Lecture 5

Classification II

ISLR 4, ESL 4

Jilles Vreeken
Aleksandar Bojchevski

UNIVERSITÄT DES SAARLANDES
CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Bayesian Classification

Bayesian Methods

- Bayes' formula

Probability of the input, given the output, i.e. class density

Posterior (probability of the output given the input) $\Pr(Y \mid X) = \dfrac{\Pr(X \mid Y)\Pr(Y)}{\Pr(X)}$ Prior probability of the output; Prior probability of the input

- $\Pr(X)$ is a normalizing constant that only depends on the input data and often need not be computed

Bayesian classification for K classes

- use Bayes' formula to determine posterior density per class $\Pr(Y = k \mid X = x)$

$$p_k(x) = \Pr(Y = k \mid X = x) = \frac{\pi_k f_k(x)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x)}$$

class density

- we compute $p_k(x)$ by estimating the class prior probabilities π_k and the class densities $f_k(X)$
- we estimate the prior class probabilities from data, $\pi_k = \frac{1}{n}\sum_{i=1}^{n} I(y_i = k)$
- we **somehow** determine the probability density for point x for a class k
- we then classify each point to its most probable class

Linear Discriminant Analysis

Model assumptions

- every class is Gaussian-distributed

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

- all classes have the same variance

$$\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_k^2 = \sigma^2$$

The Bayesian classifier now becomes

$$p_k(x) = \frac{\pi_k f_k(x)}{\sum_{\ell=1}^K \pi_\ell f_\ell(x)} = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)}{\sum_{l=1}^K \pi_l \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_l)^2\right)}$$

- the logarithm of the numerator

$$-\frac{x^2}{2\sigma^2} + x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log \pi_k - \log\left(\sqrt{2\pi}\sigma\right)$$

Linear Discriminant Analysis

The Bayes-optimal choice is to classify $\boldsymbol{x}$ to the class with the largest discriminant

- the discriminant of a class $\boldsymbol{k}$ is the log-probability that cancels in the log odds

$$\log\left(\frac{p_k(x)}{p_l(x)}\right) = \delta_k(x) - \delta_l(x)$$

- where

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log \pi_k$$

  is the log-numerator from previous slide with the class-independent terms removed

Example Linear Discriminant Analysis

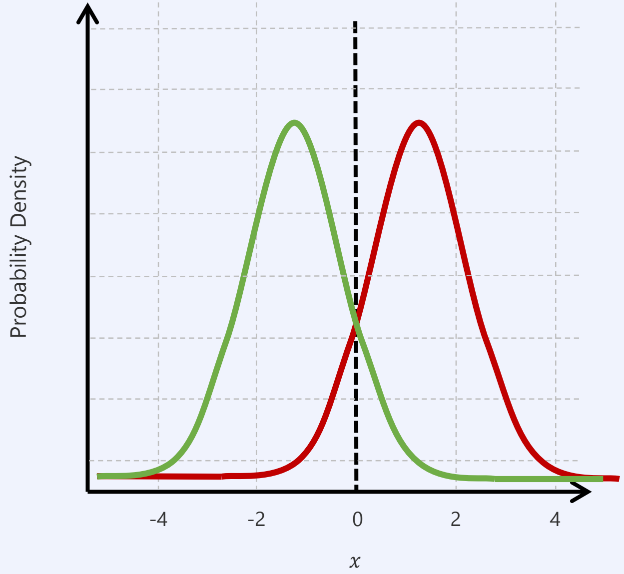
If $\pi_1 = \pi_2$ we classify an observation x to class 1 if

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

- the Bayes decision boundary is the set of points for which both discriminants are equal, i.e.

$$x = \frac{\mu_1^2 - \mu_2^2}{2(\mu_1 - \mu_2)} = \frac{\mu_1 + \mu_2}{2}$$

- the figure shows two 1D normal density functions.
- the dashed line represents the Bayes decision boundary, at which an observation is equally likely to belong to either class



