Bayesian Learning

Ben Galili
Ariel Shamir
Zohar Yakhini

IDC HERZLIYA

Probabilistic Learning

- Probabilistic Formulation of a classification task
- Bayes rule
- Bayesian Decision Theory
- Minimum error rate classification
- Maximum A-Posteriori & Maximum Likelihood
- Cost considerations
- Statistical dependence and conditional independence
- Naïve Bayes classifiers

Classification

- What do we want classifiers to do?
  → We want them to classify correctly as much as possible
- How do we measure quality/performance?
  → we want the errors to be minimized.
  → To measure this we can often use a probabilistic framework, selecting classifiers that will minimize the probability of error
- In probabilistic learning we will use the training data to infer a probability structure of the data and derive a classifier from there.
- We regard our observations (measurable features in the training data, including the class variable) as random variables, coming from class dependent distributions.

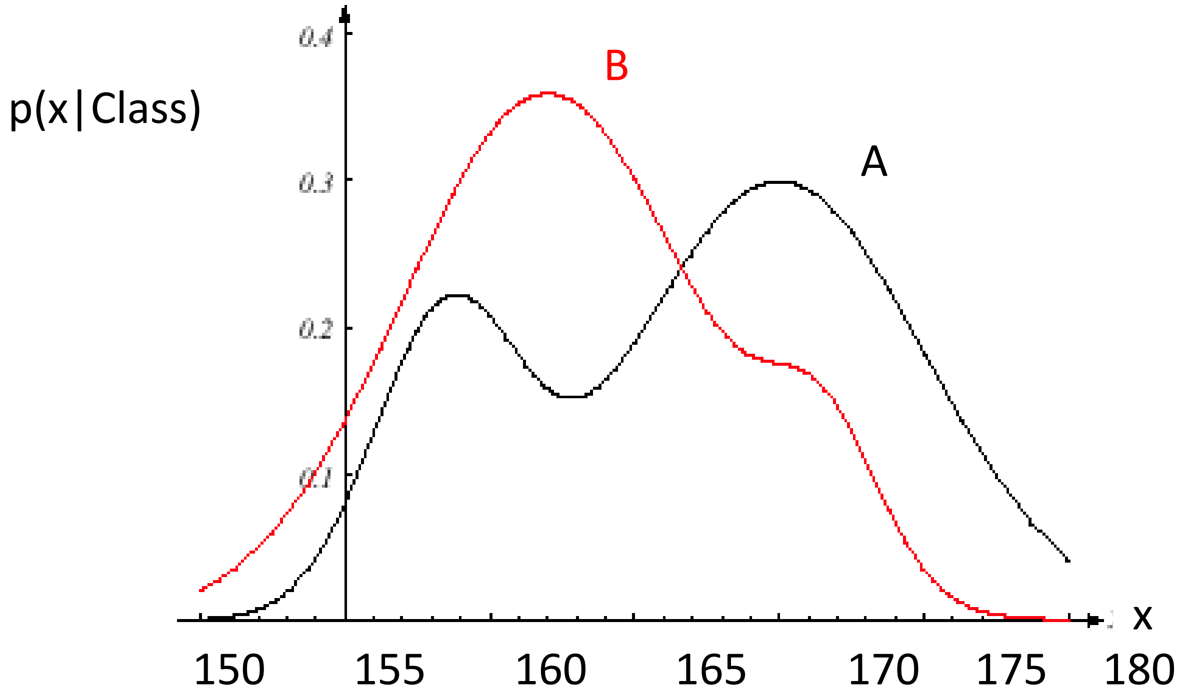
Simple approach: Using only the Prior Probability

- We have two classes A and B
- We know P(A) and P(B)
- How should we classify a new given instance?
- The best classifier:
  - Classify A if P(A)>P(B),
  - Classify B otherwise
- Note this does not use any information we may have about the features x of the observed instance.
- What is the probability of error?

More informed approach:

Use Class Conditional Information
from observed features, say height

Assume that we also know P(x|A) and P(x|B)
Example: P(height|male) and P(height| female)

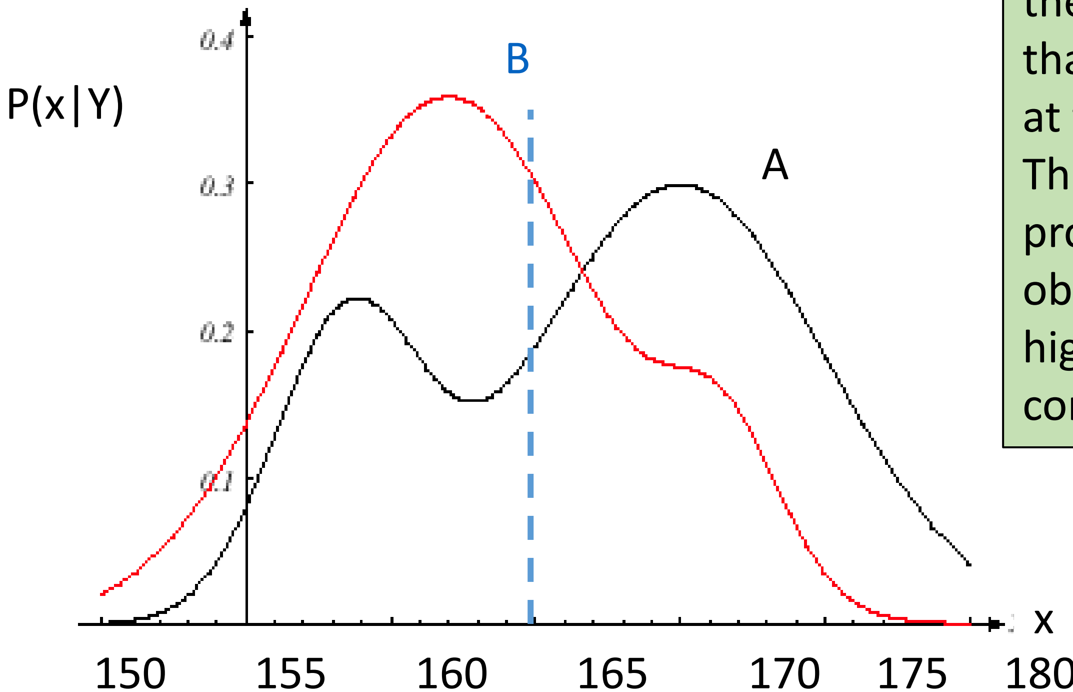


(note: not true data)

Example

Assume x = 163cm, would you say A or B?

- P(x=163| B) > P(x=163| A)



(note: not true data)

The <u>likelihood</u> of B at the observed x is higher than the <u>likelihood</u> of A at the observed x.
The conditional probability of the observation given B is higher than the conditional given A

$P(H>1.9 \mid NBA) = 0.85$

$P(H<1.9 \mid NBA) = 0.15$

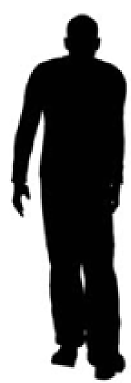
$P(H>1.9 \mid R) = 0.1$

$P(H<1.9 \mid R) = 0.9$

Which is more likely

$P(H>1.9 \mid ?)$

1.93



But we really care about

$P(? \mid H>1.9)$

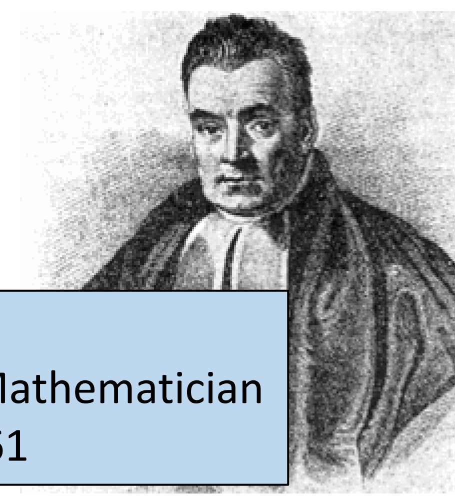
Classification using Likelihood?

