2D-PCA

Literature Review and Application

Garrett Bingham

University of North Carolina Wilmington

June 23, 2017

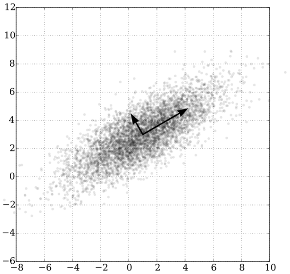
Table of contents

Overview

Review of (1D) PCA

Principal Component Analysis (PCA) transforms the data to a new coordinate system so that the greatest amount of variance lies on the first coordinate, the second greatest on the second coordinate, and so on.

Figure: PCA on Example Dataset

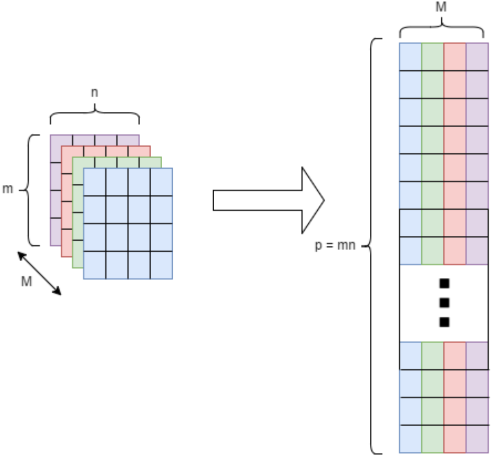


https://commons.wikimedia.org/wiki/File:GaussianScatterPCA.svg

Review of (1D) PCA

When PCA is applied to image analysis, normally the set of M images has to be first vectorized to one $p \times M$ matrix. If we center this matrix and call it $\boldsymbol{Z}$, then the principal components are the normalized eigenvalues of $\boldsymbol{Z}^T\boldsymbol{Z}$.

$\boldsymbol{Z}^T\boldsymbol{Z}$ has dimension $p \times p$, which is quite large. Thus, it is computationally expensive to calculate it explicitly.



What is 2D-PCA?

What sets 2D-PCA apart from conventional PCA is that it is based on 2D matrices rather than 1D vectors. While the covariance matrix ($\boldsymbol{Z}^T\boldsymbol{Z}$) of PCA is large ($p \times p$), 2D-PCA constructs an *image covariance matrix* directly from the original images.

The main advantage of 2D-PCA over PCA is that the covariance matrix is much smaller ($n \times n$). This means it is easier to calculate it accurately, and that it is computationally cheaper to determine the corresponding eigenvectors.

Problem Statement

Let $\boldsymbol{X}$ be an n-dimensional unit column vector. Given a set of $m \times n$ images $\boldsymbol{A}_i$, $i = 1, 2, \ldots, M$, we project image $\boldsymbol{A}_i$ onto $\boldsymbol{X}$ and obtain an m-dimensional projected vector $\boldsymbol{Y}_i$, called the projected feature vector of image $\boldsymbol{A}_i$.

$$\boldsymbol{Y}_i = \boldsymbol{A}_i\boldsymbol{X} \tag{1}$$

We want to choose $\boldsymbol{X}$ so that the projected images are as spread out as possible. If $\boldsymbol{S}_x$ denotes the covariance matrix of the projected feature vectors $\boldsymbol{Y}_i$, then we will try to maximize $J(\boldsymbol{X})$, where

$$J(\boldsymbol{X}) = \text{tr}(\boldsymbol{S}_x) \tag{2}$$

Maximizing $J(\boldsymbol{X}) = \text{tr}(\boldsymbol{S}_x)$

$$\begin{aligned} \boldsymbol{S}_x &= E(\boldsymbol{Y} - E\boldsymbol{Y})(\boldsymbol{Y} - E\boldsymbol{Y})^T \\ &= E(\boldsymbol{AX} - E\boldsymbol{AX})(\boldsymbol{AX} - E\boldsymbol{AX})^T \\ &= E[(\boldsymbol{A} - E\boldsymbol{A})\boldsymbol{X}][(\boldsymbol{A} - E\boldsymbol{A})\boldsymbol{X}]^T. \end{aligned} \tag{3}$$

