Linear Classifiers: Discriminants

Prof. Rafal Kustra

Public Health Sciences

October 12, 2016

Binary classification

- Simplest but very popular case: each object comes from one of two classes
- For convienience, lets code them $y_i \in \{0, 1\}$
- Remember that general regression problem was a conditional expectation:

$$\mathrm{E}(y|\boldsymbol{x})$$

which here becomes:

Posterior probability

$$\mathrm{E}(Y|\boldsymbol{x}) = \mathrm{P}(Y = 1|\boldsymbol{x})$$

- In multiclass problems, many classifiers return an estimate of $\mathrm{P}(y = k|\boldsymbol{x})$ rather than just class y.
- These posterior probabilities are central from decision theory point of view

Linear Classifiers

- As in linear regression we start with the linear predictor in $\boldsymbol{x}$:

$$\theta(\boldsymbol{x}) = \sum_j \beta_j x_j$$

- In regression case this is used to approximate the conditional expectation directly, i.e.

$$\widehat{y}(\boldsymbol{x}) = \mathsf{E}(y|\boldsymbol{x}) = \theta(\boldsymbol{x})$$

- At least in binary case, this could also work in classification and would approximate posterior probability
- But direct LS approach is inconvienient:
  1. $\theta(\boldsymbol{x})$ can easily be outside of $[0, 1]$ interval
  2. minimizing square distance between $\theta(x)$ and $y = (0, 1)$ means far away points even on a good side can badly influence $\widehat{\beta}$ estimate.

OLS for K-class Classification

- For two classes, one could code response y as 0/1 and run OLS
- This would estimate $\mathrm{P}(Y = 1|\boldsymbol{x})$. It is also equivalent to setting up *two* responses, y_1, y_2 (two columns of now-matrix, Y) each being an indicator for it's class. Solution β (a $p \times 2$ matrix) is:

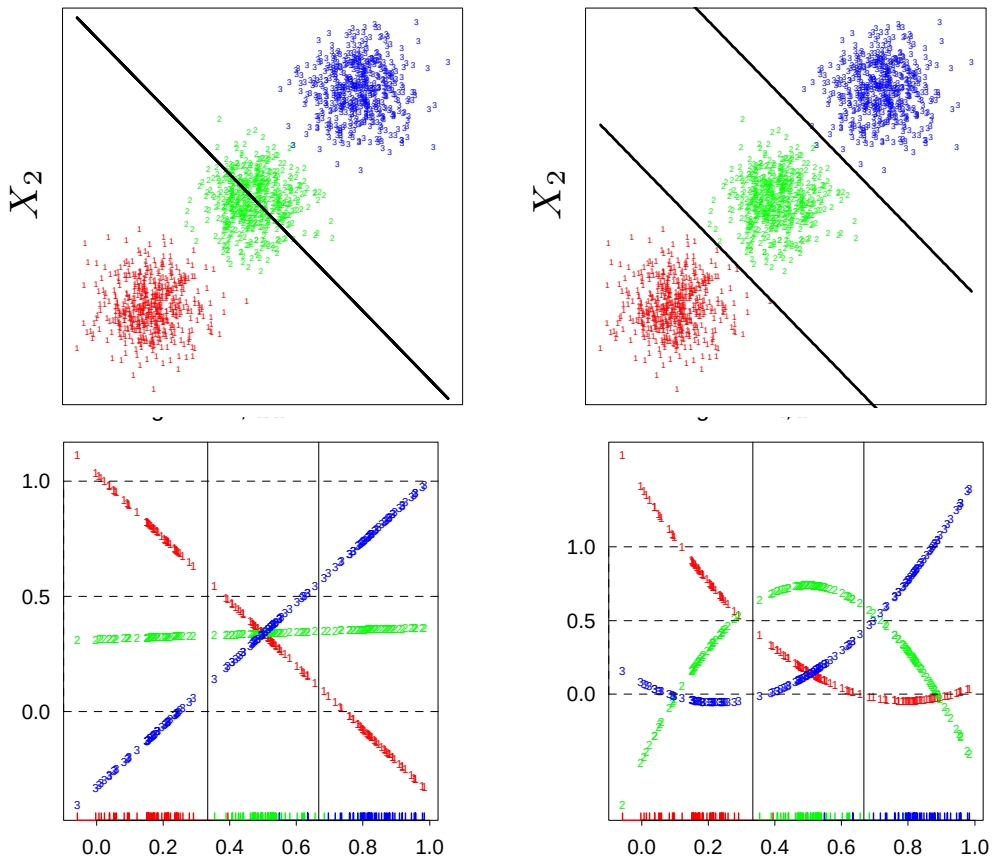
$$\widehat{\beta} = (X'X)^{-1}X'Y$$

  with corresponding two columns of $\widehat{Y}$
- The two columns of $\widehat{Y}$ are just reflections of each other (check!). Remember contrasts.
- One could extend this approach to K-class, coding each as binary column of Y. This gives you K columns of $\widehat{Y}$, each as an estimate of $\mathrm{P}(Y = k|\boldsymbol{x})$
- For $K > 2$ we have a potential of *masking* as a problem with this approach (in addition to inproper range of $\widehat{Y}$)

Classifier Boundary

- In p-dim space of predictors, a set of points ('curve') where classifier switches from predicting class A to class B
- For Bayes classifier: a set of points where posterior probabilities for two (or more) classes are equal: predicting either of these classes leades to same expected cost
- For OLS classifier above: set of points where $\widehat{y}_1 = \widehat{y}_2$ for class 1 and 2 (i.e., fitted values corresponding to column 1 or 2 of response *matrix* Y)

