Structured Principal Component Analysis

Kristin M. Branson and Sameer Agarwal
Deptartment of Computer Science and Engineering
University of California, San Diego
{kbranson,sagarwal}@cs.ucsd.edu

Abstract

Many tasks involving high-dimensional data, such as face recognition, 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. Structured Principal Component Analysis (SPCA) reduces the dimensionality of the data by choosing a small number of features to represent larger sets of similar features. The pairwise similarity of two features is measured using the Chi-squared distance between the joint distributions of the class and the data for each feature. SPCA groups the original features of the data into clusters of similar features using the Normalized Cut algorithm. As features in a cluster are similar and thus redundant, an entire cluster can be represented by a small number of Principal Components extracted from each cluster. SPCA method was tested on two face recognition databases, the Ekman and Friesen Pictures of Facial Affect Database and the Yale Face Database, with encouraging results.

1 Introduction

Many tasks require learning a classifier from a small number of high dimensional training samples. These tasks are particularly difficult because the potential complexity of a classifier increases exponentially with the dimensionality of the data.

For example, consider the task of learning a facial expression classifier from the Pictures of Facial Affect (POFA) Database (Ekman and Friesen, 1976). This database consists of 240 x 292 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. $70,080$ x $6 = 420,480$ weights must be learned, one for each pair of input and output unit. 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 directly 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. This solution is effective if there is redundancy in the data. There is a high amount of redundancy in the pixels of face images. Not only are pixels in the same region of the face similar, but faces are nearly symmetric – an entire half of

the face is essentially redundant. The classical approaches to dimensionality reduction are Principal Component Analysis (PCA) and Fisher's Linear Discriminant Analysis (LDA).

PCA is an unsupervised method that selects, from all linear transformations of the original features, the orthonormal features that minimize the sum-squared difference between the original data and the values of the data projected along these orthonormal features. This is equivalent to maximizing the variance of the projected data. The data is represented as the projection of the original high-dimensional training data on these feature vectors. While PCA chooses the features that best represent the data, it does not necessarily choose the features that best discriminate the data. For example, if pixel p of a face image varies a lot for each actor but not for each expression, p will be highly represented in the components selected by PCA, even though it does not aid in discriminating one expression from another.

LDA attempts to alleviate this problem by using a supervised criterion to choose a set of orthonormal features from all linear transformations of the original features. The orthonormal features are selected to minimize the variance of the data within each class while maximizing the variance of the means of each class of data. The standard tradeoff between these two goals is to maximize the quotient: the variance of the means of each class divided by the summed variance within each class. LDA only depends on the mean and variance of the data. These two statistics are sufficient to describe the data only if the data is normally distributed. If this assumption does not hold, then it is not clear that LDA is optimizing the right criterion. For example, there are two types of smiles represented in the POFA database, smiles in which the teeth show and smiles in which they do not. A pixel in the smile is thus bimodally distributed for samples in the happy class. In addition, LDA is limited in the maximum number of features it can select. LDA can produce at most $c-1$ features, where c is the number of classes. In the case of expression recognition, there are only six classes, thus LDA can only determine five features.

A new algorithm proposed in this paper, Structured Principal Component Analysis (SPCA), reduces the dimensionality of the data by replacing a large set of similar features by a small number of highly representative features. SPCA uses a supervised measure of similarity of two features, the Chi-squared distance between the joint distributions of the class label and the data for each feature. Thus, SPCA is a supervised algorithm that makes small, reasonable assumptions about the data. SPCA structures the original features of the data into clusters of similar features using the Normalized Cut algorithm. SPCA extracts the representative principal components from each cluster. Thus, SPCA produces a small number of features that best represent the original features for the classification task at hand.

2 SPCA Algorithm Description

SPCA is actually an algorithmic framework. The measure used to estimate pairwise similarity, the clustering algorithm used to group similar variables, and the method used to choose a representative feature from each cluster can all be varied. In this section, we describe the instantiation of the SPCA framework we have experimented with.

