Applications of Pinciple Component Analysis and Linear Discriminant Analysis on Consulting Survey Data

Ruinan(Victor) Zhang

March 24, 2017

1 Abstract

This paper explains the concepts of a statistical techinique, linear discriminant analysis (LDA) and how it can be used on analyzing surveying data and visualization for business consulting. One situation that LDA can be extremely helpful is to compare and contrast customers' expectations and actual ratings on products. I will also explain how LDA works for visualization and process of implementing an online application using R language through ShinyIO.

2 Motivation

The project is inspired by company, Smari Inc. The challenge was to provide statistical evidence for consulting business decisions by visualizing collected survey data. One specific question is how to evaluate competitors' performance on specific features. For example, cellphone brands may interested to know what do the customers think of them in terms of design, user-friendness, system-smoothiness. The goal of this project is to build an open-source online app for consulting companies to upload their dataset and construct visulizations to help them to answer such consulting questions.

In the sample dataset provided by the company, the customers rated on some features on a brand of recorders and the importance level of these features on such product within in a range of 1 throught 7. In this specific case, the challenge is to extract information from these survery dataset and visualize the distance between the customers' expection and rating on the recorder.

3 Overview and Interpretation on Visualization

The statistical method linear discriminant analysis is first applied to the dataset to transform the dataset into a dimension where the discriminance between features are maximized, and then the datset is plotted on the new dimensions for visualization. Here is the generated LDA plot on the sample dataset:

To explain this dataset a little bit, response 1 through 20 are customers' ratings on features of specific brand of recorders, and response 'q51_1' through 'q51_20' are customers' ratings on importance of these features. On this graph, the dots represent customers' actual ratings while the vectors represent customers' expectations. For example, survery question'1' asks about visual appealing and the fact that the vector for 'q1' is pointing into a different direction from the dot for 'q1' means this product's visual performance is way off from customers' expectations.

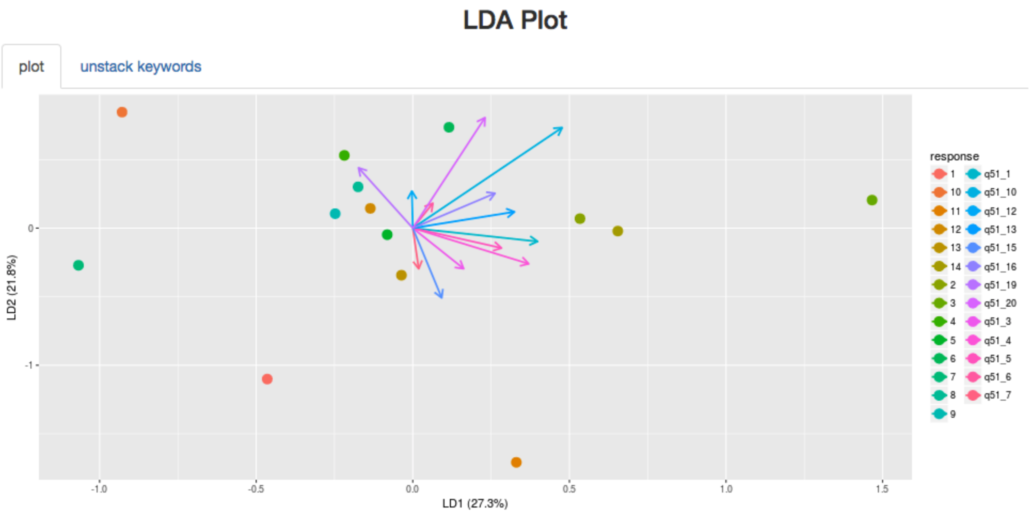

Figure 1: Linear Discriminant Plot on Sample Data

4 Statistics Concepts Involved

4.1 Linear Discriminant Analysis

Linear discriminant analysis(LDA) is a method used in statistics to find a linear combination of features that characterizes or separates two or more classes of objects or event. The idea of LDA is to find a linear combination of variables to model the difference between the classes of a particular set of variable (usually categorical variables). LDA technique is frequently used as dimensionality reduction before classifications. LDA can be visualized similar to PCA as they both generate set of linear coefficients.

