Classification

Data Analysis for Networks - DataNets'19

Anastasios Giovanidis

Sorbonne-LIP6

January 30, 2019

Bibliography

B.1 Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. "An introduction to statistical learning: with applications in R". Springer Texts in Statistics. ISBN 978-1-4614-7137-0
Chapter 2, Chapter 4
DOI 10.1007/978-1-4614-7138-7

Classification Setting

We have seen how to fit models to data when the response y_i to the input x_i is quantitative (e.g. "0.57", "24", "-24.3", etc.)

Question: How do we choose models and define their accuracy, when y_i's are qualitative?

Examples: ("Yes", "No"), ("Red", "Blue", "Green"), ("Malaria", "Yellow Fever", "Flu") or more generally:

☞ ("Class 1", "Class 2", . . . , "Class M")

Training Accuracy

Suppose we have training observations:
$D_n = \{(x_1, y_1),\ (x_2, y_2), \dots, (x_n, y_n)\}$, with $y_1,\ \dots,\ y_n$ qualitative.

Consider a fitting model with an estimate $\hat{y}_i = \hat{f}(x_i)$.
We use the training error rate:

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left(y_i \neq \hat{y}_i\right).$$

This is the fraction of incorrect classifications:

- $\hat{y}_i$ is the predicted class label for the i-th observation using $\hat{f}$.
- $\mathbf{1}\left(y_i \neq \hat{y}_i\right) = 0$ for correct classification, else 1.
- Similar to MSE_{train} in regression!

Test Accuracy

Most interested in the error rates of the classifier to test observations $(x_o, y_o) \notin D_n$, not used in training.

Again for an estimate $\hat{y}_o = \hat{f}(x_o)$ we use the test error rate:

$$Ave\left(\mathbf{1}\left(y_o \neq \hat{y}_o\right)\right).$$

☞ A good classifier is the one for which the test error is smallest !

Bayes Classifier

Optimal Classifier: Assign each observation to the most likely class, given its predictor values:

$$\max_{1 \leq j \leq M} \quad Pr\left(Y = j \mid X = x_o\right)$$

- We consider *conditional probabilities* given the observed x_o.

☞ In a two-class problem

$$Pr\left(Y = 1 \mid X = x_o\right) + Pr\left(Y = 2 \mid X = x_o\right) = 1:$$

$$\begin{array}{ll} \text{Class 1,} & \text{if } Pr\left(Y = 1 \mid X = x_o\right) > 0.5 \\ \text{Class 2,} & \text{if } Pr\left(Y = 2 \mid X = x_o\right) > 0.5 \end{array}$$

☞ Decision boundary $Pr\left(Y = 1 \mid X = x_o\right) = Pr\left(Y = 2 \mid X = x_o\right)$

Bayes example

orange region: $Pr(Y = \text{"}orange\text{"} \mid X) > 0.5$

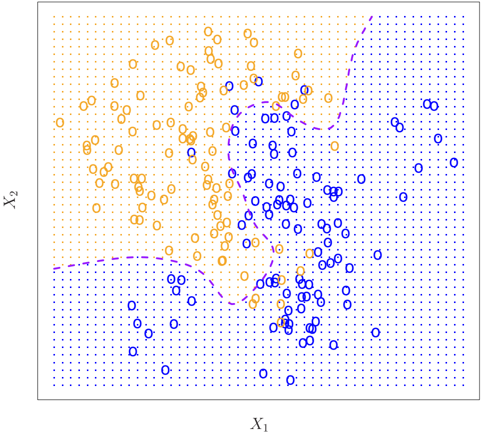

Figure: Bayes classifier : D_{100} data-set and 2 classes (blue, orange). [1]

[1] Source [B.1]

Bayes classifier cont'd

- Orange shaded region: $Pr(Y = \text{"}orange\text{"} \mid X) > 0.5$.
- Blue shaded region: $Pr(Y = \text{"}blue\text{"} \mid X) > 0.5$.
- The dashed line: Bayes decision boundary.
- Circles that fall in regions with different colour: misclassifications

☞ Bayes classifier produces lowest test error rate (irreducible) !

$$Test\ Error(x_o) = 1 - \max_j Pr\left(Y = j \mid X = x_o\right)$$

Drawback...

There is one problem however: For real data we do not know the conditional distribution $P(Y|X)$,

(unless we have generated data ourselves, in which case we know the joint distribution $P(X, Y)$).

