An Efficient Algorithm for Learning Distances that Obey the Triangle Inequality

Arijit Biswas
http://www.xrci.xerox.com/profile-main/67
Xerox Research Centre India
Bangalore, India

David Jacobs
http://www.cs.umd.edu/~djacobs/
Computer Science Department
University of Maryland
College Park, USA

Abstract

Semi-supervised clustering improves performance using constraints that indicate if two images belong to the same category or not. Success depends on how effectively these constraints can be propagated to the unsupervised data. Many algorithms use these constraints to learn Euclidean distances in a vector space. However, distances between images are often computed using classifiers or combinatorial algorithms that make distance learning difficult. In such a setting, we propose to use the triangle inequality to propagate constraints to unsupervised data. First, we formulate distance learning as a metric nearness problem where a brute-force Quadratic Program (QP) is used to modify the distances such that the total change in distances is minimized but the final distances obey the triangle inequality. Then we propose a much faster version of the QP that enforces only a subset of the inequalities and can be applied to real world clustering datasets. We show experimentally that this efficient QP produces stronger clustering results on face, leaf and video image datasets, outperforming state-of-the-art methods for constrained clustering. To gain insight into the effectiveness of this algorithm, we analyze a special case of the semi-supervised clustering problem, and show that the subset of constraints that we sample still preserves key properties of the distances that would be produced by enforcing all constraints.

1 Introduction

Semi-supervised clustering [3] of images has been an interesting problem for machine learning and computer vision researchers for decades. Pairwise constrained clustering [9, 27, 33] is a popular paradigm for semi-supervision that uses knowledge about whether two images belong to the same category (must-link constraint) or not (can't-link constraint). Performance of constrained clustering algorithms can be improved if the supervision on some image pairs is used to modify the pairwise distances of other image pairs, for which no supervision is available.

As an example of a semi-supervised problem, suppose we have access to surveillance video data from an airport and we would like to group the face images based on identity. This makes it easier to scan the people who have been present in the airport. Automatic clustering of these images is extremely hard because of the large number of clusters and large variation in pose, expression and lighting present in these faces. However we know that faces that are tracked through the video should be faces of the same person, giving

Pages 10.1-10.13
DOI: https://dx.doi.org/10.5244/C.29.10

rise to many useful must-link constraints. Also, if two face tracks appear together in some frames, then faces in these two tracks must be of different persons, even if they look very similar, providing several can't-link constraints. These constraints can be used to improve face clustering in videos [13, 31, 33].

There are several excellent metric learning approaches when the image distances can be represented as Euclidean distances in vector spaces [15, 34]. However, in many cases distances are computed robustly [17, 21], on manifolds [19, 28, 30] or as the output of algorithms [8, 25] or classifiers [6]. These do not lead to a natural embedding in a metric space, and may not even obey the triangle inequality. We are particularly interested in semi-supervised clustering of fine-grained categories. If we look at the top 10 distance measures in two important domains (LFW faces [1] and leaf shapes [2]) related to fine-grained classification, we find that 80% of the methods use non-vector space distances. These distances can be used for clustering images, but are not suitable for existing metric learning algorithms.

In order to propagate constraints from supervised to unsupervised pairs, some structure must be assumed on the set of possible distances. Otherwise, the distance between supervised pairs could be altered without affecting the distance between unsupervised pairs. Perhaps the weakest assumption that we can make about a distance is that it obeys the triangle inequality. Enforcing the triangle inequality allows us to propagate constraints; if a constraint alters one distance, other distances must also change to maintain triangle inequalities. For many interesting distances, the triangle inequality is not guaranteed to hold. However, we empirically find that the triangle inequality almost always holds for distances computed for fine-grained classification even when not explicitly enforced. For example, with the distances we use in Pubfig [6, 24] faces and Leafsnap [23], 99.98% and 97.48% of the triangle inequality constraints hold respectively. This strongly motivates us to enforce the triangle inequalities when we alter distances to incorporate the pairwise constraints. We then find empirically that by enforcing the triangle inequality we can improve performance on several real world datasets.

Our main contribution is to formulate distance learning with pairwise constraints as a metric nearness problem[1] [11, 16] and then provide an efficient algorithm to solve metric nearness for clustering. First, we formulate a quadratic optimization problem, where the pairwise distances between images are modified such that pairwise constraints and triangle inequality constraints are satisfied as much as possible. Since enforcing $O(N^3)$ triangle inequalities is computationally expensive, we propose a graph based approach, where only $O(n(M+C))$ triangle inequalities are sampled for use in the QP (N is the total number of images, n is the number of nearest neighbors in the n-nearest neighbor graph, M and C are the number of must-link and can't-link constraints respectively). We empirically show that this sampling approach works well in practice. We use the distances obtained by our approach along with a constrained clustering algorithm [32] to achieve state-of-the-art clustering results. The proposed approach is a general framework to learn distances using any kind of pairwise distances between all the images to be clustered and does not require any vector space representation of images.

