August 2003

Learning High Dimensional Correspondences from Low Dimensional Manifolds

Ji Hun Ham
University of Pennsylvania

Daniel D. Lee
University of Pennsylvania, ddlee@seas.upenn.edu

Lawrence K. Saul
University of Pennsylvania, lsaul@cis.upenn.edu

Presented at the 20th International Conference on Machine Learning (ICML 2003) Workshop: The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, held 21-24 August 2003 in Washington, DC.

Learning High Dimensional Correspondences from Low Dimensional Manifolds

Abstract
Many *different* high dimensional data sets are characterized by the *same* underlying modes of variability. When these modes of variability are continuous and few in number, they can be viewed as parameterizing a low dimensional manifold. The manifold provides a compact shared representation of the data, suggesting correspondences between the high dimensional examples from different data sets. These correspondences, though naturally induced by the underlying manifold, are difficult to learn using traditional methods in supervised learning. In this paper, we generalize three methods in unsupervised learning—principal components analysis, factor analysis, and locally linear embedding—to discover subspaces and manifolds that provide common low dimensional representations of different high dimensional data sets. We use the shared representations discovered by these algorithms to put high dimensional examples from different data sets into correspondence. Finally, we show that a notion of "self-correspondence" between examples in the same data set can be used to improve the performance of these algorithms on small data sets. The algorithms are demonstrated on images and text.

Comments
Presented at the 20th International Conference on Machine Learning (ICML 2003) Workshop: The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, held 21-24 August 2003 in Washington, DC.

Learning High Dimensional Correspondences from Low Dimensional Manifolds

Ji Hun Ham JHHAM@SEAS.UPENN.EDU
Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA

Daniel D. Lee DDLEE@SEAS.UPENN.EDU
Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA

Lawrence K. Saul LSAUL@SEAS.UPENN.EDU
Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104 USA

Abstract

Many *different* high dimensional data sets are characterized by the *same* underlying modes of variability. When these modes of variability are continuous and few in number, they can be viewed as parameterizing a low dimensional manifold. The manifold provides a compact shared representation of the data, suggesting correspondences between the high dimensional examples from different data sets. These correspondences, though naturally induced by the underlying manifold, are difficult to learn using traditional methods in supervised learning. In this paper, we generalize three methods in unsupervised learning—principal components analysis, factor analysis, and locally linear embedding—to discover subspaces and manifolds that provide common low dimensional representations of different high dimensional data sets. We use the shared representations discovered by these algorithms to put high dimensional examples from different data sets into correspondence. Finally, we show that a notion of "self-correspondence" between examples in the same data set can be used to improve the performance of these algorithms on small data sets. The algorithms are demonstrated on images and text.

1. Introduction

Many problems in statistical pattern recognition blur the common distinction between methods in supervised and unsupervised learning. A traditional problem in supervised learning is classification: the mapping of multivariate inputs to discrete outputs. Decision trees, neural networks, and support vector machines provide solutions to these problems when the input examples are high dimensional. Suppose, however, that the desired outputs are not discrete labels, but are themselves high dimensional examples from another data set. In such a problem, the goal is not to classify the inputs, but to learn their high dimensional correspondences. One solution to this problem is to learn common low dimensional representations for different high dimensional data sets. Dimensionality reduction has been extensively studied in the framework unsupervised learning, but not generally with the idea of putting high dimensional examples from different data sets into correspondence. For these reasons, the problem of learning high dimensional correspondences is best tackled by a mixture of supervised and unsupervised methods.

In this paper, we investigate automatic methods for learning correspondences between high dimensional examples from different data sets (Hotelling, 1936). For example, one data set $\{\boldsymbol{x}^{\mathbf{1}}\}$ could consist of images of an object taken from multiple viewpoints, and another data set $\{\boldsymbol{x}^{\mathbf{2}}\}$ could consist of images of a different object from different viewpoints. Given an image of the first object, is it possible for a learning algorithm to determine the corresponding view of the second object?

We assume that the algorithm is given a subset of examples that are in correspondence, $\{\boldsymbol{x}_i^{\mathbf{1}} \leftrightarrow \boldsymbol{x}_i^{\mathbf{2}}\}$, and a larger set of examples $\{\boldsymbol{x}_j^{\mathbf{1}}\}$ and $\{\boldsymbol{x}_k^{\mathbf{2}}\}$ with unknown correspondence. Simple regressions do not work well for such data because of the high dimensionality of the examples and the small number of labelled correspondences. This is shown in the first two rows of Figure 1, where a linear perceptron and a backpropagation neural network have been trained to map images of one object from rotated viewpoints into corresponding images of another object. Because the number

