High Dimensional Classification Problems

Hao Helen Zhang

Linear Binary Classification for High Dimensional Data

- Curse of Dimensionality
- Sparse LDA, penalized LDA, Nearest Shrunken Centroid (NSC)
- Penalized Logistic Regression

High Dimensional Data

- Let d represent the data dimension.
- Let n represent the sample same size.

In the old days, we lived in the low-dimensional world ($d < n$)

- the physical space commonly modeled with just three dimensions.
- $n \sim 10 - 10^2, d \sim 10$.
- It is uncommon to encounter $d = 3$ or $d = 4$
- For example, Iris Data Set

Curse of Dimensionality

We now live in the world of high dimensionality

- d is at the scale of hundreds, thousands, millions, ...
- microarray gene expression: $d \sim 10^2 - 10^4, n \sim 10 - 100$
- SNP data: $d \sim 10^6, n \sim 10^2 - 10^3$

One common feature: $d > n$ or $d \gg n$.

Curse of dimensionality (Bellman 1961)

- the phenomena arise when analyzing and organizing data in high-dimensional spaces that do not occur in low-dimensional settings.

Curse of Dimensionality Phenomena

When the dimensionality d increases, the volume of the space increases so fast that the available data becomes sparse.

- High dimensional functions tend to have more complex features than low-dimensional functions, and hence harder to estimate.
- Local methods are *less local* when the dimension d increases
- Neighborhoods with fixed k points are less concentrated as d increases.

Illustration 1

Suppose the points are uniformly distributed in a d-dimensional unit hypercube.

Question: If we want to construct a hypercube neighborhood of x_0 to capture a fraction r of the observations, what is the edge length l of this cube?

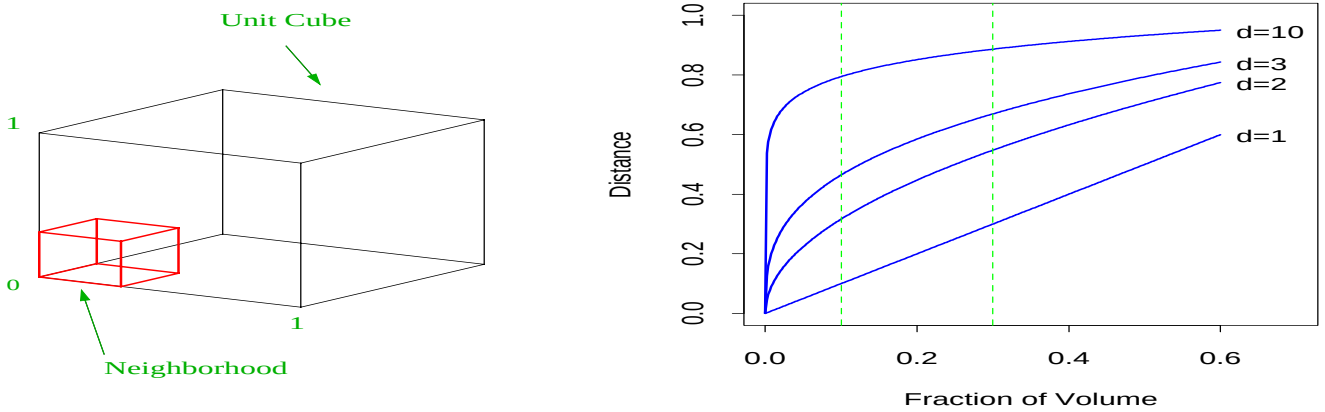

Figure 2.6: *The curse of dimensionality is well illustrated by a subcubical neighborhood for uniform data in a unit cube. The figure on the right shows the side-length of the subcube needed to capture a fraction r of the volume of the data, for different dimensions p. In ten dimensions we need to cover 80% of the range of each coordinate to capture 10% of the data.*

Answer to Illustration 1

Since the volume of cube

$$l^d = r,$$

we have $l = r^{1/d}$.

