-decomposition process in order to make the algorithm work for large images. This approach will also be combined with an entropy-based active learning to pick out the queries more wisely than random selection. Finally, we analyze the cases where the heuristics make the algorithm fail to enforce the constraints properly. Note that the goal here is not to give a perfect clustering solution, but to improve it as much as possible by getting information from the user.

2. SEMI-SUPERVISED IMAGE SEGMENTATION

In this section we explain how interactive spectral clustering with affinity propagation works for binary segmentation. Then an existing sampling strategy that is used for data reduction is explained. Finally active query selection strategy is demonstrated briefly.

2.1. Constrained Spectral Clustering

Let $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]_{d \times N}$ be the matrix containing feature vectors of all data points. In graph-based clustering methods, the data is represented by an undirected weighted graph. The goal is to bipartition the constructed graph into two clusters C_1 and C_2 such that the total weights of the *cut* edges is minimum while avoiding too small clusters by balancing the volumes. In this paper, \mathbf{K} is the adjacency (similarity) matrix and \mathbf{D} is the corresponding diagonal degree matrix where $d_{ii} = \sum_{j=1}^N k_{ij}$. Let us first define \mathbf{f} as the indicator vector:

$$f_i = \begin{cases} 1 & , \mathbf{x}_i \in C_1 \\ -r_v & , \mathbf{x}_i \in C_2 \end{cases} \quad (1)$$

where $r_v = (\sum_{\mathbf{x}_i \in C_1} d_{ii}) / (\sum_{\mathbf{x}_i \in C_2} d_{ii})$. Then normalized graph-cut problem can be formalized by the following optimization:

$$\mathbf{f} = \arg \min_{\mathbf{D}\mathbf{f} \perp \mathbf{1}_N} \frac{\mathbf{f}^T (\mathbf{D} - \mathbf{K}) \mathbf{f}}{\mathbf{f}^T \mathbf{D} \mathbf{f}}, \quad (2)$$

where $\mathbf{1}_N$ is the all-ones vector with N entries. The discrete optimization problem in (2) can be shown to be NP-hard [9]. Spectral clustering gives a solution by relaxing the discreteness of the characteristic function \mathbf{f} [9]. Constrained spectral clustering with affinity propagation is based on the main assumption that \mathbf{f} is realization of a Gaussian Process such as:

$$p(\mathbf{f}) = \frac{1}{(2\pi|\mathbf{K}_0|)^{N/2}} e^{-\frac{1}{2}\mathbf{f}^T \mathbf{K}_0^{-1} \mathbf{f}}, \quad (3)$$

where the covariance matrix of this process is considered to be equal to the unconstrained adjacency matrix \mathbf{K}_0 . Note that this formulation is very similar to Gaussian random field model [1, 10] except that the covariance matrix in that model is set to be the unnormalized Laplacian of the data graph $\mathbf{\Delta} = \mathbf{D} - \mathbf{K}_0$. However this cannot be employed directly when constraints are in the form of pairwise links.

Must- or cannot-links between two points \mathbf{x}_i and \mathbf{x}_j make their corresponding soft labels closer or further by putting a zero-mean Gaussian distribution on their difference or summation respectively ($f_i \pm f_j \sim \mathcal{N}(0, \varepsilon^2)$). The variance ε^2 can also be different for each case.

Denoting the set of must-link by \mathcal{M} and cannot-links by \mathcal{C} , the posterior distribution can be obtained by simplifying the multiplication of two Gaussian distributions:

$$p(\mathbf{f}|\mathcal{M} \cup \mathcal{C}) = p(\mathbf{f})p(\mathcal{M} \cup \mathcal{C}|\mathbf{f}) \sim e^{-\frac{1}{2}\mathbf{f}^T(\mathbf{K}_0^{-1} + \mathbf{M})\mathbf{f}}, \quad (4)$$

where \mathbf{M} is a sparse matrix (for details see [5]). The matrix $\mathbf{K}_0^{-1} + \mathbf{M}$ is inverse of the updated affinity matrix and can be used to do constrained spectral clustering. More details of this algorithm are discussed in section 3.1.

