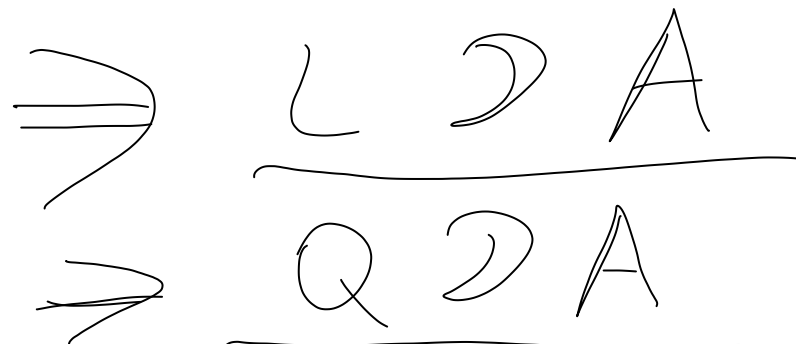


Chapter 4 - classification from

Hastie et al. (2009) and James et al. (2013)

Lecturers: Geir Storvik and Aliaksandr Hubin

LDA, QDA

- ▶ $p = 1 : p_k(\mathbf{x}) = N(\mu_k, \sigma^2)$
 - ▶ $p > 1 : p_k(\mathbf{x}) = N(\mu_k, \Sigma)$
 - ▶ $p = 1 : p_k(\mathbf{x}) = N(\mu_k, \sigma_k^2)$
 - ▶ $p > 1 : p_k(\mathbf{x}) = N(\mu_k, \Sigma_k)$
 - ▶ Bayes theorem
- 

$$\Pr(Y = k | \mathbf{x}) = \frac{\Pr(Y = k) p(\mathbf{x} | Y = k)}{p(\mathbf{x})} = \frac{\Pr(Y = k) p_k(\mathbf{x})}{p(\mathbf{x})}$$

- ▶ Classification rule

$$\hat{y} = \arg \max_k \pi_k p_k(\mathbf{x})$$

LDA, QDA

Estimation LDA

$$\frac{1}{2} \mathbf{x}^T (\hat{\Sigma}_2^{-1} - \hat{\Sigma}_1^{-1}) \mathbf{x} + (\hat{\mu}_1^T \hat{\Sigma}_1^{-1} - \hat{\mu}_2^T \hat{\Sigma}_2^{-1}) \mathbf{x}$$

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i: y_i = k} \mathbf{x}_i, \quad n_k = \sum_i I(y_i = k)$$

$$\hat{\Sigma} = \frac{1}{n - K} \sum_{k=1}^K \sum_{i: y_i = k} (\mathbf{x}_i - \hat{\mu}_k)(\mathbf{x}_i - \hat{\mu}_k)^T$$

$$\mathbf{x}^T \mathbf{A} \mathbf{x} + 2 \mathbf{B} \mathbf{x} + \frac{c}{n_k} \geq 0$$

Estimating QDA

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i: y_i = k} \mathbf{x}_i, \quad n_k = \sum_i I(y_i = k)$$

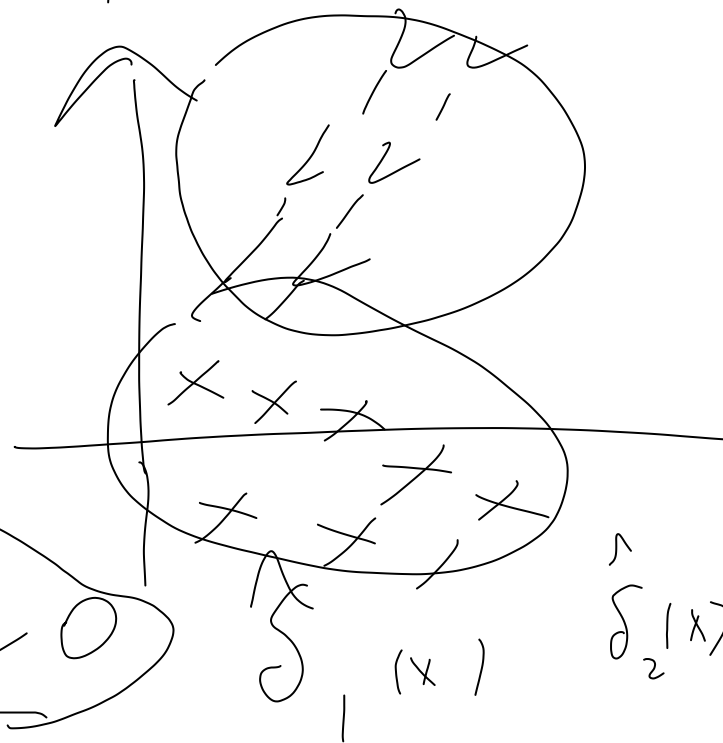
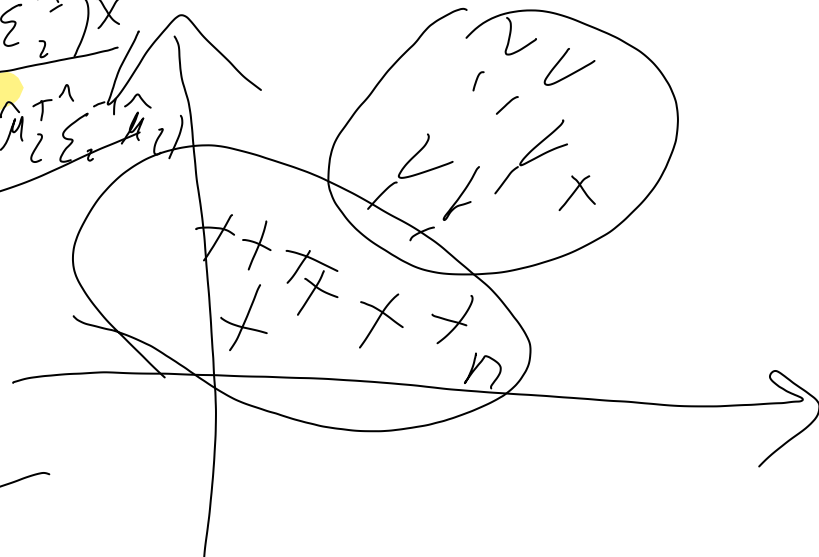
$$\hat{\Sigma}_k = \frac{1}{n_k - 1} \sum_{i: y_i = k} (\mathbf{x}_i - \hat{\mu}_k)(\mathbf{x}_i - \hat{\mu}_k)^T$$

$$\hat{\pi}_k = \frac{n_k}{n}$$

$$\hat{\delta}_1(\mathbf{x}) - \hat{\delta}_2(\mathbf{x}) \geq 0$$

$$\hat{\delta}_k(\mathbf{x}) = \log(\pi_k) - \frac{1}{2} \log(|\hat{\Sigma}_k|) + (\mathbf{x} - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1} (\mathbf{x} - \hat{\mu}_k)$$

$$= \log(\hat{\pi}_k) - \frac{1}{2} \log(|\hat{\Sigma}_k|) - \frac{1}{2} \mathbf{x}^T \hat{\Sigma}_k^{-1} \mathbf{x} + \hat{\mu}_k^T \hat{\Sigma}_k^{-1} \mathbf{x} - \frac{1}{2} \hat{\mu}_k^T \hat{\Sigma}_k^{-1} \hat{\mu}_k$$



in our
example

Regularized discriminant analysis - prevent overfitting

- ▶ Allow to choose between the common and class specific covariance matrices creates a form of regularization:

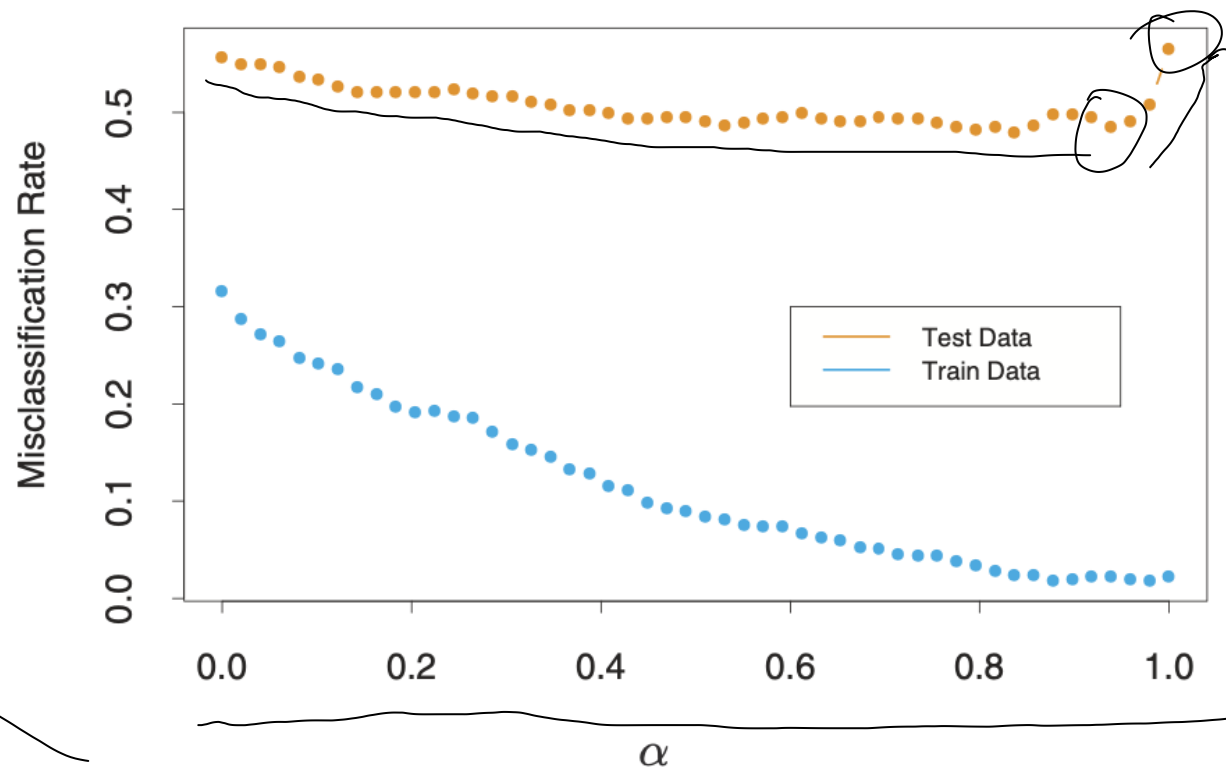
$$\hat{\Sigma}_k(a) = a\hat{\Sigma}_k + (1 - a)\hat{\Sigma}, a \in [0, 1]. \quad (1)$$

- ▶ Also possible to choose between the covariance structure or a shared common variance with no dependence in a similar way:

$$\hat{\Sigma}(\gamma) = \gamma\hat{\Sigma} + (1 - \gamma)\hat{\sigma}^2\mathbf{I}, a \in [0, 1]. \quad (2)$$

- ▶ Mixing the three variants of covariance could also be possible

Regularized Discriminant Analysis on the Vowel Data



Computations of LDA or QDA

- Compute eigen decomposition of each covariance matrix:

$$\hat{\Sigma}_k = \underbrace{U_k D_k U_k^T}, \text{ where } \underbrace{U_k} \text{ is orthonormal and } D_k \text{ is diagonal.} \quad (3)$$

- As Σ_k^{-1} decomposes as $U_k^T D_k^{-1} U_k$ we simplify computing $\delta_k(x)$ to

$$(x - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1} (x - \hat{\mu}_k) = \underbrace{[U_k^T (x - \hat{\mu}_k)]^T} \underbrace{D_k^{-1}} [U_k^T (x - \hat{\mu}_k)]. \quad (4)$$

$$\log |\hat{\Sigma}_k| = \sum_l \log d_{kl}, \quad (5)$$

where d_{kl} are eigen values

- This allows to avoid costly matrix inversion and gives matrix determinants for free.
- We can sphere the data with the common covariance prior to analysis by setting

$$X \leftarrow \underbrace{D^{-0.5} U^T X}$$

where $\hat{\Sigma} = U D U^T$ and get the common covariance as identity

K-nearest neighbors

- ▶ k-nearest neighbor: Let $\mathcal{N}_0 \subset \{1, \dots, n\}$ be the k indexes with $\mathbf{x}_i$ nearest to $\mathbf{x}_0$.
- ▶ Estimation of $\Pr(Y = k | \mathbf{X} = \mathbf{x}_0) = E[I(Y = k | \mathbf{X} = \mathbf{x}_0)]$:

$$\hat{\Pr}(Y = k | \mathbf{X} = \mathbf{x}) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = k)$$

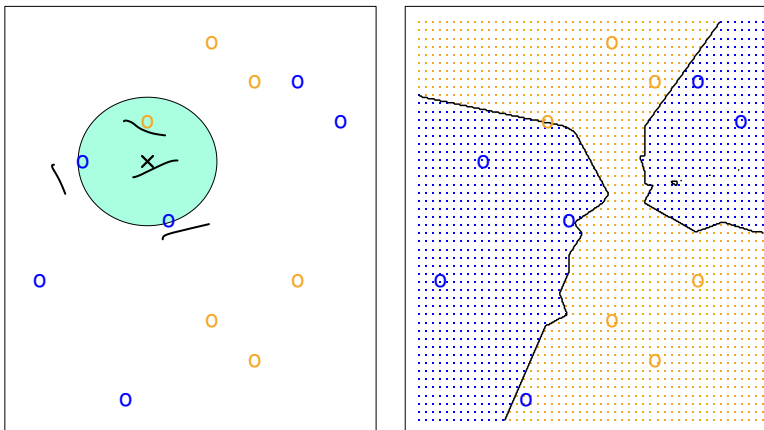
Handwritten derivation:

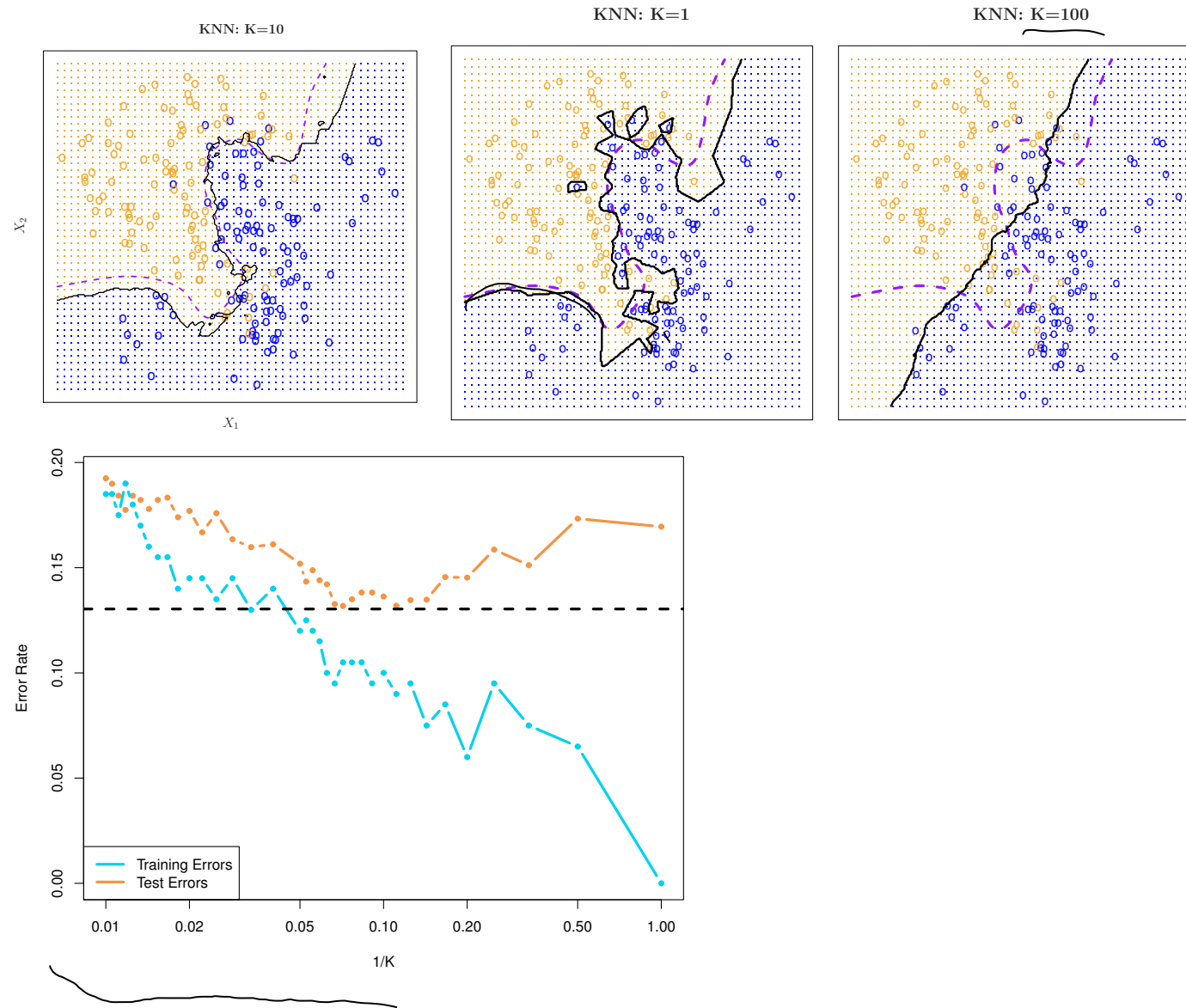
$$E[I(Y = k | \mathbf{X} = \mathbf{x}_0)] = \sum_{j=1}^K \Pr(Y = j | \mathbf{X} = \mathbf{x}_0) I(j = k) = \Pr(Y = k | \mathbf{X} = \mathbf{x}_0)$$

