# CMSC702 Supervised Learning Methods

Hector Corrada Bravo

April, 2014

## Introduction

Today we discuss the classification setting in detail. Our setting is that we observe for each subject $i$ a set of $p$ predictors (covariates) $x_i$, and *qualitative* outcomes (or classes) $g_i$, which can takes values from a discrete set $G$. In this class we assume that predictors are genomic measurements (gene expression for now), and that we have many more measurements than samples (i.e. $p << N$, where $N$ is the number of subjects). For gene expression, predictors are real numbers, and we can think of input, or feature, space, as Euclidean space.

Since our prediction $\hat{G}(x)$ will always take values in the discrete set $G$, we can always divide the input space into a collection of regions taking the same predicted values.

The question is: what is the best sub-division of this space? The boundaries of this partition can be smooth or rough depending on the prediction function.

For an important class of procedures, these *decision boundaries* are linear. This is what we will refer to as linear methods for classification. We will see that these can be quite flexible (much more than linear regression).

Suppose we have $K$ classes labeled $1, \dots, K$ and a single predictior $X$. We can define a $0-1$ indicator for each class $k$, and perform regression for each of these. We would end up with a regression function $f_k(x) = \hat{\beta}_{0k} + \hat{\beta}_{1k}x$ for each class $k$.

The decision boundary between class $k$ and $l$ is simply the set for which $\hat{f}_k(x) = \hat{f}_l(x)$, i.e., $\{x : \hat{\beta}_{0k} + \hat{\beta}_{1k}x = \hat{\beta}_{0l} + \hat{\beta}_{1l}x\}$ which is a hyperplane.

Since the same holds for any pair of classes, the division of the input space is piecewise linear.

This regression approach is one of a number of methods that model a *discriminant function* $\delta_k(x)$ for each class, and classifies $X$ to the class with the largest value of the discriminant function.

Methods that model the posterior probability $Pr(G = k|X = x)$ are also in this class. If this is a linear function of $x$ then we say it is linear. Furthermore, if it is a monotone transform of a linear function of $x$, we will sometimes say it is linear. An example is *logistic regression.*

Both decision boundaries shown in the next Figure are linear: one obtained with linear regression, the other using quadratic terms.

## Linear regression of an indicator matrix

Each response is coded as a vector of 0-1 entries. If $G$ has $K$ classes, then $Y_k$ for $k = 1, \dots, K$ is such that $Y_k = 1$, if $G = k$ and 0 otherwise.

These are collected in a vector $Y = (Y_1, \dots, Y_k)$ and the $N$ training vectors are collected in an $N \times K$ matrix, we denote by $\mathbf{Y}$.

For example, if we have $K = 5$ classes

$$\begin{array}{ccccccc} 1 & & 0 & 0 & 0 & 0 & 1 \\ 2 & & 0 & 0 & 0 & 1 & 0 \\ 3 & \to & 0 & 0 & 1 & 0 & 0 \\ 4 & & 0 & 1 & 0 & 0 & 0 \\ 5 & & 1 & 0 & 0 & 0 & 0 \end{array}$$

On the left is the original data, and on the right, the coded data.

To fit with regression to each class simulatneously, we can simply use the same matrix multiplication trick

