Lecture 11: Classification

Jing Ma, Statistics, TAMU

13 November, 2019

Recap

- Linear regression
- Training and test error
- ANOVA

This lecture

- Classification
  - Logistic regression
  - Linear Discriminant Analysis

Classification

- Regression involves predicting a continuous-valued response, like tumor size.

Classification

- Regression involves predicting a continuous-valued response, like tumor size.
- Classification involves predicting a categorical response:
  - Cancer versus Normal
  - Tumor Type 1 versus Tumor Type 2 versus Tumor Type 3

Classification

- Regression involves predicting a continuous-valued response, like tumor size.
- Classification involves predicting a categorical response:
  - Cancer versus Normal
  - Tumor Type 1 versus Tumor Type 2 versus Tumor Type 3
- Classification problems tend to occur even more frequently than regression problems in the analysis of biomedical data.

Classification

- Regression involves predicting a continuous-valued response, like tumor size.
- Classification involves predicting a categorical response:
  - Cancer versus Normal
  - Tumor Type 1 versus Tumor Type 2 versus Tumor Type 3
- Classification problems tend to occur even more frequently than regression problems in the analysis of biomedical data.
- Just like regression,
  - Classification cannot be blindly performed in high-dimensions because you will get zero training error but awful test error;
  - Properly estimating the test error is crucial.

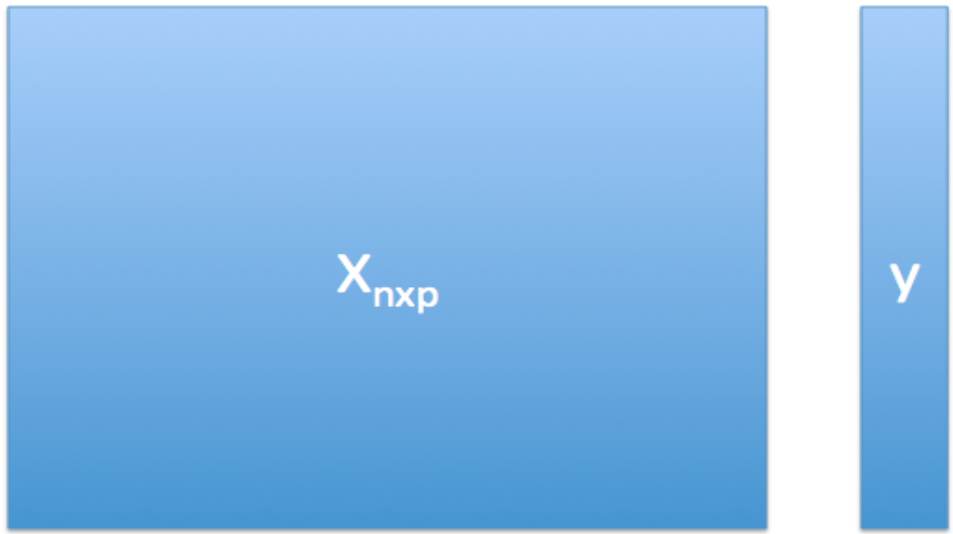
Classification

There are many approaches out there for performing classification.

We will mainly discuss logistic regression and LDA.

The classification task

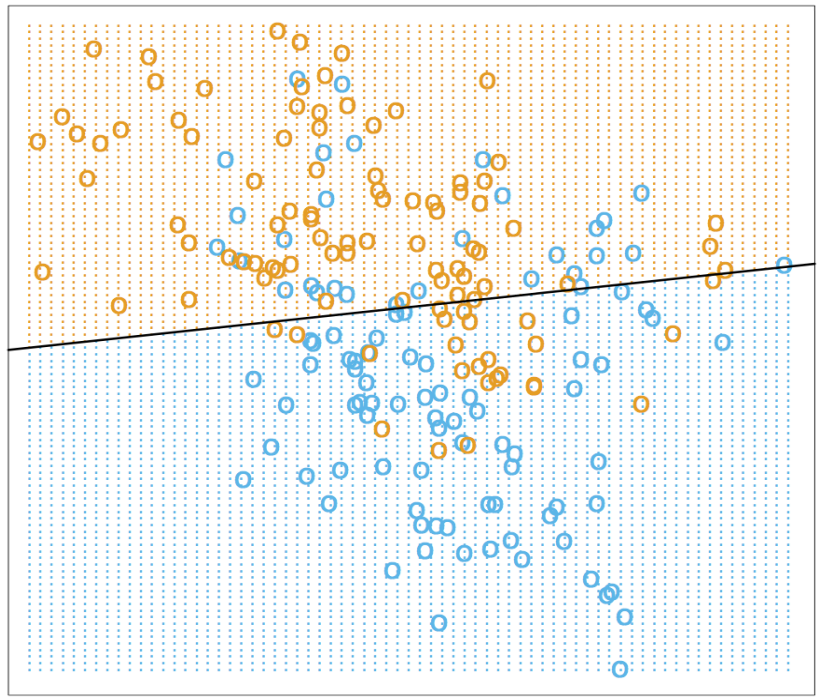
- Similar to regression, the classification problem is supervised learning:



- The only difference is that the response y is a categorical variable with (in general) K categories.
- We mostly focus on the case of $K = 2$, i.e. cancer vs benign, but the ideas are the same.

Classification using linear regression

- There is really nothing preventing us from doing this: we can fit a linear regression with a categorical response!



Classification using linear regression

- Consider the simple case of $p = 1$, a single predictor.
- Suppose that y_i can be 1 or -1 (positive or negative) with equal probability.
- In this case, no intercept is needed ($\bar{y} = 0$) so the linear regression tries to find β that minimizes the RSS:

$$\|y - x\beta\|_2^2 = \sum_i (y_i - \beta x_i)^2$$

- As we discussed before,

$$x_i\hat{\beta} = \hat{y}_i$$

- In this case, we set

$$C_i = \begin{cases} 1 & x_i\hat{\beta} > 0 \\ -1 & x_i\hat{\beta} \leq 0 \end{cases}$$

Classification using linear regression

- This model assumes that the two classes can be separated with a line, which is somewhat unrealistic!
- Suppose $y_i = 1$
  - Suppose $\hat{y}_i = 0.1$; then $(y_i - \hat{y}_i) = 0.9$

Classification using linear regression

- This model assumes that the two classes can be separated with a line, which is somewhat unrealistic!
- Suppose $y_i = 1$
  - Suppose $\hat{y}_i = 0.1$; then $(y_i - \hat{y}_i) = 0.9$
  - On the other hand, if $\hat{y}_i = -0.1$, $(y_i - \hat{y}_i) = 1.1$

  These are not very different!