- ▶ Classification rule:

$$\hat{Y} = \arg \max_k \hat{\Pr}(Y = k | \mathbf{X} = \mathbf{x})$$

Classify test point with $\mathbf{X} = \mathbf{x}^*$ to the class with the highest number of cases among the K nearest points.





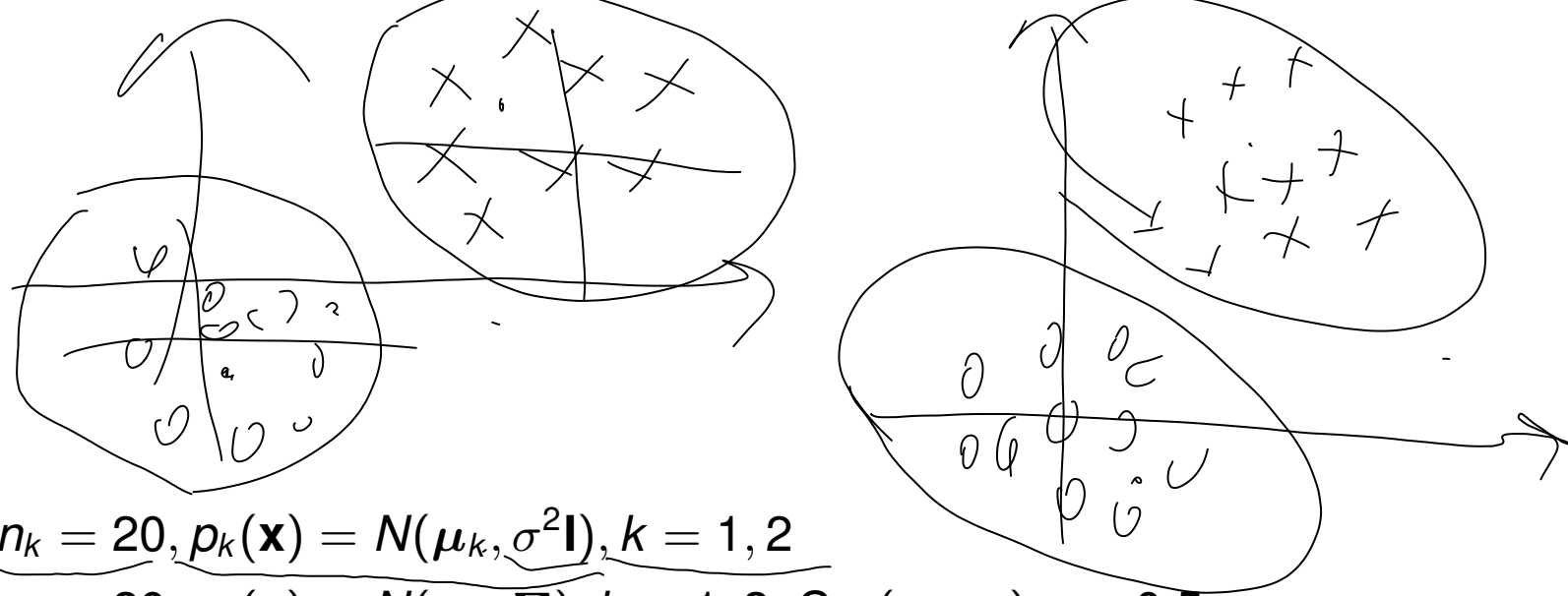
Example: phoneme data

- ▶ Language recognition
- ▶ 5 different phoneme's:
 - ▶ "sh" as in "she"
 - ▶ "dcl" as in "dark"
 - ▶ "iy" as vocal in "she"
 - ▶ "aa" as vocal in "dark"
 - ▶ "ao" as vocal in "water".
- ▶ Sound divided into 256 frequencies
- ▶ $n = 4509$ sound sequences

aa	ao	dcl	iy	sh
695	1022	757	1163	872

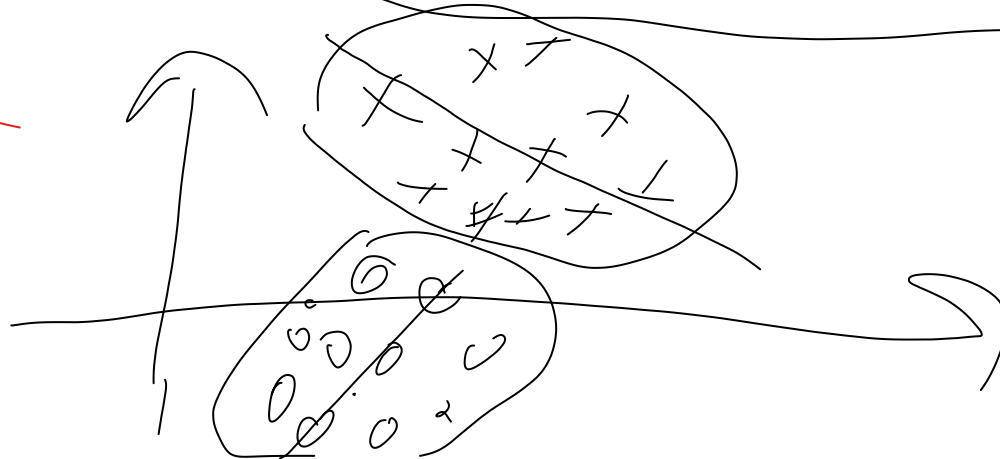
- ▶ Here: Only considering "aa" and "ao"
- ▶ `phoneme_logistic_lda_gda.R`