- Maybe try the Rule:
  - Classify A if P(x|A)>P(x|B),
  - Classify B otherwise
- Problem?
- What we want is the rule:
  - "Classify A if P(A|x) > P(B|x)"
- Not the same – why? prior probabilities also matter.
- In our M/F example we assumed $P(A) \approx P(B)$. But, in some cases, like in the NBA example, even if P(x|A)>P(x|B) it may be the case that $P(A) \ll P(B)$ (that is: the probability of A is very very small in the first place although the specific value x is much more common in A than in B).

MAP: Maximum A-Posteriori

- So - we want to assess P(A|x) and P(B|x), that is – given x (the observation) we want to know the most probable “true state of nature”.
  Is it A or B?
- Our classifier should be:
  - Classify A if P(A|x)>P(B|x),
  - Classify B otherwise
- However, we do not directly know these ‘posterior’ probabilities!
- The solution:

$$\boxed{P(A|x) = \frac{P(x|A)P(A)}{P(x)}}$$



T Bayes
English Mathematician
1702-1761

Components of the posterior probability formula

Likelihood, or Class Conditional

Prior

posterior

$$P(A|x) = \frac{P(x|A)P(A)}{P(x)}$$

Multi-class Bayes/MAP classifiers

We classify an instance with a feature vector $\vec{x}$ into

$$\boxed{C(\vec{x}) = \underset{i=1..k}{\operatorname{argmax}} \frac{P(\vec{x}|A_i)P(A_i)}{P(\vec{x})}}$$

We can drop $P(\vec{x})$ as it is constant with respect to i:

$$C(\vec{x}) = \underset{i=1..k}{\operatorname{argmax}} \; P(\vec{x}|A_i)P(A_i)$$

The principle of Bayes Classification

- Classification depends both on the class conditional information (the likelihood) and on the prior distribution.
- The binary case:
  - Classify as A if $P(x|A)P(A) > P(x|B)P(B)$
  - Classify as B otherwise
- Note: $P(x)$ is removed from the denominator on both sides because it is the same.
- What if $P(x) = 0$ (such as in continuous distributions)?

Minimum Error Rate Classification

- Whenever we observe a value x, what is the probability of error?
  - *If we decide B then* $P(error \mid x) = P(A \mid x)$
  - *If we decide A then* $P(error \mid x) = P(B \mid x)$
- The Bayes decision is therefore the one that minimizes the probability of error at the observed x
- Using Bayes decision as our model h
  - $P(error \mid x) = \min[P(B \mid x), P(A \mid x)]$
  - *If we really knew the complete probability structure (which we normally don't …) we could estimate:*

$$Error_P(h) = \int P(error \mid x)\, dP(x)$$

Loss = Cost of Wrong Decision

- Assume, as above, that we have k different classes: A_i , $1 \leq i \leq k$
- Upon observing x, we need to assign our instance to one of the A_i s
(and we apply the Bayes/MAP approach)
- Wrong decisions lead to a loss! Loss may depend on which j was misclassified into i. We represent this as a cost function:

$$\lambda_{ij} = \text{Cost}\big(h(x) = A_i \ \wedge \ x \in A_j\big)$$

- For example, a most simple zero-one loss:

$$\lambda_{ij} = \begin{cases} 0 & i = j \\ 1 & i \neq j \end{cases}$$

Minimum Cost of Error Bayes Classification

- Whenever we observe a particular x, what is the expected risk of classifying into A_i , under a general cost function?:

$$R(Choose\ A_i|x) = \sum_{j \neq i} \lambda_{ij} P(A_j|x)$$

- The Bayes cost based decision will be the one that minimizes this cost of error
- That is – in general, having observed x we classify it into

$$\underset{i}{\operatorname{argmin}} \sum_{j \neq i} \lambda_{ij} P(A_j|x)$$

Bayes MAP Classifiers

Under the zero-one loss function:

$$C(\vec{x}) = \underset{i=1..k}{\operatorname{argmax}} \; P(A_i|\vec{x}) = \underset{i=1..k}{\operatorname{argmax}} \; P(\vec{x}|A_i)P(A_i)$$

Under a general cost function:

$$C(\vec{x}) = \underset{i=1..k}{\operatorname{argmin}} \; \sum_{j \neq i} \lambda_{ij} P(\vec{x}_j|A_j)P(A_j)$$

How To Evaluate the Conditional Probabilities/Densities

- In general – how do we estimate distributions?
- We can use the training set to compute a histogram of values for features per class. We can then use the histograms as estimates of the conditional probabilities.
- We can also infer a model, as we discussed last time, using, for example, MLE.
- Note – we need to infer class dependent models. Parameters may (should) be different for each class.

Example: Fisher's *Iris* Data Set

R.A. Fisher
British statistician
and geneticist
1890-1962

- Fisher's Iris data set is a multivariate data set introduced by Ronald Fisher in his 1936 paper:
  The use of multiple measurements in taxonomic problems
- Became a typical basic test case for many statistical classification techniques in machine learning

Fisher's *Iris* Data Set

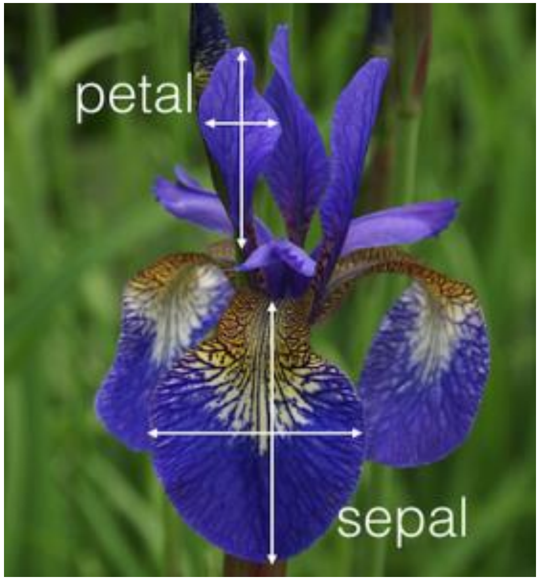
- 50 samples from each of three species of Iris:
  Iris setosa, Iris virginica and Iris versicolor.
- Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.