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

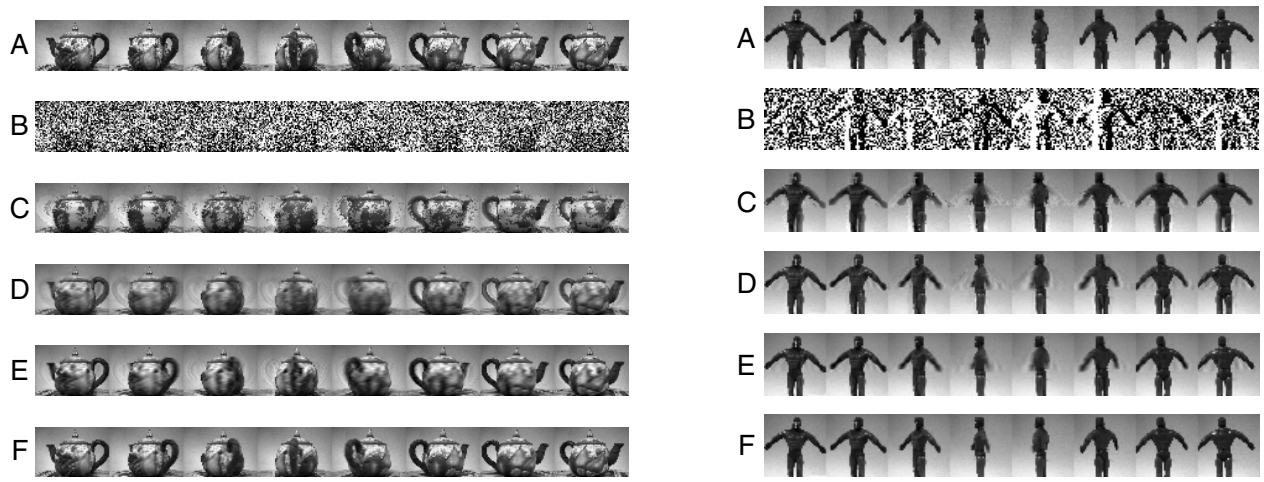

Figure 1. On the left are reconstructed images of a teapot, given the corresponding image of an action figure from the same viewpoint. On the right are reconstructed images of the action figure, given the corresponding view of the teapot. The original images are given in row (A); the reconstructions were generated by a linear perceptron (B), a backpropagation neural network (C), principal components analysis (D), factor analysis (E), and constrained locally linear embedding (F). Each of the data sets consisted of 200 grayscale images, with 120 in labelled correspondence. The reconstructions were generated by first embedding the data onto 15-dimensional subspaces and 3-dimensional manifolds as described in the text.

of images in labelled correspondence is very small relative to the number of parameters that must be estimated, these models severely overfit the data and lead to extremely poor generalization.

What makes this correspondence problem tractable? Often, the variability within a high dimensional data set, such as the image variability of objects from different viewpoints, is characterized by a low dimensional manifold (Hinton et al., 1997; Seung & Lee, 2000). In the remainder of this paper, we generalize three methods in unsupervised learning—principal components analysis, factor analysis, and locally linear embedding (Roweis & Saul, 2000; Saul & Roweis, 2003)—to discover subspaces and manifolds that provide common low dimensional representations of different high dimensional data sets. We then show how to use these shared representations to put high dimensional examples from different data sets into correspondence. As shown in Figure 1, this approach leads to much better reconstructions of corresponding images from other data sets.

2. Mathematical formulation

We consider two data sets; the first set $\{\boldsymbol{x^1}\}$ consists of n_1 vectors of dimensionality d_1 so that $\boldsymbol{x_i^1} \in \Re^{d_1}$ for $i = 1, ..., n_1$. The second set $\{\boldsymbol{x^2}\}$ consists of n_2 vectors of dimensionality d_2. Among the vectors in the two sets, we are given a small number $n_c < n_1, n_2$ of pairs in one-to-one correspondence. Without loss of generality, we assume the corresponding pairs are given by $\boldsymbol{x_i^1} \leftrightarrow \boldsymbol{x_i^2}$ for $i = 1, ..., n_c$.

The data in the two sets can be summarized by forming the matrix:

$$X = \begin{bmatrix} \boldsymbol{x_i^1} & \boldsymbol{x_j^1} & \boldsymbol{?} \\ \boldsymbol{x_i^2} & \boldsymbol{?} & \boldsymbol{x_k^2} \end{bmatrix}. \tag{1}$$

X is a $(d_1 + d_2) \times (n_1 + n_2 - n_c)$ matrix, where the top rows consists of elements from the first data set and the bottom rows consists of elements from the second data set. The first n_c columns of matrix X consist of the augmented vectors from the two sets which are in correspondence. The other columns consist of vectors with unknown correspondence: x_j^1 for $j = (n_c + 1), ..., n_1$ and x_k^2 for $k = (n_c + 1), ..., n_2$. We can also write matrix X in block matrix form:

$$X = \left[\; X_c \; \middle| \; X_s \; \right] = \left[\begin{array}{c|cc} X_c^1 & X_s^1 & X_r^1 \\ X_c^2 & X_r^2 & X_s^2 \end{array} \right]. \tag{2}$$

The submatrix X_c denotes the leftmost n_c columns of X which contain corresponding vectors from the two data sets and are fully known. The columns of X_s contain vectors from only a single data set, and thus contain the unknown portions X_r^1 and X_r^2 that need to be reconstructed.

