
Label Preserving Dimensionality Reduction

Kristin M. Branson

Computer Science and Engineering
U.C. San Diego
kbranson@cs.ucsd.edu

Abstract

Many tasks, such as face recognition, require learning a classifier from a small number of high dimensional training samples. These tasks suffer from the curse of dimensionality: the number of training samples required to accurately learn a classifier increases exponentially with the dimensionality of the data. One solution to this problem is dimensionality reduction. Common methods for dimensionality reduction such as Principal Component Analysis (PCA) and Fisher's Linear Discriminant Analysis (LDA) have undesirable properties. PCA is an unsupervised algorithm, while LDA can only produce a small number of components and assumes the data is normally distributed. A new algorithm framework proposed in this paper, Structured Principal Component Analysis (SPCA), does not have these undesirable properties. SPCA structures the data in a supervised way, then applies PCA to each structure. SPCA first clusters together similar features of the data, using a supervised similarity measure that makes small, reasonable assumptions about the data. From each cluster of similar features, a small number of components are chosen to represent that cluster. The instantiation of SPCA employed in this paper uses the normalized cut criterion to cluster the features. The similarity measure used is the same-class Chi-squared distance between each feature over all the training data. Finally, PCA is used to extract a small number of components from each cluster. This algorithm was tested on two face recognition databases, with encouraging results.

1 Introduction

Many tasks require learning a classifier from a small number of high dimensional training samples. Because the potential complexity of a classifier increases exponentially with the dimensionality of the data, it is difficult to accurately learn a classifier with data of this type (Duda et al., 2001).

As a concrete example, consider the task of facial expression classification of the Pictures of Facial Affect (POFA) Database (Ekman and Friesen, 1976). This database consists of 240 x 292 greyscale pixel images of 14 actors performing one of six expressions. The dimensionality of each data sample is the number of pixels in the image, 70,080. The goal is to determine a classifier, say a perceptron, which will accurately classify novel images.

A simple classifier like a perceptron has an input unit for each pixel and an output unit for each expression. A weight must be learned for each pair of input and output units. As there are 70,080 inputs and 6 outputs, there are 420,480 weights that must be learned. The number of training samples needed to accurately and confidently estimate these weights grows exponentially with the number of weights (Bishop, 1995). As the POFA database has only 110 training examples, an accurate perceptron cannot be learned from this high dimensional data.

One solution to this problem is dimensionality reduction: choosing a small(er) set of features to represent the data. Two of the most popular methods for dimensionality reduction are Principal Component Analysis (PCA) and Fisher's Linear Discriminant Analysis (LDA). Both PCA and LDA select orthonormal features from all linear transformations of the input features using the mean and covariance statistics of the training data. PCA and LDA differ in that the criterion used to choose the features for LDA is supervised, while the criterion for PCA is unsupervised.

PCA has the obvious disadvantage that it does not use one of the most important features of the training data, namely the class labels. LDA has the disadvantage that it can find at most $c - 1$ components to project onto, where c is the number of classes in the task. For example, in expression recognition, c is 6. Both PCA and LDA only consider the mean and covariance statistics of the training data, thus are both disadvantaged if the distribution of the data is not normal.

In this paper, I propose an algorithm for dimensionality reduction that does not suffer from any of these shortcomings, Structured Principal Component Analysis (SPCA). SPCA is a supervised algorithm that makes small and reasonable assumptions about the distribution of the data. SPCA uses graph segmentation to find the features that best represent the original features for the classification task. SPCA clusters the features so that the similarity of features within a cluster is maximized and the similarity of features in separate clusters is minimized. As a supervised measure of similarity between features is used, the clustering of features is supervised. Each cluster of features is represented by a small number of features. Thus, SPCA produces a small number of features that best represent the original features for the classification task at hand.