Comparisons



$p = 2, K = 2$

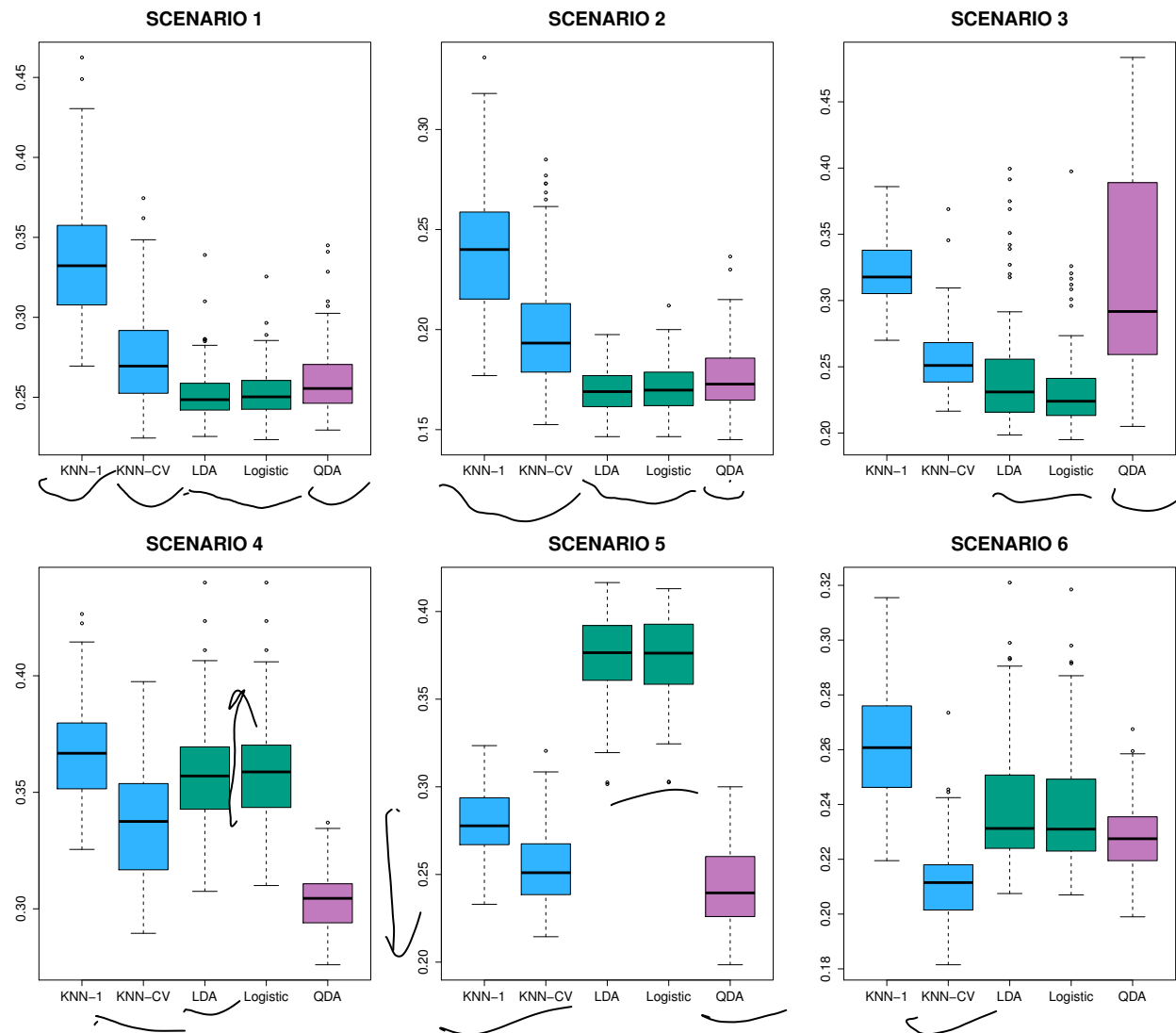
- Scenario 1: $n_k = 20, p_k(\mathbf{x}) = N(\mu_k, \sigma^2 \mathbf{I}), k = 1, 2$
- Scenario 2: $n_k = 20, p_k(\mathbf{x}) = N(\mu_k, \Sigma), k = 1, 2, \text{Cor}(x_1, x_2) = -0.5$
- Scenario 3: Like Scenario 1 but t -distribution and $n_k = 50$.
- Scenario 4: $n_k = 20, p_k(\mathbf{x}) = N(\mu_k, \Sigma_k), k = 1, 2, \text{Cor}(x_1, x_2) = -0.5$ for $k = 1$ and $\text{Cor}(x_1, x_2) = 0.5$ for $k = 2$
- Scenario 5: Logistic regression model with quadratic terms and all pairwise interactions $p(\mathbf{x}, \beta) = \frac{e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2}}{1 + e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2}}$
- Scenario 6: Logistic regression model with a complex non-linearity



$$p(\mathbf{x}, \beta) = \frac{e^{f(\mathbf{x}, \beta)}}{1 + e^{f(\mathbf{x}, \beta)}}$$

$f(\mathbf{x}, \beta)$ - non-linear function

Comparisons based on simulations



No uniformly best performing method!

Recall Logistic regression

$$Y \sim \text{Bernoulli}(p(x; \beta)) = (p(x; \beta))^Y (1 - p(x; \beta))^{1-Y}$$

- ▶ Method for **classification** with **two** possible outcomes
- ▶ Model $(\mathbf{x}^T \boldsymbol{\beta} = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p)$:

$$p(\mathbf{x}; \boldsymbol{\beta}) = \Pr(Y = 1 | \mathbf{x}; \boldsymbol{\beta}) = \frac{e^{\mathbf{x}^T \boldsymbol{\beta}}}{1 + e^{\mathbf{x}^T \boldsymbol{\beta}}} = 1 - \Pr(Y = 0 | \mathbf{x}; \boldsymbol{\beta})$$

Note:

$$\log \left(\frac{p(\mathbf{x}; \boldsymbol{\beta})}{1 - p(\mathbf{x}; \boldsymbol{\beta})} \right) = \mathbf{x}^T \boldsymbol{\beta}$$

Part of **Generalized linear models** (GLM)

Recall Logistic regression

- ▶ Method for **classification** with **two** possible outcomes
- ▶ Model ($\mathbf{x}^T \boldsymbol{\beta} = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$):

$$p(\mathbf{x}; \beta) = \Pr(Y = 1 | \mathbf{x}; \beta) = \frac{e^{\mathbf{x}^T \beta}}{1 + e^{\mathbf{x}^T \beta}} = 1 - \Pr(Y = 0 | \mathbf{x}; \beta)$$

Note:

$$\log \left(\frac{p(\mathbf{x}; \beta)}{1 - p(\mathbf{x}; \beta)} \right) = \mathbf{x}^T \beta$$

Part of **Generalized linear models** (GLM)

- ▶ Data: $\{(x_1, y_1), \dots, (x_N, y_N)\}$ (assumed independent)
- ▶ Multinomial generalization top K classes (softmax function):

$$p(\mathbf{x}; \beta_k) = \Pr(Y = k | \mathbf{x}; \beta_k) = \frac{e^{\mathbf{x}^T \beta_k}}{1 + \sum_{l=1}^{K-1} e^{\mathbf{x}^T \beta_l}}$$

$V \sim \text{multinomial}(\vec{p})$

Fitting Logistic regression

- ▶ Logistic regression models are usually fit by **maximum likelihood**, using the conditional likelihood of Y given X :

$$\ell(\beta) = \sum_{i=1}^N \log p(x_i; \beta),$$

- ▶ Let $p_1(x; \beta) = p(x; \beta)$, and $p_2(x; \beta) = 1 - p(x; \beta)$
- ▶ The **log-likelihood** then becomes

$$\begin{aligned} \ell(\beta) &= \sum_{i=1}^N \{y_i \log p(x_i; \beta) + (1 - y_i) \log (1 - p(x_i; \beta))\} \\ &= \sum_{i=1}^N \left\{ y_i \beta^T x_i - \log \left(1 + e^{\beta^T x_i} \right) \right\}, \end{aligned}$$

where $\beta = \{\beta_{10}, \beta_1\}$, and we assume that the vector of inputs x_i includes the constant term 1

Fitting Logistic regression (Newton-Raphson)

