Improving Embeddings by Flexible Exploitation of Side Information

Ali Ghodsi
University of Waterloo

Dana Wilkinson
University of Waterloo

Finnegan Southey
Google Inc.

Abstract

Dimensionality reduction is a much-studied task in machine learning in which high-dimensional data is mapped, possibly via a non-linear transformation, onto a low-dimensional manifold. The resulting embeddings, however, may fail to capture features of interest. One solution is to learn a distance metric which prefers embeddings that capture the salient features. We propose a novel approach to learning a metric from side information to guide the embedding process.

Our approach admits the use of two kinds of side information. The first kind is class-equivalence information, where some limited number of pairwise "same/different class" statements are known. The second form of side information is a limited set of distances between pairs of points in the target metric space. We demonstrate the effectiveness of the method by producing embeddings that capture features of interest.

1 Introduction

Many machine learning approaches use distances between data points as a discriminating factor. In some cases, the plain Euclidean distance between points is meaningful and such methods can work well. Such distances are the starting point of many popular dimensionality-reduction techniques, such as MDS [Cox and Cox, 2001], LLE [Roweis and Saul, 2000], Isomap [Tenenbaum *et al.*, 2000], LEM [Belkin and Niyogi, 2003], and SDE [Weinberger and Saul, 2004]. Often, however, this distance does not capture the distinction one is trying to characterize. Approaches such as kernelized methods address this issue by mapping the points into new spaces where Euclidean distances may be more useful. An alternative approach is to construct a new distance metric over the points and use it in place of Euclidean distances, as explored by Xing *et al.* [Xing *et al.*, 2003]. In this approach, some amount of *side information* is employed to learn a distance metric that captures the desired distinction.

In some situations it may be possible to obtain a small amount of information regarding the similarity of some points in a particular data set. Consider a large collection of images. While it would be expensive to have a human examine and label the entire set, it might be feasible to have a human examine a small subset and provide information on how pairs of images relate to each other. Similarly, some expensive experiment might yield useful similarity information for a subset of the data. This paper will show how two kinds of such side information can be used in a preprocessing step for embedding techniques, leading to embeddings that capture the target properties. The method improves over earlier work in this area by using standard "off-the-shelf" optimization methods and by allowing more flexibility in the side information used.

The first kind of side information identifies pairs of points that belong to the same class or pairs that belong to different classes. Note that this information is about the class-equivalence/inequivalence of points but does not give the actual class labels. Consider four points, $x_1, x_2, x_3,$ and x_4. Given side information that x_1 and x_2 are in the same class, and that x_3 and x_4 also share a class, we cannot be certain whether the four points fall into one or two classes.

The second kind of side information takes the form of partial information about the similarity of points in the form of distances. For some pairs of points, one may have distances corresponding to an informative metric space, obtained by some expensive measurement. Molecular conformation problems are a good example. Some of the distances between pairs of atoms in a given molecule can be determined by nuclear magnetic resonance spectroscopy, but the procedure is costly and certainly not guaranteed to provide all such distances. Determining the rest of the distances can help in classifying the conformation of the molecule (see [Crippen and Havel, 1988] for more details).

These two kinds of side information can be used to learn a new distance metric. The distances between points in this new space can then be used with any embedding technique. We start by showing how class-equivalence side information can be used to learn such a metric and show the effect of using this new metric with a variety of embedding techniques. We then show how multiple applications of this method can be used to combine several learned distance metrics together into one in order to capture multiple attributes in the embedding. We also discuss how the class-equivalence approach can be kernelized, allowing for nonlinear transformations of the metric. Finally, we formulate a method for using the second type of side information, where we have partial infor-

mation about desirable target distances between some pairs of points. Experimental results demonstrate the value of this preprocessing step.

1.1 Related Work

Our use of class equivalence relations to learn a metric follows [Xing *et al.*, 2003]. In that work, a new distance metric is learned by considering side information. Xing *et al.* used side information identifying pairs of points as "similar". They then construct a metric that minimizes the distance between all such pairs of points. At the same time, they attempt to ensure that all "dissimilar" points are separated by some minimal distance. By default, they consider all points not explicitly identified as similar to be dissimilar. They present algorithms for optimizing this objective and show results using the learned distance for clustering, an application in which an appropriate distance metric is crucial. Other work on learning distance metrics has been primarily focused on classification, with side information in the form of class labels [Globerson and Roweis, 2006; Weinberger *et al.*, 2006; Goldberger *et al.*, 2005].