2.2. Novelty Selection

There are several sampling techniques that we can use. Here, as in [8], we use *novelty selection* method where the points are sampled such that there is least amount of information redundancy between them; more specifically we pick the first pixel, then go through the rest one by one and add a pixel if the distance between its feature vector to the closest point already selected, is less than a threshold δ_s . The constrained clustering algorithm can be run on a graph constructed over this subset of points as a sparse-labeling step; the full-labeling process can be performed simply by using nearest neighbors (NN) or Kernel Density Estimation (KDE); here, for the sake of simplicity, we've used the former one. In the following sections, we denote S as the set of selected samples for which the number of entries $|S| = N_s(\delta_s)$ depends on the threshold δ_s .

Figure 1 shows a case where we sampled an image using a threshold of $\delta_s = 0.2$; this gave us a subset of pixels which has less than one percent (0.8%) of the whole number of pixels (figure 1b). In this example we used, as features, only the spatial and intensity information of the samples. The first row of the figure shows the result of the unconstrained spectral clustering and the rest illustrates how putting few constraints on the samples can lead to desired clusters although we're working only on a small portion of dataset. In the second row the user tried to cluster the man from his background (7 must-links plus 14 cannot-links) while in the third row his hat is the only targeted object (no must-links and 11 cannot-links); as can be seen in the first case since the heterogeneity of the desired object is higher and more similar to background we needed more constraints.

2.3. Laplacian Eigendecomposition

Besides sub-sampling the image, to further ease the task of spectral clustering, we take advantage of the iterative invariant-subspace estimation techniques. For estimating the first two eigenvectors of the Laplacian matrix we have used subspace (orthogonal) iterations [11] which is a generalization of the power method. It starts from an orthonormal initial set of vectors and at each iteration they're multiplied by the matrix followed by an orthonormalization process (e.g. QR factorization) which is required to preserve linear independence between the vectors. However, because orthogonalization is expensive for large matrices, it's preferable to perform it after each fix number of iterations; this corresponds to *multi-step* subspace iteration that we have used here [12]. Empirically we set the intervals between consecutive factorizations equal to 200 iterations. More details about this iterative method is discussed in section 3.2.

2.4. Choosing Constraints Actively

Choosing a point from the sample pool to put constraints on, is not trivial and has been studied widely under the title of *active learning*. All such algorithms select queries such that putting constraints on them has the largest effect and therefore the user needs to have minimum interaction to get the desired result. One way of doing this is *uncertainty sampling* [13], which chooses the query as the point with posterior distribution of highest entropy, hence least certainty for the algorithm. Let y be the estimated class label for the feature vector \mathbf{x} . Then Shannon entropy of the posterior is given by:

$$H(y|\mathbf{x}) = -\sum_y p(y|\mathbf{x}) \log p(y|\mathbf{x}), \quad (5)$$

We consider binary segmentation so y has only two possible values (say 1 and 2); the posterior for each class, at a pixel $\mathbf{x} \in S$, is proportional to the multiplication of likelihoods computed using KDE and the empirical class priors:

$$P(y|\mathbf{x}) \propto p(\mathbf{x}|y)P(y) = \begin{cases} \left(\frac{1}{N_1+N_2}\right) \sum_{\mathbf{x}_k \in C_1} K(\mathbf{x}, \mathbf{x}_k) & , y=1 \\ \left(\frac{1}{N_1+N_2}\right) \sum_{\mathbf{x}_k \in C_2} K(\mathbf{x}, \mathbf{x}_k) & , y=2 \end{cases} \quad (6)$$

where N_i is the number of samples in the C_i cluster and K can be extracted from the affinity matrix. Our goal is to pick out a sample that the algorithm has the least confidence about its label; we use the absolute value of the log-posteriors as a criterion for the query selection:

$$\begin{aligned} \mathbf{x}_q &= \arg \max_{\mathbf{x} \in S} H(y|\mathbf{x}) \\ &= \arg \min_{\mathbf{x} \in S} \left| \log \left[\frac{P(y=1|\mathbf{x})}{P(y=2|\mathbf{x})} \right] \right|. \end{aligned} \quad (7)$$

3. IMPLEMENTATION DETAILS

Efficient implementation of different blocks in previous sections needs more detailed analysis which is done in this section. It should be noted that we use the same heuristic of making all the negative elements in the resulted new affinity matrix equal to zero.