2.1 A Supervised Similarity Measure Between Variables

A supervised measure of the similarity of two features is required. Thus the similarity of two features u and v is the similarity of the pairs (u,c) and (v,c), where c is the class label. Two random variables, u and v are the same if they have the same distribution, hence the distance between these variables can be measured by the Chi-squared distance between their respective distributions, f_u and f_v, defined as

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

Two pairs of random variables are the same if the distribution of each pair is the same, that is if the joint distributions of the pairs of random variables are the same. The distance between

these pairs of random variables can be measured by the Chi-squared distance between their respective joint distributions. If we consider u, v, and c to be random variables, then the supervised distance between features u and v is the Chi-squared distance between the joint distributions of the pairs (u,c) and (v,c):

$$\begin{aligned} d(u,v) = \chi^2(f_{uc}, f_{vc}) &= \frac{1}{2}\int\int \frac{(f_{uc}(x,c') - f_{vc}(x,c'))^2}{f_{uc}(x,c') + f_{vc}(x,c')} dx dc' \\ &= \frac{1}{2}\sum_{c'}\int \frac{(f_{uc}(x,c') - f_{vc}(x,c'))^2}{f_{uc}(x,c') + f_{vc}(x,c')} dx \end{aligned}$$

As $f_{uc}(x,c') = p(x|c=c')p(c')$, this is equivalent to

$$d(u,v) = \sum_{c'}\left[\frac{1}{2}\int \frac{(f_{u|c}(x,c') - f_{v|c}(x,c'))^2}{f_{u|c}(x,c') + f_{v|c}(x,c')} dx\right] p(c') = \sum_{c'}\chi^2(f_{u|c}, f_{v|c})p(c')$$

Thus, the pairwise distance between two variables is defined to be the weighted sum of the Chi-squared distances between their class-conditional distributions. Since in practice we only have samples from these distributions available to us, we replace the integral in the above expression by a sum and the analytical densities with empirical histograms to get

$$d(u,v) = \frac{1}{2}\sum_{c}\sum_{i=1}^{nbins} \frac{(f_{u|c}(i) - f_{v|c}(i))^2}{f_{u|c}(i) + f_{v|c}(i)} p(c)$$

where $f_{u|c}(i)$ is the fraction of data samples of class c for which the value of feature u falls in bin i and p(c) is the relative frequency of class c.

To further justify the choice of the class-conditional distance, let us consider a concrete example from facial expression classification in which feature u is a pixel in the left eye of the image and feature v is the mirror of this pixel in the right eye of the image. As faces are nearly symmetric over a vertical divide, u has nearly the same intensity as v in each image. Since u does not provide much additional information given v (and vice-versa), ideally u and v will be grouped in the same cluster.

The distribution of all the face images of class c for feature u, $f(u|c)$ is similar to the distribution of all the face images of class c for feature v, $f(v|c)$. For example, the distribution of the happy face images for feature u is similar to the distribution of the happy face images for feature v. Thus, in this example we can require that similar features have similar within-class distributions. However, as eyes express emotion, the eyes change in different expressions (for example the eyes in a happy expression are slitted more than the eyes in a fear expression). The distribution of the happy training samples for feature u will be different from the distribution of the fearful training samples for feature v. It does not make sense to require that similar features have similar class-conditional distributions across classes (i.e. requiring $f_{u|happy}$ be near $f_{v|fearful}$). Thus, intuitively, the summed class-conditional distances between distributions makes sense.

We chose the Chi-squared distance because it is simple and makes small, reasonable assumptions about the data. The Chi-squared distance is based on the assumption that the fraction of the data samples in a bin is normally distributed. The standard error for a normal distribution is the square root of the mean. As the mean is unknown, an approximation is the observed fraction of data samples in a bin. The bottom term in this sum, $f_{u|c}(i) + f_{v|c}(i)$, is then an approximation of the standard error on the squared difference $(f_{u|c}(i) - f_{v|c}(i))^2$.