Masking Geometry

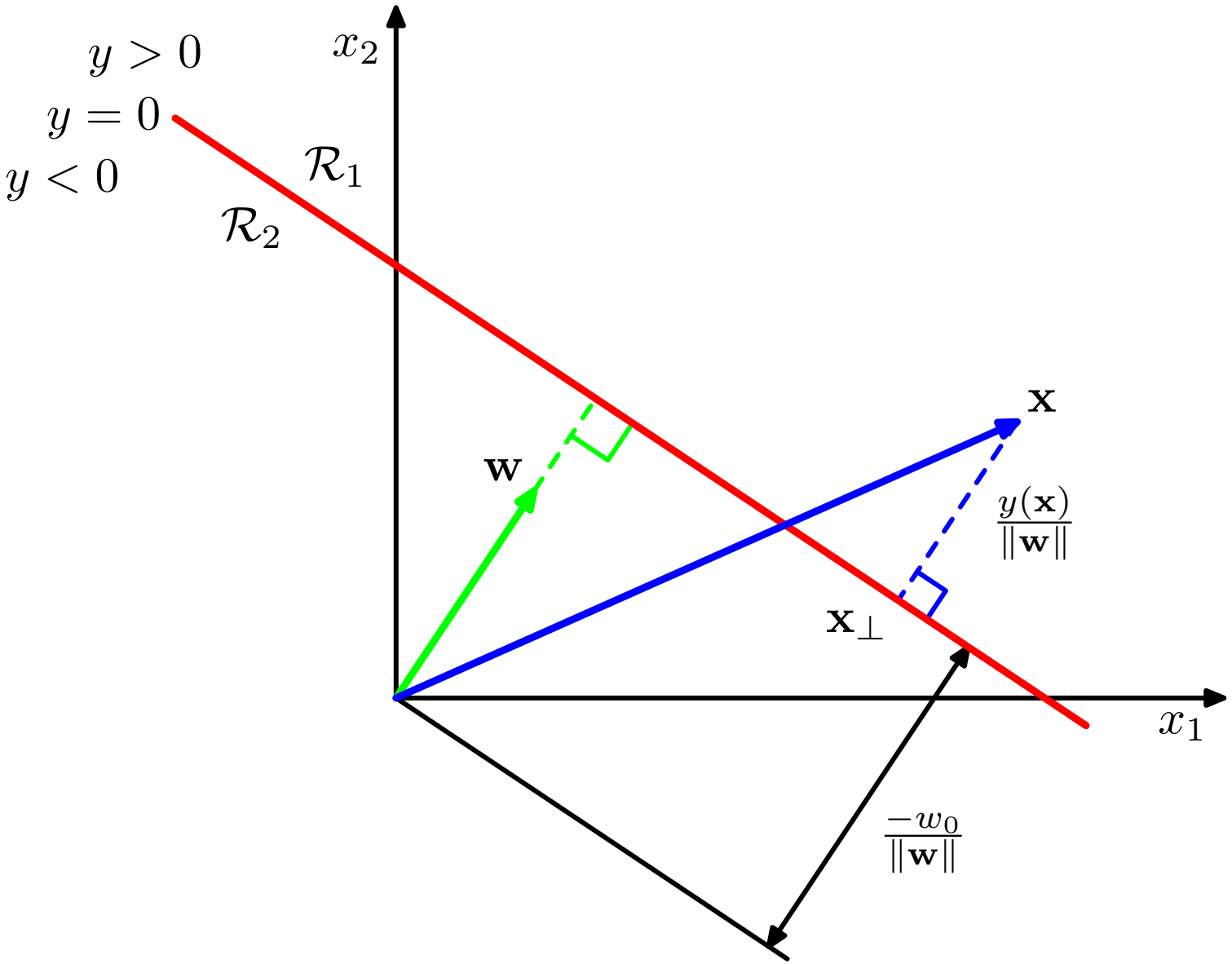


Linear Discriminants

- One way to deploy linear functions of $\boldsymbol{x}$ is through *discriminant* functions
- We will seek $\widehat{\beta}$ that predicts the class of object $\boldsymbol{x}$ by the sign of $\theta(\boldsymbol{x})$, i.e. in binary case, we predict class $y = 1$, if $\theta(\boldsymbol{x}) > 0$ and class $y = 0$ otherwise
- This creates a linear boundary, $\mathcal{D}$ characterized by all points $\boldsymbol{x}$ such that $\theta(\boldsymbol{x}) = 0$
- As it stands boundary $\mathcal{D}$ passes through the origin. Adding an intercept term we remedy that:

$$\theta(\boldsymbol{x}) = \alpha + \beta^T \boldsymbol{x}$$

- In this formulation one can see that α locates a surface $\alpha/||\beta||$ units away from origin
- Vector β is a normal to the decision hyperplane: we can think of classification procedure as projecting points $\boldsymbol{x}$ on (extension of) β and checking if they lie to the left or right of $\alpha/||\beta||$


$y > 0$
$y = 0$
$y < 0$
x_2
$\mathcal{R}_1$
$\mathcal{R}_2$
$\mathbf{x}$
$\mathbf{w}$
$\frac{y(\mathbf{x})}{\|\mathbf{w}\|}$
$\mathbf{x}_\perp$
x_1
$\frac{-w_0}{\|\mathbf{w}\|}$

Fisher Discriminant Analysis for 2 classes

- In 1936, Fisher proposed finding a discriminant direction, β such that when points from two classes were projected onto the line two things happen:
  1. The class means would be well separated
  2. The spread within classes was small
- This leads to finding vector β that maximizes the spread but also minimizes the within-class variance. For K classes spread of means can be measured by *between-class* variance:

$$B = \sum_{k=1}^{2} (\overline{\boldsymbol{x}}_k - \overline{\boldsymbol{x}}) (\overline{\boldsymbol{x}}_k - \overline{\boldsymbol{x}})^T = \frac{1}{2} (\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_2)(\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_2)^T$$

- The common within-class covariance matrix can be estimated by:

$$W = (N - K)^{-1} \sum_i \left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{k(i)}\right) \left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{k(i)}\right)^T$$

- The direction of the line onto which projected data would maximize the spread of class means while minimizing the within-class spread is $\boldsymbol{a}$:

$$\boldsymbol{a} = \underset{\alpha}{\text{argmax}} \frac{\alpha^T B \alpha}{\alpha^T W \alpha}$$

