MS1b Statistical Data Mining
Part 2: Supervised Learning
Parametric Methods

Yee Whye Teh
Department of Statistics
Oxford

http://www.stats.ox.ac.uk/~teh/datamining.html

Outline

Supervised Learning

So far we have been interested in using EDA and clustering techniques to understand high-dimensional data, useful for hypothesis generation. If a response (or grouping) variable occured in examples, it was merely to 'validate' that the discovered clusters or projections are meaningful.
We now move to supervised learning where in addition to having n observations of a p-dimensional predictor variable X, we also have a response variable $Y \in \mathcal{Y}$.

- Classification: group information is given and $\mathcal{Y} = \{1, \ldots, K\}$.
- Regression: a numerical value is observed and $\mathcal{Y} = \mathbb{R}$.

Given training data (X_i, Y_i), $i = 1, \ldots, n$, the goal is to accurately predict the class or response Y of new observations, when only the predictor variables X are observed.

Outline

Regression example: Boston Housing Data

The original data are 506 observations on 13 variables X; medv being the response variable Y.

```
crim    per capita crime rate by town
zn      proportion of residential land zoned for lots
        over 25,000 sq.ft
indus   proportion of non-retail business acres per town
chas    Charles River dummy variable (= 1 if tract bounds river;
        0 otherwise)
nox     nitric oxides concentration (parts per 10 million)
rm      average number of rooms per dwelling
age     proportion of owner-occupied units built prior to 1940
dis     weighted distances to five Boston employment centers
rad     index of accessibility to radial highways
tax     full-value property-tax rate per USD 10,000
ptratio pupil-teacher ratio by town
b       1000(B - 0.63)^2 where B is the proportion of blacks by to
lstat   percentage of lower status of the population
medv    median value of owner-occupied homes in USD 1000's
```

```
> str(X)
'data.frame': 506 obs. of  13 variables:
 $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
 $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
 $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87
 $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0
 $ rm     : num  6.58 6.42 7.18 7.00 7.15 ...
 $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9
 $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
 $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
 $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
 $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2
 $ black  : num  397 397 393 395 397 ...
 $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...

> str(Y)
num[1:506] 24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

Goal: predict median house price $\hat{Y}(X)$, given 13 predictor variables X of a new district.

Classification example: Lymphoma data

Revisiting the lymphoma gene expression data. Now in the supervised setting.
We have gene expression measurements of $n = 62$ patients for $p = 4026$ genes. These form the predictor variable matrix X.
For each patient, the subtype of cancer is available in a n dimensional vector Y with entries in $\{0, 1\}$.

```
> str(X)
'data.frame': 62 obs. of  4026 variables:
 $ Gene 1    : num  -0.344 -1.188  0.520 -0.748 -0.868 ...
 $ Gene 2    : num  -0.953 -1.286  0.657 -1.328 -1.330 ...
 $ Gene 3    : num  -0.776 -0.588  0.409 -0.991 -1.517 ...
 $ Gene 4    : num  -0.474 -1.588  0.219  0.978 -1.604 ...
 $ Gene 5    : num  -1.896 -1.960 -1.695 -0.348 -0.595 ...
 $ Gene 6    : num  -2.075 -2.117  0.121 -0.800  0.651 ...
 $ Gene 7    : num  -1.8755 -1.8187  0.3175  0.3873  0.0414 ...
 $ Gene 8    : num  -1.539 -2.433 -0.337 -0.522 -0.668 ...
 $ Gene 9    : num  -0.604 -0.710 -1.269 -0.832  0.458 ...
 $ Gene 10   : num  -0.218 -0.487 -1.203 -0.919 -0.848 ...
 $ Gene 11   : num  -0.340  1.164  1.023  1.133 -0.541 ...
 $ Gene 12   : num  -0.531  0.488 -0.335  0.496 -0.358 ...
 $ Gene 13   : num   0.0846  0.4820  1.5254  0.0323 -0.7563 ...
 $ Gene 14   : num  -1.2011 -0.0505 -0.8799  0.7518 -0.9964 ...
 $ Gene 15   : num  -0.9588 -0.0554 -1.0008  0.2502 -1.0235 ...