$\mu_1 = -1.25$
$\mu_2 = 1.25$
$\sigma_1 = \sigma_2 = 1$

Fitting Univariate LDA Models

In general, we do not know the underlying class densities

- we estimate these using the finite training sample

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

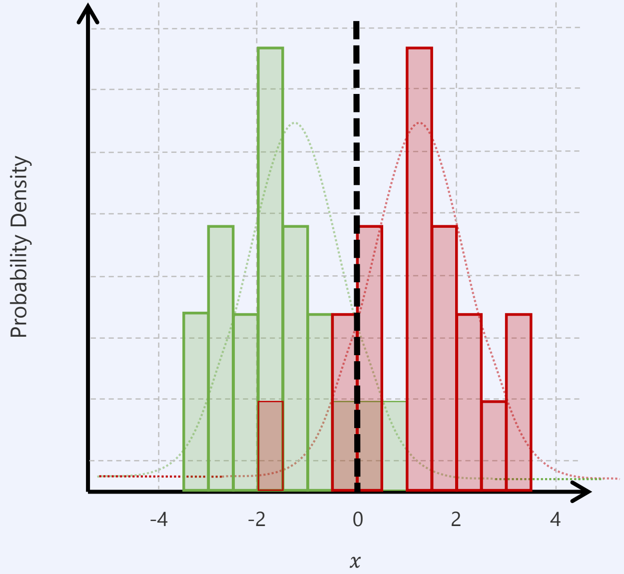
$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

$$\pi = n_k / n$$

- we assign x to the class with the largest fitted discriminant

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log \hat{\pi}_k$$

- note that the discriminants are linear (!)



LDA fit over 20 samples per class, fitted decision boundary in dashed black. Bayes error 10.6%, LDA test error 11.1%

Multivariate LDA

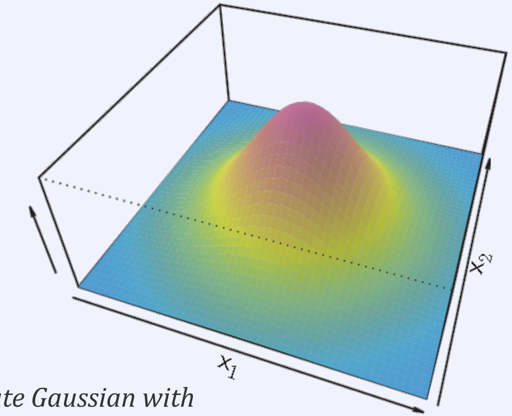
Model **assumptions**

- each class is a multivariate Gaussian
- the covariance matrix is the same for all classes

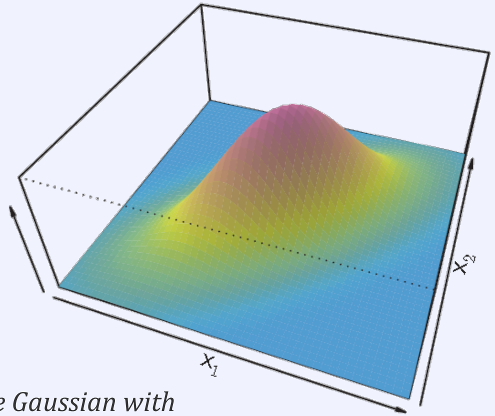
$$f_{k(x)} = \frac{1}{(2\pi)^{\frac{p}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}^{-1}(x-\mu_k)\right)$$

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

- $\mathbf{\Sigma}$ is the $p \times p$ covariance matrix of the inputs $\mathbf{\Sigma} = \mathrm{Cov}(x)$



Multivariate Gaussian with two uncorrelated predictors



Multivariate Gaussian with two correlated predictors (0.7)

Multivariate LDA

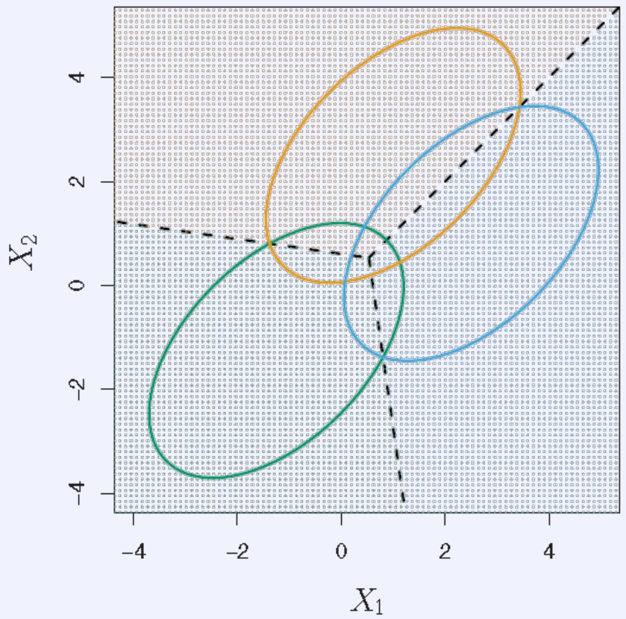
Model **assumptions**

- each class is a multivariate Gaussian
- the covariance matrix is the same for all classes

$$f_{k(x)} = \frac{1}{(2\pi)^{\frac{p}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}^{-1}(x-\mu_k)\right)$$

