e-PGPathshala

Subject : Computer Science

Paper: Machine Learning

Module: Dimensionality Reduction - I

Module No: CS/ML/27

Quadrant I – e-text

Welcome to the e-PG Pathshala Lecture Series on Machine Learning.Machine learning generally deals with a large amount of high dimension data.In this module we will discuss an important aspect of machine learning – the appropriate reduction of this dimensionality.

Learning Objectives:

The learning objectives of this module are as follows:

- To explain the Curse of DimensionalityTo discuss the concept of Dimensionality Reduction
- To outline the concepts of feature selection and feature reduction
- To explain in detail the Principal Component Analysis method of Dimensionality Reduction

27.1 Curse of Dimensionality

In any machine learning problem, if the number of observables or features is increased then it takes more time to compute, more memory to store inputs and intermediate results and more importantly much more data samples are needed for the learning. From a theoretical point of view, increasing the number of features should lead to better performance. However in practice, the inclusion of more features leads to a decrease in performance. This aspect is called the curse of dimensionality and is basically because the number of training examples required increases exponentially as dimensionality increases.A lot of machine learning methods have at least O(nd^2) complexity where n is the number of samples and d is the dimensioanlity. A typical example is the need to estimate covariance matrix for which as d becomes large the number of samples is of the O(nd^2) which involves huge computations. In other words if the number of features that is dimension d is large, the number of samples n, may be too small for accurate parameter estimation.

For example, the covariance matrix has d^2parameters (Figure 27.1).

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \cdots & \sigma_{1d} \\ \vdots & \ddots & \vdots \\ \sigma_{d1} & \cdots & \sigma_d^2 \end{bmatrix}$$

Figure 27.1 Covariance Matrix

If themodel parameters are highthen there will be*overfitting.*An interesting paradox is thatif n <d^2we are better off assuming that features are uncorrelated, even if we know this assumption is wrong.We are likely to avoid overfitting because we fit a model with less parameters (Figure 27.2)

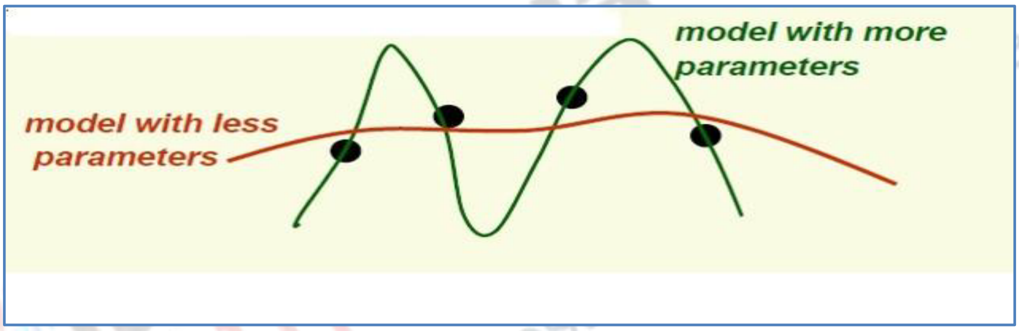

Figure 27.2 Model with More and Less Number of Parameters

Suppose we want to use the nearest neighbour approach with k = 1 (*1NN*). Suppose we start with only one feature, this feature is not discriminative, i.e. it does not separate the classes well (Figure 27.3).

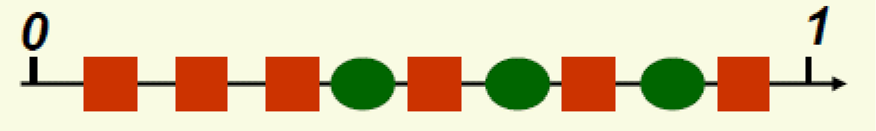

Figure 27.3 1NN with only One Feature

Now we decide to use 2 features. For the 1NN method to work well, we need a lot of samples, i.e. samples have to be dense. To maintain the same density as in 1D (9 samples per unit length), how many samples do we need? As we discussed we need 9^2samples to maintain the same density as in *1D.*Of course, when we go from 1 feature to 2, no one gives us more samples, we still have 9. This is way too sparse for *1NN* to work well (Figure 27.4).

