# Machine Learning, Machine Learning (extended)

## 4 – Supervised Learning: Bayesian Classification

Kashif Rajpoot

k.m.rajpoot@cs.bham.ac.uk

School of Computer Science

University of Birmingham

# Outline

- Supervised learning
- Classification
  - Probabilistic vs non-probabilistic
  - Generative vs discriminative
- Refresher: probability
- Bayesian classification
- Naïve Bayes classification
- Gaussian classification

# Supervised learning

- Regression
  - Minimised loss (e.g. least squares)
  - Maximum likelihood

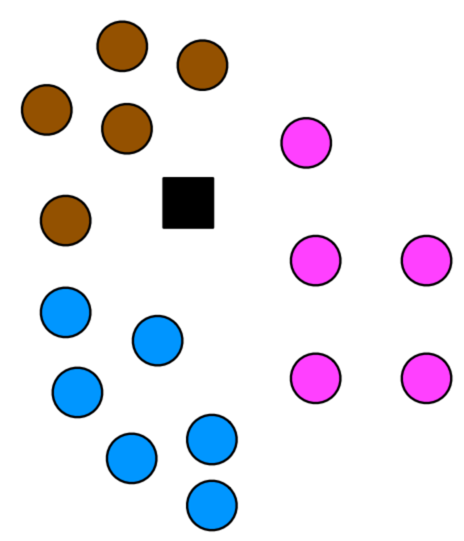
- Classification
  - Generative (e.g. Bayesian)
  - Instance-based (e.g. k-NN)
  - Discriminative (e.g. SVM)

# Classification

- A set of N objects with attributes (usually vector) $\boldsymbol{x}_n$
- Each object has an associated target label $t_n$
- Binary classification

  $t_n \in \{0,1\}$ or $t_n \in \{-1,1\}$
- Multi-class classification

  $t_n \in \{1,2,\dots,C\}$



- Classifier learns from $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N$ and $t_1, t_2, \dots, t_N$ so that it can later classify $\boldsymbol{x}_{new}$

# Probabilistic vs non-probabilistic classification

- Probabilistic classifiers produce a probability of class membership

$$P(t_{\text{new}} = k | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$$

$$P(t_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$$

$$P(t_{\text{new}} = 0 | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$$

- Non-probabilistic classifiers produce a hard assignment

$$t_{\text{new}} = 1 \text{ or } t_{\text{new}} = 0$$

# Probabilistic vs non-probabilistic classification

- Probabilities provide us with more information
  - $P(t_{new} = 1) = 0.6$ is more useful than $t_{new} = 1$
  - Confidence level

- Particularly important where cost of misclassification is high and imbalanced
  - Diagnosis: telling a diseased person they are healthy is much worse than telling a healthy person they are diseased

# Generative vs discriminative classification

- Generative classifiers generate a model for each class, based on training samples available
  - Data in each class can be seen as *generated* by some model
  - For new test samples, they assign these samples to the class that suits best (e.g. by probability measure)

- In contrast, discriminative classifiers attempt to explicitly define the decision boundary that separates the classes
  - Intuitively, these methods are for binary class problems but can be extended to multi-class problems

# Bayesian classifier

- A classifier built on Bayes rule
  - Builds a probabilistic model of the data, embedding prior knowledge
  - Allows us to extract prior knowledge from observed data

- Generative approach
  - Builds a model from training objects
  - Any new objects can be classified based on the probabilistic model specification

# Refresher: probability

- Conditional probability
- Joint probability
- Marginal probability
- Bayes rule

# Refresher: Conditional probability

- When the outcome of an event is affected (i.e. conditioned) by the outcome of another event

- For example: we toss a coin and then tell the result
  - $P(X = 1)$: probability of coin landing head
  - $P(X = 0)$: probability of coin landing tail
  - $P(Y = 1)$: probability of telling coin landed head
  - $P(Y = 1)$: probability of telling coin landed tail

- $P(Y = y|X = x)$: probability of telling coin landed $y$, given that coin has landed $x$

# Refresher: Conditional probability

- If we always tell the true outcome:
  - $P(Y = 1|X = 1) =?$
  - $P(Y = 1|X = 1) = 1$

  - $P(Y = 0|X = 0) =?$
  - $P(Y = 0|X = 0) = 1$

  - $P(Y = 0|X = 1) =?$
  - $P(Y = 0|X = 1) = 0$

  - $P(Y = 1|X = 0) =?$
  - $P(Y = 1|X = 0) = 0$

# Refresher: Conditional probability

- If we tell the true head outcome only 80% times:
  - $P(Y = 1|X = 1) =?$
  - $P(Y = 1|X = 1) = 0.8$

  - $P(Y = 0|X = 0) =?$
  - $P(Y = 0|X = 0) = 1$

  - $P(Y = 0|X = 1) =?$
  - $P(Y = 0|X = 1) = 0.2$

  - $P(Y = 1|X = 0) =?$
  - $P(Y = 1|X = 0) = 0$

# Refresher: Joint probability

- What is the probability that the coin lands heads and we say heads?
  - This is joint probability (i.e. probability of two or more variables)
  - $P(Y = y, X = x)$

- In case of no dependence between variables:
  - $P(Y = y, X = x) = P(Y = y)P(X = x)$

- In case of dependence between variables:
  - $P(Y = y, X = x) = P(Y = y|X = x)P(X = x)$, or
  - $P(Y = y, X = x) = P(X = x|Y = y)P(Y = y)$

# Refresher: Joint probability

- What is the probability that the coin lands heads and we say heads?
  - $P(Y = 1, X = 1) = P(Y = 1|X = 1)P(X = 1) = ?$
  - $P(Y = 1, X = 1) = P(Y = 1|X = 1)P(X = 1) = 0.8\text{x}0.5 = 0.4$

- Other joint probabilities
  - $P(Y = 0, X = 1) = P(Y = 0|X = 1)P(X = 1) = ?$
  - $P(Y = 0, X = 1) = P(Y = 0|X = 1)P(X = 1) = 0.2\text{x}0.5 = 0.1$
  - $P(Y = 1, X = 0) = P(Y = 1|X = 0)P(X = 0) = ?$
  - $P(Y = 1, X = 0) = P(Y = 1|X = 0)P(X = 0) = 0\text{x}0.5 = 0$
  - $P(Y = 0, X = 0) = P(Y = 0|X = 0)P(X = 0) = ?$
  - $P(Y = 0, X = 0) = P(Y = 0|X = 0)P(X = 0) = 1\text{x}0.5 = 0.5$

# Refresher: Marginal probability

- What is the probability of telling that the coin landed heads, or telling that the coin landed tails?
  - $P(Y = 1) =?$
  - $P(Y = 0) =?$
  - Where is $X$?
- $X$ has been marginalized from the joint distribution $P(Y = y, X = x)$
  - $P(Y = y) = \sum_x P(Y = y, X = x)$
  - For coin toss example:
    - $P(Y = y) = P(Y = y, X = 0) + P(Y = y, X = 1)$
- $P(Y = 1) = P(Y = 1, X = 0) + P(Y = 1, X = 1)$
- $P(Y = 1) = 0 + 0.4 = 0.4$
- $P(Y = 0) =?$