The problem for a learning algorithm is to fill in the unknown parts of matrix X. A traditional supervised learning approach would be to take the known matrix X_c as training data, then to use the columns of X_s^1 or X_s^2 as input to obtain the predicted outputs X_r^1 and X_r^2. However, if the number of known correspondences n_c is small, this approach may not generalize properly as seen from the examples in Figure 1. On the other hand, an alternative approach is to treat this data matrix as an unsupervised learning problem with

missing data that need to be reconstructed. In the following sections, we describe three different techniques that can be used for solving this problem.

3. Linear subspaces

One popular approach in unsupervised learning is to model the data as lying on a linear subspace embedded in a high dimensional space. The projections of the data along the bases of the linear subspace then constitutes a low dimensional representation of the data. In the context of the correspondence problem, this low dimensional representation should be consistent between the two different data sets. By treating the correspondence problem as a large missing data problem, it is possible to simultaneously learn a consistent low dimensional representation as well as fill in the unknown high dimensional missing values. In order to handle this missing data problem, we consider these algorithms within a generative model framework.

3.1. Principal components analysis

Principal components analysis (PCA) is a standard method in statistical analysis (Jolliffe, 1986). As a generative model, it can be considered as the limiting case of a probabilistic generative model mediated by a small number d_ξ of latent variables $\boldsymbol{\xi}$: $\boldsymbol{x} = W\boldsymbol{\xi} + \boldsymbol{\eta}$, where W is a d_x by d_ξ weight matrix, and $\boldsymbol{\eta}$ is a Gaussian random noise term whose variance goes to 0.

Given a data matrix X, learning involves estimating the optimal model parameters W as well as inferring the hidden variables $\boldsymbol{\xi}$. In the case of learning correspondences with PCA, we treat the data in X as examples of dimensionality $d_x = d_1 + d_2$, where all the entries in X_c are fully known, and all the examples in X_s contain missing entries. Because of these missing entries, the model parameters cannot be estimated by diagonalizing the covariance matrix as is usually the case with PCA. In order to properly fill in these missing entries and simultaneously learn the appropriate model parameters, we use the Expectation-Maximization (EM) (Dempster et al., 1977) algorithm. As described in (Saul & Rahim, 1999; Roweis, 1998; Tipping & Bishop, 1999), the EM algorithm iteratively estimates the posterior distributions of the hidden and missing variables, and then reestimates the model parameters based upon the estimated posterior distributions. With each iteration, the likelihood of the data under the model distribution increases monotonically until convergence is achieved.

We applied the EM algorithm for PCA on images of the teapot and action figure in Figure 1. The algorithm finds a $d_\xi = 15$ dimensional linear subspace which best fits the variability in the pixel values of both sets of images simultaneously. Given the image of one object with unknown correspondence, the algorithm first projects this image onto this low dimensional subspace to determine the appropriate values of the hidden variables ξ that best describes the known image. These components are then projected back into the high dimensional space of the second image object to reconstruct the unknown image with the corresponding view. Although this reconstruction corresponds to a linear mapping between the two data sets, the use of the low dimensional subspace leads to better reconstructions than the purely supervised learning techniques.

3.2. Factor analysis

Factor analysis (FA) is a more general linear latent variable model, described by the generative equation:

$$\boldsymbol{x} = \boldsymbol{\mu} + W\boldsymbol{\xi} + \boldsymbol{\eta}. \tag{3}$$

In this case, $\boldsymbol{\mu}$ is a vector of dimensionality d_x giving the mean of the distribution, W is known as the factor loading matrix, $\boldsymbol{\xi}$ describes the low dimensional latent variables, and $\boldsymbol{\eta}$ is uncorrelated Gaussian random noise with diagonal covariance $\Psi = \langle \boldsymbol{\eta}\boldsymbol{\eta}^T \rangle$.

Learning a factor analysis model involves estimating the model parameters $\boldsymbol{\mu}$, W, and Ψ from the data and requires inferring the posterior distributions of the hidden variables ξ. If the data X has missing values, estimating these parameters also involves inferring the distributions over these missing variables as well. In this regard, the EM algorithm is convenient for simultaneously filling in these missing values as well as learning the model parameters. The derivation of the EM algorithm for factor analysis is straightforward, although more complicated than that for PCA. The details of the derivation and the expressions used for the E-step and M-step are provided in the appendix.

The algorithm was used to determine the optimal parameters of a 15-dimensional factor analysis model for the images in Figure 1. Given an image with unknown correspondence, the factor analysis model first projects the image onto the low dimensional subspace described by the basis in the factor loading matrix. However, in contrast to PCA which assumes that the pixel noise covariance $\Psi \to 0$, the factor analysis model weights the different pixel values according to the noise estimates in determining the optimal low dimensional representation. This representation is then used to project back into the high dimensional space of the second image object to construct the appropriate reconstructions.