2 Background

2.1 Principal Component Analysis

The most common method for dimensionality reduction is Principal Component Analysis (PCA). PCA is an unsupervised algorithm that selects orthonormal features from all linear transformations of the input features. The data is represented as the projection of the original high-dimensional training data on these feature vectors. The projection $\tilde{\mathbf{x}}_i$ of training sample, \mathbf{x}_i , onto the feature vectors represented as columns of the matrix V is:

$$\tilde{\mathbf{x}}_i = V^T \mathbf{x}_i.$$

PCA selects the features that minimize the sum squared error of the distance between a sample and its projection. That is, given training data $D = \mathbf{x}_i, i = 1, \dots, n$, PCA finds the features, the column vectors of V , which minimize

$$J(V) = \frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - V\tilde{\mathbf{x}}_i\|^2.$$

The vectors V that minimize J are also the vectors that maximize the projected scatter, $V^T S V$, where

$$S = \sum_{i=1}^n (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T.$$

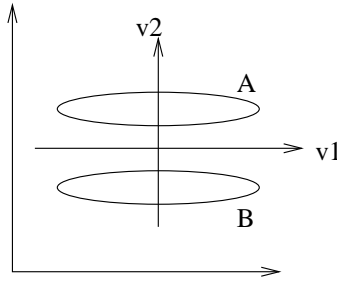


Figure 1: Two classes of data and the direction of the first principal component and the first component selected by LDA.

Thus, the columns of V are the eigenvectors of S , sorted in order of decreasing eigenvalues. Thus, these vectors are also in the direction of the greatest variance of the data. The M features selected are the first M eigenvectors of S (Bishop, 1995).

PCA is an unsupervised algorithm that chooses the components that best represent the data. It does not necessarily choose the components that would be best for discriminating one class of data from another. For example, consider the two classes of data shown in Figure 1. PCA will choose the vector in the direction of greatest variance of all the data, \mathbf{v}_1 . While the classes of data are linearly separable, the classes are not linearly separable when projected on the first principal component.

2.2 Fisher's Linear Discriminant Analysis

Fisher's Linear Discriminant Analysis (LDA) attempts to alleviate this problem. LDA is a supervised method of dimension reduction that attempts to find the linear combination of features which will represent the most information useful for discriminating between classes. LDA will choose the optimal component, \mathbf{v}_2 to discriminate the two classes shown in Figure 1.

Criteria used for LDA choose the projection space with the maximum separation between the data in different classes. LDA does this by maximizing the scatter of the means of the projected data of each class while minimizing the scatter of the projected data within each class. All criteria for LDA try to maximize the between class scatter in the projected space:

$$\tilde{S}_B = \sum_{i=1}^c (\tilde{\mathbf{m}}_i - \tilde{\mathbf{m}})(\tilde{\mathbf{m}}_i - \tilde{\mathbf{m}})^T$$

while minimizing the within class scatter in the projected space:

$$\tilde{S}_W = \sum_{i=1}^c \sum_{\mathbf{x} \in D_i} (\tilde{\mathbf{x}} - \tilde{\mathbf{m}}_i)(\tilde{\mathbf{x}} - \tilde{\mathbf{m}}_i)^T,$$

where $\tilde{\mathbf{m}}_i$ is the mean of the projected data of class i , $\tilde{\mathbf{m}}$ is the mean of all the projected data, and c is the number of classes.

Thus, LDA finds the matrix of column feature vectors, V , that maximize $V^T S_B V$ and minimize $V^T S_W V$, where S_W and S_B are the within and between class scatters of the unprojected data. The standard criterion function maximized is:

$$J(V) = \frac{|V^T S_B V|}{|V^T S_W V|}.$$

Finding V that maximizes $J(V)$ reduces to solving the generalized eigenvector problem: $S_B \mathbf{v} = \lambda S_W \mathbf{v}$ (Duda et al., 2001).

Because the rank of S_B is at most $c - 1$, where c is the number of classes, at most $c - 1$ features can be extracted using LDA. In the example of facial expression classification of the POFA database, this is only five features, which is often too few to result in good generalization. In addition, the criterion function used for LDA is somewhat arbitrary. While it is true that maximizing the between-class variance and minimizing the within-class variance is desirable for separating classes, there are other measures that are as appropriate. In addition, maximizing between-class scatter while minimizing within-class scatter does not exactly translate into the most common criterion function for LDA, involving the ratio of the determinants of these matrices. For example, using the trace of the matrices has been suggested. Finally, LDA uses only the mean and variance statistics of the data in the criterion function used to choose the features. If the data is not normally distributed, then LDA ignores many potential attributes of the data.

3 Structured Principal Component Analysis

In this section, I will discuss the details of the SPCA algorithm. First, I will discuss the graph segmentation used to cluster the original variables of the data. Second, I will discuss the supervised similarity measure used in the graph segmentation algorithm. Third, I will discuss the specifics of the current implementation of SPCA.