The assumption that the fraction of samples in a bin is normally distributed is reasonable, as this is the normalized sum of a number of random variables, which in the limit approaches a normal distribution. In addition, the Chi-squared distance is meaningful even for non-normal distributions. Only the measure of the standard error, the denominator in the Chi-squared term is based on the assumption of normality. Thus, the assumptions made by this algorithm are more reasonable and less costly than those made by LDA.

2.2 Graph Segmentation Using the Normalized Cut Criterion

SPCA uses the Normalized Cut algorithm to cluster the features so that features in the same cluster are similar, while features in different clusters are dissimilar (Shi and Malik, 2000). Thus, SPCA clusters the features so that the intra-cluster affinity is maximized while the inter-cluster affinity is minimized, where affinity is a measure of group similarity. The similarity between features u and v is inversely proportional to the distance between them, $d(u,v)$: $W(u,v) = e^{-d(u,v)^2/\sigma}$ (σ is a constant that describes what distances are considered far). The inter-cluster affinity between clusters S_1 and S_2 is:

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

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

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

The criterion function minimized by Normalized Cut is:

$$NCut(S_1,S_2) = Aff(S_1,S_2)\left(\frac{1}{Aff(S_1,S_1 \cup S_2)} + \frac{1}{Aff(S_2,S_1 \cup S_2)}\right).$$

This quantity increases with inter-cluster affinity and decreases with intra-cluster affinity.

The membership vector, $\mathbf{y}$, that indicates which cluster each feature should be in, can be approximated by solving a generalized eigenvector problem, $W\mathbf{y} = \lambda D\mathbf{y}$. The pairwise affinity matrix, W, is an N x 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, $D(u,u) = \sum_{v=1}^{N} W(u,v)$ (Weiss, 1999).

The above formulation can be extended to a k-partitioning of the graph by using additional eigenvectors (Malik et al., 2001; Ng et al., 2002). We do so by stacking the 2^{nd} to the k^{th} eigenvectors columnwise, normalizing the rows of the resulting matrix, and performing k-means clustering on them.

Given that our data is high-dimensional, solving the eigenvector problem is a computationally intensive task. However, our high dimensional data is highly redundant, i.e. there are a large number of features in our data that are similar to each other, implying that a number of rows of our weight matrix W are similar to each other. Having made this observation, we approximate the eigenvector decomposition by solving the problem for a random sample from the data and extrapolating the resulting eigenvectors to the full dataset. This is known as the Nyström approximation. The original eigenvector problem has complexity $O(N^3)$ in the dimensionality of the data. Using the Nyström approximation we can compute the eigenvalue decomposition in $O(s^3N)$, where s is the number of samples used. Empirical evidence shows that for data with a clear clustering structure, a fairly small number of samples can be used to approximate the eigenvectors to a small error (Fowlkes et al., 2001).

2.3 Representation of Each Cluster

SPCA clusters the features of the data into groups that, because of their high affinity, can be represented by a small number of components. To reduce the dimensionality of the data, all the features in a cluster are represented by a small number of features. The concise representation closest to the actual data in each cluster is the top principal components of the data. PCA chooses the components which minimize the sum-squared distance between the projected data and the original data. These components are the eigenvectors of the covariance matrix, ranked in order of the corresponding eigenvalue. Thus, there are two parameters in SPCA: the number of clusters and the number of principal components extracted from each cluster.

3 Experiments

The SPCA algorithm was compared to PCA and LDA on three sets of data. The first set described is a synthetic set designed to demonstrate the weaknesses of PCA and LDA. The second set is the Ekman and Friesen POFA database, with the task of expression recognition. The third set is the Yale Face database, with the task of identity recognition. SPCA achieves 100% accuracy on the synthetic data, compared to SPCA and LDA which did no better than chance. SPCA also outperforms PCA and LDA on the POFA database. SPCA outperforms PCA on the Yale database and has similar performance to LDA.

3.1 Synthetic Data

PCA and LDA both have weaknesses that limit their effectiveness. If a feature with high variance over the data is uncorrelated with the class labels of the data, PCA will highly represent this feature because of its variance, perhaps neglecting features with smaller variance but more correlated with the classification of the data.