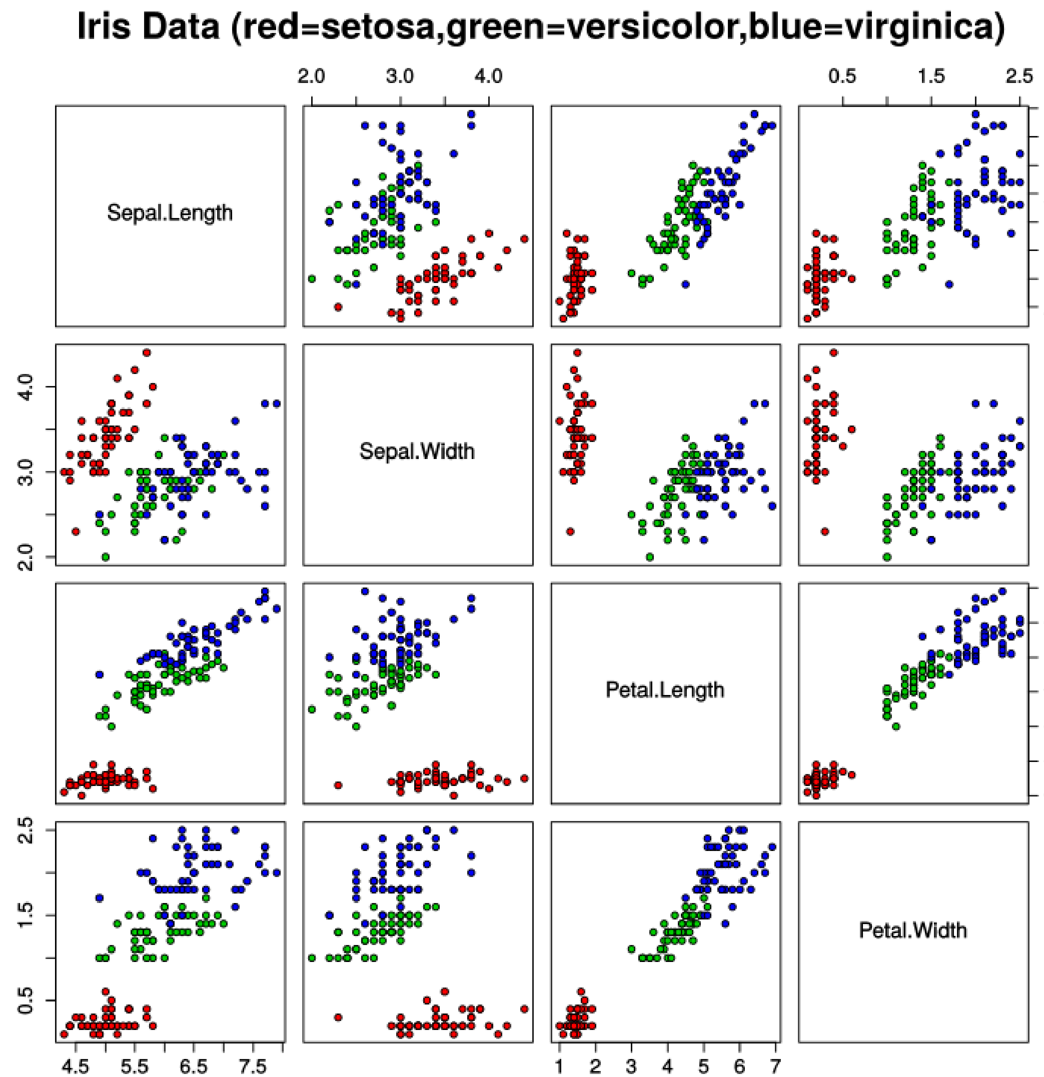
setosa

versicolor

virginica

The Data in a Scatter Plot





Evaluating class conditional probabilities/densities in the Fisher Iris Dataset

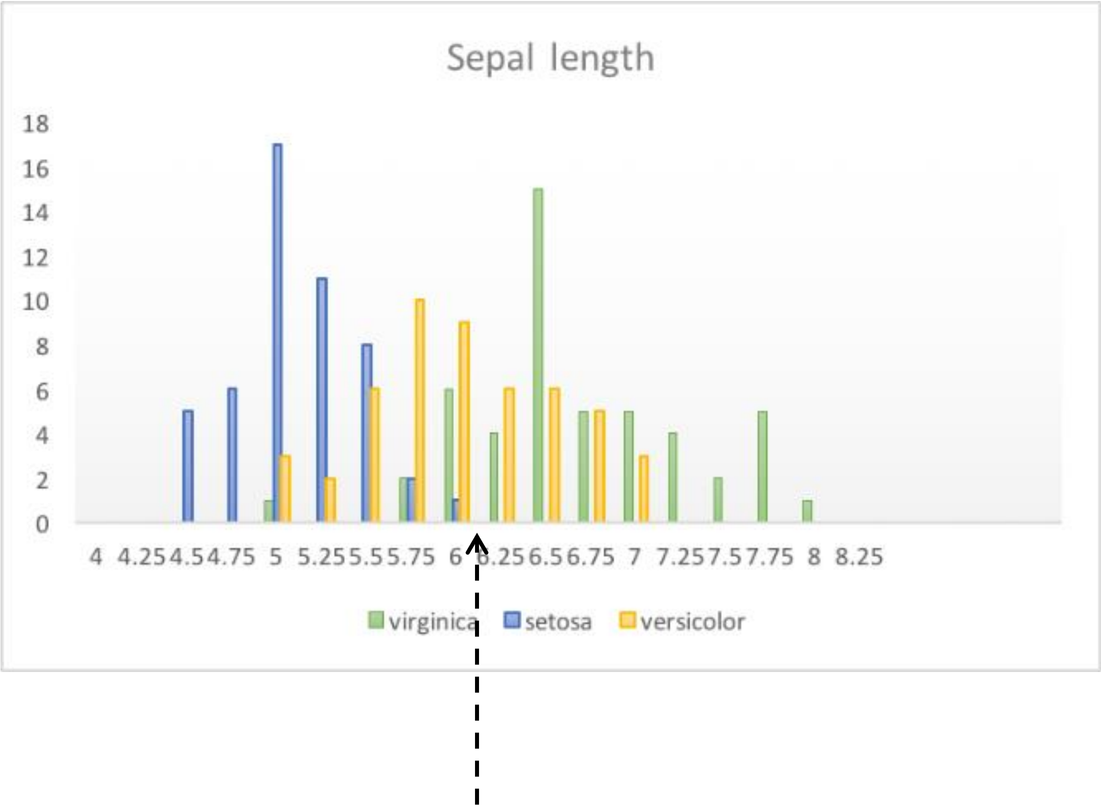
$$P(\vec{x}|A_i) = \ ?$$

Option1: use the actual data

P(sepal length = x | Setosa)

= (count of Setosa w sepal length = x)/(total Setosa)