Bayes classifier serves as an unreachable gold standard!

If we do not know exactly $P(Y|X)$ we can try to estimate it.

Classifiers

We will consider in this lecture the following classifiers:

- K-Nearest-Neighbours classifier (KNN)
- Logistic Regression (LR)
- Linear Discriminant Analysis (LDA)
- Quadratic Discriminant Analysis (QDA)

KNN classifier

So, how does the KNN classifier works?

- Choose a positive integer K.
- Given a test observation $x_o \notin D_n$, the KNN classifier identifies the K points in the training data closest to x_o, the set $\mathcal{N}_K(x_o)$.
- The conditional probability for class j at x_o is estimated as:

$$Pr\left(Y = j \mid X = x_o\right) = \frac{1}{K} \sum_{i \in \mathcal{N}_K(x_o)} \mathbf{1}\left(y_i = j\right). \qquad (1)$$

- Calculate the estimates for all classes $j = 1, \ldots, M$ and
- Finally, Apply Bayes: classify x_o to the class with the largest estimated probability.

KNN example

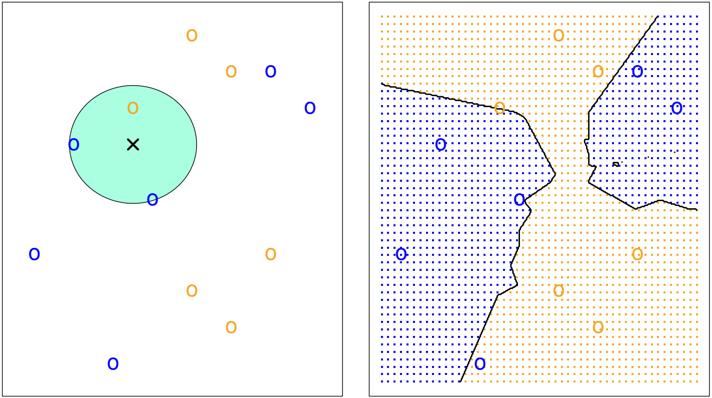

Figure: KNN classifier ($K = 3$) : D_{12} data-set and 2 classes. [2]

[2] Source [B.1]

Optimal Choice of K

Despite its simplicity KNN can give classifiers surprising close to Bayes. Choice of K is important:

- If $K = 1$, very flexible decision boundary $\rightarrow$
  Low Training Error ($= 0$) but! High Test Error.
- As K increases (less flexibility)
  Training Error increases but the Test Error may not !
- Find optimal K^* with minimum Test Error ($\bigcup$ shape)
- If $K = 100$ decision boundary close to linear.

Variance vs Bias Tradeoff

or

Flexibility vs Interpretability

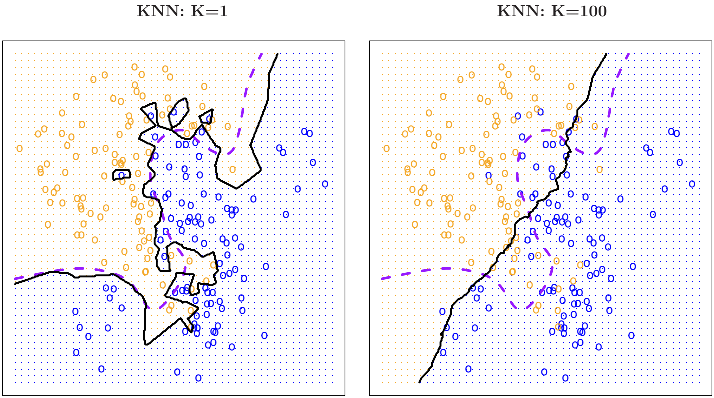

Figure: KNN with $K = 1$ (left) and $K = 100$ (right). [3]

[3] Source [B.1]

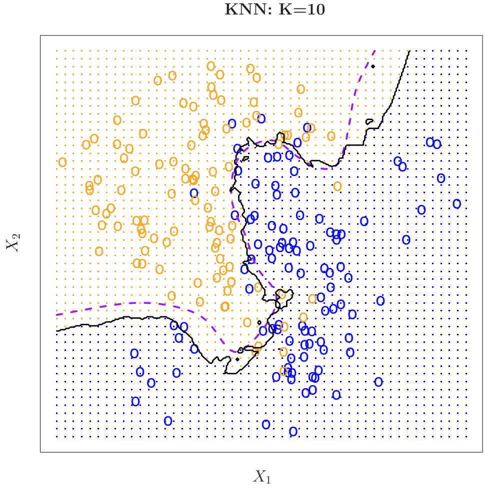

Figure: KNN with $K = 10$ close to Bayes optimal. [4]