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

- $\mathbf{\Sigma}$ is the $p \times p$ covariance matrix of the inputs $\mathbf{\Sigma} = \mathrm{Cov}(x)$



2D synthetic data example with three classes. Ellipses contain 95% of the class probability mass, the Bayes decision boundaries are dashed

(ESL 4.4.3)

Multivariate LDA

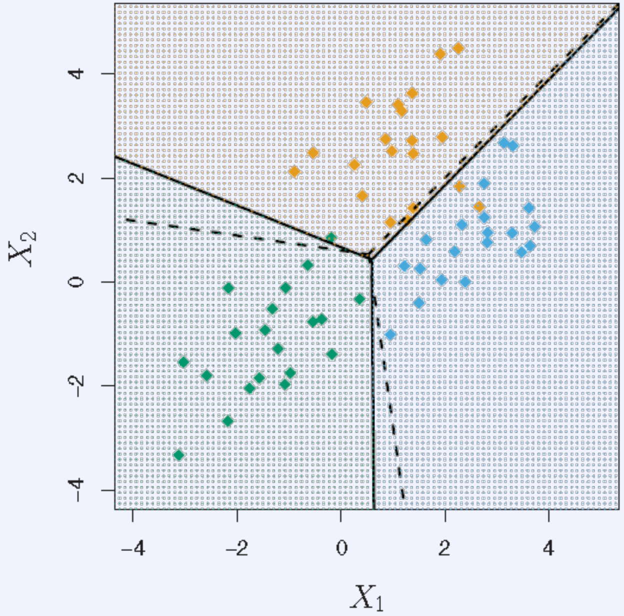
Model assumptions

- each class is a multivariate Gaussian
- the covariance matrix is the same for all classes

$$f_{k(x)} = \frac{1}{(2\pi)^{\frac{p}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}^{-1}(x-\mu_k)\right)$$

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

- $\mathbf{\Sigma}$ is the $p \times p$ covariance matrix of the inputs $\mathbf{\Sigma} = \mathrm{Cov}(x)$
- model is fitted using sample estimates similar to the 1D case
- μ easy, but $\mathbf{\Sigma}$ is the hardest to estimate



LDA fit of data set comprising 20 samples from each class, decision boundary in black

Multivariate LDA

Example **default** with **balance** and **student** as inputs

- training error for LDA is 2.75%
- data is highly unbalanced, we have only 3,33% positives
- the No-only classifier has an error of already only 3,33%

Sensitivity $\text{Sens} = TP/(TP+FN) = TP/P^*$

- fraction of correctly predicted positives

Specificity $\text{Spec} = TN/(TN+FP) = TN/N^*$

- fraction of correctly predicted negatives

- No $\text{Sens} = \frac{0}{333} = 0\%$, $\text{Spec} = \frac{9{,}667}{9{,}667} = 100\%$
- LDA $\text{Sens} = \frac{81}{333} = 24.3\%$, $\text{Spec} = \frac{9{,}644}{9{,}667} = 99.8\%$
- LDA approximates the Bayes classifier, minimizing error on all observations

LDA Model Results

Prediction	True Default Status No (−)	Yes (+)	Total
No (−)	9,644	252	9,896
Yes (+)	23	81	104
Total	**9,667**	**333**	**10,000**

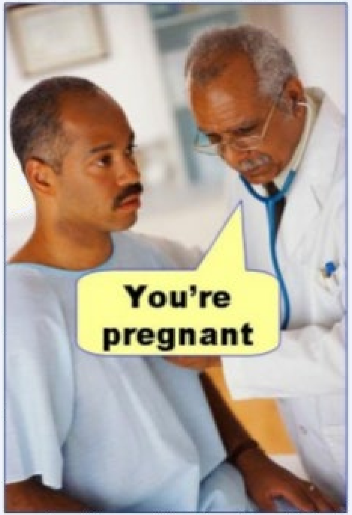
Prediction	True Default Status No (−)	Yes (+)	Total
No (−)	TN	FN	N
Yes (+)	FP	TP	P
Total	N^*	P^*	n