3.1. Computing Constrained Affinity Matrix

The set of all constraints Ω can be broken into several individual links $\Omega = \bigcup_{t=1}^T \Omega_t$ where each Ω_t is either a must- or a cannot-link; keeping in mind that we can apply constraints sequentially we'll focus only on t 'th constraint:

$$\mathbf{K}_t = (\mathbf{K}_{t-1}^{-1} + \mathbf{M}_t)^{-1} = \left(\mathbf{K}_0^{-1} + \sum_{i=1}^t \mathbf{M}_i \right)^{-1}. \quad (8)$$

Each constraint matrix \mathbf{M}_t can be written in a permuted form:

$$\mathbf{M}_{t_p} := \mathbf{P}_t^T \mathbf{M}_t \mathbf{P}_t = \begin{bmatrix} (\mathbf{\Gamma}_t)_{2 \times 2} & 0 \\ 0 & 0 \end{bmatrix}_{N_s + N_s}$$

where Γ_t has the form $\frac{1}{\varepsilon^2} \begin{bmatrix} 1 & \pm 1 \\ \pm 1 & 1 \end{bmatrix}$ (with negative for must-link and positive for cannot-link constraints). Using Wood-bury inversion identity, (8) can be rewritten as $\mathbf{K}_t = \mathbf{K}_{t-1} - \mathbf{K}_{t-1}(\mathbf{I} - \mathbf{M}_t \mathbf{K}_{t-1})^{-1} \mathbf{M}_t \mathbf{K}_{t-1}$ where \mathbf{I} is the identity matrix. Since using permutation leads to considerable simplifications, similar to \mathbf{M}_{t_p} we also define $\mathbf{K}_{t-1_p} =: \mathbf{P}_t^T \mathbf{K}_{t-1} \mathbf{P}_t$, then it can be easily shown that

$$\mathbf{K}_{t_p} = \mathbf{K}_{t-1_p} - \mathbf{K}_{t-1_p} (\mathbf{I} - \mathbf{M}_{t_p} \mathbf{K}_{t-1_p})^{-1} \mathbf{M}_{t_p} \mathbf{K}_{t-1_p} \quad (9)$$

$$= \mathbf{P}^T \mathbf{K}_t \mathbf{P}.$$

Therefore in order to compute the next constrained affinity matrix, we can use permutations of \mathbf{K}_{t-1} and \mathbf{M}_t and then permute back to original indices to get the result. It's better to do so because deriving the permuted update is cleaner: let us first divide \mathbf{K}_{t-1_p} into four blocks as below:

$$\begin{bmatrix} & 2 & N_s - 2 \\ 2 & \mathbf{K}_{11} & \mathbf{K}_{12} \\ N_s - 2 & \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix}.$$

Then after some matrix manipulations, the second term in (9) can be computed by the following matrix products:

$$\mathbf{K}_{t-1_p} - \mathbf{K}_{t_p} = \begin{bmatrix} \mathbf{K}_{11} \\ \mathbf{K}_{12} \end{bmatrix} (\Gamma_t \mathbf{K}_{11} + \mathbf{I})^{-1} \Gamma_t \begin{bmatrix} \mathbf{K}_{11}^T & \mathbf{K}_{12}^T \end{bmatrix}.$$

Note that the inversion appeared in above equation is over a 2×2 matrix and therefore very easy to compute. Doing further matrix algebra, eventually we can write each column of the new affinity matrix as the linear combination of the same column and the constrained columns in the old matrix; more particularly assuming that the t 'th constraint is between points \mathbf{x}_i and \mathbf{x}_j both in S , one can verify for $k = 1, \dots, N_s$:

Must-link:

$$\mathbf{K}_{t,k} = \mathbf{K}_{t-1,k} - \left(\frac{k_{t-1,ik} - k_{t-1,jk}}{\alpha} \right) (\mathbf{K}_{t-1,i} - \mathbf{K}_{t-1,j}) \quad (10a)$$

Cannot-link:

$$\mathbf{K}_{t,k} = \mathbf{K}_{t-1,k} - \left(\frac{k_{t-1,ik} - k_{t-1,jk}}{\beta} \right) (\mathbf{K}_{t-1,i} - \mathbf{K}_{t-1,j}) \quad (10b)$$

where $\mathbf{K}_{t,k}$ is the k 'th column of \mathbf{K}_t and:

$$\begin{aligned} \alpha &= k_{t-1,ii} + k_{t-1,jj} - 2k_{t-1,ij} + \varepsilon^2, \\ \beta &= k_{t-1,ii} + k_{t-1,jj} + 2k_{t-1,ij} + \varepsilon^2. \end{aligned} \quad (11)$$

From equations (11), (10a) and (10b) we can observe that as ε gets larger, the constraint becomes less effective. As in the original paper [5], we put ε close to zero in order to have

hard constraints; however, it should be noted that it cannot get smaller than the squared root of the machine's precision (epsilon). Also it is clear that points which are far from \mathbf{x}_i and \mathbf{x}_j won't take any effects.