> str(Y)
 num [1:62] 0 0 0 1 0 0 1 0 0 0 ...
```

Goal: predict 'cancer class' $\hat{Y}(X) \in \{0, 1\}$, given 4026 predictor variables X (gene expressions) of a new patient.

Loss

Suppose we have trained a classifier or learner so that, upon observing a new predictor variable $X \in \mathbb{R}^p$, a prediction $\hat{Y} \in \mathcal{Y}$ is made.
How good is the prediction? We can use any loss function $L : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}^+$ to measure the loss incurred. Typical loss functions

- Misclassification error for classification

$$L(Y, \hat{Y}) = \begin{cases} 0 & Y = \hat{Y} \\ 1 & Y \neq \hat{Y} \end{cases} .$$

- Squared error loss for regression

$$L(Y, \hat{Y}) = (Y - \hat{Y})^2.$$

Alternative loss functions often useful. For example, non-equal misclassification error often appropriate. Or 'likelihood'-loss $L(Y, \hat{Y}) = -\log \hat{p}(Y)$, where $\hat{p}(k)$ is the estimated probability of class $k \in \mathcal{Y}$.

Risk and empirical risk minimization

For a given loss function L, the risk R of a learner is given by the expected loss

$$R(\hat{Y}) = \mathbb{E}(L(Y, \hat{Y})),$$

where $\hat{Y} = \hat{Y}(X)$ is a function of the random predictor variable X.
Ideally, we want to find a learner or procedure that minimizes the risk. The risk is unknown, however, as we just have finitely many samples.
Empirical risk minimization can be used, where one is trying to minimize –instead of the risk $R(\hat{Y})$– the empirical risk

$$R_n(\hat{Y}) = \mathbb{E}_n(L(Y, \hat{Y}) = \frac{1}{n} \sum_{i=1}^{n} L(Y_i, \hat{Y}_i).$$

The expectation is with respect to the empirical measure and hence just a summation over the observations.

The Bayes classifier

What is the optimal classifier if the joint distribution (X, Y) were known? The distribution f of a random predictor variable X can be written as

$$f(X) = \sum_{k=1}^{K} f_k(X) P(Y = k),$$

where, for $k = 1, \ldots, K$,

- the prior probabilities over classes are $P(Y = k) = \pi_k$
- and distributions of X, conditional on $Y = k$, is $f_k(X)$.