Confusion matrix

Type-1 error
False positive

Type-2 error
False negative

Types of Errors – a handy guide



Type I error
(false positive)

Type II error
(false negative)

Multivariate LDA

Biasing the classifier trades sensitivity for specificity

$$\log((p_k(x))/(p_l(x))) = \delta_k(x) - \delta_l(x)$$

- move the decision threshold between class **no** or **yes** from

  $$\Pr(\,\textbf{default} = \text{yes} \mid X = x\,) = 0.5$$

- we can increase sensitivity by choosing
  $\Pr(\,\textbf{default} = \text{yes} \mid X = x\,) < 0.5$
  as this assigns more points to class **yes**
- for $\Pr(\,\textbf{default} = \text{yes} \mid X = x\,) < 0.2$
  - Sens = $195/333 = 58.6\%$
  - Spec = $9{,}432/9{,}667 = 97.6\%$
  - Error = $373/10{,}000 = 3.73\%$

For a threshold of 0.5 we get
Sens = 24.3%, Spec = 99.8%, Error=2.75%

Prediction	**True Default Status** No (−)	Yes (+)	Total
No (−)	9,644	252	9,896
Yes (+)	23	81	104
Total	**9,667**	**333**	**10,000**

Prediction	**True Default Status** No (−)	Yes (+)	Total
No (−)	9,432	138	9,570
Yes (+)	235	195	430
Total	**9,667**	**333**	**10,000**

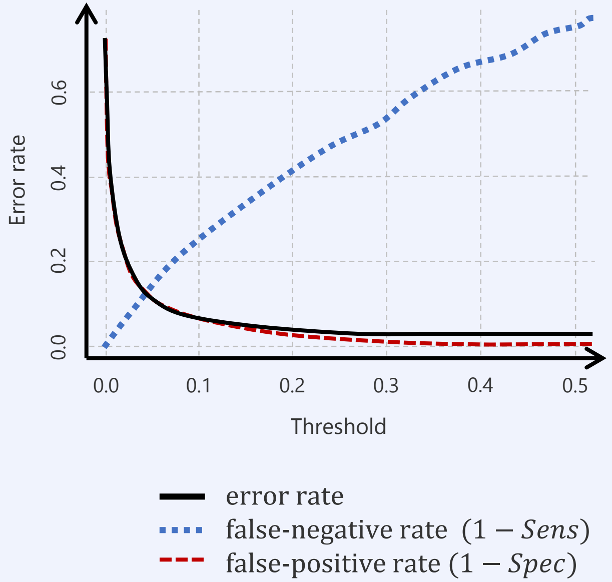
While for a threshold of 0.2 we have
Sens = 58.6%, Spec = 97.6%, Error=3.73%

Multivariate LDA

Biasing the classifier trades sensitivity for specificity

$$\log\big((p_k(x))/(p_l(x))\big) = \delta_k(x) - \delta_l(x)$$

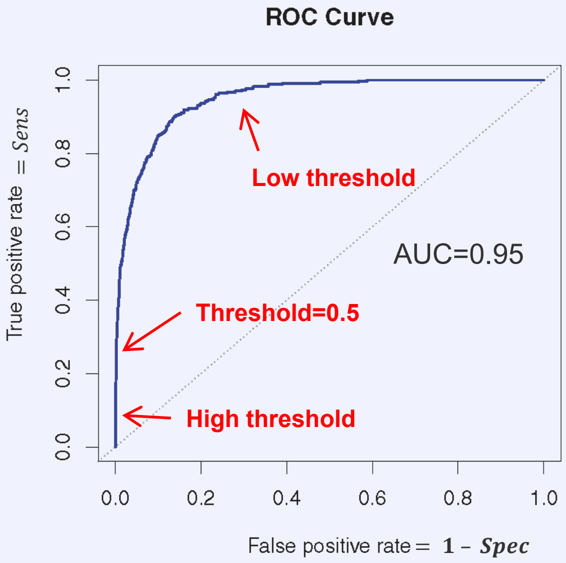
- move the decision threshold between class **no** or **yes** from
  $$\Pr(\,\mathbf{default} = \text{yes} \mid X = x\,) = 0.5$$
- we can increase sensitivity by choosing
  $\Pr(\,\mathbf{default} = \text{yes} \mid X = x\,) < 0.5$
  as this assigns more points to class **yes**
- for $\Pr(\,\mathbf{default} = \text{yes} \mid X = x\,) < 0.2$
  - $\text{Sens} = 195/333 = 58.6\%$
  - $\text{Spec} = 9{,}432/9{,}667 = 97.6\%$
  - $\text{Error} = 373/10{,}000 = 3.73\%$
- error rates change smoothly when we move the threshold