2 Learning a Metric from Class-Equivalence Side Information

We will start with the simpler similar/dissimilar pair case, formalizing this notion of side information and stating an objective that will be optimized using standard semidefinite programming software. The use of this kind of side information allows one to select a characteristic for distinction. For example, one may have several images of faces. One sensible cluster is by presence or absence of a beard. Another is by the presence or absence of glasses. Different indications of similarity allow the capturing of either distinction. The following takes the same basic approach as [Xing *et al.*, 2003] but offers a simpler optimization procedure using "off-the-shelf" optimization methods instead of their iterative method.

2.1 Derivation

Given a set of t points, $\{x_i\}_{i=1}^t \subseteq R^n$, we identify two kinds of class-related side information. First, a set class-equivalent (or *similar*) pairs of points

$$S: \quad (x_i, x_j) \in \mathcal{S} \quad \text{if } x_i \text{ and } x_j \text{ are similar}$$

and, second, a set of class-inequivalent (*dissimilar*) pairs

$$O: \quad (x_i, x_j) \in \mathcal{O} \quad \text{if } x_i \text{ and } x_j \text{ are dissimilar}$$

We then wish to learn a matrix A that induces a distance metric $D^{(A)}$ over the points

$$D^{(A)}(x_i, x_j) = \|x_i - x_j\|_A = \sqrt{(x_i - x_j)^T A (x_i - x_j)}$$

where $A \succeq 0$.

We define the following loss function, which, when minimized, attempts to minimize the squared induced distance between similar points and maximize the squared induced distance between dissimilar points

$$L(A) = \sum_{(x_i,x_j)\in\mathcal{S}} \|x_i - x_j\|_A^2 - \sum_{(x_i,x_j)\in\mathcal{O}} \|x_i - x_j\|_A^2$$

The optimization problem then becomes

$$\min_A L(A) \text{ s.t. } A \succeq 0 \text{ and } \mathrm{Tr}(A) = 1 \tag{1}$$

The first constraint (positive semidefiniteness) ensures a Euclidean metric. The second excludes the trivial solution where all distances are zero. The constant in this constraint is arbitrary, affecting only the scale of the space. This objective will be optimized using standard semidefinite programming software and so it must be converted to a linear objective. Expanding the loss function

$$\begin{aligned} L(A) \quad = \quad & \sum_{(x_i,x_j)\in\mathcal{S}} (x_i - x_j)^T A (x_i - x_j) - \\ & \sum_{(x_i,x_j)\in\mathcal{O}} (x_i - x_j)^T A (x_i - x_j) \end{aligned}$$

each squared distance term must be converted. We start by observing that $\mathrm{vec}(XYZ) = (Z^T \otimes X)\mathrm{vec}(Y)$, where $\mathrm{vec}()$ simply rearranges a matrix into a vector by concatenating columns and $\otimes$ is the Kronecker product. Note that $(x_i - x_j)^T A (x_i - x_j) = \mathrm{vec}((x_i - x_j)^T A (x_i - x_j))$ because the left-hand side is a scalar. Using this and the fact that $(a^T \otimes b^T) = \mathrm{vec}(ba^T)^T$, we can rewrite the squared distance terms as

$$\begin{aligned} & (x_i - x_j)^T A (x_i - x_j) \\ = \quad & \mathrm{vec}((x_i - x_j)^T A (x_i - x_j)) \\ = \quad & ((x_i - x_j)^T \otimes (x_i - x_j)^T)\mathrm{vec}(A) \\ = \quad & \mathrm{vec}((x_i - x_j)(x_i - x_j)^T)^T \mathrm{vec}(A) \\ = \quad & \mathrm{vec}(A)^T \mathrm{vec}((x_i - x_j)(x_i - x_j)^T) \end{aligned}$$

The linear loss function is then

$$\begin{aligned} L(A) \quad = \quad & \sum_{(x_i,x_j)\in\mathcal{S}} \mathrm{vec}(A)^T \mathrm{vec}((x_i - x_j)(x_i - x_j)^T) - \\ & \sum_{(x_i,x_j)\in\mathcal{O}} \mathrm{vec}(A)^T \mathrm{vec}((x_i - x_j)(x_i - x_j)^T) \\ = \quad & \mathrm{vec}(A)^T \left[\sum_{(x_i,x_j)\in\mathcal{S}} \mathrm{vec}((x_i - x_j)(x_i - x_j)^T) \right. \\ & \left. - \sum_{(x_i,x_j)\in\mathcal{O}} \mathrm{vec}((x_i - x_j)(x_i - x_j)^T) \right] \end{aligned}$$

This form, along with the two constraints from (1), can be readily submitted to an SDP solver to optimize the matrix A[1]. Aside from this convenient form, this formulation has other advantages over that used by Xing *et al.*, especially with respect to the side information we can convey.