If we define

$$\boldsymbol{G}_t = E[(\boldsymbol{A} - E\boldsymbol{A})^T(\boldsymbol{A} - E\boldsymbol{A})], \tag{4}$$

then

$$\text{tr}(\boldsymbol{S}_x) = \boldsymbol{X}^T\boldsymbol{G}_t\boldsymbol{X}. \tag{5}$$

Principal Component Vectors

Let $\bar{\boldsymbol{A}}$ denote the average image. Then $\boldsymbol{G}_t$ can be evaluated as

$$\boldsymbol{G}_t = \frac{1}{M} \sum_{j=1}^{M} (\boldsymbol{A}_j - \bar{\boldsymbol{A}})^T (\boldsymbol{A}_j - \bar{\boldsymbol{A}}). \tag{6}$$

The $\boldsymbol{X}$ that maximizes $J(\boldsymbol{X})$ is the eigenvector of $\boldsymbol{G}_t$ corresponding to the largest eigenvalue. Our first principal component of the image $\boldsymbol{A}_i$ is then $\boldsymbol{Y}_{i1} = \boldsymbol{A}_i \boldsymbol{X}_1$, and the following principal components $\boldsymbol{Y}_{ij} = \boldsymbol{A}_i \boldsymbol{X}_j$ correspond to the next largest eigenvalues.

Note that in 2D-PCA, the principal components are vectors, not scalars.

Application

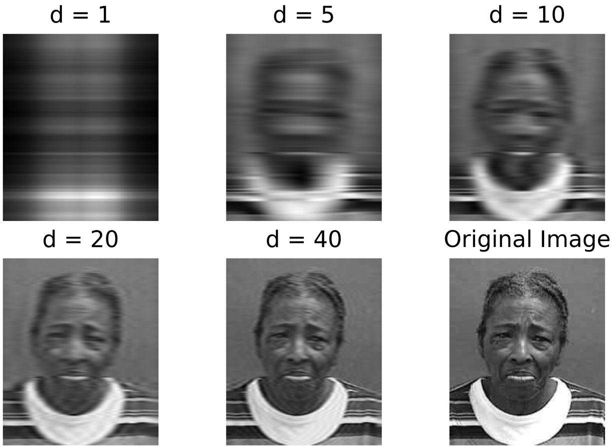
Methodology

In PCA, the principal components and eigenvectors can be combined to reconstruct the original data. Similarly, in 2D-PCA we can reconstruct face image $\boldsymbol{A}_i$ from $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_d$, which are the first d eigenvectors of $\boldsymbol{G}_t$, and $\boldsymbol{Y}_{i1}, \boldsymbol{Y}_{i2}, \ldots, \boldsymbol{Y}_{id}$, the first d principal components of image $\boldsymbol{A}_i$.

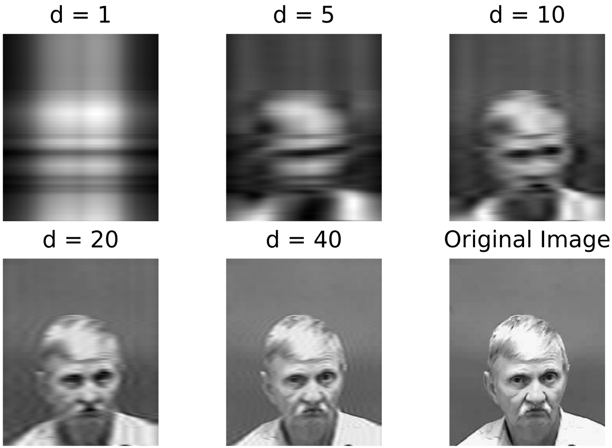
$$\tilde{\boldsymbol{A}}_i = \sum_{k=1}^{d} \boldsymbol{Y}_{ik} \boldsymbol{X}_k^T. \tag{7}$$

$\tilde{\boldsymbol{A}}_i$ is the reconstructed subimage of $\boldsymbol{A}_i$. If $d = n$, then we get perfect reconstruction, and $\tilde{\boldsymbol{A}}_i = \boldsymbol{A}_i$.

Results



Results



Future Studies

1D vs 2D

Wei-Shi Zheng, J.H. Lai, and Stan Z. Li. 1d-lda vs. 2d-lda: when is vector-based linear discriminant analysis better than matrix-based? *Pattern Recognition*, 41(7):2156–2172, 2008.