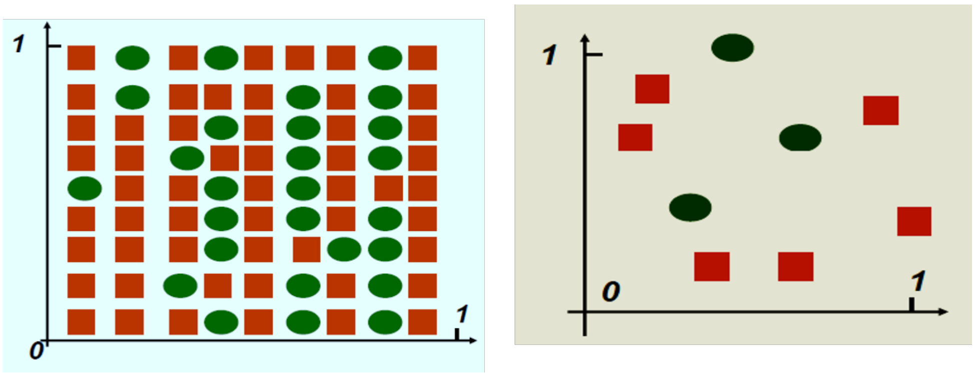

Figure 27.4 1NN with only Two Features

Things are even more of a problem when we decide to use 3 features (Figure 27.5). If *9* was dense enough in 1D, in 3D we need $9^3=729$ samples. In general, if *n* samples is dense enough in *1D* - Then in *d* dimensions we need n^d samples.

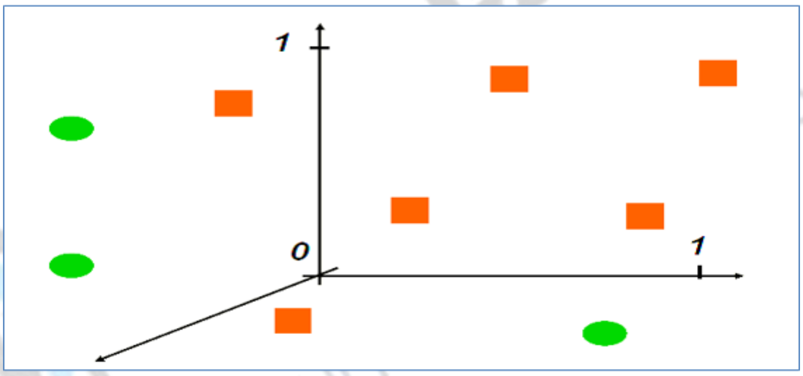

Figure 27.5 1NN with Three Dimensions

27.2 Dimensionality Reduction

Some features (dimensions) bear little useful information, which essentially means that we can drop some features. In dimensionality reduction high-dimensional points are projectedto a low dimensional space while preserving the “essence” of the data. In projecting from a higher dimensional space to a lower dimensional space the distances are preserved as well as possible. After this projection the learning problems are solved in low dimensions. An illustration is shown in Figure 27.6.

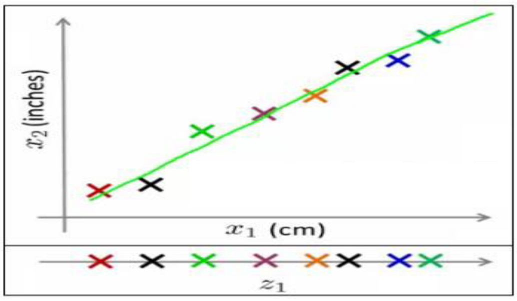

Figure 27.6 From 2D to 1D

Let us assume that we have data with dimension d. Let us reduce the dimensionality to k<d by discarding unimportant features or combining several features in one and then use the resulting k-dimensional data set for learning for classification problem (e.g. parameters of probabilities P(x|C) or learning for regression problem (e.g. parameters for model y=g(x|Thetha)).

For a fixed number of samples, as we add features, the graph of classification error is shown in Figure 27.7. We see that there exists an optimal number of features which results in minimum classification error.

