Knowledge Driven Dimension Reduction For Clustering

Ian Davidson
Department of Computer Science
University of California - Davis
davidson@cs.ucdavis.edu

Abstract

As A.I. algorithms are applied to more complex domains that involve high dimensional data sets there is a need to more saliently represent the data. However, most dimension reduction approaches are driven by objective functions that may not or only partially suit the end users requirements. In this work, we show how to incorporate general-purpose domain expertise encoded as a graph into dimension reduction in way that lends itself to an elegant generalized eigenvalue problem. We call our approach Graph-Driven Constrained Dimension Reduction via Linear Projection (GCDR-LP) and show that it has several desirable properties.

1 Introduction and Motivation

Consider the following situation: you wish to apply a unsupervised data-mining approach such as clustering to some high-dimensional data such as text, video or audio. However, most algorithms have time/space complexity at least linear with respect to the number of dimensions and will hence take a long time to converge on your data set. Furthermore, you strongly believe that not all dimensions are necessary and transforming the data to a lower dimensional space will make the problem not only computationally easier but also allow for patterns to more easily be discovered. Approaches such as principal component analysis (PCA), factor analysis (FA) or singular value decomposition (SVD) are appropriate in domains where little background knowledge is known. However, you do have knowledge in the form of what pairs of objects are similar or dis-similar. For example this could be derived from a small amount of labels on the data or manually examining a small number of instances. From this information **a graph can be constructed** with each node being an instance and an edge indicating the relationship between the instances, a positive edge-weight represents similarity and a negative edge-weight dis-similarity. This graph will typically be small since domain knowledge is known only on a small subset of the instances but can also incorporate other information such as local geometry to preserve the nearest neighbors of each point. We list a general version of the problem below:

Problem 1 ***The Graph-Driven Constrained Dimension Reduction Problem.*** *Given a weighted graph G and a data set of points D in r dimensional space where the nodes in G correspond to a* ***subset*** *of the points in D. Find a projection of this space into a lower s dimension space so that the pair of nodes/points in G with a positive edge are close together and those with a negative edge are far apart.*

We will provide more detail on our solution to this problem later but it is important to note the problem of focus in this paper is different to spectral clustering (dimension reduction) in two keys ways. Firstly, we are projecting the entire space D occupies not just the points in G or D. Secondly, we do not formulate the problem as some form of min-cut and then solve a relaxed version of the problem.

Our work aims to find a reduced dimension space based on the graph and is conceptually most similar to **encoding** a list of must-link (similar) and cannot-link (dissimilar) instance-level constraints into clustering algorithm which was first introduced to the data mining and machine learning communities by Wagstaff and Cardie [Wagstaff et. al. 2001] with significant extensions by Basu and collaborators [Basu et. al. 2004]. Xing and collaborators [Xing et. al. 2002] introduced the idea of learning a distance function then performing clustering with it. In this context, the points that are part of a must-link (cannot-link) constraint should be close together (far apart). However, their approach does not perform dimension reduction and as we experimentally show fairs poorly in high dimensional data. Furthermore, we allow weights in our graph so can model degrees of belief in these propositions.

Though each of the above types of approaches were shown to yield impressive results particularly at improving the cluster accuracy when measured on extrinsic labels they have several significant limitations. Firstly, we show in Table 1 (columns 3, 4 and 5) that these approaches do not fair well when the data is best clustered in a reduced/lower dimensional space. This is to be expected since they implicitly assume (like most clustering algorithms) that dimension reduction has been performed prior to their application. However, we also show that for these data sets that classic unguided dimension reduction techniques such as PCA perform poorly (see Table 1, columns 6 and 7). Secondly, trying to simultaneously find a good clustering while also satisfying the constraints can be quite limiting. Algorithms that attempt to satisfy all constraints such as the COP-k-means algorithm are known to not converge when dealing with even relatively few number of constraints [Davidson and Ravi 2007] and

constrained spectral-clustering formulations [Coleman et. al. 2008] only exist for $k > 2$ for just must-link constraints and not cannot-link constraints.

Our aim is to create an approach to address the dimension reduction problem specifically for unsupervised learning with the aid of hints/constraints with the following properties.