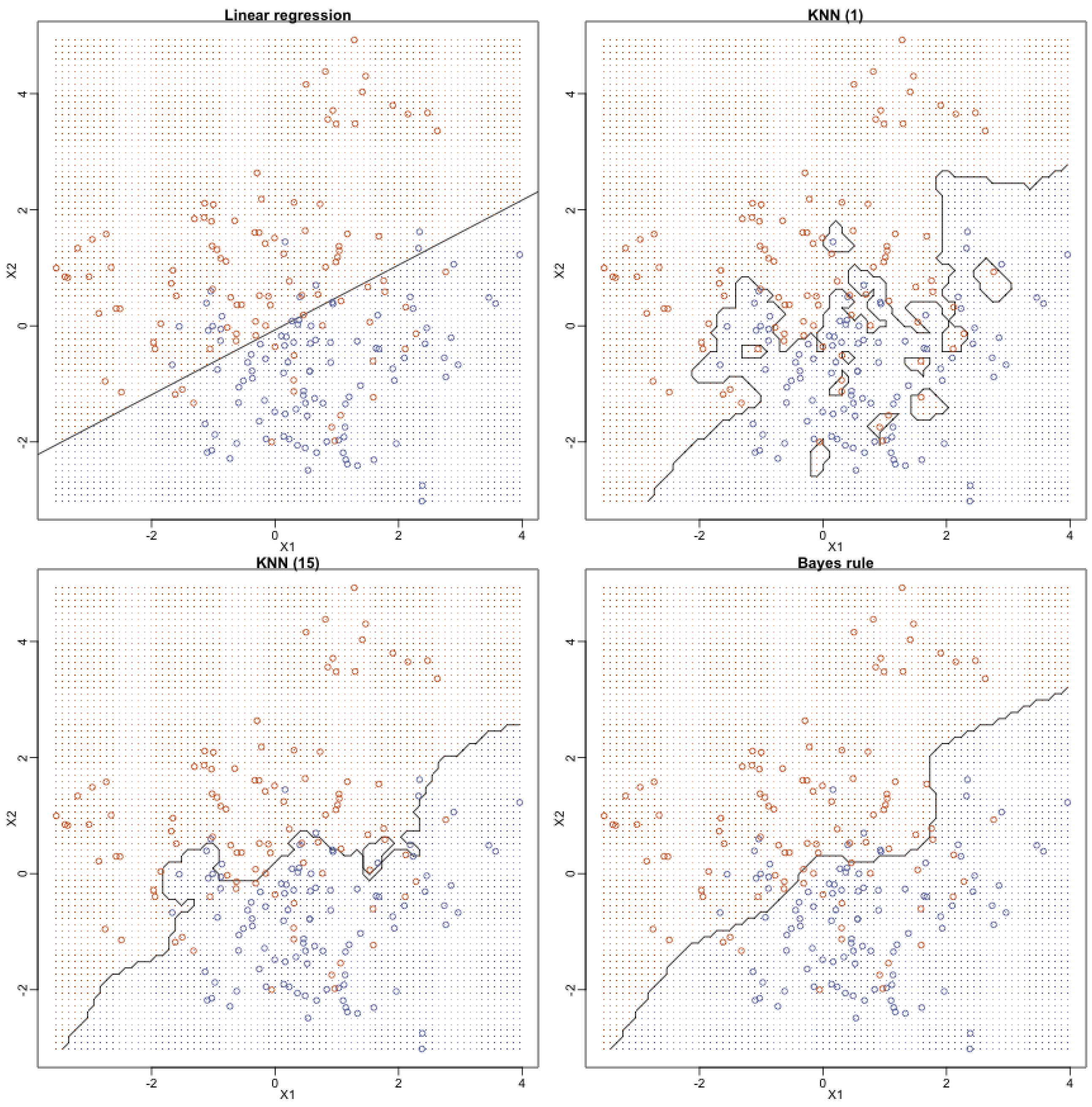


Figure 1: Decision boundaries

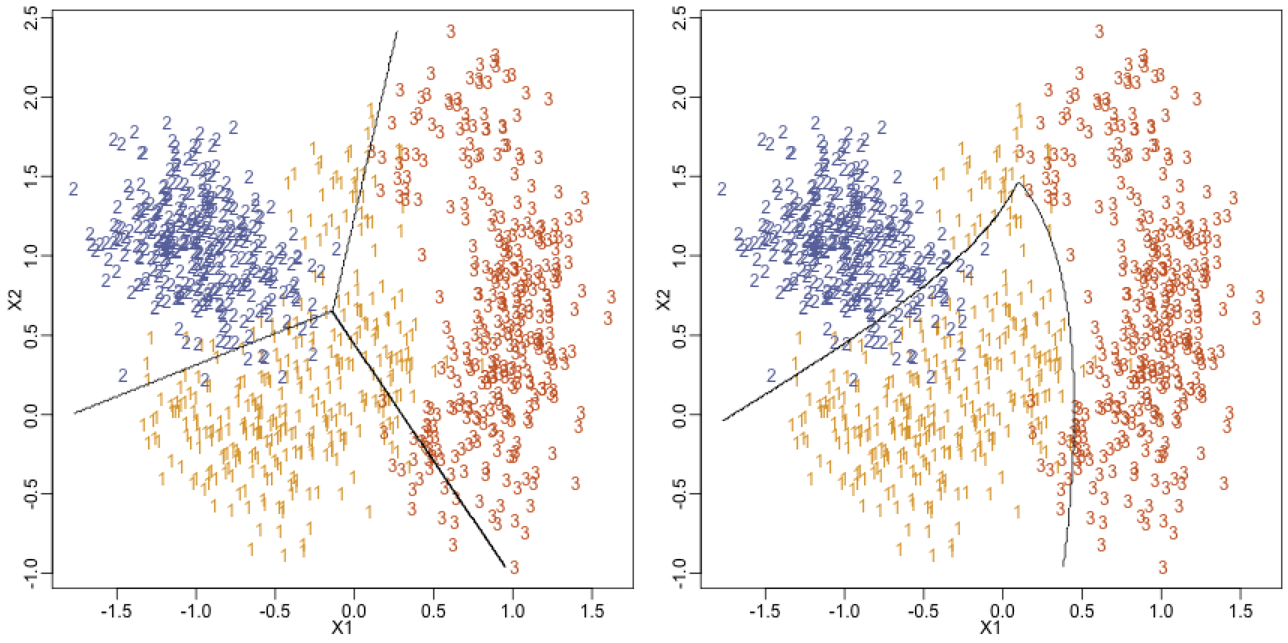


Figure 2: Linear decision boundaries

$$\hat{\mathbf{Y}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$

Notice the matrix

$$\hat{\mathbf{B}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$

has the coefficients for each regression in the columns. So, for any point $x$ we can get the prediction by

- compute the fitted output $\hat{f}(x) = [(1, x)\hat{\mathbf{B}}]$, a $K$ vector
- identify the largest component, and classify accordingly

$$\hat{G}(x) = \arg\max_{k \in G} \hat{f}_k(x)$$

What is the rationale for this approach?

Recall the *expected prediction error* we discussed last time. What do we do for qualitative data? Because the elements of $G$ are not numbers, taking expectations doesn't mean anything. However, we can define a *loss function*. Say we have three elements $G = \{A, B, C\}$, we can define the loss function like this:

$$L(\hat{G}, G) = \begin{cases} 0 & \text{if } \hat{G} = G \\ 1 & \text{if } \hat{G} \neq G \end{cases}$$

This can be thought of as a distance matrix with 0s in the diagonals and 1s everywhere else.

