# MATH 748: Weekly Report

Oct 4-10, 2021

Nayeong Kim (nkim10@sfsu.edu)

# 6 Binary Classification: LDA and LR

## Reminder: LDA and LR

(1) LDA (Linear Discriminant Analysis)
Assume $X|Y=j \sim MVN(\mu_j, \Sigma)$ for $j=0,1$. In other words,

$$log\frac{P(Y=1|X=x)}{P(Y=0|x=x)} = \frac{\pi_1 f_1(x)}{\pi_0 f_0(x)} = \beta_0 + \beta_1^T x$$

Following Bayes classifier: $\hat{Y} = \underset{j=0,1}{\text{argmax}} P(Y=j|X=x)$. Equivalently, $\hat{Y}=1$ if $\beta_0+\beta_1^T x > 0$ and $\hat{Y}=0$ if $\beta_0+\beta_1^T x < 0$. Decision boundary is $\beta_0+\beta_1^T x = 0$.

(2) LR (Logistic Regression)
The model is as below.

$$log\frac{P(Y=1|X=x)}{P(Y=0|X=x)} = log\frac{p}{1-p} = \beta_0+\beta_1^T x \quad \text{where } p = P(Y=1|X=x)$$

$$\Rightarrow P(Y=1|X=x) = \frac{e^{\beta_0+\beta_1^T x}}{1+e^{\beta_0+\beta_1^T x}}$$

Estimators are MLE of conditional likelihood. Assume $Y_i = y_i|x_i \sim p^{y_i}(1-p)^{1-y_i}$ where $y_i = 0,1$. The joint likelihood is

$$\text{JCL} = \prod_{i=0}^{n} p(x_i;\beta)(1-p(x_i;\beta)^{1-y_i}$$

$$log\text{JCL} = \sum_{i=1}^{n} y_i log\, p(x_i;\beta) + (1-y_i) log(1-p(x_i;\beta))$$

We can get $\hat{\beta}$ by *Newton-Raphson* method.

## 6.5 Relation between LDA and OLS(Ordinary Least Square)

(1) Ordinary Least Squre
Assume $y = \beta_0+\beta_1^T x+\varepsilon$ and $\hat{y}=1$ if $\beta_0+\beta_1^T x > 0.5$.

(2) Relation between $(\hat{\beta_0}^{LDA}, \hat{\beta_1}^{LDA})$ and $(\hat{\beta_0}^{OLS}, \hat{\beta_0}^{OLS})$

- $\hat{\beta_1}^{LDA} \propto \hat{\beta_1}^{LS}$
  The slope coefficients are identical up to a scalar multiplier.
- $\hat{\beta_0}^{LDA} = \hat{\beta_0}^{OLS}$ if $n_0 = n_1$, $\hat{\beta_0}^{LDA} \neq \hat{\beta_0}^{OLS}$ elsewise (where $n_0$ and $n_1$ are sample sizes)

## 6.6 Relation between LDA and LR

(1) Common things
They have the same linear boundary by setting $log\frac{p}{1-p} = \beta_0 + \beta_1^T x$.

(2) Different things

- Where the linear-logit form is from:
  **LDA**: Due to the assumption $\Sigma_0 = \Sigma_1$. In indirect way.
  **LR**: Due to the construction. In direct way.
- Difference in marginal distribution assumptions for $X$:
  **LDA**: $p(x) = \Pi_0 f_0(\mu_0, \Sigma) + \Pi_1 f_1(\mu_1, \Sigma)$ where $f_i(\mu_i, \Sigma) \sim \text{MVN}(\mu_i, \Sigma)$
  **LR**: Arbitrary. (More general.)
- Difference in parameter estimation:
  **LDA**: By maximizing marginal density of $X$

$$\hat{\beta}^{LDA} = \underset{\beta}{\text{argmax}} \sum_{i=1}^{n} log\ p(x_i)$$

  **LR**: By maximizing conditional density of $y_i|X = x_i$

$$\hat{\beta}^{LR} = \underset{\beta}{\text{argmax}} log \sum_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}$$

LDA is easier to compute and LDA works better when Gaussian assumption is reliable. Meanwhile, LR is more robust. In practice, they give similar results.

# 7 Binary Classification: High Dimension

This section is about high dimension classification problems.

## 7.1 High Dimension

This subsection is about high dimensional data.

Let $p$ be the dimension of the data and $n$ be the sample size. In the old days, $n$ used to be larger than $p$. In now days, various data can have larger dimensions than sample sizes: $p > n$ or even $p >> n$. For example, genetic data has a huge dimension.