- The domain knowledge can be represented as a weighted graph which can represent both domain knowledge such as similar/dissimilar instances but also other properties such as local geometry.
- The approach is general-purpose being usable in a wide variety of mining algorithms and easy to implement.
- The approach is fast and computable in closed form.
- The approach produces a mapping everywhere to a reduced dimensional space not just for constrained points.
- The new dimensions are easily interpretable.

We show that we can do all of the above by formulating our constrained dimension reduction problem as a **linear** dimension reduction problem that gives rise to a generalized eigenvalue problem. A closed form solution to the problem exists that is easily implementable in MATLAB and whose result is easily understood as producing a new set of dimensions that are a linear combination of the old ones. We call our approach Graph-Driven Constrained Dimension Reduction via Linear Projection (GCDR-LP) for clustering.

We begin this paper by formally describing the approach in section 2. The approach requires construction of a weighted constraint graph and we discuss several ways of doing this in section 3. We then show in section 4 experimentally that clustering approaches that use constraints fair poorly on many UCI data sets with additional noise dimensions but our approach does better. Our approach works well at reducing dimensionality for facial image data performing better than the unconstrained eigen-faces (PCA-style) approach as shown in section 5. We describe related and future work and finally conclude by summarizing our contributions.

2 The GCDR-LP Approach

To reduce the dimensionality of the points we propose creating a *linear* relationship between their old positions and new positions of the form $q = A^T x$ where A is a $r \times s$ matrix, x the point in the higher dimensional space (described by a column vector) and q the point in the lower dimensional space. Therefore the points were originally in a r dimensional space and will be reduced to a s dimensional space. Where as approaches such as PCA are guided by an objective function that finds the projection that maximizes the data variance, our approach will be guided by a user-defined constraint graph that captures their knowledge of the problem. Note that $A = \{a_1 \dots a_s\}$ and that the i^{th} column vector in A specifies that the i^{th} dimension of the reduced space as being some linear combination of the higher dimensions. It should be immediately noticed that the mapping is linear and is global/constant regardless of where the point is in the original space. It is left to future work to explore non-linear and possibly local mappings, but that such transformation will most likely come at the cost of the efficient and easy implementation that our approach gives.

It is now left to describe how A is calculated and for that we need to introduce the notion of a constraint graph. A constraint graph $G(V, E, W)$ consists of a vertex for each point and positive edge-weights indicating similarity and negative edge-weights indicating dissimilarity between the vertices/points they are incident on. The absence of an edge (zero weight) indicates that no knowledge is known about the points. In the next section we describe ways to create constraint graphs, but in the mean time, consider that the only non-zero weights in the graph are positive weights if two instances are similar and negative weights if they are dissimilar with the magnitude of the weight indicates the degree of belief in this proposition.

Definition 1 ***Constraint Graph Definition***. *Let $G(V, E, W)$ be a basic constraint graph with the properties that for each pair of similar instances (i,j) $w_{i,j} > 0$ and for each pair of dissimilar instances $w_{i,j} < 0$, else $w_{i,j} = 0$. Note that implicitly $w_{i,i} = +1 \; \forall \; i$, that $w_{i,j} = w_{j,i}$ and there is a requirement that $\forall \; i : \; \sum_j w_{i,j} > 0$.*

Given the definition of G a reasonable objective function is to map the points onto a single dimension (line) so as to minimize the distance between the constrained points multiplied by their weight pair. In the following derivation the column vector **a** specifies a points location on this line as a linear combination of the points position in the original space. Since the weight is negative for dissimilar instances and positive for similar instances this emphasizes our desired result that similar points are close together and dissimilar points are far apart. Formally:

Constrained Dimension Reduction Objective Function

$$\arg\min_{\mathbf{a}} \frac{1}{2} \sum_{i,j} (q_i - q_j)^2 w_{i,j} \tag{1}$$

$$= \arg\min_{\mathbf{a}} \frac{1}{2} \sum_{i,j} (\mathbf{a}^T x_i - \mathbf{a}^T x_j)^2 w_{i,j}$$

We now show how equation 1 can be converted into a generalized eigenvalue problem that is easily solvable in MATLAB or any other package that can compute eigen-vectors and the corresponding eigenvalues.