3.1 Graph Segmentation Using Eigenvectors

SPCA clusters the features so that features in the same cluster are similar, while features in different clusters are dissimilar (Shi and Malik, 2000a). Thus, SPCA clusters the features so that the intra-cluster affinity is maximized while the inter-cluster affinity is minimized (Hamerly, 2002). The affinity is a measure of group similarity. If $W(u, v)$ is the similarity between feature u and feature v , then the inter-cluster affinity between clusters S_1 and S_2 is:

$$Affinity(S_1, S_2) = \sum_{u \in S_1, v \in S_2} W(u, v).$$

Similarly, the intra-cluster affinity of cluster S is:

$$Affinity(S, S) = \sum_{u \in S, v \in S} W(u, v).$$

The membership vector, \mathbf{y} , that indicates which cluster each feature should be in, can be approximated by solving a generalized eigenvector problem, thus finding the clusteration of features which maximize and minimize the between-cluster and within-cluster affinities is tractable (Shi and Malik, 2000a).

The generalized eigenvector problem focuses on two matrices, the pairwise affinity matrix and the degree matrix (Weiss, 1999). The pairwise affinity matrix, W , is an $N \times N$ matrix, where N is the original number of features in the data. Each element of the affinity matrix is the pairwise similarity $W(u, v)$ between two features, u and v . The degree matrix, D , is a diagonal matrix in which each diagonal element represents the total similarity of a feature to all other features. That is, $D(u, u) = \sum_{v=1}^N W(u, v)$ (Shi and Malik, 2000b).

The cluster membership vector that maximizes within-cluster affinity and minimizes between-cluster affinity is the solution \mathbf{y} with the second smallest eigenvalue λ to the system:

$$(D - W)\mathbf{y} = \lambda D\mathbf{y}.$$

\mathbf{y} is actually an approximate solution, since \mathbf{y} is a continuous-valued vector, not discrete-valued. Thus, thresholding is required to determine the actual cluster membership.

3.2 A Supervised Similarity Measure Between Variables

Two features are similar if the distribution of the training data for each of these features are similar in each class, but not across different classes. As a concrete example, let us consider again the facial expression classification task. For example, consider feature u to be a pixel in the left eye of the image while feature v is the mirror of this pixel in the right eye of the image. As faces are more or less symmetric over a vertical divide, u will look the same as v for each facial expression. Ideally, u and v will be grouped in the same segment. The distance between the distribution of all the happy training samples for u and the distribution of all the happy training samples for v will be small. Similarly, with all other classes, the distributions for u and v will not be far apart. However, since the eyes tend to change a lot in different expressions (for example the eyes in a happy expression are slitted much more than the eyes in a fear expression), the distribution of the happy training samples for feature u will be very different from the distribution of the fearful training samples for feature v . Similarly, with all other combinations of different classes, the distance between the distributions are not necessarily similar if two features are similar. Thus, a supervised measure of the similarity between feature distributions is the sum of the distance between the feature distributions for each class.

The distance metric chosen for this algorithm is the Chi-squared distance between two distributions, f_u and f_v :

$$\chi^2(f_u, f_v) = \frac{1}{2} \int \frac{(f_u(x) - f_v(x))^2}{f_u(x) + f_v(x)} dx.$$

The Chi-squared distance can be approximated using the training data by binning the data. Thus, if 11 equal bins are chosen to cover the range $[-2, 2]$ (where values beyond the limits of this range are placed in the closest end bin), then $f(2)$ is the number of data samples falling in the second bin, data samples in the range $[-1.75, -1.5]$, $f(3)$ is the number of data samples falling in the third bin, data samples in the range $[-1.5, -1.25]$, and so on. The approximation for the Chi-squared distance between features u and v is:

$$\frac{1}{2} \sum_{c \in C} \sum_{i=1}^{n_{bins}} \frac{(f_{cu}(i) - f_{cv}(i))^2}{f_{cu}(i) + f_{cv}(i)},$$

where $f_{cu}(i)$ is the number of data samples of class c for which the value of feature u falls in bin i .

The Chi-squared distance is based on the assumption that the count of the number of samples in a bin is normally distributed. The bottom term in this sum, $f_{cu}(i) + f_{cv}(i)$, is then an approximation of the standard error on the squared difference $(f_{cu}(i) - f_{cv}(i))^2$. The standard error for a normal distribution is the square root of the mean. As the mean is unknown, an approximation is the actual count.