- When d=1,
  - If $r = 0.01$ then $l = 0.01$; if $r = 0.1$ then $l = 0.1$.
- When d=10,
  - If $r = 0.01$ then $l = 0.63$; if $r = 0.1$, then $l = 0.80$.
  - In order to capture 1% (or 10%) of the data, we must cover 63% (80%) of the range of each input.

Illustration 2

If $n = 100$ represents a dense sample for one single input, we need $n = 100^{10}$ to have the same sampling density with $d = 10$.

- The number of required points increases exponentially to maintain the same sampling density.

Statistical and Data Mining Challenges

Challenges:

- To estimate multivariate functions with the same accuracy as in low dimensions, n needs to grow exponentially with d.
- Organizing data often relies on detecting areas where objects form groups with similar properties. When d is large, all objects are sparse and dissimilar in many ways, which makes organization strategies less efficient.
- Computational burden
  - best subset selection: there are $\binom{d}{k}$ models of a given size k.

Challenges in High Dimensional Classification Problems

- redundant or useless covariates for prediction.
- strong multi-collinearity among X's
- When $d > n$, not enough data (or degree-of-freedom) to determine all the parameters uniquely.

Consequently, when $d > n$,

- Linear/logistic regression do not have unique solutions
- LDA does not guarantee the sample covariance matrix to be positive definite, thus not invertible
- "Overfitting": no bias on the training data but high variance

Rule of thumb: for a moderate d (say $d < 15$), n should at least be five times or more than d to get a stable solution.

Modern High-dimensional Classifiers

- LDA-type methods
  - Naive Bayes (NB), Nearest Shrunken Centroid (NSC)
  - sparse LDA, regularized LDA, penalized LDA
- penalized logistic regression
- large-margin methods
  - Support Vector Machines (SVMs)
- classification tree
  - random forest
- boosting

Review of Linear Discriminant Analysis

Recall that LDA classify a point $\mathbf{x}$ to "1" if

$$\beta_0 + \mathbf{x}^T \beta > 0,$$

where $\beta_0 = \log \frac{\pi_1}{\pi_0} - \frac{1}{2}(\mu_1 + \mu_0)^T \Sigma^{-1}(\mu_1 - \mu_0)$, and

$$\beta = \Sigma^{-1}(\mu_1 - \mu_0).$$

Classical LDA use the sample covariance matrix $\hat{\Sigma}$ to estimate Σ

$$\hat{\Sigma} = \sum_{k=0}^{1} \sum_{Y_i = k} (\mathbf{x}_i - \hat{\mu}_k)(\mathbf{x}_i - \hat{\mu}_k)^T / (n-2),$$

where $\hat{\mu}_k = \sum_{Y_i = k} \mathbf{x}_i / n_k$ for $k = 0, 1$.

Challenges to LDA

However, when $d > n$,

- the sample covariance matrix $\hat{\Sigma}$ is not full rank and hence not invertible.
- classical LDA does not work any more.

How to adapt LDA to high dimensional data?

High Dimensional LDA

Key Idea 1: replace $\hat{\Sigma}$ by an alternative positive definite matrix

- Independence rule (proposed by Bickel & Levina 2008)
- Features annealed independence rule (FAIR; Fan and Fan 2008)
- Cai et al. (2010), Rothman et al. (2008), Cai & Zhou (2012).

Procedure: first obtain an invertible $\hat{\Sigma}$, and then apply the LDA.

Naive Bayes (NB) Rule

Bickel and Levina (2004)

- also known as Independence Rule (IR)

Main idea: replace $\hat{\Sigma}$ by $\hat{D} = \text{diag}(\hat{\Sigma})$

- $\hat{D}$ is always positive definite
- equivalent to treating all variables as independent within groups

Naive Bayes (IR) decision rule:

- estimate μ_{kj} by

$$\hat{\mu}_{kj}^{IR} = (1 - r_{jn})_+ \hat{\mu}_{kj}, \quad k = 0, 1; \quad j = 1, \ldots, d,$$

  where r_{jn} is a tuning parameter.
- plug $\hat{\mu}^{IR}$ and $\hat{D}$ into the LDA.