Expanding equation 1 we obtain

$$\arg\min_{\mathbf{a}} \frac{1}{2} [\sum_{i,j} \mathbf{a}^T x_i w_{i,j} x_i^T \mathbf{a} - \sum_{i,j} \mathbf{a}^T x_i w_{i,j} x_j^T \mathbf{a} - \sum_{i,j} \mathbf{a}^T x_j w_{i,j} x_i^T \mathbf{a} + \sum_{i,j} \mathbf{a}^T x_j w_{i,j} x_j^T \mathbf{a}] \tag{2}$$

Let D be a diagonal matrix such that the entry $d_{i,i} = \sum_j w_{i,j}$. Then note that the first expression of the equation 2 equals $\sum_i \mathbf{a}^T x_i d_{i,i} x_i^T \mathbf{a}$ since w is only summed over j and this can be written as $\mathbf{a}^T X D X^T \mathbf{a}$. Also, due to symmetry (by virtue that the similarity/dissimilarity is symmetrical) then the fourth expression of the above equation will also yield this exact same result. Similarly, due to symmetry the second and

third expressions are equal and together they yield the result $-\mathbf{a}^T XWX^T\mathbf{a}$ and hence the objective function is:

$$\arg\min_{\mathbf{a}} \mathbf{a}^T XDX^T\mathbf{a} - \mathbf{a}^T XWX^T\mathbf{a} \quad (3)$$
$$= \arg\min_{\mathbf{a}} \mathbf{a}^T X(D-W)X^T\mathbf{a}$$

However, equation 3 is unbounded so we add the constraint $\mathbf{a}^T XDX^T\mathbf{a} = 1$ to remove scaling issues and turn the problem into a constrained optimization problem of the form:

$$\arg\min_{\mathbf{a}} \mathbf{a}^T X(D-W)X^T\mathbf{a} \quad (4)$$
$$\textbf{subject to: } \mathbf{a}^T XDX^T\mathbf{a} = 1$$

We can turn equation 4 into a unconstrained problem by casting it as a Lagrange multiplication problem with λ being the Lagrange multiplier and noting that the constraint should be rewritten to equal 0.

$$\arg\min_{\mathbf{a}} \mathbf{a}^T X(D-W)X^T\mathbf{a} - \lambda.(\mathbf{a}^T XDX^T\mathbf{a} - 1) \quad (5)$$
$$= \arg\min_{\mathbf{a}} \mathbf{a}^T X(D-W)X^T\mathbf{a} - \lambda.\mathbf{a}^T XDX^T\mathbf{a} + \lambda \quad (6)$$

taking the first order derivative with respect to $\mathbf{a}$ *yields*

$$\arg\min_{\mathbf{a}} X(D-W)X^T\mathbf{a} - \lambda.XDX^T\mathbf{a} + 0 \quad (7)$$
$$\arg\min_{\mathbf{a}} X(D-W)X^T\mathbf{a} = \lambda.XDX^T\mathbf{a} \quad (8)$$

We note that equation 8 is precisely in the form of a generalized eigenvalue problem with $\mathbf{a}$ being the eigen-vector of the corresponding smallest eigenvalue which can be efficiently solved for in closed form. The smallest eigenvalues' eigenvectors that are the solutions to this problem are the $a_1 \ldots a_s$ discussed earlier and describe the lower dimensional space the points are mapped to and since they are eigen-vectors form an orthonormal basis. Since we require all entries in D to be positive (see definition 1) then with appropriate normalization of X the expression XDX^T is symmetrical and also diagonally dominant and hence is positive definite. Therefore our generalized eigenvalue problem will only have real eigenvalues.

A valid question (raised by one of the reviewers) that we shall leave for future work is the relationship (if any) between our objective function given in equation 1 and the objective function shown below in equation 9. In this modified objective function A is a $r \times s$ matrix solved for all at once.

$$\arg\min_{A} \sum_{i,j} ||A^T x_i - A^T x_j||^2 w_{i,j} \quad (9)$$

2.1 Some Simple Illustrative Examples

To illustrate and verify the approach consider the example in Figure 1 which consists of four clusters of points each at a different corner of the cube. Throughout this section we use the basic constraint graph in definition 1 with $w_{i,j} = 1$ for similar instances and $w_{i,j} = -1$ for dissimilar instances. It should be noted that just like PCA it is left up to the user to determine how many dimensions to reduce the data to.

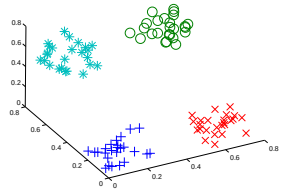

Figure 1: The original data in three dimensional space.

