Constraints as Features

Shmuel Asafi
Tel Aviv University
shmulikasafi@gmail.com

Daniel Cohen-Or
Tel Aviv University
cohenor@gmail.com

Abstract

In this paper, we introduce a new approach to constrained clustering which treats the constraints as features. Our method augments the original feature space with additional dimensions, each of which derived from a given Cannot-link constraints. The specified Cannot-link pair gets extreme coordinates values, and the rest of the points get coordinate values that express their spatial influence from the specified constrained pair. After augmenting all the new features, a standard unconstrained clustering algorithm can be performed, like k-means or spectral clustering. We demonstrate the efficacy of our method for active semi-supervised learning applied to image segmentation and compare it to alternative methods. We also evaluate the performance of our method on the four most commonly evaluated datasets from the UCI machine learning repository.

1. Introduction

Clustering is a fundamental problem in data analysis and is widely used in many fields. Unconstrained or unsupervised clustering problems, where none of the data is labelled, remain an active research field in many domains. However, sometimes some supervision is given on a subset of the data in the form of labelled data, or as pairs of constraints; namely, Must-link and Cannot-link constraints that define a pair of data points that should or should not be associated with the same cluster [1].

Must-links and Cannot-links impose direct constraints on the specified pair of data points; however, the main challenge is that they should have a non-local influence on the clustering result and should affect data points that are further away from the specified pairs. Must-link constraints are usually considered an easier case than Cannot-link constraints, since Must-link constraints are transitive and can be represented as a shortening of the distance between the pair to zero or some other small constant [2, 3]. On the other hand, Cannot-link constraints are non-transitive and have no obvious geometric interpretation [2, 6, 7] and therefore considered a difficult problem.

In this paper, we introduce a new approach for realizing constrained clustering, focusing on Cannot-link constraints. The key idea is to embed the data points into a higher dimensional space, which augments the given space with additional dimensions derived from Cannot-link constraints. The data points are augmented with additional coordinates, each of which is defined by one of the given Cannot-link constraints. The all-pairs distances among the data points in the augmented space combines both the original distances and the distances of points according to each Cannot-link constraint. The actual clustering is then performed by some convenient clustering technique, like k-means or spectral clustering.

The strength of our approach stems from the fact that the realization of a Cannot-link constraint does not impose an early hard classification of data points by the two ”poles” of the given Cannot-link constraint. The technique that we introduce augments the space with additional coordinates that softly distribute the influence of the Cannot-link constraints, and defer the hard decisions to the actual clustering phase. We present a method to define scalar values to each data point that admit to their spatial inference by the introduction of a Cannot-link constraint, thereby defining the augmented coordinates (see Figure 1).

We demonstrate the performance of our method on a number of well known datasets from the UCI benchmark, and to other semi-supervised methods. We also show how it performs as the basic machinery of an active learning application for image segmentation. (no good)

2. Background

The problem of constrained clustering has been a subject of research in the context of semi-supervised learning where only a (typically small) subset of the data is labeled. Wagstaff et al. [5] has first formulated the constraints in the form of pair-wise Must-Link and Cannot-Link constraints. The early work in constrained clustering approaches considered the constraints by directly modifying existing clustering methods by including explicit actions induced by the Must-Link or Cannot-link constraints,e.g.,