Classification using linear regression

- This model assumes that the two classes can be separated with a line, which is somewhat unrealistic!
- Suppose $y_i = 1$
  - Suppose $\hat{y}_i = 0.1$; then $(y_i - \hat{y}_i) = 0.9$
  - On the other hand, if $\hat{y}_i = -0.1$, $(y_i - \hat{y}_i) = 1.1$

  These are not very different!
- However, in the first case, $C_i = 1$ and in the second case, $C_i = -1$

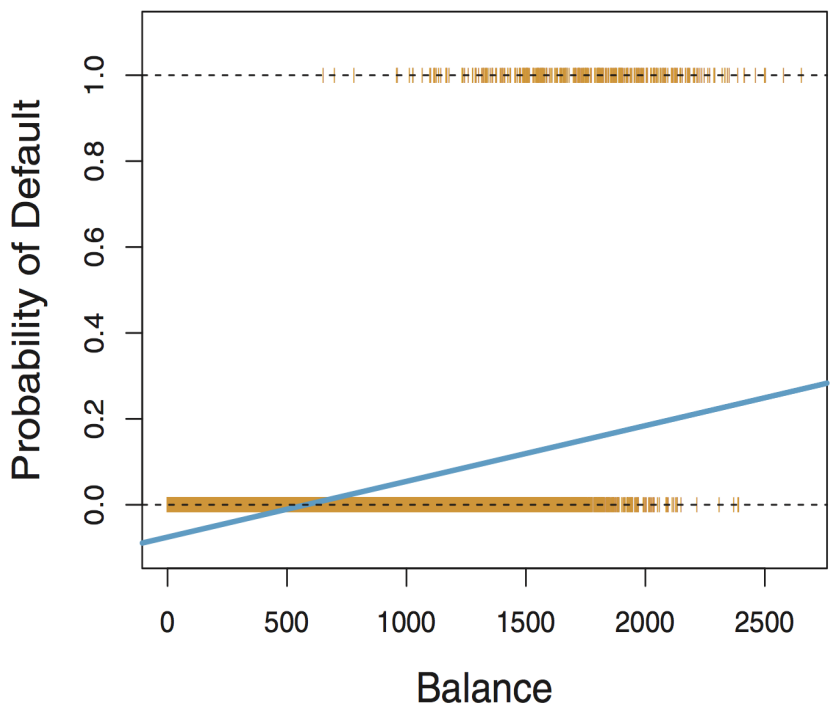
Classification using linear regression

- This model assumes that the two classes can be separated with a line, which is somewhat unrealistic!
- Suppose $y_i = 1$
  - Suppose $\hat{y}_i = 0.1$; then $(y_i - \hat{y}_i) = 0.9$
  - On the other hand, if $\hat{y}_i = -0.1$, $(y_i - \hat{y}_i) = 1.1$

  These are not very different!
- However, in the first case, $C_i = 1$ and in the second case, $C_i = -1$
- This suggest that so *sum of squared errors may not be the best loss function for categorical variables*!

Drawbacks of linear regression for classification

- If we code the values of y as 0 and 1 (instead of -1 and 1), then $X\hat{\beta}$ from linear regression *gives an estimate of the probability* $P(y = 1 \mid X)$, which is sensible.
- However, there is no guarantee that the estimated probabilities are in fact between 0 and 1!! And, in general, they are actually not!



Drawbacks of linear regression for classification

There is also a serious problem if y has more than 2 categories!

- Suppose that we are trying to predict the medical condition of a patient in the emergency room on the basis of her symptoms, and there are three possible diagnoses: stroke, drug overdose, and epileptic seizure
- We could consider a quantitative response as

$$Y = \begin{cases} 1 & \text{if stroke} \\ 2 & \text{if drug overdose} \\ 3 & \text{if epileptic seizure} \end{cases}$$

- Unfortunately, this coding implies an ordering on the outcomes, putting drug overdose in between stroke and epileptic seizure
- In practice there is no particular reason that this needs to be the case and one could choose any other equally reasonable coding

Logistic regression

- Logistic regression is the straightforward extension of linear regression to the classification setting.
- For simplicity, suppose $y \in \{0, 1\}$: a two-class classification problem.
- Logistic regression assumes a parametric model

$$P(y = 1 \mid X) = \frac{\exp(\beta_0 + \beta_1 X)}{1 + \exp(\beta_0 + \beta_1 X)}.$$

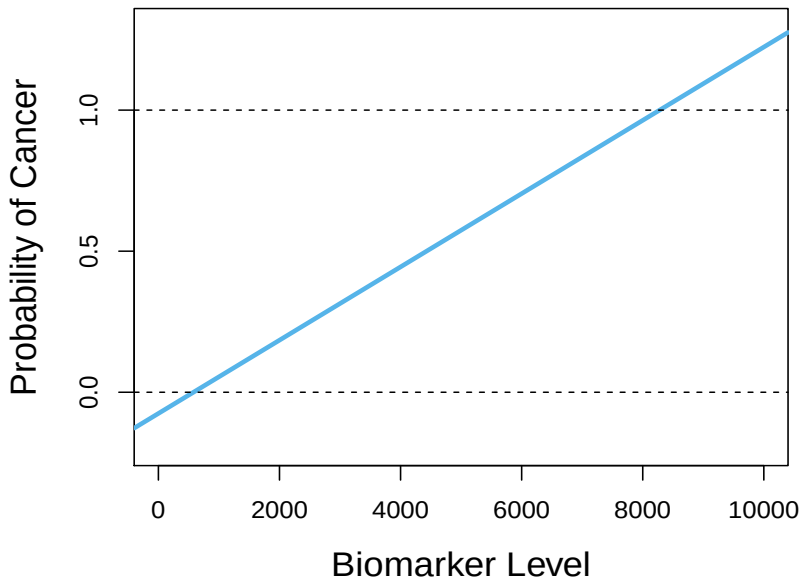
Logistic regression

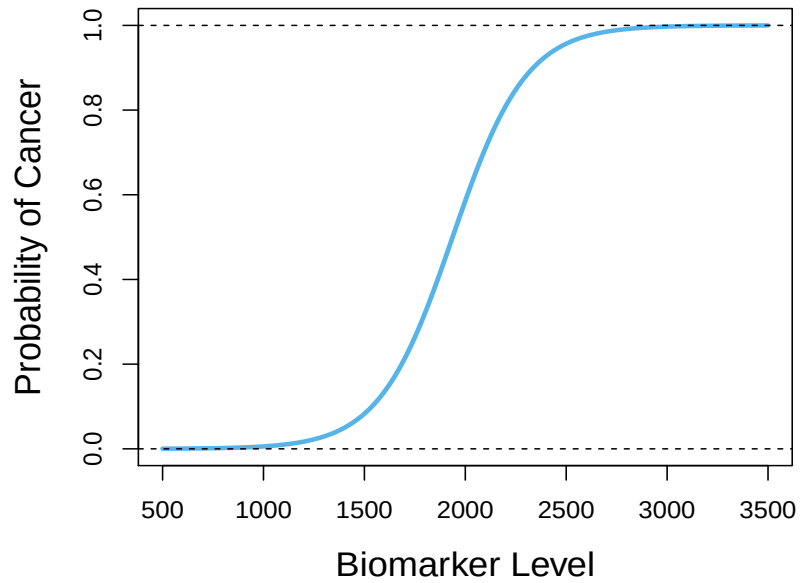
- Taking log and doing some algebra, we can see that

$$\log\left(\frac{P(y=1 \mid X)}{1-P(y=1 \mid X)}\right) = \beta_0 + \beta_1 X$$

- $\log \frac{P(y=1|X)}{1-P(y=1|X)} = \log \frac{P(y=1|X)}{P(y=0|X)}$ is the log-odds, or logit transform
- This means that logistic regression is a *linear model* in the new, transformed domain. These types of models are called generalized linear models.
- We usually fit this model using glm in R.

Logistic vs linear regression





- Left: linear regression.
- Right: logistic regression.

The ALL/AML leukemia data

We will illustrate logistic regression using gene expression data from the leukemia ALL/AML study of Golub et al. (1999). Note there are 27 ALL (code 0) and 11 AML (code 1) patients.

```
require(multtest)
data(golub)
# golub.cl
# data.frame(gene1=golub[68,],cl=golub.cl) %>%
#   mutate(cl=factor(cl)) %>%
#   ggplot(aes(x=cl,y=gene1)) + geom_boxplot()
# teststat = mt.teststat(golub, golub.cl)
```

Logistic regression on the leukemia data

```
mydata = data.frame(gene1=golub[68,],cl=golub.cl) %>%
  mutate(cl=factor(cl))
leukemia_lr = glm(cl~gene1,data=mydata,family = 'binomial')
# summary(leukemia_lr)
```

Logistic regression on the leukemia data

What we get from `summary(leukemia_lr)`:

- Call
- Deviance residuals are a measure of model fit. This part of output shows the distribution of the deviance residuals for individual cases used in the model.
- Coefficients, their standard errors, the z-statistic, and the associated p-value. `gene1` is statistically significant.
- The logistic regression coefficients give the change in the log odds of the outcome for a one unit increase in the predictor variable.
  - For every one unit change in `gene1`, the log odds of class 1 (versus class 0) increases by 3.1335.