Recall that LDA assumes the class-conditional distribution of the data over each feature is normal. Suppose the data does not have this property, for instance if a feature's data for one class is bimodally distributed and for another class is normally distributed. This is the case for the pixels in the smiles (which may or may not show teeth) of happy faces versus pixels in the mouths of sad faces. As LDA chooses the components that separate the class means as much as possible, it will choose to offset the means of the bimodal and normal distributions. This could result in one of the modes of the bimodal distribution being projected to nearly the same value as the mean of the normal distribution.

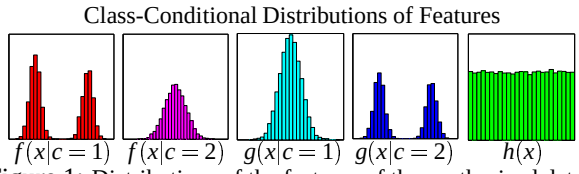


Figure 1: Distributions of the features of the synthesized data.

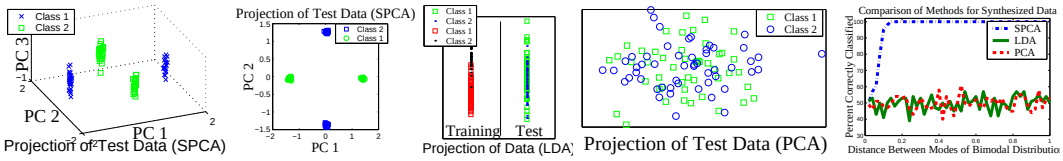
With these limitations in mind, we synthesized 100 training and 100 test samples, all i.i.d. Each sample has 1000 features with three possible distributions, $f(x|c)$, $g(x|c)$, and $h(x|c)$. Only the features with distribution f or g are useful in classification. These distributions are shown in Figure 1. $f(x|c=1)$ and $g(x|c=2)$ are bimodal distributions, with modes ± 0.5 and a standard deviation of 1. $f(x|c=2)$ and $g(x|c=1)$ are normal distribution with mean 0 and standard deviation 1. $h(x)$ is uniformally distributed between 0 and 1. 100 features have distribution f, 100 features have distribution g, and 800 features have distribution h. The optimal dimensionality reduction technique for this data set would ignore all 800 features of distribution h and use any of the features of distribution f or g.

Figure 2 shows the projection of the data on the features chosen by SPCA, LDA, and PCA. When grouping the data into three clusters, SPCA put all but two of the features with distribution f in one cluster, all but one of the features with distribution g in the second cluster, and all the rest of the features in the third cluster. Thus the first and second principal components generated by SPCA are useful in discriminating the data, while the third is not. SPCA performs equally well when only two clusters of the features are found.

As hypothesized, LDA was not able to separate the test data. It was able to find a projection to separate the training data, but this projection relied heavily on the features of distribution h which aren't correlated with the classification. Thus, when generalizing to the test data, LDA fails.

PCA was distracted by the 800 features of distribution h that were not correlated with the classification, and thus was unable to separate the training data or the test data.

In fact, SPCA performs well while the other two algorithms fail on data in which the



(a) (b) (c) (d) (e)

Figure 2: (a) Projection of the test data on the features chosen by SPCA, 3 clusters (b) 2 clusters (c) Projection of the training data and test data on the feature chosen by LDA. (d) Projection of the test data on the top two Principal Components chosen by PCA. (e) Results of SPCA, LDA, and PCA followed by a nearest neighbor classifier on classifying the synthetic data, with varying distance between the modes of the bimodal distributions.

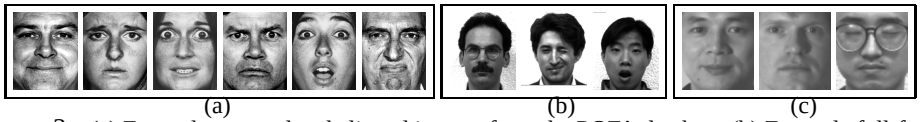
separation between the modes of the bimodal distribution is small. For separations greater than 0.1, SPCA achieves 100% accuracy using a nearest-neighbor classifier. No matter how small the separation, LDA and PCA are not able to separate the data, despite the distributions approaching a normal distribution, as shown in Figure 3.