- ▶ Set the **derivatives** of log likelihood $\sum_{i=1}^N \left\{ y_i \beta^T x_i - \log \left(1 + e^{\beta^T x_i} \right) \right\}$ to **zero**:

$$\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^N x_i (y_i - p(x_i; \beta)) = 0$$

we get $p + 1$ equations nonlinear in β

- ▶ Since x_i is 1, the first score equation specifies that $\sum_{i=1}^N y_i = \sum_{i=1}^N p(x_i; \beta)$ the expected number of class ones matches the observed number
- ▶ To solve the score equations, we use the Newton-Raphson algorithm, which requires the second-derivative or Hessian matrix

$$\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} = - \sum_{i=1}^N x_i x_i^T p(x_i; \beta) (1 - p(x_i; \beta))$$

- ▶ Starting with β^{old} , a single Newton update is:

$$\beta^{\text{new}} = \beta^{\text{old}} - \left(\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} \right)^{-1} \frac{\partial \ell(\beta)}{\partial \beta}$$

where the derivatives are evaluated at β^{old}

Fitting Logistic regression (IRLS)

- ▶ Let $\mathbf{y}$ denote the vector of y_i values, $\mathbf{X}$ the $N \times (p+1)$ matrix of x_i values
- ▶ Let $\mathbf{p}$ the vector of fitted probabilities with i th element $p(x_i; \beta^{\text{old}})$
- ▶ Let $\mathbf{W}$ a $N \times N$ diagonal matrix of weights with i th diagonal element $p(x_i; \beta^{\text{old}})(1 - p(x_i; \beta^{\text{old}}))$
- ▶ Then we have

$$\frac{\partial \ell(\beta)}{\partial \beta} = \mathbf{X}^T (\mathbf{y} - \mathbf{p})$$

$$\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} = -\mathbf{X}^T \mathbf{W} \mathbf{X}$$

- ▶ And the Newton step becomes

$$\begin{aligned} \beta^{\text{new}} &= \beta^{\text{old}} + (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T (\mathbf{y} - \mathbf{p}) \\ &= (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} (\mathbf{X} \beta^{\text{old}} + \mathbf{W}^{-1} (\mathbf{y} - \mathbf{p})) \\ &= (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} \mathbf{z} \end{aligned}$$

Handwritten notes: $+ (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T (\mathbf{y} - \mathbf{p})$ and $\beta^{\text{old}} (\mathbf{y}^T \mathbf{W} \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{W} \mathbf{X})$

Properties of a fitted Logistic regression

$$\hat{\beta} \xrightarrow{n \rightarrow \infty} N(\beta, \Sigma(\beta)) = N(\beta, (X^T W X)^{-1}) = N(\beta, \Sigma)$$

- ▶ From **standard MLE theory** we have that β is **consistent** (i.e., converges to the true β) and the distribution of β converges to $N(\beta, (X^T W X)^{-1})$
- ▶ **Wald test** which can be used to test for exclusion of a term.
- ▶ They do not require iterative fitting, and are **based on the maximum-likelihood fit** of the current model

$$z = \frac{\hat{\beta}_j - \beta_j}{\hat{\sigma}_j} \sim N(0, 1)$$

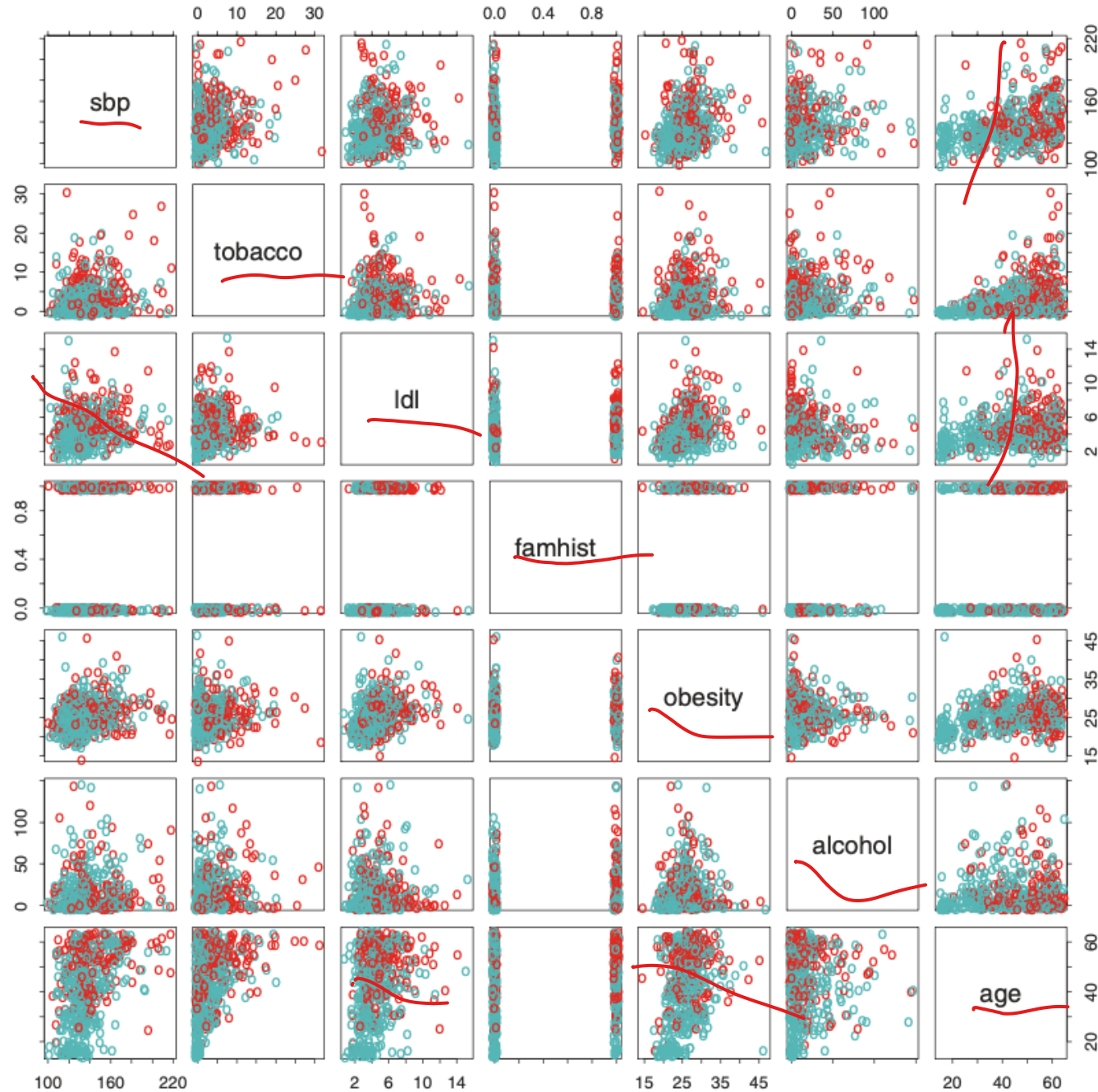
also significant

significant

$z \geq 1.96 \text{ or } z \leq -1.96$

Example: South African Heart Disease (Idl - bad cholesterol)

$$Y_i \in \{0, 1\}$$



Fit a logistic regression with all covariates

- **Wald test:** Z score **greater** than approximately 2 in absolute value is significant at the 5% level of significance

	Coefficient	Std. Error	Z Score
(Intercept)	-4.130	0.964	-4.285
sbp	0.006	0.006	1.023
tobacco	0.080	0.026	3.034
ldl	0.185	0.057	3.219
famhist	0.939	0.225	4.178
obesity	-0.035	0.029	-1.187
alcohol	0.001	0.004	0.136
age	0.043	0.010	4.184

- Systolic blood pressure (sbp) is not significant!
- Nor is obesity, and its sign is negative.
- Result of the correlation between the set of predictors.
- Marginally, both sbp and obesity are significant, and with positive sign.

Perform step-wise model selection

- ▶ **Drop the least significant coefficient**, and refit the model until no further terms can be dropped from the model
- ▶ Or use the residual deviance of a fitted model which is minus twice its log-likelihood, and the deviance between two models is the difference of their individual residual deviance (in analogy to sums-of-squares). **Drop the covariate resulting in the best deviance between two models** with and without it.