Xing *et al.* require at least one dissimilar pair in order to avoid the trivial solution where all distances are zero. The constraint on the trace that we employ means that we need not place any restrictions on pairings. Side information can consist of similar pairs only, dissimilar pairs only, or any combination of the two; the method still avoids trivial solutions.

[1] We use the MATLAB SDP solver SeDuMi [Sturm, 1999]

Furthermore, in the absence of specific information regarding dissimilarities, Xing *et al*. assume that all points not explicitly identified as similar are dissimilar. This information may be misleading, forcing the algorithm to separate points that may, in fact, be similar. The formulation presented here allows one to specify only the side information one actually has, partitioning the pairings into similar, dissimilar, and unknown.

2.2 Results

Once a metric has been learned, the new distances can be used with any embedding technique. To demonstrate the benefits of this approach, we generated embeddings with and without the preprocessing step, informing the preprocessed embeddings about some characteristic of interest, and then examined the result. To show that the side information truly informs the embedding, two sets of informed embeddings were generated from the same data set, each with side information pertaining to two different characteristics. Structure in the resulting embeddings that captures the two different characteristics within a single data set is evidence that the method works as intended.

Embeddings were generated using a variety of techniques including MDS, Isomap, LEM, LLE, and SDE. The data set consisted of 200 images of faces and two distinctions are identified by side information: faces with beards vs. faces without beards and faces with glasses vs. faces without glasses. In one set of experiments (which we will call *all-similar*), all similar pairs were identified but no dissimilar pairs. The second set (which we will call *five-pairs*), simulating a situation where labelling is expensive, identifies only four similar pairs and one dissimilar pair. The pairs were selected at random. Techniques using the preprocessor are labelled as "Equivalence-Informed" (e.g., Equivalence-Informed MDS). In each case, a two-dimensional embedding is shown. The two classes are marked with X and O, respectively. Additionally, a subset of the images are displayed on the plot (plots with all images are unreadable). Some plots have been omitted due to space constraints.

Inspection reveals that, in general, the informed versions of these embeddings manage to separate the data based on the target property, whereas the uninformed versions are typically chaotic. Even when separation is poor in the informed embedding, there is usually much more structure than in the uninformed embedding. Equivalence-Informed MDS (Figures 1 and 2, five-pairs beards and all-pairs glasses omitted) offers mixed results, especially with five-pairs for glasses vs. no glasses (Figure 2), but MDS is a comparatively crude embedding technique in any case. Isomap with all-pairs works very well (Figure 3, glasses omitted) and with five-pairs it still manages to group the two classes but does not separate them well (Figure 4, glasses omitted). LEM separates well with all-pairs (Figure 5, glasses omitted) but shows weak separation in the five-pairs case (Figure 6, glasses omitted). The same effective grouping and varying separation occurs for LLE (figures omitted). Finally, SDE groups well, but is weak in separation in all cases (Figures 7 and 8, glasses omitted).

These results show that the side information can be effectively exploited to capture characteristics of interest where uninformed embedding techniques fail to automatically discover them. Moreover, two different characteristics (beards and glasses) have been captured within the same data set, even when using only small quantities of class-equivalence side information. Finally, the preprocessor is effective with a wide range of embedding techniques.

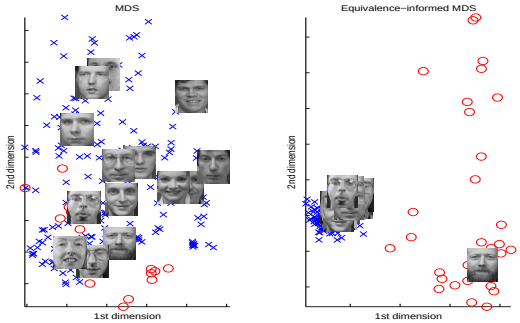

Figure 1: *MDS and Equivalence-Informed MDS with (un)bearded distinction (all equivalent pairs)*

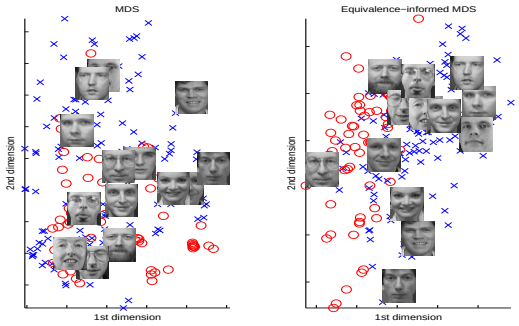

Figure 2: *MDS and Equivalence-Informed MDS with glasses/no glasses distinction (4 equivalent/1 inequiv. pairs)*