[4] Source [B.1]

Variance vs Bias Tradeoff

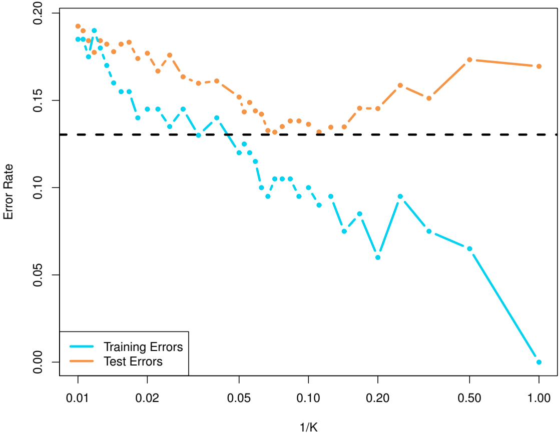

Figure: Training/Test Error Rate. [5]

[5] Source [B.1]

What if... Linear Regression?

Suppose we have again two classes: 'Class 1', 'Class 2'.

- What if we used Linear Regression for the $P(Y|X)$?
- Let 'Class 1': $Y = 0$ and 'Class 2': $Y = 1$.
- We assume that the linear model describes the 0/1 data,

$$y_i = \beta_0 + \beta_1 x_i + \epsilon$$

and we look for the regression line

$$\mathbb{E}[y_i|x_i] = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

☞ Since $y_i \in \{0, 1\}$ then $\mathbb{E}[y_i|x_i] = Pr(y_i = 1|x_i) = \hat{\beta}_0 + \hat{\beta}_1 x_i$.

Wrong Shape ! less than 0, more than 1

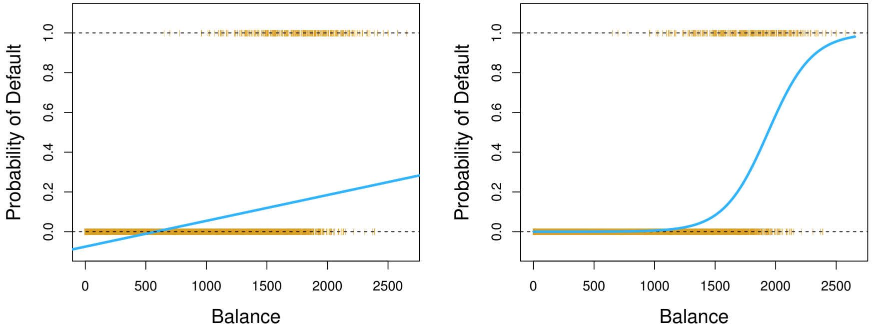

Figure: $Pr(Y = 1|X)$. Linear vs Sigmoidal fit. [6]

[6] Source [B.1]

Logistic Regression

Suppose for the two-class problem $Pr\,(Y = 1|X)$ follows the logistic function.

$$p(X) := Pr\,(Y = 1|X) = \frac{e^{\beta_o + \beta_1 X}}{1 + e^{\beta_o + \beta_1 X}} \tag{2}$$

- For $X \to -\infty$: $p(X) \to 0$
- For $X \to +\infty$: $p(X) \to 1$
- It is an S-shaped curve.

☞ We need to fit $\beta_o,\ \beta_1$ in the non-linear logistic function.

Logistic fit

We consider a Training data-set D_n with $Y_n = (0, 0, 1, \ldots, 0, 1)$.

- We don't want to use *MSE* fit $\rightarrow$ complicated expressions.
- Better use: log-likelihood function.

What is the likelihood $g(D_n)$ of the data-sample?

$$g(D_n) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'}))$$

because we assumed that for any X

$$Y = \begin{cases} 1, & p(X) \\ 0, & 1 - p(X) \end{cases}$$

and for all $x_i \in D_n$ we know what is the y_i answer.