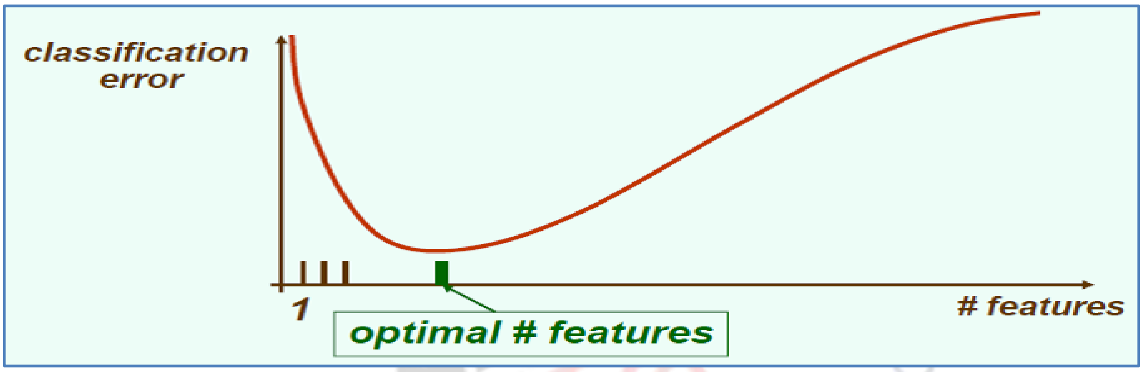

Figure 27.7 Optimal Number of Features for Classification

Thus for each fixed sample size ***n***, there is an optimal number of features that we should use. In dimensionality reduction we strive to find that set of features that are effective.

27.3 Why Dimensionality Reduction?

Why do we need to carry out dimensionality reduction?. Essentially it reduces time complexity since there will be less computation and therefore more efficient learning. It also reduces space complexity since there are less number of parametersresulting in more efficient storage. Moreover simpler models are more robust on small datasets since complex models may result in overfitting especially in the case of smaller datasets. The dimensionality reduction helps in data visualization (structure, groups, outliers, etc) specifically when reduction is to 2 or 3 dimensions. Most machine learning and data mining techniques may not be effective for high dimensional data since the intrinsic dimension (that is the actual features that decide the classification) may be small,for example the number of genes actually responsible for a disease may be small but the dataset may contain a large number of other genes as features. The dimension-reduced data can be used for visualizing, exploring and understanding the data. In addition cleaning the data will allow simpler models to be built later.

27.4 The Process of Dimensionality Reduction

The process of reducing the number of random variables (features) used to represent the samples under consideration can be carried out by combining, transforming or selecting features. We have to estimate which features can be removed from the data. Several features can be combined together without loss or even with gain of information (e.g. income of all family members for loan application), however we need to estimate which features to combine from the data.

The simplest way to carry out dimensionality reduction is to keep just one variable and discard all others. However this is too simplistic and not reasonable. Another simple way is to weigh all variables equally but again this is not reasonable unless all variables have the same variance. Another method is to weigh the features based on some criteria and find the weighted average. However the issue is the choice of the criterion. The basic issues for dimensionality reduction are two fold namely how do we represent the data, whether we use vector space and what is the criteria to be used in carrying out the reduction process.Dimensionality can be reduced basically using two methods: feature reduction and feature selection. We will discuss these methods in section 27.6.

27.5 Criterion for Reduction

There are many criteria that can be used for dimensionality reduction. These include criteria that are mainly geometric based and information theory based. These criteria need to capture the variation in data since these variations are “signals” or information contained in the data. We need to normalize each variable first and then discover variables or dimensions that are highly correlated or dependent (Figure 27.8). When variables are highly related they can be combined to form a simpler representation.

