Introduction to Data Mining
Homework 1

151250189, 翟道京, zhaidj@smail.nju.edu.cn

2018 年 4 月 20 日

1 LDA, NCA and PCA

Linear Discriminant Analysis(LDA) and Neighborhood component Analysis (NCA) are two widely used methods for dimensionality reduction. Please compare them with PCA and answer what are their rationales for data reduction.

Solution.

(a) Linear Discriminant Analysis (LDA)[1]

Linear Discriminant Analysis (LDA) is most commonly used as dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications. The goal is to project a dataset onto a lower-dimensional space with good class-separability in order avoid overfitting ("curse of dimensionality") and also reduce computational costs.

In a nutshell, often the goal of an LDA is to project a feature space (a dataset n-dimensional samples) onto a smaller subspace k (where $k \leq n-1$) while maintaining the class-discriminatory information. And the LDA approach can be summarized in 5 steps:[2]

1 *Compute the d-dimensional mean vectors for the different classes from the dataset.*

2 *Compute the scatter matrices (in-between-class and within-class scatter matrix).*

The within-class scatter matrix $S_{\mathbf{W}}$ is computed by the following equation:

$$S_{\mathbf{W}} = \sum_{i=1}^{c} S_i$$

where S_i is scatter matrix for i_{th} class,

$$S_i = \sum_{\mathbf{x} \in D_i}^{n} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T$$

[1] Wikipedia: Linear discriminant analysis, `https://en.wikipedia.org/wiki/Linear_discriminant_analysis`

[2] Blog: Linear Discriminant Analysis by Sebastian Raschka: `http://sebastianraschka.com/Articles/2014_python_lda.html`

and $\mathbf{m}_i$ *is the mean vector for the* i_{th} *class, say*

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in D_i}^{n} \mathbf{x}.$$

The between-class scatter matrix $S_\mathbf{B}$ *is computed by the following equation:*

$$S_\mathbf{B} = \sum_{i=1}^{c} N_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T$$

where $\mathbf{m}$ *is the overall mean vector, and* N_i *is the sizes of the respective classes.*

3 *Compute the eigenvectors* $(\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_d)$ *and corresponding eigenvalues* $(\lambda_1, \lambda_2, ..., \lambda_d)$ *for the scatter matrices.*

4 *Sort the eigenvectors by decreasing eigenvalues and choose* k *eigenvectors with the largest eigenvalues to form a* $d \times k$ *dimensional matrix* $\mathbf{W}$ *(where every column represents an eigenvector).*

5 *Use this* $d \times k$ *eigenvector matrix to transform the samples onto the new subspace. This can be summarized by the matrix multiplication:* $\mathbf{Y} = \mathbf{X} \times \mathbf{X}$ *(where* $\mathbf{X}$ *is a* $n \times d$*-dimensional matrix representing the* n *samples, and* $\mathbf{y}$ *are the transformed* $n \times k$*-dimensional samples in the new subspace).*

(b) LDA vs. PCA

Both Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) are linear transformation techniques that are commonly used for dimensionality reduction. PCA can be described as an "unsupervised" algorithm, since it "ignores" class labels and its goal is to find the directions (the so-called principal components) that maximize the variance in a dataset. In contrast to PCA, LDA is "supervised" and computes the directions ("linear discriminants") that will represent the axes that that maximize the separation between multiple classes.