We theoretically analyze a simplified case in which only one pairwise constraint is present, to gain insights into our fast approach. Our sampling approach is based on the intuition that clustering is predominately affected by small distances, and is not sensitive to the exact value of larger distances. We prove that our sampling approach produces the same set of small distances that would be obtained by enforcing all constraints.

[1] Given a dissimilarity matrix, find the "nearest" matrix of distances that satisfy the triangle inequalities.

We perform experiments on leaf and face image datasets and show that distances obtained by our method achieve state-of-the-art clustering results. We also run experiments on a real world video dataset, where extracted faces are clustered based on identity. Our approach outperforms recent constrained clustering methods on this video dataset.

2 Related Work

Our work draws on prior work in constrained clustering and metric learning. One of the earliest constrained clustering algorithms was proposed by Wagstaff et al [32] in 2001. The authors in [32] proposed an algorithm in which points are assigned to clusters such that none of the specified constraints are violated. The authors in [5] proposed a method in which a Hidden Markov Random Field (HMRF) is formulated along with must-link and can't-link constraints and MAP estimation is performed on the HMRF. The authors in [4] proposed a soft constrained clustering algorithm, where the cost function for clustering takes care of the pairwise constraints. However this approach is more suited for the active constrained clustering setup proposed in [4], where constraints are obtained in an interactive manner. The authors in [26, 27] proposed spectral clustering methods where pairwise constraints are propagated to the entire dataset. However spectral methods usually require vector representation of the images for clustering. In [22] the authors proposed a method for constrained clustering where the must-links are propagated between other image pairs, by computing the shortest path between all pairs. A straight-forward modification to the Floyd-Warshall algorithm [14] is used to perform the fast all-pair-shortest-path computation.

There is a significant amount of past research work on metric learning (see [7]). However much of this work is directed at the case in which objects are embedded in a vector space [15, 34]. Perhaps the most similar past work is on non-parametric kernel learning methods [20, 35], where a kernel matrix is learned from the similarity matrix and user provided constraints by using semi-definite programming. However we demonstrate in the experimental results that such approaches do not work well for image datasets, especially when there are many clusters.

Recently several researchers have applied constrained clustering techniques to faces in videos. In [13], the authors proposed a method for metric learning from constraints and used that for face identification. The authors in [31] also proposed an approach to cluster face images with pairwise constraints for movie content analysis. In the most recent work on face clustering [33], a method was proposed where a Hidden Markov Random Field based objective function is formulated that incorporates the pairwise constraints; it is optimized using the simulated field algorithm [12].

Although there has been past research work on metric nearness [11, 16] they do not take pairwise constraints into account and are not scalable to real world clustering datasets. The authors in [11, 16] propose a triangle fixing algorithm to obtain a globally optimal solution. Their algorithm iterates through the triangle inequalities, optimally enforcing any inequality that is not satisfied. Although this method gives an exact solution for 100 points in around 15 seconds, it takes 3 minutes, 50 minutes and 6 hours (scales as $O(N^3)$) for datasets of size 200, 500 and 1000 points respectively with a C++ implementation [11]. That prevents us from applying this method to real world clustering datasets of size 1000 or more images and necessitates the development of an efficient method for triangle inequality enforcement.

3 Our Approach

Semi-supervised clustering with pairwise constraints is most useful when the constraints are propagated to other image pairs as well. In this paper we propose an approach for learning

new distances using only the initial pairwise distances between images. We formulate a quadratic program to modify the distances such that total distance modification is minimized but the final distances obey the triangle inequality. We define the ideal scenario QP in Section 3.1 in which all constraints are enforced and discuss the high computation cost for solving the ideal case. Next, in Section 3.2 we propose a graph based QP formulation which is much faster and applicable to larger datasets in practice. In Section 3.3, we theoretically demonstrate why the graph based approach is a good alternative to the brute-force method.

3.1 Problem Formulation

We begin with a set of N unlabeled images $\mathcal{U}$ ($\boldsymbol{x} \in \mathcal{U}$) from K classes. We are also provided with initial distances between all the image pairs, i.e., $d_I(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is given $\forall i, j$ (note that d_I denotes initial distance). However we do not have any vector representation of the images. We have access to a set of must-link and can't-link constraints. Let $\mathcal{M}$ denote the set of must-link constraints such that any pair $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{M}$ implies that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to the same class. Similarly $\mathcal{C}$ denotes the set of can't-link constraints such that any pair $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{C}$ implies that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to different classes.