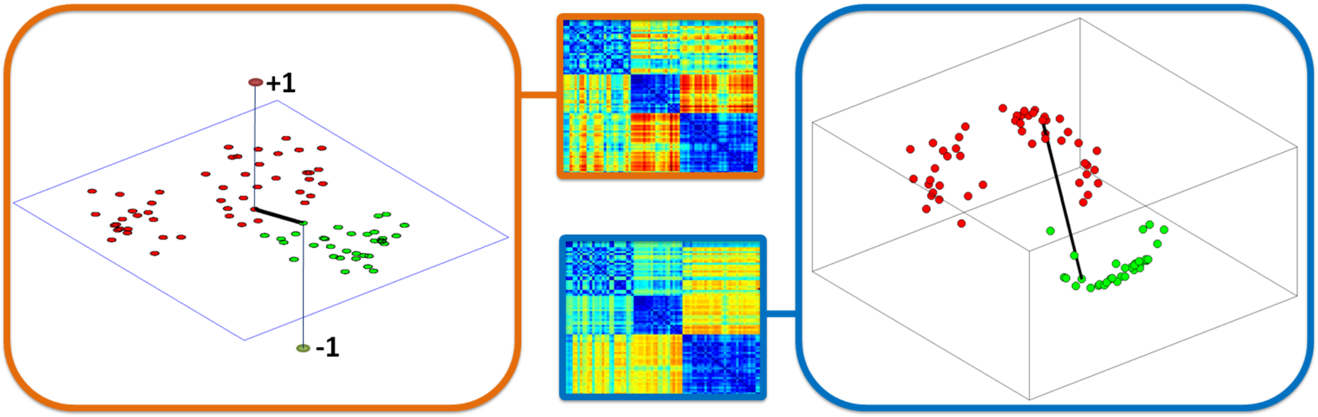

Figure 1. An augmented dimension derived from a Cannot-link constraint. On the left is the original two-dimensional dataset. The colouring is the ground-truth clustering of the data. The Cannot-link constraint is marked by a black line between a pair of points. On the right is the high-dimensional space where an extra dimension was derived from the Cannot-link constraint and augmented to the original space. In the middle are the corresponding distance matrices: the bottom distance matrix can much more easily be clustered to the two ground-truth clusters.

[1, 8, 4, 2]. These methods were only considering hard-constraints, where any inconsistency in the given constraints led to poor results or to break [8, 4].

Kamvar et al. [2] interrupt the Must-Link constraints geometrically by setting a zero-distance between all such pairs, and then recalculate the distance matrix using an all-pair-shortest-path algorithm which enforces the triangle inequality and thereby restores a metric property. However, they were unable to treat the Cannot-link in analog fashion since it remains unclear which distance should be associated with the constraint, and unlike the Must-link, Cannot-link are not transitive. In our work, we adopt their solution to the Must-Link constraints, and introduce a novel technique that also interprets the Cannot-Link constraints geometrically.

Another approach to consider the pair-wise constraints is to learn and define a distance metric that respects the given constraints. These methods attempt to find an optimal linear transformation that warps the standard Euclidean metric into a Mahalanobis one [9, 10, 11, 25]. The advantage of this approach is that the new distances can then be used with standard unconstrained clustering algorithms. The decoupling of clustering method and the pair-wise constraints data embedding is also realized in our approach. However, the metric learning methods do not embed the constraints in the dataset itself, thus they might not be able to warp or bend the space significantly enough to respect the given constraints.

A recent approach is to apply the constraints into spectral clustering methods. Spectral clustering considers the affinity matrix A, which expresses the similarity of pairs of points ($A_{i,j}$), rather than directly the distance matrix. Several attempts were made to redefine the affinity matrix itself according to the given constraints. The idea was first presented by Kamvar and Klein [12] by applying simple local changes to the affinity matrix: Must-Link pairs were set to $A_{i,j} = 1$ meaning they are very similar, and Cannot-Link pairs were set to $A_{i,j} = 0$ meaning they are very dissimilar. This method avoids the problem of determining a proper constant-value for Cannot-Link pairs. However, although elegant, such a naive realization of the concept leads to only limited local geometric impact, and requires many constraints to be placed before a significant effect takes place. Some attempts have been made at improving this method by propagating the constraints themselves to nearby pairs [13, 26, 27]. Other methods propagate the constrained entries in the affinity matrix to nearby entries [14]. Since the affinity matrix is derived directly from the distance matrix, these methods are usually effective with Must-Link constraints, but remain limited when dealing with Cannot-Link constraints [15, 16, 17].

Recently, Wang et al. [18] presented a spring-based method where the actual location of data points in the feature space is altered to agree with the given constraints. Must-Link constraints attract points together and Cannot-Link constraints push points apart, while springs are attached to all pairs of points. Solving the spring-system re-embeds the data points and redefines the distance matrix. However, a Cannot-link constraint repulses the constraints pair together with their nearby points and often creates cluttered regions and conflicts. Figure 2 illustrates the difference between our method and the spring-system based method. As demonstrated, the main caveat of this approach is that the displacements of the points occur within the original dimensions, leaving the relations among the data points in their original subspace untouched. Our method avoids these problems by not altering the original dimensions, but adding a new dimension for each Cannot-Link constraint instead.

3. Overview

We present a constraint clustering approach, where the input points $\{X_i\}_{i=1}^{K}$ are M-dimensional vectors. Typically, each dimension represents a measurement of a feature, and the distance $D_{i,j}$ between X_i and X_j expresses the similarity between them in that feature space. However, in most cases, the feature space is imperfect, so without any constraint, the clustering of the points is likely to be erroneous. The key idea is to convert a given constrained clustering problem to an unconstrained one. More specifically, we assume that we are given a set of points X_i and the constraints are given as Must-links and Cannot-links pairs. The constraints are assumed to be sparse, constituting a semi-supervised clustering problem.

Once the constrained problem is converted to an unconstrained one, any clustering method can be used. Some unconstrained clustering techniques, like spectral methods, do not require as input the embedding of the points in space, but only a distance matrix $D_{i,j}$ that encodes the distances between every pair of points in the feature space. The method that we present takes as input a distance matrix $D_{i,j}$ (extracted from a feature space of M dimensions) and N pairs of constraints and creates an altered distance matrix $\tilde{D}_{i,j}$ (representing distances measured in an augmented space of $M+N$ dimensions). The embedding of the points X in the augmented feature space is also available, if needed.