Another observation is that the diagonal terms corresponding to the affected points shrink as we get more and more constraints. On the other hand, a zero on-diagonal term might cause the whole column to shrink to zero. So in our implementation, we normalize columns and rows of the new affinity matrix each time a new constraint is arrived:

$$\text{normalized } \mathbf{K}_t = \mathbf{\Lambda}_t \mathbf{K}_t \mathbf{\Lambda}_t, \quad (12)$$

where $\mathbf{\Lambda}_t$ is a diagonal matrix and $\lambda_{ii} = \frac{1}{\sqrt{k_t(i,i)}}$.

Besides the gradual shrinkage, on-diagonal terms might become zero at once before normalization; this happens if one of the following cases take place for a point \mathbf{x}_k , $k = 1, \dots, N_s$:

$$\begin{aligned} \alpha &\leq [k_{t-1,ik} - k_{t-1,ij}]^2 \\ \beta &\leq [k_{t-1,ik} + k_{t-1,ij}]^2. \end{aligned} \quad (13)$$

If either conditions in (13) occurs the corresponding diagonal term becomes non-positive, therefore such constraints should be avoided.

3.2. Large Matrix Eigendecomposition

Suppose $\hat{\mathbf{v}}$ is an estimated eigenvector of a matrix \mathbf{A} and $\hat{\lambda}$ the corresponding estimated eigenvalue from the Rayleigh quotient:

$$\hat{\lambda} = \frac{\hat{\mathbf{v}}^T \mathbf{A} \hat{\mathbf{v}}}{\hat{\mathbf{v}}^T \hat{\mathbf{v}}},$$

then residual vector of the estimation is defined by $\mathbf{r} = \mathbf{A} \hat{\mathbf{v}} - \hat{\lambda} \hat{\mathbf{v}}$ satisfying the following inequality for a symmetric \mathbf{A} [12]:

$$|\lambda - \hat{\lambda}| \leq \|\mathbf{r}\|_2 \quad (14)$$

So \mathbf{r} is usually used as the stopping criterion in iterative eigen-decomposition methods. Here, we compute a matrix containing the residual vectors \mathbf{r}_1 and \mathbf{r}_2 for the estimated first two eigenvectors respectively; we monitor norm-2 of these two vectors terminating the iterations if they fall below the threshold 10^{-5} which from equation (14), implies that violation of the estimated eigenvalues from the actual ones are at most 10^{-5} .

To further speed up the eigen-decomposition process, we follow *locking* strategy which simply means freezing the eigenvectors that have already converged; obviously such freezing process will start from eigenvectors associated with larger eigenvalues since their convergence rates are higher.

4. EXPERIMENTAL RESULTS

For evaluating the active constrained spectral clustering method that is described on interactive image segmentation we simulate user constraints using the ground truth

segmentation labels. After running the unconstrained spectral clustering (using [14]) once, we start generating constraints sequentially through an automatic procedure shown in Algorithm 1. Steps 5 to 9 are related to constraints generation after taking query from the active learning process. Steps 6 and 8 are different because from (10b) putting cannot-links between two points which are far away from each other won't effect the affinity matrix.

For each image that is shown in this section, the initial adjacency matrix is computed using Gaussian affinity on normalized unit-variance feature vectors which contain spatial and nonspatial information. Effects of constraints are monitored by computing Rand index as the pairwise accuracy of the yielded clustering results. Figure 2 shows the results on the 'football' image (figure 2a) where the sampling threshold is set to be $\delta_s = 0.2$. The only nonspatial feature used here is the gray value of pixels. As can be seen from the first row, unconstrained clustering result is not satisfying because the ball is very similar to its background in terms of the intensity. We used $Max = 100$ in Algorithm 1, therefore 100 pairs of must- and cannot-links are generated sequentially ($2 \times 100 = 200$ constraints in total). In the second row we show the final clustering result on the selected samples (figure 2d) and also the whole image after generalization (2e). For comparison purposes, we have also tried random query selection method which differs only on step 4. Figure 2e shows that active learning could help the algorithm achieve a higher accuracy with much less fluctuations.