# Refresher: Bayes rule

- Joint probability can be estimated as (assuming dependence between variables):
  - $P(Y = y, X = x) = P(Y = y|X = x)P(X = x)$, or
  - $P(Y = y, X = x) = P(X = x|Y = y)P(Y = y)$

- By equating the right hand sides:
  - $P(X = x|Y = y)P(Y = y) = P(Y = y|X = x)P(X = x)$, so:

$$P(X = x|Y = y) = \frac{P(Y = y|X = x)P(X = x)}{P(Y = y)}$$

- For coin toss example, this is finding the probability that the coin landed in a particular way given what was said about the outcome

# Refresher: Bayes rule

- What is the probability of coin landing head if it was told as head?
  - $P(X=1|Y=1)=?$
  - $\frac{P(Y=1|X=1)P(X=1)}{P(Y=1)}=\frac{0.8\text{x}0.5}{0.4}=1$
- What is the probability of coin landing tail if it was told as tail?
  - $P(X=0|Y=0)=?$
  - $\frac{P(Y=0|X=0)P(X=0)}{P(Y=0)}=\frac{1\text{x}0.5}{0.6}=0.83$
- What is the probability of coin landing head if it was told as tail?
  - $P(X=1|Y=0)=?$
- What is the probability of coin landing tail if it was told as head?
  - $P(X=0|Y=1)=?$

# Refresher: Bayes rule

$$P(X = x|Y = y) = \frac{P(Y = y|X = x)P(X = x)}{P(Y = y)}$$

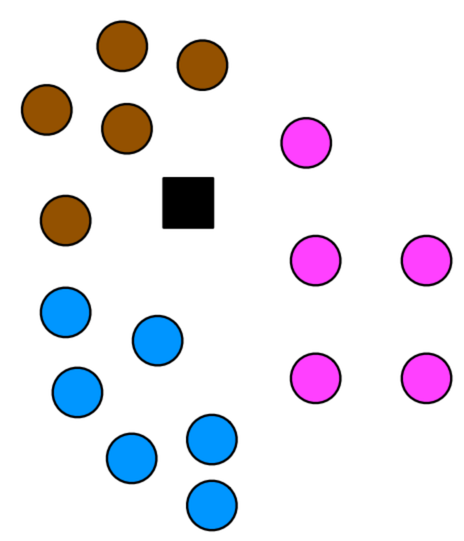
- $P(X = x)$ denotes prior belief: prior probability of event $x$ occurring, before seeing any data for prediction
- $P(Y = y|X = x)$ denotes class-conditional likelihood: probability of the observed data $y$ occurring, given that event $x$ has occurred
- $P(Y = y)$ denotes data evidence: marginal probability of observed data $y$
  - $P(Y = y) = \sum_x P(Y = y, X = x) = \sum_x P(Y = y|X = x)P(X = x)$
- $P(X = x|Y = y)$ denotes posterior probability: probability of event $x$ occurring after seeing the new data $y$

# Classification

- A set of N objects with attributes (usually vector) $\boldsymbol{x}_n$
- Each object has an associated target label $t_n$
- Binary classification

  $t_n \in \{0,1\}$ or $t_n \in \{-1,1\}$
- Multi-class classification

  $t_n \in \{1,2,\ldots,C\}$



- Classifier learns from $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ and $t_1, t_2, \ldots, t_N$ so that it can later classify $\boldsymbol{x}_{new}$

# Classification

- Let's begin with a simpler problem formulation
- A set of N objects with discrete-valued scalar attribute $x_n$
- Each object $x_n$ has an associated target label $t_n$
- Classifier learns from $x_1, x_2, \ldots, x_N$ and $t_1, t_2, \ldots, t_N$ so that it can later classify $x_{new}$
- Let's turn Bayes rule in to a classifier that can predict $t_{new}$ for unseen data $x_{new}$

# Bayesian classification

$$P(t_{new} = c|x_{new}) = \frac{P(x_{new}|t_{new} = c)P(t_{new} = c)}{P(x_{new})}$$

What is $x_{new}$?

- $P(t_{new} = c)$ - prior belief: prior probability of label $c$ occurring, before seeing any data for prediction
- $P(x_{new}|t_{new} = c)$ - class-conditional likelihood: probability of the data $x_{new}$ occurring, given that label $c$ is true
- $P(x_{new})$ - data evidence: marginal probability of data $x_{new}$
  - $P(x_{new}) = \sum_{c=1}^{C} P(x_{new}, t_{new} = c) = \sum_{c=1}^{C} P(x_{new}|t_{new} = c)P(t_{new} = c)$
- $P(t_{new} = c|x_{new})$ - posterior probability: probability of label $c$ occurring after seeing the data $x_{new}$

# Bayesian classification

$$P(t_{new} = c | x_{new}) = \frac{P(x_{new} | t_{new} = c) P(t_{new} = c)}{P(x_{new})}$$

can be written as:

$$P(c | x_{new}) = \frac{P(x_{new} | c) P(c)}{P(x_{new})}$$

- The Bayesian classifier estimates probability for all candidate class labels $c \in \{1, 2, \ldots, C\}$
  - In this context, each class label $c$ can also be termed as *candidate hypothesis* whose probability is estimated
  - Let's call set of all target labels or candidate hypotheses as hypotheses space $H = \{1, 2, \ldots, C\}$

# Bayesian classification

- Bayesian classifier aims to assign target label $c \in H$ that has maximum posterior probability

$$\underset{c \in H}{argmax}\, P(c|x_{new})$$

$$\underset{c \in H}{argmax} \frac{P(x_{new}|c)P(c)}{P(x_{new})}$$

$$\underset{c \in H}{argmax}\, P(x_{new}|c)P(c)$$

- Note that $P(x_{new})$ is independent of target label $c$
  - i.e. it is same for all target labels, thus can be omitted
- This is called *maximum a posterior* hypothesis

# Bayesian classification

- $P(x_{new}|c)$ denotes *class-conditional distribution*, specific to class $c$, evaluated at $x_{new}$
  - We need a class-conditional distribution for each class $c$, at each discrete-valued scalar attribute $x_{new}$
  - This distribution can be estimated from training data

- $P(c)$ denotes *prior probability* that can be set as:
  - Uniform prior: $P(c) = \frac{1}{C}$ i.e. each class is equally probable
  - Class size prior: $P(c) = \frac{N_c}{N}$ i.e. class prior as per its frequency in training observations

# Bayesian classification

- In cases where the prior is uniform for all labels

$$\underset{c \in H}{argmax}\, P(x_{new}|c)P(c)$$

becomes

$$\underset{c \in H}{argmax}\, P(x_{new}|c)$$

- This is called *maximum likelihood* hypothesis

# Bayesian classification

- *Maximum a posterior* (MAP) hypothesis
  - $\underset{c \in H}{argmax}\, P(x_{new}|c)P(c)$

- *Maximum likelihood* (ML) hypothesis
  - $\underset{c \in H}{argmax}\, P(x_{new}|c)$

- Typically, MAP estimate is used for Bayesian classification since it is flexible regarding *prior* use

# Bayesian classification: Example

- Cancer diagnosis problem
  - A patient takes a lab test and the result comes back positive for cancer.
  - It is known that the test returns a correct positive result in only 98% of the cases and a correct negative result in only 97% of the cases.
  - Furthermore, only 0.008 (i.e. 0.8%) of the entire population has cancer.