- This direction is one which is a compromise between the line that is perpendicular to the line joining the two class means, and the line which is perpendicular to the principal axis of the *pooled* covariance structure
- Another way to look at it is to assume that the points in different classes have different means but the same covariance matrix, Σ
- Then the variance of an observation, $\boldsymbol{x}_i$ projected on any vector, α, is:

$$\text{Var}(\boldsymbol{x}_i^T \alpha) = \alpha^T \Sigma \alpha$$

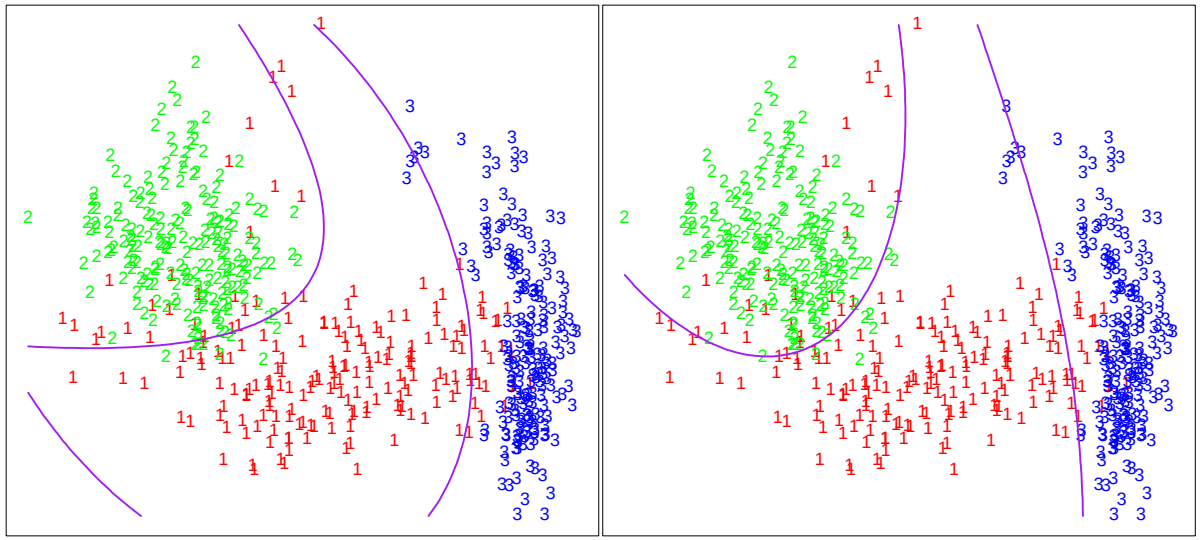


Figure 4.6: *Two methods for fitting quadratic boundaries. The left plot shows the quadratic decision bound-*

- The within-class covariance matrix, W, is a standard estimate of the common covariance of points with different means. Hence the denominator of the Fisher criterion is an estimate of the variance of an observation projected on vector, α
- It is easy to check the the solution to FDA criterion is:

$$\boldsymbol{a} \propto W^{-1}(\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_2)$$

  and is usually normalized so that $\boldsymbol{a}^T W \boldsymbol{a} = 1$
- Classification is done by measuring the distance of a new point, $\boldsymbol{x}_0$, to the class means *in the projected space* (here a line) and choosing the class with the closest mean, ie:

$$C(\boldsymbol{x}_0) = \underset{k}{\operatorname{argmin}}\, \boldsymbol{x}_0^T \boldsymbol{a} - \overline{\boldsymbol{x}}_k^T \boldsymbol{a} = \underset{k}{\operatorname{argmin}}\, \boldsymbol{a}^T(\boldsymbol{x}_0 - \overline{\boldsymbol{x}}_k)$$

LDA for more classes

- With two classes there is one hyperplane separating the classes in a linear classifier. With K classes one needs $K-1$ hyperplanes
- Another way to look at this: the Fisher criterion:

$$\frac{\alpha^T B \alpha}{\alpha^T W \alpha}$$

has rank $K-1$, because of the numerator (unless the dimensionality of the data is smaller than number of classes). JN Rao in 1956 extended the FDA to LDA for multiple classes, by finding up to $K-1$ directions, $\boldsymbol{a}_k$, starting with the maximum (Fisher) one, $\boldsymbol{a}_1$ and subsequent ones, $\boldsymbol{a}_k$, for $k = 2, \ldots, K-1$ subject to:

$$\boldsymbol{a}_k^T W \boldsymbol{a}_j, \quad j = 1, \ldots, k-1$$

This is a kind-of orthogonality constraint in the metric of pooled covariance structure