Comments on Naive Bayes (NB) Rule

Although this treatment is a model misspecification, theoretical studies show surprisingly that IR can outperform a rule that intends to model all the correlation.

- Bickel and Levina (2004)

Sparse LDA

Key Idea 2: improve the estimation of $\Sigma^{-1}(\mu_1 - \mu_0)$

- An accurate estimate of Σ does not guarantee better classification rule.
- The estimation of means also play a role.
  - Fan & Fan (2008) provided proofs that show, even when the independence structure is true, the signal can be swamped by the noises from estimating the means.

What truly matters is the estimation of the product $\Sigma^{-1}(\mu_1 - \mu_0)$

- Sparse LDA, penalized LDA, regularized LDA

Discriminating Functions for LDA

Classify to "1" if

$$\begin{aligned}
\log \frac{\Pr(Y=1|\mathbf{X}=\mathbf{x})}{\Pr(Y=0|\mathbf{X}=\mathbf{x})} &> 0 \\
\log[g_1(\mathbf{x})] + \log \pi_1 &> \log[g_0(\mathbf{x})] + \log \pi_0 \\
-\frac{1}{2}(\mathbf{x}-\mu_1)^T\Sigma^{-1}(\mathbf{x}-\mu_1) + \log \pi_1 &> -\frac{1}{2}(\mathbf{x}-\mu_0)^T\Sigma^{-1}(\mathbf{x}-\mu_0) + \log \pi_0 \\
-(\mathbf{x}-\mu_1)^T\Sigma^{-1}(\mathbf{x}-\mu_1) + 2\log \pi_1 &> -(\mathbf{x}-\mu_0)^T\Sigma^{-1}(\mathbf{x}-\mu_0) + 2\log \pi_0
\end{aligned}$$

Define the discriminating functions for each class:

$$\delta_k(\mathbf{x}) = -(\mathbf{x}-\mu_k)^T\Sigma^{-1}(\mathbf{x}-\mu_k) + 2\log \pi_k, \quad k = 0, 1.$$

Classify a point $\mathbf{x}$ to "1" if

$$\delta_1(\mathbf{x}) > \delta_0(\mathbf{x}).$$

Another Interpretation of LDA

The LDA rule classifies a point $\mathbf{x}$ to Class 1, if

$$(\mathbf{x}-\mu_1)^T\Sigma^{-1}(\mathbf{x}-\mu_1)-2\log\pi_1 < (\mathbf{x}-\mu_0)^T\Sigma^{-1}(\mathbf{x}-\mu_0)-2\log\pi_0,$$

and to Class 0, otherwise. Here μ_1 and μ_0 are the centroid of Class 1 and Class 0.

- The term
$$(\mathbf{x}-\mu_k)^T\Sigma^{-1}(\mathbf{x}-\mu_k), k=0,1$$
is known as the **Mahalanobis distance** of $\mathbf{x}$ to each class.
- This can be regarded as a "normalized" distance after taking into account the standard deviations.

The LDA rule: classify a point to its nearest centroid.

Nearest Shrunken Centroids (NSC)

Tibshirani et al. (2002).

- propose to use "de-noised" versions of the centroids

Main ideas:

- shrink the class centroids towards the overall centroids after standardizing by the within-class standard deviation for each variable (or "gene" used in the paper)
- This standardization has the effect of giving higher weight to variables (genes) whose expression is stable within samples of the same class.

Algorithm of Nearest Shrunken Centroids (NSC)

NSC algorithm:

1. replace $\hat{\Sigma}$ by a diagonal estimator

$$\tilde{\Sigma} = \hat{D} + s_0^2 \mathbf{I},$$

where $\hat{D} = \text{diag}(\hat{\Sigma})$, $s_0 > 0$ is a small constant, and $\mathbf{I}$ is the identity matrix.