Prediction on new data

Suppose we have the expression of gene1 measured on some new patients. Can we predict the type of leukemia (ALL or AML)?

```
set.seed(1)
newdata1 <- with(mydata, data.frame(gene1 = sample(gene1,4)))
# type could also be "link"
newdata1$clP <- predict(leukemia_lr, newdata = newdata1, type = "response")
newdata1
```

```
##      gene1        clP
## 1 -1.28137 0.02783370
## 2 -1.00702 0.06335107
## 3 -1.06221 0.05383201
## 4 -0.57605 0.20698545
```

Linear discriminant analysis

- Recall that the Bayes classifier suggests assigning observation i to class h for which

$$p_h(x) = P(Y = h \mid X = x)$$

  is the largest.

Linear discriminant analysis

- Recall that the Bayes classifier suggests assigning observation i to class h for which

$$p_h(x) = P(Y = h \mid X = x)$$

  is the largest.

- However, as we discussed, calculating $p_h(x)$ is in general difficult!

Linear discriminant analysis

One approach for making this problem easier is to use the Bayes theorem to write

$$p_h(x) = P(Y = h \mid X = x) = \frac{\pi_h \ f_h(x)}{\sum_{l=1}^{H} \pi_l \ f_l(x)}$$

Linear discriminant analysis

One approach for making this problem easier is to use the Bayes theorem to write

$$p_h(x) = P(Y = h \mid X = x) = \frac{\pi_h \ f_h(x)}{\sum_{l=1}^{H} \pi_l \ f_l(x)}$$

Here

- $\pi_h = P(Y = h)$ is the prior probability
- $f_h(x) = P(X = x \mid Y = h)$ is the density function of X for an observation coming from class h
- $p_h(x)$ is the posterior density of y given the data x

Linear discriminant analysis

To use the Bayes theorem

$$p_h(x) = P(Y = h \mid X = x) = \frac{\pi_h f_h(x)}{\sum_{l=1}^{H} \pi_l f_l(x)}$$

we need to estimate π_h and f_h:

Linear discriminant analysis

To use the Bayes theorem

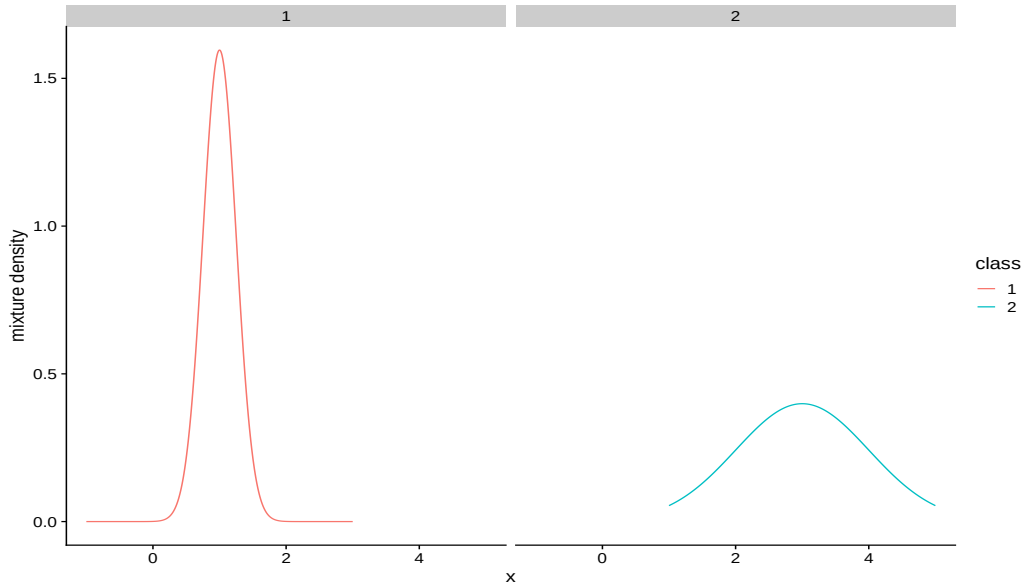
$$p_h(x) = P(Y = h \mid X = x) = \frac{\pi_h f_h(x)}{\sum_{l=1}^{H} \pi_l f_l(x)}$$

we need to estimate π_h and f_h:

- π_h is often easy to estimate: if we have a random sample, $\hat{\pi}_h = 1/n \sum_i I(Y_i = h)$
- However, estimating f_h can be very challenging, especially in high dimensions
- One solution is to assume a parametric form for f_h

Linear discriminant analysis

- In LDA, we assume $f_h(x)$ is the normal density, $N(\mu_h, \sigma_h)$.



- This is equivalent to assuming that our data is generated from a mixture of normal distributions.

$$X \sim \sum_{h=1}^{H} \pi_h \phi(\mu_h, \sigma_h).$$

Linear discriminant analysis

- LDA further assumes that $\sigma_h = \sigma \ \forall h$: all classes share a common variance

Linear discriminant analysis

- LDA further assumes that $\sigma_h = \sigma \ \forall h$: all classes share a common variance
- With this assumption, the decision boundary only depends on means μ_h and is always linear (hence the name)

LDA for $p = 1$

- To start, suppose that $p = 1$

LDA for $p = 1$

- To start, suppose that $p = 1$
- Following the main assumption of LDA, suppose $\sigma_h = \sigma$

LDA for $p = 1$

- To start, suppose that $p = 1$
- Following the main assumption of LDA, suppose $\sigma_h = \sigma$
- We can then show that an observation with covariate x is classified to the class h with the largest value of

$$x\frac{\mu_h}{\sigma^2} - \frac{\mu_h^2}{2\sigma^2} + \log(\pi_h)$$

LDA for $p = 1$

- To start, suppose that $p = 1$
- Following the main assumption of LDA, suppose $\sigma_h = \sigma$
- We can then show that an observation with covariate x is classified to the class h with the largest value of

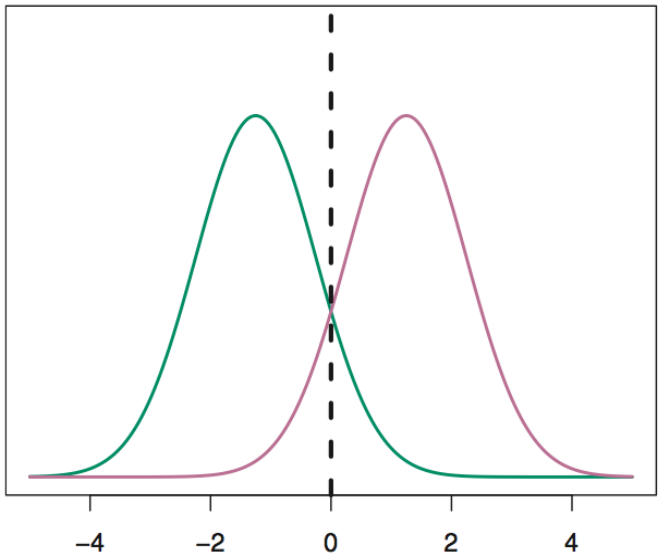
$$x\frac{\mu_h}{\sigma^2} - \frac{\mu_h^2}{2\sigma^2} + \log(\pi_h)$$