We learn a new set of distances for all the image pairs given the pairwise constraints $\mathcal{M}$ and $\mathcal{C}$. A quadratic optimization problem is formulated to find these distances. Let us assume that the set of final distances are given by $d_F(\boldsymbol{x}_i, \boldsymbol{x}_j)$, which we can obtain by solving the following quadratic optimization problem:

$$
\begin{aligned}
\underset{d_F}{\text{minimize}} \quad & \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \notin \mathcal{M} \cup \mathcal{C}} \left(d_F(\boldsymbol{x}_i, \boldsymbol{x}_j) - d_I(\boldsymbol{x}_i, \boldsymbol{x}_j)\right)^2 \\
\text{subject to} \quad & (i) \;\; d_F(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq U, \;\; (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{M} \\
& (ii) \;\; d_F(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq L, \;\; (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{C} \\
& (iii) \;\; d_F(\boldsymbol{x}_i, \boldsymbol{x}_j) + d_F(\boldsymbol{x}_j, \boldsymbol{x}_k) \geq d_F(\boldsymbol{x}_i, \boldsymbol{x}_k), \forall i, j, k \\
& (iv) \;\; d_F(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0, \forall i, j
\end{aligned}
\tag{1}
$$

In the formulation in Eq. 1, distances between all image pairs are modified such that the constraints (i) to (iv) are satisfied. The objective function minimizes the total sum of changes in pairwise distances. Constraint (i) causes distances corresponding to must-link constraints to be reduced so they are upper-bounded by U, a user defined constant. Similarly, constraints in (ii) move can't-link image pairs as far as possible. The triangle inequality constraints are added in (iii). These triangle inequality constraints propagate information about must-link and can't-link constraints to other image pairs. The final set of constraints in (iv) ensure that all the quadratic optimization variables, i.e., the pairwise distances, remain non-negative. This formulation is similar to [□] except that we also take pairwise constraints into account.

In this QP formulation, there are large number of QP variables ($O(N^2)$) and triangle inequality constraints ($O(N^3)$). Also with all the pairwise and triangle inequality constraints there may not exist a feasible solution. To avoid these issues we reduce the size of the QP significantly by determining which triangle inequalities are crucial for clustering using a novel graph-based formulation.

3.2 Graph Based QP (DistLQP)

In this section we describe an efficient approach, based on enforcing a subset of the constraints in equation 1. We make use of the intuition that when a dataset is clustered, comparatively smaller distances usually determine the clusters. When two points are far apart, the exact distance between them is not critical in determining the clusters. We can build a nearest neighbor graph to determine which distances are small and possibly important for

clustering. Distances smaller than or equal to a fixed threshold t_h are called nearest neighbors or "small" and distances larger than t_h are called "large". We focus our method on accurately determining the small distances, while allowing some inaccuracy in the large distances. Although we use a hard threshold t_h for proofs, in practice we find it more reasonable to adapt the threshold locally. So we use an n-nearest neighbor graph, in which the parameter n is selected depending upon the problem domain. We present our algorithm in this section and analyze some of its properties in Section 3.3.

First, a n-nearest neighbor graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is created from the N images, where each node (image) is connected by edges to its n nearest neighbors. We also add edges when there is a must-link or can't-link constraint between two images but the corresponding distances are not lower than U or more than L respectively. If two images are already very close and must-linked, it is not required to modify the corresponding distance. Similarly, if two images are very far and are can't-linked, we do not need to change the corresponding distances. $\mathcal{E}$ denotes the nearest neighbor edges and must-link/can't-link edges of the graph.

One of our chief contributions is to determine which triangle inequalities out of $O(N^3)$ possible inequalities should be enforced in the QP. We only enforce triangle inequalities for triplets that contain at least one nearest neighbor edge and at least one must-link/can't-link edge. This reduces the number of triangle inequalities to $O(n(M+C))$. In Section 3.3, we prove that under some simplified assumptions even with this small subset of triangle inequalities we obtain distances with the same key properties as those obtained by enforcing all the constraints. Now $\mathcal{E}$ is augmented to make $\mathcal{E}' = \mathcal{E} \cup \mathcal{E}_{\mathcal{A}}$, where $\mathcal{E}_{\mathcal{A}}$ includes edges between two nodes that are connected to a third node with one nearest neighbor edge and one must-link/can't-link edge in $\mathcal{E}$. All edges in $\mathcal{E}'$ are used as the variables of the QP.

We also add slack variables to the must-link and can't-link constraints in the QP such that a feasible solution can be obtained even when all the pairwise constraints are not satisfied. Slack variables could also be added to the triangle inequality constraints if required, however that would increase the number of QP variables significantly. We have not found this to be necessary. Our final QP (with the graph-based formulation and the slack variables for pairwise constraints) which we optimize is provided below ($d'_F(\boldsymbol{x}_i, \boldsymbol{x}_j)$ denotes the final distances obtained from the following QP):