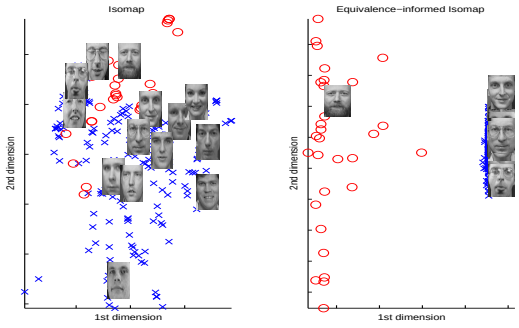

Figure 3: *Isomap and Equivalence-Informed Isomap with (un)bearded distinction (all equivalent pairs)*

Finally, Figures 9 and 10 are provided to give a numerical notion of the improvement offered by informed embeddings. The plots show the misclassification rate of k-means clustering in recovering the true classes of the images, after applying each of the uninformed and informed embeddings (indicated by, e.g., "MDS" and "Inf. MDS"). The informed methods achieve a lower error rate than their uninformed counterparts by better separating the data (except 4-pairs SDE).

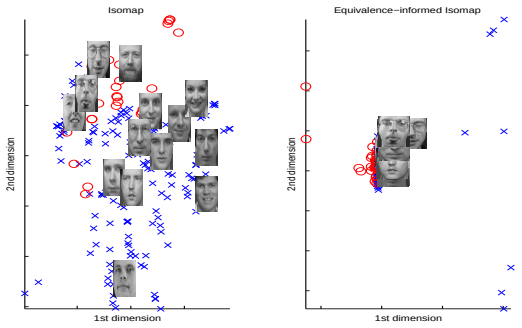

Figure 4: *Isomap and Equivalence-Informed Isomap with (un)bearded distinction (4 equivalent/1 inequivalent pairs)*

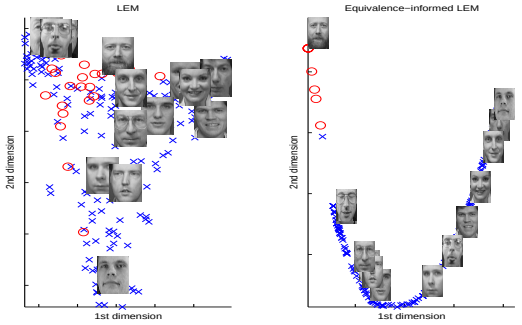

Figure 5: *LEM and Equivalence-Informed LEM with (un)bearded distinction (all equivalent pairs)*

3 Multiple-Attribute Metric Learning with Class-Equivalence Side Information

In some cases, there may be more than one distinction to capture in data (e.g., glasses vs. no glasses **and** beards vs. no beards). The method above can be extended to construct a distance metric using multiple sets of side information, each corresponding to a different criterion. Multiple metrics are learned and then combined to form a single metric.

Suppose there are k different sets of side information over the same set of points. Using the optimization described above, k transformations, $A_1, \cdots, A_k$ can be learned. From each A_i, the dominant eigenvector v_i can be taken and a new matrix assembled where each column is one of these eigenvectors, $\bar{A} = [\ v_1 \quad \cdots \quad v_k\ ]$. This combined matrix defines a rank k linear subspace onto which we can project the data. Applying singular value decomposition to $\bar{A}$, one can form a matrix U from the k eigenvectors corresponding to the largest eigenvalues. The final transformation is then $\hat{A} = UU^T$, which is an orthonormal basis combining the distinctions drawn by each kind of side information.

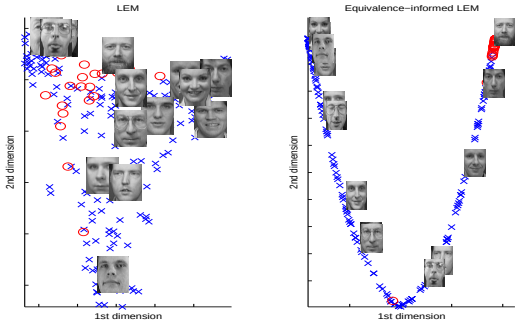

Figure 6: *LEM and Equivalence-Informed LEM with (un)bearded distinction (4 equivalent/1 inequivalent pairs)*

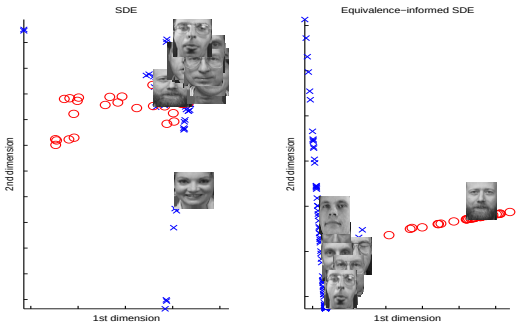

Figure 7: *SDE and Equivalence-Informed SDE with (un)bearded distinction (all equivalent pairs)*