Given this scenario, the problem is to construct a 'good' classifier $\hat{Y}$ which assigns classes to observations

$$\hat{Y} : \mathcal{X} \to \left\{1, \ldots, K\right\}$$

We are interested in finding the classifier $\hat{Y}$ that minimises the risk under 0-1 loss, the *Bayes Classifier*.

$$\begin{aligned} R(\hat{Y}) &= \mathbb{E}\Big[L(Y, \hat{Y}(X))\Big] \\ &= \mathbb{E}\Big[\mathbb{E}[L(Y, \hat{Y}(x)\Big|X = x]\Big] \\ &= \int_{\mathcal{X}} \mathbb{E}\Big[L(Y, \hat{Y}(x))\Big|X = x\Big] f(x)dx \end{aligned}$$

For the Bayes classifier, minimizing $\mathbb{E}\Big[L(Y, \hat{Y}(x))\Big|X = x\Big]$ for each x suffices.

That is, given $X = x$, want to choose $\hat{Y}(x) \in \{1, \ldots, K\}$ such that the expected conditional loss is as small as possible.

Can write $\mathbb{E}\Big[L(Y,\hat{Y}(x))\Big|X=x\Big] = \sum_{k=1}^{K} L(k,\hat{Y}(x))P(Y=k|X=x)$.
Choosing $\hat{Y}(x)=m$ with $m\in\{1,\ldots,K\}$, the r.h.s. is simply

$$\mathbb{E}\Big[L(Y,\hat{Y}(x))|X=x\Big] = 1-P(Y=m|X=x).$$

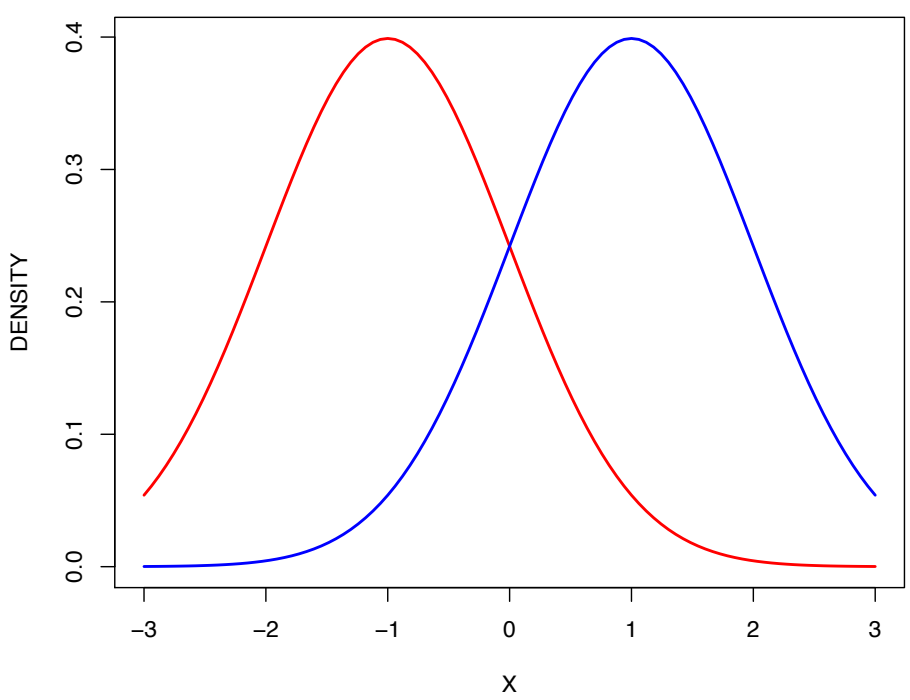