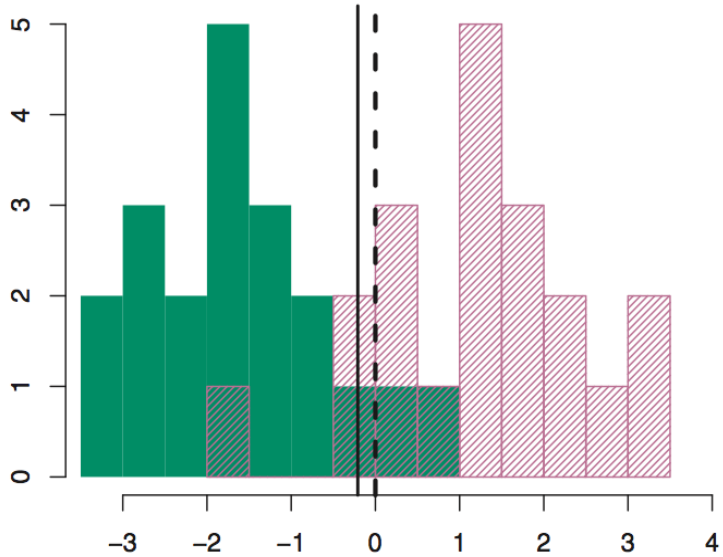
- If we further assume that $H = 2$ and $\pi_1 = \pi_2 = 0.5$, then the decision boundary is

$$\frac{\mu_1 + \mu_2}{2}$$

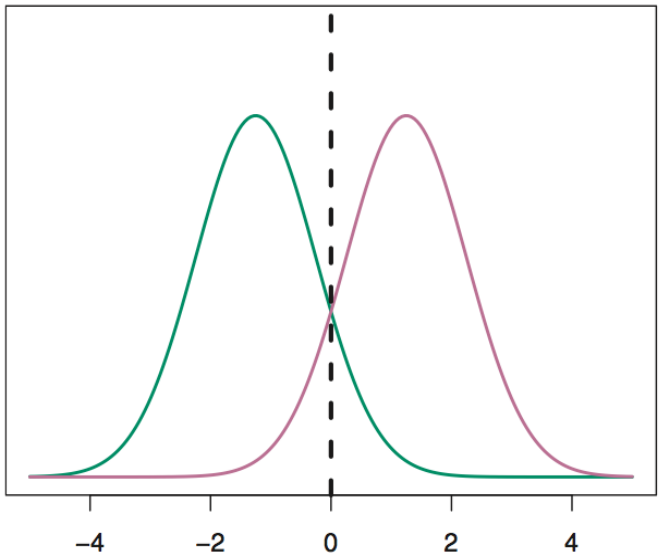
which is clearly **linear**!

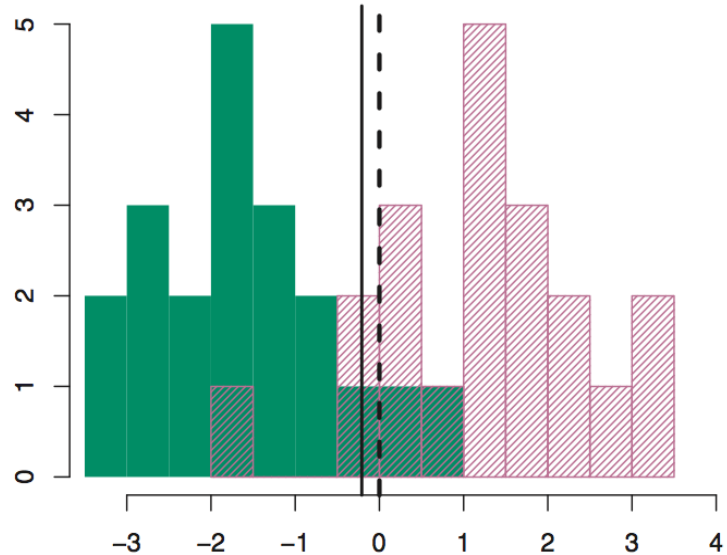
LDA for $p = 1$





LDA for $p = 1$

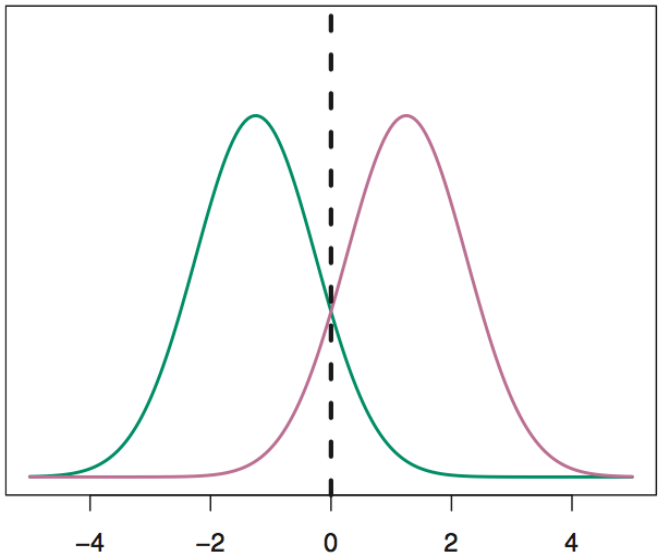


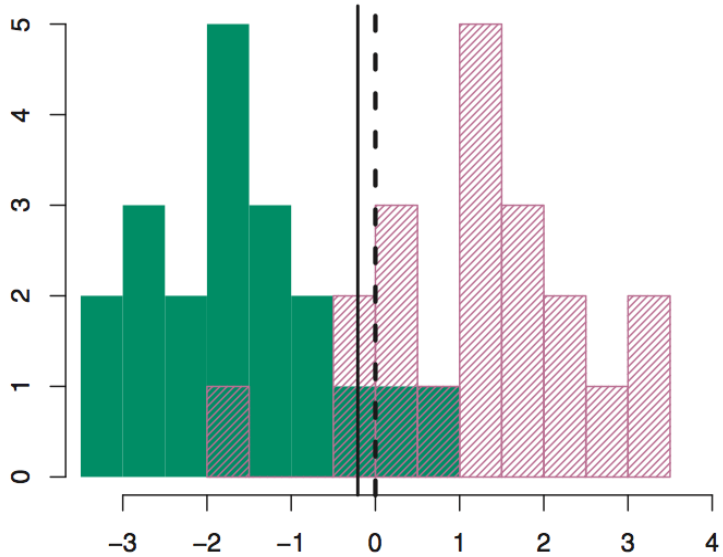


- To make this work, we need to estimate the parameters. The ML estimates are given by $\hat{\pi}_h = n_h/n$ and

$$\hat{\mu}_h = \frac{1}{n_h} \sum_{i:y_i=h} x_i \qquad \hat{\sigma}_h^2 = \frac{1}{n-H} \sum_{h=1}^{H} \sum_{i:y_i=h} (x_i - \hat{\mu}_h)^2$$

LDA for $p = 1$





- To make this work, we need to estimate the parameters. The ML estimates are given by $\hat{\pi}_h = n_h/n$ and

$$\hat{\mu}_h = \frac{1}{n_h} \sum_{i:y_i=h} x_i \qquad \hat{\sigma}_h^2 = \frac{1}{n-H} \sum_{h=1}^{H} \sum_{i:y_i=h} (x_i - \hat{\mu}_h)^2$$