Log-likelihood maximization

The log-likelihood function, is then equal to

$$
\begin{aligned}
\ell(\beta_0, \beta_1; D_n) &= \log(g(D_n)) \quad (3)\\
&= \sum_{i:y_i=1} \log p(x_i) + \sum_{i':y_{i'}=0} \log\left(1 - p(x_{i'})\right)\\
&= \sum_{i=1}^{n} \left\{y_i \log p(x_i) + (1 - y_i)\log\left(1 - p(x_i)\right)\right\}\\
&\overset{\frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}}{=} \sum_{i=1}^{n} \left\{y_i\left(\beta_0 + \beta_1 x_i\right) - \log\left(1 + e^{\beta_0+\beta_1 X}\right)\right\}
\end{aligned}
$$

☞ We want to $\max_{\beta_0,\beta_1} \ell(\beta_0, \beta_1;\ D_n)$.

Newton's algorithm

We follow standard process:

- $\nabla \ell(\beta_0, \beta_1; D_n) = \begin{bmatrix} \frac{\partial \ell}{\partial \beta_o} \\ \frac{\partial \ell}{\partial \beta_1} \end{bmatrix}$
- $\nabla^2 \ell(\beta_0, \beta_1; D_n) = \begin{bmatrix} \frac{\partial^2 \ell}{\partial {\beta_0}^2} & \frac{\partial^2 \ell}{\partial \beta_o \partial \beta_1} \\ \frac{\partial^2 \ell}{\partial \beta_1 \partial \beta_0} & \frac{\partial^2 \ell}{\partial {\beta_1}^2} \end{bmatrix} < 0$ negative-definite
- Hence the log-likelihood logistic function is strictly concave.

$$\begin{bmatrix} \beta_0^{(k+1)} \\ \beta_1^{(k+1)} \end{bmatrix} = \begin{bmatrix} \beta_0^{(k)} \\ \beta_1^{(k)} \end{bmatrix} - \left(\nabla^2 \ell(\beta_0, \beta_1; D_n)\right)^{-1} \cdot \nabla \ell(\beta_0, \beta_1; D_n)$$

"What are the odds?"

One can see the logistic expression of the predictions from a different point-of-view:

$$q(x_i) := \frac{p(x_i)}{1 - p(x_i)} \quad = \quad e^{(\beta_0 + \beta_1 x_i)}.$$

☞ odds function: often used in... Horse-racing!

"What are the odds ?"

- If $q(x_i) = 1/4$, then $p(x_i = 1) = 0.2$
- If $q(x_i) = 9/1$, then $p(x_i = 1) = 0.9$.

The logits (or log-odds)

One can see the logistic expression from a different point-of-view:

$$Q(x_i) := \log\left(\frac{p(x_i)}{1-p(x_i)}\right) \quad = \quad \beta_0 + \beta_1 x_i.$$

Here we come back to the expression for the Linear Regression!

Separating hyperplane: For $p = 0.5$, we get the "linear" boundary

$$0 = \beta_0 + \beta_1 x_{i,1} \quad (+\beta_2 x_{i,2} + \ldots + \beta_K x_{i,K}), \qquad \textit{for } K \geq 1.$$

e.g. for $K = 1$, it is a point $x_{bound} = -\beta_0/\beta_1$. (left: 1, right: 0)

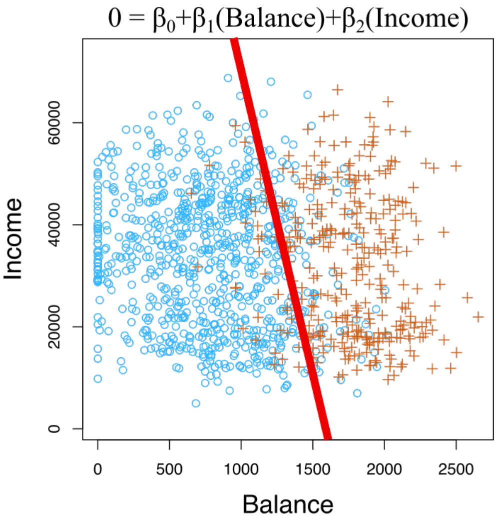

Figure: The boundary separates "blue" from "orange". [7]

[7] Source [B.1]

Test Data (Logistic)

If we have test input data $x_o \notin D_n$, how do we choose its Class?
Say $x_o = (x_{o,1}, x_{o,2}, \ldots, x_{o,K})$.