We also applied the algorithm on a retinal image, where the vessels are to be segmented. Although retinal vessel segmentation is a very challenging problem, there are few significant efforts for making it user-interacted; [15, 16] are two examples. In figure 3 we have used the output of the Frangi filter [17] followed by a localized histogram equalization as the nonspatial feature. We found that for such complicated vessel structures selecting must-links as the closest point with the same class gives better results. Besides, because of the non-convexity of shapes in the image, for each query we put four additional cannot links (so 1 must-link and 5 cannot-links). In this case, the accuracy doesn't increase very fast and also have some fluctuations. This indicates that although active queries are selected according to the maximally uncertain pixel principle, the constraints provided for them are unsatisfactory for complicated structures. However, still we can see after getting $6 \times 30 = 180$ constraints the result (figure 3g) is significantly more similar to the ground truth (figure 3h).

5. CONCLUSION

In this paper, we have employed constrained spectral clustering to image segmentation after novelty-based down-sampling. Implementation details of the algorithm are also discussed to show how to segment with constrained spectral clustering efficiently. Moreover, some deficiencies with the main algorithm and also the uncertainty-based active learning are illustrated. Despite there being significant rooms for improvement, the results indicate clearly that few user inputs can be effectively used to interactively segment images to a solution. Future work will include modifying the algorithm to exclude heuristics.

Acknowledgments

This project was supported by grants from the National Center for Research Resources (5P41RR012553-14) and the National Institute of General Medical Sciences (8 P41 GM103545-14) from the National Institutes of Health.

References

1. Zhu, Xiaojin; Ghahramani, Zoubin; Lafferty, John. Semi-supervised learning using gaussian fields and harmonic functions. Proceedings of the Twentieth International Conference on Machine Learning. 2003:912–919.

2. Kulkarni, Mayuresh; Nicolls, Fred. Interactive image segmentation using graph cuts. Twentieth Annual Symposium of the Pattern Recognition Association of South Africa; 2009.
3. Kumar, Nimit; Kummamuru, Krishna. Semisupervised clustering with metric learning using relative comparisons. *IEEE Transactions on Knowledge and Data Engineering*. Apr.2008 20
4. Heiler, Matthias; Keuchel, Jens; Schnörr. Semidefinite clustering for image segmentation with a-priori knowledge. *Proceedings of the 27th DAGM conference on Pattern Recognition*; 2005. p. 309-317.
5. Lu, Zhengdong; Carreira-Perpiñán, Miguel Á. Constrained spectral clustering through affinity propagation. *IEEE Conference on Computer Vision and Pattern Recognition*; Jun. 2008 p. 1-8.
6. Houhou, Nawal; Bresson, Xavier; Szlam, Arthur. Semi-supervised segmentation based on non-local continuous min-cut. *Proceedings of the 2nd International Conference on Scale Space and Variational Methods in Computer Vision*; 2009. p. 112-123.
7. Klement, Vladimír; Oberhuber, Tomáš.; Šev ovi , Daniel. Constrained level-set method and its applications to image segmentation. *Applied Mathematics and Computation*. May.2011
8. Paiva, António RC.; Tasdizen, Tolga. Fast semi-supervised image segmentation by novelty selection; *IEEE International Conference on Aucostic Speech and Signal Processing*; Mar. 2010 p. 1054-1057.
9. Shi, Jianbo; Malik, Jitendra. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Learning*. Aug.2000 22
10. Verbeek, Jakob; Vlassis, Nikos. Semi-supervised learning with gaussian fields. *Pattern Recognition*. Jun.2006 39:1864–1875.
11. Golub, Gene H.; Van Loan, Charles F. *Matrix computations*. 3. Johns Hopkins University Press; Baltimore, MD, USA: 1996.
12. Saad, Yousef. *Numerical Methods for Large Eigenvalue Problems*. Society for Industrial and Applied Mathematics; Philadelphia, PA: 2011.
13. Settles, Burr. *Computer Sciences Technical Report 1648*. University of Wisconsin; Madison: 2009. Active learning literature survey.
14. Ng, Andrew Y.; Weiss, Yair; Jordan, Michael I. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processind Systems*. 2001; 14:849–856.
15. You, Xinge; Peng, Qinmu; Yuan, Yuan; Cheung, Yiu-ming; Lei, Jiajia. Segmentation of retinal blood vessels using the radial projection and semi-supervised approach. *Pattern Recognition*. Jan. 2011 44:2314–2324.
16. Ruggeri, Alfredo; Poletti, Enea; Fiorin, Diego; Tramontan, Lara. From laboratory to clinic: the development of web-based tools for the estimation of retinal diagnostic parameters. *33rd Annual International Conference of the IEEE EMBS*; Aug. 2011
17. Frangi, Alejandro F.; Niessen, Wiro J.; Vincken, Koen L.; Viergever, Max A. Multiscale vessel enhancement filtering. *Medical Image Computing and Computer Assisted Intervention*. 1998; 1496:130–137.