- The picture is very similar if $H > 2 \ldots$ or if $p > 1$

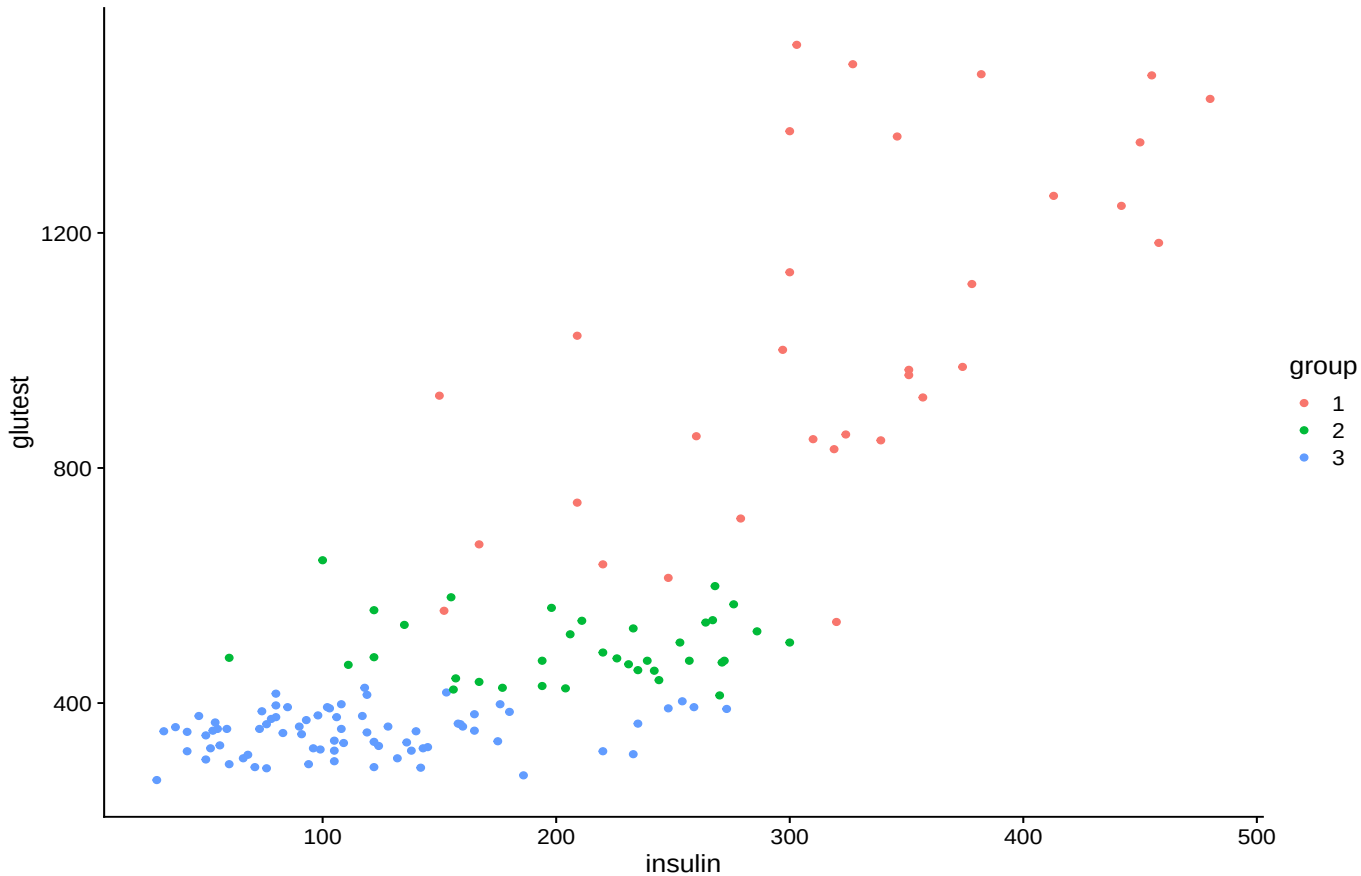
The diabetes data

```
diabetes = read_csv("data/diabetes.csv")
diabetes
```

```
## # A tibble: 144 x 7
##       id relwt glufast glutest steady insulin group
##    <dbl> <dbl>   <dbl>   <dbl>  <dbl>   <dbl> <dbl>
##  1     1  0.81      80     356    124      55     3
##  2     3  0.94     105     319    143     105     3
##  3     5  1         90     323    240     143     3
##  4     7  0.91     100     350    221     119     3
##  5     9  0.99      97     379    142      98     3
##  6    11  0.9       91     353    221      53     3
##  7    13  0.96      78     290    136     142     3
##  8    15  0.74      86     312    208      68     3
##  9    17  1.1       90     364    152      76     3
## 10    19  0.83      85     296    116      60     3
## # ... with 134 more rows
```

LDA on the diabetes data

```
diabetes$group %<>% factor
ggdb = ggplot(mapping = aes(x = insulin, y = glutest)) +
  geom_point(aes(colour = group), data = diabetes)
ggdb
```



LDA on the diabetes data

```
library("MASS")
diabetes_lda = lda(group ~ insulin + glutest, data = diabetes)
# diabetes_lda
```

LDA on the diabetes data

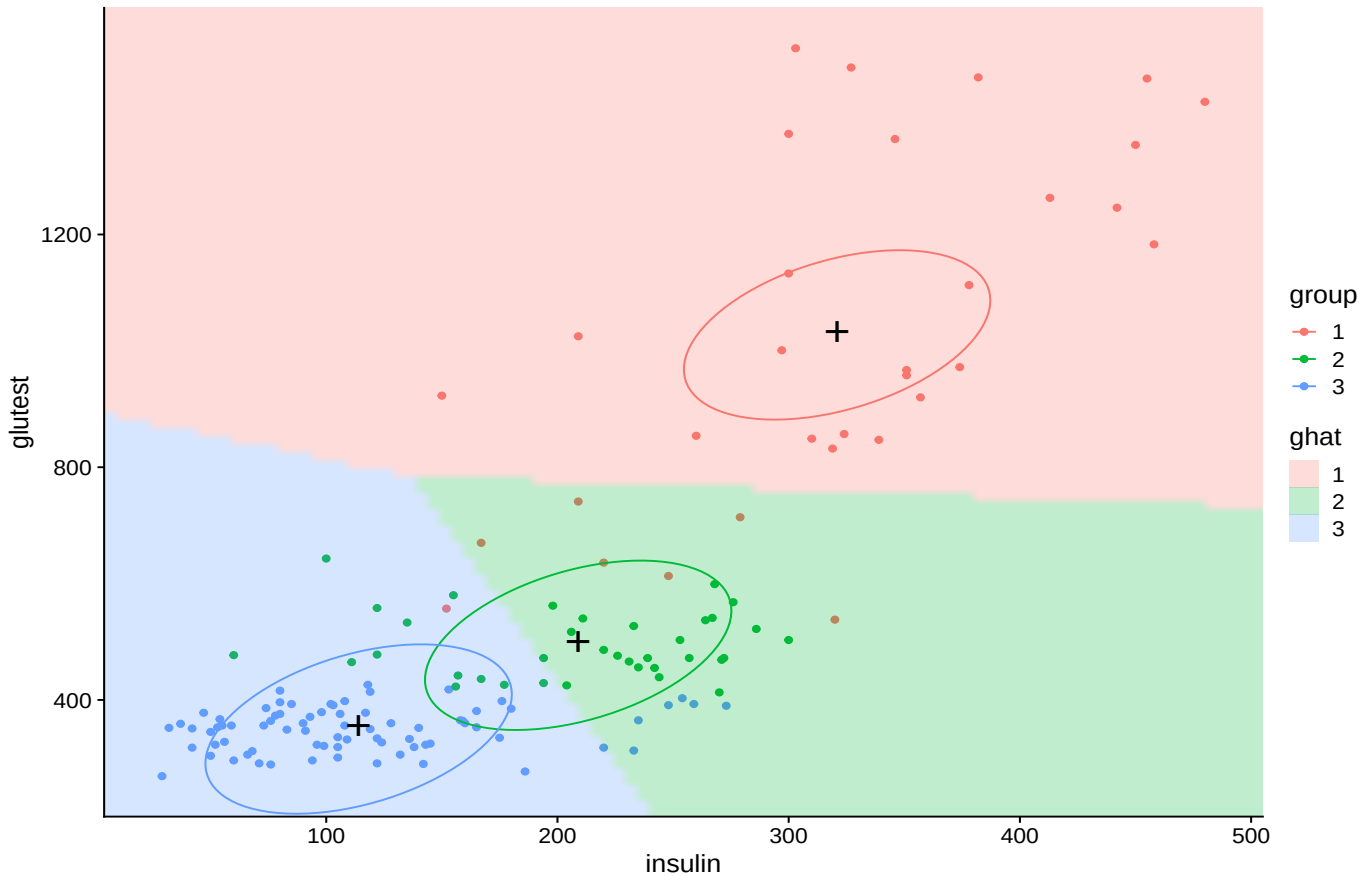
```
ghat = predict(diabetes_lda)$class
table(ghat, diabetes$group)
```

```
##
## ghat  1  2  3
##    1 25  0  0
##    2  6 24  6
##    3  1 12 70
```