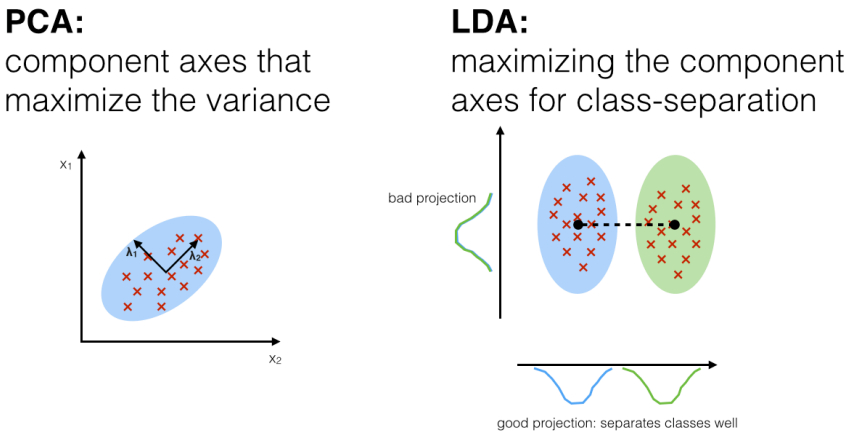

图 1: LDA vs. PCA

Although it might sound intuitive that LDA is superior to PCA for a multi-class classification task where the class labels are known, this might not always the case. For example, comparisons between classification accuracies for image recognition after using PCA or LDA show that PCA tends to outperform LDA if the number of samples per class is relatively small (PCA vs. LDA, A.M. Martinez et al., 2001 [3]*). In practice, it is also not uncommon to use both LDA and PCA in combination: E.g., PCA for dimensionality reduction followed by an LDA.* [4]

(c) Neighborhood component Analysis (NCA)

Proposed by Goldberger et al[5]*, neighborhood components analysis aims at "learning" a distance metric by finding a linear transformation of input data such that the average leave-one-out (LOO) classification performance is maximized in the transformed space. The key insight to the algorithm is that a matrix* $\mathbf{A}$ *corresponding to the transformation can be found by defining a differentiable objective function for* $\mathbf{A}$*, followed by use of an iterative solver such as conjugate gradient descent. One of the benefits of this algorithm is that the number of classes* k *can be determined as a function of* $\mathbf{A}$*, up to a scalar constant. This use of the algorithm therefore addresses the issue of model selection.*

In order to define $\mathbf{A}$*, we define an objective function describing classification accuracy in the transformed space and try to determine* $\mathbf{A}^*$ *such that this objective function is maximized.*

$$\mathbf{A}^* = argmax_{\mathbf{A}} f(\mathbf{A})$$

1 Leave-one-out (LOO) classification

Consider predicting the class label of a single data point by consensus of its k-nearest neighbours with a given distance metric. This is known as leave-one-out classification. However, the set of nearest-neighbours C_i *can be quite different after passing all the points through a linear transformation. Specifically, the set of neighbours for a point can undergo discrete changes in response to smooth changes in the elements of* $\mathbf{A}$*, implying that any objective function* $f(\cdot)$ *based on the neighbours of a point will be piecewise-constant, and hence not differentiable.*

2 Solution

We can resolve this difficulty by using an approach inspired by stochastic gradient descent. Rather than considering the k-nearest neighbours at each transformed point in LOO-classification, we'll consider the entire transformed data set as stochastic nearest neighbours. We define these using a softmax function of the squared Euclidean distance between a given LOO-classification point and each other point in the transformed space:

[3] `https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=908974`

[4] Blog: Linear Discriminant Analysis by Sebastian Raschka: `http://sebastianraschka.com/Articles/2014_python_lda.html`

[5] J. Goldberger, G. Hinton, S. Roweis, R. Salakhutdinov. (2005) Neighbourhood Components Analysis. Advances in Neural Information Processing Systems. 17, 513-520, 2005.

$$p_{ij} = \begin{cases} \frac{e^{-||Ax_i - Ax_j||^2}}{\sum_k e^{-||Ax_i - Ax_k||^2}}, & \text{if } j \neq i \\ 0, & \text{if } j = i \end{cases}$$

The probability of correctly classifying data point i is the probability of classifying the points of each of its neighbours C_i:

$$p_i = \sum_{j \in C_i} p_{ij}$$

where p_{ij} is the probability of classifying neighbour j of point i.