1. What is the probability that this patient has cancer?
2. What is the probability that this patient does not have cancer?
3. What is the diagnosis?

# Bayesian classification: Example

- Cancer diagnosis problem
  - A patient takes a lab test and the result comes back positive for cancer.
  - It is known that the test returns a correct positive result in only 98% of the cases and a correct negative result in only 97% of the cases.
  - Furthermore, only 0.008 (i.e. 0.8%) of the entire population has cancer.

  - $P(cancer|+) = P(+|cancer)P(cancer) =?$
  - $P(\neg cancer|+) = P(?\,|?\,)P(?\,) =?$

# Bayesian classification

- In the discussion (and example) so far, the data has only one discrete-valued attribute

$$\underset{c \in H}{argmax}\, P(x_{new}|c)P(c)$$

  - What if data has several discrete-valued attributes?

- Classification problem addressed was:
  - Classifier learns from $x_1, x_2, \dots, x_N$ and $t_1, t_2, \dots, t_N$ so that it can later classify $x_{new}$

- Now the classification problem becomes:
  - Classifier learns from $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N$ and $t_1, t_2, \dots, t_N$ so that it can later classify $\boldsymbol{x}_{new}$

# Naïve Bayes assumption

- Instead of $\underset{c \in H}{argmax}\, P(x_{new}|c)P(c)$, we need to estimate:

$$\underset{c \in H}{argmax}\, P(\boldsymbol{x}_{new}|c)P(c) = \underset{c \in H}{argmax}\, P\left(x_{new}^{1}, x_{new}^{2}, \dots, x_{new}^{d} \middle| c\right)P(c)$$

- Recall that $P(\boldsymbol{x}_{new}|c)$ denotes *class-conditional distribution*, specific to class $c$, evaluated at $\boldsymbol{x}_{new}$
  - It is extremely difficult to fit class-conditional distribution for each class $c$, for a high dimensional data
- Naïve Bayes assumption: attributes that describe data are conditionally independent given a hypothesis

$$P(\boldsymbol{x}_{new}|c) = \prod_{i=1}^{d} P\left(x_{new}^{i} \middle| c\right)$$

  - It is a simplifying assumption, obviously it may be violated in reality
  - In spite of that, it works well in practice

# Naïve Bayes classification

- Naïve Bayes classifier: uses the Naïve Bayes assumption and estimates the *maximum a posterior* (MAP) hypothesis

$$\underset{c \in H}{argmax}\, P(\boldsymbol{x}_{new}|c)P(c) = \underset{c \in H}{argmax} \prod_{i=1}^{d} p\left(x_{new}^{i}|c\right) P(c)$$

- Note that $p\left(x_{new}^{i}|c\right)$ denotes probability for $i^{\text{th}}$ attribute value

- Each attribute has discrete values (in discussion so far)
  - Continuous attribute values to be discussed later…

- Very simple but practical classification algorithm

- Successful applications:
  - Medical diagnosis
  - Text classification

# Naïve Bayes classification: Learning to diagnose

- Given the patients data (symptoms and diagnosis):

| chills | runny nose | headache | fever | Flu? |
|---|---|---|---|---|
| Y | N | Mild | Y | N |
| Y | Y | No | N | Y |
| Y | N | Strong | Y | Y |
| N | Y | Mild | Y | Y |
| N | N | No | N | N |
| N | Y | Strong | Y | Y |
| N | Y | Strong | N | N |
| Y | Y | Mild | Y | Y |

- What is the probability for the following?

| | | | | |
|---|---|---|---|---|
| Y | N | Mild | N | ? |

# Naïve Bayes classification: Learning to diagnose

- What is the probability for the following?

| chills | runny nose | headache | fever | Flu? |
|---|---|---|---|---|
| Y | N | Mild | N | ? |

- $P(flu = Y|\boldsymbol{x}_{new}) = P(\boldsymbol{x}_{new}|flu = Y)P(flu = Y) =?$
- Using Naïve Bayes classifier:
  - $P(flu = Y|\boldsymbol{x}_{new}) =$
    $$P(chills = Y|flu = Y)$$
    $$P(runny\ nose = N|flu = Y)$$
    $$P(headache = Mild|flu = Y)$$
    $$P(fever = N|flu = Y)$$
    $$P(flu = Y) =?$$
- $P(flu = N|\boldsymbol{x}_{new}) = P(\boldsymbol{x}_{new}|flu = N)P(flu = N) =?$

# Naïve Bayes classification: Learning to classify

- Given the tennis playing data:

| outlook | temperature | humidity | wind | Play tennis? |
|---|---|---|---|---|
| sunny | hot | high | weak | N |
| sunny | hot | high | strong | N |
| overcast | hot | high | weak | Y |
| rain | mild | high | weak | Y |
| rain | cool | normal | weak | Y |
| rain | cool | normal | strong | N |
| overcast | cool | normal | strong | Y |
| sunny | mild | high | weak | N |
| sunny | cool | normal | weak | Y |
| rain | mild | normal | weak | Y |
| sunny | mild | normal | strong | Y |
| overcast | mild | high | strong | Y |
| overcast | hot | normal | weak | Y |
| rain | mild | high | strong | N |

# Naïve Bayes classification: Learning to classify

- Classify the new data:

| outlook | temperature | humidity | wind | Play tennis? |
| --- | --- | --- | --- | --- |
| sunny | cool | high | strong | ? |

- $P(play = Y|\boldsymbol{x}_{new}) = P(\boldsymbol{x}_{new}|play = Y)P(play = Y) = ?$

- $P(play = N|\boldsymbol{x}_{new}) = P(\boldsymbol{x}_{new}|play = N)P(play = N) = ?$

# Gaussian classification

- In the discussion (and examples) so far, the data has discrete-valued attributes (e.g. ‘sunny’, ‘Y’)
  - What if the attributes are real-valued?
- For discrete-valued attributes, we used probability distributions to estimate attribute value relation with class label
- For real-valued attributes, we need to use probability density function (pdf) to estimate attribute value relation with class label
- Gaussian classifier: Bayes or Naïve Bayes classifier that utilizes Gaussian pdf