A quantitative comparison of the errors from these algorithms in reconstructing the corresponding images is plotted in Figure 2. In general, the linear manifold representations lead to better reconstruction error than the purely supervised techniques, with the factor analysis model outperforming PCA. However, neither of these techniques ap-

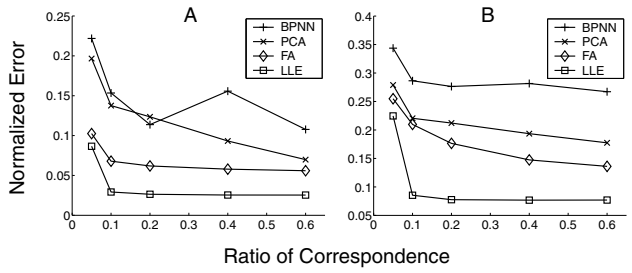

Figure 2. Normalized reconstruction error for the teapot images (A) and the action figure images (B) for four of the algorithms shown in Figure 1. Errors from the perceptron algorithm were too large to be plotted on the same scale.

proach the performance of an algorithm that uses a nonlinear low dimensional representation as described in the following section.

4. Nonlinear manifolds

There is no reason to assume that the high dimensional examples would lie on a purely low dimensional linear subspace as assumed by PCA and FA. In order to model the nonlinear low dimensional structure of this data, we generalize a recently-developed algorithm known as locally linear embedding (LLE) (Saul & Roweis, 2003) to handle correspondences.

The basic LLE algorithm is to model the data nonparametrically as consisting of locally linear patches lying on a low dimensional manifold embedded in the high dimensional space. For each vector from a data set $X = \{\boldsymbol{x_i}\}$, the algorithm computes the following:

1. Finds nearest neighbors $\boldsymbol{x_j}$ of $\boldsymbol{x_i}$ based on the Euclidean distance $\|\boldsymbol{x_i} - \boldsymbol{x_j}\|^2$.

2. Computes weights W_{ij} such that $\|\boldsymbol{x_i} - \sum_j W_{ij}\boldsymbol{x_j}\|^2$ is minimized, subject to the constraint $\sum_j W_{ij} = 1$.

Once the local reconstruction weights W are determined, LLE constructs a low dimensional representation of the data by finding low dimensional vectors $Y = \{\boldsymbol{y_i}\}$ which minimize:

$$\sum_i \|\boldsymbol{y_i} - \sum_j W_{ij}\boldsymbol{y_j}\|^2 \tag{4}$$

subject to the constraint $\langle \boldsymbol{y_i}\boldsymbol{y_i}^T \rangle = I$. A nice feature of LLE is that this minimization can be efficiently performed by computing the eigenvectors of $M = (I - W)(I - W)^T$ with the smallest eigenvalues. The low dimensional representation Y can then be formed by taking a small number of these eigenvectors as the rows of Y.

Given two data sets X^1 and X^2, LLE can be performed separately on the two sets by computing the nearest neighbors and local weights W^1 and W^2. Then, two separate low dimensional representations Y^1 and Y^2 can be calculated by diagonalizing $M^1 = (I - W^1)(I - W^1)^T$ and $M^2 = (I - W^2)(I - W^2)^T$ respectively. This is equivalent to minimizing the combined cost:

$$\min \quad \mathrm{tr}(Y^1 - W^1Y^1)(Y^1 - W^1Y^1)^T + \tag{5}$$

$$\mathrm{tr}(Y^2 - W^2Y^2)(Y^2 - W^2Y^2)^T. \tag{6}$$

However, in order to account for correspondences between the two data sets, the two representations should be equivalent for those examples that are in correspondence. If the matrices Y^1 and Y^2 are partitioned into $Y^1 = \begin{bmatrix} Y^1_c & Y^1_s \end{bmatrix}$ and $Y^2 = \begin{bmatrix} Y^2_c & Y^2_s \end{bmatrix}$, then the shared correspondence implies the constraint $Y^1_c = Y^2_c$. The minimization of Equation 6 with this constraint can be efficiently computed by first partitioning M^1 and M^2 as:

$$M^1 = \begin{bmatrix} M^1_{cc} & M^1_{cs} \\ M^1_{sc} & M^1_{ss} \end{bmatrix} \tag{7}$$

$$M^2 = \begin{bmatrix} M^2_{cc} & M^2_{cs} \\ M^2_{sc} & M^2_{ss} \end{bmatrix}. \tag{8}$$

and then finding the eigenvectors with the smallest eigenvalues of:

$$M' = \begin{bmatrix} M^1_{cc} + M^2_{cc} & M^1_{cs} & M^2_{cs} \\ M^1_{sc} & M^1_{ss} & \mathbf{0} \\ M^2_{sc} & \mathbf{0} & M^2_{ss} \end{bmatrix}. \tag{9}$$

This optimization gives rise to low dimensional representations of the two data sets where the points in correspondence are constrained to be equivalent. Intuitively, this causes the two different manifold structures to align with each other to bring these common points into correspondence. This is illustrated on some simulated data in Figure 3 where two different manifolds were sampled to generate the data sets X^1 and X^2. Due to the low sampling density, LLE applied separately to the data sets gives rise to a nonfaithful representation. However, constraining equivalent points in the two data sets to have the same underlying two-dimensional representation causes the underlying uniform manifold structure to become apparent.