The Bayes Classifier chooses the class with the greatest posterior probability

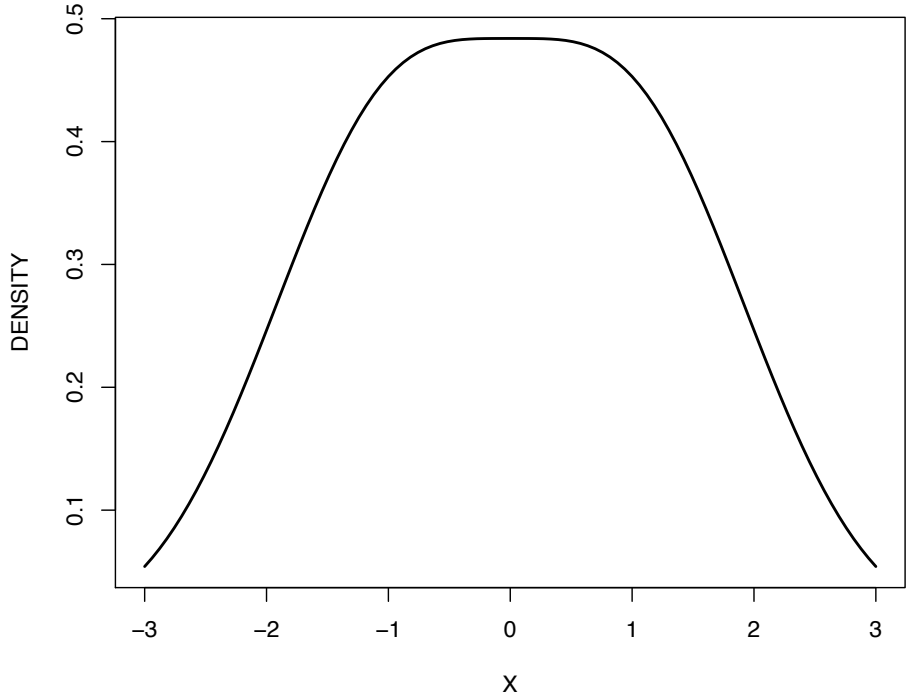
$$\begin{aligned}\hat{Y}(x) &= \underset{k=1,\ldots,K}{\arg\max}\, P(Y=k|X=x) = \underset{k=1,\ldots,K}{\arg\max}\, \frac{\pi_k f_k(x)}{\sum_{k=1}^{K}\pi_k f_k(x)}\\ &= \underset{k=1,\ldots,K}{\arg\max}\ \pi_k f_k(x).\end{aligned}$$

The Bayes classifier is optimal in terms of misclassification error.

Take a simple example, where π_k and f_k *are* known for $k = 1, \ldots, K$. Choose two classes $\{1, 2\}$.
Suppose $X \sim \mathcal{N}(\mu_Y, 1)$, where $\mu_1 = -1$ and $\mu_2 = 1$ and assume equal priors $\pi_1 = \pi_2 = 1/2$.
So $f(x) = \frac{1}{2} f_1(x) + \frac{1}{2} f_2(x)$, where

$$f_1(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{(x-(-1))^2}{2}) \quad \text{and} \quad f_2(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{(x-1)^2}{2}).$$

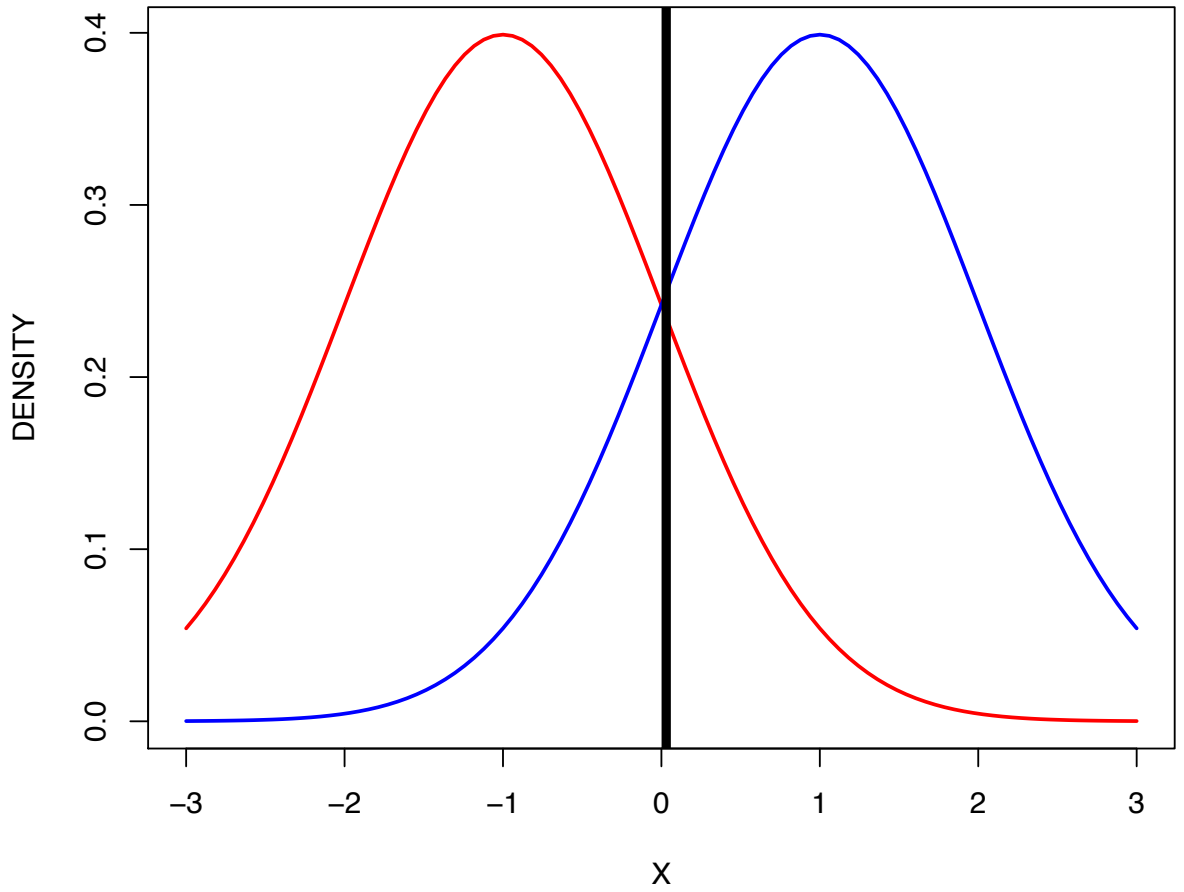




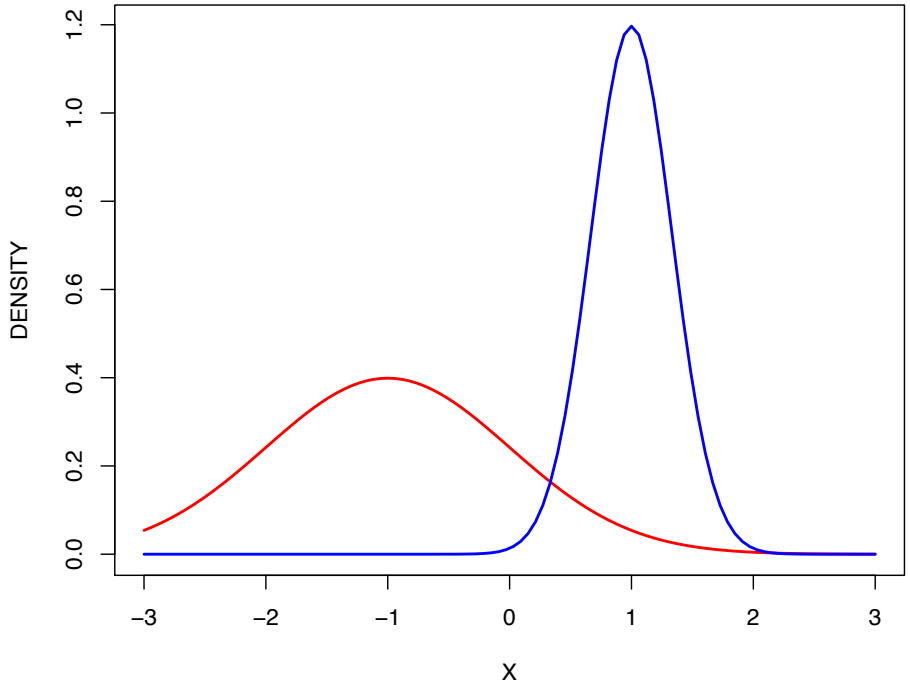
How do you classify a new observation $x = 0.1$?
Optimal classification is