The assumption that the count of the number of samples in a bin is normally distributed is reasonable, as this is really the sum of a number of (somewhat) random features, and the sum of random features approaches a normal distribution. In addition, the Chi-squared distance is meaningful even for non-normal distributions. Thus, the assumptions made by this algorithm are more reasonable and less costly than those made by PCA and LDA.

3.3 Implementation Details

The affinity matrix used in the clustering algorithm is high dimensional for high dimensional data. If N is the dimensionality of the data, then the affinity matrix is an $N \times N$ matrix. Solving a generalized eigenvector problem for $N \times N$ matrices is intractable for large N . However, often the number of training samples used to determine the affinity and

degree matrices, n , is relatively small. The number of actual segments in the variables is closer to n than to N (Fowlkes et al., 2001). Thus, the Nyström method for approximating the solutions to eigenvectors of the normalized affinity matrix are valid in this application. The Nyström method has been shown to work well for graph segmentation of relatively small sets of high dimensional data, and runs in time polynomial in the number of training samples, not the number of dimensions ($O(n^3)$). Thus, the Nyström approximation to the generalized eigenvector problem was implemented.

Once the segmentation of variables has been obtained, each segment must be represented by a small number of variables. SPCA uses the top few Principal Components of each segment to represent the variables in that segment.

4 Experiments

The SPCA algorithm was tested on face recognition tasks. Face recognition can greatly benefit from a supervised method of dimensionality reduction because small details are required for a certain recognition task and there is a lot of noise in faces that must be generalized over. As discussed before, expression classifiers must generalize over identity (and identity classifiers must generalize over expression). In addition, classifiers of both identity and expression recognition must generalize over different lighting and occlusion effects. SPCA was tested on the Ekman and Friesen Pictures of Facial Affect Database and the Yale Face Database.

4.1 The Ekman and Friesen POFA Database

The POFA Database consists of 110 images of 14 actors posing one of six expressions (happy, sad, fearful, angry, surprised, and disgusted), plus neutral. These expressions are considered “universal expressions,” meaning that regardless of culture, the same emotions evoke these expressions. The expressions are posed by actors trained to move some of 44 muscle groups identified by Ekman as displaying the posed expression. All expressions were recognizable by at least 70% of those tested on them.

Expression recognition of the POFA database is difficult because the classifier must be able to generalize over different identities to focus on the expression. This is particularly difficult because the faces of two actors differ more than faces of the same actor posing different expressions. Figure 2 shows images from the POFA database and the differences for different expressions and actors. This database is also difficult, as stated before, because the number of training samples is small, while the dimensionality of the data is high.

To compare SPCA with previous experiments in which PCA and LDA performed well, the same preprocessing of the images of the data set is performed. This preprocessing begins with aligning the images so that the eyes and the bottom of the top row of teeth are in the same position for all images, and cropped so that mostly with z-scoring (normalizing the data so that the mean intensity value for each pixel is zero and the standard deviation is one). Next, the images are subsampled and convolved with Gabor wavelet jets, each composed of 40 Gabor filters of five different scales and eight different orientations, resulting in a 40,600 dimensional vector. Gabor filters are responsive to edges and are biologically inspired. The different orientations and scales aid in improving invariance to small translations and rotations of the data. Finally, the outputs of the Gabor filters are z-scored.

After preprocessing, the dimensionality of the data is reduced using PCA, LDA, or SPCA. A perceptron is learned and tested on novel test images.

PCA performs best when 50 principal components are extracted. Classification is best when a hold out set is used for early stopping, and both weight decay and momentum are used

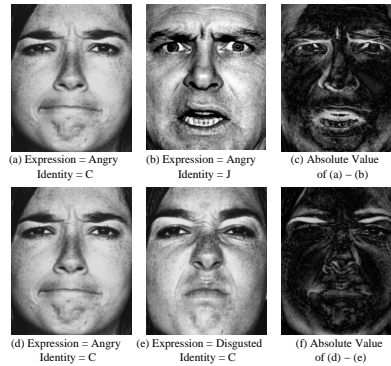


Figure 2: (a) and (b) are images of different actors posing the same expression. (d) and (e) are image of the same actor posing different expressions. (c) is the absolute value of the difference between (a) and (b). (f) is the absolute value of the difference between (d) and (e). The lighter the pixel, the larger the difference.