Notice that in some applications this loss function may not be appropriate. For instance, saying someone has cancer when they don't is not as bad as saying they don't have cancer when they actually do. For now, let's stick to 1 and 0.

We can now write

$$EPE = E_X \sum_{k=1}^{K} L(G_k, \hat{G}(X)) Pr(G = k|X)$$

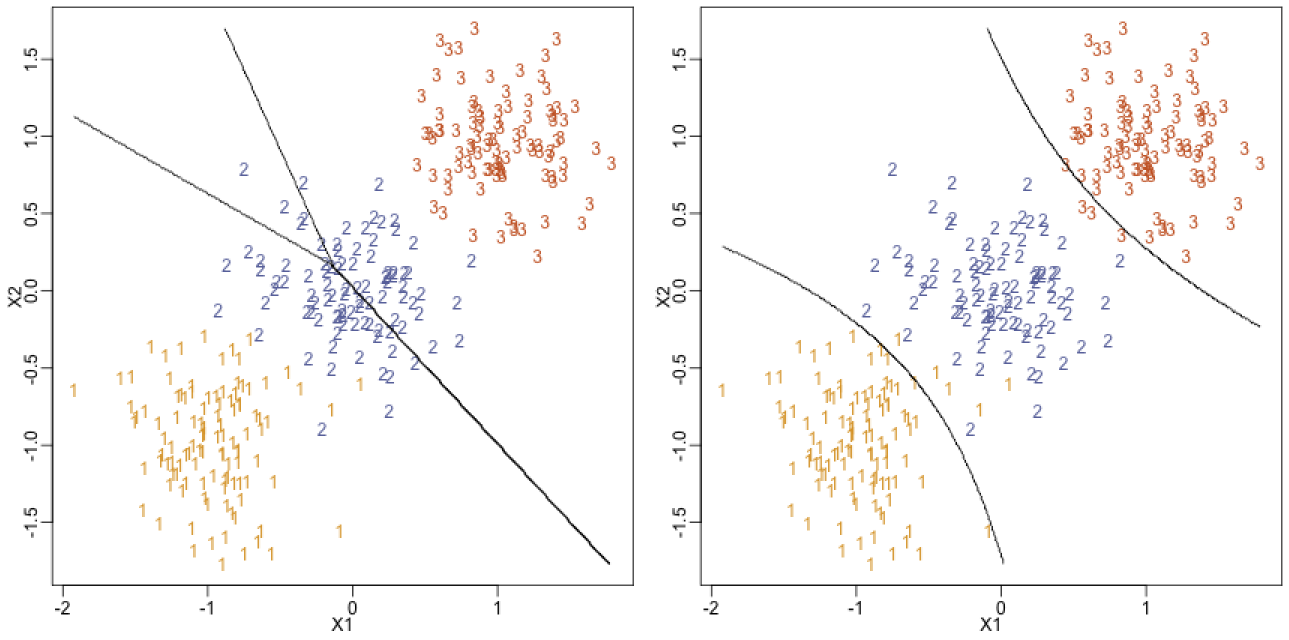


Figure 3: Linear regression problem

The minimizer of EPE is known as the *Bayes classifier*, or *Bayes decision rule*.

$$\hat{G}(X) = \max_{g \in G} Pr(g|X = x)$$

So, why don't we use it? Typically we don't know $Pr(g|X = x)$, just like in the regression setting we don't know $f(x) = E[Y|X = x]$.

Note: If we simulate data like we do below, then we can compute the Bayes decision rule since we know $Pr(g|X = x)$.

Note: If we use the encoding 0/1 encoding above for the two-class case, then we see the relationship between the Bayes classifier and the regression function since $Pr(G = g|X = x) = E(Y|X = x)$ in this case.

The real issue here is, how good is the linear approximation here? Alternatively, are the $\hat{f}_k(x)$ good estimates of $Pr(G = k|X = x)$.

We know they are not great since we know $\hat{f}(x)$ can be greater than 1 or less than 0. However, as we have discussed, this may not matter if we get good predictions.

A more conventional way of fitting this model is by defining target $t_k$ as the vector with a 1 in the $k$th entry and 0 otherwise, such that $y_i = t_k$ if $g_i = k$. Then the above approach is equivalent to minimizing

$$\min_{\mathbf{B}} \sum_{i=1}^{N} \|y_i - \{(1, x_i)\mathbf{B}\}'\|^2.$$

A new observation is classified by computing $\hat{f}(x)$ and choosing the closes target

$$\arg\min_{k} \|\hat{f}(x) - t_k\|^2$$

Because of the rigid nature of linear regression, a problem arises for linear regression with $K >= 3$. The next figure shows an extreme situation. Notice that decision boundaries can be formed by eye to perfectly discriminate the three classes. However, the decision boundaries from linear regression do not do well.

Why does linear regression miss the classification in this case? Notice that the best direction to separate these data is a line going through the centroids of the data. Notice there is no information in the projection orthogonal to this one.

If we then regress $Y$ on the transformed $X$, then there is barely any information about the second class. This is seen clearly in the left panel of the next Figure.