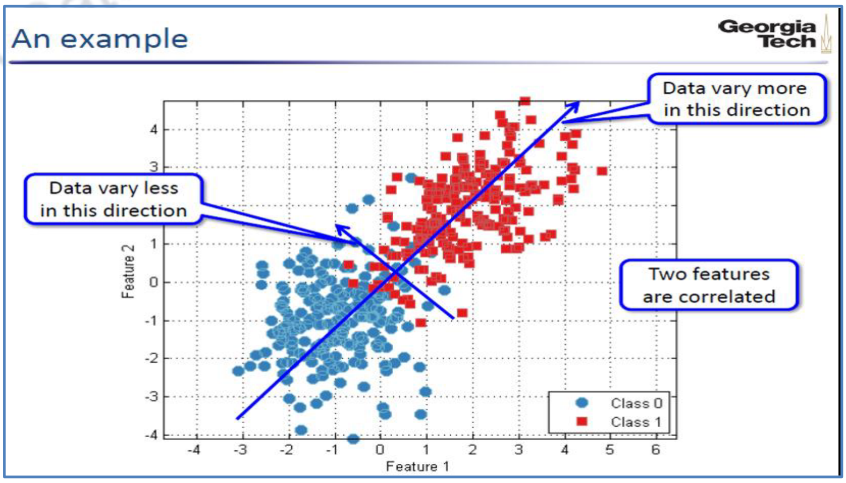

Figure 27.8 Dependency of Features

Courtesy: http://www.cc.gatech.edu/~lsong/teaching/CSE6740fall13/lecture1.pdf

27.6 Feature Reduction and Feature Selection

As already discussed there are two approaches to dimensionality reduction namely feature reduction and feature selection. The concept underlying the two methods is shown in Figure 27.9.

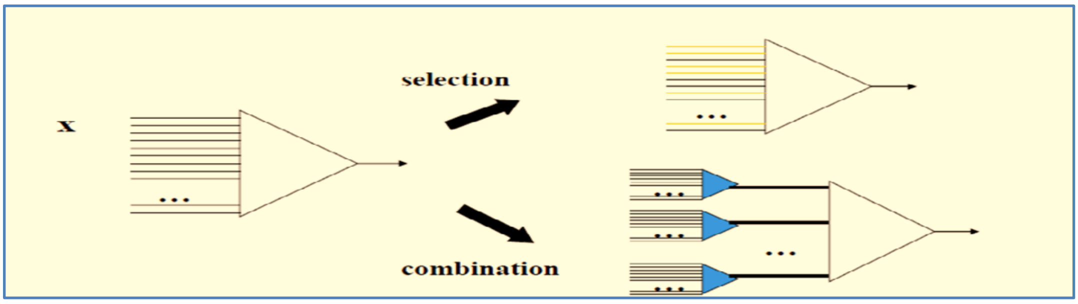

Figure 27.9 Illustration of Feature Reduction and Feature Selection

27.6.1 Feature Reduction

In feature reduction all the original features are used however the features are linearly or non-linearly transformed.Here we combine the high dimensional inputs to a smaller set of features; and then use this transformed data for learning. A simple example is shown in Figure 27.10 where a vector x is transformed to a vector y. Ideally, the new vector *y* should retain all information from x that is important for learning.

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 + x_2 \\ x_3 + x_4 \end{bmatrix} = \boldsymbol{y}$$

Figure 27.10 Feature Reduction

27.6.2 Feature Selection

In feature selection only a smaller subset of the original features are selected from a large set of inputs and this new data is used for learning. Generally the features are ranked on a defined “interestingness” measure. Then we can select and use x% of features with top ranking based on feature values.

27.7 Dimensionality Reduction with Feature Reduction

Feature reduction refers to the mapping of the original high-dimensional data onto a lower-dimensional space.Criterion for feature reduction can be different based on different problem settings. In an unsupervised setting the criteria could try to minimize the information loss, in supervised setting it could try to maximize the class discrimination.

Given a set of data points of p variables we compute the linear transformation (projection) (Figure 27.11).

$$G \in \Re^{p \times d} : x \in \Re^{p} \to y = G^{T} x \in \Re^{d} \quad (d << p)$$

27.7.1Applications of Feature Reduction:

Applications of feature reduction include the following where the high dimensional input data is transformed into a lower dimensional one before carrying out machine learning:

- Face recognition
- Handwritten digit recognition
- Text mining
- Image retrieval
- Microarray data analysis
- Protein classification

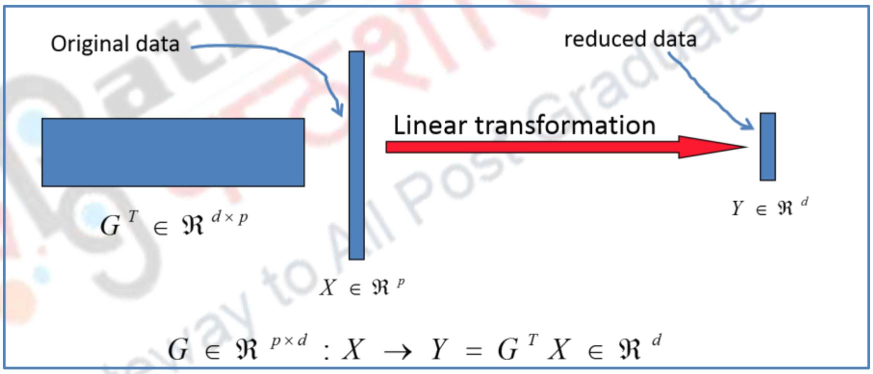
Original data
reduced data
Linear transformation
$G^{T} \in \Re^{d \times p}$
$X \in \Re^{p}$
$Y \in \Re^{d}$
$G \in \Re^{p \times d} : X \to Y = G^{T} X \in \Re^{d}$

