PCA and KPCA

16BCE1259
Shushil Kumar Ravishankar

What is PCA?

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components.

In PCA, we are interested to find the directions (components) that maximize the variance in our dataset and we project the entire set of data (without class labels) onto the subspace with maximum variance.

General approach:

1. Take the whole dataset consisting of d-dimensional samples ignoring the class labels
2. Compute the d-dimensional mean vector (i.e., the means for every dimension of the whole dataset)
3. Compute the scatter matrix (alternatively, the covariance matrix) of the whole data set
4. Compute eigenvectors (e_1,e_2,...,e_d) and corresponding eigenvalues (λ_1,λ_2,...,λ_d)
5. Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a d×k dimensional matrix W (where every column represents an eigenvector)
6. Use this d×k eigenvector matrix to transform the samples onto the new subspace. This can be summarized by the mathematical equation: $y=W^T\times X$ (where X is a d×1-dimensional vector representing one sample, and y is the transformed k×1-dimensional sample in the new subspace.)

What is KPCA?

The “classic” PCA approach described above is a linear projection technique that works well if the data is linearly separable. However, in the case of linearly inseparable data, a nonlinear technique is required if the task is to reduce the dimensionality of a dataset.

The basic idea to deal with linearly inseparable data is to project it onto a higher dimensional space where it becomes linearly separable. Let us call this nonlinear mapping function ϕ so that the mapping of a sample **x** can be written as $\mathbf{x} \rightarrow \phi(\mathbf{x})$, which is called “kernel function.”

Now, the term “kernel” describes a function that calculates the dot product of the images of the samples **x** under ϕ.

$$\kappa(x_i, x_j)= \phi(x_i).\phi(x_j)^T$$

Advantages of Dimensionality Reduction:

1. It reduces the time and storage space required.
2. Removal of multi-collinearity improves the performance of the machine learning model.
3. It becomes easier to visualize the data when reduced to very low dimensions such as 2D or 3D.

Observations on Big Mart Sales dataset:

1. PCA

After using PCA to transform points to a n-dimension vector space, its new accuracy score is found by using a MLP Classifier to classify the dataset.

It was found that the accuracy is highest for n=8,11.

As the original vector space already contains 11 dimensions, we will only consider n=8 for dimensional reduction.

Therefore, after undergoing a dimensionality reduction from a 11-dimension vector space to an 8-dimension vector space, the accuracy of the MLP Classifier is 91.79%

This means that the dataset is more separable when transformed to an 8-dimension vector space than any other vector space.

2. KPCA

After using PCA to transform points to a n-dimension vector space using different kernels, its new accuracy score is found by using a MLP Classifier to classify the dataset.

Highest accuracy is given by cosine kernel with reduction to 8 dimensions (43.76%).

The accuracy score of other kernels are:

- **RBF:** 39.85% for 6 components
- **Poly:** 39.81% for 1 component

When the dataset is used without applying kpca with a MLP Classifier, the accuracy is 28.33%.

This implies that the dataset is transformed to a more easily separable data points on applying the PCA with appropriate kernel function.