The calculation of the distance matrix $\tilde{D}$ is performed in two steps. First, the Must-links constraints modify D by shortening constrained pairs and running the all-pair-shortest-path algorithm as described by Kamvar et al. [2], yielding the distance matrix $\hat{D}$. Then to respect the Cannot-link constraints, $\hat{D}$ is augmented by adding additional coordinates to yield $\tilde{D}$. The modified distance matrix $\hat{D}_{i,j}$ is first embedded in M dimensional feature-space. Then, to account for the Cannot-link constraints new coordinates are introduced in additional dimensions, where each coordinate is derived directly from one Cannot-Link constraint. As a result, each additional coordinate can be regarded as a new feature representing a constraint. Assuming N pairs of Cannot-link constraints, the augmented feature space is now in $M + N$ dimensions, and the $\tilde{D}$ distance matrix can be defined, respectively.

In Section 4 we elaborate more on the initial modification of the feature space to respect the Must-link constraints. In Section 5 we present the conversion of a Cannot-link constraint to a novel feature dimension. In Section 6 we discuss the elementary requirements of an interactive semi-supervised clustering system, where in Section 7 we demonstrate such a system with a simple active semi-supervised image segmentation application. In Section 8 we evaluate the performance of our method on the UCI benchmark and compare it to other methods.

4. Converting Constraints to Features

The first step is to modify the input distance matrix D by respecting all the Must-link constraints to define $\hat{D}$. Following [2] we recalculate all-pair-shortest-path using the following steps:

- Extract the minimal distance d_{min} greater than zero from the input distance matrix D.
- Initialize the updated matrix $\hat{D} = D$
- For every Must-Link constrained pair (i, j), shorten their distance to $\hat{D}_{i,j} = d_{min}$.
- Restore triangle-inequality by updating $\hat{D}_{x,y} = min(\hat{D}_{x,y}, \hat{D}_{x,i} + \hat{D}_{i,j} + \hat{D}_{j,y})$, for every pair x, y.

The above algorithm shortens the entries in the distance matrix D between constrained pairs to a minimum constant and then updates all other distances in the matrix appropriately so the triangle-inequality holds. The restoration of the triangle-inequality for each pair x, y is a single operation and does not necessitate an iterative process since the shortest path must either be the existing path between x, y or the path that goes directly through i, j. Since now $\hat{D}$ defines a valid metric, it can be used to re-embed the data points back into feature space. We choose to use a positive constant d_{min} instead of zero, so as to avoid two distinct points being re-embedded in the same exact location.

The implementation requires going through every pair of points for every Must-Link constraint, thereby its time-complexity is $O(N \cdot K^2)$, where N is the number of Must-Link constraints, and K is the number of data points.

Once $\hat{D}$ is available, we need to account for the N Cannot-link constraints to define $\tilde{D}$:

$$\tilde{D}^{p}_{i,j} = \hat{D}^{p}_{i,j} + \sum_{c=1..N} (\alpha \cdot D^{(c)}_{i,j})^{p}, \tag{1}$$

where $D^{(c)}$ are constrained distance matrices, N is the number of Cannot-Link constraints and α is a factor determining the degree of influence the Cannot-Link constraints have. α can also be a function of N to account for cases where the number of constraints is relatively large with respect to M. The altered distance matrix $\tilde{D}$ is calculated in p-norm. In our implementation we used the Euclidean norm ($p = 2$) and set $\alpha = d_{max}$, where d_{max} is the maximum distance in $\hat{D}$, so that constrained distances are normalized to the scale of distances in $\hat{D}$.

In the following we show how to define $D^{(c)}$ for a given Cannot-link constraint.