Figure 2 shows the performance of our approach for various constraint graphs where all the points of a given type (*,o,+,x) are labeled as similar/dissimilar to another type. In the left image we see that with two non-interacting constraints (Dissimilar(+,o) and Similar(x,*)) that a desirable result is achieved with the similar tagged points separating the dissimilar tagged points. In the middle image of Figure 2 we test the transitivity property of our approach since Similar(+,x),Similar(x,*) $\rightarrow$ Similar(+,*) and get a reasonable result, given the limitations of a linear transformation, where the 'x' points are surrounded by the '*' and '+' points. To obtain the ideal solution where all three point types are mapped to the same region would require a non-linear transformation of the space given the symmetry of the data.

However, our approach does have limitations. In the right image of Figure 2 we explicitly add Dissimilar(o,*) to Similar(+,x) and find that an undesirable results is obtained. By looking at Figure 1 we see that no linear and global transformation could satisfy both of these constraints.

A valid question is how the transformation progresses as the number of edges increases. Figure 3 captures the progressive transformation as more edges are added. We see that initially only the points around the single pair of 'x' and '+' constrained points overlap but as the number of constraints increase so does the amount of overlap until the two subpopulations overlap completely after the introduction of ten edges.

3 Creating the Constraint Graph

When creating the constraint graph, it helps if the weights are envisioned as penalties that are charged if the constraints are not well satisfied. Any manner of methods of creating constraint graphs could be used so long as the following holds:

1. A positive penalty means the points should be close together, a negative penalty far apart and no penalty meaning the points are unconstrained.
2. To help ensure real solutions to our generalized eigenvalue problem, the sum of penalties on a single point must be greater than zero.
3. The constraints should be consistent and give rise to a feasible clustering [Davidson and Ravi 2007], otherwise the results may be meaningless. In situations where constraints are generated solely from the ground truth (such as labels) the constraints generated will be consistent.

The simplest way of creating the constraint graph is by initializing the matrix W to the identity matrix (i.e. every point is most similar to itself) and then adding in a

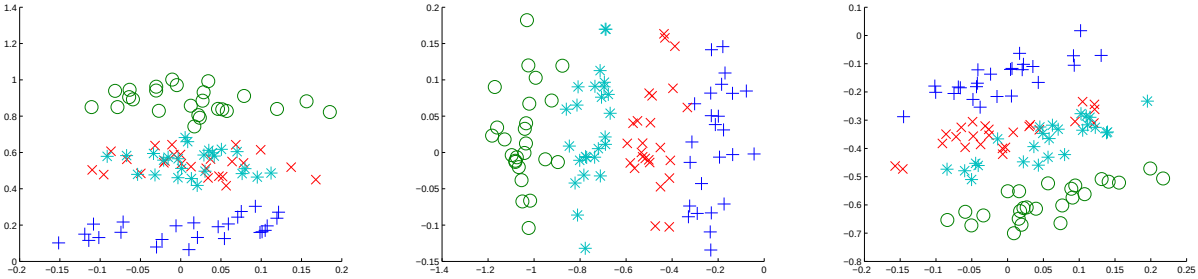

Figure 2: The transformed data in 2D space with constraints Dissimilar(+,o) & Similar(x,*) (left), Similar(x,+) & Similar(x,*) (middle) and Similar(+,x) & Dissimilar(o,*) (right).

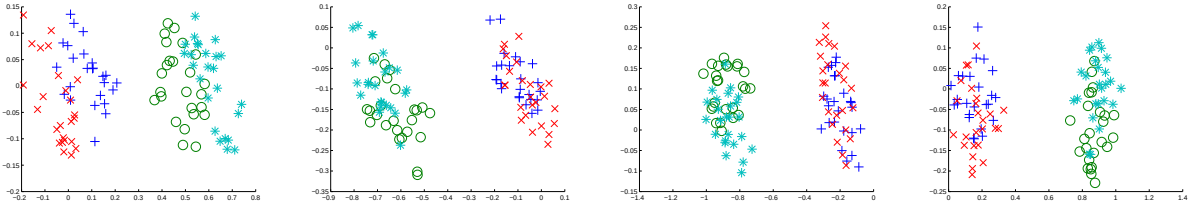

Figure 3: The transformed data in 2D space with constraints Similar(x,+) for 1,3,5 and 10 constraints going left to right. Note the more constraints the more '+' and 'x' overlap and align.

