Principal Component Analysis

Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process.

So to sum up, the idea of PCA is simple — reduce the number of variables of a data set, while preserving as much information as possible.

**STEP 1: STANDARDIZATION**

The aim of this step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis.

More specifically, the reason why it is critical to perform standardization prior to PCA, is that the latter is quite sensitive regarding the variances of the initial variables. That is, if there are large differences between the ranges of initial variables, those variables with larger ranges will dominate over those with small ranges (For example, a variable that ranges between 0 and 100 will dominate over a variable that ranges between 0 and 1), which will lead to biased results. So, transforming the data to comparable scales can prevent this problem.

Mathematically, this can be done by subtracting the mean and dividing by the standard deviation for each value of each variable.

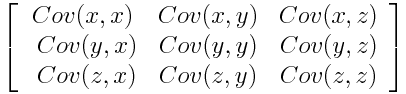
Principal Component Analysis Standardization

Once the standardization is done, all the variables will be transformed to the same scale.

### STEP 2: COVARIANCE MATRIX COMPUTATION

The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them. Because sometimes, variables are highly correlated in such a way that they contain redundant information. So, in order to identify these correlations, we compute the covariance matrix.

The covariance matrix is a p × psymmetric matrix (where p is the number of dimensions) that has as entries the covariances associated with all possible pairs of the initial variables. For example, for a 3-dimensional data set with 3 variables x, y, and z, the covariance matrix is a 3×3 matrix of this from:

Covariance Matrix for 3-Dimensional Data

Since the covariance of a variable with itself is its variance (Cov(a,a)=Var(a)), in the main diagonal (Top left to bottom right) we actually have the variances of each initial variable. And since the covariance is commutative (Cov(a,b)=Cov(b,a)), the entries of the covariance matrix are symmetric with respect to the main diagonal, which means that the upper and the lower triangular portions are equal.

**What do the covariances that we have as entries of the matrix tell us about the correlations between the variables?**

It’s actually the sign of the covariance that matters:

* if positive then: the two variables increase or decrease together (correlated)
* if negative then: One increases when the other decreases (Inversely correlated)

Now, that we know that the covariance matrix is not more than a table that summaries the correlations between all the possible pairs of variables, let’s move to the next step.

### STEP 3: COMPUTE THE EIGENVECTORS AND EIGENVALUES OF THE COVARIANCE MATRIX TO IDENTIFY THE PRINCIPAL COMPONENTS

Eigenvectors and eigenvalues are the linear algebra concepts that we need to compute from the covariance matrix in order to determine the **principal components** of the data. Before getting to the explanation of these concepts, let’s first understand what do we mean by principal components.

Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables. These combinations are done in such a way that the new variables (i.e., principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components. So, the idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on.

Geometrically speaking, principal components represent the directions of the data that explain a **maximal amount of variance**, that is to say, the lines that capture most information of the data. The relationship between variance and information here, is that, the larger the variance carried by a line, the larger the dispersion of the data points along it, and the larger the dispersion along a line, the more the information it has. To put all this simply, just think of principal components as new axes that provide the best angle to see and evaluate the data, so that the differences between the observations are better visible.

Logistic Regression

Logistic regression is another powerful supervised ML algorithm used for binary [classification](https://www.sciencedirect.com/topics/computer-science/classification) problems (when target is categorical). The best way to think about logistic regression is that it is a linear regression but for classification problems. Logistic regression essentially uses a [logistic function](https://www.sciencedirect.com/topics/computer-science/logistic-function) defined below to model a binary output variable. The primary difference between linear regression and logistic regression is that logistic regression's range is bounded between 0 and 1. In addition, as opposed to linear regression, logistic regression does not require a linear relationship between inputs and output variables. This is due to applying a nonlinear log transformation to the odds ratio.

Text

Description automatically generated

Logit function is defined as the natural log of the odds. A probability of 0.5 corresponds to a logit of 0, probabilities smaller than 0.5 correspond to negative logit values, and probabilities greater than 0.5 correspond to positive logit values. Logistic function ranges between 0 and 1 (P∈[0,1]) while logit function can be any real number from minus infinity to positive infinity (P∈[−∞, ∞]).

Text, letter

Description automatically generated

Random Forest

## Introduction

Random forest is a ***Supervised Machine Learning Algorithm* that is *used widely in Classification and Regression problems***. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing ***continuous variables*** as in the case of regression and ***categorical variables*** as in the case of classification. It performs better results for classification problems.

## Working of Random Forest Algorithm

Before understanding the working of the random forest we must look into the ensemble technique. ***Ensemble***simplymeans combining multiple models. Thus a collection of models is used to make predictions rather than an individual model.

*Ensemble uses two types of methods*:

1. **Bagging**– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example,  Random Forest.