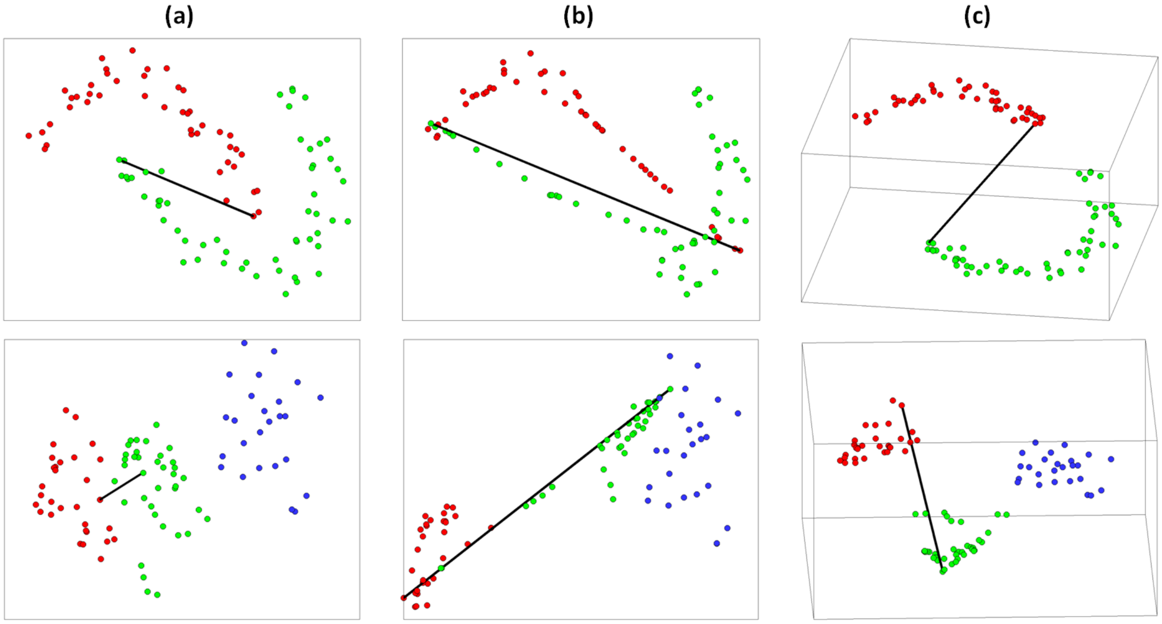

Figure 2. Two comparisons of our method to the spring-system based method. In each row: (a) original feature space with groundtruth clustering and a Cannot-Link constraint in black. (b) the spring-system re-embeds the points in an undesired clutter. (c) our method augments a dimension in which the desired points are distanced without damaging local structure or creating a clutter of points.

5. Converting a Cannot-link constraint to a feature

For each Cannot-Link constraint we derive a new feature dimension encoded in $D^{(c)}$. The framework of our method is not coupled with any specific algorithm for deriving the new dimension, but the algorithm should obey the following guidelines:

- The constrained points in a pair should be placed as far apart as possible by giving them extreme values along the new dimension: one minimum and the other maximum.

- All other data points should be assigned a scalar that expresses their relation with the constrained pair, or their likelihood to be clustered with one of the constrained pair.

We have chosen to implement a distance derivation for the new dimension based on the Diffusion Maps, since the distances between points in the diffusion space better reflect their likelihood to be clustered together [20]. Notice, however, that we are not using the Diffusion Map to make clustering decisions directly. The calculation of the Diffusion Map will be explained below.

Given a Cannot-link constraint, its endpoints $c1$ and $c2$ are given fixed extreme values 1 and -1, respectively. All other points are given values according to the following equation:

$$v_i = \frac{(\varphi(i, c2) - \varphi(i, c1))}{(\varphi(i, c2) + \varphi(i, c1))}, \tag{2}$$

where $\varphi(x, y)$ is the distance between points x and y in the Diffusion Map, and v_i is the value given to point i. Points that are closer to $c1$ in the Diffusion Map get values closer to 1, points closer to $c2$ get values closer to -1, and points that are as close to both get values close to 0. Figure 3 shows the 'closeness' values in the Diffusion Map and demonstrates why it is preferable over Euclidean distances.

We denote the *constrained distance matrix* between points in the new dimension derived from constraint c by:

$$D^{(c)}_{i,j} = |v_i - v_j| \tag{3}$$

The distance matrix of the original dataset can be interpreted as an undirected graph that encodes "walking distances" between pairs of points. A "random traveller" on that graph is more likely to walk on short edges than on long ones; this likelihood is the affinity between two points and is usually calculated using a Gaussian kernel:

$$A_{i,j} = e^{-\frac{D^2_{i,j}}{\sigma^2}}, \tag{4}$$

where σ is the kernel's width, and the affinity matrix A is normalized to create a stochastic matrix.

A Diffusion Map is a re-embedding of the dataset that places two points close together if there is a likely "walking path" between them. Nadler et al. [20] showed that eigen-analysis of matrix A is equivalent to the calculation of random walk probabilities. Thus, each point x in the dataset is re-embedded using coordinates derived from the eigenvectors and eigenvalues of matrix A:

$$\Psi_t(x) = (\lambda_1^t \psi_1(x), \lambda_2^t \psi_2(x), ..., \lambda_K^t \psi_K(x)), \qquad (5)$$

where λ_i is the i'th eigenvalue, ψ_i is the i'th eigenvector, and t is a "walking time" or scaling parameter which we derive automatically from the dataset following [19].