$w_{i,j} = +1$ if i and j are similar and -1 if they are dissimilar. In addition we modify the constraint graph in two ways to maintain local geometry and maintain consistency. In addition to the constraints embedded as weights, we add to each entry $w_{i,j}$ the amount $\frac{1}{k}$ if j is one of the $k = 5$ nearest neighbors of i to preserve the local geometry. We also propagate constraints due to transitivity and entailment in the graph. Transitivity is simply $Similar(x,y), Similar(y,z) \rightarrow Similar(x,z)$ and entailment $Similar(a,b), Similar(x,y), Dissimilar(a,x) \rightarrow Dissimilar(a,y), Dissimilar(b,x), Dissimilar(b,y)$.

We use the above constraint graph creation approach throughout this paper, but note that more complex approaches may be warranted if more domain knowledge exists.

4 Experimental Results - UCI

In this section we artificially create a situation that many practitioners face: The data contains useful features that can be used for clustering but many additional superfluous columns are present and it is difficult to separate out apriori the useful and superfluous columns.

To recreate this problem we take UCI data sets which contain useful features and add many randomly generated features so that clustering in the enlarged space yields poor results. To achieve this we take the following data sets with number of extrinsic labels (k), instances (n) and dimensions (m) in parentheses, Iris($k = 3$, $n = 150$, $m = 4$), Wine($k = 3$, $n = 178$, $m = 13$), Pima($k = 2$, $n = 768$, $m = 8$), Ionsphere($k = 2$, $n = 351$, $m = 34$), Glass($k = 6$, $n = 214$, $m = 10$) and Protein-Yeast($k = 6$, $n = 1484$, $m = 8$) and add in twenty columns of uniformly-distributed random numbers. We take twenty data points and use their labels to generate all possible entries in the graph (i.e. all possible pairwise constraints). If two points have the same label a similar-edge is generated between them otherwise a dissimilar-edge is generated. For these edges the corresponding must-link and cannot-link constraints are generated so as to compare results against constrained clustering algorithms. We cluster the data for k equaling the number of extrinsic labels. We then try several approaches. Firstly, we cluster the data in the enlarged space using regular k-means and COP-k-means algorithms. Next we perform metric learning using Xing et al's approach [Xing et. al. 2002] and cluster with k-means. Finally, we perform a variety of dimension reduction techniques including our own: PCA+k-means, PCA+COP-k-means and SSDR (see related work in section 6). Supervised dimension reduction approaches such as LDA are not applicable as only twenty data points are labeled and these approaches do not fair well in such problems. We apply each algorithm to 100 generated constraint sets and randomly restart each clustering algorithm 100 times, setting k to be the number off extrinsic labels. We report in Table 1 the average accuracy (Rand index) each obtained when measured on the instance labels but have scaled the results so that a value of 0.5 is the performance of guessing the most popular class.

As expected the base-line k-means algorithm performs poorly, often obtaining results only slightly better than always guessing the most popular class. This is to be expected since the algorithm assumes all dimensions are important and does no implicit features selection. Similar results are obtained by COP-k-means which though having the benefits of the constraints must **satisfy them and simultaneously find a useful clustering** in the higher-dimensional space. The worst performing approach is metric learning [Xing et. al. 2002] which perform worse than regular k-means. This is to be expected since the objective function of this and other metric learning algorithms do not explicitly try to find lower dimensional

spaces. Also, when learning a full metric their approach has no closed form solution and maybe converging to a poor result. The performance of the PCA dimension reduction algorithm with k-means is a mixture of hit and miss with respect to performance improvement over regular k-means as is the addition of PCA to constrained COP-k-means clustering. This is to be expected as the objective function of PCA attempts to find the projection that maximizes the variance which is most likely associated with the columns with random data. With the exception of the Ionsphere data set (which others have reported show no accuracy improvement with the addition of constraints) the GCDR-LP algorithm outperforms all other algorithms. This is not only indicative of the algorithm's performance but the general method of using hints/constraints for dimension reduction and then performing clustering for this type of problem.

5 Experimental Results - Images

In a second type of problem typically faced by practitioners, the available data is very high dimensional, but there are no nuisance columns. Instead the clusters are more easily identifiable in a lower dimensional space. This problem is common when dealing with data such as images, video and audio. We take the CMU faces data set [Mitchell 1997] which consists of controlled portrait images and cluster the data for $k = 2$. We measure performance and obtain edges/constraints using a variety of labels including gender:{female,male}, facial orientation:{up,down} and facial features:{glasses,no-glasses}.