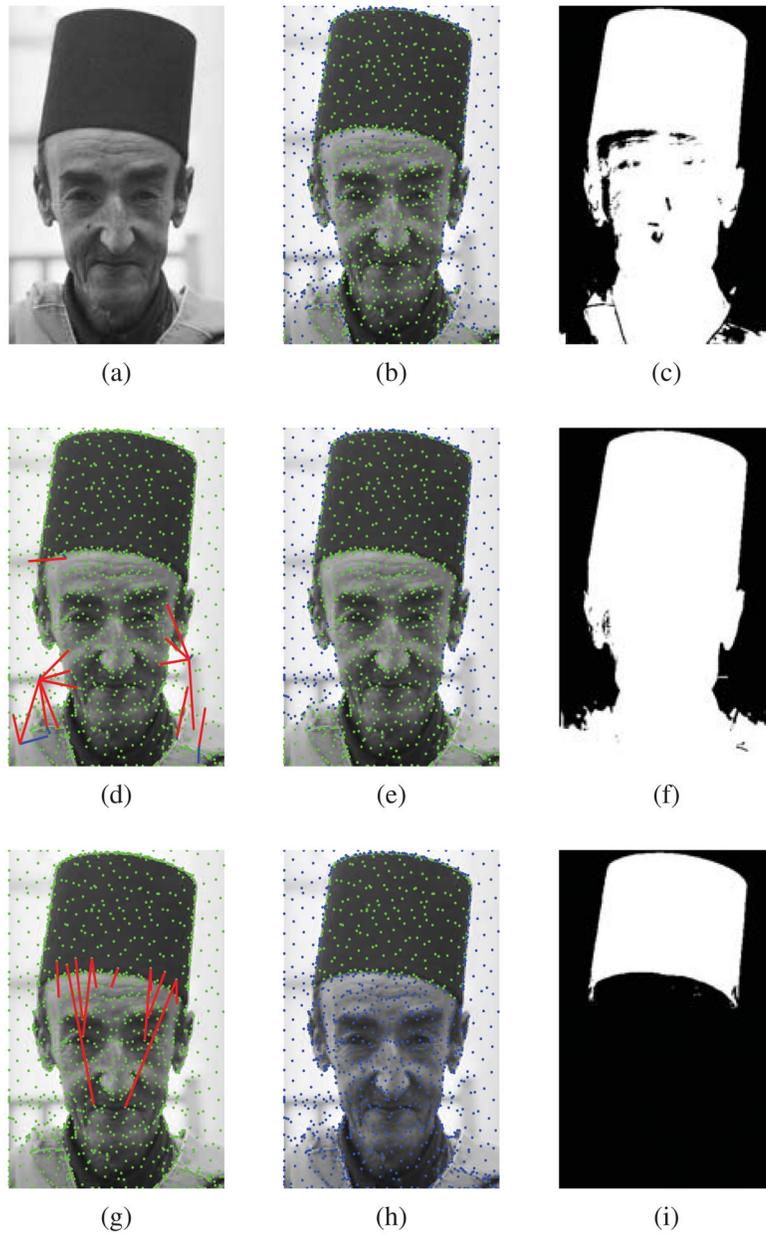


Fig. 1. Putting constraints on sampled pixels: (a) the original image (476×321); (b) clustered sampled pixels using no constraints (1228 samples); (c) result of generalizing the unconstrained clustering; (d,g) two set of constraints: blue lines are must-links and red lines are cannot-links; (e,h) clustered sampled pixels after applying the constraints; (f,i) result of generalizing the constrained clusterings.