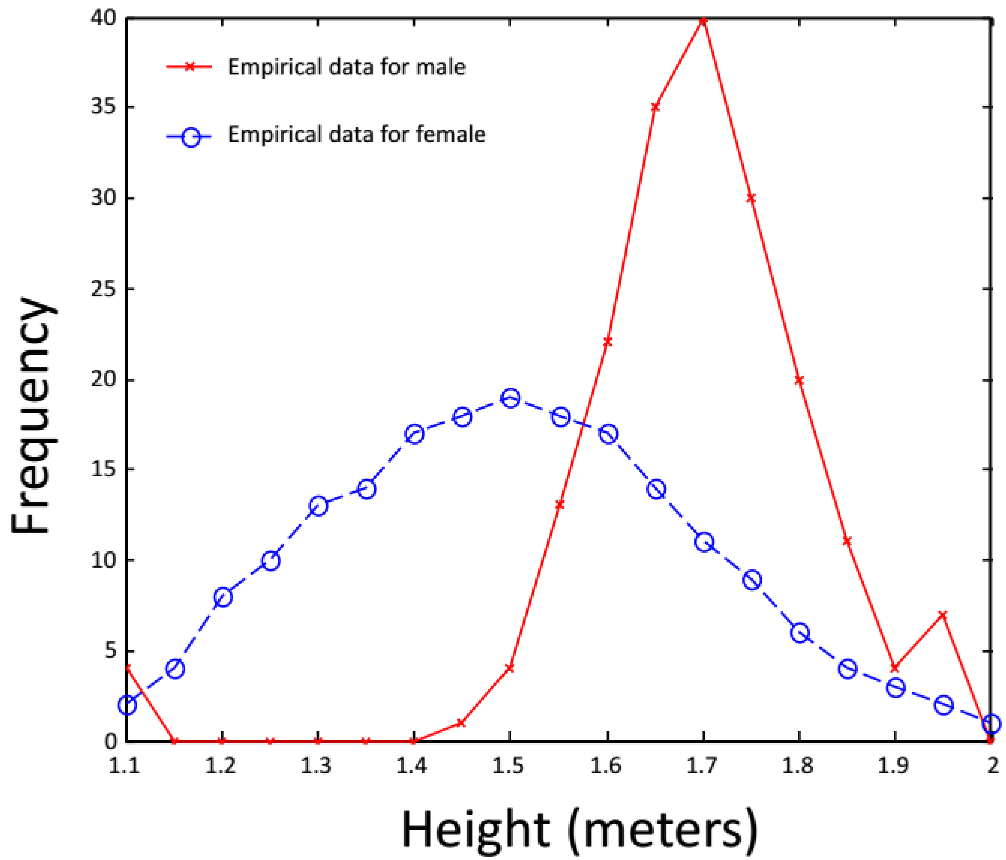
# Gaussian classification: Learning to predict gender

- Given heights of 190 male and 190 females, can a classifier learn to predict gender?

- Attribute?
  - Height

- Class labels?
  - $t_{male} = 1$
  - $t_{female} = 0$

- $c \in \{0,1\}$



# Gaussian classification: Learning to predict gender

- Let's recall maximum a posterior estimate from Bayes rule:

$$p(c|x_{new}) = p(x_{new}|c)p(c)$$

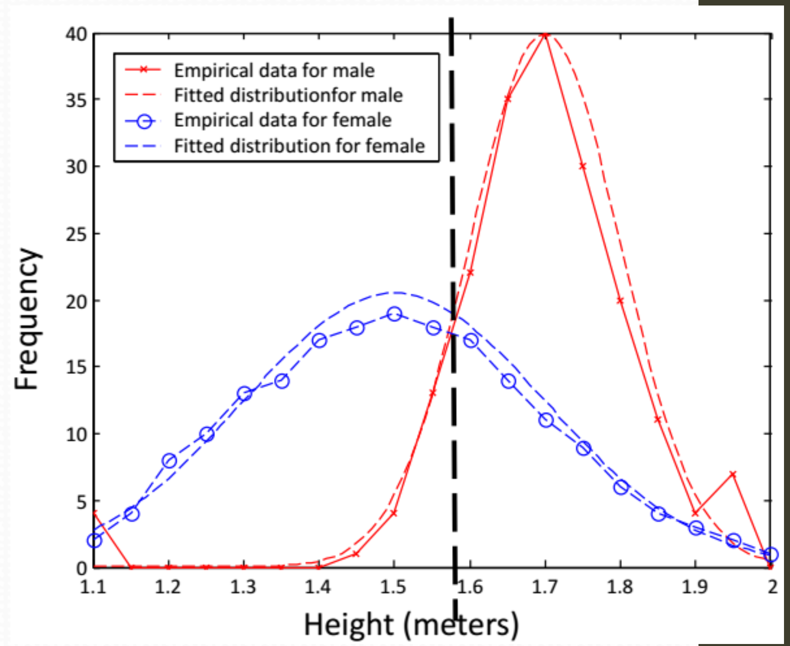
- We need class prior:
  - $p(c = 1) = ?$
  - $p(c = 0) = ?$
- We will also need class-conditional likelihood:
  - $p(x_{new}|c = 1)$: probability that a male has height $x_{new}$
  - $p(x_{new}|c = 0)$: probability that a female has height $x_{new}$
- Posterior
  - $p(c = 1|x_{new})$: probability that height $x_{new}$ is a male
  - $p(c = 0|x_{new})$: probability that height $x_{new}$ is a female

# Gaussian classification: Learning to predict gender

- Class-conditional likelihood $p(x_{new}|c)$
  - Class-conditional distribution can be modelled as a Gaussian pdf, for each class
- Univariate Gaussian pdf

$$p(x_{new}|\mu_c, \sigma_c^2) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left\{-\frac{x_{new}-\mu_c}{2\sigma_c^2}\right\}$$

- How to estimate $\mu_c$ and $\sigma_c^2$?
  - $\mu_c = \frac{1}{N_c}\sum_{n=1}^{N_c} x_n$
  - $\sigma_c^2 = \frac{1}{N_c}\sum_{n=1}^{N_c}(x_n - \mu_c)^2$
- We will have a separate Gaussian pdf for each class
  - $\mu_0$ and $\sigma_0^2$: mean and variance for female class
  - $\mu_1$ and $\sigma_1^2$: mean and variance for male class



# Gaussian classification: Learning to predict gender

- By estimating class-conditional likelihood and class prior, posterior can be estimated:

$$p(c|x_{new}) = p(x_{new}|c)p(c)$$

- Since $p(x_{new}|c) = p(x_{new}|\mu_c, \sigma_c^2)$, we get:

$$p(c|x_{new}) = p(x_{new}|\mu_c, \sigma_c^2)p(c)$$

- Recall that

$$p(x_{new}|\mu_c, \sigma_c^2) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left\{-\frac{x_{new} - \mu_c}{2\sigma_c^2}\right\}$$

- Thus, the posterior estimate for prediction is:

$$p(c|x_{new}) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left\{-\frac{x_{new} - \mu_c}{2\sigma_c^2}\right\} p(c)$$

# Gaussian classification

- In the Gaussian classifier discussion so far, the data has only one attribute
  - i.e. one-dimensional input $x_n$
  - Univariate Gaussian pdf was sufficient to estimate class-conditional likelihood