We compare our approach against the eigen-faces approach which is a standard method of performing dimension reduction on facial images. The eigen-face approach calculates a huge $m \times m$ covariance matrix where m is the number of pixels in the image and then finds the eigen-vectors of this matrix and in doing so projects the data along the dimension of most variance as per PCA. Note that both approaches require the calculation of eigen-vectors, however, the PCA and eigen-face approaches require the additional step of calculating the covariance matrix.

The experimental results are shown in Table 2. The data sets were sampled so that there were equal number of each class. For each problem 100 similar-edges and 100 dissimilar-edge constraints were generated. As we can see with no dimension reduction the k-means algorithm performs no better than random guessing. The eigen-faces approach performed significantly better as has been reported previously, this is so since the images are controlled for light and distance and hence the eigen-vector approach chooses the pixels that are most variable/different/informative across the different images. Conversely, our approach uses only the constraint-graph to perform the dimension reduction by mapping similar images close together and dis-similar images far apart and given these hints are obtained from the ground truth are useful for improving clustering accuracy. Given the aims of both approaches are orthogonal, a valid question is: "Can the two approaches be combined?" To explore this question we first performed eigen-faces on the data and then GCDR-LP on the already reduced data sets. Performance results are promising as the last column in Table 2 indicates and the combination of approaches seems reasonable. Eigen-faces finds the most discriminating points and GCDR-LP finds the subset of those that are most useful for satisfying the constraints.

6 Related and Future Work

There have been several attempts to perform semi-supervised dimension reduction. Bar-Hillel and collaborators [Bar-Hillel et. al. 2005] add an intermediate step for Relevant Component Analysis but their work is only limited to must-link/similar constraints. The work of Tang and Zhong [Tang and Zhong 2006] and Zhang et al [Zhang et. al. 2007] use an objective function similar to that of Xing et' al [Xing et. al. 2002]. Their objective function sums (in the lower dimensional space) the distances between each pair of cannot-linked points less the sum of the distances between each pair of must-linked points and attempt to maximize this function. However, there approach has several limitations. Firstly, by not modeling the constraint graph, all constraints are created as equally important which may be undesirable. Similarly, in the work of [Zhang et. al. 2007] all unconstrained points are treated equally meaning that the algorithm will attempt to preserve the mapping between the distances between all pairwise combinations of unconstrained points. In our formulation the introduction of the constraint graph allows us the flexibility to model constraints of different importance and clearly emphasize what local geometry is important. We saw that in Table 1 that this additional flexibility translated into a significant improvement in performance over the SSDR approach of Zhang et al [Zhang et. al. 2007] that extends the work of Tang and Zhong [Tang and Zhong 2006].

Our work has the benefit of being a linear transformation and a logical next step is to explore non-linear transformation that make use of constraints for dimension reduction. Though there exists well understood and mature work for non-linear dimension reduction [Roweis and Saul 2000], it is not straight-forward to extend this work for constraint-graphs. In particular, in these approaches the reduced space only defines distances between points **in the training set**, which poses problems since typically the number of constrained points is very small and a subset of all points available. Furthermore, these approaches rarely have closed form solutions as ours does and hence will not scale well for the large amounts of data found in mining tasks. Finally, it would also be interesting to determine if our form of dimension reduction is useful for classification algorithms.

7 Conclusion

We propose the graph-driven constrained dimension reduction by linear projection (GCDR-LP) approach that given a weighted graph attempts to find a series of dimensions that are linear combinations of the old dimensions. The objective function of our approach essentially tries to find a low dimensional space that makes the points/nodes in the graph with a positive edge-weight closer together and those with a negative edge-weight further apart. The constraint graph can be created in any number of ways and we explored also having additional entries for each instances k-nearest neighbors so as to

Dataset	k-means	COP-k-means	Xing+k-means	PKM	PCA+ k-means	PCA+ COP-k-means	SSDR+ k-means	GCDR-LP+ k-means
Iris	58%	54%	48%	49%	51%	59%	59%	**68%**
Wine	54%	49%	45%	46%	46%	57%	55%	**61%**
Pima	53%	52%	51%	53%	55%	52%	54%	**59%**
Ionsphere	61%	58%	53%	52%	62%	59%	58%	60%
Glass	63%	64%	59%	58%	59%	62%	61%	**66%**
Protein	59%	55%	53%	55%	60%	58%	59%	**68%**