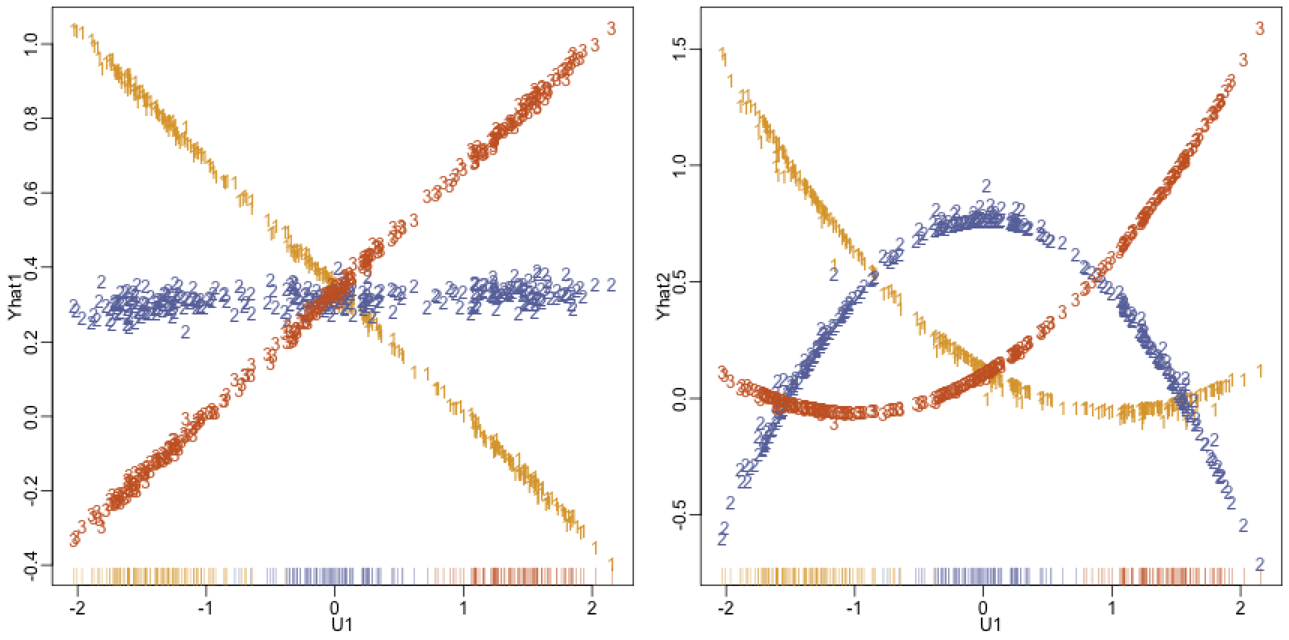


Figure 4: linear regression problem 2

However, by making the regression function a bit more flexible, we can do a bit better. One way to do this is to use quadratic terms (there are 3, what are they?) In the last two Figures, this linear regression version including quadratic terms does much better.

However, if we increase the number of classes to $K = 4$ we would then need to start adding the cubic terms and now are dealing with lots of variables.

## Linear Discriminant Analysis

Decision theory tells us that we should know the class posteriors $Pr(G|X = x)$ for optimal classification.

Suppose $f_k(x)$ is the class conditional density of $X$ in class $G = k$, and let $\pi_k$ be the prior probability of class $k$, with $\sum_{k=1}^K \pi_k = 1$. A simple application of Bayes' theorem gives us

$$Pr(G = k|X = x) = \frac{f_k(x)\pi_k}{\sum_{l=1}^K f_l(x)\pi_l}$$

Notice that havign quantities $f_k(x)$ is almost equivalent to having the $Pr(G = k|X = x)$ provided by Bayes' rule.

Suppose we model each class conditional density as multivariate Gaussian:

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma_k}^{-1}||^{1/2}} \exp\{-\frac{1}{2}(x - \mu_k)'\mathbf{\Sigma_k}^{-1}(x - \mu_k).$$

The next Figure shows regions that contain 95% of the data for three bivariate distributions with different means, but the same covariance structure. The covariance structure make these ellipses rather than circles.

The lines are the Bayes decision rules.

Linear Discriminant Analysis (LDA) arises when we assume that the covariance is the same for all classes. In this case, we see that discriminant functions are simply

$$\delta_k(x) = x'\mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k\mathbf{\Sigma}^{-1}\mu_k + \log \pi_k$$

Notice: if we assume $\pi_k = 1/K$ then the last term is not needed. In any case, notice this is a linear function of $x$!.

In practice, we do not have the means $\mu_k$ or the covariance structure $\mathbf{\Sigma}$. The strategy is to use training data to estimate these.