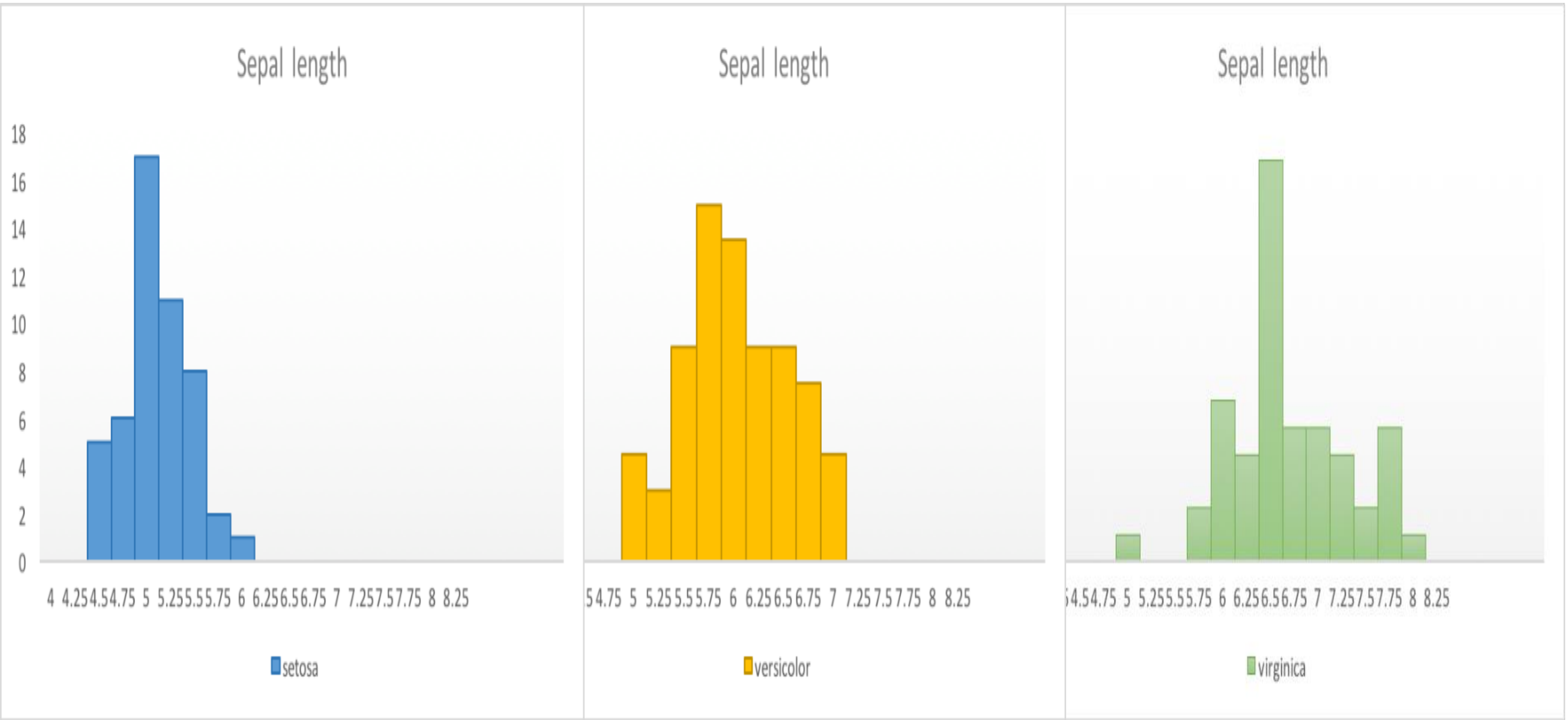
Sampling – information from data only …



$$P(x|A_i) = 0\ ?$$

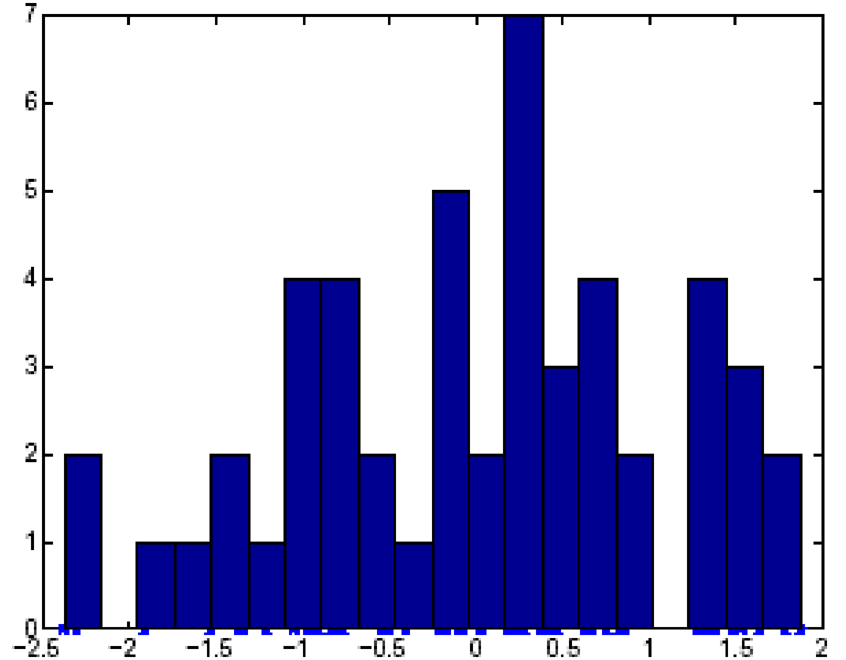
What if the sepal length of a new instance is 6.1?

Option2: Histograms



A Histogram as Density Estimation (Binning)

- In 1D we have m real values and we divide the real line into k non-overlapping bins: $[c_i\text{-}h, c_i+h)$, $i = 1 \dots k$
- There are different approaches for determining k

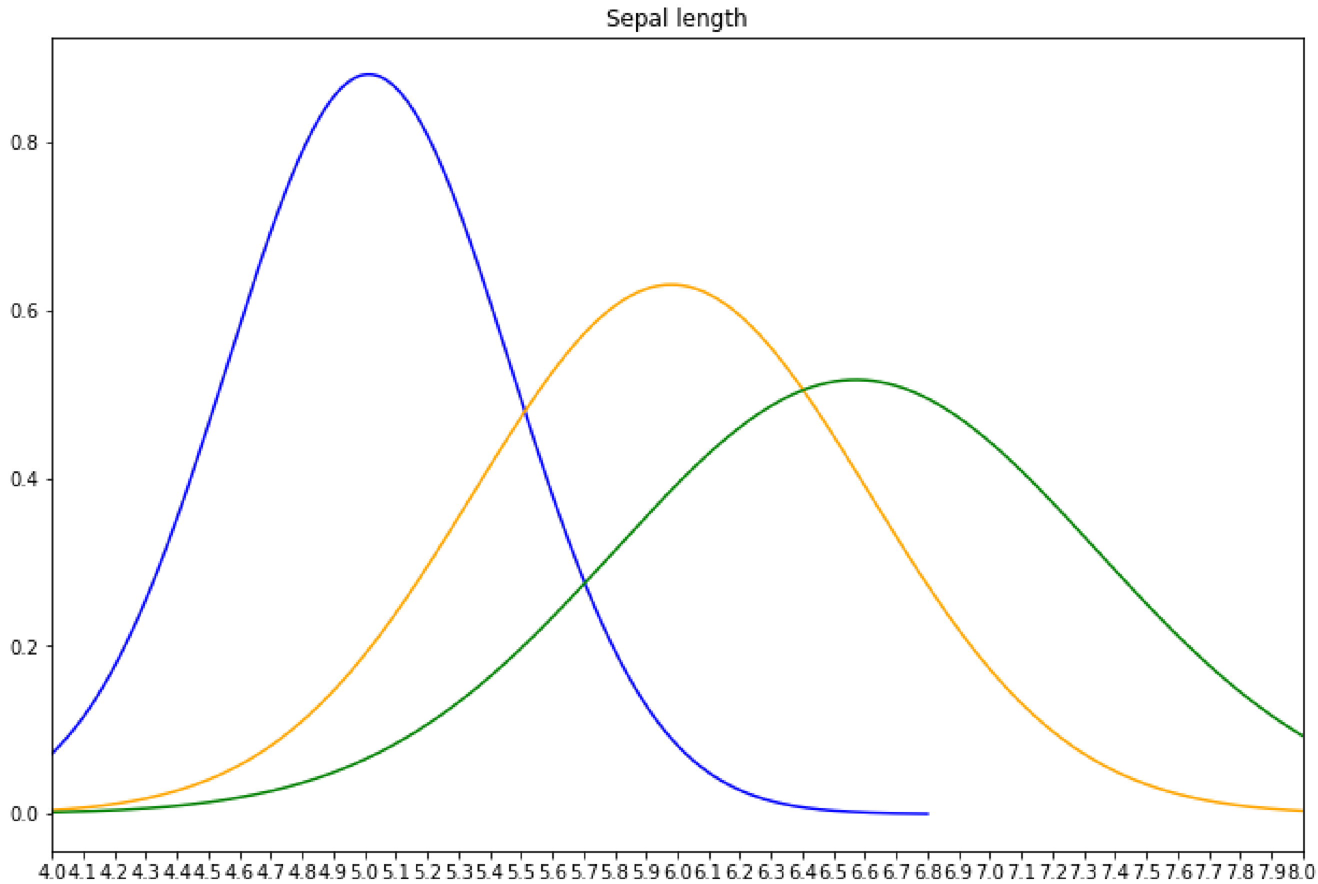


- The resulting density estimate will be:

$$p(x) = \frac{\{number\ of\ samples\ in\ the\ bin\ containing\ x\}}{\{total\ number\ of\ samples\}}$$