Much like how we extended PCA to 2D-PCA, this article extends LDA to 2D-LDA. It also contrasts the two methods, which is helpful when trying to decide which to implement. The 2D models may be more accurate and computationally efficient, but in image processing they can miss out on some of the local structure of the data.

nD-PCA

Hongchuan Yu and M. Bennamoun. 1d-pca, 2d-pca to nd-pca. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 4, pages 181–184, 2006.

2D-PCA is further generalized to n dimensions. This isn't directly applicable to us, since we're working with 2D images.

2D Kernel PCA

V. D. M. Nhat and S. Lee. Kernel-based 2dpca for face recognition. In *2007 IEEE International Symposium on Signal Processing and Information Technology*, pages 35–39, December 2007.

Making use of the kernel trick allows us to perform 2D-PCA in a high dimensional space where the data is better separated.

Blockwise 2D Kernel PCA

Armin Eftekhari, Mohamad Forouzanfar, Hamid Abrishami Moghaddam, and Javad Alirezaie. Block-wise 2d kernel pca/lda for face recognition. *Information Processing Letters*, 110(17):761–766, 2010.

2D Kernel PCA has certain computational challenges. Blockwise 2D Kernel PCA breaks an image up into blocks, and 2D Kernel PCA is performed on each block. This drastically lowers computation time, and also captures some of the local structure of the data. Blockwise 2D Kernel LDA is also discussed.

Supervised PCA

Elnaz Barshan, Ali Ghodsi, Zohreh Azimifar, and Mansoor Zolghadri Jahromi. Supervised principal component analysis: visualization, classification and regression on subspaces and submanifolds. *Pattern Recognition*, 44(7):1357–1371, 2011.

Traditional PCA seeks to maximize the variance of the data, without regard to data class. If the classes of the data points are known, then Supervised PCA can be used to maximize *dependency* between the data points and their corresponding classes.

Correlation Tensor Analysis

Y. Fu and T. S. Huang. Image classification using correlation tensor analysis. *IEEE Transactions on Image Processing*, 17(2):226–234, February 2008.

Correlation Tensor Analysis is like a combination between 2D-PCA and Supervised PCA. The covariate data is left in a tensor instead of being vectorized. Class labels are also utilized like in Supervised PCA.

This Paper!

Jian Yang, D. Zhang, A. F. Frangi, and Jing-yu Yang. Two-dimensional pca: a new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):131–137, January 2004.

I based my presentation on this paper.

Classification Method

Once an image has been decomposed into its d principal component vectors, we can combine these into an $m \times d$ matrix $\boldsymbol{B}_i = [\boldsymbol{Y}_{i1}, \dots, \boldsymbol{Y}_{id}]$. This is called the feature matrix of image $\boldsymbol{A}_i$.

The distance between two feature matrices $\boldsymbol{B}_i$ and $\boldsymbol{B}_j$ is then

$$d(\boldsymbol{B}_i, \boldsymbol{B}_j) = \sum_{k=1}^{d} \|\boldsymbol{Y}_{ik} - \boldsymbol{Y}_{jk}\|_2. \tag{8}$$

This can be used in image recognition. If two images are close, they could be the same person. Other distance metrics can be used.

Distance Matrix

It is possible to compute distances among all individuals, giving us a very large distance matrix. Can we learn anything from this distance matrix? Are people of a certain gender or race grouped together? Applying k-means clustering or plotting a dendrogram would be an interesting project.

Note: the distance matrix can get large very quickly, it's size is $O(M^2)$.

Open Questions

Accuracy vs Computation Time

One of the main reasons one would choose 2D-PCA over PCA or some other classifier is for decreased computation time. A study into the accuracy vs computation time tradeoff would be insightful.

Optimal Parameter Selection

What value of d is low enough to save computational resources but high enough to accurately classify an image? Does this choice of d vary with number of images in a dataset or with image dimensions?

Robustness

2D-PCA has been used for facial recognition with variations in time (two weeks), facial expression, and lighting condition. Would it still perform well with people of different age?

Conclusion

Conclusion

We've heard many times that there is no single best model for every situation. 2D-PCA was developed specifically for the purpose of face recognition. The researchers understood their data and their objective to the point that they were able to generalize a well-known method (PCA) to a new one that fit their needs.

Can we do the same?