PCA and LDA Analysis Using R

Pattabiraman V

PCA and LDA

- Principal Component Analysis (PCA): iis used to identify the combination of attributes (principal components, or directions in the feature space) that account for the most variance in the data.
- Linear Discriminant Analysis (LDA) is tries to identify the attributes that account for the most variance *between classes*. In particular, LDA, in contrast to PCA, is a supervised method, using known class labels.

Introduction to PCA

- **Introduction**
- Improving predictability and classification one dimension at a time! “Visualize” 30 dimensions using a 2D-plot!
- PCAs primary purpose is **NOT** as a ways of feature removal! PCA can reduce dimensionality but **it wont reduce the number of features / variables in your data.**
- **Visualization methods**
- What this means is that you might discover that you can explain 99% of variance in your 1000 feature dataset by just using 3 principal components but you still need those 1000 features to construct those 3 principal components, this also means that in the case of predicting on future data you still need those same 1000 features on your new observations to construct the corresponding principal components.

Working Principles of PCA

- **Standardize the data** (Center and scale).
- **Calculate the Eigenvectors and Eigenvalues from the covariance matrix or correlation matrix** (One could also use Singular Vector Decomposition).
- **Sort the Eigenvalues in descending order and choose the *K* largest Eigenvectors** (Where *K* is the desired number of dimensions of the new feature subspace $k \leq d$).
- **Construct the projection matrix W from the selected *K* Eigenvectors.**
- **Transform the original dataset X via W to obtain a *K*-dimensional feature subspace Y.**

Property of PCA

- property of PCA is that our components are sorted from largest to smallest with regard to their standard deviation (**Eigenvalues**). So let's make sense of these:
- ***Standard deviation:*** This is simply the **eigenvalues** in our case since the data has been centered and scaled (**standardized**)
- ***Proportion of Variance***: This is the amount of variance the component accounts for in the data,
  ie. **PC1** accounts for **>44% of total variance** in the data alone!
- ***Cumulative Proportion***: This is simply the accumulated amount of explained variance, ie. if we used **the first 10 components** we would be able to account for **>95% of total variance** in the data.

Introduction to LDA

- Linear Discriminant Analysis (LDA) can be seen from two different angles.
- First, classify a given sample of predictors to the class with highest posterior probability.
- It minimizes the total probability of misclassification. To compute it uses Bayes' rule and assume that follows a Gaussian distribution with class-specific mean and common covariance matrix.
- Second tries to find a linear combination of the predictors that gives maximum separation between the centers of the data while at the same time minimizing the variation within each group of data.

Introduction to LDA

- The second approach is usually preferred in practice due to its dimension-reduction property and is implemented in many R packages, as in the *lda* function of the *MASS* package.
- **Step 1: Load Necessary Libraries**
- *library(MASS)*
- *library(ggplot2)*
- **Load the Data -** use the built-in **iris** dataset in R.
- **#attach *iris* dataset to make it easy to work with** *attach(iris)*
- **#view structure of dataset**
- *str(iris)*

Working with LDA

- Dataset contains 5 variables and 150 total observations.
- For this example we'll build a linear discriminant analysis model to classify which species a given flower belongs to.
- use the following predictor variables in the model:
- Sepal.length, Sepal.Width, Petal.Length, Petal.Width
- use them to predict the response variable *Species*, which takes on the following three potential classes:
- Setosa, versicolor, virginica

Working with LDA

- **Step 3: Scale the Data**
- One of the key assumptions of linear discriminant analysis is that each of the predictor variables have the same variance. An easy way to assure that this assumption is met is to scale each variable such that it has a mean of 0 and a standard deviation of 1.
- We can quickly do so in R by using the **scale()** function
- We can use the apply() function to verify that each predictor variable now has a mean of 0 and a standard deviation of 1:

Working with LDA

- **Step 4: Create Training and Test Samples**
- split the dataset into a training set to train the model on and a testing set to test the model on:
- **Step 5: Fit the LDA Model**
- use the lda() function from the **MASS** package to fit the LDA model to our data:

Interpretation of LDA output

- **Prior probabilities of group:** These represent the proportions of each Species in the training set. For example, 35.8% of all observations in the training set were of species *virginica*.
- **Group means:** These display the mean values for each predictor variable for each species.
- **Coefficients of linear discriminants:** These display the linear combination of predictor variables that are used to form the decision rule of the LDA model. For example:
- **LD1:** .792*Sepal.Length + .571*Sepal.Width – 4.076*Petal.Length – 2.06*Petal.Width
- **LD2:** .529*Sepal.Length + .713*Sepal.Width – 2.731*Petal.Length + 2.63*Petal.Width
- **Proportion of trace:** These display the percentage separation achieved by each linear discriminant function.