The distance between two points x, y in the Diffusion Map is then simply:

$$\varphi(x, y) = |\Psi_t(x) - \Psi_t(y)| \qquad (6)$$

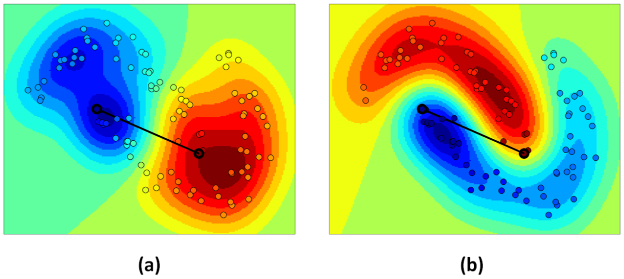

Figure 3. Illustration of distances in Euclidean vs. Diffusion spaces. The two elements marked black are the constrained endpoints. The coloring reflects how close are other elements to the two endpoints. (a) Distances in Euclidean map. (b) Distances in Diffusion map.

6. Interactive Semi-Supervised Clustering

Our method was designed to work well in interactive Semi-supervised set-ups, where a user is first given an unconstrained clustering result, and is then expected to add constraints one-by-one or few at a time, until the clustering result becomes satisfying. An interactive constrained clustering method should follow these guideline in order to be effective:

- Calculations should be relatively efficient in time-complexity, since the system is interactive.
- Every addition of a constraint should have a noticeable effect, otherwise the user will be discouraged quickly.
- Early constraints should have relatively significant effects, later constraints should have finer effects. This allows the user to lead the system into stable convergence.

The time complexity of our method is reasonable, allowing interactive semi-supervision. Adding any single Must-Link constraint is a $O(K^2)$ operation since it only includes updating the all-pair-shortest-path matrix $D_{(SP)}$; adding a single Cannot-Link constraint can also be achieved in $O(K^2)$ assuming the Diffusion Map is calculated in a preprocess. This is achieved by calculating v_i for each element ($O(K)$), calculating $D_{i,j}^{(c)}$ for each pair ($O(K^2)$) and updating $\tilde{D}_{i,j}^{p}$ by an element-wise sum (also $O(K^2)$). Of course, the unconstrained clustering algorithm itself must also be time-efficient.

Our method enables a quick convergence to a desirable stable state. The first Cannot-Link constraints added by the user have an immediate and significant effect on the resulting distance matrix $\tilde{D}$. This effect is a result of α (in Eq 1) having a constant value, which is determined by the maximal distance in the original distance matrix. Thus, the first constrained distance matrix $D^{(c)}$ is given a weight which is equivalent to the weight of the original distance matrix. Additional constraints all share the same total weight α, so the incremental addition of more constraints causes each constraint to become gradually less significant. Notice, that the order by which constraints are added does not affect the final result. Different sequences produce different intermediary distance matrices, but produce the same final distance matrix.

7. Constrained Image Segmentation

We have implemented a simple constrained image segmentation application to demonstrate the applicability of our method to interactive semi-supervised clustering problems. The application prompts the user to select an image and a groundtruth segment labelling of its pixels. First, the image is over-segmented into super-pixels using the method provided by [21]. Each super-pixel is represented as a data point with five features: average X coordinate, average Y coordinate and average RGB values. The constructed feature space is fairly naive and does not convey many of the usable features to segmentation like textures and edges. We believe this represents many real-life problems, where an ideal feature space is not available, thereby putting much responsibility on the supervision.

An initial unconstrained segmentation is constructed using standard Spectral Clustering, taking the distances between the feature vectors in L2-norm. The user can then either insert a manual constraint by clicking on the desired pixels (for either Must-Link or Cannot-Link constraints), or she can request the application to insert a random constraint respecting the groundtruth labelling. The application constructs the constrained segmentation using 4 different methods: Constraints as Features, Spectral Learning (Kamvar 2003) [12], Affinity Propagation [14] and Constrained Spectral Clustering (CSC) [15]. Only a 2-way clustering algorithm was provided for the latter method; thus only 2-segment examples (foreground-background) were tested

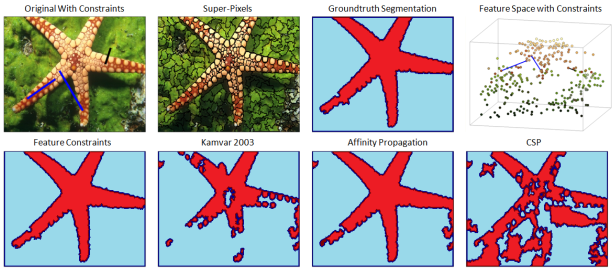

Figure 4. Result on Semi-Supervised image segmentation. Top (left-to-right): Original image with a total of 3 constraints (two Must-Links in blue and one Cannot-Link in black), oversegmentation to super-pixels, ground truth segmentation, 3D view of the Feature Space with constraints. Bottom (left-to-right): Our result, and other compared methods. It is clear that our algorithm has a superior result, though Affinity Propagation is a close match.