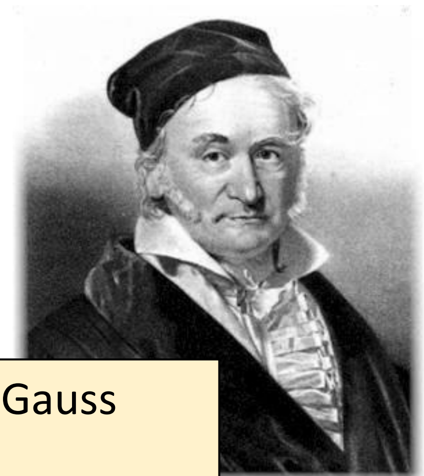
Option 3: parametric approximation

For example: Normal …



Normal Distribution: Parameters

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

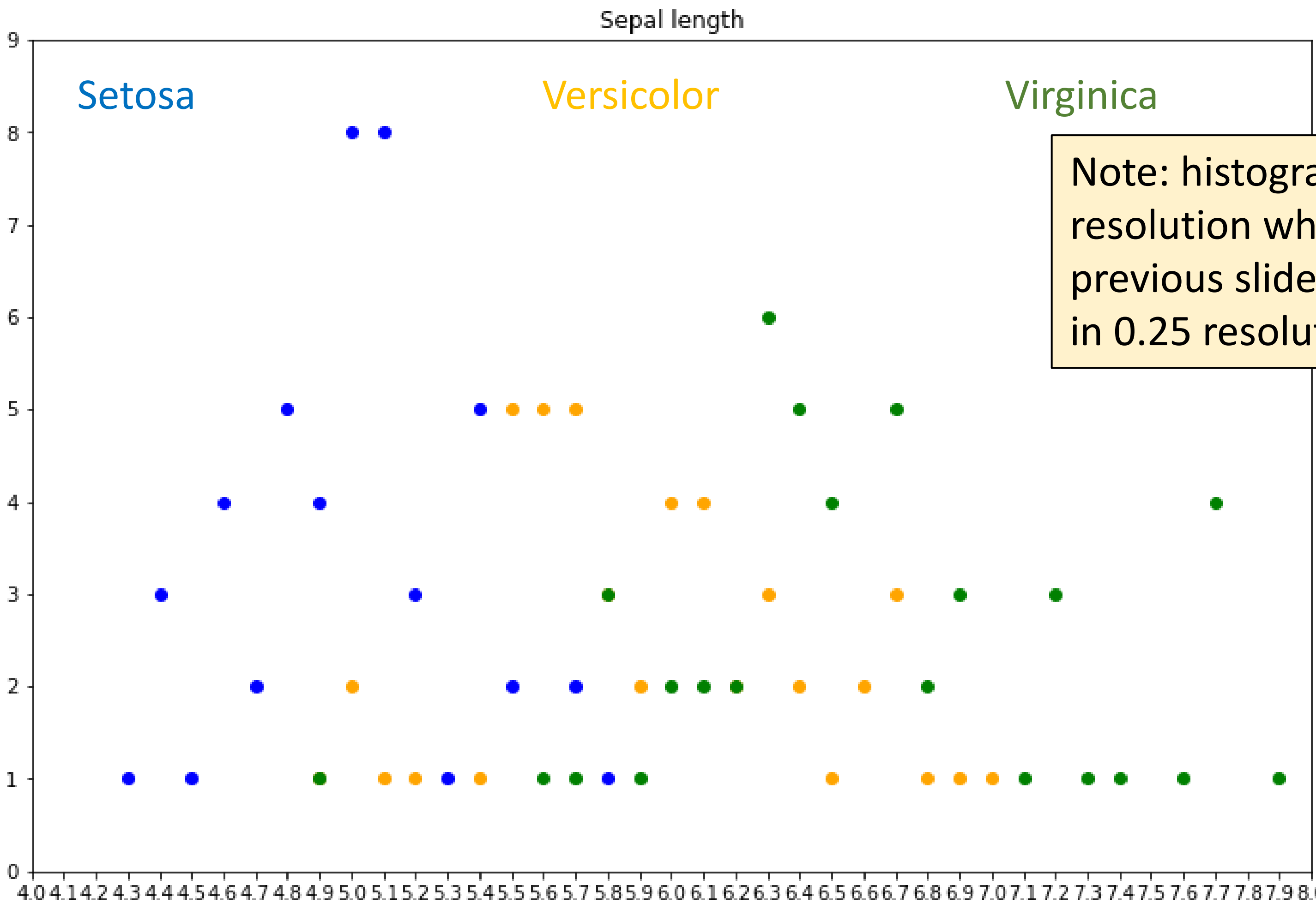


Carl Friedrich Gauss
1777-1855
German Mathematician

- Normal/Gauss distributions are determined by two parameters: μ and σ.
- Given m values of a normal variable X, the MLE estimates for the mean and variance of X are:

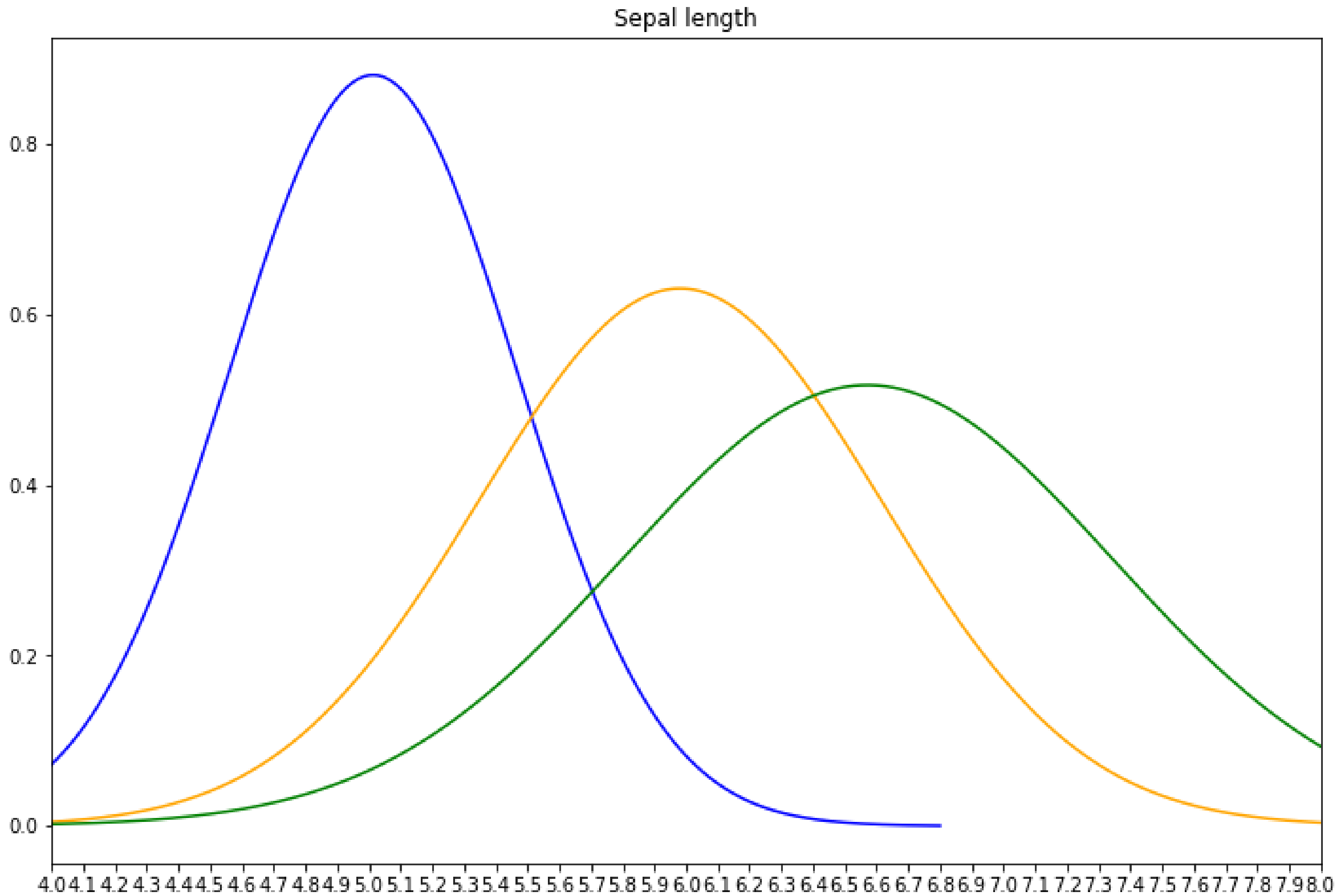
$$\hat{\mu} = \frac{1}{m}\sum_{k=1}^{m} x_k \quad ; \quad \hat{\sigma}^2 = \frac{1}{m}\sum_{k=1}^{m}(x_k - \mu)^2$$