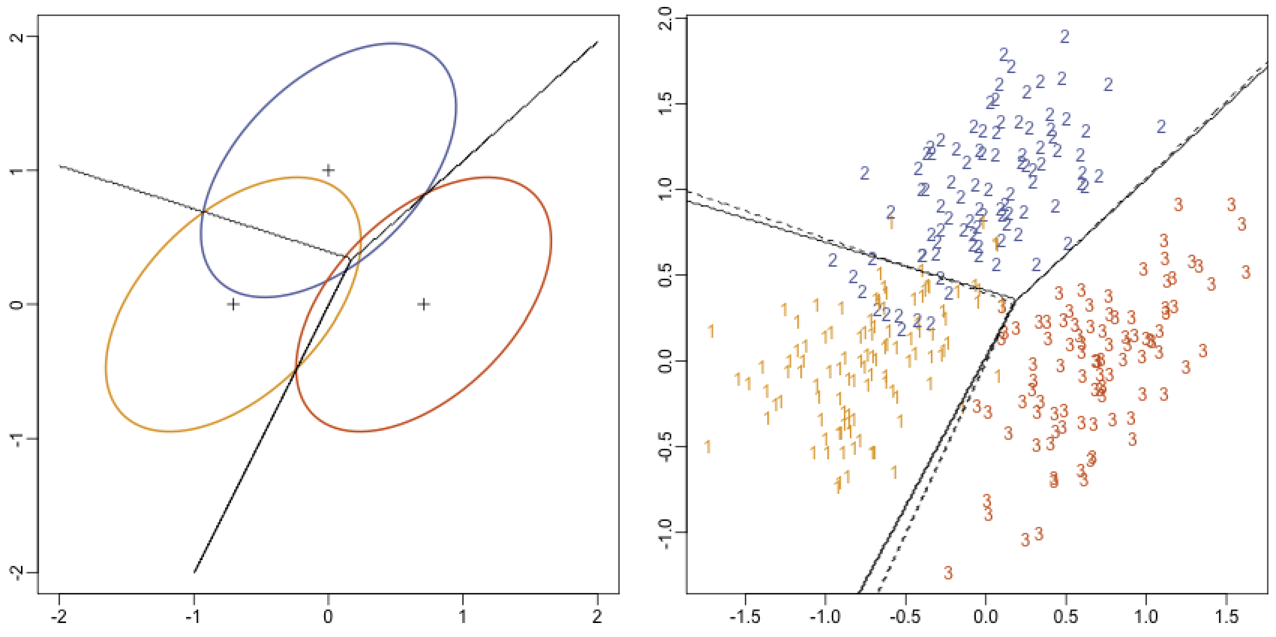


Figure 5: Lda illustration

In the next figure we see some outcomes in a three class simulation. The distributions used to create them are those shown on the left panel. The Bayes decision rules are shown dashed and the estimated LDA discriminant functions are shown as solid lines.

To estimate the parameters we simply:

- $\hat{\pi}_k = \frac{N_k}{N}$, where $N_k$ is the observed number of subjects in class $k$
- $\hat{\mu}_k = \frac{1}{N_k} \sum_{g_i = k} x_i$
- $\hat{\mathbf{\Sigma}} = \frac{1}{N-K} \sum_{k=1}^{K} \sum_{g_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)'$

Technical note: for two classes LDA is almost the same as regression, the coefficients of each are proportional, but unless $N_1 = N_2$ the intercepts are different.

If we assume that each class has its own correlation structure, the discriminant functions are no longer linear. Instead, we get

$$\delta_k(x) = -\frac{1}{2} \log |\mathbf{\Sigma}_k|^{-1} - \frac{1}{2}(x - \mu_k)' \mathbf{\Sigma}_k^{-1} (x - \mu_k)$$

The decision boundary is now described with a quadratic function. This is therefore called *quadratic discriminant analysis* (QDA). Next we plot LDA and QDA decision boundaries for the same data.

Note: when the number of covariates grow, the number of things to estimate in the covariance matrix gets very large. One needs to be careful.

Note: the LDA classifier is implemented in `R` by function `MASS::lda`.

## Computations for LDA and QDA

Suppose we compute the eigen-decomposition of each $\hat{\mathbf{\Sigma}}_k = \mathbf{U}_k \mathbf{D}_k \mathbf{U}_k'$, where $\mathbf{U}_k$ is a $p \times p$ matrix with orthonormal columns and $\mathbf{D}_k$ is a diagonal matrix of positive eigenvalues $d_{kl}$. The ingredients for $\delta_k(x)$ are:

- $(x - \hat{\mu}_k)' \mathbf{\Sigma}_k^{-1} (x - \hat{\mu}_k) = [\mathbf{U}_k'(x - \hat{\mu}_k)'] \mathbf{D}_k^{-1} [(x - \hat{\mu}_k) \mathbf{U}_k]$
- $\log |\mathbf{\Sigma}_k|^{-1} = \sum_l \log d_{kl}$