Define the objective function using LOO classification, this time using the entire data set as stochastic nearest neighbours:

$$f(\mathbf{A}) = \sum_i \sum_{j \in C_i} p_{ij} = \sum_i p_i$$

Note that under stochastic nearest neighbours, the consensus class for a single point i is the expected value of a point's class in the limit of an infinite number of samples drawn from the distribution over its neighbours $j \in C_i$ i.e.: $P(Class(X_i) = Class(X_j)) = p_{ij}$. Thus the predicted class is an affine combination of the classes of every other point, weighted by the softmax function for each $j \in C_j$ where C_j is now the entire transformed data set.

This choice of objective function is preferable as it is differentiable with respect to $\mathbf{A}$ (denote $x_{ij} = x_i - x_j$):

$$\begin{aligned} \frac{\partial f}{\partial A} &= -2A \sum_i \sum_{j \in C_i} p_{ij} \left(x_{ij} x_{ij}^T - \sum_k p_{ik} x_{ik} x_{ik}^T \right) \\ &= 2A \sum_i \left(p_i \sum_k p_{ik} x_{ik} x_{ik}^T - \sum_{j \in C_i} p_{ij} x_{ij} x_{ij}^T \right) \end{aligned}$$

Obtaining a gradient for $\mathbf{A}$ means that it can be found with an iterative solver such as conjugate gradient descent. Note that in practice, most of the innermost terms of the gradient evaluate to insignificant contributions due to the rapidly diminishing contribution of distant points from the point of interest. This means that the inner sum of the gradient can be truncated, resulting in reasonable computation times even for large data sets.

3 Alternative formulation

"Maximizing $f(\cdot)$ is equivalent to minimizing the L_1-distance between the predicted class distribution and the true class distribution (ie: where the p_i induced by $\mathbf{A}$ are all equal to 1). A natural alternative is the KL-divergence, which induces the following objective function and gradient:" (Goldberger 2005)

$$g(A) = \sum_i \log\left(\sum_{j \in C_i} p_{ij}\right) = \sum_i \log(p_i)$$

$$\frac{\partial g}{\partial A} = 2A \sum_i \left(\sum_k p_{ik} x_{ik} x_{ik}^T - \frac{\sum_{j \in C_i} p_{ij} x_{ij} x_{ij}^T}{\sum_{j \in C_i} p_{ij}}\right)$$

In practice, optimization of $\mathbf{A}$ *using this function tends to give similar performance results as with the original.*

(d) Experiments in Dimensionality Reduction using NCA

Goldberger. et al[6] *investigated the use of linear dimensionality reduction using NCA (with nonsquare* $\mathbf{A}$*) for visualization as well as reduced-complexity classification on several datasets.*

In figure 2 it shows their examples of 2-D visualization. First, they generated a synthetic three dimensional dataset (shown in top row of figure 2) which consists of 5 classes (shown by different colors). In two dimensions, the classes are distributed in concentric circles, while the third dimension is just Gaussian noise, uncorrelated with the other dimensions or the class label. If the noise variance is large enough, the projection found by PCA is forced to include the noise (as shown on the top left of figure 2). (A full rank Euclidean metric would also be misled by this dimension.) The classes are not convex and cannot be linearly separated, hence the results obtained from LDA will be inappropriate (as shown in figure 2). In contrast, NCA adaptively finds the best projection without assuming any parametric structure in the low dimensional representation. They have also applied NCA to the UCI "wine" dataset, which consists of 178 points labeled into 3 classes and to a database of gray-scale images of faces consisting of 18 classes (each a separate individual) and 560 dimensions (image size is 20×28). The face dataset consists of 1800 images (100 for each person). Finally, they applied our algorithm to a subset of the USPS dataset of handwritten digit images, consisting of the first five digit classes ("one" through "five"). The grayscale images were downsampled to 8 × 8 pixel resolution resulting in 64 dimensions.

As can be seen in figure 2 when a two-dimensional projection is used, the classes are consistently much better separated by the NCA transformation than by either PCA (which is unsupervised) or LDA (which has access to the class labels). Of course, the NCA transformation is still only a linear projection, just optimized with a cost function which explicitly encourages local separation.