$$
\begin{aligned}
\underset{d'_F, \xi_m, \xi_c}{\text{minimize}} \quad & \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{E}' - \mathcal{M} \cup \mathcal{C}} (d'_F(\boldsymbol{x}_i, \boldsymbol{x}_j) - d_I(\boldsymbol{x}_i, \boldsymbol{x}_j))^2 + \lambda_1 \sum_m \xi_m + \lambda_2 \sum_c \xi_c \\
\text{subject to } & (i)\ d'_F(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq U + \xi_m, \quad (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{M} \cap \mathcal{E}' \\
& (ii)\ d'_F(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq L - \xi_c, \quad (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{C} \cap \mathcal{E}' \\
& (iii)\ d'_F(\boldsymbol{x}_i, \boldsymbol{x}_j) + d'_F(\boldsymbol{x}_j, \boldsymbol{x}_k) \geq d'_F(\boldsymbol{x}_i, \boldsymbol{x}_k), \forall i, j, k \\
\text{s.t. } & E = \{(\boldsymbol{x}_i, \boldsymbol{x}_j), (\boldsymbol{x}_i, \boldsymbol{x}_k), (\boldsymbol{x}_j, \boldsymbol{x}_k)\} \subset \mathcal{E}' \text{and } \exists\ e_p, e_q \in E \text{ s.t. } e_p \neq e_q, e_p \in \mathcal{M} \cup \mathcal{C} \text{ and } e_q \in \mathcal{E} \\
& (iv)\ d'_F(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0, \forall i, j,\ \xi_m \geq 0, \forall m, \xi_c \geq 0, \forall c
\end{aligned}
\tag{2}
$$

where variables ξ_m and ξ_c refer to the slack variables corresponding to the must-link and can't-link constraints respectively. λ_1 and λ_2 are user defined constants. Eq. 2 gives a standard convex problem, which we solve using the interior point method for quadratic optimization. We will refer to our proposed approach for distance learning as DistLQP (**Dist**ance **L**earning with **Q**uadratic **P**rogramming) from now on.

In Figure 1a, we consider a simple 2-d example with six points $\boldsymbol{x}_i$, where $i = 1, .., 6$. The nearest neighbor edges are shown in black, the must-link edge is shown in green and the can't link edge is shown in red. In this scenario, distances $d(\boldsymbol{x}_1, \boldsymbol{x}_3)$, $d(\boldsymbol{x}_2, \boldsymbol{x}_3)$, $d(\boldsymbol{x}_3, \boldsymbol{x}_4)$, $d(\boldsymbol{x}_4, \boldsymbol{x}_5)$, $d(\boldsymbol{x}_5, \boldsymbol{x}_6)$, $d(\boldsymbol{x}_1, \boldsymbol{x}_2)$, $d(\boldsymbol{x}_2, \boldsymbol{x}_4)$, $d(\boldsymbol{x}_3, \boldsymbol{x}_5)$, $d(\boldsymbol{x}_4, \boldsymbol{x}_6)$ are used as variables in the QP. Triangles

(a) (b)

Figure 1: (a) A simple example in 2-d Euclidean space (green denotes must-link edge and red denotes can't-link edge) to explain our proposed approach. (b) Some example leaf and face images.

$\Delta x_1x_2x_3$, $\Delta x_2x_3x_4$, $\Delta x_3x_4x_5$ and $\Delta x_4x_5x_6$ are used for the triangle inequality constraints.

We note that our approach is developed for clustering images, in which all distances and constraints are available when we learn the new distances. The out of sample problem is also of interest, in which a query image retrieves its nearest neighbors using the learned distances. We have performed preliminary experiments using our approach to learn distances for retrieval, but have not found that it significantly improves performance. This may be an interesting topic for future research.

3.3 Theoretical Analysis

Although our graph based approach is much faster than the brute-force version and empirically found to be better than the state-of-the-art approaches, we theoretically analyze a simple scenario to obtain further insight into its properties. We consider the case in which one constraint is added to a set of distances that obey the triangle inequality (as mentioned in the introduction, the distances we use generally do obey the triangle inequality). We state a theorem in this section (proof in supplementary material) which shows that in this case, even with the subset of triangle inequalities we enforce, key properties of the brute-force QP solution are preserved.

Note that $d_I(\boldsymbol{x}_i,\boldsymbol{x}_j)$ are the initial distances, $d_F(\boldsymbol{x}_i,\boldsymbol{x}_j)$ are the distances obtained by solving the QP in equation 1 and $d'_F(\boldsymbol{x}_i,\boldsymbol{x}_j)$ are the distances obtained by solving the QP in equation 2. Due to space constraints we provide all proofs of our lemmas and the theorem in the supplementary material.

Now we state a theorem and show that even with the triangle inequality constraints that we enforce in QP in equation 2, distances are changed in a similar way as they would have, had we enforced all the triangle inequality constraints (as in equation 1). We note that distances that are smaller than or equal to a fixed threshold t_h are called nearest neighbors or "small" and distances larger than t_h are called "large" for all d_I, d_F and d'_F.