- What if the data has multiple attributes?
  - i.e. d-dimensional input $\boldsymbol{x}_n = \{x_n^1, x_n^2, \dots, x_n^d\}$
  - Multivariate Gaussian pdf will be needed for estimation of class-conditional likelihood

$$p(\boldsymbol{x}_{new}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \\ = \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}_c|}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}_{new} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c{}^{-1}(\boldsymbol{x}_{new} - \boldsymbol{\mu}_c)\right\}$$

# Gaussian classification

- Multivariate Gaussian pdf for class-conditional likelihood

$$p(\boldsymbol{x}_{new}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) = \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}_c|}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}_{new} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c{}^{-1}(\boldsymbol{x}_{new} - \boldsymbol{\mu}_c)\right\}$$

where

$$\boldsymbol{\mu}_c = \frac{1}{N_c}\sum_{n=1}^{N_c} \boldsymbol{x}_n$$

$$\boldsymbol{\Sigma}_c = \frac{1}{N_c}\sum_{n=1}^{N_c} (\boldsymbol{x}_n - \boldsymbol{\mu}_c)(\boldsymbol{x}_n - \boldsymbol{\mu}_c)^T$$

What is the difference between $\boldsymbol{x}_n$ and $\boldsymbol{x}_{new}$?

# Gaussian classification

- By estimating class-conditional likelihood and class prior, posterior can be estimated:

$$p(c|\boldsymbol{x}_{new}) = p(\boldsymbol{x}_{new}|c)p(c)$$

- Since $p(\boldsymbol{x}_{new}|c) = p(\boldsymbol{x}_{new}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$, we get:

$$p(c|\boldsymbol{x}_{new}) = p(\boldsymbol{x}_{new}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)p(c)$$

- Recall that

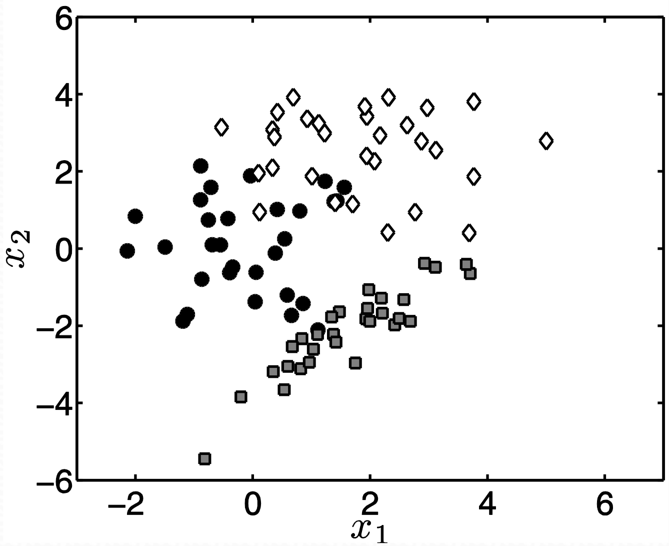
$$p(\boldsymbol{x}_{new}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) = \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}_c|}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}_{new} - \boldsymbol{\mu}_c)^T {\boldsymbol{\Sigma}_c}^{-1}(\boldsymbol{x}_{new} - \boldsymbol{\mu}_c)\right\}$$

- Thus, the posterior estimate for prediction is:

$$p(c|\boldsymbol{x}_{new}) = \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}_c|}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}_{new} - \boldsymbol{\mu}_c)^T {\boldsymbol{\Sigma}_c}^{-1}(\boldsymbol{x}_{new} - \boldsymbol{\mu}_c)\right\} p(c)$$

# Gaussian classification

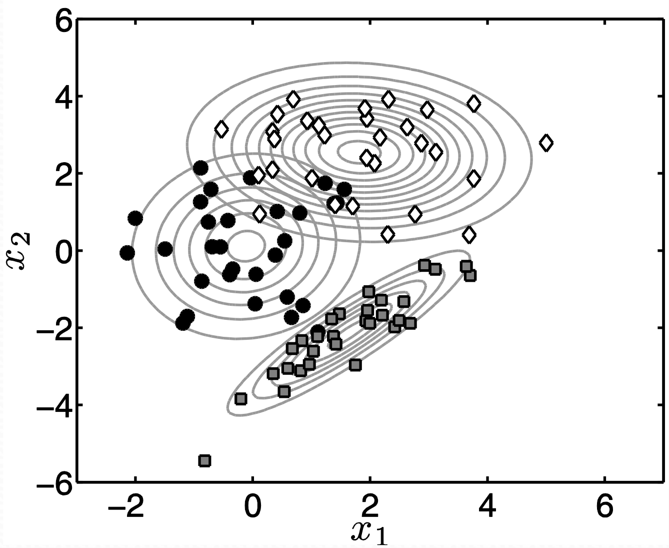
- Example: three class 2d data (30 samples each)
  - 1: black circles
  - 2: white diamonds
  - 3: grey squares




- Class conditional prior

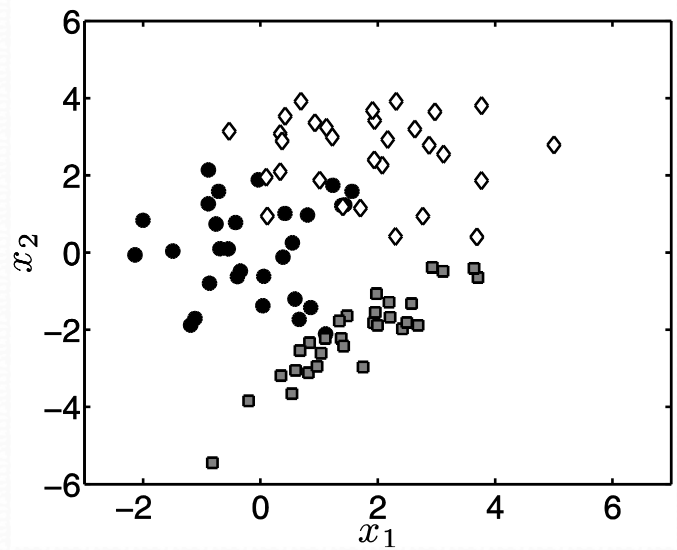
$$p(\boldsymbol{x}_{new}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) = \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}_c|}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}_{new} - \boldsymbol{\mu}_c)^T {\boldsymbol{\Sigma}_c}^{-1}(\boldsymbol{x}_{new} - \boldsymbol{\mu}_c)\right\}$$

- Density contours



# Gaussian classification: Making predictions

- Example: three class 2d data (30 samples each)
  - 1: black circles
  - 2: white diamonds
  - 3: grey squares