Conditional distribution of sepal length



Note: histogram in 0.1 resolution whereas in previous slides it was in 0.25 resolution

Normal Conditional Distributions using MLE



Back to Fisher's irises

- Assume we measured the sepal length of a specific flower and we get 5.2cm.
  Which of the three species is it?
- Using MAP we are looking for the larger of
  - P(versicolor |sepal length = 5.2)
  - P(virginica |sepal length = 5.2)
  - P(setosa |sepal length = 5.2)
- We now use the Bayes formula to compute these

Using Bayes

- P(versicolor |sepal length = 5.2) =

= P(sepal length = 5.2|versicolor) P(versicolor) /P(sepal length = 5.2)

- P(virginica |sepal length = 5.2) =

= P(sepal length = 5.2|virginica) P(virginica) /P(sepal length = 5.2)

- P(setosa |sepal length = 5.2) =

= P(sepal length = 5.2|setosa) P(setosa) /P(sepal length = 5.2)

But since we assumed that the priors are the same

P(versicolor) = P(virginica) = P(setosa)

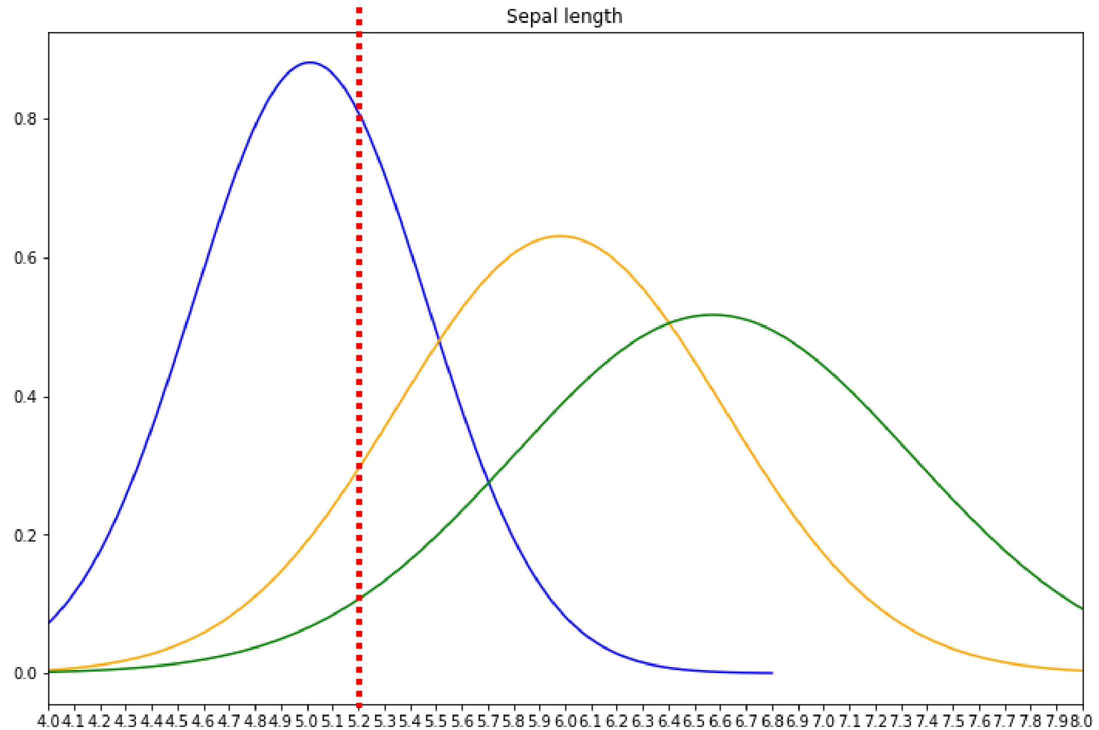
we just use the likelihoods for classification

Compare Class Conditional Probabilities; The Gauss version

Which one is larger?

1. P(sepal length = 5.2|versicolor)
2. P(sepal length = 5.2|virginica)
3. $\boxed{\text{P(sepal length = 5.2|setosa)}}$

What is the advantage of this approach over a histogram approach?



How to Measure Classification Performance?

- How do we know if we managed to build a good classifier?
- We can measure the error rate on the training set – count the misclassified examples (43/150 = 29%)
- In the Bayes classifier approach we only learned the distributions from the data.
  We may still suffer from overfitting.
  We need to use a “test set”, one not used in the learning process.
- Also - misclassification represents six types of errors:
  1. versicolor classified as virginica
  2. versicolor classified as setosa
  3. virginica classified as versicolor
  4. virginica classified as setosa
  5. setosa classified as virginica
  6. setosa classified as versicolor

Confusion Matrix

Classified Species

True Species

	versicolor	virginica	setosa
versicolor	31 (20%)	14 (9%)	5 (3%)
virginica	12(8%)	37 (25%)	1 (0.7%)
setosa	11 (7.3%)	0 (0%)	39 (26%)

Wrong classification - error

Correct classification

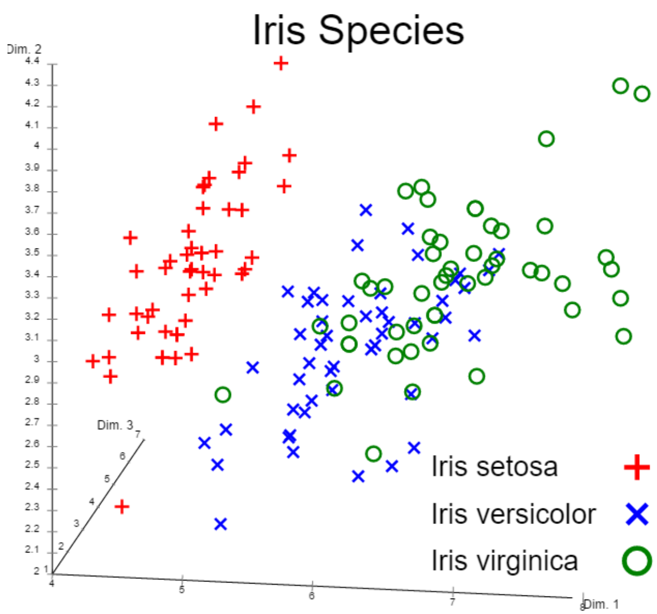
What can be learned from the matrix?

Cost considerations?

Can we compute the expected cost of the classification?

Multi Dimensional Feature Spaces

- Each instance observed consists of many features
- Assume we have
  d features
  and
  m instances
- That is: our training data consists of m labeled instances of the form
  $$\vec{x} = (x_1, x_2, \ldots, x_d)$$
- There may be dependencies between the features
- For instance, the width and the length (both sepal and petal) are not totally independent



What are d and m here?