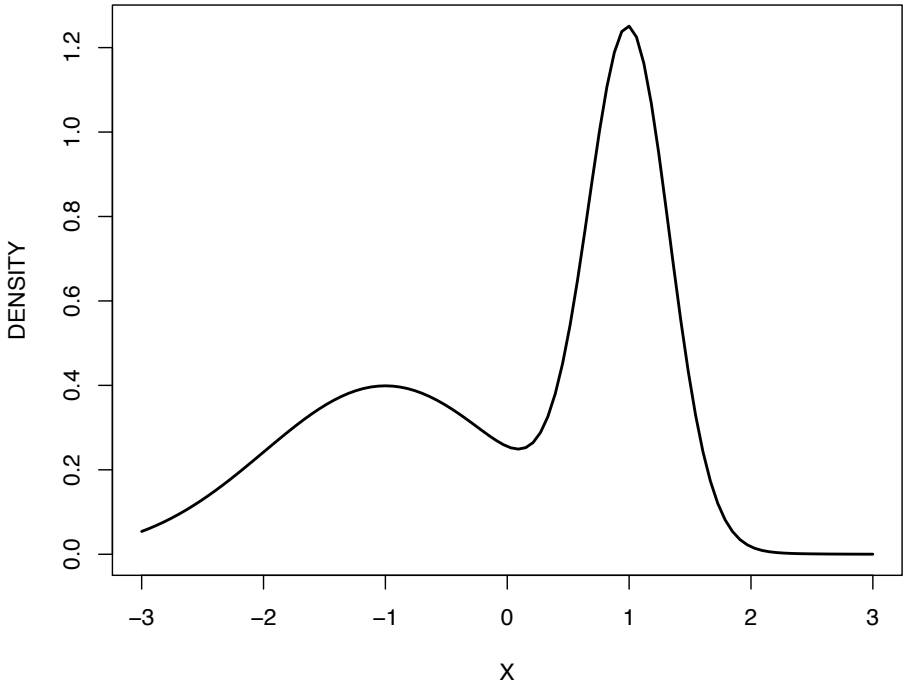
$$\hat{Y}(x) = \underset{k=1,\ldots,K}{\arg\max}\ \pi_k f_k(x),$$

which is class 1 if $x < 0$ and class 2 if $x \geq 0$.

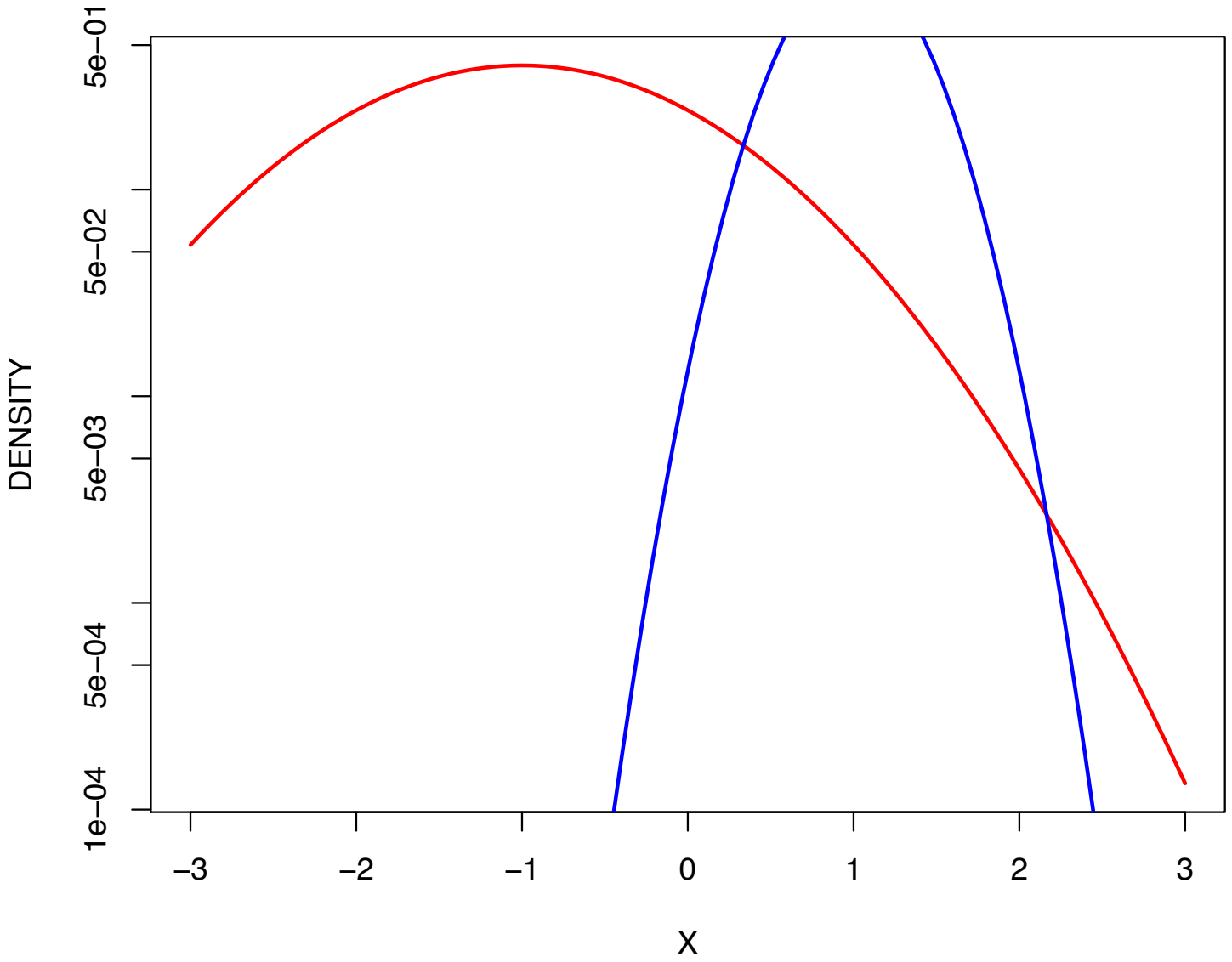


How do you classify a new observation x if now the standard deviation is still 1 for class 1 but 1/3 for class 2 ?





Looking at density in a log-scale, optimal classification is class 2 if and only if $x \in [-0.39, 2.15]$.



Plug-in classification

The Bayes Classifier chooses the class with the greatest posterior probability

$$\hat{Y}(x) \quad = \quad \underset{k=1,\ldots,K}{\arg\max}\, \pi_k f_k(x).$$

Unfortunately, we usually know neither the conditional class probabilities nor the prior probabilities.
Given

- estimates $\hat{\pi}_k$ for π_k and $k = 1, \ldots, K$ and
- estimates $\hat{f}_k(x)$ of conditional class probabilities,

the *plug-in classifiers* chooses the class

$$\hat{Y}(x) = \underset{k=1,\ldots,K}{\arg\max}\, \hat{\pi}_k \hat{f}_k(x).$$

Linear Discriminant Analysis will be an example of plug-in classification.