We also applied the constrained LLE algorithm to generate the corresponding views of the objects in Figure 1. In order to reconstruct the unknown correspondence of an example in data set X^1, the low dimensional representation in Y^1 is first computed. Then by finding the nearest points in Y^2 to this representation, the corresponding image is reconstructed by interpolating the appropriate examples in X^2. As shown in Figure 2, this technique gives rise to the best

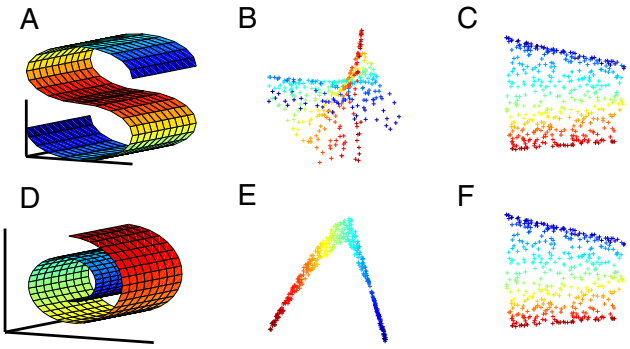

Figure 3. 400 points are each sampled randomly from the S-curve (A) and the Swiss roll (D). Low dimensional coordinates from LLE when applied separately to the S-curve (B) and Swiss roll (E) are not very uniform due to the low sampling density. With the same number of samples and 240 pairs of points in correspondence, the combined LLE representations for the S-curve (C) and Swiss roll (F) use the correspondence between the points to yield a smoother, more uniform representation.

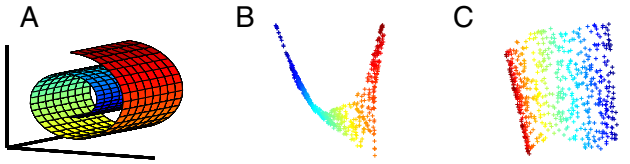

Figure 4. LLE applied to data from 600 points sampled from the Swiss roll in (A). Applying LLE directly on a single data set with all 600 points results in the representation in (B). Splitting the 600 points into two data sets with 480 points each, with 360 points in correspondence and applying the constrained LLE algorithm results in the representation shown in (C).

reconstruction error among all the algorithms that we studied, using a manifold of considerably lower dimensionality ($d = 3$) than the subspaces ($d = 15$) modeled by PCA and FA.

5. Self-correspondence

A convenient way to form correspondences from a single data set is to select overlapping subsets from that data set. Given data X with n examples, we can first choose n_c examples and form X_c. We split the remaining $n - n_c$ examples into two sets X_s^1 and X_s^2. Then we can treat $X^1 = [X_c\ X_s^1]$ and $X^2 = [X_c\ X_s^2]$ as two separate data sets with the first n_c examples in direct correspondence.

The advantage of splitting the data set in this manner is illustrated in Figure 4. When conventional LLE is applied to the single data set, the small sampling of points results in a distorted representation. However, with the same data points split into two data sets, the correspondences places constraints on the low dimensional representation that results in a smoother, more uniform representation.

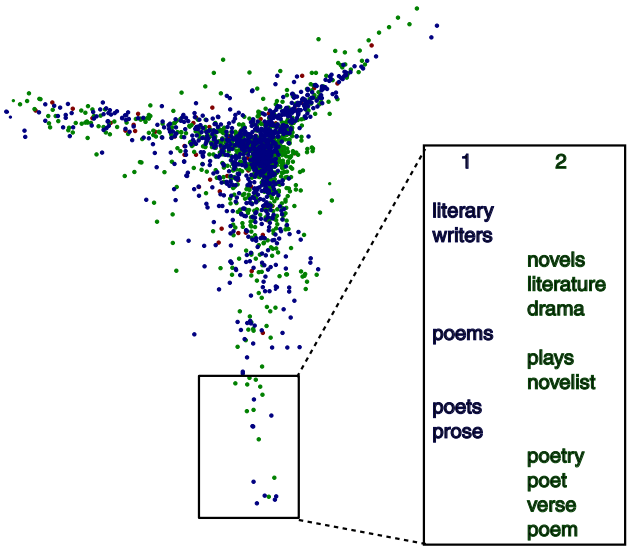

Figure 5. Correspondences used to map related words in encyclopedia articles according to their occurrence in the documents. One set of 968 words and another of 969 words were used with 90 in correspondence. The resulting low dimensional represention maps related words between the two data sets onto each other.