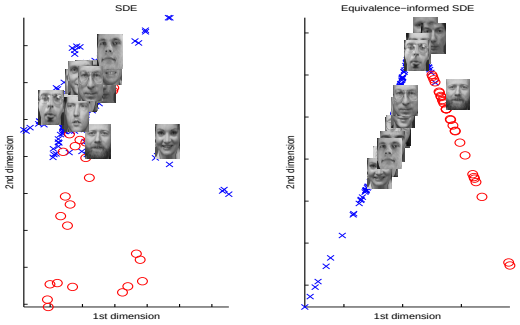

Figure 8: *SDE and Equivalence-Informed SDE with (un)bearded distinction (4 equivalent/1 inequivalent pairs)*

3.1 Results

We demonstrate the effectiveness of this method by creating a single embedding that captures two distinctions. We repeat the earlier experiments but using both the (un)bearded and glasses/no glasses criteria together. Results were generated for all the embedding methods and their multiple-attribute, equivalence-informed versions but some are omitted for space. In all cases, the side information consisted of all similar pairs and no dissimilar pairs.

The informed embeddings tend to discover the four distinct categories of faces (combinations of the presence and absence of beards and glasses). Informed MDS (Figure 11) separates

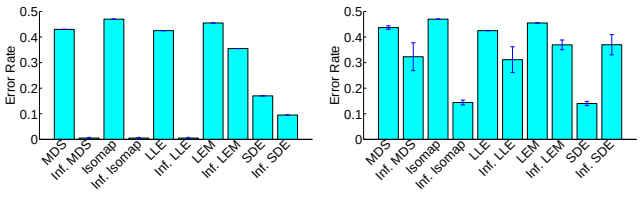

Figure 9: *Clustering Error for beards vs. no beards (Left: all equivalent pairs Right: 4 equiv/1 inequiv pairs)*

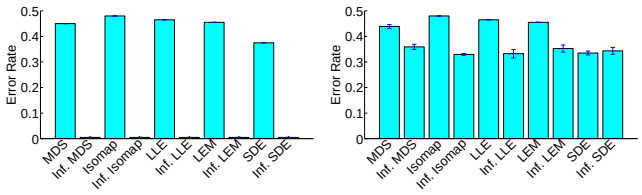

Figure 10: *Clustering Error for glasses vs. no glasses (Left: all equivalent pairs Right: 4 equiv/1 inequiv pairs)*

these quite well, as do Informed Isomap (Figure 12) and Informed LLE (Figure 13). Results with Informed LEM (figure omitted) are roughly grouped. Finally, Informed SDE (Figure 14) does has trouble separating two categories, but a lot of useful structure is present. In all cases, the informed embeddings have distinct structure that the uninformed embeddings fail to capture. The only feature the uninformed embeddings seem to capture is the relative darkness of the images.

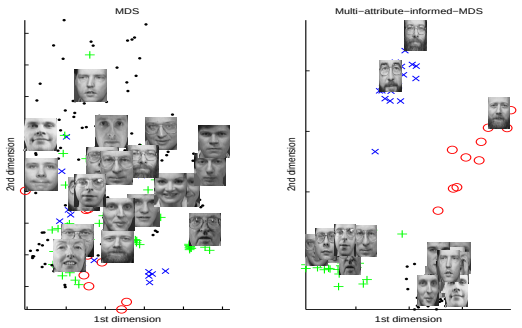

Figure 11: *MDS and Multi-Attribute Informed MDS with (un)bearded and (no)glasses attributes (all equivalent pairs)*

These results show that the multiple-attribute approach can capture the desired properties in a single embedding, while the uninformed versions fail to capture either of the properties of interest. Again, the method is straightforward to apply and works with a variety of embedding techniques.