Use the fitted values of $\beta_0, \beta_1, \ldots, \beta_K$

- Either calculate $p(x_o) = \frac{e^{\beta_0+\beta_1 x_o}}{1+e^{\beta_0+\beta_1 x_o}}$ and check if $>, =, < 0.5$,
- or check the position of x_o related to the boundary:
  $\beta_0 + \beta_1 x_{o,1} + \beta_2 x_{o,2} + \ldots + \beta_K x_{o,K} >, =, < 0$.

e.g. $\beta_0 + \beta_1 x_{o,1} + \beta_2 x_{o,2} + \ldots + \beta_K x_{o,K} > 0 \Rightarrow p(x_o) > 0.5$

☞ We need not always use the value of 0.5 for the boundary...

Multiple Logistic Regression

We have implied that the Logistic Regression is generalised to higher than 1 dimension:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \ldots + \beta_K X_K,$$

where $X = (X_1, \ldots, X_K)$ are K predictors.

Equivalently,

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\beta_K X_K}}{1+e^{\beta_0+\beta_1 X_1+\beta_K X_K}}.$$

☞ $\beta_0, \ldots, \beta_K$ are estimated by the maximum likelihood method.

Example

Using the data set Default we want to decide, whether an individual is likely to default on its bank account.

$X = (\text{balance}, \text{ income}, \text{ student[Yes]})$, so $K = 3$.
$Y = \text{default[Yes]}$.

- First consider only balance, $K = 1$.

	Coefficient	Std. error	Z-statistic	P-value
`Intercept`	-10.6513	0.3612	-29.5	<0.0001
`balance`	0.0055	0.0002	24.9	<0.0001

☞ 1-unit increase in balance is associated to $\beta_1 = 0.0055$ units increase in log-odds of default.

Example (predictions)

default[Yes] probability for an individual with $\text{balance} = 1000$ EUR

$$\hat{p}(\text{balance} = 1000) \quad = \quad \frac{e^{-10.6513+0.0055x1000}}{1 + e^{-10.6513+0.0055x1000}} = 0.00576$$

- Now consider binary student[Yes], $K = 1$.

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−3.5041	0.0707	−49.55	<0.0001
student[Yes]	0.4049	0.1150	3.52	0.0004

$$\hat{p}(\text{student[Yes]} = 1) = 0.0431 \quad > \quad \hat{p}(\text{student[Yes]} = 0) = 0.0292$$

Conclusion 1: Students are more likely to default.

Example (multiple)

- Now consider the entire X vector, $K = 3$.

	Coefficient	Std. error	Z-statistic	P-value
`Intercept`	-10.8690	0.4923	-22.08	<0.0001
`balance`	0.0057	0.0002	24.74	<0.0001
`income`	0.0030	0.0082	0.37	0.7115
`student[Yes]`	-0.6468	0.2362	-2.74	0.0062

Paradox: Conclusion 2: Students are less likely to default !!!! ($\beta_{\text{student[Yes]}} < 0$)

Why? The student[Yes] and balance predictors are correlated.

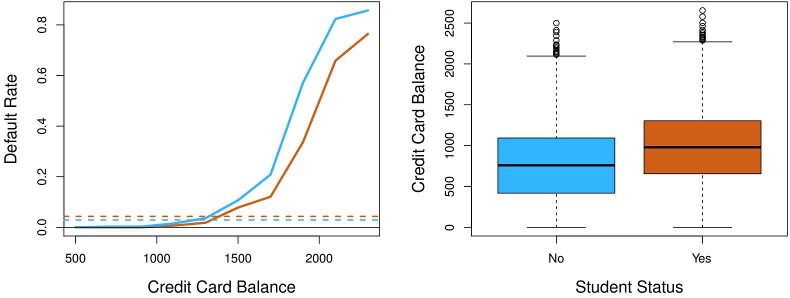

Figure: Students tend to have higher debts in the US/GB/D. [8]

Conclusion 1: For the same credit-card $\mathrm{balance}$ a student is less likely to default.

[8] Source [B.1]

Logistic Regression for > 2 Classes

We can easily generalise to M classes:

$$
\begin{aligned}
\log \frac{Pr(Class = 1|X = x)}{Pr(Class = M|X = x)} &= \beta_{1,0} + \beta_1^T x \\
&\dots \\
\log \frac{Pr(Class = M-1|X = x)}{Pr(Class = M|X = x)} &= \beta_{M-1,0} + \beta_{M-1}^T x \\
Pr(Class = M|X = x) &= \frac{1}{1 + \sum_{m=1}^{M-1} \exp(\beta_{m,0} + \beta_m^T x)}
\end{aligned}
$$

- We need $M - 1$ log-odds. • The probabilities sum-up to 1.
- The choice of denominator class is arbitrary. • Max likelihood.