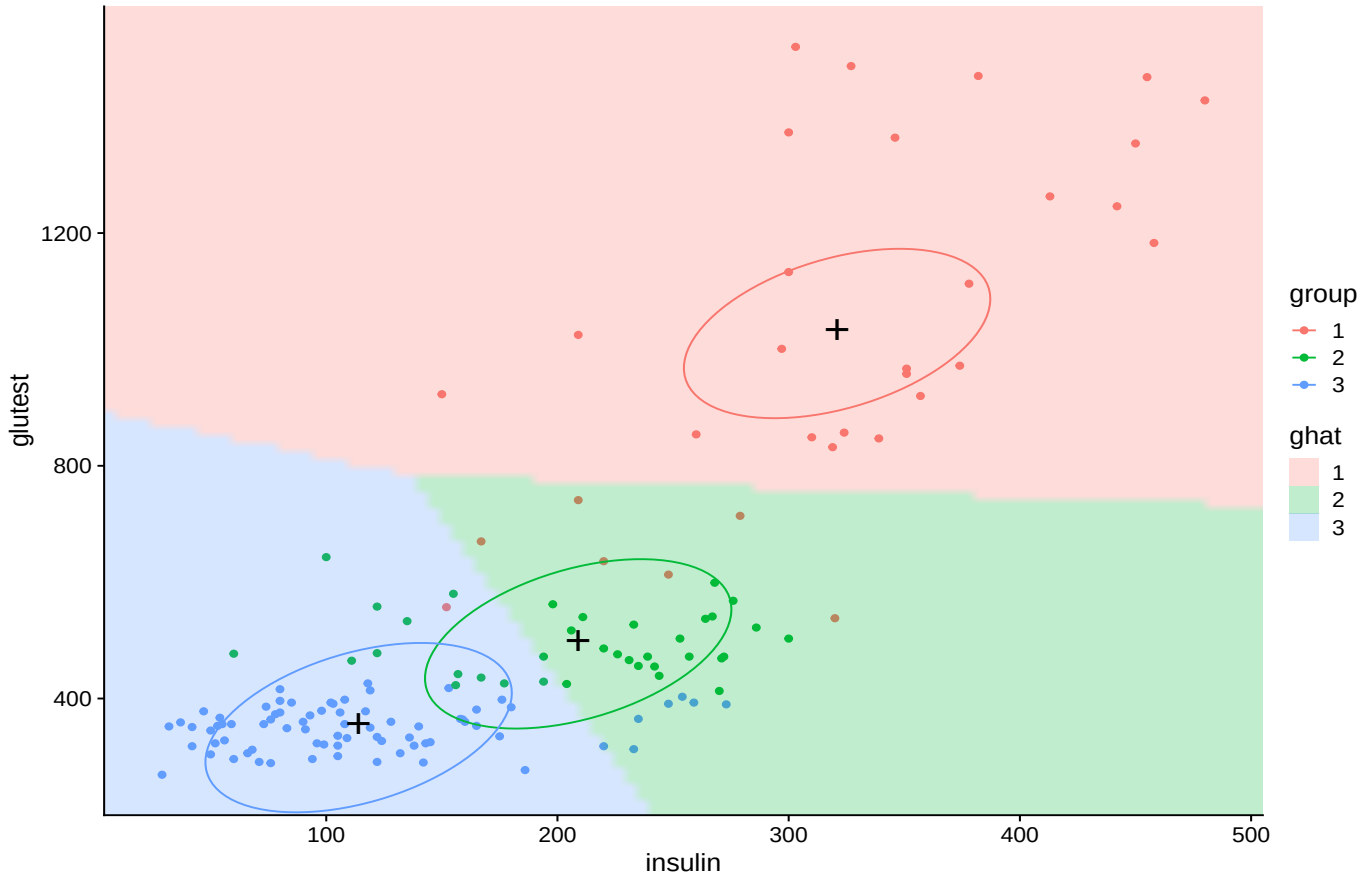
```
mean(ghat != diabetes$group)
```

```
## [1] 0.1736111
```