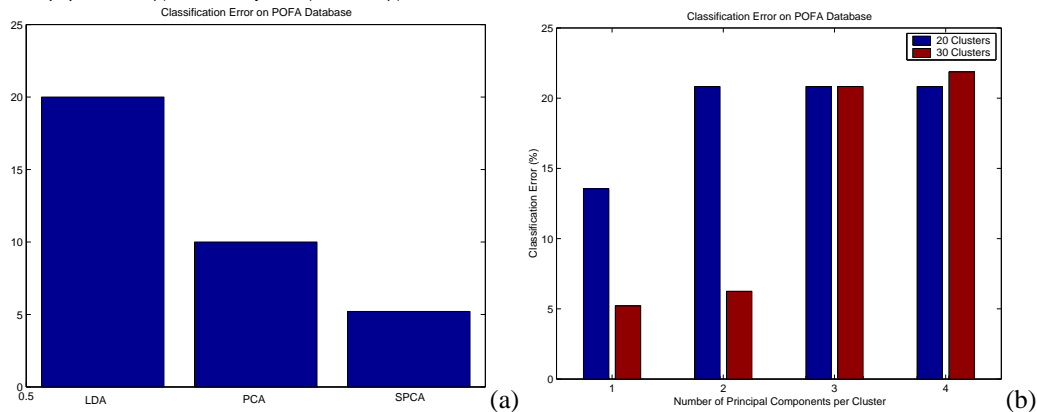


Figure 3: (a) Optimal classification error on the POFA database using LDA, PCA, and SPCA. (b) Classification error using SPCA with 20 or 30 clusters of features, and between one and four Principal Components extracted from each cluster.

in training the perceptron. The perceptron is trained on images of all actors but two, the holdout actor and the verification actor. The perceptron that performs best on the images from the hold out set is used to classify the images from the verification set. Classification error is used to evaluate the performance of the classifier. The optimal performance of PCA is 10% error (Dailey et al., 2000). Using the same training and perceptron set-up, the optimal performance of LDA is 20% error (this performance is achieved using an alternate criterion function; with the standard criterion function, LDA achieves 25% classification error).

SPCA outperforms both LDA and PCA. With the features clustered into 30 clusters and 1 principal component extracted from each cluster, SPCA achieves a 5.2% classification error. The comparison of classification error of LDA, PCA, and SPCA is shown in Figure 3(a). The results of using 20 and 30 clusters of features and extracting between one and four principal components from each of these clusters is shown in Figure 3(b).

As SPCA does not necessarily choose clusters that vary much, I tried adding an additional layer of PCA at the end of the dimensionality reduction. As perceptrons sometimes are

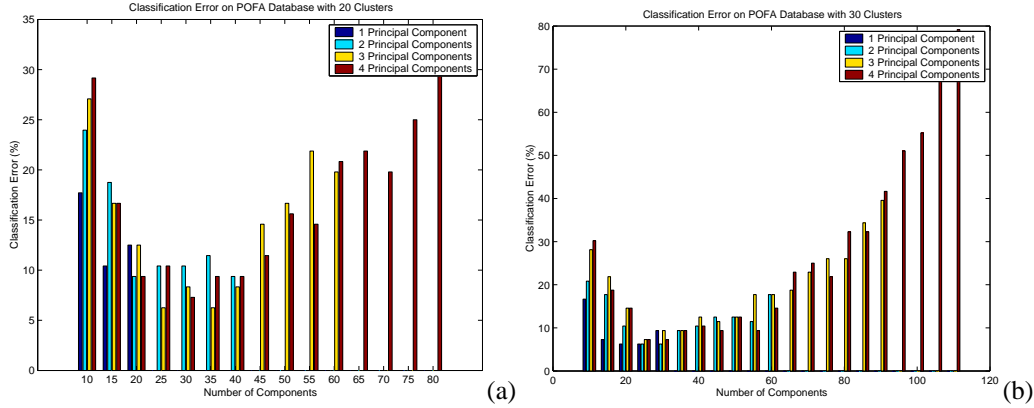


Figure 4: Results of using SPCA plus an extra layer of PCA. Each set of bars in these graphs represents a different number of principal components extracted from each cluster. The x-axis ranges over the number of principal components extracted in the extra layer of PCA. Graph (a) is the results for 20 clusters of features and graph (b) is for 30 clusters of features.

sensitive to unnecessary input (e.g. the weights learned for a feature that is always zeros in the training data are completely random and could be very large), my hypothesis was that this would help the perceptron's performance. While this addition sped up training of the perceptron (the perceptron trained optimally in five iterations), it did not improve classification performance. The optimal performance of SPCA with the extra layer of PCA is 6.25% error. The results using different numbers of clusters, principal components extracted per cluster, and principal components extracted in the extra layer of PCA are shown in Figure 4.