We also performed constrained LLE on a set of documents by splitting the words that occurred in the documents into two data sets. Each word was represented by a feature vector that described the count of that word over the set of the documents. Using only a few common words between the two data sets, the algorithm was able to successfully match related words from the different data sets as illustrated in Figure 5.

6. Discussion

When considering high dimensional examples from disparate data sets, it is valuable to model them as low dimensional manifolds in order to map correspondences between them. PCA and factor analysis can be easily extended to handle corresponding data sets by using the EM algorithm to handle the missing data examples. It should be noted that the PCA model is similar to an earlier algorithm (Tenenbaum & Freeman, 2000) that used a bilinear model to separate style (viewpoint in our image example) from content (identity of object), but the optimization algorithm used was not the same as the EM algorithm.

In order to model manifolds, we see that LLE can be easily extended to handle constraints introduced by correspondences. In our tests, the low dimensional nonlinear structure discovered by contrained LLE gave the best performance in reconstructing the unknown corresponding views

of an object. This approach to mapping correspondences differs from graph matching algorithms that use combinatorial optimization techniques (Barrow & Popplestone, 1971; Gold & Rangarajan, 1996). For certain problems, a combinatorial approach may be appropriate, but it is not clear how combinatorial algorithms would perform with large amounts of missing data or sparsely sampled data.

We also note that high dimensional correspondences can be used to better learn the underlying low dimensional structure of data. This is illustrated by using self-correspondences from a single data set to estimate more faithful representations. Choosing overlapping subsets and using the correspondence information to obtain better estimates is complementary to "bootstrapping" and other techniques for parameter estimation where multiple subsets of a small data set are chosen and assumed to be independent (Efron & Tibshirani, 1993; Niyogi et al., 1998). Also, the algorithm can easily be generalized from two subsets to multiple partitions of a data set. The optimal number of partitions to use for the best representation is currently under investigation.

This work shows that a little supervised knowledge about correspondences can go a long way towards improving the performance of traditional unsupervised learning algorithms. We are currently investigating how these correspondences may possibly be learned as well. We believe that problems with partially labelled correspondences present many new opportunities for research in machine learning.

Acknowledgements

We gratefully acknowledge O. Naroditsky for providing us with the pose correspondence images, and also to G. Chechik, M. I. Jordan, C. J. Taylor and K. Daniilidis for useful discussions.

References

Barrow, H., & Popplestone, R. (1971). Relational descriptions in picture processing. *Machine Intelligence*, *6*, 377–396.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, *39*, 1–38.

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap, monographs on statistics and applied probability, 57*. Chapman and Hall/CRC, Boca Raton, FL.

Gold, S., & Rangarajan, A. (1996). A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*, 377–388.

Hinton, G., Dayan, P., & Revow, M. (1997). Modeling the manifolds of images of handwritten digits. *IEEE Transactions on Neural Networks*, *8*, 65–74.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, *28*, 321–377.

Jolliffe, I. (1986). *Principal component analysis*. Springer-Verlag.

Niyogi, P., Girosi, F., & Poggio, T. (1998). Incorporating prior information in machine learning by creating virtual examples. *Proceedings of the IEEE*, *86*, 2196–2209.

Roweis, S. (1998). EM algorithms for PCA and SPCA. *Advances in Neural Information Processing Systems*, *10*.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*, 2323–2326.

Saul, L., & Rahim, M. (1999). Maximum likelihood and minimum classification error factor analysis for automatic speech recognition. *IEEE Transactions on Speech and Audio Processing*, *8*, 115–125.

Saul, L. K., & Roweis, S. T. (2003). Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, *4*, 119–155.

Seung, H. S., & Lee, D. D. (2000). The manifold ways of perception. *Science, 290*, 2268–2269.

Tenenbaum, J., & Freeman, W. (2000). Separating style and content with bilinear models. *Neural Computation*, *12*, 1247–1283.

Tipping, M., & Bishop, C. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, *61*, 611–622.

A. Factor analysis with missing data

A.1. Latent variable model

The data model has the form of $\boldsymbol{x} = \boldsymbol{\mu} + W\boldsymbol{\xi} + \boldsymbol{\eta}$, where $\boldsymbol{\mu}$ is a size d_x mean vector, W is a d_x by d_ξ factor loading matrix, $\boldsymbol{\xi}$ a size d_ξ hidden vector, and $\boldsymbol{\eta}$ is a size d_ξ noise vector. The random vectors $\boldsymbol{x}$, $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ are assumed to have the following distributions:

$$P(\boldsymbol{\xi}) = \frac{1}{(2\pi)^{d_\xi/2}} \exp\left\{-\frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{\xi}\right\}, \tag{10}$$