Theorem 1. *If there are N points* $\{\boldsymbol{x}_1,\boldsymbol{x}_2,...,\boldsymbol{x}_N\}$ *with a must-link/can't-link constraint present among them, for any fixed threshold* t_h *and for any pair* $\{\boldsymbol{x}_i,\boldsymbol{x}_k\}$,

I. if $d_I(\boldsymbol{x}_i,\boldsymbol{x}_k) > t_h$ and $d_F(\boldsymbol{x}_i,\boldsymbol{x}_k) > t_h$, then $d'_F(\boldsymbol{x}_i,\boldsymbol{x}_k) > t_h$.
II. if $d_I(\boldsymbol{x}_i,\boldsymbol{x}_k) > t_h$ and $d_F(\boldsymbol{x}_i,\boldsymbol{x}_k) \leq t_h$, then $d'_F(\boldsymbol{x}_i,\boldsymbol{x}_k) = d_F(\boldsymbol{x}_i,\boldsymbol{x}_k)$.
III. if $d_I(\boldsymbol{x}_i,\boldsymbol{x}_k) \leq t_h$ and $d_F(\boldsymbol{x}_i,\boldsymbol{x}_k) > t_h$, then $d'_F(\boldsymbol{x}_i,\boldsymbol{x}_k) = d_F(\boldsymbol{x}_i,\boldsymbol{x}_k)$.
IV. if $d_I(\boldsymbol{x}_i,\boldsymbol{x}_k) \leq t_h$ and $d_F(\boldsymbol{x}_i,\boldsymbol{x}_k) \leq t_h$, then $d'_F(\boldsymbol{x}_i,\boldsymbol{x}_k) = d_F(\boldsymbol{x}_i,\boldsymbol{x}_k)$.

We prove that when one must-link or can't-link constraint is added distances learned using only a subset of triangle inequalities will be exactly the same as the distances learned by the brute-force approach except when a large ($> t_h$) distance remains large using the brute-force approach. Since large distances are unlikely to affect clustering, we prove that if a large distance remains large using the brute-force approach, it will also be large using our fast approach but we are not required to compute the exact change in those distances.

We note that using our method we can modify a maximum of $4n$ distances with each pairwise constraint, which is much higher than the one pairwise distance modification when no constraint propagation happens. If there are a total of $(M+C)$ pairwise constraints available, $4(M+C)n$ pairwise distances may get modified.

4 Experimental Results

We run experiments on three different domains, leaves, faces and detected faces from video images. We describe the datasets below:

- **Leaf-1042 [9]** (Figure 1b): This data set contains 1042 leaf images from 62 classes. Chi-square square distances [9] between curvature histograms are used as the initial distances for our algorithm.
- **Face-1000 & Face-10000 [24]** (Figure 1b): These datasets are subsets of the Pubfig dataset. The complete Pubfig dataset has around 58,797 celebrity images of 200 persons. Distances are calculated based on the output of a pre-trained classifier [6]. Face-1000 has 1000 images from 50 classes, with each class having 20 images. Face-10000 is a much larger dataset with 10000 images from 50 classes, where each class has 200 images.
- **BF0502-subset-1 and BF0502-subset-2 [18]**: These datasets contain extracted faces from videos. The original video dataset BF0502 [18] has 27,504 extracted faces of 11 main cast members and others from the TV series "Buffy the Vampire Slayer". BF0502-subset-1 has 687 faces of 6 persons and is exactly same as the BF0502-subset used in [33]. Since BF0502-subset-1 had only 6 clusters we created another subset called BF0502-subset-2 which contains 600 face images of 11 persons[2]. Unlike the other datasets, previous work on this video data has shown strong results using vector space representations. So one goal of these experiments is to provide a comparison to a state-of-the-art method on its home turf.

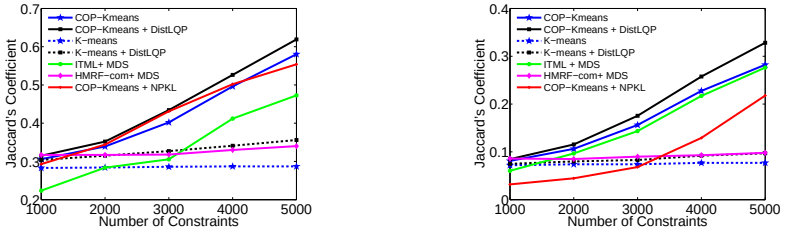
  Since our method does not require a vector space representation, we are free to make use of a non-vector space, robust distance on this data, which we find improves results. For each image pair, the squared difference is computed in each dimension. Then only the smallest 80% of those differences are summed and the square root is taken to obtain a robust distance. Note that while such robust distances are common in computer vision, they cannot be represented as a Euclidean distance in any vector space.

We describe the algorithms below that we compare in our experimental evaluation. We compare our approach to previous vector space methods by first using Multidimensional Scaling (MDS) [10] to embed the images in a vector space.