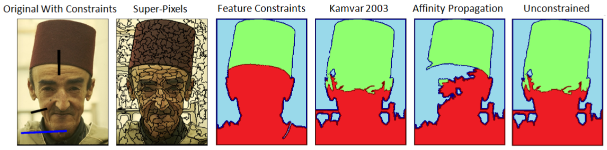

Figure 5. Result on Semi-Supervised image segmentation. Left-to-right: Original image with a total of 3 constraints (one Must-Links in blue and two Cannot-Link in black), oversegmentation to super-pixels, our result, other compared methods and unconstrained result. It is clear that our algorithm has a superior result.

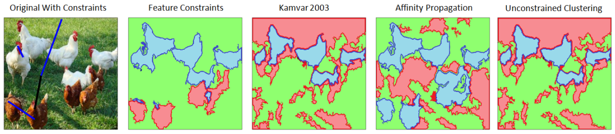

Figure 6. Result on Semi-Supervised image segmentation. Left-to-right: Original image with a total of 4 constraints (three Must-Links in blue and one Cannot-Link in black), our result, other compared methods and unconstrained result. Our algorithm has a superior result, although not perfect.

here. The method introduced by [26] was also attempted at this application, but failed to produce comparable results.

We have run our application on several images taken from the Berkeley Segmentation Dataset[22]. Two of our results are shown in Figures 4, 5. It is visually clear that our method outperforms the competition. In Figure 7 we show the results for random constraints drawn according to the ground truth segmentation, for which we have also compared to Information Theory Metric Learning (ITML) [25]. Our method reaches a good result with very few constraints, and is able to quickly improve when more constraints are added.

We have also compared our constrained algorithm to the other methods, with random constraints over the 117 images with four or less ground-truth segments that are in the Berkeley Segmentation Dataset. Each image was tested with 50, 100, 200 and 400 constraints; each test result is the average result of 10 random constraints-set. Table 1 shows a histogram, counting which method outperformed the rest for a total of 468 tests. Over all the 4,680 runs of random constraints-sets, our method (denoted as ”Feature”) outperformed the rest in 2,324 cases, Affinity Propagation and ITML both separately in 1,097 cases, Spectral Learning in 102 cases, and Constrained Spectral Clustering in only 60 cases. This ratio is consistent with the average results.

8. Results on UCI Datasets

We evaluate the performance of our method on the four most commonly evaluated datasets from the UCI machine learning repository [23]: Iris, Wine, Hepatitis and Ionosphere. Clustering results are evaluated using the Adjusted Rand Index [24]. Our results are compared with five other methods: Spectral Learning (Kamvar 2003)[12], Affinity Propagation [14], Constrained Spectral Clustering (CSC)[15], Information Theory Metric Learning (ITML) [25] and E2CP [26]. The Affinity Propagation, Constrained Spectral Clustering and E2CP methods were implemented and evaluated using the authors’ own MATLAB code; The ITML method of finding a Mahanabolis distance metric was implemented using the authors’ code, after which a standard implementation of Spectral Clustering was used. The Spectral Learning method was evaluated using our own implementation following the implementation description provided by the authors [12]. Implementation of the CSP method was only available for 2-way clustering, thus following [15], we reduced the datasets to the two most difficult classes to discern.

The methods were evaluated using 10 to 200 random Must-Link and Cannot-Link constraints, drawn randomly from the groundtruth clustering. We take average, minimum and maximum results over 50 different random sets of constraints for each number of constraints. Results are presented in Figures 8 and 9. Our method is more successful than other methods in most cases, and in other cases it does only slightly worse than the closest competition. The unconstrained results of each method is different, because each has a different implementation of the diffusion map or of K-means. The Constrained Spectral Clustering (CSP) and Spectral Learning (Kamvar 2003) methods require relatively many constraints to be placed before they make a significant impact, and they usually exhibit a monotonically increasing function of clustering accuracy against number of constraints. On the other hand, the Affinity Propagation method and our method are able to achieve a significant effect with relatively few constraints, but suffer from non-monotony: the addition of more constraints might some-