Table 1: Results of applying a variety of algorithms to UCI data sets with 20 columns of random noise added and 20 similar and dissimilar constraints/edges. The first four techniques cluster in the higher dimensional space, the latter four reduce the dimensionality to the original number of dimensions and then perform clustering. Results are averaged over 100 constraint sets and randomly restarting the clustering algorithm 100 times for each. Results in bold show statistically significant better results than next best technique using a student pair-wise t-test at 95% CI.

Data-set	k-means	Eigen-faces k-means	Eigen-faces COP-k-means	GCDR-LP +k-means	Eigen-faces-then-GCDR-LP+k-means
Female/Male	51%	65%	62%	70%	**73%**
Up/Down	52%	66%	63%	73%	**76%**
Sunglasses/Not	54%	70%	66%	78%	**83%**

Table 2: Results of applying a variety of algorithms to CMU Faces data sets of 128x128 pixels using 100 similar and dissimilar constraints/edges. Results are averaged over 100 sets of constraints/edges and 100 random restarts of the clustering algorithm. Results in bold show statistically significant better results than next best technique using a student pair-wise t-test at 95% CI.

maintain the underlying local geometry. Our problem formulation is easily solved as a generalized eigen-value problem which is implementable in MATLAB and has a closed form solution. This has advantages over metric learning techniques that do not perform dimension reduction or have closed form solutions and hence may converge to a poor solution.

After the transformation any number of algorithms could be used to cluster the data and in this work we explored k-means and have also used agglomerative hierarchical clustering (results not shown). We show that our approach is useful for performing dimension reduction to help non-hierarchical clustering algorithms such as k-means which outperforms k-means, COP-k-means, PCA+k-means, PCA+COP-k-means, metric learning approach (Xing et' al [Xing et. al. 2002])+k-means, PKM [Basu et. al. 2004] and SSDM [Zhang et. al. 2007]. This result not only shows the utility of our algorithm but the general approach of separating the constraint satisfaction and clustering problems. For the CMU faces database we show the approach of using constraints for dimension reduction produces better results than eigen-faces and can be used in conjunction with eigen-faces to obtain even better results.

8 Acknowledgments

The author thanks the anonymous reviewers for their excellent comments and the NSF for support of this work via GRANT IIS-0801528 CAREER:Knowledge Enhanced Clustering with Constraints.

References

[Bar-Hillel et. al. 2005] A. Bar-Hillel, T. Hertz, N. Shental, D. Weinshall, Learning a Mahalanobis Metric from Equivalence Constraints,*JMLR* 6:937-965, 2005

[Basu et. al. 2004] S. Basu, M. Bilenko and R. J. Mooney, Active Semi-Supervision for Pairwise Constrained Clustering, 4^{th} *SIAM DM Conference*, 2004.

[Coleman et. al. 2008] T. Coleman, J. Saunderson and A. Wirth, Spectral Clustering with Inconsistent Advice, International Conference on Machine Learning, 2008.

[Davidson and Ravi 2007] I. Davidson and S. S. Ravi, "The Complexity of Non-Hierarchical Clustering with Instance and Cluster Level Constraints", *Data Mining and Knowledge Discovery*, Vol. 14, No. 1, Feb. 2007.

[Mitchell 1997] T. Mitchell, *Machine Learning*, McGraw Hill, 1997.

[Roweis and Saul 2000] S. Roweis, L. K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, *Science*, vol 290, 22 December 2000.

[Tang and Zhong 2006] W. Tang and S. Zhong, Pairwise Constraints-Guided Dimensionality Reduction", SIAM DM Workshop FSDM'06, 2006.

[Wagstaff et. al. 2001] K. Wagstaff, C. Cardie, S. Rogers and S. Schroedl, "Constrained K-means Clustering with Background Knowledge", ICML, 2001.

[Xing et. al. 2002] E. Xing et. al., Distance metric learning with application to clustering with side-information, NIPS 15, 2002.

[Zhang et. al. 2007] D. Zhang et. al., Semi-supervised dimensionality reduction, *SIAM DM Conference* 2007.