- **K-means [29][3]:** In this approach the distances corresponding to the must-link pairs are set to zero and can't-link pairs are set to a very high value. This version of K-means works better than the K-means without any distance modification.
- **K-means + DistLQP:** Distances learned from our approach are used along with the K-means clustering algorithm.
- **COP-Kmeans [32]:** This is a traditional constrained clustering algorithm, which avoids constraint violations.
- **COP-Kmeans + DistLQP:** Distances learned from our approach are used along with the constrained clustering algorithm COP-Kmeans.

[2] We will publish the details of all subsets of data sets used, to allow others to experiment with the same datasets.

[3] Since we do not have the vector representation of images we compute the medoid instead of the mean in the update step.



(a) Leaf-1042 (b) Face-1000

Figure 2: Comparison of our proposed distance learning approach with other constrained clustering methods.

K-means [29]	K-means+DistLQP	COP-Kmeans [32]	COP-Kmeans +DistLQP	ITML [15]+MDS
0.0784	0.0803	0.0818	**0.0831**	0.0711

Table 1: Results for Face-10000 (HMRF-com [33] and NPKL [20] were not fast enough for this dataset).

- **ITML [15] + MDS [10]:** MDS is used to project the distances back to a vector space and Information Theoretic Metric Learning (ITML) is used to learn distances.
- **HMRF-com [33] + MDS [10]:** MDS is used to project the distances back to vector space and HMRF-com is used for clustering.
- **COP-Kmeans + NPKL [20]:** Distances are learned using a non-parametric Kernel Learning (NPKL) approach and are used along with COP-Kmeans.

The major steps in our approaches K-means + DistLQP and COP-Kmeans + DistLQP are described now. First, we have the initial distances and the pairwise constraints. We set the distances corresponding to must-link to 0 and can't-link to a high value. We learn a new set of distances using our approach. Next, K-means or COP-kmeans is run with the new distances. Finally, we get a clustering and evaluate the clustering solution. In the baseline K-means and COP-Kmeans all of the above steps are used except where we learn a new set of distances using our approach.

We use Jaccard's coefficient [9], which varies from 0 to 1 (1 is the best), as the clustering evaluation metric for Leaf-1042, Face-1000 and Face-10000. For BF0502-subset-1 and BF0502-subset-2, we use the same evaluation metric as used by [33]. Each algorithm is run with 25 random initializations and with 5 different random sets of pairwise constraints for a fixed constraint set size. The average of all of them is reported. Since the video data has one fixed set of constraints, the average of 25 random initializations are reported. For Leaf-1042 and Face-1000, the total number of constraints is varied from 1000 to 5000, with 20% of the constraints being always must-link constraints. For Face-10000, one fixed constraint size of $25,000$ (20% must-link) is used. However with the video dataset BF0502-subset-1 we have a fixed set of 687 must-link constraints and 180 can't-link constraints. For BF0502-subset-2, we have 600 must-link and 189 can't-link constraints available. These constraints are produced following the same method as used by [33]. The nearest neighbor graph size n is set to a fixed value of 20 for all datasets. The must-link threshold U is always set to a low value of 0.01, whereas the can't-link threshold varies from one dataset to another depending on the maximum pairwise image distance. We set that to $L = \frac{max(d_I(\boldsymbol{x}_i,\boldsymbol{x}_j),\forall i,j)}{\alpha}$, where α is a parameter. α is set to be 8 for leaves, 3 for faces and 2 for the video subsets. λ_1 is set to be 10 and λ_2 is set to be 1 for all the datasets.

Algorithms	BF0502-subset-1	BF0502-subset-2
K-means [29]	0.3640	0.2578
K-means + DistLQP	0.3749	0.2635
COP-Kmeans [32]	0.3865	0.2603
COP-Kmeans + DistLQP	0.4035	**0.2695**
ITML [15]	0.3566	0.2177
HMRF-com [33]	**0.4576**	0.2633
COP-Kmeans+NPKL [20]	0.3223	0.2440

Table 2: Results for BF0502-subset-1 and BF0502-subset-2 (confusion matrix used in [33]).

4.1 Discussion of Results

Our approach outperforms all other algorithms for all datasets except for BF0502-subset-1, where HMRF-com uses the actual vector representation of the images, and only six clusters are present. HMRF-com usually works well with a small number of clusters and when the actual vector representation of the images are available. However when the number of clusters increases, this approach does not perform well because there are not usually enough samples from each class to learn the parameters of the Gaussian distributions used in their approach.