☞ For multiple classe, discriminant analysis is more popular...

Linear Discriminant Analysis (LDA)

For classification of two or multiple classes, we often use the LDA classifier:

- Again, the class boundaries are linear.
- Instead of modelling $Pr\,(Y = k|X = x)$ directly as in LR, it does this indirectly by modelling $Pr\,(X = x|Y = k)$.
- It makes use of the Bayes' Theorem and the Bayes classifier.
- It assumes that the distribution of X's is approximately Normal, (or Gaussian).

Bayes' Theorem in Classification

We want to calculate the conditional probability for each class

$$\begin{aligned} Pr\left(Y=k|X=x\right) &\overset{Bayes'}{=} \frac{Pr\left(X=x|Y=k\right)Pr\left(Y=k\right)}{Pr\left(X=x\right)} \\ &\overset{Total}{=} \frac{Pr\left(X=x|Y=k\right)Pr\left(Y=k\right)}{\sum_{m=1}^{M}Pr\left(X=x|Y=m\right)Pr\left(Y=m\right)} \\ &= \frac{f_k(x)\cdot\pi_k}{\sum_{m=1}^{M}f_m(x)\cdot\pi_m} \end{aligned} \tag{4}$$

☞ We need the conditional probability of X given the class, and the frequency of each class.

☞ Given these, we can choose for $X=x_o$, the class with $\max_{1\le j\le M} Pr\left(Y=j|X=x_o\right)$ (Bayes classifier).

LDA for 1 predictor $K = 1$

We can **assume** that $f_k(x)$ is normal or Gaussian.

- For $K = 1$:

$$f_k(x) = \frac{1}{\sigma_k\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right),$$

  μ_k and σ_k^2 are the mean and variance for the k-th class.
- Let us further assume that $\sigma_1^2 = \sigma_2^2 = \ldots = \sigma_M^2 = \sigma^2$, hence there is a shared variance among all classes.
- The π_m's are also called prior probabilities.

Q: Is the gaussian assumption reasonable?

LDA ($K = 1$)

Plugging in (4), we get:

$$Pr\left(Y = k | X = x\right) \quad = \quad \frac{\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_k)^2\right) \cdot \pi_k}{\sum_{m=1}^{M} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_m)^2\right) \cdot \pi_m}$$

Unknowns: π_m, μ_m, $\forall m$, and σ.

LDA ($K = 1$) classification

We take the log in the above expression. We then assign for $X = x$, the class m^* such that

$$
\begin{aligned}
m^* &= \arg \max_{1 \le m \le M} Pr\left(Y = m | X = x\right) \\
&= \arg \max_{1 \le m \le M} \log Pr\left(Y = m | X = x\right) \\
&= \arg \max_{1 \le m \le M} \left\{ x \cdot \frac{\mu_m}{\sigma^2} - \frac{\mu_m^2}{2\sigma^2} + \log(\pi_m) \right\} \qquad (5) \\
&= \arg \max_{1 \le m \le M} \{ x \cdot c_1 + c_0 \} \quad (\mathit{linear!})
\end{aligned}
$$

Estimating the decision function

For each m we have the linear discriminant function function of x:

$$\delta_m(x) \quad = \quad x \cdot \frac{\mu_m}{\sigma^2} - \frac{\mu_m^2}{2\sigma^2} + \log(\pi_m),$$

and to calculate it from the dataset D_n we use the estimates:

$$\begin{aligned} \hat{\mu}_m &= \frac{1}{n_m} \sum_{i:y_i=m} x_i, \\ \hat{\sigma}^2 &= \frac{1}{n-M} \sum_{m=1}^{M} \sum_{i:y_i=m} (x_i - \hat{\mu}_m)^2, \\ \hat{\pi}_m &= \frac{n_m}{n}. \end{aligned}$$

2-class example

In the case of $M = 2$ classes, suppose $\pi_1 = \pi_2$ additionally. Then the discriminant functions become:

$$\begin{aligned}\delta_1(x) &= x \cdot \frac{\mu_1}{\sigma^2} - \frac{\mu_1^2}{2\sigma^2} + \log(\pi_1) \\ \delta_2(x) &= x \cdot \frac{\mu_2}{\sigma^2} - \frac{\mu_2^2}{2\sigma^2} + \log(\pi_2)\end{aligned}$$

so that x is assigned class 1, if $\delta_1(x) > \delta_2(x)$ or,

$$2x(\mu_1 - \mu_2) > \mu_1^2 - \mu_2^2$$

The decision boundary are the points x, s.t.