ROC Curves

Receiver-Operating Characteristic (ROC) curves plot $Sens$ against $1 - Spec$ for all thresholds

- Area Under the ROC-Curve (AUC) measures the quality of a classifier independent of the choice of that threshold
- optimally $Spec = Sens = 1$ for any threshold ($AUC = 1$)
- random classifier performs on the diagonal ($AUC = 0.5$)
- if the ROC curve goes below the diagonal, we can improve accuracy by inverting the classifier



ROC curves are not influenced by imbalance of the data

- balance only affects locations of a threshold along the curve

Quadratic Discriminant Analysis (QDA)

We give up the assumption that the covariances of all classes are all the same

For QDA we have

$$f_{k(x)} = \frac{1}{(2\pi)^{\frac{p}{2}}|\mathbf{\Sigma}_k|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}_k^{-1}(x-\mu_k)\right)$$

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

For LDA we had

$$f_{k(x)} = \frac{1}{(2\pi)^{\frac{p}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}^{-1}(x-\mu_k)\right)$$

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

Quadratic Discriminant Analysis (QDA)

We give up the assumption that the covariances of all classes are all the same

For QDA we have

$$f_{k(x)} = \frac{1}{(2\pi)^{\frac{p}{2}}|\mathbf{\Sigma}_k|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}_k^{-1}(x-\mu_k)\right)$$

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

For LDA we had

$$f_{k(x)} = \frac{1}{(2\pi)^{\frac{p}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}^{-1}(x-\mu_k)\right)$$

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

Quadratic Discriminant Analysis (QDA)

In QDA every class has its own covariance matrix

$$f_{k(x)} = \frac{1}{(2\pi)^{\frac{p}{2}}|\mathbf{\Sigma}_k|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}_k^{-1}(x-\mu_k)\right)$$

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

- class boundaries are now quadratic curves
- we fit a different covariance matrix estimate per class
- LDA has $(2K+p+1)p/2$ parameters,
- QDA has $Kp(p+3)/2$ parameters

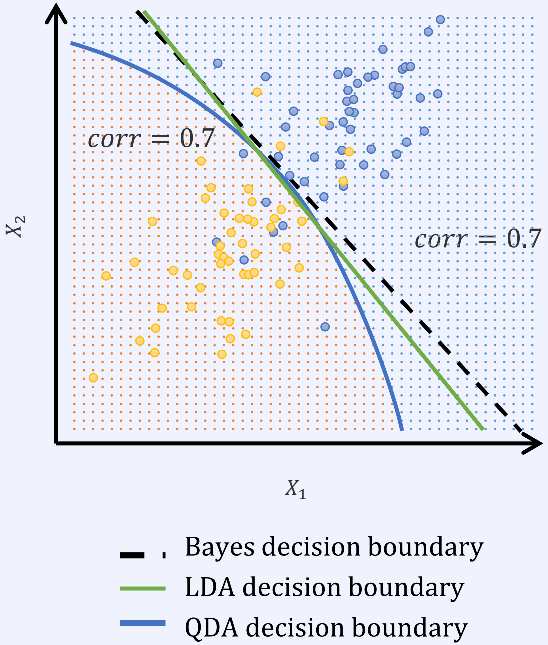
Example

- for $p=4, K=2$, LDA has 18 parameters, QDA has 28 parameters
- for $p=8, K=2$, LDA has 52 parameters, QDA has 88 parameters