Also when we perform MDS to convert the distances to a vector space representation, the representation is often inaccurate. When these vectors are used along with HMRF-com and ITML, their performance degrades because of the poor vector space representations. We used 60 dimensional MDS for all the datasets except the video subsets, where we had access to the original vector space representations. We analyze the results of each dataset below:

- **Leaf-1042 (see Figure 2a):** Our approach outperforms all other algorithms by a significant margin. For example, with only $5,000$ pairwise constraints ($\sim 1\%$ of all possible pairwise constraints), our approach improves the Jaccard's Coefficient from 0.58 (with the second best approach) to 0.62 (relative improvement of 7%). The proposed method takes around 4 minutes (in matlab) with $5,000$ pairwise constraints to learn a new set of distances for this dataset.
- **Face-1000 (see Figure 2b):** In this dataset also our approach outperforms all other approaches by a significant margin. For example, when $5,000$ pairwise constraints (only $\sim 1\%$ of all possible constraints) are available, our approach improves the Jaccard's Coefficient from 0.28 (with the second best approach) to 0.33 (relative improvement of 18%). Our approach takes around 2 minutes to learn a new set of distances with $5,000$ pairwise constraints for this dataset.
- **Face-10000 (see Table 1):** Although the proposed approach improves clustering even for this dataset, it is still far from perfect. From the performance of all the algorithms, it seems that clustering such a large dataset is an extremely hard problem and it is probably hard to obtain a reasonable clustering with state-of-the-art image features. It takes around 40 minutes for our approach to learn the new distances with the $25,000$ constraints we use. However HMRF-com [33] and NPKL [20] were not fast enough that we could run these algorithms for Face-10000.
- **BF0502-subset-1 and BF0502-subset-2 (see Table 2):** For these two datasets we used the original vector space representation (after dimensionality reduction with Principal Component Analysis as suggested by [33]) for HMRF-com [33] and ITML [15]. We did not have to do Multidimensional scaling to obtain a vector space representation for

these datasets. Although for BF0502-subset-1, HMRF-com outperforms our approach by a significant margin, we note that we do not use the actual vector space representation in our approach. However for BF0502-subset-2, we outperform all other approaches, which demonstrates that HMRF-com's performance may be more sensitive to the number of clusters. For the video subsets, it took around 1.3 minutes to learn a new set of distances for clustering. We note that while our approach is not really designed for this scenario, in which a vector space representation of the images is available, it still performs very well, outperforming other approaches when the number of clusters is 11. In part, this is because our approach offers the freedom to make use of robust distances that deviate from the original vector space.

Since we formulate distance learning with pairwise constraints as a metric nearness problem, we compare the runtime of our approach with the past work [] on metric nearness. We obtain a new set of distances using $5,000$ pairwise constraints in about $2-4$ minutes for Leaf-1042 and Face-1000 using a matlab implementation. Whereas the previous metric nearness approach in [] takes as long as 6 hours for obtaining a new set of distances using one set of pairwise constraints (C++ implementation) and produces similar final clustering solutions (Jaccard's Coefficient of 0.623 compared to 0.62 using our method on Leaf-1042) for these datasets. We also run our approach on Face-10000 (a dataset of $10,000$ images) in around 40 minutes where the past approach can take several days to get the new distances. We note that a C++ implementation could further improve the runtime of our approach.

5 Conclusion

In this paper we propose a novel method for learning distances between images when pairwise must-link and can't-link constraints are provided. Our method uses only pairwise distances between all images and does not require the vector space representation of the images. We apply these learned distances for clustering images. We run experiments for leaf and face clustering. We also use our approach for clustering faces extracted from videos. We demonstrate that our approach outperforms other approaches and produces state-of-the-art clustering results. Although the experimental results are shown for images, the proposed approach can be applied to any other domain.

References

[1] http://vis-www.cs.umass.edu/lfw/.

[2] http://www.dabi.temple.edu/ shape/mpeg7/results.html.

[3] Eric Bair. Semi-supervised clustering methods. *CoRR*, abs/1307.0252, 2013. URL http://arxiv.org/abs/1307.0252.

[4] Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. Active semi-supervision for pairwise constrained clustering. In *ICDM*, 2004. ISBN 0-89871-568-7. URL http://www.siam.org/meetings/sdm04/proceedings/sdm04_031.pdf.

[5] Sugato Basu, Mikhail Bilenko, and Raymond J. Mooney. A probabilistic framework for semi-supervised clustering. In *ACM SIGKDD*, 2004.

[6] Peter Belhumeur. Personal communication with author. 2013.

[7] Aurélien Bellet, Amaury Habrard, and Marc Sebban. A survey on metric learning for feature vectors and structured data. *CoRR*, abs/1306.6709, 2013. URL http://arxiv.org/abs/1306.6709.

[8] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *NIPS*, pages 831–837. MIT Press, 2000.

[9] Arijit Biswas and David W. Jacobs. Active image clustering: Seeking constraints from humans to complement algorithms. In *CVPR*. IEEE, 2012. ISBN 978-1-4673-1226-4. URL http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6235193.

[10] I. Borg and P. Groenen. *Modern multidimensional scaling, theory and applications*. Springer - Verlag , New York, 1997.