$$P(\boldsymbol{x}|\boldsymbol{\xi}) = \frac{1}{(2\pi)^{d_x/2}|\Psi|^{1/2}} \exp\left\{-\frac{1}{2}\left[\boldsymbol{x} - W\boldsymbol{\xi} - \boldsymbol{\mu}\right]^T \Psi^{-1}\left[\boldsymbol{x} - W\boldsymbol{\xi} - \boldsymbol{\mu}\right]\right\}, \text{ and} \tag{11}$$

$$P(\boldsymbol{\eta}) = \frac{1}{(2\pi)^{d_\xi/2}|\Psi|^{1/2}} \exp\left\{-\frac{1}{2}\boldsymbol{\eta}^T\Psi^{-1}\boldsymbol{\eta}\right\}, \tag{12}$$

where Ψ is a diagonal matrix $cov(\eta)$. To perform factor analysis with missing data, we introduce new random vector $\boldsymbol{y}$ of dimension $d_y = d_x + d_\xi$:

$$\boldsymbol{y} = \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{\xi} \end{pmatrix}$$

which has the distribution

$$P(\boldsymbol{y}) = P(\boldsymbol{x}, \boldsymbol{\xi}) = \frac{|A|^{1/2}}{(2\pi)^{d_y/2}} \exp\left\{-\frac{1}{2}\left[\boldsymbol{y} - \boldsymbol{b}\right]^T A\left[\boldsymbol{y} - \boldsymbol{b}\right]\right\}, \tag{13}$$

where the parameters A and b are

$$A = \begin{bmatrix} \Psi^{-1} & -\Psi^{-1}W \\ -W^T\Psi^{-1} & I_{d_\xi} + W^T\Psi^{-1}W \end{bmatrix}, \text{ and } \quad \boldsymbol{b} = \begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{0} \end{pmatrix}. \tag{14}$$

Let *v* and *h* be sets of indices for visible and hidden variables, i.e., $v = \{1, 2, \cdots, d_x\}$ and $h = \{d_x + 1, \cdots, d_y\}$. Also let *u* (for unobservable) be a set of indices in $\{1, 2, \cdots, n_y\}$ for elements of $\boldsymbol{y}$ which are either hidden or missing, and *o* (for observable) be the other indices for observed variables. Note $h \subset u$. Using this notation, A_{uo} is a sub-matrix whose (i, j)th entry is $A_{u_i o_j}$, where u_i and o_j is the ith and the jth element of *u* and *o*, respectively.

Let's partition $\boldsymbol{y}$, $\boldsymbol{b}$, and *A* as:

$$\boldsymbol{y} = \begin{pmatrix} \boldsymbol{y_o} \\ \boldsymbol{y_u} \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} \boldsymbol{b_o} \\ \boldsymbol{b_u} \end{pmatrix}, \quad \text{and} \quad A = \begin{bmatrix} A_{oo} & A_{ou} \\ A_{uo} & A_{uu} \end{bmatrix}. \tag{15}$$

The conditional probability of unobserved variables in $\boldsymbol{y}$ given observed variables in $\boldsymbol{y}$ is, as a function of $\boldsymbol{y_u}$ and the deviations $\boldsymbol{\delta y_u} = \boldsymbol{y_u} - \overline{\boldsymbol{y_u}}$:

$$P(\boldsymbol{y_u}|\boldsymbol{y_o}) = \exp\left\{-\frac{1}{2}\left[\boldsymbol{y_u} - \overline{\boldsymbol{y_u}}\right]^T \overline{\boldsymbol{\delta y_u}\boldsymbol{\delta y_u}^T}^{-1}\left[\boldsymbol{y_u} - \overline{\boldsymbol{y_u}}\right]\right\}, \tag{16}$$

where the overline $\overline{\cdot}$ means conditional expectation $E_{\boldsymbol{y_u}}[\,\cdot\,|\boldsymbol{y_o}]$. After algebraic manipulations, we get the mean $\overline{\boldsymbol{y_u}}$ and the covariance matrix $\overline{\boldsymbol{\delta y_u}\boldsymbol{\delta y_u}^T}$

$$\overline{\boldsymbol{y_u}} = A_{uu}^{-1}\left[A_{uu}\boldsymbol{b_u} - A_{uo}(\boldsymbol{y_o} - \boldsymbol{b_o})\right], \text{ and } \overline{\boldsymbol{\delta y_u}\boldsymbol{\delta y_u}^T} = A_{uu}^{-1}. \tag{17}$$

These results represent the expected values of the unknown variables in the probabilistic factor analysis model.