Example LDA vs. QDA

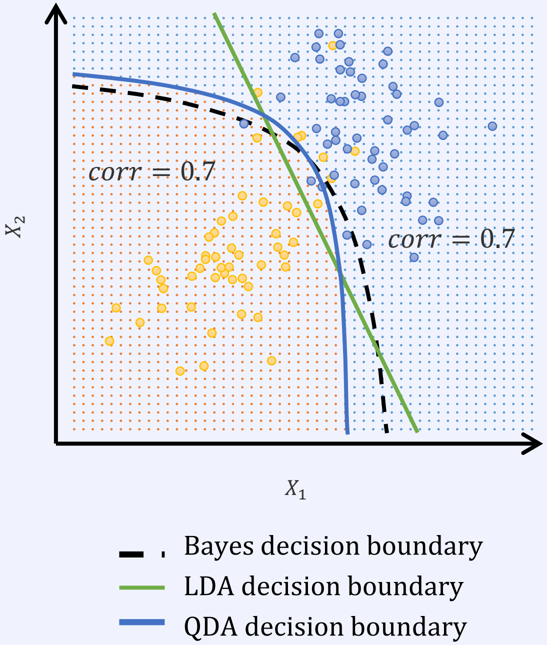
Two-class problem with $\Sigma_1 = \Sigma_2$

QDA overtrains



Two-class problem with $\Sigma_1 \neq \Sigma_2$

LDA overtrains



Fitting LDA and QDA Models

Again, we use sample estimates

- $\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$
- $\widehat{\boldsymbol{\Sigma}} = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$
- $\widehat{\boldsymbol{\Sigma}}_k = \frac{1}{n_k-K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$
- $\pi_k = n_k / n$

To simplify calculation we use the eigenvalue decomposition of the covariance matrices

$$\widehat{\boldsymbol{\Sigma}}_k = \boldsymbol{U}_k \boldsymbol{D}_k \boldsymbol{U}_k^T$$

- $\boldsymbol{U}_k$ is a $\boldsymbol{p} \times \boldsymbol{p}$ orthonormal matrix
- $\boldsymbol{D}_k$ is a diagonal matrix of decreasing positive eigenvalues $\boldsymbol{d}_{kl}$

The main terms in the discriminants,

$$\delta_k(x) = -\frac{1}{2} \log\left|\widehat{\boldsymbol{\Sigma}}_k\right| - \frac{1}{2} (x - \mu_k)^T \widehat{\boldsymbol{\Sigma}}_k^{-1} (x - \mu_k) + \log \pi_k$$

then turn into

$$\log\left|\widehat{\boldsymbol{\Sigma}}_k\right| = \sum_l \log d_{kl}$$

$$(x - \hat{\mu}_k)^T \widehat{\Sigma}_k^{-1} (x - \hat{\mu}_k) = \left[\boldsymbol{U}_k^T (x - \hat{\mu}_k)\right]^T D_k^{-1} \left[U_k^T (x - \hat{\mu}_k)\right]$$

The LDA estimator

- Step 1: Normalize $\boldsymbol{X}$ to spherical covariance
$$X^* \leftarrow \boldsymbol{D}^{-1/2} \boldsymbol{U}^T X$$
- Step 2: Classify to the closest class centroid in the transformed space, where distance is weighted by the class prior probabilities $\boldsymbol{\pi}_k$

Comparing Different Classifiers

Comparison of the Classification Methods

We now know four classifiers: $\boldsymbol{k}$-NN, LDA, QDA and logistic regression

- when should we use which?

Logistic regression and LDA are surprisingly closely related

- univariate binary setting $\quad p_2(x) = 1 - p_1(x)$
- log-odds for LDA are $\quad \log\frac{p_1(x)}{1-p_1(x)} = c_0 + c_1 x$

  (difference of two linear discriminants)
- while for logistic regression $\quad \log\frac{p_1(x)}{1-p_1(x)} = \beta_0 + \beta_1 x$

Similar, but different

- $\boldsymbol{\beta_0}$ and $\boldsymbol{\beta_1}$ are maximum likelihood estimates
- $\boldsymbol{c_0}$ and $\boldsymbol{c_1}$ are estimated from sample mean and variance of Gaussian distribution
- relationship extends to multivariate data: LR and LDA often give similar results – but not always!
- LDA makes stronger assumptions

Comparison of the Classification Methods

We now know four classifiers: $\boldsymbol{k}$-NN, LDA, QDA and logistic regression

- when should we use which?

k-NN is nonparametric and tends to work better for strongly nonlinear settings

- it does not allow for inference, i.e. we do not get a model that we can learn from