	Coefficient	Std. Error	Z score
(Intercept)	-4.204	0.498	-8.45
tobacco	0.081	0.026	3.16
ldl	0.168	0.054	3.09
famhist	0.924	0.223	4.14
age	0.044	0.010	4.52

Interpret tobacco

$$\log \frac{p(x_i; \beta)}{1 - p(x_i; \beta)} = x_i^T \beta \log \left[\frac{p(x_i; \beta)}{1 - p(x_i; \beta)} \right] / \frac{p(x_i^*; \beta)}{1 - p(x_i^*; \beta)}$$

$x_i^T \beta$

- Variable tobacco is measured in total lifetime usage in kilograms, with a median of 1.0kg for the controls and 4.1kg for the cases
- An increase of 1kg in lifetime tobacco usage accounts for an increase in the odds ratio of coronary heart disease of $x_i^* = (x_{i1}, \dots, x_{ij-1}, x_j + \Delta, x_{j+1}, \dots, x_{in})$

$$\exp(0.081 \times (x_{\text{tobacco}} - x_{\text{tobacco}}^*)) = \exp(0.081) = 1.084$$

or 8.4%

- Likewise, an approximate 95% confidence interval of $\exp(0.081 \pm 2 \times 0.026) = (1.03, 1.14)$

	Coefficient	Std. Error	Z score
(Intercept)	-4.204	0.498	-8.45
tobacco	<u>0.081</u>	<u>0.026</u>	3.16
ldl	0.168	0.054	3.09
famhist	0.924	0.223	4.14
age	0.044	0.010	4.52

$$\frac{\Delta \beta}{\exp(\Delta \beta)}$$

L_1 regularized Logistic Regression

if $\lambda_1 = 1 \Rightarrow$ lasso
 if $\lambda_2 = 0 \Rightarrow$ ridge
 if $\lambda_1 > 0, \lambda_2 > 0 \Rightarrow$ elastic net

- ▶ L_1 penalty used in the lasso can be used for variable selection and shrinkage with any linear regression model.
- ▶ For logistic regression, we would maximize a penalized version becomes:

$$\max_{\beta_0, \beta} \left\{ \sum_{i=1}^N \left[y_i (\beta_0 + \beta^T x_i) - \log (1 + e^{\beta_0 + \beta^T x_i}) \right] - \lambda \sum_{j=1}^p |\beta_j| \right\}$$

λ for parameterization of glmnet in R

- ▶ R package **glmnet** can fit coefficient paths for very large logistic regression problems efficiently (large in N or p). Its algorithms can exploit sparsity in the predictor matrix $\mathbf{X}$
- ▶ L_2 penalties $\lambda \sum_{j=1}^p (\beta_j)^2$ are also possible to be addressed. As well as the mix of L_1 and L_2 penalties $\lambda_1 \sum_{j=1}^p |\beta_j| + \lambda_2 \sum_{j=1}^p (\beta_j)^2$

1 - strength of penalty
 2 - choosing between L_1, L_2 or mix

Logistic regression or LDA?

- ▶ **Log-posterior odds** between classes are linear functions of x :

$$\begin{aligned}\log \frac{\Pr(Y = k | X = x)}{\Pr(Y = K | X = x)} &= \log \frac{\pi_k}{\pi_K} - \frac{1}{2} (\mu_k + \mu_K)^T \Sigma^{-1} (\mu_k - \mu_K) \\ &\quad + x^T \Sigma^{-1} (\mu_k - \mu_K) \\ &= \alpha_{k0} + \alpha_k^T x.\end{aligned}$$

- ▶ This **linearity** is a consequence of the **Gaussian** assumption for the **class densities**, as well as the assumption of a **common covariance matrix**
- ▶ The linear **logistic model** by construction has **linear logits**:

$$\log \frac{\Pr(Y = k | X = x)}{\Pr(Y = K | X = x)} = \beta_{k0} + \beta_k^T x$$

- ▶ Although they have **exactly the same form**, the **difference** lies in the way the linear **coefficients** are **estimated**

Logistic regression or LDA?

- We can **write** the **joint density** of X and Y as

$$\Pr(X, Y = k) \neq \Pr(X) \Pr(Y = k | X)$$

where $\Pr(X)$ denotes the marginal density of the inputs X .

- ▶ For both LDA and logistic regression, the **second term** on the right has the **logit-linear** form

$$\Pr(Y = k \mid X = x) = \frac{e^{\beta_{k0} + \beta_k^T x}}{1 + \sum_{l=1}^{K-1} e^{\beta_{l0} + \beta_l^T x}}$$

where we have again arbitrarily chosen the last class as the reference

- ▶ The **logistic regression** model leaves the **marginal density** of X as an **arbitrary** density function $\Pr(X)$, and fits the parameters of $\Pr(Y | X)$ by maximizing the conditional likelihoods $\Pr(Y = k | X)$
- ▶ Whilst $\Pr(X)$ is **totally ignored**, we can think of this marginal density as **uniform** with a probability of $1/N$ at each observation
- ▶ If one has $\Pr(X)$ **available** externally (like population census for demographic variables) that could be used for **post-stratification purpose**

Logistic regression or LDA?

$$p(x) = \int_y p(x, y) dy = \int p(x|y) p(y) dy$$

- ▶ With **LDA** we directly have a **joint density**

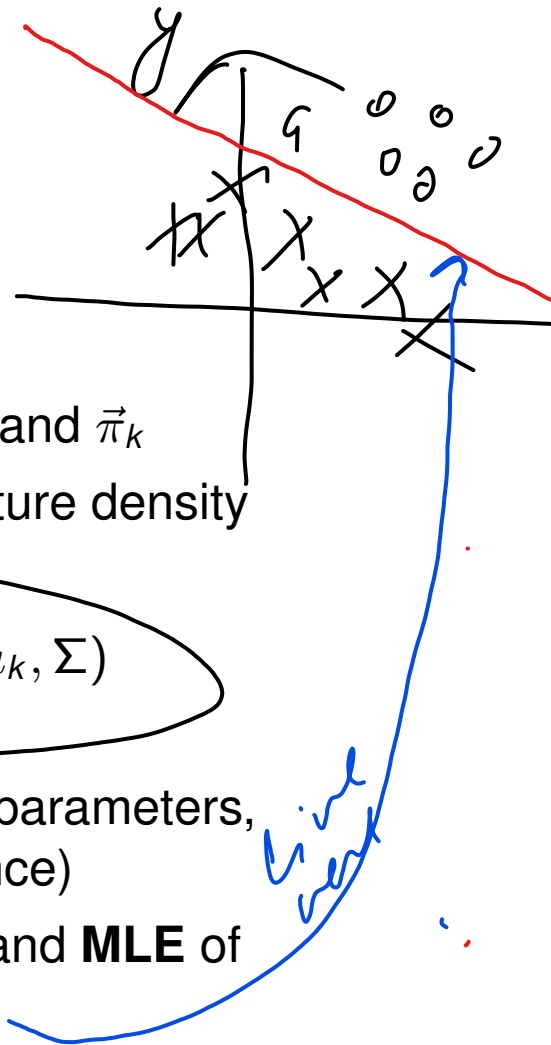
$$\Pr(X, Y = k) = \phi(X; \mu_k, \Sigma) \pi_k$$

where ϕ is the Gaussian density function.