For a high-dimensional data, we have to think of *Curse of Dimensionality*. As $p$ increases, volume of the space increases so fast that the available data becomes very sparse. Therefore, i) high dimensional functions are harder to estimate, ii) local mehtods are less local with high dimensional data set iii) high dimensional data set make it more difficult to find neighbors.

Let us think of a $p$-dim hyper cube having uniformly distributed points in it. If we want to capture a fraction $r$ of the observations, the edge length $I$ of this small hyper cube. Then $I^p = r$. When $p = 100$, to get 0.01 of the observations, we must cover 0.95 length of the input.

In another illustration, we can think of how many samples are required for $p$ features. Assuming $n$ samples are required to represent a dense sample for one feature, we need $n^p$ to have the same sample density with $p$ features.

(1) Statistical and data mining challenges

a. We need a large sample size. (We've shown that $n$ needs to grow exponentially with $p$.)

b. Computational burden (e.g. Best subset selection requires $O(p!)$ computations.)

c. Redundant or useless features

d. Strong multi-co-linearity among features

e. When $p > n$, not enough data to determine all the parameters uniquely.

These are the challenges for high-dimensional data analysis. Furthermore, there is *rules of thumb* which say that $n$ should be at least 5 times or more than $p$ to get stable solutions for $p < 15$.

(2) Modern high dimensional classifier

- Modified LDA
- Modified LR
- SVM, Tree, Boosting : *Will be covered later*

## 7.2 Modified LDA

(1) Key Idea 1: Replace $\hat{\Sigma}$ by an alternative positive definite matrix.
In [1] (Bickel, Levina, 2008), *Indeppendent Rule* is mentioned. With this, we can assume that $\hat{\Sigma}$ is a diagonal matrix. This reduces the burden on the high complexity of $\Sigma$ by reducing $p \times p$ dimension to $p$ dimension.

Despite of the imperfect model specification, it can outperform a rule that intends to model all the correlation.

(2) Key Idea 2: Improve the estimate $\Sigma^{-1}(\mu_1 - \mu_0)$

- NSC(Nearest Shrunken Centroids) [2]
  We can interpret LDA as classifying points to its nearest centroid. The distance here is the *Mahalanobis distance*: $\sigma_j(x) = -(x-\mu_j)^T\Sigma^{-1}(x-\mu_j) + 2log\pi_j$.
  In [2], it proposes to use *denoised* version of the centroid. Do standardization for each feature and assume $\widetilde{\Sigma} = D + s_0^2 I$

- DSDA(Direct Sparse Discriminant Analysis)
  Idea: Relabel $y_0 = -\frac{n}{n_0}$, $y_1 = \frac{n}{n_1}$ and run linear regression.

$$\hat{\beta_1}^{OLS} \propto \hat{\beta_1}^{LDA}$$

$$(\hat{\beta}_0, \hat{beta}^{DSDA}) = \underset{\beta}{\text{argmax}} \sum_{i=1}^{n} (y_i - \beta_0 - \beta^T x_i)^2 + p_\lambda(\beta)$$

  $p_\lambda(\beta)$ is a penalty term. $x$ is classified to be class 1 if $x^T\hat{\beta}^{DSDA} + \hat{\beta}_0 > 0$

## 7.3 Modified Logistic Regression

Original LR gets the estimator of $\beta$:

$$\hat{\beta}^{LR} = \underset{\beta}{\text{argmax}} \sum_{i=1}^{n} log\ p^{y_i}(1-p)^{1-y_i} = \underset{\beta}{\text{argmax}}\text{JCL}$$

In modified LR, it uses regularization. In other words, it gives the penalty in the loss function: $J(\beta)$ to limit the size of the parameters.

$$\hat{\beta}^{LR} = \underset{\beta}{\text{argmin}} - \text{JCL} + \lambda J(\beta)$$

There are various ways to give the penalty term. *Lasso* gives $J(\beta) = \sum|\beta_j|$ and *Ridge* gives $J(\beta) = \sum\beta_j^2$.

# 8 KNN

## 8.1 Need for Nonlinear Classifier

When we simulate the probability model composing multiple normal models, linear models don't perform well. For implementing more flexible models, it is necessary to allow nonlinear boundaries for decision models. There are some exaples of nonlinear classifiers: NN(Nearest Neighbor Classifier), SVM, trees, random forests and boosting.

## 8.2 KNN

The idea of KNN($k$ Nearest Neighbors) is to classify object based on $k$ closest training examples in the sample space.

# References

[1] Covariance regularization by thresholding Bickel, Levina, 2008

[2] Diagnosis of multiple cancer types by shrunken centroids of gene expression Tibshirani et al, 2002