[11] J Brickell, I. S. Dhillon, Suvrit Sra, and J Tropp. The metric nearness problem. *SIAM J. Matrix Analysis and Applications*, April 23 2008. URL http://eprints.pascal-network.org/archive/00004443/;http://link.aip.org/link/?SML/30/375.

[12] G. Celeux, F. Forbes, and N. Peyrard. EM procedures using mean field-like approximations for markov model-based image segmentation. *Pattern Recognition*, 2003. URL http://www.sciencedirect.com/science/article/B6V14-4568660-1/2/4af8b0aff5b1d9033e8418e4665c0233.

[13] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Unsupervised metric learning for face identification in TV video. 2011.

[14] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[15] Jason V. Davis, Brian Kulis, Prateek Jain 0002, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *ICML*, 2007. URL http://doi.acm.org/10.1145/1273496.1273523.

[16] Inderjit S. Dhillon, Suvrit Sra, and Joel A. Tropp. Triangle fixing algorithms for the metric nearness problem. In *NIPS*, 2004. URL http://books.nips.cc/papers/files/nips17/NIPS2004_0770.pdf.

[17] B. Erol and F. Kossentini. A robust distance measure for the retrieval of video objects. In *SSIAI*, pages 40–44, 2002.

[18] M. R. Everingham, J. Sivic, and A. Zisserman. Hello! my name is buffy: Automatic naming of characters in TV video. In *BMVC*, 2006. URL http://www.bmva.org/bmvc/2006/papers/340.pdf.

[19] Oren Freifeld and Michael J. Black. Lie bodies: A manifold representation of 3D human shape. In Andrew W. Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *ECCV (1)*, volume 7572 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2012. ISBN 978-3-642-33717-8. URL http://dx.doi.org/10.1007/978-3-642-33718-5.

[20] Steven C. H. Hoi, Rong Jin, and Michael R. Lyu. Learning nonparametric kernel matrices from pairwise constraints. In *ICML*, 2007. URL http://doi.acm.org/10.1145/1273496.1273542.

[21] David W. Jacobs, Daphna Weinshall, and Yoram Gdalyahu. Classification with non-metric distances: Image retrieval and class representation. *IEEE Trans. Pattern Anal. Mach. Intell*, 2000. URL http://doi.ieeecomputersociety.org/10.1109/34.862197.

[22] Dan Klein, Sepandar D. Kamvar, and Christopher D. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *ICML*, 2002.

[23] Neeraj Kumar, Peter N. Belhumeur, Arijit Biswas, David W. Jacobs, W. John Kress, Ida C. Lopez, and João V. B. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *ECCV, 2012*. URL http://dx.doi.org/10.1007/978-3-642-33709-3.

[24] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*. IEEE, 2009. URL http://dx.doi.org/10.1109/ICCV.2009.5459250.

[25] H. B. Ling and D. W. Jacobs. Shape classification using the inner-distance. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(2):286–299, February 2007. URL http://dx.doi.org/10.1109/TPAMI.2007.41.

[26] Z. D. Lu and M. A. Carreira Perpinan. Constrained spectral clustering through affinity propagation. In *CVPR*, 2008. URL http://dx.doi.org/10.1109/CVPR.2008.4587451.

[27] Zhiwu Lu and Horace Ho-Shing Ip. Constrained spectral clustering via exhaustive and efficient constraint propagation. In *ECCV*. Springer, 2010. URL http://dx.doi.org/10.1007/978-3-642-15567-3.

[28] Y. M. Lui and J. R. Beveridge. Grassmann registration manifolds for face recognition. In *ECCV*, pages II: 44–57, 2008. URL http://dx.doi.org/10.1007/978-3-540-88688-4_4.

[29] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *5th Berkeley Symp. on Mathematics Statistics and Probability*, 1967.

[30] F. R. Schmidt, M. Clausen, and D. Cremers. Shape matching by variational computation of geodesics on a manifold. In *DAGM*, pages 142–151, 2006. URL http://dx.doi.org/10.1007/11861898_15.

[31] Nicholas Vretos, Vassilios Solachidis, and Ioannis Pitas. A mutual information based face clustering algorithm for movie content analysis. *Image Vision Comput*, 2011. URL http://dx.doi.org/10.1016/j.imavis.2011.07.006.

[32] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schroedl. Constrained K-means clustering with background knowledge. In *ICML*, 2001.

[33] Baoyuan Wu, Yifan Zhang, Bao-Gang Hu, and Qiang Ji. Constrained clustering and its application to face clustering in videos. In *CVPR*, 2013.

[34] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart J. Russell. Distance metric learning with application to clustering with side-information. In *NIPS*, 2002. URL http://books.nips.cc/papers/files/nips15/AA03.pdf.

[35] Jinfeng Zhuang, Ivor W. Tsang, and Steven C. H. Hoi. SimpleNPKL: simple non-parametric kernel learning. In *ICML*, 2009. URL http://doi.acm.org/10.1145/1553374.1553537.