Figure 27.11 Feature Reduction – Linear Transformation

Here we combine the old features ***x*** to create a new set of features y based on the some function of x as given below:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \to f\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}\right) = \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} = y \quad \text{with } k < d$$

The best ***f***(***x***) is most likely to be a non-linear function. Assuming, ***f***(***x***) is a linear mapping, it can be represented by a matrix ***W***as given below:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \Rightarrow W \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} w_{11} & \cdots & w_{1d} \\ \vdots & & \vdots \\ w_{k1} & \cdots & w_{kd} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} \quad \text{with } k < d$$

27.7.2 Approaches to Finding Optimal Transformation

There are basically three approaches for finding optimal transformations which are listed below:

- **Principal Components Analysis (PCA):** Seeks a projection that preserves as much information in the data as possible (in a least-squares sense).
- **Fisher Linear Discriminant:** Find projection to a line such that samples from different classes are well separated
- **Singular value decomposition (SVD):** Transforming correlated variables into a set of uncorrelated ones

In this module we will be discussing the first approach namely PCA.

27.8 Principle Component Analysis (PCA)

Let us first discuss the basis of PCA. Main idea: seek most accurate data representation in a lower dimensional space.PCA was introduced by Pearson (1901) and Hotelling (1933) to describe the variation in a set of multivariate data in terms of a set of uncorrelated variables.We typically have a data matrix of *n* observations on *p* correlated variables $x_1, x_2, \ldots x_{p.}$ PCA looks for a transformation of the $x_i$into *p* new variables y_i that are uncorrelated.

27.8.1 Example in 2-D

Let us project the 2-D data to 1-D subspace (a line) in such a way that the projection error is minimum. Figure 27.12 shows the cases of two such projections where the right hand side projection shows small projection error and therefore is a good line to project to. We need to note that the good line to use for projection lies in the direction of largest variance. After the data is projected on the best line, need to transform the coordinate system to get 1D representation for vector *y*(Figure 27.13).Note that new data ***y*** has the same variance as old data ***x*** in the direction of the green line