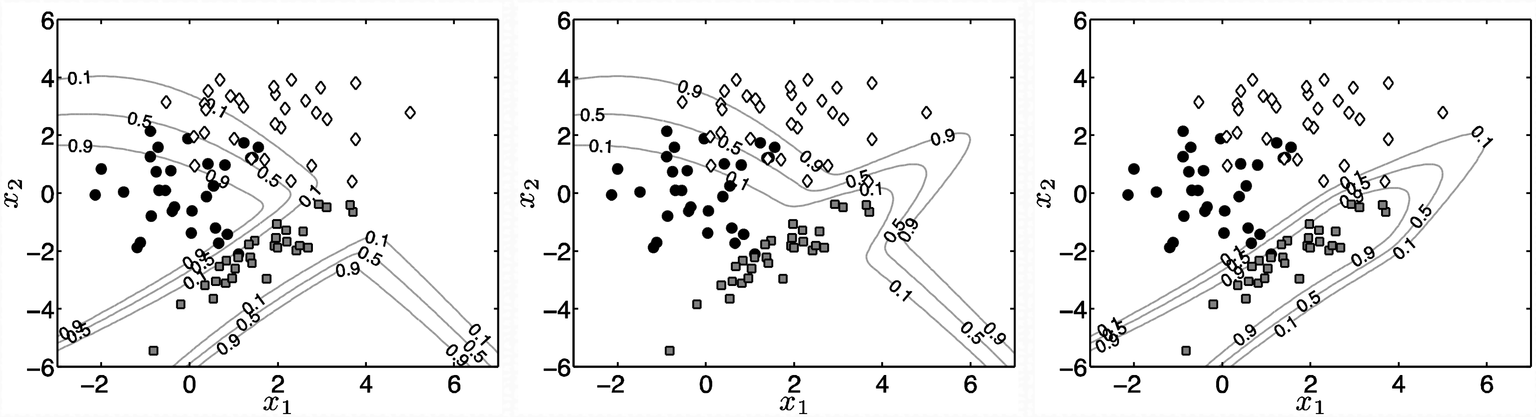



- Posterior $p(c|\boldsymbol{x}_{new})$ for $\boldsymbol{x}_{new} = [2{,}0]^T$

| $c$ | $p(\boldsymbol{x}_{new}\|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ | $p(c)$ | $p(c\|\boldsymbol{x}_{new}) = p(\boldsymbol{x}_{new}\|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)p(c)$ | $\boldsymbol{P}(c\|\boldsymbol{x}_{new})$ |
|---|---|---|---|---|
| 1 | 0.0138 | 1/3 | 0.0046 | 0.6890 |
| 2 | 0.0061 | 1/3 | 0.0020 | 0.3024 |
| 3 | 0.0002 | 1/3 | 0.0001 | 0.0087 |

# Gaussian classification: Making predictions

- By evaluating the Gaussian classifier on a grid of many $\boldsymbol{x}_{new}$ values, we can estimate and draw the classification probability contours




(a) $P(T_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$ (b) $P(T_{\text{new}} = 2|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$ (c) $P(T_{\text{new}} = 3|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$

- The steepness of density contours for class 3 partly explains the odd behaviour in (a) and (b)

# Gaussian classification

- How many parameters to estimate multivariate Gaussian pdf with 2d data, for each class?
  - $\boldsymbol{\mu}_c = \frac{1}{N_c}\sum_{n=1}^{N_c} \boldsymbol{x}_n$
  - $\boldsymbol{\Sigma}_c = \frac{1}{N_c}\sum_{n=1}^{N_c}(\boldsymbol{x}_n - \boldsymbol{\mu}_c)(\boldsymbol{x}_n - \boldsymbol{\mu}_c)^T$
  - Five parameters: two for $\boldsymbol{\mu}_c$ and three for $\boldsymbol{\Sigma}_c$
- In general, for $D$ dimensional data, we need to estimate $D + D + \frac{D(D-1)}{2}$ parameters
  - For 10-dimensional data, 65 parameters need to be estimated for each class
- What if the training data size is small (e.g. 30 samples)?

# Gaussian classification: Naïve Bayes assumption

- To "overcome" lack of data, Naïve Bayes assumption can be utilized
  - i.e. each attribute is assumed to be independent

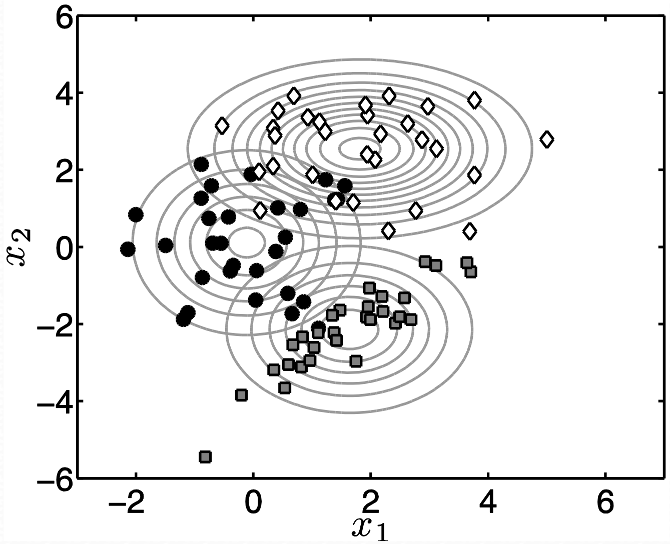
- Class-conditional multivariate distribution is factorized in to product of $D$ univariate distributions, for each class

$$p(\boldsymbol{x}_{new}|c) = \prod_{d=1}^{D} p(x_{new}^{d}|c) = \prod_{d=1}^{D} p(x_{new}^{d}|\mu_c, \sigma_c^2)$$

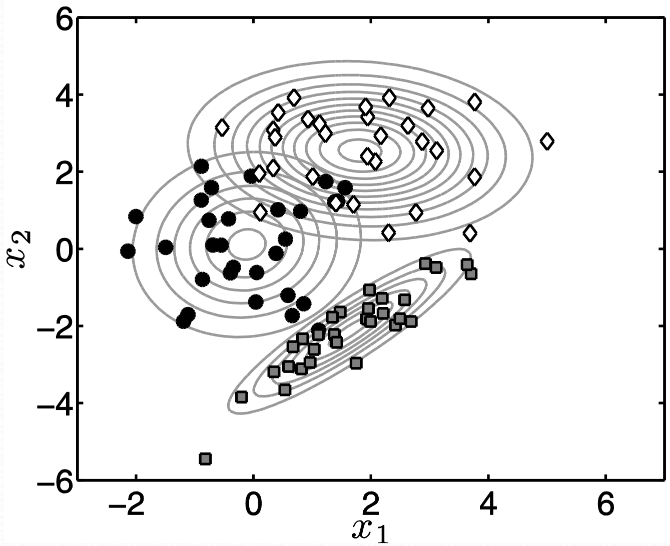
- Each univariate distribution relies on only two parameters: mean $\mu_c$ and variance $\sigma_c^2$
  - Thus, only $2 * D$ parameters need to be estimated with Naïve Bayes assumption