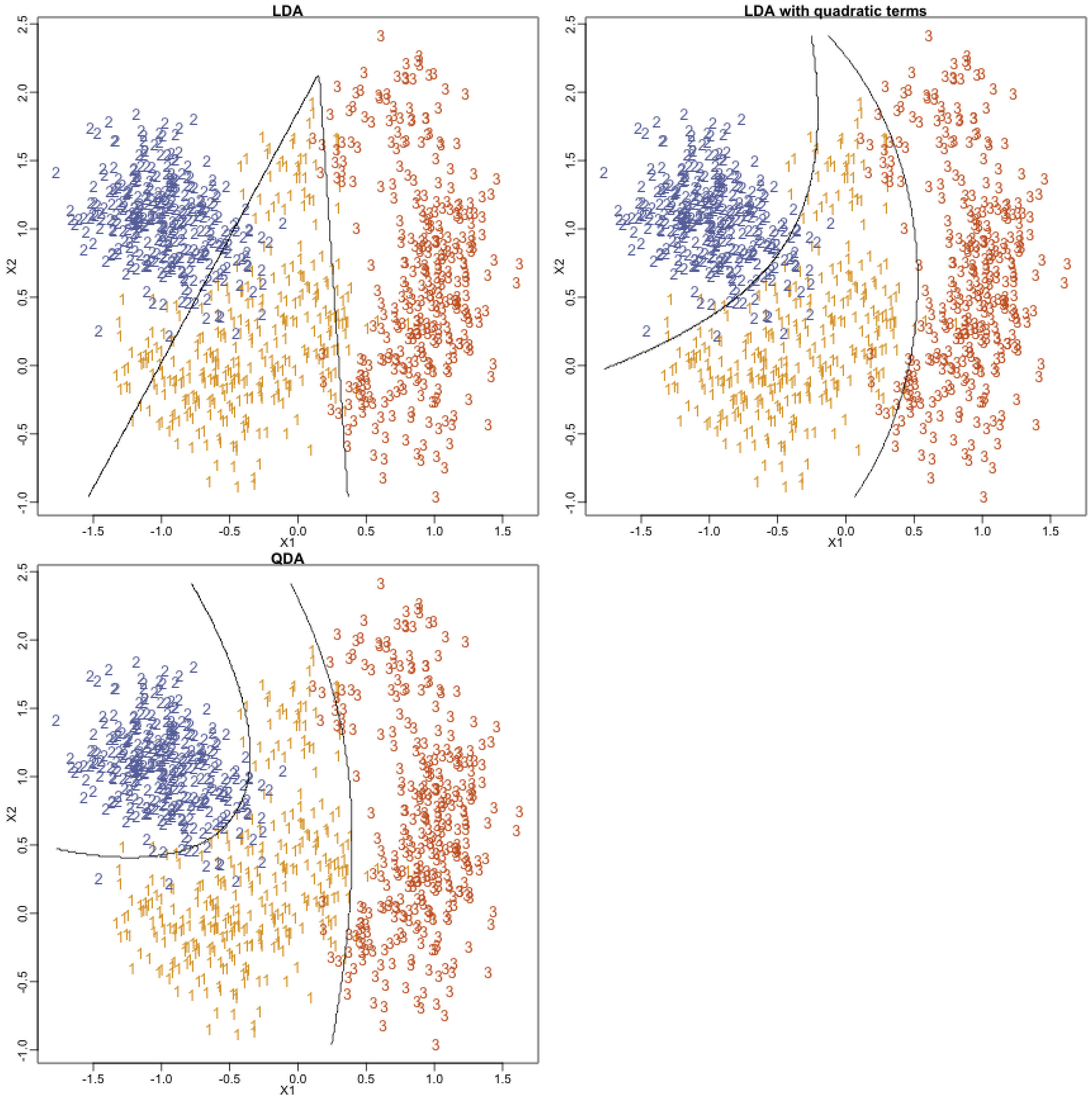


Figure 6: QDA illustration

Notice this is much easier to compute since $\mathbf{D}_k$ is a diagonal matrix!

Given this, we can now compute and interpret the LDA classifier as follows:

- *Sphere* the data with respect to the common covariance estimate $\hat{\mathbf{\Sigma}}$ to get $X^* = \mathbf{D}^{1/2}\mathbf{U}'X$. The common covariance estimate of $X^*$ is now the identity matrix!
- Classify to the closes class centroid ($\mu_k$s) in the transformed space, correcting for the effect of the class prior probabilities $\pi_k$.

Section 4.3.3 of Hastie, Tibshirani and Friedman, has a nice discussion of how LDA is related to the solution of the problem: *find the linear combination $Z = a'X$ such that the between-class variance is maximized relative to the within-class variance.*

## The curse of dimensionality

We have discussed the bias-variance tradeoff, and how flexible classifiers can, when model selection is properly applied, give us much better predictive performence. However, why would we ever consider a high-bias method such as linear regression? Is properly tuned KNN always better?

Consider the case where we have many covariates. We want to use local regression. These methods can be generalized to cases where we have more than one or two predictors. Basically, we need to define distance and look for small multi-dimensional "balls" around the target points. With many covariate this becomes difficult. To understand why we need some mathematical intuition. Let's try our best to see why.

Imagine we have equally spaced data and that each covariate is in $[0, 1]$. We want to something like kNN with a local focus that uses 10% of the data in the local fitting. If we have $p$ covariates and we are forming $p$-dimensional cubes, then each side of the cube must have size $l$ determined by $l \times l \times \cdots \times l = l^p = .10$. If the number of covariates is p=10, then $l = .1^{1/10} = .8$. So it really isn't local! If we reduce the percent of data we consider to 1%, $l = 0.63$. Still not very local. If we keep reducing the size of the neighborhoods we will end up with very small number of data points in each average and thus with very large variance. This is known as *the curse of dimensionality.*

## Diagonal LDA and shrunken centroids

In high-dimensional problems, we might not have enough data to compute all parameters of the covariance matrix needed for LDA. A nice, but powerful, form of regularization is to estimate the covariance matrix assuming that predictors are independent, i.e., the covariance matrix is *diagonal*. How can we interpret this in terms of principal components?