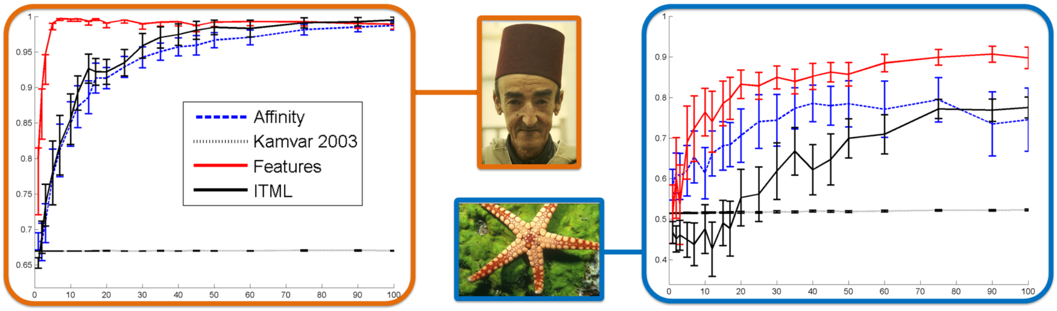

Figure 7. Result on Semi-Supervised image segmentation. Performance of four methods on two images, evaluated using the Adjusted Rand Index (vertical) against a ground truth segmentation. Evaluated for 1 to 100 randomly chosen constraints (horizontal). Average, minimum and maximum results over 50 iterations.

	50 Constraints					100 Constraints					200 Constraints					400 Constraints				
	30	60	90	117	All	30	60	90	117	All	30	60	90	117	All	30	60	90	117	All
Features	**13**	**17**	**11**	**13**	**54**	**14**	**20**	**16**	**12**	**62**	**15**	**22**	**20**	**18**	**75**	**15**	**16**	12	**16**	**59**
Affinity	12	8	**11**	7	38	10	5	5	6	26	8	0	0	1	9	8	5	**14**	4	31
ITML	5	4	7	5	21	6	5	8	8	27	7	7	9	7	30	8	9	13	7	27
Kamvar	0	1	1	2	4	0	0	1	1	2	0	1	1	1	3	0	0	1	0	1

Table 1. Image Segmentation Comparison of images from BSD. Test results of 117 images with 50, 100, 200 and 400 random constraints. Each test is broken to top 30 images (with best maximal result), top 31 to 60, top 61 to 90, bottom 91 to 117, and overall; this shows that our method outperforms the rest over images that are easy or hard to segment.

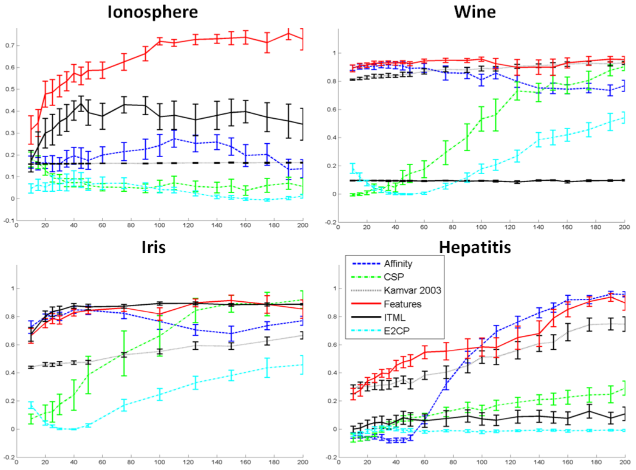

Figure 8. Results on the most common UCI datasets. All datasets were reduced to 2-way clustering problems by selecting the hardest two classes to discern. The figure shows clustering results evaluated using the Adjusted Rand Index on the vertical scale, number of constraints on the horizontal scale.

times cause less accurate clustering. The running time of all methods is comparable, spectral analysis being the most prominent factor in all of them.

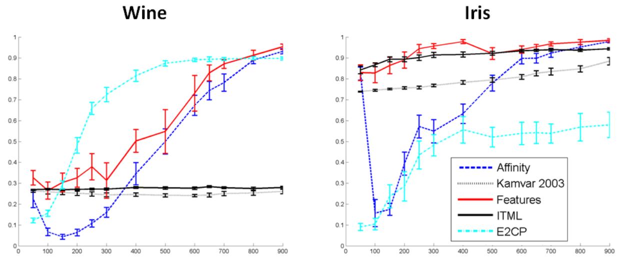

Figure 9. Results on the 3-way datasets of Wine and Iris from UCI. These datasets were not tested on the Constrained Spectral Clustering method [15] since it's available implementation supports only 2-way clustering. The figure shows clustering results evaluated using the Adjusted Rand Index on the vertical scale, number of constraints on the horizontal scale.

9. Conclusions and Discussion

We have presented a technique for constrained clustering where new features derived from pair-wise constraints are augmented to the original data feature space. Our algorithm is decoupled from the actual clustering as it merely redefines or warps the distances among the points in the feature space. We have performed an extensive evaluation to demonstrate that our method outperforms alternative methods in most cases.

Similarly to commonly used techniques in optimization, whereby a constrained problem of the sort $\min f(x)$ s.t. $c(x) = 0$ is converted into the unconstrained problem $\min f(x) + \mu c(x)$, in our setting the hard Cannot-link constraints are incorporated into an unconstrained system where those constraints are not guaranteed to be satisfied.