SPCA outperforms PCA with a smaller number of features (30) than PCA requires (50). This suggests that the SPCA-reduced features are more dense with information.

4.2 The Yale Face Database

The Yale Database (Belhumeur and Kriegman, 1997) consists of 165 images of 15 actors. The images of the actors are under different lighting, expressions, and occlusion effects. This data set is difficult, particularly for PCA, because the classifier must generalize over all these distractions (Belhumeur et al., 1996). This dataset was used in the first paper proposing LDA for dimensionality reduction in face recognition tasks, instead of PCA. Thus, LDA performs extremely well on the data set while PCA performs poorly.

As with the POFA data set, the Yale database is first preprocessed. The images are centered and cropped, z-scored, convolved with Gabor wavelet jets, and finally z-scored again. PCA achieved a 10% error, while LDA makes only one error in 165 tests, a 0.61% error (Belhumeur et al., 1996). SPCA makes no errors when the features are clustered into 20 groups and four principal components are extracted from each group. In addition, the perceptron trains extremely quickly on the SPCA-reduced data, as it trains optimally in five iterations of backpropagation. The results of SPCA with different number of principal components extracted from each group is shown in Figure 5. Both LDA and PCA do not perform as well on a close-cropped face in which the outline of the face is excluded. Future work includes testing SPCA on these close-cropped faces.

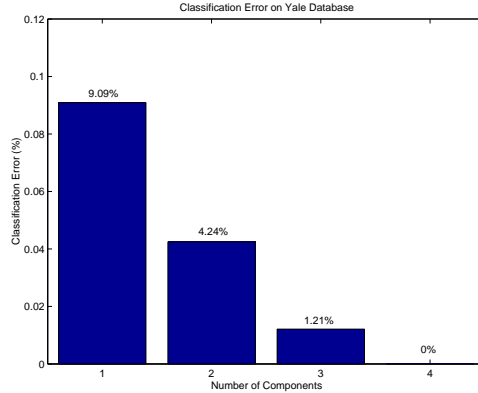


Figure 5: Results of using SPCA on the Yale database, varying the number of Principal Components extracted from each cluster.

5 Conclusion

Structured PCA first structures the original features into a number of clusters of similar features, then applies PCA to each of these clusters of similar features to obtain a small number of features that represent the data. Preliminary results show that the hypothesis that SPCA does not have the faults of PCA and LDA appears to be true, judging by the performance of these three algorithms on two respected databases of faces. Future work includes further testing of SPCA on these data sets. Most critically, the number of clusters of features can be varied.

In addition, other algorithms for dimensionality reduction that more directly address the faults of PCA and LDA can be tried. For example, a hierarchical LDA algorithm is proposed. While LDA is restricted to only $c - 1$ components, a hierarchical LDA algorithm would first find these $c - 1$ components, then find the components in the null space of the first $c - 1$ components, using the same criterion function.

An algorithm that addresses the assumption of normality in LDA would be a distance LDA algorithm which considers the distance between the distributions of different classes. Like the similarity measure used in SPCA, this algorithm would choose the components which minimize the distance between the distributions of the features of samples of the same class. It would also try to maximize the distance between distributions of different classes.

References

- Belhumeur, P. N., Hespanha, J., and Kriegman, D. J. (1996). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In *ECCV (1)*, pages 45–58.
- Belhumeur, P. N. and Kriegman, D. J. (1997). The yale face database.
- Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford University Press, Oxford.
- Dailey, M. N., Cottrell, G. W., and Adolphs, R. (2000). A six-unit network is all you need to discover happiness. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, Mahwah, NJ. Erlbaum.
- Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. Wiley-Interscience, New York.
- Ekman, P. and Friesen, W. (1976). *Pictures of Facial Affect*. Consulting Psychologists, Palo Alto, CA.

- Fowlkes, C., Belongie, S., and Malik, J. (2001). Efficient spatiotemporal grouping using the nystrom method.
- Hamerly, G. (2002). Presentation of 'segmentation using eigenvectors: A unifying view.
- Shi, J. and Malik, J. (2000a). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.
- Shi, J. and Malik, J. (2000b). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.
- Weiss, Y. (1999). Segmentation using eigenvectors: A unifying view. In *ICCV (2)*, pages 975–982.