LDA on the diabetes data

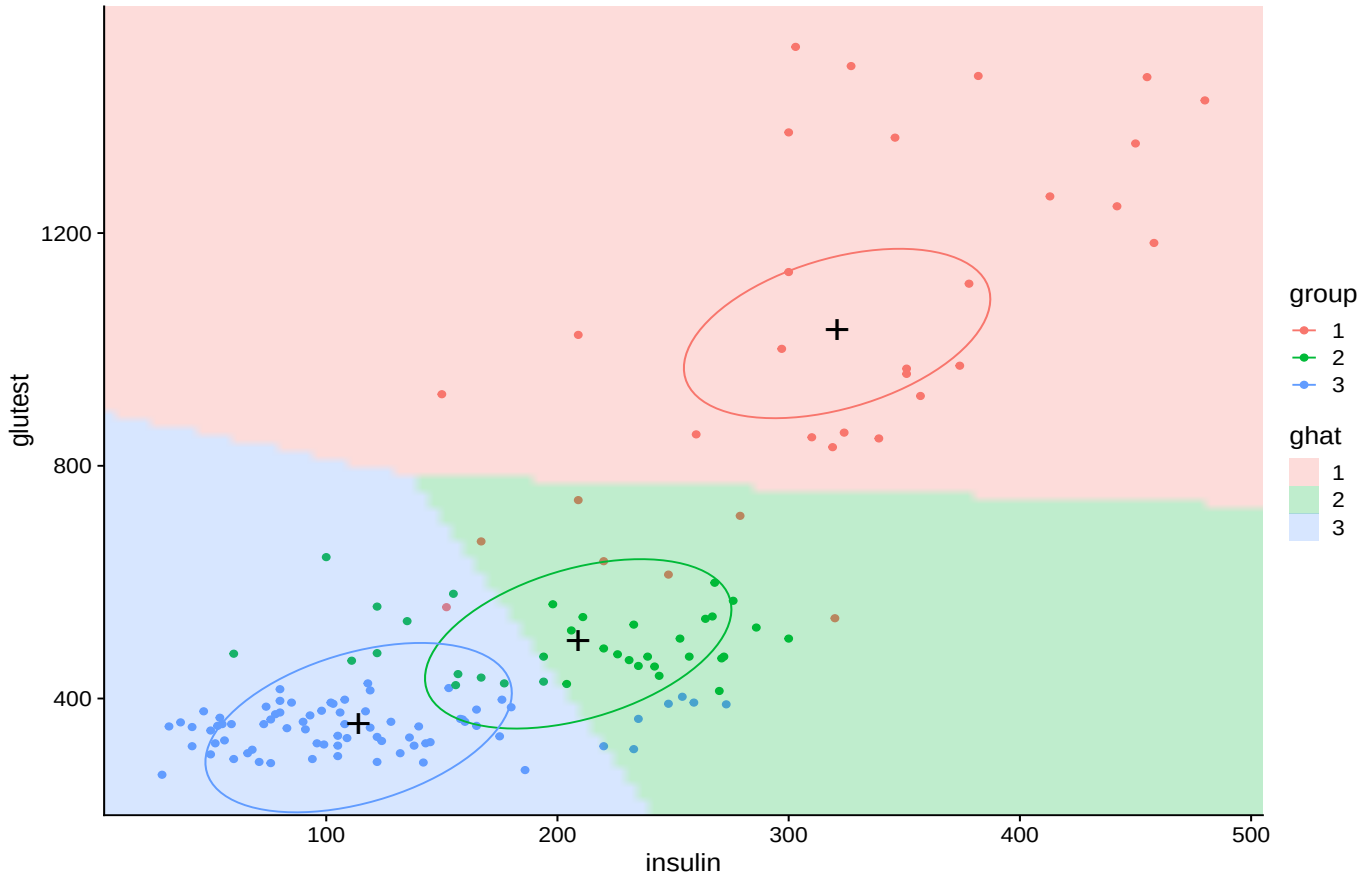


LDA on the diabetes data



Why is the boundary between the prediction regions for group 1 and 2 not perpendicular to the line between the cluster centers?

LDA on the diabetes data



How confident would you be about the predictions in those areas of the 2D plane that are far from all of the cluster centers?

Hiiragi mouse embryo single cell expression data

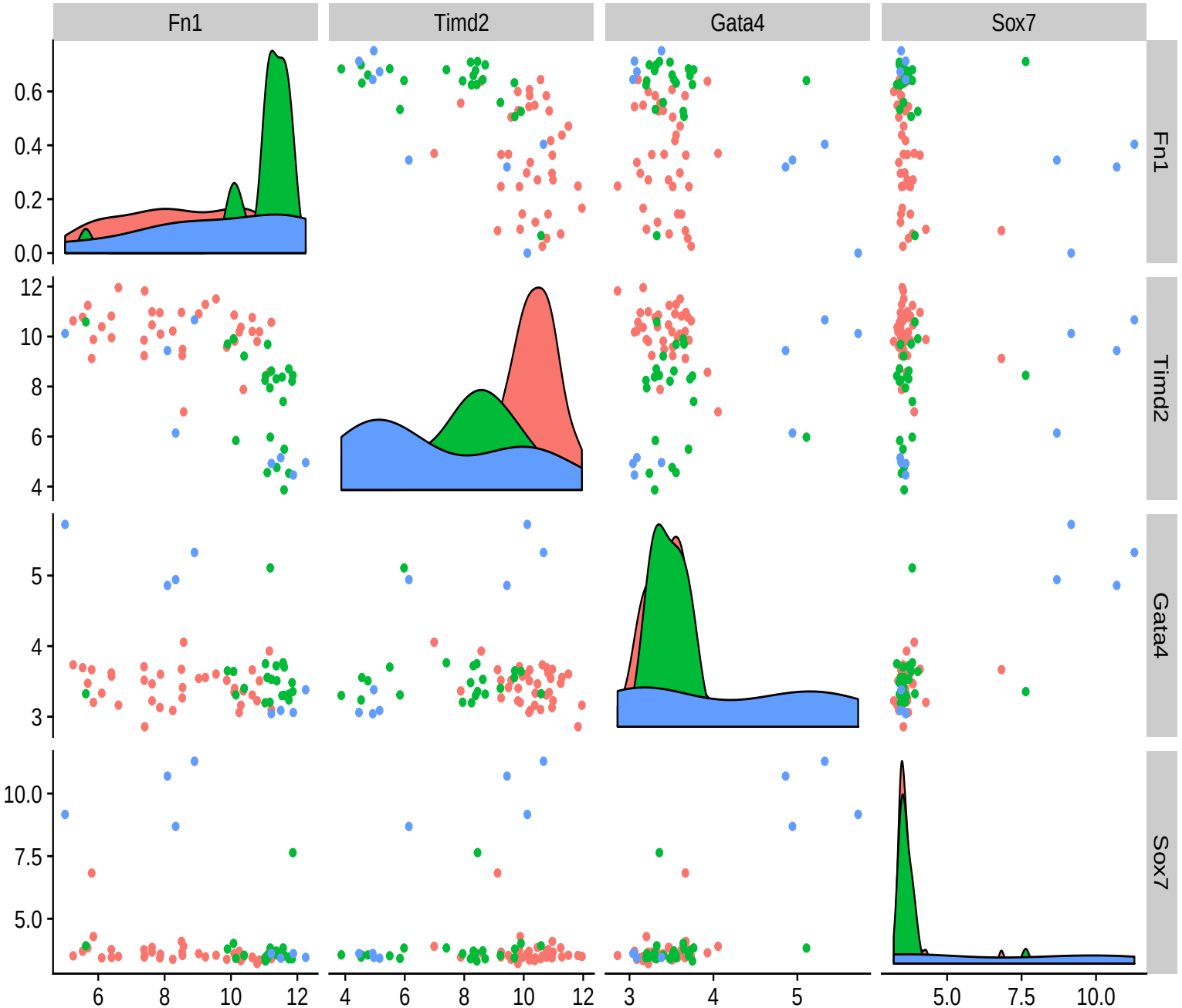
```
library("Hiiragi2013"); library("GGally"); library("dplyr")
data("x")
probes = c("1426642_at", "1418765_at", "1418864_at", "1416564_at")
embryoCells = t(Biobase::exprs(x)[probes, ]) %>%
  as_tibble %>%
  mutate(Embryonic.day = x$Embryonic.day) %>%
  dplyr::filter(x$genotype == "WT")
annotation(x)
```

```
## [1] "mouse4302"
```

```
library("mouse4302.db")
anno = AnnotationDbi::select(mouse4302.db, keys = probes,
                             columns = c("SYMBOL", "GENENAME"))
anno
```