- Classification of new point, $\boldsymbol{x}_0$ again happens by first projecting the new point and class means onto the space spanned by discriminant directions, $\boldsymbol{a}_k$, and finding the closest class mean to the new point using Euclidean distance. Let $A = [\boldsymbol{a}_1\ \boldsymbol{a}_2\ \cdots\ \boldsymbol{a}_{K-1}]$ be a matrix with $K-1$ columns being discriminant directions. Then:

$$C(\boldsymbol{x}_0) = \underset{k}{\operatorname{argmin}} \, ||A(\boldsymbol{x}_0 - \overline{\boldsymbol{x}}_k)||$$

Classification via Maximum Likelihood

- Lets enter a Gaussian world:

$$f_k(\boldsymbol{x}) = N_p(\mu_k, \Sigma)$$

  so that class-densities are Gaussian with *the same* covariance matrix, Σ and class-specific means
- The joint log-likelihood of data separates:

$$-1/2p\log[(2\pi)|\Sigma|]\sum_i \left\{(\boldsymbol{x}_i - \boldsymbol{\mu}_{k(i)})^T\Sigma^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_{k(i)})\right\}$$

- The ML estimator (or Bayes estimator, under the same prior probabilities) assigns $\boldsymbol{x}_0$ to that class, k, that has maximum likelihood for $\boldsymbol{x}_0$, or equivalently *minimum Mahalonobis distance* to class mean:

$$\boldsymbol{x}_0 - \boldsymbol{\mu}_k)^T\Sigma^{-1}(\boldsymbol{x}_0 - \boldsymbol{\mu}_k$$

- One inserts the MLE estimates of μ_k, Σ in above
- It is an easy theorem to show that minimum Mahalonibis distance is same as Euclidean distance in the space of LDA directions

Logistic Regression

- In LDA with Gaussian assumption we assumed a distribution for the $\boldsymbol{X}$ vector of inputs.
- In Logistic regression we go straight after the posterior, $P(y|\boldsymbol{x})$. For *binary classification* code two classes of y as $(0, 1)$. We need a link between $\mathrm{E}(y|x) = \mathrm{P}(y = 1|x)$ and x.
- Logit link is most popular:

$$P(y = 1|\boldsymbol{x}; \beta) = \frac{exp(\boldsymbol{x}^T \beta)}{1 + \exp(\boldsymbol{x}^T \beta)}$$

- This leads to the inverse function:

$$\log\left(\frac{\mathrm{P}(y = 1|\boldsymbol{x}; \beta)}{1 - \mathrm{P}(y = 1|\boldsymbol{x}; \beta)}\right) = \boldsymbol{x}^T \beta = \sum_j \beta_j x_j$$

the LHS is frequently called *log-odds* at $\boldsymbol{x}$

Parameter estimation

- We have a model for predicting posterior probabilities given any $\boldsymbol{x}$ input that depends on parameter vector, β
- Maximum-likelihood estimate of β starts with forming a log-likelihood function
- The model is conditional on $\boldsymbol{x}$: the only randomness is in y which is binary
- Given a training set with y_i, $i = 1, \ldots, N$ where each $y_i \in \{0, 1\}$ we have N Bernoulli trials, each with probability, $p(\boldsymbol{x}_i)$. We can write down a likelihood as follows:

$$\prod_i p(\boldsymbol{x}_i)^{y_i}(1 - p(\boldsymbol{x}_i))^{1-y_i}$$

which leads to log-likelihood:

$$\mathcal{L}(\beta) = \sum_i y_i \log(p(\boldsymbol{x}_i))(1 - y_i) \log(1 - p(\boldsymbol{x}_i))$$

- or:

$$\sum_i y_i \log\left(\frac{p(\boldsymbol{x}_i)}{1-p(\boldsymbol{x}_i)}\right) + \log\left(1-p(\boldsymbol{x}_i)\right)$$

- In terms of β this becomes:

$$\sum_i \sum_j y_i x_{ij} \beta_j - \sum_i \log\left(1+\exp(\sum_j x_{ij}\beta_j)\right)$$

- MLE estimates of β are of course obtained by maximizing the log-likelihood. As this is a non-linear function, iterative process is used. Normally a slightly modified version of well-known Newton-Raphson function minimizer, called Iteratively Reweighted Least Squares is used. Book has the details in Ch. 4.3.3.