QDA is a compromise between LDA and $\boldsymbol{k}$-NN

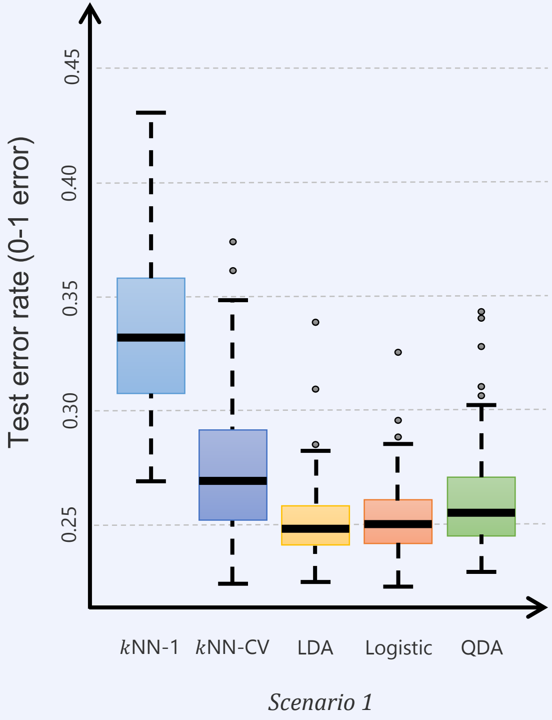
Comparing the Classification Methods

Scenario 1

- 100 random training data sets, $p = 2$ predictors, $K = 2$ classes
- 20 observations per class
- observations in different classes uncorrelated normal variables with different means and the same variance (spherical Gaussian)
- this matches the LDA assumptions of LDA

Observations

- LDA works very well
- logistic regression assumes a linear decision boundary, performs only slightly worse than LDA
- k-NN overtrains, as does QDA



V

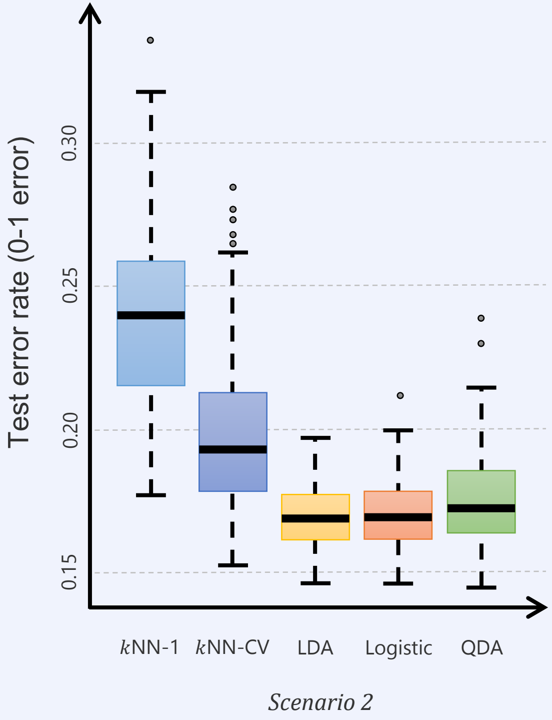
Comparing the Classification Methods

Scenario 2

- 100 random training data sets, $p = 2$ predictors, $K = 2$ classes
- like scenario 1, but predictors in each class now have a correlation of -0.5 (elliptical multivariate Gaussian)

Observations

- relative performances are similar to scenario 1



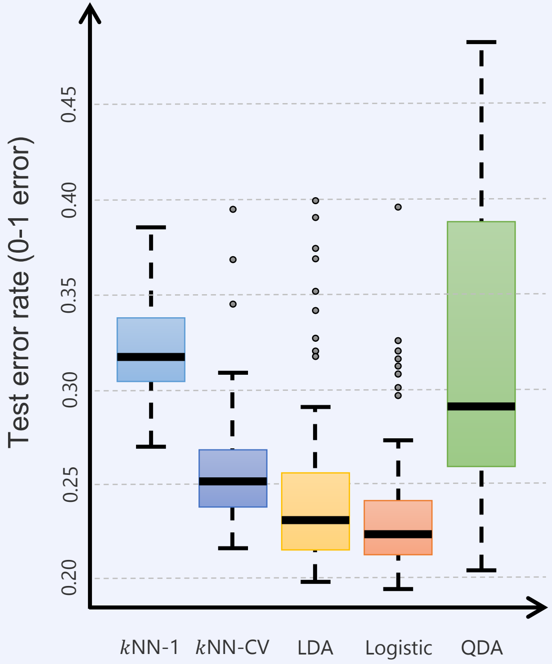
Comparing the Classification Methods

Scenario 3

- 100 random training data sets, $p = 2$ predictors, $K = 2$ classes
- X_1 and X_2 are generated using a $\boldsymbol{t}$-distribution
- more extreme points than with a Gaussian
- decision boundary is linear, but, setup violates LDA assumption