These experiments on the synthetic data set show that SPCA is robust to features that are uncorrelated with classification, unlike PCA. They also show that SPCA is robust to non-normal distributions of the data, unlike LDA.

3.2 The Ekman and Friesen POFA Database

SPCA, PCA, and LDA were tested on the Ekman and Friesen Database of Pictures of Facial Affect (Ekman and Friesen, 1976). This data set includes 14 trained actors posing six expressions: Happiness, Sadness, Fear, Anger, Surprise, and Disgust (plus Neutral). There are 110 greyscale images in this data set, 96 of which are not neutral. Examples from the Ekman and Friesen POFA are shown in Figure 3.

An expression classifier must generalize over identity and concentrate only on the expression in an image. This is particularly difficult because the difference between the images of two different actors posing the same expression is greater than the difference between the same actor posing two different expressions. As PCA selects the features in the direction of greatest variance, these will encode more for identity than expression. A supervised algorithm would be able to find a more accurate and concise representation that is tailored to expression recognition. However, PCA significantly outperforms LDA, by a margin of 10% accuracy. We hypothesized that this was partially due to the limited number of components LDA can extract ($6 - 1 = 5$). Even trying different criterion functions which allow LDA to produce more features does not greatly improve LDA's performance.



(a) (b) (c)

Figure 3: (a) Example cropped and aligned images from the POFA database (b) Example full-face images from the Yale database (c) Example closely-cropped images from the Yale database

To compare SPCA with previous experiments in which PCA and LDA performed well, we performed the image preprocessing. The images were aligned so that the eyes and the bottom of the top row of teeth were in the same position for all images, and cropped inside the contours of the face. Aligning the images is necessary because PCA and LDA are extremely sensitive to small translations in the images. Cropping the images is necessary to avoid confusing PCA with uninformative data, like the background and hair. Next, the images were 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 lines and 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 were z-scored (normalized

so that the mean intensity value for each pixel is zero and the standard deviation is one). After preprocessing, the dimensionality of the data is reduced using PCA, LDA, or SPCA. A perceptron is learned from images of 12 of the actors, and training is stopped at the best performance on a held out actor. The perceptron is evaluated on images of a novel actor.

SPCA only finds clusters of features with high affinity, not necessarily clusters which are important to classification. Clusters differ in number of elements and correlation with the classification, yet the number of principal components extracted from each is the same. Thus, each cluster is weighted equally in the output features from SPCA. For the POFA data set, we added an extra layer of PCA to weight the principal components extracted by SPCA by the amount of variance of the data projected on them. Thus, if k clusters are found and n principal components are extracted from each cluster, then PCA is performed on the projection of the data onto the kn principal components extracted by SPCA.

This extra layer proved necessary when using a perceptron for classification, as a perceptron is easily influenced by input variables that have small variance in the training data. For example, suppose a feature has a nearly constant value for all the training data and a slightly higher value for one training sample of class c. The perceptron will find this variable useful in determining class c from other classes, and thus could weight its inputs to classify an example as class c if ever the value of this variable differs from the mean. If the inconsistent value for this variable for one training sample is merely noise, then the perceptron will mistake all test examples with an inconsistent value for this variable as class c.

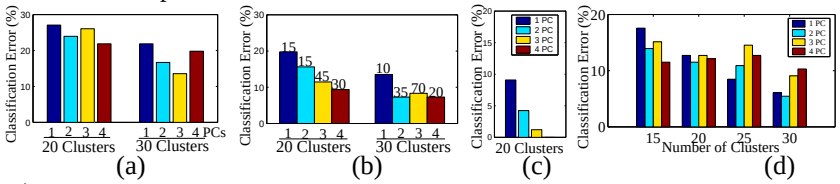


Figure 4: Comparison of parameter settings on POFA and Yale databases. (a) POFA, 20 and 30 clusters, without an extra layer of PCA (b) POFA, 20 and 30 clusters, with an extra layer of PCA. The numbers above each bar are the number of principal components extracted in the extra layer of PCA (c) Yale, Full-face images (d) Yale, Closely-Cropped images

With an extra layer of PCA added, SPCA achieves 92.7% accuracy on this data set, compared to 90% accuracy achieved by PCA (using 50 principal components), and 79.3% accuracy achieved by LDA. These are the optimal results obtained by SPCA, PCA and LDA.