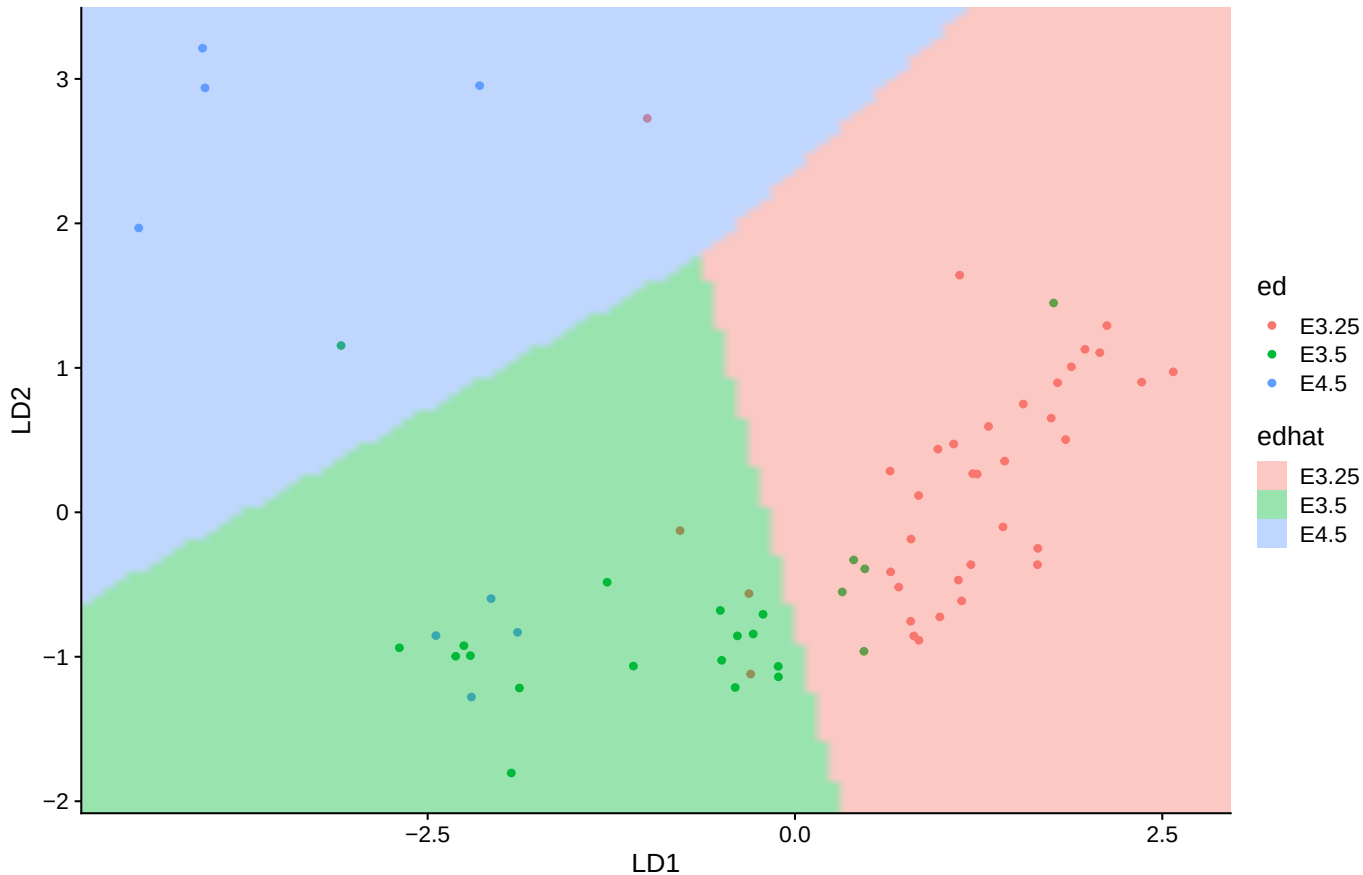
```
##      PROBEID SYMBOL                                                GENENAME
## 1 1426642_at    Fn1                                              fibronectin 1
## 2 1418765_at  Timd2 T cell immunoglobulin and mucin domain containing 2
## 3 1418864_at  Gata4                                  GATA binding protein 4
## 4 1416564_at   Sox7                SRY (sex determining region Y)-box 7
```

Hiiragi mouse embryo single cell expression data



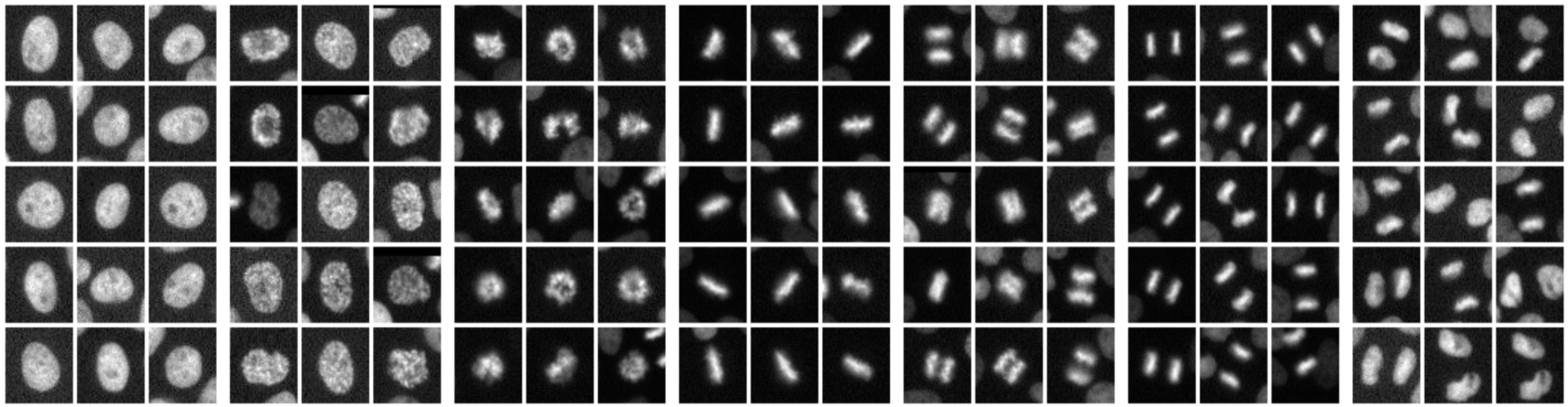
LDA classification regions for Embryonic.day

```
##          LD1  LD2
## Fn1    -0.2 -0.4
## Timd2   0.5  0.0
## Gata4 -0.1 -0.6
## Sox7   -0.7  0.5
```

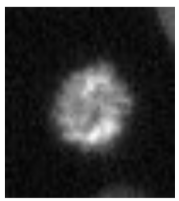


Morphological phenotyping

Provide Human Annotation to a small set of cells:

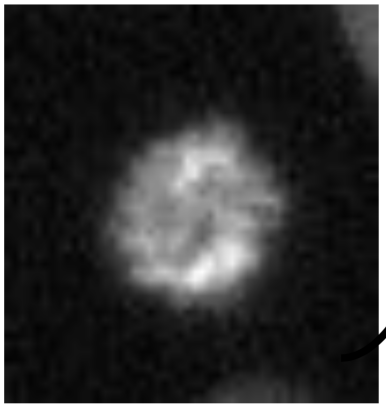


inter pro prometa meta earlyana lateana telo



Which mitotic phase is this?
Can we do this automatically?

Automatic classification workflow in reality



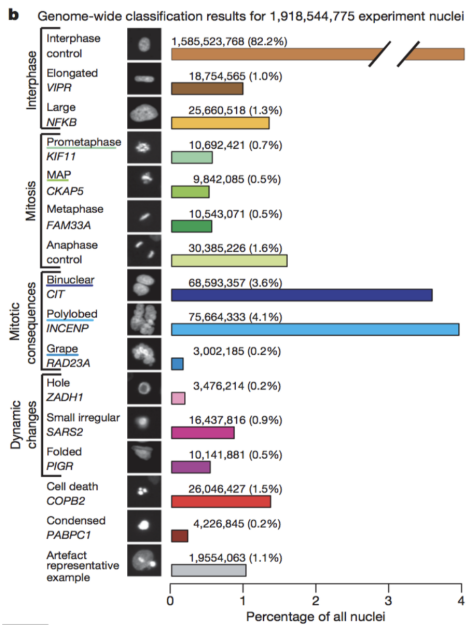
Preprocessing
e.g. normalization, background subtraction, …

Feature extraction
e.g. lightness, nucleus area, excentricity, …

Classification

Prophase

Metaphase



Summary

- Logistic regression
- Linear Discriminant Analysis (LDA)
- Question: when will LDA fail? What to do then?