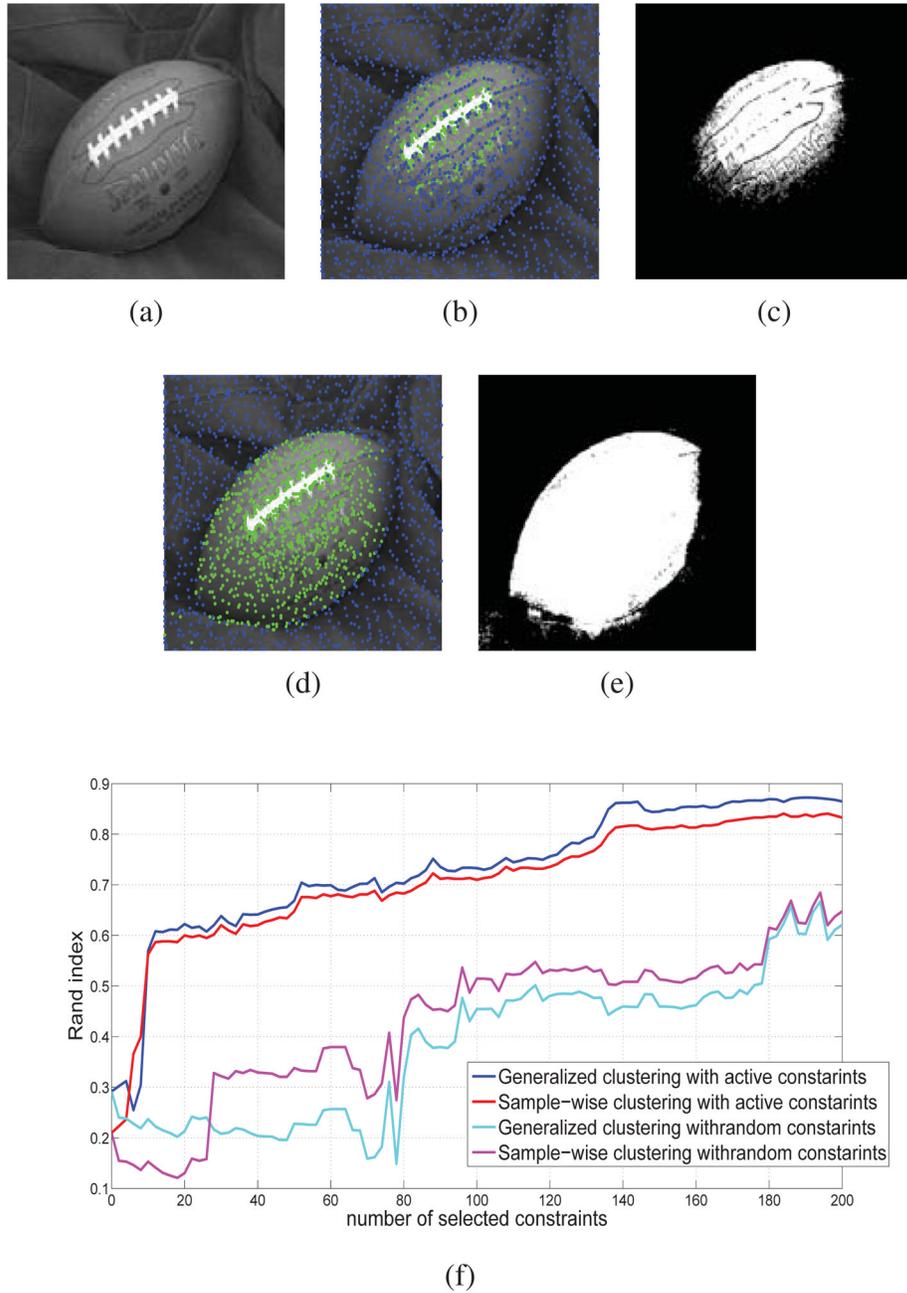
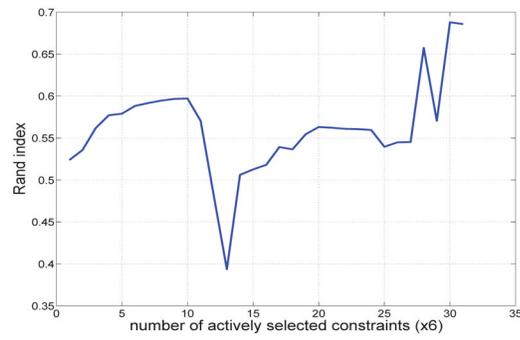