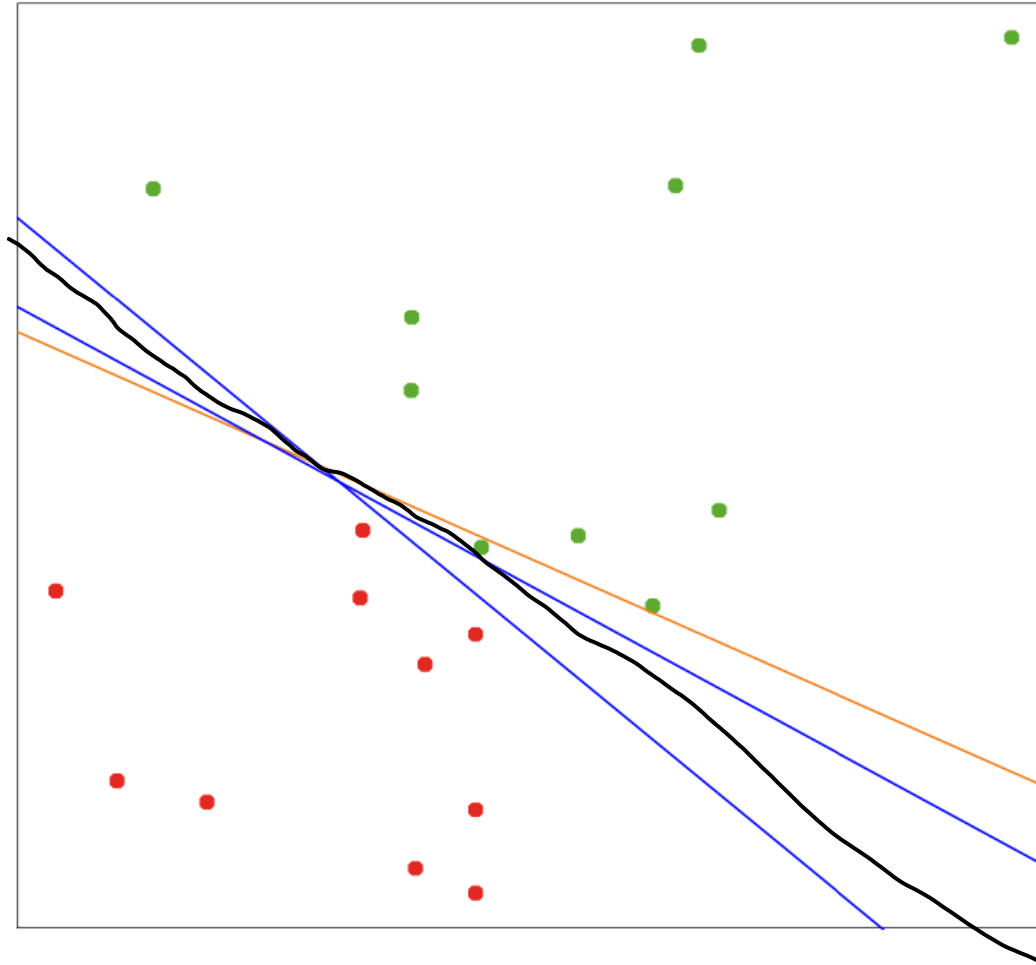
- ▶ Standard normal theory leads easily to the estimates $\hat{\mu}_k$, $\hat{\Sigma}$, and $\vec{\pi}_k$
- ▶ The marginal density $\Pr(X)$ plays a role here and has a mixture density

$$\Pr(X) = \sum_{k=1}^K \Pr(X|Y = k) \Pr(Y = k) = \sum_{k=1}^K \pi_k \phi(X; \mu_k, \Sigma)$$

- ▶ Additional **assumptions give more information** about the parameters, and hence can estimate them more **efficiently** (lower variance)
- ▶ For **perfectly separated data**, the likelihood is unbounded and **MLE** of the parameters are **undefined**
- ▶ The **LDA** coefficients for the same data are **still well defined**



Separating hyperplanes



- ▶ Blue lines are two of the infinitely many possible hyperplanes separating the classes
- ▶ Orange line is the least squares solution to the problem, obtained by regressing the $-1/1$ response Y on X (with intercept)

Separating hyperplanes

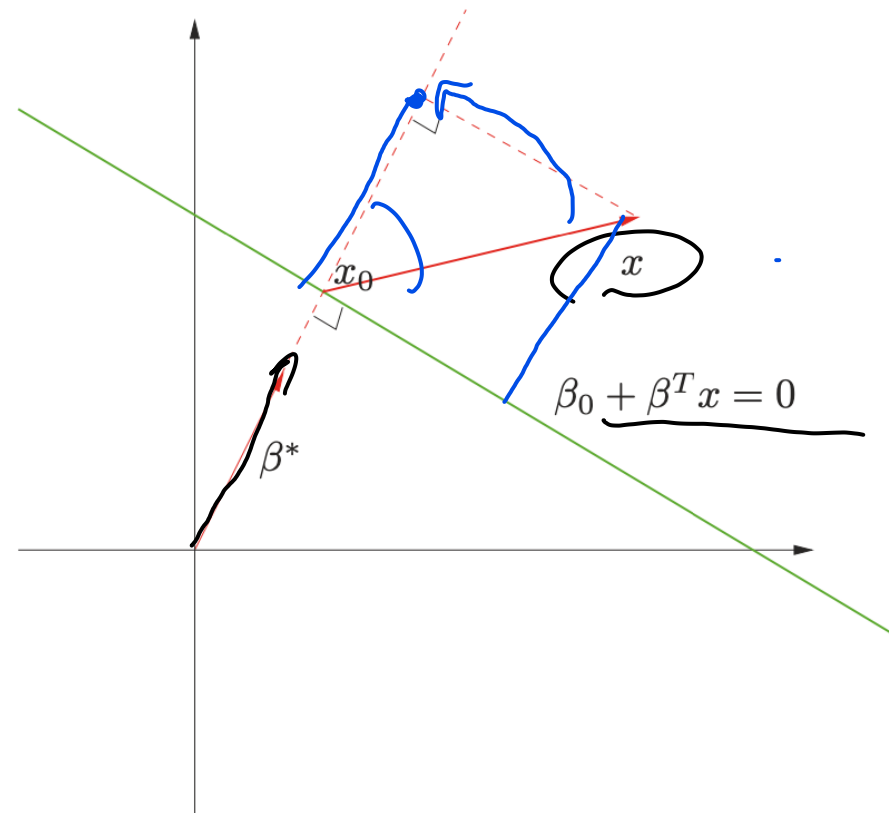
- ▶ This least squares solution does not do a perfect job in separating the points, and makes one error.
- ▶ This is the same boundary found by LDA in light of its equivalence with linear regression in the two-class case (Recall Exercise 4.2)
- ▶ Lines inducing separating hyperlanes:

$$\left\{ x : \hat{\beta}_0 + \bar{\beta}_1 x_1 + \bar{\beta}_2 x_2 = 0 \right\}$$

are called **linear perceptrons** in the engineering literature

- ▶ We shall learn how to build optimal separating hyperplanes but first we need to recall some preliminaries

The linear algebra of a hyperplane



- ▶ A unit vector $\beta^* = \beta / \|\beta\|$ perpendicular (normal) to the surface of L .
- ▶ For any point x_0 in L , $\beta^T x_0 = -\beta_0$.
- ▶ The signed distance of any point x to the closest point on L is given by

$$\beta^{*T} (x - x_0) = \frac{1}{\|\beta\|} (\beta^T x + \beta_0) \quad (6)$$

Rosenblatt's Perceptron Learning Algorithm

- ▶ The algorithm tries to find a separating hyperplane by minimizing the distance of misclassified points to the decision boundary
- ▶ If $y_i = 1$ is misclassified then $x_i^T \beta + \beta_0 < 0$ and if $y_i = -1$ is misclassified then $x_i^T \beta + \beta_0 > 0$ giving

$$D(\beta, \beta_0) = - \sum_{i \in \mathcal{M}} y_i (x_i^T \beta + \beta_0)$$

where $\mathcal{M}$ indexes the set of misclassified points. The quantity is non negative and proportional (with factor $\frac{1}{\|\beta\|}$) to the distance of the misclassified points to the decision boundary