We will now want to estimate the higher dimensional conditional distributions:

$$P(\vec{x}|A_i) = P\big((x_1, x_2, \dots, x_d)|A_i\big)$$

Multivariate distributions - a refresher …

- Rolling two dice is a multivar distribution. Our distribution is defined over the space of all pairs $(i, j), i = 1 \ldots 6$ and $j = 1 \ldots 6$.
- When we assume two independent fair dice then the probability distribution function is uniform over all 36 possible outcomes.
- Can you construct a distribution over all pairs so that the induced distribution for each individual die is fair (uniform) but that over the pairs is not uniform?
- The dbns for the two individual dice are called <u>marginals</u>.
- The same marginals can be coupled into many different joint distributions

The normal density function

Again - density functions for Gaussian r.vs:

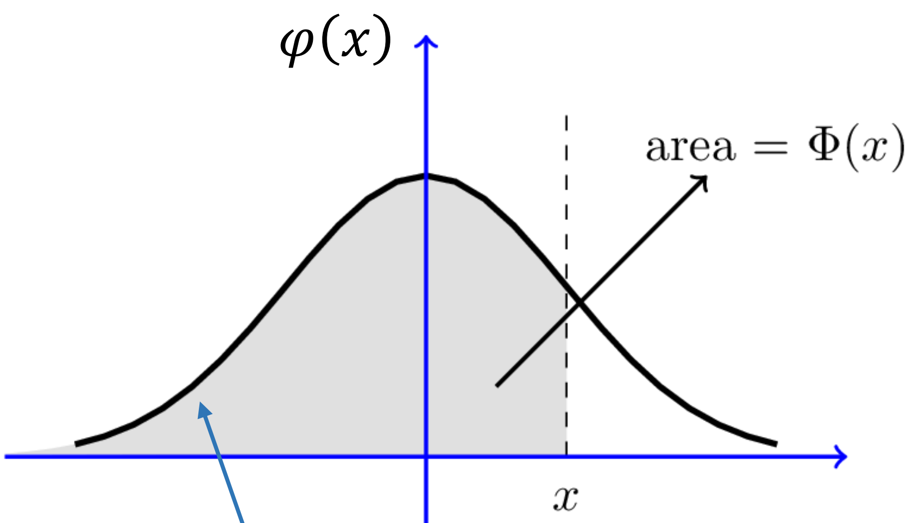
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

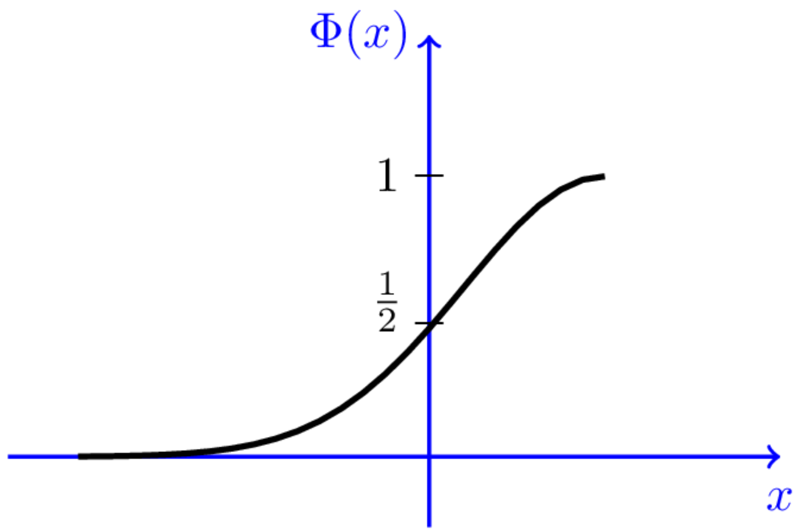
We then say that the r.v X is normally distributed with mean μ and standard deviation σ.
We write $X \sim N(\mu, \sigma)$

A random variable that has a normal distribution with
$\mu = 0$ and $\sigma = 1$
is called <u>Standard Normal</u>.
The density function then becomes:

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$$

The CDF of a standard normal is often called Φ





$$\varphi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$$

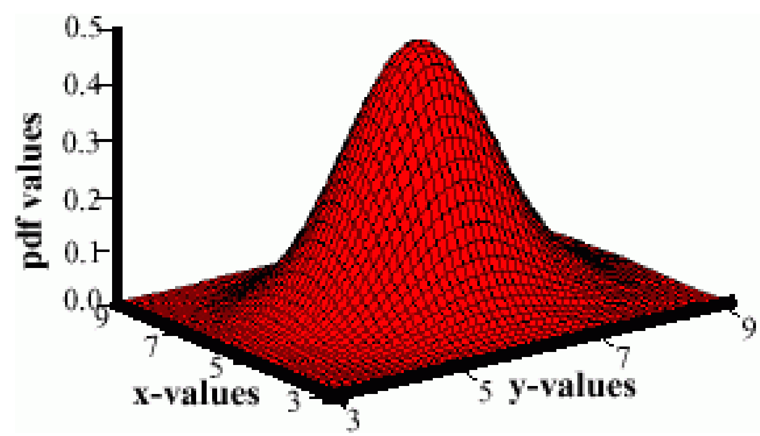
Multivariate Normal Distributions

- **A multivariate normal distribution is defined by its (multi D) pdf:**

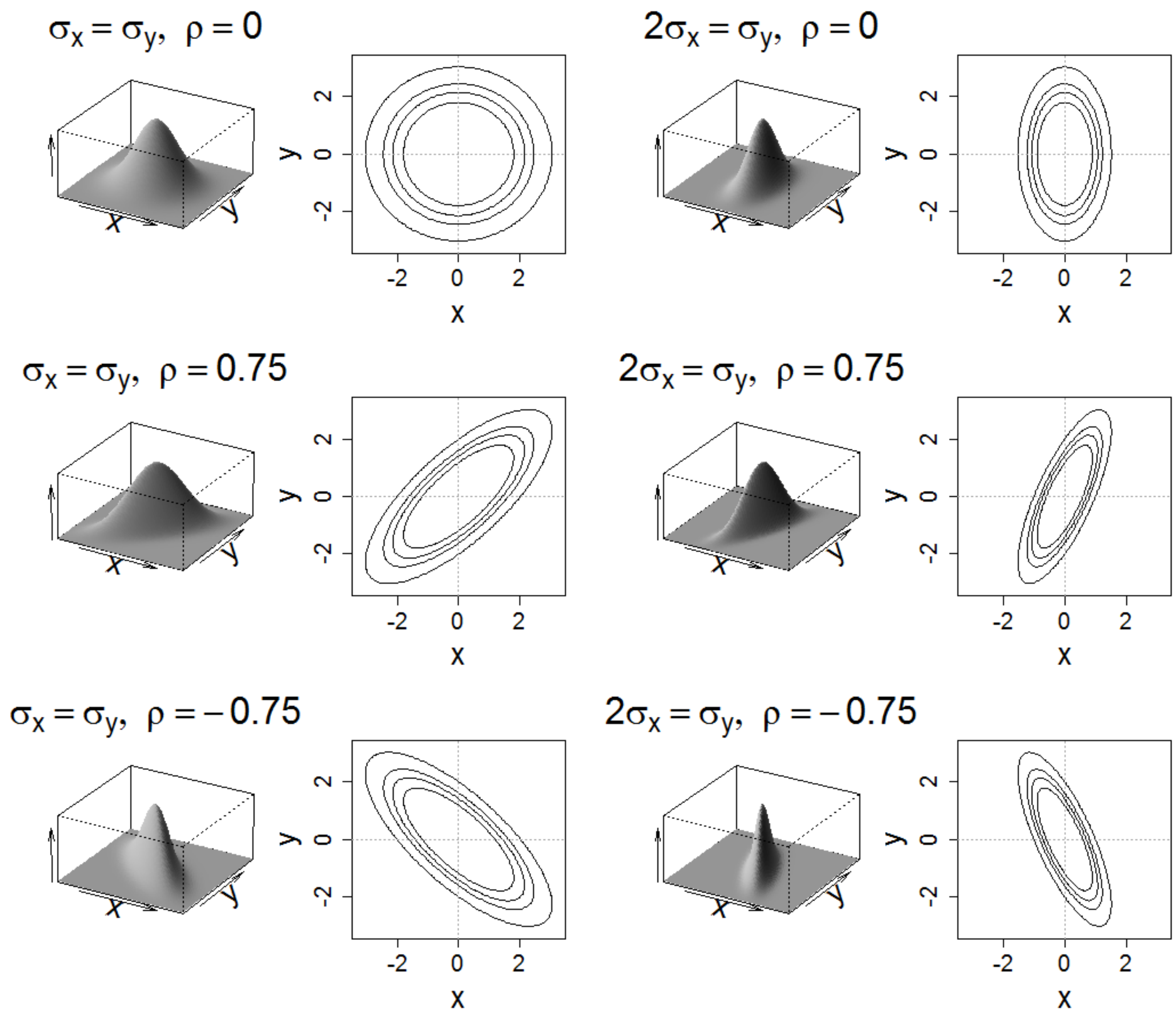
$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp[\frac{-1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})]$$

where μ represents the mean (vector) and Σ represents the covariance matrix.