4 Kernelizing Metric Learning

Not uncommonly, nonlinear transformations of the data are required to successfully apply learning algorithms. One efficient method for doing this is via a kernel that computes a

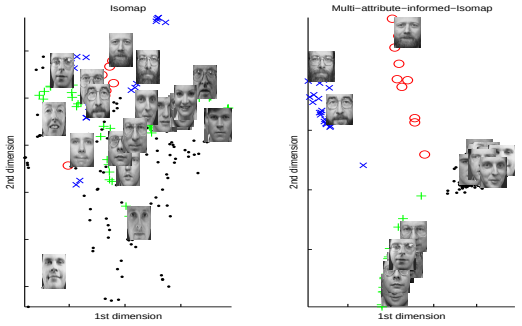

Figure 12: *Isomap and Multi-Attribute Informed Isomap with (un)bearded and (no)glasses attributes (all equivalent pairs)*

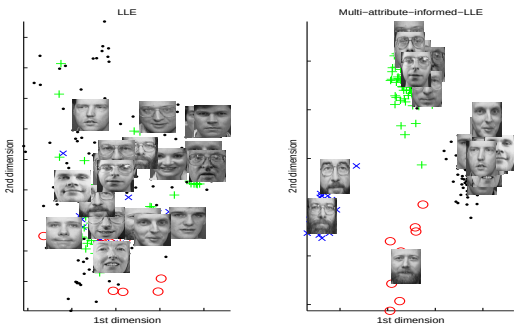

Figure 13: *LLE and Multi-Attribute Informed LLE with (un)bearded and (no)glasses attributes (all equivalent pairs)*

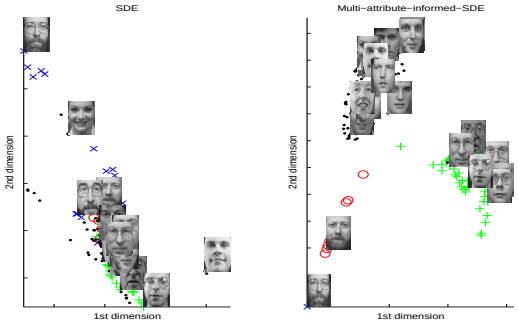

Figure 14: *SDE and Multi-Attribute Informed SDE with (un)bearded and (no)glasses attributes (all equivalent pairs)*

similarity measure between any two data points. In this section, we show how to learn a metric in the feature space implied by a kernel, allowing our use of side information to be extended to nonlinear mappings of the data.

Conceptually, the points are mapped into a feature space by some nonlinear mapping $\Phi()$ and a metric is learned in that space. Actually applying the mapping is often undesirable (e.g., the feature vectors may have infinite dimension), so we employ the well known *kernel trick*, using some kernel $K(x_i, x_j)$ that computes inner products between feature vectors without explicitly constructing them.

The squared distances in our objective have the form $(x_i - x_j)^T A(x_i - x_j)$. Because A is positive semidefinite, it can be decomposed into $A = WW^T$. W can then be expressed as a linear combination of the data points, $W = X\beta$, via the kernel trick. Rewriting the squared distance

$$\begin{aligned} & (x_i - x_j)^T A(x_i - x_j) \\ = \; & (x_i - x_j)^T WW^T(x_i - x_j) \\ = \; & (x_i - x_j)^T X\beta\beta^T X^T(x_i - x_j) \\ = \; & (x_i^T X - x_j^T X)\beta\beta^T(X^T x_i - X^T x_j) \\ = \; & (X^T x_i - X^T x_j)^T \mathcal{A}(X^T x_i - X^T x_j) \end{aligned}$$

where $\mathcal{A} = \beta\beta^T$, we have now expressed the distance in terms of the matrix to be learned, $\mathcal{A}$, and inner products between data points, which can be computed via the kernel, K. The optimization of $\mathcal{A}$ proceeds as in the non-kernelized version but with one additional constraint. The rank of $\mathcal{A}$ must be t because β is t by n and $\mathcal{A} = \beta\beta^T$. However, this con-

straint is problematic, so we drop it during the optimization and then find a rank t approximation of $\mathcal{A}$ via singular value decomposition.

5 Using Partial Distance Side Information

We will now discuss the use of the second kind of side information mentioned at the beginning of this paper. Recall that in this case, we have side information that gives exact distances between some pairs of points in some space natural to the data. Such partial distance information can be used to inform our learned metric. Given a set of similarities in the form of pairs for which distances are known

$$S: \quad (x_i, x_j) \in \mathcal{S} \quad \text{if the target distance } d_{ij} \text{ is known}$$

the following cost function is employed, which attempts to preserve the set of known distances

$$L(A) = \sum_{(x_i,x_j)\in\mathcal{S}} \left\| \|x_i - x_j\|_A^2 - d_{ij} \right\|^2$$