A.2. EM algorithm

The EM algorithm consists of two steps. The E-step uses the expected values derived in the previous section, and the M-step involves maximizing the auxiliary function Q:

$$\left[\tilde{\boldsymbol{\mu}}, \tilde{W}, \tilde{\Psi}\right] = \arg\max\ Q \tag{18}$$

$$= \arg\max \left\langle -\frac{1}{2}\left[\overline{\boldsymbol{y}} - \tilde{\boldsymbol{\mu}}\right]^T \tilde{A}\left[\overline{\boldsymbol{y}} - \tilde{\boldsymbol{\mu}}\right] - \frac{1}{2}\mathrm{Tr}\left[\tilde{A}\overline{\boldsymbol{\delta y_u}\boldsymbol{\delta y_u}^T}\right] + \frac{1}{2}\log|\tilde{A}|\right\rangle \tag{19}$$

$$= \arg\max \left\{ -\frac{1}{2}\mathrm{Tr}\left(\tilde{A}K\right) + \frac{1}{2}\log|\tilde{A}|\right\}, \tag{20}$$

where K, $\overline{\boldsymbol{y}}$, and $\tilde{\boldsymbol{\mu}}$ are defined as

$$K = \left\langle \left[\overline{\boldsymbol{y}} - \tilde{\boldsymbol{\mu}}\right]\left[\overline{\boldsymbol{y}} - \tilde{\boldsymbol{\mu}}\right]^T + \overline{\boldsymbol{\delta y_u}\boldsymbol{\delta y_u}^T}\right\rangle, \quad \overline{\boldsymbol{y}} = \begin{pmatrix} \boldsymbol{y_o} \\ \overline{\boldsymbol{y_u}} \end{pmatrix} \text{ and } \quad \tilde{\boldsymbol{\mu}} = \begin{pmatrix} \tilde{\boldsymbol{\mu_v}} \\ \mathbf{0} \end{pmatrix} \tag{21}$$

by the assumption that hidden variables $\boldsymbol{\xi}$ are zero-mean. The bracket $\langle\cdot\rangle$ stands for the average over examples. From the Schur complement one can easily verify

$$\left|\tilde{A}\right| = \left|\begin{bmatrix} \tilde{\Psi}^{-1} & -\tilde{\Psi}^{-1}\tilde{W} \\ -\tilde{W}^T\tilde{\Psi}^{-1} & I_{d_\xi} + \tilde{W}^T\tilde{\Psi}^{-1}\tilde{W} \end{bmatrix}\right| = \left|\tilde{\Psi}^{-1}\right|. \tag{22}$$

To further simplify the calculation let R be defined as

$$R = \langle \overline{\boldsymbol{y}\boldsymbol{y}}^T + \overline{\boldsymbol{\delta y}\boldsymbol{\delta y}^T}\rangle\ - \langle\overline{\boldsymbol{y}}\rangle\langle\overline{\boldsymbol{y}}\rangle^T, \tag{23}$$

with

$$\boldsymbol{\delta y} = \begin{pmatrix} \boldsymbol{y_o} - \boldsymbol{y_o} \\ \boldsymbol{y_u} - \overline{\boldsymbol{y_u}} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\delta y_u} \end{pmatrix}. \tag{24}$$

Then, the maximum value of Q is achieved by simultaneously solving

$$\frac{\partial Q}{\partial \tilde{W}} = \frac{\partial Q}{\partial \tilde{\Psi}} = \frac{\partial Q}{\partial \tilde{\boldsymbol{\mu}}} = \mathbf{0}. \tag{25}$$

Using R defined above, the solution to

$$\frac{\partial Q}{\partial \tilde{W}} = \frac{\partial}{\partial \tilde{W}} - \frac{1}{2}\mathrm{Tr}\left\{-\tilde{\Psi}^{-1}\tilde{W}R_{hv} - \tilde{W}^T\tilde{\Psi}^{-1}R_{vh} + (I + \tilde{W}^T\tilde{\Psi}^{-1}\tilde{W})R_{hh}\right\} = \mathbf{0} \text{ is} \tag{26}$$

$$\tilde{W} = R_{vh}(R_{hh})^{-1}, \tag{27}$$

and the solution to

$$\frac{\partial Q}{\partial \Psi^{-1}} = -\frac{1}{2}\mathrm{Tr}\left(R_{vv} - R_{vh}\tilde{W}^T - \tilde{W}R_{hv} + \tilde{W}^T R_{hh}\tilde{W} - \Psi\right) = \mathbf{0} \text{ yields} \tag{28}$$

$$\tilde{\Psi}_{ii} = \left[\begin{pmatrix} I & -\tilde{W} \end{pmatrix} R \begin{pmatrix} I & -\tilde{W} \end{pmatrix}^T\right]_{ii}. \tag{29}$$

Finally, setting $\frac{\partial Q}{\partial \tilde{\boldsymbol{\mu}}_v} = \mathbf{0}$ gives

$$\tilde{\boldsymbol{\mu}} = \begin{pmatrix} I & -\tilde{W} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \langle\overline{\boldsymbol{y}}\rangle. \tag{30}$$

These equations determine the closed form expressions for updating parameters $\tilde{\boldsymbol{\mu}}$, $\tilde{W}$, and $\tilde{\Psi}$ which are guaranteed to monotonically increase the likelihood.