2. **Boosting**– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example,  ADA BOOST, XG BOOST

#### Bagging

Bagging, also known as ***Bootstrap Aggregation*** is the ensemble technique used by random forest.Bagging chooses a random sample from the data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as ***row sampling***. This step of row sampling with replacement is called***bootstrap***. Now each model is trained independently which generates results. The final output is based on majority voting after combining the results of all models. This step which involves combining all the results and generating output based on majority voting is known as ***aggregation***.

Steps involved in random forest algorithm:

Step 1: In Random forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on ***Majority Voting or Averaging***for Classification and regression respectively.

### Important Features of Random Forest

1. Diversity**-**Not all attributes/variables/features are considered while making an individual tree, each tree is different.

2. Immune to the curse of dimensionality- Since each tree does not consider all the features, the feature space is reduced.

3. Parallelization**-**Each tree is created independently out of different data and attributes. This means that we can make full use of the CPU to build random forests.

4.  Train-Test split**-**In a random forest we don’t have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.

5.  Stability**-**Stability arises because the result is based on majority voting/ averaging.

Naive Bayes(BernoulliNB)

Bernoulli Naive Bayes is a part of the family of Naive Bayes. It only takes binary values. The most general example is where we check if each value will be whether or not a word that appears in a document. That is a very simplified model. In cases where counting the word frequency is less important, Bernoulli may give better results. In simple words, we have to count every value binary term occurrence features i.e. a word occurs in a document or not. These features are used rather than finding the frequency of a word in the document.

To understand it in layman’s terms,Bernoulli distribution has two mutually exclusive outcomes:P(X=1)=p or P(X=0)=1-p.In BernoulliNB theorem, we can have multiple features but each one is assumed to be binary valued variable i.e. boolean. Therefore, this class requires samples to be represented as binary-valued feature vectors. In case, any other kind of data, is provided, then a BernoulliNB instance may binarize its input.

The decision rule for Bernoulli naive Bayes is based on





According to the decision rule formula,x needs to be binary. Think about the formula in the case where xi=1 and the case where xi=0. So i is the event where xi=1 or the event where xi=0.

Decision Tree

Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets (or subpopulations) based on most significant splitter / differentiator in input variables. Decision trees are built using a heuristic called recursive partitioning. This approach is more commonly known as divide and conquer because it uses the feature values to split the data into smaller and smaller subsets of similar classes. At the beginning of the root node, which represents our entire dataset. The algorithm will choose a feature that is the most predictive of the target class. The examples are then partitioned into groups of distinct values of this feature; this decision forms the first set of tree branches. The algorithm continues to divide-and-conquer the nodes, choosing the best candidate feature each time until a stopping criterion is reached.

This could occur if

* All, or nearly all, of the examples at the node have the same class.
* There are no more remaining features to distinguish among examples.
* The tree has grown to a predefined size limit.

Types of Decision Trees

Types of decision tree is based on the type of target variable we have. It can be of two types:

1. Categorical Variable Decision Tree: Decision Tree which has categorical target variable then it called as categorical variable decision tree.

Example:- In above scenario of Kids example, where the target variable was “Kids will consume Nutritional drink or not” i.e. YES or NO.

2. Continuous Variable Decision Tree: Decision Tree has continuous target variable then it is called as Continuous Variable Decision Tree.

Example:- Let’s say we have a problem to predict whether a customer will pay his renewal premium with an insurance company (yes/ no). Here we know that income of customer is a significant variable but insurance company does not have income details for all customers. Now, as we know this is an important variable, then we can build a decision tree to predict customer income based on occupation, product and various other variables. In this case, we are predicting values for continuous variable.

Important Terminology related to Decision Trees

• Root Node: It represents entire population or sample and this further gets divided into two or more homogeneous sets.

• Splitting: It is a process of dividing a node into two or more sub-

nodes.

• Decision Node: When a sub-node splits into further sub-nodes,

then it is called decision node.

• Leaf/ Terminal Node: Nodes do not split is called Leaf or Terminal

node.

• Pruning: When we remove sub-nodes of a decision node, this

process is called pruning. You can say opposite process of splitting.

• Branch / Sub-Tree: A sub section of entire tree is called branch or

sub-tree.

• Parent and Child Node: A node, which is divided into sub-nodes is

called parent node of sub-nodes where as sub-nodes are the child

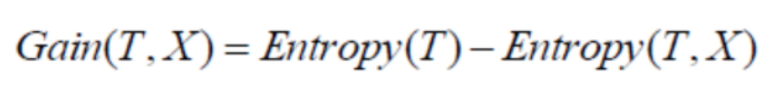
of parent node.

# C4.5

C4.5 decision tree is a modification over the ID3 Decision Tree. C4.5 uses the Gain Ratio as the goodness function to split the dataset, unlike ID3 which used the Information Gain.

The Information Gain function tends to prefer the features with more categories as they tend to have lower entropy. This results in overfitting of the training data. Gain Ratio mitigates this issue by penalising features for having a more categories using a formula called Split Information or Intrinsic Information.





Text

Description automatically generated