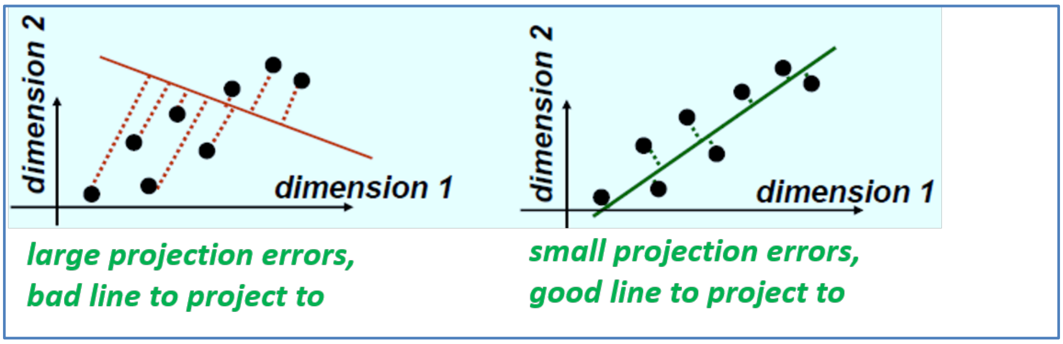

Figure 27.12 2-D to 1-D Projection

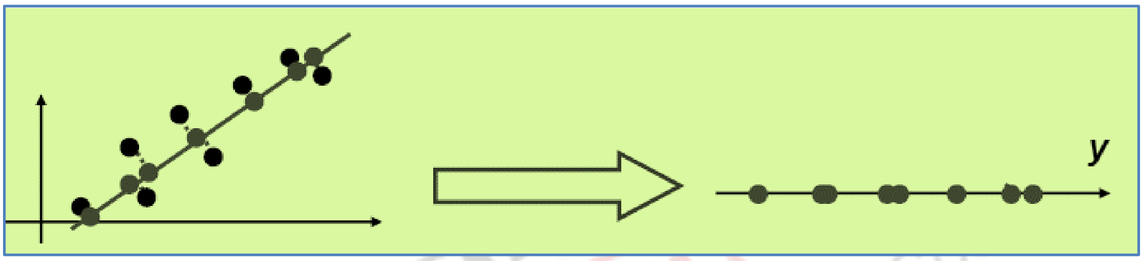

Figure 27.13 1-D Projection of the Data

27.8.2 PCA Methodology

PCA projects the data along the directions where the data varies the most. These directions are determined by the **eigenvectors of the covariance matrix** corresponding to the largest eigenvalues.The magnitude of the eigenvalues corresponds to the variance of the data along the eigenvector directions.Let us assume that d observables are linear combination of k<d vectors. We would like to work with this basis as it has lesser dimension and have all(almost) required information. Using this bais we expect data is uncorrelated or otherwise we could have reduced it further. We choose the projection that shows the large variance or otherwise the featues bear no information (

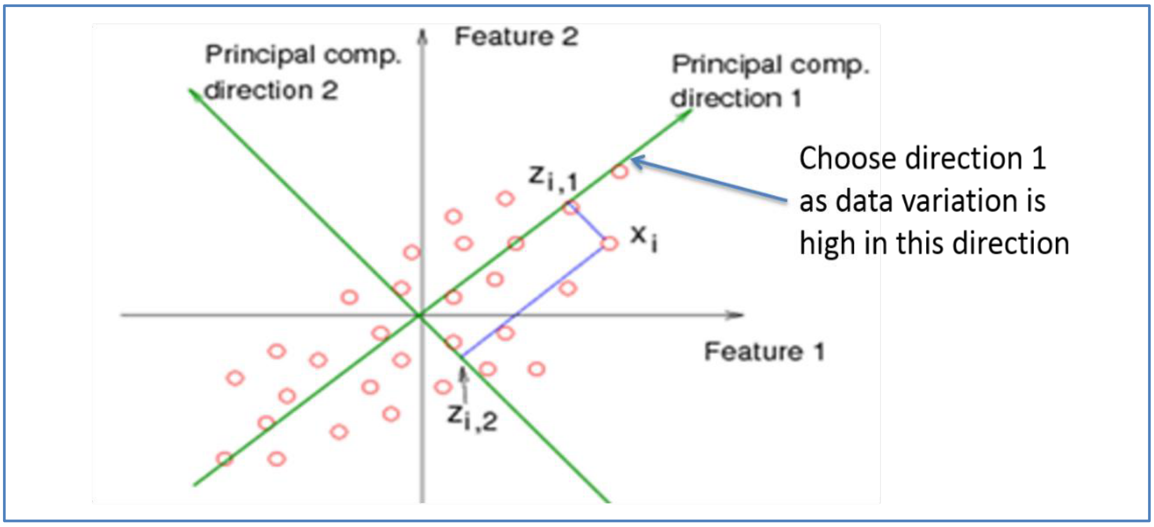

Figure 27.14 Direction of Principal Components

We choose directions such that a total variance of data will be maximum Figure 27.14), that is we choose a projection that maximizestotal variance. We choose directions that are orthogonal and try to minimize correlation.When we consider a **d-dimensional feature** space, we need to choose k<d orthogonal directions which maximize total variance. We first calculate d by d symmetric **covariance matrix** estimated from samples. Then we select k largest eigenvalue of the covariance matrix and the associated k eigenvector.

