**Principal component analysis** (**PCA**) is a **statistical** procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). **KNN**has been used in **statistical** estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique

|  |  |  |
| --- | --- | --- |
| **K Nearest Neighbors - Classification** |  |  |
| K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970’s as a non-parametric technique. |  |  |
|  |  |  |
| **Algorithm** |  |  |
| A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor. |  |  |
|  |  |  |
| http://www.saedsayad.com/images/KNN_similarity.png |  |  |
|  |  |  |
| It should also be noted that all three distance measures are only valid for continuous variables. In the instance of categorical variables the Hamming distance must be used. It also brings up the issue of standardization of the numerical variables between 0 and 1 when there is a mixture of numerical and categorical variables in the dataset. |  |  |
| http://www.saedsayad.com/images/KNN_hamming.png |  |  |
| Choosing the optimal value for K is best done by first inspecting the data. In general, a large K value is more precise as it reduces the overall noise but there is no guarantee. Cross-validation is another way to retrospectively determine a good K value by using an independent dataset to validate the K value. Historically, the optimal K for most datasets has been between 3-10. That produces much better results than 1NN. |  |  |
|  |  |  |
| *Example*: |  |  |
| Consider the following data concerning credit default. Age and Loan are two numerical variables (predictors) and Default is the target. |  |  |
| http://www.saedsayad.com/images/KNN_example_1.png |  |  |
| We can now use the training set to classify an unknown case (Age=48 and Loan=$142,000) using Euclidean distance. If K=1 then the nearest neighbor is the last case in the training set with Default=Y. |  |  |
|  |  |  |
| D = Sqrt[(48-33)^2 + (142000-150000)^2] = 8000.01  >> Default=Y |  |  |
|  |  |  |
| http://www.saedsayad.com/images/KNN_example_2.png |  |  |
| With K=3, there are two Default=Y and one Default=N out of three closest neighbors. The prediction for the unknown case is again Default=Y. |  |  |
|  |  |  |
| **Standardized Distance** |  |  |
| One major drawback in calculating distance measures directly from the training set is in the case where variables have different measurement scales or there is a mixture of numerical and categorical variables. For example, if one variable is based on annual income in dollars, and the other is based on age in years then income will have a much higher influence on the distance calculated. One solution is to standardize the training set as shown below. |  |  |
|  |  |  |
| http://www.saedsayad.com/images/KNN_example_3.png |  |  |
| Using the standardized distance on the same training set, the unknown case returned a different neighbor which is not a good sign of robustness. |  |  |