The *diagonal-covariance LDA* discriminant score function for class $k$ is

$$\delta_k(x^*) = -\sum_{j=1}^{p} \frac{(x_j^* - \bar{x}_{kj})^2}{s_j^2} + 2\log \pi_k.$$

Here $x^* = (x_1^*, x_2^*, \ldots, x_p^*)'$ is a vector corresponding to a new observation, $s_j$ is the pooled within-class estimate of the standard-deviation of predictor $j$, and $\bar{x}_{kj} = \sum_{g_i=k} x_{ij}/N_k$ is the mean of the $N_k$ values for predictor $j$ in class $k$. We call $\bar{x}_k = (\bar{x}_{k1}, \bar{x}_{k2}, \ldots, \bar{x}_{kp})'$ the *centroid* of class $k$. The discriminant score is then the standardized distance between observation $x^*$ and class centroid $\bar{x}_k$ adjusted by the class prior probability $\pi_k$.

The classification rule is then

$$\hat{G}(x^*) = \arg\max_{k=1,\ldots,K} \delta_k(x^*).$$

The diagonal LDA classifier is equivalent to *nearest centroid* classifier after appropriate standarization (and shift from the class priors).

To help with interpretation, we want to reduce the number of predictors used in the classifier. For instance, diagonal LDA uses all predictors when computing the standarized distance to each centroid. Preferrably, we want to remove predictors that do not contribute to the class predictions. Intuitively, these would be predictors for which the class means (or centroids) are close to each other, so that points are equi-distant to each of the class centroids.

One way of filtering predictors in this case is to use the two-sample $t$-statistic for each predictor:

$$t_j = \frac{\bar{x}_{2j} - \bar{x}_{1j}}{s_j}$$

The $t_j$-statistic gives a measure of how *significant* is the difference in the class means for predictor $j$. If the $t_j$ values were normally distributed, then one may consider values with $|t_j| > 2$ significant.

Similarily, consider the following statistic:

$$d_{kj} = \frac{\bar{x}_{kj} - \bar{x}_j}{m_k s_j},$$

with $m_k^2 = 1/N_k - 1/N$. Notice that with constant within-class variance $\sigma^2$, the variance of the contrast $\bar{x}_{kj} - \bar{x}_j$ is $m_k^2\sigma^2$. This gives a measure of how *significant* is the difference between the class $k$ mean for predictor $j$, and the overall mean for predictor $j$. In the two-class setting, how are the two statistics related?

The *nearest shrunken centroid* method uses a version of this statistic to *regularize* the nearest centroid predictor by *shrinking* the class means towards the overall mean for each predictor. Let

$$d_{kj} = \frac{\bar{x}_{kj} - \bar{x}_j}{m_k(s_j + s_0)},$$

with $s_0$ a small positive constant, typically the median of the $s_j$ values, used to guard against $d_{kj}$ values resulting from predictors with values near zero (this is common in gene expression data). Suppose you threshold these values:

$$d'_{kj} = d_{kj} \cdot I\{|d_{kj}| \geq \Delta\}.$$

Then $d'_{kj} = d_{kj}$ if the standarized class mean is "significantly" different from the overall mean, and zero otherwise. Now let's *shrink* the class means $\bar{x}_{kj}$ towards the overall mean as

$$\bar{x}_{kj} = \bar{x}_j + m_k(s_j + s_0)d'_{kj}.$$

What happens to class centroids with no *significant* difference with the overall centroid? Thus, unless a predictor has a *significant* difference to the overall mean for at least one class, it is useless for classification. How is this related to filtering predictors using the $t$-statistic?

Now, the thresholding we used is called *hard thresholding*, and it has some problems. Besides, selection, we want to use *shrinkage* to pool data when estimating $p$-dimensional centroids. Therefore, shrinking towards the overall mean (is estimated with all $N$ points) is a good idea. So to do both selection and to pool data to get estimates we can use *soft thresholding*.

$$d'_{kj} = \text{sign}(d_{kj})(|d_{kj}| - \Delta)_+,$$

where $(u)_+$ corresponds to the positive part of $u$:

$$(u)_+ = \begin{cases} u & \text{if } u > 0 \\ 0 & \text{o.w.} \end{cases}$$

The class centroids are shrunken towards the overall centroid as before. $\Delta$ is a parameter to be chosen (cross-validatated prediction error can be used). The hard-threshold and soft-threshold operations are illustrated in Figure 7.

The shrunken centroids method is implemented in `R` in package `pamr`.

### Relationship to Naive-Bayes

There is an elegant connection between a penalized Naive Bayes classifier and the shrunken centroids method (PAM). Diagonal LDA can be motivated using a slightly different approach as a Naive Bayes classifier, as pictured here:

In this case $g$ represents class, and $x_j$ is expression for gene $j$. The naive Bayes classifier assumes that $x_j|g = k \sim N(\mu x_{kj}, \sigma^2)$, that is, expression is independent and normally distributed in each class. Given a training set $\{(\mathbf{x}_1, g_1), \ldots, (\mathbf{x}_n, g_n)\}$, parameters for the naive Bayes model can be estimated by maximizing log-likelihood:

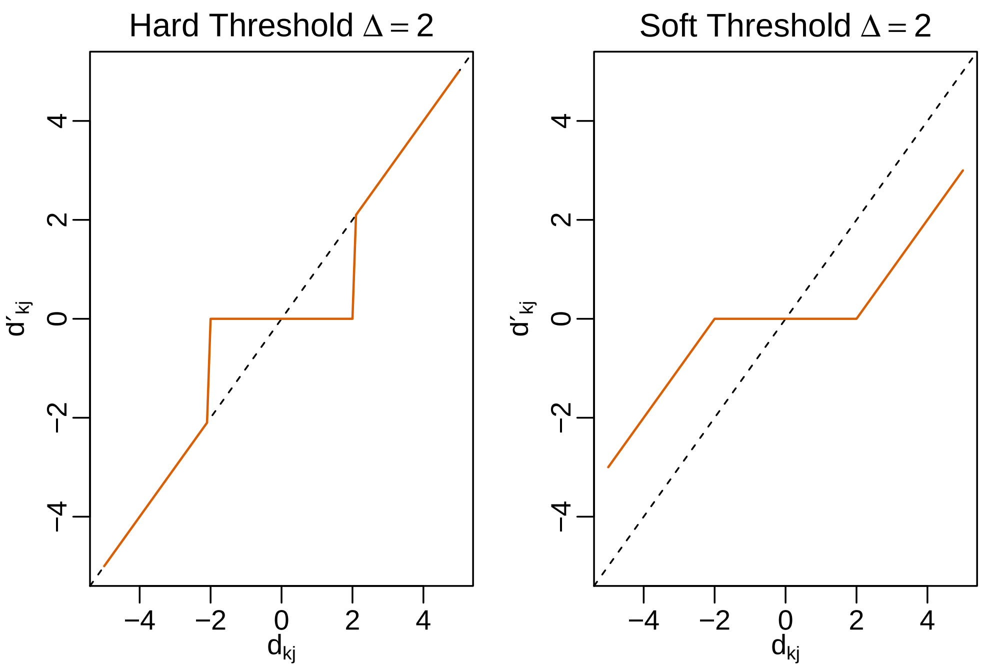
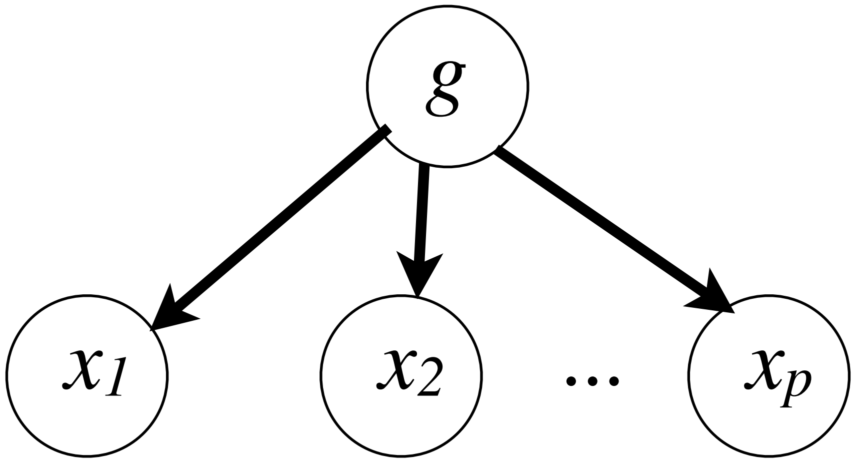


Figure 7: Thresholding operations



.

Figure 8: A naive Bayes classifier

$$\max \sum_{ji} log\{P(G = g_i|x_j)\}.$$

The estimates given above for diagonal LDA are exactly the maximizers of this optimization problem.

Now, suppose we re-parameterize the class means for each gene as $\mu x_{kj} = \mu_j + \mu_{kj}$ where $\mu x_j$ is the overall mean, and *regularize* the solution to the naive Bayes/diagonal LDA problem using an $l1$-penalty as follows:

$$\min_{\mu_j, \mu_{jk}} \left\{ \frac{1}{2} \sum_{j=1}^{p} \sum_{k=1}^{K} \sum_{i \in C_k} \frac{(x_{ij} - \mu_j - \mu_{jk})^2}{s_j^2} + \lambda \sum_{j=1} \sum_{k=1}^{K} \sqrt{N_k} \frac{|\mu_{jk}|}{s_j} \right\}.$$

You can show that the solution to this regularized optmization problem is the nearest-shrunken centroid. That is, PAM, or the nearest-shrunken centroid method, is the solution to an $l1$-regularized naiveBayes classifier, where deviation of class means from the overall mean is penalized.

## Quadratic regularization for linear classifiers

### Regularized Discriminant Analysis

Suppose we want to use LDA but not require that the variance is diagonal. If $p >> N$ then the non-diagonal $p$-by$p$ estimate $\hat{\mathbf{\Sigma}}$ is of rank $K$ and therefore $\hat{\mathbf{\Sigma}}^{-1}$ is undefined. That means we can't compute the discriminant direction $\hat{\mathbf{\Sigma}}^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$. The *regularization* approach in this case is to *shrink* $\hat{\mathbf{\Sigma}}$ towards its diagonal:

$$\hat{\mathbf{\Sigma}}(\gamma) = \gamma \hat{\mathbf{\Sigma}} + (1 - \gamma)\text{diag}(\hat{\mathbf{\Sigma}}),$$

with $\gamma$ a regularization parameter to be chosen. This can be combined with shrunken centroids, but distances are now "warped".