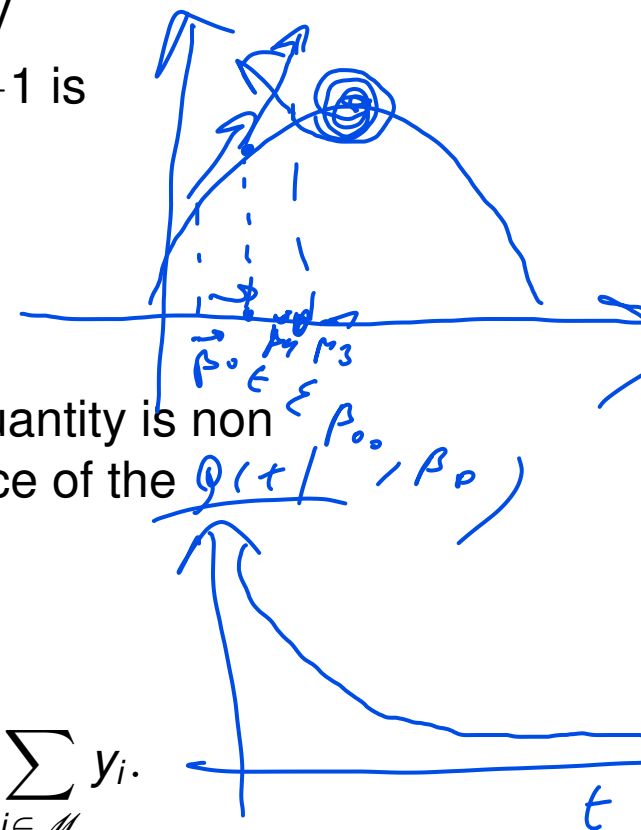
- ▶ The gradient (assuming $\mathcal{M}$ is fixed) is given by

$$\frac{\partial D(\beta, \beta_0)}{\partial \beta} = - \sum_{i \in \mathcal{M}} y_i x_i, \text{ and } \frac{\partial D(\beta, \beta_0)}{\partial \beta_0} = - \sum_{i \in \mathcal{M}} y_i.$$

- ▶ Use stochastic gradient descent **SGD** (explain on whiteboard), i.e. at each step i is sampled randomly from $\mathcal{M}$ and then a step is taken as

$$\begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} \leftarrow \begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} + \rho \begin{pmatrix} y_i x_i \\ y_i \end{pmatrix}$$

where ρ is the learning rate of SGD (first order method).



$$g_t = g_0 / (1 + t)$$

Problems with Rosenblatt's Perceptron Learning Algorithm

- ▶ If the classes are linearly separable, the algorithm converges to a separating hyperplane in a finite number of steps **but...**
- ▶ The finitenumber of steps to convergence can be very large. The smaller the gap the longer the time to find it
- ▶ When the data are not separable, the algorithm will not converge, and go in cycles. The cycles can be long and therefore hard to detect.
- ▶ This problem can often be eliminated by overfitting through seeking a hyperplane no in the original space, but in a much enlarged space obtained by creating nonlinear tranformations of original covariates
- ▶ When the data are separable, there are many solutions, and which one is found depends on the starting values
- ▶ This problem can be resolved by means of adding constraints to the separating hyperplanes

Optimal separating hyperplanes

- Consider the optimization problem

$$\max_{\beta, \beta_0, \|\beta\|=1} M : \text{s. t. } y_i \left(x_i^T \beta + \beta_0 \right) \geq M, i = 1, \dots, N$$

where we are maximizing distance to the closest point from either class

- The set of conditions ensure that **all** the points are **at least a signed distance M from the decision boundary** defined by β and β_0 , and we seek the **largest such M** and associated parameters
- We can get rid of the $\|\beta\| = 1$ constraint by replacing the conditions with

$$\frac{1}{\|\beta\|} y_i \left(x_i^T \beta + \beta_0 \right) \geq M,$$

or equivalently

$$y_i \left(x_i^T \beta + \beta_0 \right) \geq M \|\beta\|.$$

- Since for any β and β_0 satisfying these inequalities, any positively scaled multiple satisfies them too, we reparametrize $\|\beta\| = 1/M$:

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 : \text{s. t. } y_i \left(x_i^T \beta + \beta_0 \right) \geq 1, i = 1, \dots, N$$

Lagrange multipliers method for constrained optimization

- ▶ Convex optimization (quadratic criterion with linear inequality constraints)
- ▶ The Lagrange function, to be minimized w.r.t. β and β_0 , is

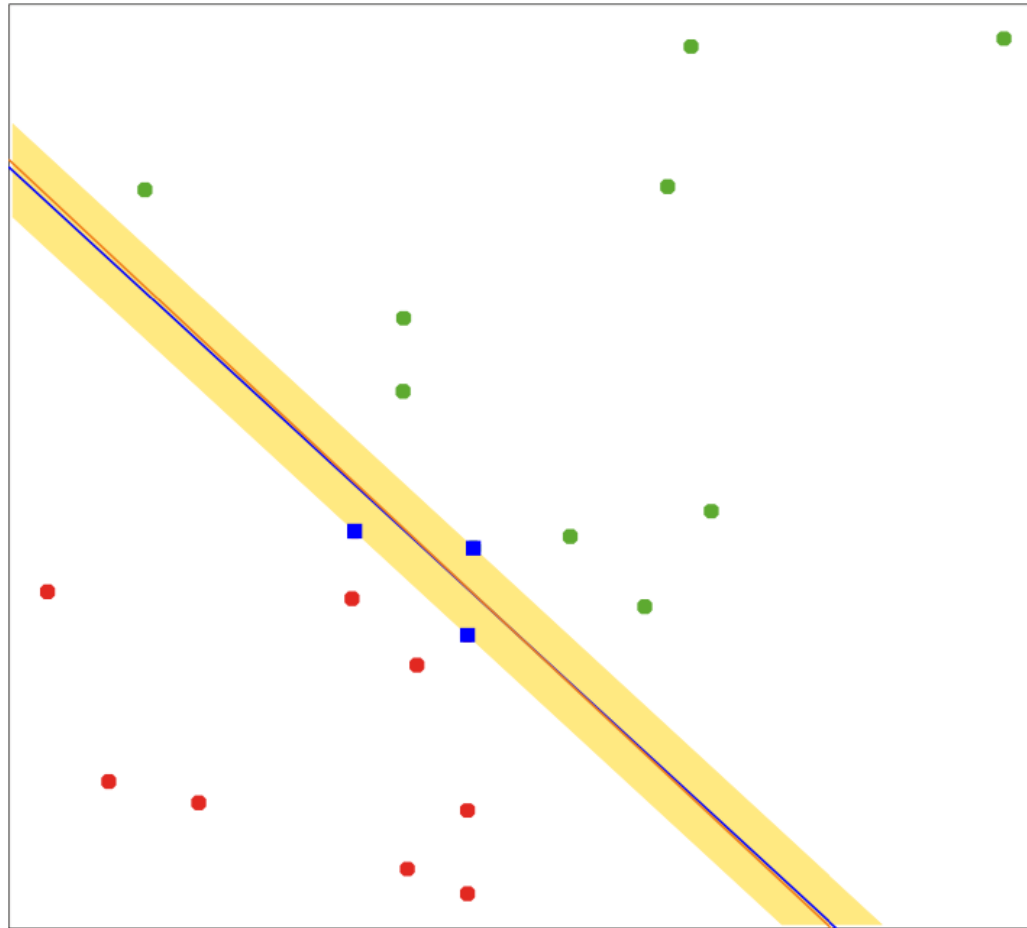
$$L_P = \frac{1}{2} \|\beta\|^2 - \sum_{i=1}^N \alpha_i \left[y_i \left(x_i^T \beta + \beta_0 \right) - 1 \right], \alpha_i y_i > 0, i = 1, \dots, N$$

- ▶ Set the derivatives to zero and obtain:

$$\beta = \sum_{i=1}^N \alpha_i y_i x_i, \text{ and } \sum_{i=1}^N \alpha_i y_i = 0, \text{ and } y_i \left(x_i^T \beta + \beta_0 \right) - 1 = 0, i = 1, \dots, N$$

- ▶ Further resolved by numeric tools
- ▶ From $y_i \left(x_i^T \beta + \beta_0 \right) - 1 = 0, i = 1, \dots, N$ it follows that x_i is on the boundary of the slab

Optimal separating hyperplanes



- ▶ The shaded region delineates the maximum margin separating the two classes.
- ▶ There are three support points indicated, which lie on the boundary of the margin, and the optimal separating hyperplane (blue line) bisects the slab
- ▶ The boundary found using logistic regression (red line) is very close to the optimal separating hyperplane

References

- T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.
- G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.