SPCA proved to be relatively insensitive to the number of clusters and the number of principal components extracted from each cluster. Figure 4(a) and (b) show the results of varying these parameters. While SPCA performs better with 30 clusters than 20 clusters, the classification error difference is small (only 2%). In addition, using 30 clusters, optimal results are obtained extracting two and four principal components from each cluster, and extracting three principal components is only 1% worse in classification error.

3.3 The Yale Face Database

The Yale Database (Belhumeur and Kriegman, 1997) consists of images of 15 actors under 11 different conditions, including different lighting, facial expressions, and occlusion effects. Identity recognition is difficult, particularly for PCA, because the classifier must generalize over all these distractions (Belhumeur et al., 1996). This data set was used in the first paper proposing LDA instead of PCA for dimensionality reduction in face recognition tasks. Thus, LDA performs extremely well on the data set while PCA performs poorly.

Two experiments were performed, one in which the images were cropped outside the face contour (full-face images) and one in which the images were cropped inside the face contour (closely-cropped images). Examples of each are shown in Figure 3. The preprocessing of the Yale database wass the same as that of the POFA database (aligning, cropping, Ga-

bor filtering, and z-scoring). After preprocessing, the dimensionality of the data is reduced using PCA, LDA, or SPCA. A perceptron is trained by packpropagation on 164 of the samples and tested on a novel image.

SPCA achieves 100% accuracy on the full-face samples, compared to LDA which obtains 99.4% classification accuracy and PCA which obtains 10% classification accuracy. On the closely-cropped samples, LDA outperforms SPCA. LDA achieves 97% classification accuracy, compared to SPCA with 94.6% accuracy and PCA with 76.4% accuracy. A comparison of the effects of the parameters for SPCA is shown in Figure 4(c) and (d).

4 Discussion

SPCA uses a supervised measure of similarity to cluster the features into groups of high intra-cluster affinity and low inter-cluster affinity. It extracts a small number of principal components from each cluster to represent the data. Experimentally, we have shown that the supervised measure of similarity allows SPCA to distinguish features that are correlated with the classification from those that are not. Because of this, SPCA outperforms PCA in all experiments. We have also shown that when the assumptions made by LDA do not hold, LDA performs very poorly. In these cases, we have experimentally shown that because the assumptions made by the similarity measure used by SPCA are small and reasonable, SPCA outperforms LDA. In addition, we hypothesize that additional experimentation will non-aligned databases will show that SPCA is more robust than PCA and LDA to small translations and rotations in the images.

As stated earlier, SPCA is actually a versatile framework of algorithms. In the future, we hope to experiment with other instantiations, including different methods of representing the features of each cluster. Instead of selecting from the linear combinations of the features in a cluster, we could select directly from the features in the cluster. This would be useful in applications (e.g. medical applications) in which feature selection is desired and linear combinations of features are useless.

Finally, we believe that the distance measure chosen for SPCA could be applied to LDA. In such an algorithm, the data would be projected onto the feature space which maximizes the Chi-squared distance between the distributions of the data of each class and minimizes the Chi-squared distance of the distributions within each class.

Acknowledgements

We would like to thank Serge Belongie, Gary Cottrell and GURU, and Charles Elkan for helpful discussion and advice.

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.

Ekman, P. and Friesen, W. (1976). *Pictures of Facial Affect*. Consulting Psychologists, Palo Alto.

Fowlkes, C., Belongie, S., and Malik, J. (2001). Efficient spatiotemporal grouping using the nystrom method.

Malik, J., Belongie, S., Leung, T. K., and Shi, J. (2001). Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. *NIPS 14*.

Shi, J. and Malik, J. (2000). 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.