$$x = \frac{\mu_1 + \mu_2}{2}.$$

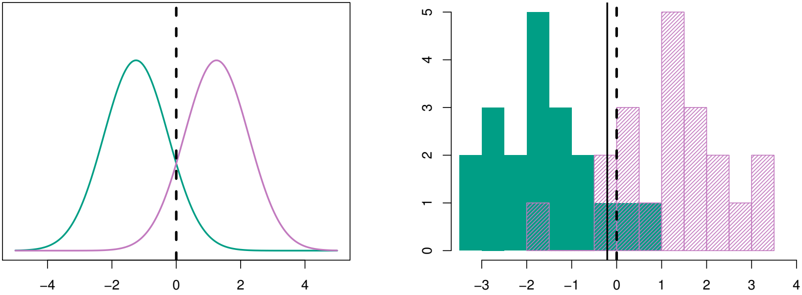

Figure: Two normal density functions and decision boundary. [9]

[9] Source [B.1]

LDA for $K > 1$ dimensions

How does the LDA perform, when the predictors X have more than 1 dimension? say $X = (X_1, \ldots, X_K)$.

☞ Assume a multivariate Gaussian distribution instead of a 1-dimensional $X \sim \mathcal{N}(\mu, \mathbf{\Sigma})$.

$$f(x) \quad = \quad \frac{1}{(2\pi)^{K/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \mathbf{\Sigma}^{-1}(x-\mu)\right).$$

• mean $\mu = (\mu_1, \ldots, \mu_K)$, • common covariance matrix $\mathbf{\Sigma}$.

Linear Discriminant Function:

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

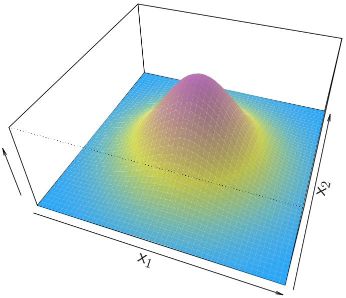

Figure: Examples of binormal distributions. [10]

[10] Source [B.1]

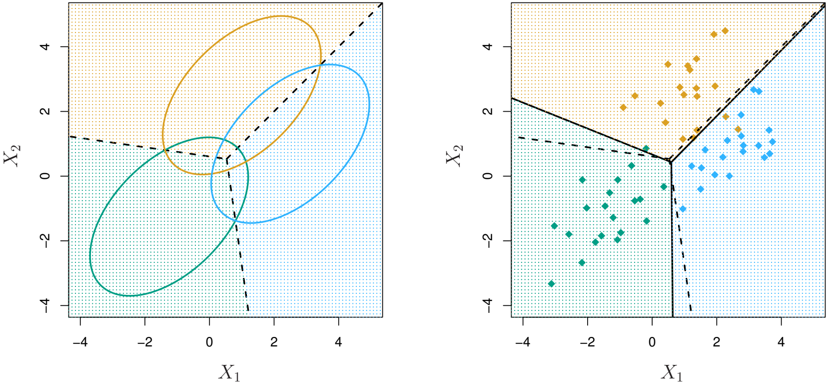

Figure: Classification for $M = 3$ classes and $K = 2$ dimensions. [11]

[11] Source [B.1]

		True default status		
		No	Yes	Total
Predicted default status	No	9,644	252	9,896
	Yes	23	81	104
	Total	9,667	333	10,000

Figure: Confusion Matrix: Predicted vs True default status. [12]

$$\textit{Error}\left[\widehat{\text{Default}} = \text{“}\textit{Yes}\text{”}\,|\text{Default} = \text{“}\textit{No}\text{”}\right] = 23/9667 \approx 0.2\%$$

$$\textit{Error}\left[\widehat{\text{Default}} = \text{“}\textit{No}\text{”}\,|\text{Default} = \text{“}\textit{Yes}\text{”}\right] = 252/333 \approx 75.7\%$$

[12] Source [B.1]

Quadratic Discriminant Analysis (QDA)

LDA assumed for each class a different mean μ_k and same covariance matrix $\mathbf{\Sigma}$.

☞ QDA assumes different covariance matrix per class. That is, an observation from the k-th class is of the form $X \sim \mathcal{N}(\mu_k, \mathbf{\Sigma}_k)$.