The idea behind Fisher's LDA is to find a seperate populations by find linear combinations $Y = a^T X$ which has expected value $E(Y) = a^T \times E(X|\pi_i) = a^T \times \mu_i$ for pipulation π_i and variance $Var(Y) = a^T \times Cov(X)a = a \times \Sigma a$. The between-class scatter matrix is defined as $B_\mu = \sum(\mu_i - \mu_0)(\mu_i - \mu_0)^T$ where μ_0

is the population mean. The optimizing criterian is the ratio between “sum of squared distances from populations to overall mean of Y” and “variance of Y”, in formula:

$$\frac{S_B}{S_W} = \frac{a^T B_\mu a}{a^T \Sigma a}$$

It turns out the elements of a which maximize $\frac{S_B}{S_W}$ are eigenvalues of $\frac{B_\mu}{\Sigma}$. Here is a brief outline of the proof:

Suppose $e_1, e_2, ...e_s$ and $\lambda_1, \lambda_2, ...\lambda_s$ are the corresponding eigenvalues for $\frac{B_\mu}{\Sigma}$, and $e_1, e_2, ...e_s$ are the correponding eigenvectors and scaled so that $e^t \Sigma e = 1$ The linear combination $a_i^T \times X$ is called the (i)th discriminant. The goal is to maximize the ratio $\frac{S_B}{S_W}$ with subject to $0 = Cov(a_i^T X, a_j^T X)$By the eigendecomposition, $\Sigma = P^T A P$ where A is a disgonal matrix with positive elements of λ_i. Let $A^{1/2}$ denote the isgonal matrix with element of $\sqrt{\lambda_i}$. We can easily have $\Sigma^{1/2} = P^T A^{1/2} P$ and $\Sigma^{-1/2} = P^T A^{-1/2} P$. Next, set

$$u = \Sigma^{1/2} a$$

so $u^T u = a^T \Sigma a$, then the maximizing criteria $\frac{S_B}{S_W}$ can be transformed into

$$\frac{u^T \Sigma^{-1/2} B_\mu \Sigma^{-1/2} u}{u^T u}$$

For $i = 1$, when $e_1 \mu_= \Sigma^{-1/2}$,

$$Var(a_1^T X) = a_1^T \Sigma a_1 = e_1^T \Sigma^{-1/2} \Sigma \Sigma^{-1/2} e_1 = e_1^T e_1 = 1$$

For $i = 1$, when $e_2 \mu_= \Sigma^{-1/2}$,

$$Var(a_2^T X) = a_2^T \Sigma a_2 = e_2^T \Sigma^{-1/2} \Sigma \Sigma^{-1/2} e_2 = e_2^T e_2 = 1$$

Continue in this fashion, and based on the fact e is scaled so that $e^t \Sigma e = 1$, we can easily get:

$$\Sigma^{-1/2} B_\mu \Sigma^{-1/2} = \lambda e$$

and then multiply $\Sigma^{-1/2}$ on the left side gives

$$\Sigma^{-1} B_\mu (\Sigma^{-1/2} e) = \lambda((\Sigma^{-1/2} e))$$

which indicates

$$\Sigma^{-1}B_{\mu} = \frac{B_{\mu}}{\Sigma} = \lambda$$

.

5 Online App on ShinyIO

5.1 Application Details

The online PCA and LDA visualization online applications can be find *https://zhangruinan.shinyapps.io/shinny* and *https://zhangruinan.shinyapps.io/LDA_plot/* Descriptions on the application details including: functionality, default dataset, can be find at *https://github.com/zhangruinan/Biplot_Anal* and *https://github.com/zhangruinan/LDA-Plot*. We now outline some of these details

- Upload excel sheet as input
- Horizontal unstacking data based on keyword matching
- Visualization choice including centroids, scatters, and vectors

5.2 Implimentation Techniques

To give a very brief view of how Shinny IO works, the online application is run by two major R scripts: server.R and ui.R. The ui.R provides customerized user interface widgets like checkboxs or text input. The input from the user interface is then passed to server.R which handles the computation and generate output. The generated output can be then passed back to UI and displayed to users on the website.

References

[1] https://onlinecourses.science.psu.edu/stat505/node/49

[2] S. Balakrishnama, A. Ganapathiraju. *Linear Discriminant Analysis*
https://www.isip.piconepress.com/publications/reports/1998/isip/lda/lda_theory.pdf

[3] Sebastian Raschka
http://sebastianraschka.com/Articles/2014_python_lda.html