27.9 Steps of PCA

Let D be the data set**D={*x*$_1$,*x*$_2$**,…,***x*$_n$**} where each ***x*$_i$** is a ***d***-dimensional vector. We need to reduce the dimension d to ***k.***

1. First we must find the sample mean of all the ***x*$_i$**

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

2. Then we subtract the sample mean from each ***x*$_i$** of the data

$$z_i = x_i - \hat{\mu}$$

3. Now we compute the scatter matrix

$$S = \sum_{i=1}^{n} z_i z_i^t$$

4. Next we compute eigenvectors $\boldsymbol{e_1},\boldsymbol{e_2},\ldots,\boldsymbol{e_k}$ corresponding to the ***k*** largest eigenvalues of the scatter matrix ***S***

5. Let $\boldsymbol{e_1},\boldsymbol{e_2},\ldots,\boldsymbol{e_k}$be the columns of matrix

$$\boldsymbol{E} = \left[\mathrm{e}_1 \cdots \mathrm{e}_k\right]$$

6. The desired ***y*** which is the closest approximation to ***x*** is $\boldsymbol{Y} = \boldsymbol{E}^t\boldsymbol{z}$

The larger the eigenvalue of ***S***, the larger is the variance in the direction of corresponding eigenvector (Figure 27.15).

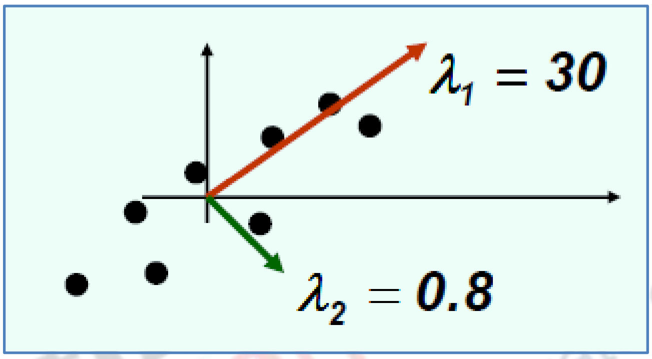

Figure 27.15 Variance and Eigen Vector

This result is exactly what we wanted. We wanted to project ***x*** into subspace of dimension ***k*** which has the largest variance. This is very intuitive that is we restrict the attention to directions where the scatter is the greatest. Thus PCA can be thought of as finding a new orthogonal basis by rotating the old axis until the directions of maximum variance are found (Figure 27.16).

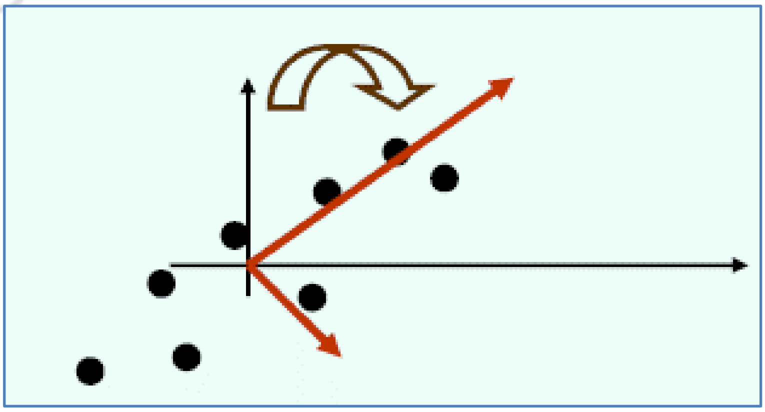

Figure 27.16 Rotation to find Direction of Maximum Variance

27.10 Practical use of PCA

- PCA is useful for **pre-processing features** before the actual classification is carried out.
- PCA does not take different classes into account it only considers the properties of different features.
- If two features x_i and x_j are redundant, then one eigen value in A is very small and **one dimension** can be dropped
- We do not need to choose between two correlating features. It is better to do the linear transform and then drop the least significant dimension. In this way both of the correlating features are utilized.
- This reduction in dimensionality enables the identification of the strongest patterns in the data
- PCA allows the capture of most of the variability of the data with a small fraction of the total set of dimensions
- It eliminates much of the noise in the data making it beneficial for data mining and other data analysis algorithms

27.11 Drawback of PCA

PCA was designed for accurate data representation and not for data classification. The primary job of PCA is to preserve as much variance in data as possible.Therefore only if the direction of maximum variance is important for classification, it will work. However sometimes, the directions of maximum variance may not be useful for the classification.

Summary

- Explained the Curse of Dimensionality
- Discussed the concept of Dimensionality Reduction
- Outlined the concepts of feature selection and feature reduction
- Explained in detail the Principal Component Analysis method of Dimensionality Reduction