# Gaussian classification: Naïve Bayes assumption

- Density contours, <u>with</u> Naïve Bayes assumption



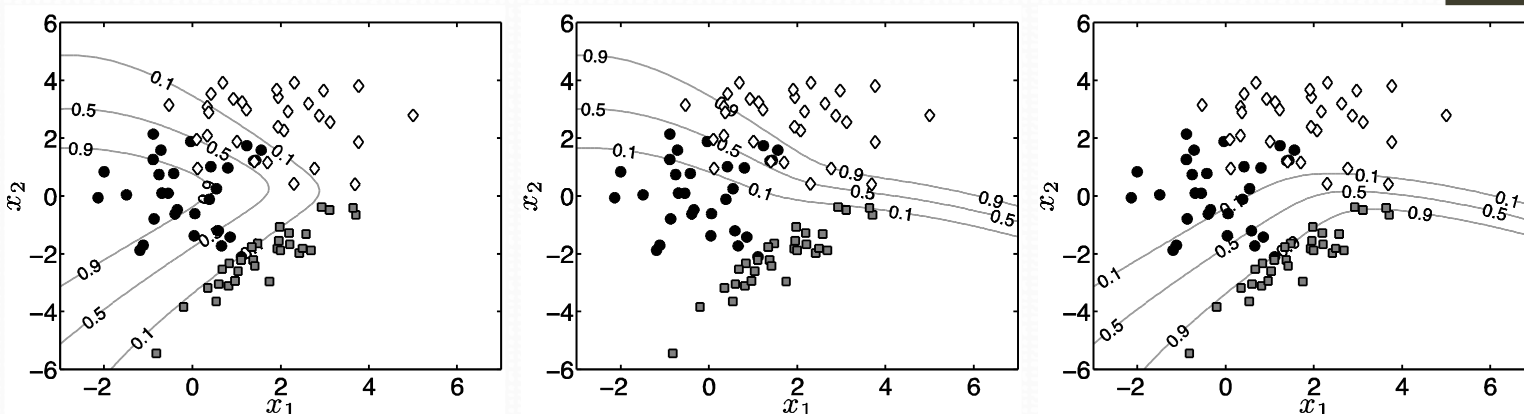

- Density contours, <u>without</u> Naïve Bayes assumption



# Gaussian classification: Naïve Bayes assumption

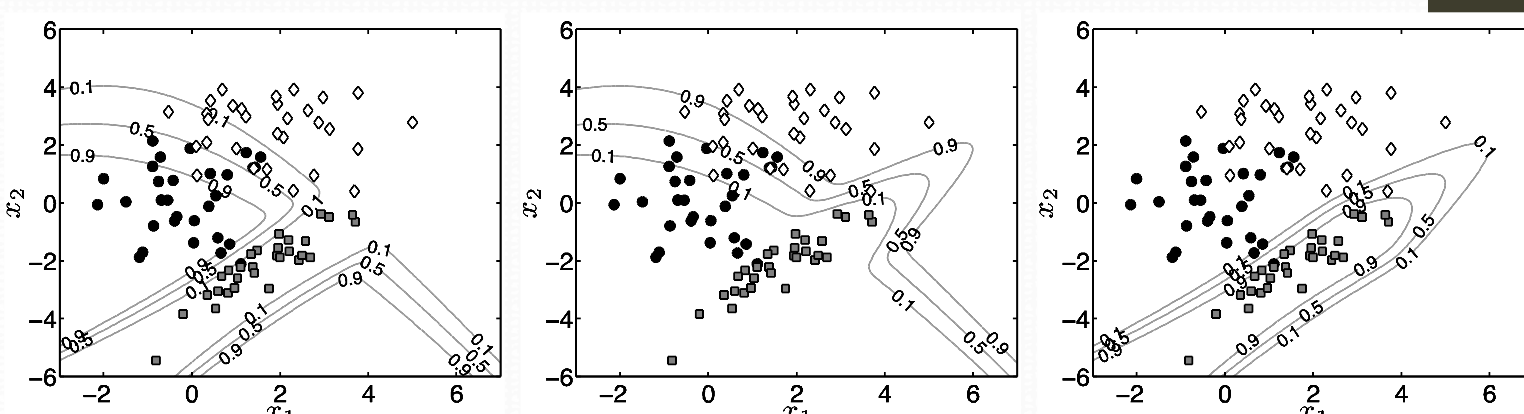
- Classification probability contours

<u>with</u>
Naïve
Bayes
assump
tion




(a) $P(T_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$ (b) $P(T_{\text{new}} = 2|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$ (c) $P(T_{\text{new}} = 3|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$

<u>without</u>
Naïve
Bayes
assump
tion



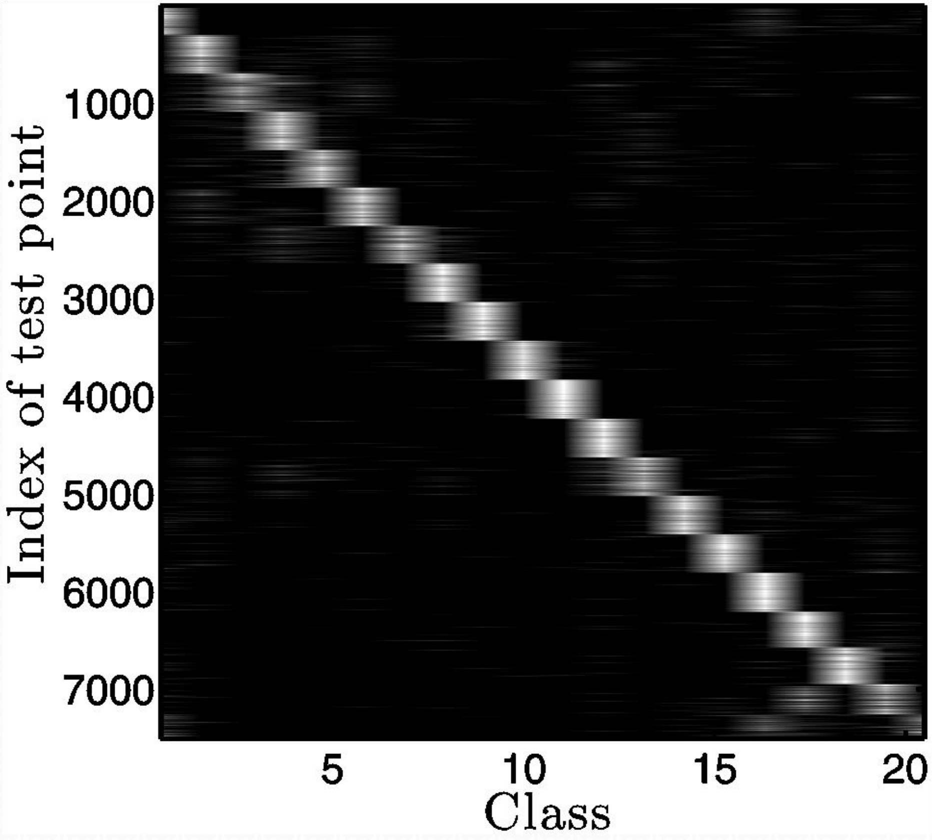

(a) $P(T_{\text{new}} = 1|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$ (b) $P(T_{\text{new}} = 2|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$ (c) $P(T_{\text{new}} = 3|\mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$

# Gaussian classification: Learning to classify text

- Automatic text classification
  - 20 newsgroups dataset consisting of 20,000 documents (covering sports, religion, computing, etc)
  - Each document is a post to one of 20 newsgroups
- Learn to classify a new document to one of these 20 newsgroups
- What are the attributes?
  - Words?
- How to encode the document as a vector of numerical values for classification?
  - *Bag-of-words*
  - *Word frequency*
- With or without Naïve Bayes assumption?