[6] J. Goldberger, G. Hinton, S. Roweis, R. Salakhutdinov. (2005) Neighbourhood Components Analysis. Advances in Neural Information Processing Systems. 17, 513-520, 2005.

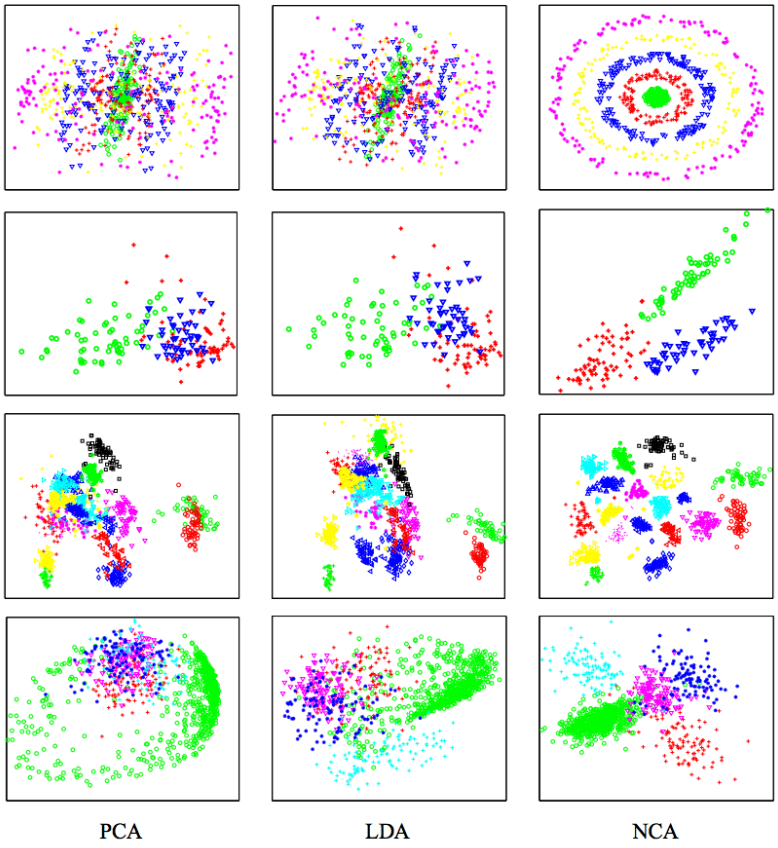

图 2: Dataset visualization results of PCA, LDA and NCA applied to (from top) the "concentric rings", "wine", "faces"and "digits"datasets.The data are reduced from their original dimensionalities (D=3,D=13,D=560,D=256 respectively) to the d=2 dimensions show.

(e) NCA vs. other Methods

Here are other Mahalanobis distance metric learning methods for labelled data. Relevant component analysis (RCA) finds a distance metric, but assumes the classes have Gaussian distributions whereas NCA makes no assumption about class distribution. Xing et al's method finds a transformation that minimises the pairwise distances between points of the same class. This assumes that all points of the one class form a single, compact and connected set. Lowe's method is very similar to NCA, but further constrains the distance metric to be diagonal.

For low rank, linear transformation of data, LDA will be optimal if the classes are Gaussian with common covariance. This is not going to be true in general. LDA also suffers from a small sample size problem when dealing with high-dimensional data when the within-class scatter matrix is nearly singular. There are newer variants of LDA that improve this instability, and also improve robustness to outliers.

In general, there are two classes of regularization assumption that are common in linear methods for classification. The first is a strong parametric assumption about the structure of the class distributions (typically enforcing connected or even convex structure); the second is

an assumption about the decision boundary (typically enforcing a hyperplane). NCA method makes neither of these assumptions, relying instead on the strong regularization imposed by restricting ourselves to a linear transformation of the original inputs.