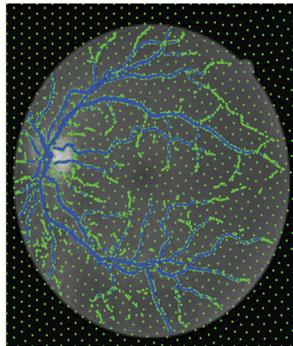
Fig. 2. Constrained clustering on the ‘football’ image: (a) the original image, the goal is to cluster the ball; (b) unconstrained sample-wise clustering result; (c) labels shown in subfigure (b) generalized to the whole pixels; (d) constrained sample-wise clustering result after putting 2×100 constraints actively; (e) labels shown in subfigure (d) generalized to the whole pixels; (e–f) accuracies vs. the number of constraints for active and random query selection strategies respectively.



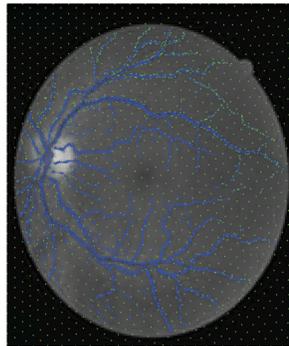
(a)



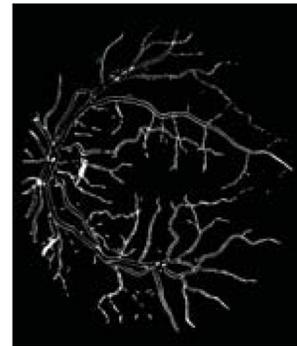
(b)



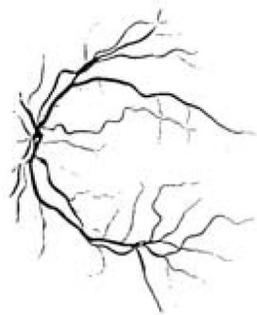
(c)



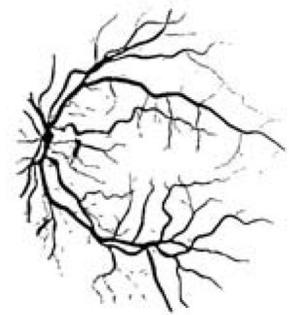
(d)



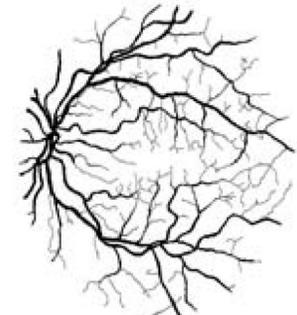
(e)



(f)



(g)



(h)

Fig. 3. Constrained clustering on a ‘retinal’ image: (a) green channel of a retinal image; (b) accuracies vs. the number of actively selected constraints; (c–d) sample-wise unconstrained and constrained clustering results; (e) difference between un/constrained full clustering results (f–g) unconstrained and constrained full-labels; (h) full ground truth.

Algorithm 1**Constrained spectral clustering with automatic constraint generation**

-
- 1: run unconstrained spectral clustering using \mathbf{K}_0 to get labels \mathbf{y}_0 ;
 - 2: $Max \leftarrow$ Maximum number of constraints;
 - 3: **for** $i = 1 \rightarrow Max$ **do**
 - 4: select \mathbf{x}_q from equation (7);
 - 5: $c_q \leftarrow$ actual class of \mathbf{x}_q ;
 - 6: find point with largest likelihood at the same class:

$$\mathbf{x}_M \leftarrow \arg \max_{\mathbf{x} \in S} P(\mathbf{x} | c_q);$$

- 7: put a must-link between \mathbf{x}_q and \mathbf{x}_M ;
- 8: find the closest point to \mathbf{x}_q with the opposite actual class $c(\mathbf{x})$:

$$\mathbf{x}_C \leftarrow \arg \min_{\substack{\mathbf{x} \in S \\ c(\mathbf{x}) \neq c_q}} \|\mathbf{x} - \mathbf{x}_q\|_2;$$

- 9: put a cannot-link between \mathbf{x}_q and \mathbf{x}_C ;
 - 10: update the affinity matrix to get \mathbf{K}_i from equation (8);
 - 11: run spectral clustering on \mathbf{K}_i to get update labels \mathbf{y}_i ;
 - 12: **end for**
-