The optimization problem is then

$$\min_A L(A) \qquad \text{s.t. } A \succeq 0$$

A convenient form for this optimization is obtained using the same approach for quadratic terms used earlier

$$\begin{aligned} L(A) &= \sum_{(x_i,x_j)\in\mathcal{S}} \left\| \|x_i - x_j\|_A^2 - d_{ij} \right\|^2 \\ &= \sum_{(x_i,x_j)\in\mathcal{S}} \left\| \text{vec}(A)^T \text{vec}(B_{ij}) - d_{ij} \right\|^2 \\ &= \sum_{(x_i,x_j)\in\mathcal{S}} \text{vec}(A)^T \text{vec}(B_{ij}) \text{vec}(B_{ij})^T \text{vec}(A) \\ &\qquad + d_{ij}^2 - 2 d_{ij} \text{vec}(A)^T \text{vec}(B_{ij}) \end{aligned}$$

where $B_{ij} = (x_i - x_j)(x_i - x_j)^T$. Note that the d_{ij}^2 term in the above is a constant so it can be dropped for the purposes of minimization, and the loss rewritten as

$$\begin{aligned} L(A) &= \text{vec}(A)^T \left[\sum_{(x_i,x_j)\in\mathcal{S}} \text{vec}(B_{ij}) \text{vec}(B_{ij})^T \text{vec}(A) \right. \\ &\qquad \left. -2 \sum_{(x_i,x_j)\in\mathcal{S}} d_{ij} \text{vec}(B_{ij}) \right] \\ &= \text{vec}(A)^T \left[Q\text{vec}(A) - 2R \right] \end{aligned}$$

where $Q = \sum_{(x_i,x_j)\in\mathcal{S}} \text{vec}(B_{ij}) \text{vec}(B_{ij})^T$ and $R = \sum_{(x_i,x_j)\in\mathcal{S}} d_{ij} \text{vec}(B_{ij})$. This objective is still quadratic but a linear objective can be obtained via the Schur complement [Boyd and Vandenberghe, 2004], as we outline here.

The Schur complement relates the positive semidefiniteness of the matrix on the left and the expression on the right by an if-and-only-if relation

$$\begin{bmatrix} X & Y \\ Y^T & Z \end{bmatrix} \succeq 0 \quad \Leftrightarrow \quad Z - Y^T X^{-1} Y \succeq 0$$

By decomposing $Q = S^T S$, a matrix of the form

$$J = \begin{bmatrix} I & S\text{vec}(A) \\ (S\text{vec(A)})^T & 2\text{vec(A)}^T R + t \end{bmatrix}$$

can be constructed. By the Schur complement, if $J \succeq 0$, then the following relation holds

$$2\text{vec}(A)^T R + t - \text{vec}(A)^T S^T S \text{vec}(A) \quad \geq \quad 0$$

Note that this quantity is scalar. As long as J is positive semidefinite, the scalar t is an upper bound on the loss

$$\begin{aligned} &\text{vec}(A)^T S^T S \text{vec}(A) - 2\text{vec}(A)^T R \\ = \quad &\text{vec}(A)^T Q \text{vec}(A) - 2\text{vec}(A)^T R \quad \leq t \end{aligned}$$

Therefore, minimizing t subject to $J \succeq 0$ also minimizes the objective. This optimization problem can be readily solved by standard semidefinite programming software

$$\min_A t \quad \text{s.t.} \quad A \succeq 0 \quad \text{and} \quad J \succeq 0$$

5.1 Results

Again, we demonstrate effectiveness by generating uninformed and informed embeddings but with target distances specified for some pairs of points. The distances used here are a portion of the distance D^A computed in Section 2.2. That is, there are 200 images of faces and two distinctions are identified by class-equivalence side information: faces with beards vs. faces without beards and faces with glasses vs. faces without glasses. For each distinction, the distance matrix D^A is computed, where no dissimilar pairs but all similar pairs are identified. Then 30 pairs of points are selected at random and their distances in D^A are used as the side information for learning a new distance metric. This new distance metric is then used to inform the embeddings.

Results using these distances are somewhat noisier than with the class-equivalence informed embeddings but still provide good structure compared with the uninformed versions, and they group the data. Informed MDS (Figure 15, glasses omitted) and Informed Isomap (Figures 16, glassed omitted) group quite effectively, with a few images placed inappropriately. Informed LEM (Figures 17 and 18), Informed LLE (figures omitted), and Informed SDE (figures omitted) all give results similar to one another, and all show noisier performance on the glasses attribute than on the beards (beards are likely easier since they create a more substantial difference in an image than glasses). All methods group beardedness quite well. These last results show that preprocessing based on partial distance information is effective, although less so than the class-equivalence information. The informed embeddings are still more useful than the uninformed.