2. for each dimension j, shrink $\hat{\mu}_j$'s to its grand mean by

$$t_{kj}^* = \frac{\hat{\mu}_{kj} - \hat{\bar{\mu}}_j}{m_k(s_j + s_0)}, \quad k = 0, 1; \quad j = 1, 2, \cdots, d$$

where

$$\hat{\bar{\mu}}_j = \frac{1}{n}\sum_{i=1}^{n} X_{ij}, \; m_k = \sqrt{\frac{1}{n_k} + \frac{1}{n}}, \; s_j^2 = \hat{D}_{jj}.$$

The presence of s_0 protect X_j's from having large t_{kj}'s by chance.

NSC

NSC shrinks t^*_{kj} towards zero

$$t'_{kj} = \text{sign}(t_{kj^*})(t^*_{kj} - \Delta)_+.$$

Then the NSC decision rule is done by

- define

$$\hat{\mu}'_{kj} = \hat{\bar{\mu}}_j + m_k(s_j + s_0)t'_{kj}, \quad j = 1, \cdots, d; \quad k = 0, 1.$$

- define the discriminating function as

$$\delta_{PAM}(\boldsymbol{X}) = \arg\min_k (\boldsymbol{X} - \hat{\boldsymbol{\mu}}'_k)^T (\tilde{\Sigma})^{-1} (\boldsymbol{X} - \hat{\boldsymbol{\mu}}'_k) - 2\log \hat{\pi}_k.$$

This brings variable selection into NSC

- When Δ is sufficiently large, one will have many X_j's with

$$\hat{\mu}'_{1j} = \hat{\mu}'_{2j} = \hat{\bar{\mu}}_j.$$

  These variables will have no effect on $\delta_{PAM}(\boldsymbol{X})$.

Practical Implementation of PAM

R Package: PAM (prediction analysis for microarrays)

Model tuning:

- s_0 is set to be the median of s_j's
- Δ is chosen by cross validation

An R package *pamr* is available.

Direct Sparse Discriminant Analysis (DSDA)

Mai et al. (2012): recast LDA as a linear regression problem

$$(\hat{\beta}_0^{ols}, \hat{\boldsymbol{\beta}}^{ols}) = \arg\min_{\beta_0, \boldsymbol{\beta}} \sum_{i=1}^{n} (Y_i - \beta_0 - \boldsymbol{X}_i^T \boldsymbol{\beta})^2,$$

where relabel $Y_2 = n/n_2, Y_1 = -n/n_2$. Y is still the class label but treated as continuous here. Then

$$\hat{\boldsymbol{\beta}}^{ols} = c(\hat{\Sigma})^{-1}(\hat{\mu}_1 - \hat{\mu}_0) \propto \hat{\boldsymbol{\beta}}^{bayes},$$

which have the same direction.

Main Idea

Consider

$$(\hat{\beta}_0, \hat{\boldsymbol{\beta}}^{DSDA}) = \arg\min_{\beta_0, \boldsymbol{\beta}} \sum_{i=1}^{n} (Y_i - \beta_0 - \boldsymbol{X}_i^T \boldsymbol{\beta})^2 + P_\lambda(\boldsymbol{\beta}),$$

where P_λ is a penalty function.

The decision rule classifies $\mathbf{x}$ to class $+1$ is

$$\boldsymbol{x}^T \hat{\boldsymbol{\beta}}^{DSDA} + \hat{\beta}_0^{opt} > 0,$$

where β_0^{opt} is discussed in Mai et al. (2012).

Penalized Logistic Regression

Minimize the penalized negative likelihood function

$$\min_{\beta} \sum_{i=1}^{n} \left(-y_i(\beta_0 + \mathbf{x}_i^T \beta) + \log(1 + e^{\beta_0 + \mathbf{x}_i^T \beta}) \right) + \lambda J(\beta).$$

Choices of penalty functions: (details later)

- LASSO $J(\beta) = \sum_{j=1}^{d} |\beta_j|$ (Tibshirani, 1996)
- bridge penalty $J(\beta) = \sum_{j=1}^{d} |\beta_j|^q$ (Frand and Friedman, 1993).
- adaptive LASSO
- SCAD
- elastic net