Our method extends the dimension of the problem. While this is the key of our method, it also incurs two intricacies: first, if the number of constraints is relatively high to the original space dimension, the complexity of the computation increases significantly. However, in most cases, we care about a sparse set of constraints. The second problem is the weight of the additional dimension relative to the original one. As we described in Section 5, in all our experiments we gave the additional dimension the same overall weight as the original ones. Although this was proved to work well, it remains as an open problem for future research.

References

[1] K. Wagstaff and C. Cardie, *Clustering with Instance-level Constraints*, The Seventeenth International Conference on Machine Learning, 2000.

[2] 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*, The Nineteenth International Conference on Machine Learning, 2002.

[3] Z. Li, J. Liu and X. Tang *Constrained Clustering via Spectral Regularization*, CVPR 2009.

[4] I. Davidson and S. S. Ravi *Clustering With Constraints: Feasibility Issues and the k-Means Algorithm*, Fifth SIAM International Conference, 2005.

[5] K. Wagstaff and C. Cardie *Clustering with Instance-level Constraints* Seventeenth International Conference on Machine Learning, 2000

[6] I. Davidson and S. S. Ravi *Agglomerative Hierarchical Clustering with Constraints: Theoretical and Empirical Results*, Lecture Notes in Computer Science, 2005.

[7] I. Davidson and S. S. Ravi emphUsing Instance-Level Constraints in Agglomerative Hierarchical Clustering: Theoretical and Empirical Results, Data Mining and Knowledge Discovery, 2009.

[8] K. Wagstaff, C. Cardie, S. Rogers and S. Schroedl *Constrained K-means Clustering with Background Knowledge*, Eighteenth International Conference on Machine Learning, 2001.

[9] E. P. Xing, A. Y. Ng, M. I. Jordan and S. Russell *Distance Metric Learning, with Application to Clustering with Side-Information*, Advances in Neural Information Processing Systems 15, 2002.

[10] S.C.H. Hoi, W.Liu and S. Chang *Semi-Supervised Distance Metric Learning for Collaborative Image Retrieval*, CVPR 2008.

[11] A. Bar-Hillel, T. Hertz, N. Shental and D. Weinshall *Learning Distance Functions using Equivalence Relations*, Twentieth International Conference on Machine Learning, 2003.

[12] S. D. Kamvar, D. Klein, C. D. Manning *Spectral Learning*, IJCAI 2003.

[13] A. Sharma, E. Lavante and R. Horaud *Learning Shape Segmentation Using Constrained Spectral Clustering and Probabilistic Label Transfer*, Eleventh European Conference on Computer Vision, 2010.

[14] Z. Lu and M. A. Carreira-Perpinan *Constrained Spectral Clustering through Afnity Propagation*, CVPR 2008.

[15] X. Wang, B. Qian and I. Davidson *On Constrained Spectral Clustering and Its Applications*, Cornell University Library, 2012.

[16] Q. Xu, M. desJardins, K. Wagstaff *Constrained Spectral Clustering under a Local Proximity Structure Assumption*, FLAIRS-05, 2005.

[17] X. Wang and I. Davidson *Active Spectral Clustering*, ICDM 2010.

[18] Y. Wang, S. Asafi, O. van Kaick, H. Zhang and D. Cohen-Or *Active Co-Analysis of a Set of Shapes*, Sigraph Asia, 2013.

[19] S. Lafon and A. B. Lee *Diffusion Maps and Coarse-Graining: A Unified Framework for Dimensionality Reduction, Graph Partitioning, and Data Set Parameterization*, IEEE Trans. on Pattern Analysis and Machine Intelligence 28(9): 1393-1403, 2006.

[20] B. Nadler, S. Lafon, R. R. Coifman and I. G. Kevrekidis *Diffusion Maps, Spectral Clustering and Eigenfunctions of Fokker-Planck Operators*, Advances in Neural Information Processing Systems 18, 2005.

[21] M. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa *Entropy Rate Superpixel Segmentation*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR 11) , Colorado Spring, 2011.

[22] D. Martin and C. Fowlkes and D. Tal and J. Malik *A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics*, Proc. 8th Int'l Conf. Computer Vision (ICCV), 2001.

[23] A. Asuncion and D. Newman. *UCI machine learning repository*, 2007.

[24] K. Y. Yeung and W. L. Ruzzo *Details of the Adjusted Rand index and Clustering algorithms*, 2001.

[25] J.V. Davis, B. Kulis, P. Jain, S. Sra and I.S. Dhillon *Information-theoretic metric learning*, ICML, 2007.

[26] P. He, X. Xu, and L. Chen *Constrained Clustering with Local Constraint Propagation*, Computer Vision ECCV, 2012.

[27] Z. Lu, H.S. Ip *Constrained Spectral Clustering via Exhaustive and Efficient Constraint Propagation*, Computer Vision ECCV, 2010.