Quadratic Discriminant Function:

$$\begin{aligned} \delta_k(x) \quad = \quad & -\frac{1}{2} x^T \mathbf{\Sigma}_k^{-1} x + x^T \mathbf{\Sigma}_k^{-1} \mu_k - \frac{1}{2} \mu_k^T \mathbf{\Sigma}_k^{-1} \mu_k - \\ & -\frac{1}{2} \log |\mathbf{\Sigma}_k| + \log(\pi_k) \end{aligned}$$

QDA is more flexible than LDA: Bias vs Variance tradeoff !

QDA examples

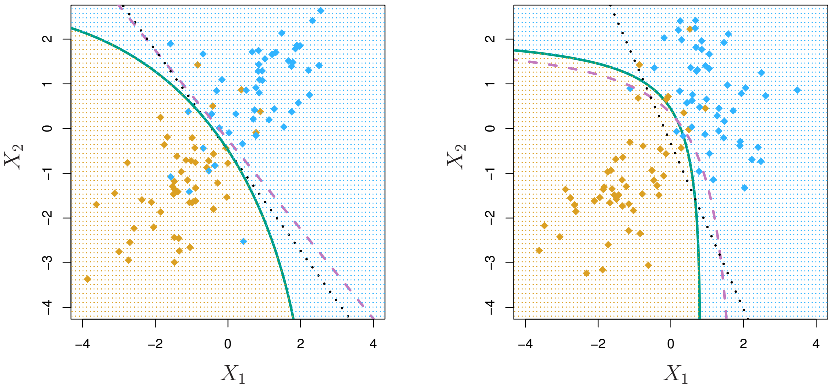

Figure: (left:) Truth common $\mathbf{\Sigma}$, (right:) Truth different $\mathbf{\Sigma}_1$, $\mathbf{\Sigma}_2$. [13]

[13] Source [B.1]

Method comparison: linear

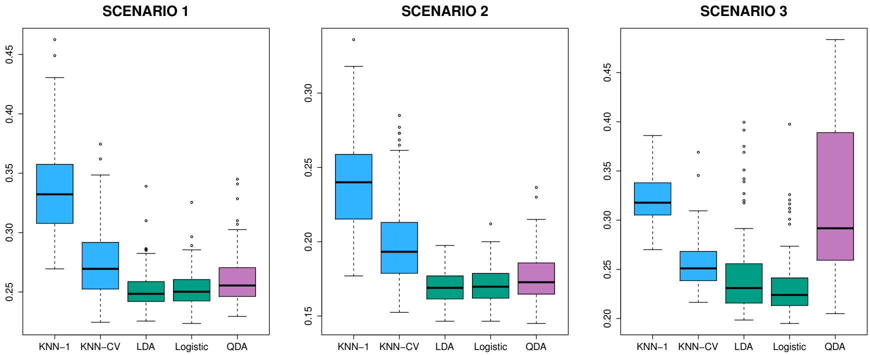

Figure: (1) uncorr., $\mathcal{N}$, $\mu_1 \neq \mu_2$, (2) corr., $\mathcal{N}$, (3) uncorr., t-distr.[14]

[14] Source [B.1]

Method comparison: non-linear

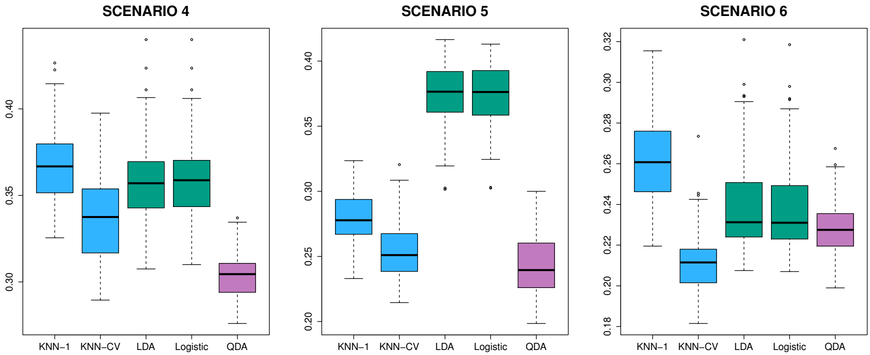

Figure: (4) corr. $\mathcal{N}$, $\boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$, (5) logistic $X_1^2, X_2^2, X_1 X_2$ (6) more-NL. [15]

[15] Source [B.1]

END