6 Conclusions

Many machine learning techniques handle complex data by mapping it into new spaces and then applying standard techniques, often utilizing distances in the new space. Learning a distance metric directly is an elegant means to this end. This research shows how side information of two forms, class-equivalence information and partial distance information, can

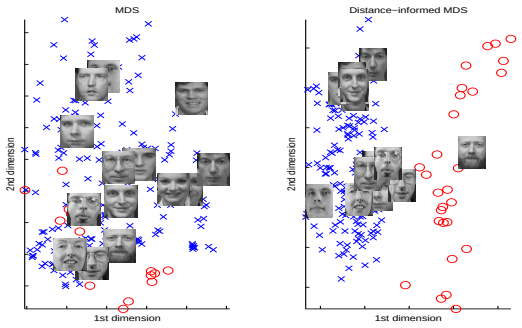

Figure 15: *MDS and Distance-Informed MDS with (un)bearded distinction (30 distances)*

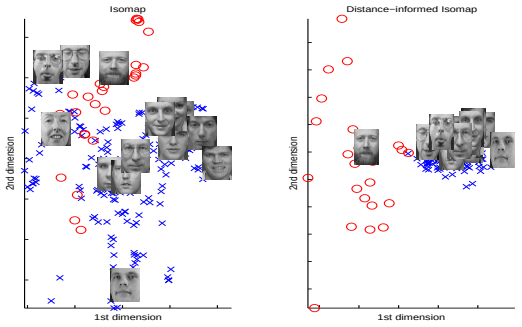

Figure 16: *Isomap and Distance-Informed Isomap with (un)bearded distinction (30 distances)*

be used to learn a new distance as a preprocessing step for embedding. The results demonstrate that side information allows us to capture attributes of the data within the embedding in a controllable manner. Even a small amount of side information can improve the embedding's structure. Furthermore, the method can be kernelized and also used to capture multiple attributes in the same embedding. Its advantages over existing metric learning methods are a simpler optimization formulation that can be readily solved with off-the-shelf software and greater flexibility in the nature of the side information one can use. In all, this technique represents a useful addition to our toolbox for informed embeddings.

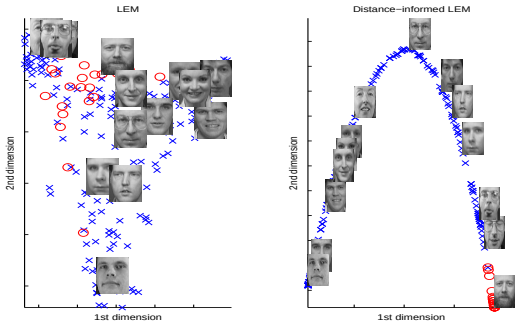

Figure 17: *LEM and Distance-Informed LEM with (un)bearded distinction (30 distances)*

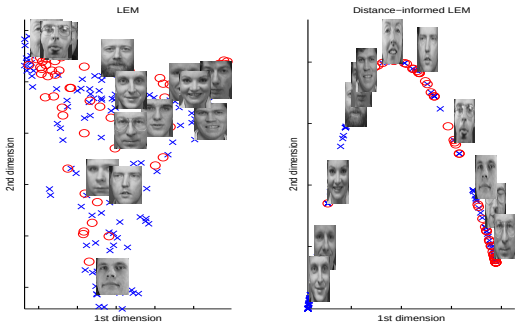

Figure 18: *LEM and Distance-Informed LEM with (no)glasses distinction (30 distances)*

References

[Belkin and Niyogi, 2003] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[Boyd and Vandenberghe, 2004] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, New York, 2004.

[Cox and Cox, 2001] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman Hall, 2nd edition edition, 2001.

[Crippen and Havel, 1988] G. Crippen and T. Havel. *Distance geometry and molecular conformation*. Research Studies Press Ltd., Letchworth, 1988.

[Globerson and Roweis, 2006] Amir Globerson and Sam Roweis. Metric learning by collapsing classes. In *NIPS-2005*, pages 451–458. 2006.

[Goldberger *et al.*, 2005] Jacob Goldberger, Sam Roweis, Geoffrey Hinton, and Ruslan Salakhutdinov. Neighbourhood components analysis. In *NIPS-2004*, pages 513–520. 2005.

[Roweis and Saul, 2000] Sam Roweis and Lawrence Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.

[Sturm, 1999] J. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods Softw.*, 11/12(1-4):625–653, 1999.

[Tenenbaum *et al.*, 2000] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[Weinberger and Saul, 2004] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *CVPR-2004*, volume II, pages 988–995, 2004.

[Weinberger *et al.*, 2006] Kilian Weinberger, John Blitzer, and Lawrence Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS-2005*, pages 1475–1482. 2006.

[Xing *et al.*, 2003] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance metric learning with application to clustering with side-information. In *NIPS-2002*, pages 505–512. 2003.