Observations

- logistic regression performs best
- QDA deteriorates because of non-normality of the data



Scenario 3

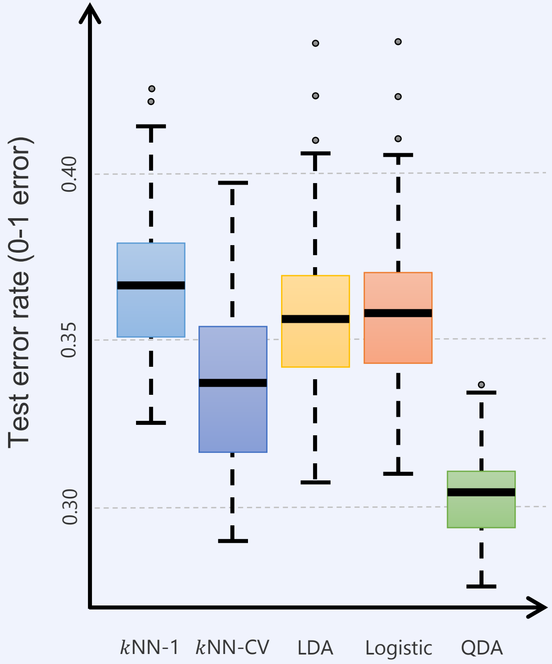
Comparing the Classification Methods

Scenario 4

- 100 random training data sets, $p = 2$ predictors, $K = 2$ classes
- class 1: normal distribution with correlation $\mathbf{0.5}$ to predictors
- class 2: normal distribution with correlation $-\mathbf{0.5}$ to predictors
- assumptions of QDA are met (but not LDA!)

Observations

- QDA outperforms all other methods



Scenario 4

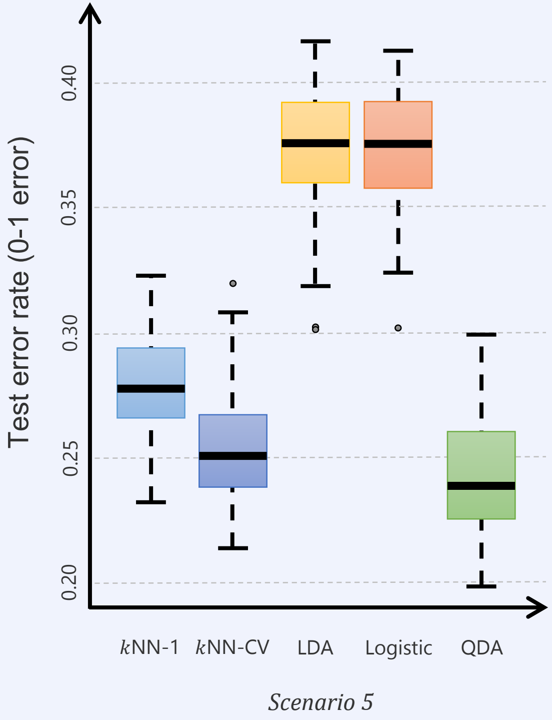
Comparing the Classification Methods

Scenario 5

- 100 random training data sets, $p = 2$ predictors, $K = 2$ classes
- two normal distributions with uncorrelated predictors
- inputs X_1^2, X_2^2 and $X_1 X_2$, not X_1 and X_2
- the decision boundary is quadratic

Observations

- QDA performs best
- kNN (CV) follows closely
- the linear methods all perform poorly



Comparing the Classification Methods

Scenario 6

- 100 random training data sets, $p = 2$ predictors, $K = 2$ classes
- like 5, but responses sampled from a complicated linear function

Observations

- even QDA cannot model data well
- k-NN-1 overtrains
- k-NN (CV) outperforms all parametric approaches
- smoothness must be chosen carefully