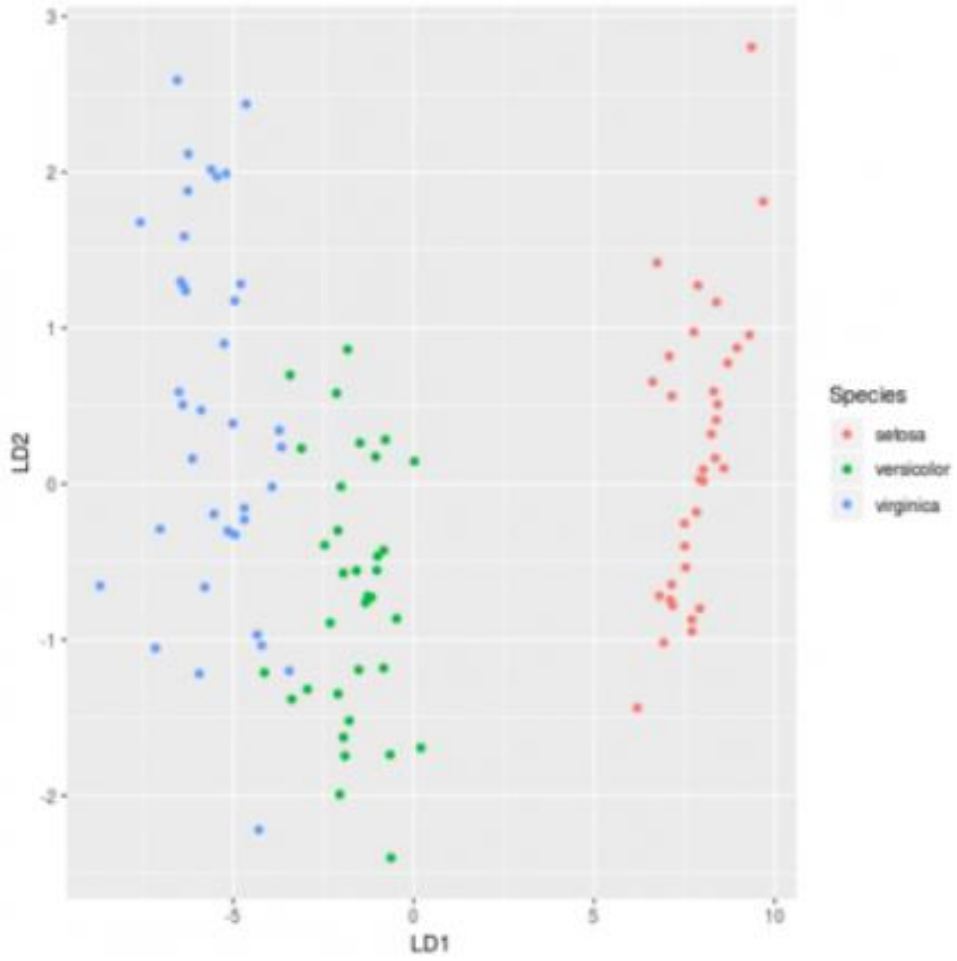
LDA Model to Make Predictions

- **Step 6: Use the Model to Make Predictions**
- fit the model using our training data, we can use it to make predictions on our test data:
- This returns a list with three variables:
- **class:** The predicted class
- **posterior:** The posterior probability that an observation belongs to each class
- **x:** The linear discriminants
- view each of these results for the first six observations in our test dataset: **head(predicted$class)**
- **head(predicted$posterior)**
- **head(predicted$x)**
- use the following code to see what percentage of observations the LDA model correctly predicted the Species for:

LDA Model to Make Predictions

- use the following code to see what percentage of observations the LDA model correctly predicted the Species for:
- **mean(predicted$class==test$Species)**
- It turns out that the model correctly predicted the Species for **100%** of the observations in our test dataset.
- In the real-world an LDA model will rarely predict every class outcome correctly, but this iris dataset is simply built in a way that machine learning algorithms tend to perform very well on it.
- **Step 7: Visualize the Results**
- create an LDA plot to view the linear discriminants of the model and visualize how well it separated the three different species in our dataset:
- **lda_plot <- cbind(train, predict(model)$x)**
- **ggplot(lda_plot, aes(LD1, LD2)) + geom_point(aes(color = Species))**

- **OUTPUT**



Thank you