Boundary regions for Logistic regression

- So far we have a model for posterior probabilities. To classify as usual we predict the class with maximum posterior probability
- Even though the probabilities are not linear in $\boldsymbol{x}$ the boundary between the two classes is.
- The boundary is defined as a set of $\boldsymbol{x}$ points where posterior probabilities are equal: here 0.5 each
- This is equivalent to the odds ratio being 1:

$$p(\boldsymbol{x}) = 1 - p(\boldsymbol{x}) \iff \frac{p(\boldsymbol{x})}{1 - p(\boldsymbol{x})} = 1 \iff \text{logit}(p(\boldsymbol{x})) = 0$$

which in logistic regression means set of $\boldsymbol{x}$ such that:

$$\boldsymbol{x}^T \widehat{\beta} = 0$$

which of course is linear in $\boldsymbol{x}$

Comparison with Gaussian LDA

- LDA starts with a Gaussian assumption on class-densities, f_k (for binary case, $k = 1, 2$):

$$f_k(\boldsymbol{x}) \equiv f_{X|k}(\boldsymbol{x}; \Sigma, \boldsymbol{\mu}_k) \propto |\Sigma|^{-p/2} \exp\left\{-(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k)\right\}$$

- With 0-1 loss, we classify by highest posterior probability, which in this case is:

$$\mathrm{P}(y = k|\boldsymbol{x}) = \frac{f_k(\boldsymbol{x}; \Sigma, \mu_k)\pi_k}{f_X(\boldsymbol{x})}$$

- Since denominator does not depend on class k it is not relevant for classification. Hence for binary case the boundary region becomes:

$$\begin{aligned} \log \frac{P(y = 1|\boldsymbol{x})}{P(y = 0|\boldsymbol{x})} &= (\boldsymbol{x} - \boldsymbol{\mu}_0)^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_0) \\ &\quad -(\boldsymbol{x} - \boldsymbol{\mu}_1)^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_1) + \log \frac{\pi_1}{\pi_0} \end{aligned}$$

which is also linear in $\boldsymbol{x}$ since $\boldsymbol{x}^T \Sigma^{-1} \boldsymbol{x}$ term cancels out

- So we have two binary procedures with quite different assumptions producing linear classification boundaries
- One crucial difference is that LDA is forced to model the whole distribution of the x space. For high-dimension problems this can be very difficult and wasteful
  1. Normality assumption may be quite wrong
  2. Even if normality is plausible, equal-covariance assumption may not be
  3. Number of parameters to estimate could be enormous: $O(p^2))$ rather than $O(p)$ that logistic regression requires
- Logistic on the other hand goes straight after posterior and can a much better choice for most problems. However:
  1. Having distributional framework for X space can be beneficial when extending the model
  2. IF we can get the distribution of X right, we can have better prediction in X different from ones in the training set

Polychotomous Logistic regression

- Quite easy to extend logistic regression to K classes
- First need to choose a reference class. This is arbitrary in a sense that model predictions will not change (but interpretation of parameters will).
- With K classes, lets code y as:

$$y_i \in (0, \ldots, K-1)$$

- As before, we will model log-odds ratio as a linear function of $\boldsymbol{x}$. Reference class used for denominator. So, for class k, $k = 1, \ldots, K-1$ we have:

$$\log \frac{\mathrm{P}(y = k|\boldsymbol{x})}{\mathrm{P}(y = 0|\boldsymbol{x})} = \boldsymbol{x}^T \beta_k$$

- Of course all those probabilities have to sum up to one across classes, which leads to inverse functions:

$$\mathrm{P}(y = k|\boldsymbol{x}; \beta) = \frac{\exp(\boldsymbol{x}^T \beta_k)}{1 + \sum_{\kappa=1}^{K-1} \exp(\boldsymbol{x}^T \beta_\kappa)}; \qquad k = 1, \ldots, K-1$$

- and for the reference class ($k = 0$):

$$\mathrm{P}(y = k|\boldsymbol{x}; \beta) \frac{1}{1 + \sum_{\kappa=1}^{K-1} \exp(\boldsymbol{x}^T \beta_\kappa)}$$

- Hence for K classes there are $K - 1$ vectors of parameters, β_k defining $K - 1$ boundaries.
- Estimation is similar as in binary case for logistic regression. The Hessian matrix is block-diagonal but the weight matrix is not which leads to slightly more complicated algorithm.