# Gaussian classification: Learning to classify text

- ~11,000 documents used as training dataset
- ~7,000 documents used as testing dataset
- Each test document is assigned a class, which has highest probability out of the 20 probability estimates
  - 78% classification accuracy



# Summary

- Generative approach to classification
- Bayes rule for classification
- Naïve Bayes classifier is a simple but effective classifier for data having several attributes
- Class-conditional likelihood
- Class prior
- Maximum a posteriori Naïve Bayes estimate

# Exercise (ungraded)

- ML – Tom Mitchell: Exercise 6.1
  - Consider the example application (disease diagnosis) of Bayes rule in Section 6.2.1. Suppose the doctor decides to order a second lab test for the same patient, and suppose the second test returns a positive result as well. What are the posterior probabilities of cancer and ¬cancer following these two tests? Assume that the two tests are independent?

# Exercise (ungraded)

- Given training data, train a Gaussian classifier (without Naïve Bayes assumption)

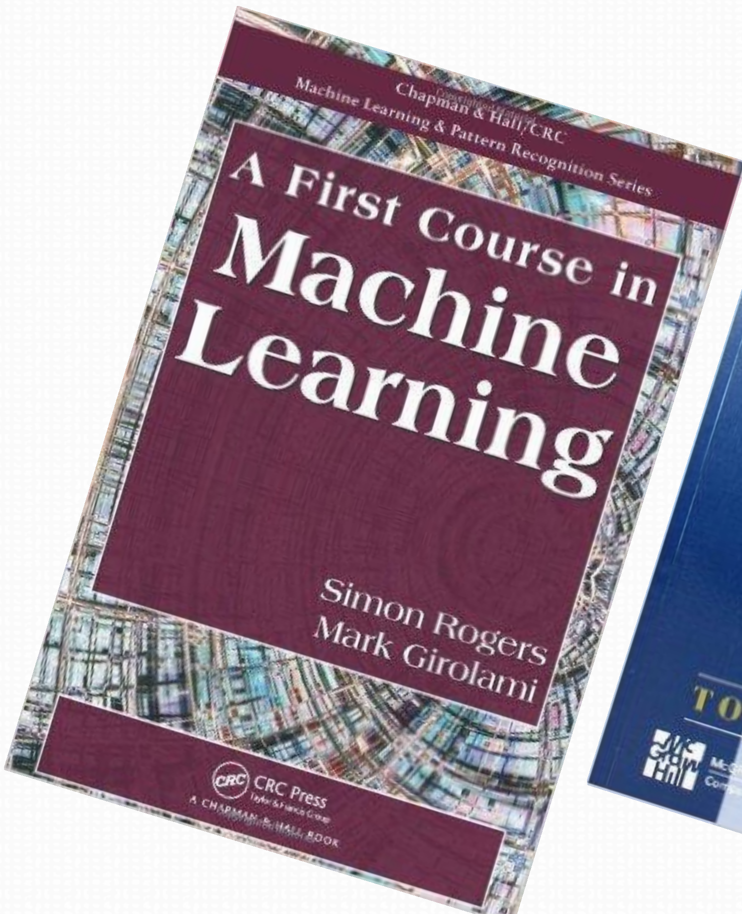
| Attribute 1 | Attribute 2 | class |
|---|---|---|
| $-4$ | $1$ | $1$ |
| $-5$ | $2$ | $1$ |
| $-3$ | $3$ | $1$ |
| $-2.5$ | $4.5$ | $1$ |
| $-4$ | $5$ | $1$ |
| $3$ | $1$ | $2$ |
| $3.5$ | $0$ | $2$ |
| $4$ | $0.5$ | $2$ |
| $4$ | $-1$ | $2$ |
| $3.5$ | $-1$ | $2$ |

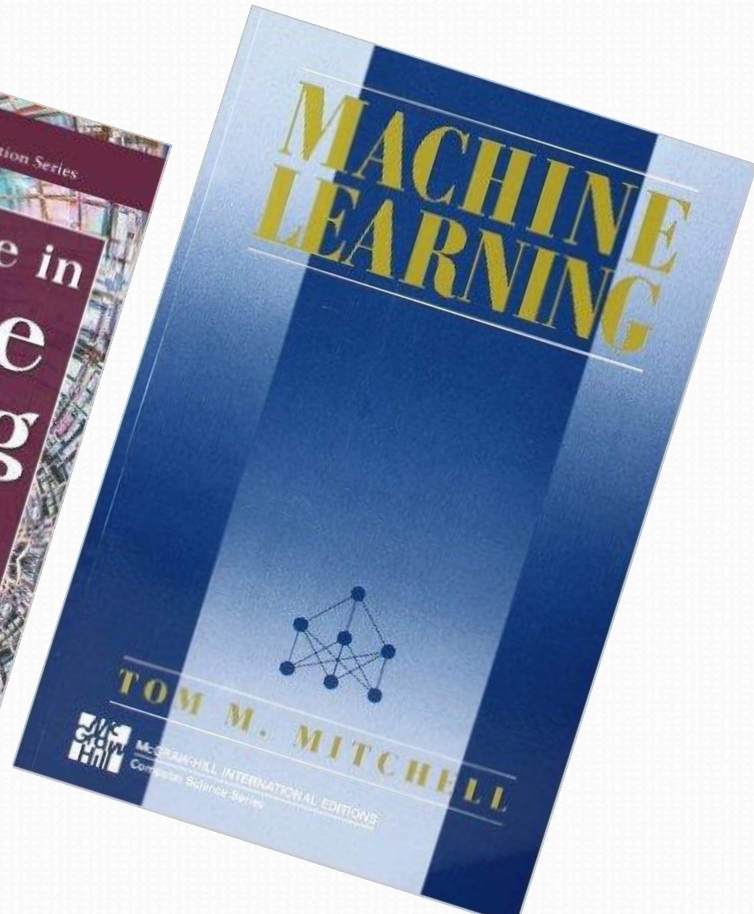
- Predict a new sample by estimating posterior using this Gaussian classifier

| Attribute 1 | Attribute 2 | class |
|---|---|---|
| $-2$ | $2$ | ? |

# Exercise (ungraded)

- Try MATLAB code – plotcc.m (from FCML book website)
- Try MATLAB code – bayesclass.m (from FCML book website)


Author’s material (Simon Rogers)

- Ata Kaban’s material from previous years

That's All

Thank You