- **The covariance is always symmetric and positive semidefinite.**
- **How does the shape vary as a function of the covariance?**
- **Will be further discussed next week**



2D joint Gaussians



So far ....

$$\mathrm{C}(\vec{x}) = \ \operatorname{argmax}_{i=1\ldots k}\ \{P(\vec{x}|A_i)P(A_i)\}$$

$$\Downarrow$$

Need to estimate $P(\vec{x}|A_i)$ and $P(A_i)$

- Estimating Probabilities and Densities:
  - parametric vs. non-parametric or data based
  - For example: Gauss vs Histogram
- Estimating in 1D or in higher dimensions;

Parametric Bayes classification

- We can use a (multidimensional) parametric model (e.g Gaussian) that is learned for each one of the classes separately to assess the likelihoods.
- Important: this approach allows us to classify an instance with feature values that we have not seen in the training. The entire feature space is covered.
- Disadvantages?

Multidimensional classification

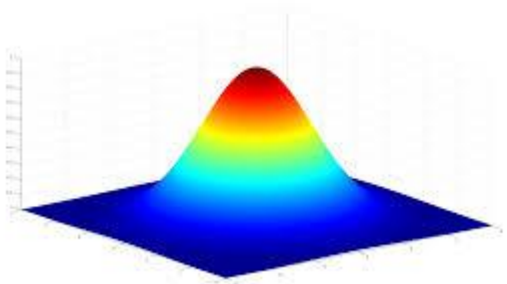
- We have a multiclass classification task with k classes $A_1 \ldots A_k$
- Each instance $x \in X$ is described by a set of attributes $x = (x_1, x_2, \ldots, x_d)$ with $x_j \in V_j$ where V_j is the space of possible values attainable by feature j.
  (These can be $\mathbb{R}$ or some other infinite space or maybe finite discrete sets)

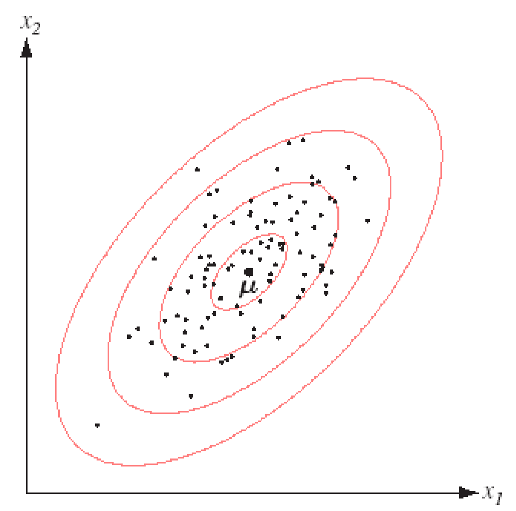
- Given x and using MAP we classify x into:

$$
\begin{aligned}
v_{MAP} &= \arg\max_i P\left(A_i \mid (x_1, x_2, \ldots, x_d)\right) = \\
&= \arg\max_i \frac{P\left((x_1, x_2, \ldots, x_d) \mid A_i\right) P(A_i)}{P\left((x_1, x_2, \ldots, x_d)\right)} \\
&= \arg\max_i P\left((x_1, x_2, \ldots, x_d) \mid A_i\right) P(A_i)
\end{aligned}
$$

Estimating Gaussian distributions in high dimensions

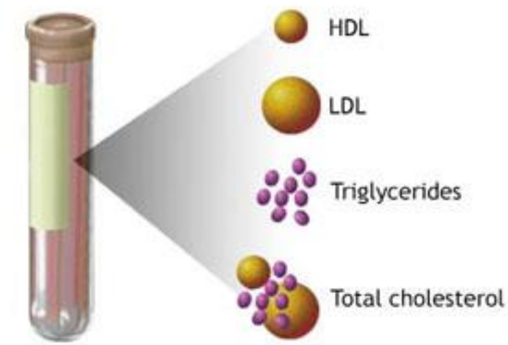
- Samples drawn from a normal population tend to fall in a single cloud or cluster whose center is determined by the vector of means and shape by the covariance matrix
- The mean can be estimated easily, but estimating the covariance requires us to learn $d(d+1)/2$ parameters.

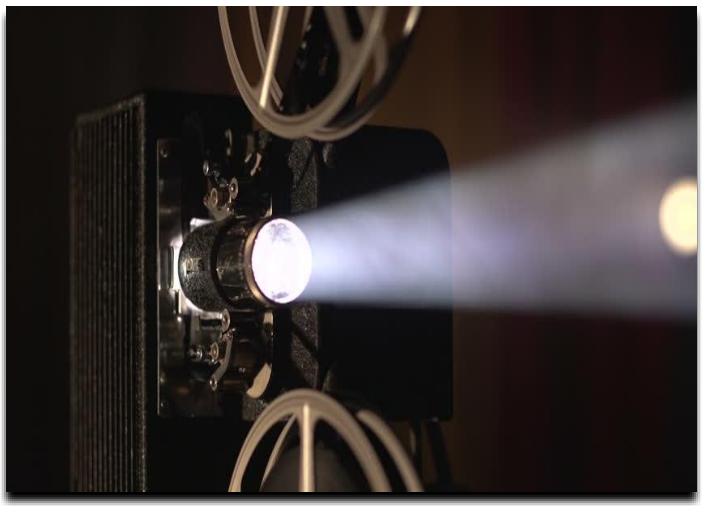




<u>Conditional</u> independence

- Is the blood cholesterol level of a person independent of the number of movies watched by that person so far?
- No – they are both related to the age of the person.
- But – they are conditionally independent given the age.
- Presumably …, socioeconomic and behavioral factors ignored …
- Notation: $X \perp Y \mid C$





Conditional Independence - Definition

The features are conditionally independent given the class if
<u>for all</u> relevant multidimensional feature values $(\vec{x})$ AND
<u>for all</u> possible classes values (i),
we have:

$$P\left(\left(x_1, x_2, \ldots, x_d\right) \mid A_i\right) = \prod_{j=1 \ldots d} P\left(x_j \mid A_i\right)$$

Naïve Bayes – the Conditional Independence Assumption

- Naïve Bayes Classification makes the useful simplifying assumption that feature values are conditionally independent given the class.
- Is this always true?
  Example in the HWA

Naïve Bayes classifiers

Classify an instance with observed properties $\vec{x}$ as

$$\underset{i}{\operatorname{argmax}} P(A_i)P(\vec{x}|A_i) =$$

$$\underset{i}{\operatorname{argmax}} P(A_i)\prod_{j=1}^{d} P(x_j|A_i)$$

Note: the first step in using Naïve Bayes Classifiers is to estimate the conditional distributions for all single features and all classes

We will do a use case example in the HW

Naïve Bayes vs Full Bayes

- Naïve:

$$C(\vec{x}) = \underset{i}{\operatorname{argmax}} P(A_i) \prod_{j=1}^{d} P\left(x_j | A_i\right)$$

Learn the conditional marginals from the data.

Learn from the data in the same way.

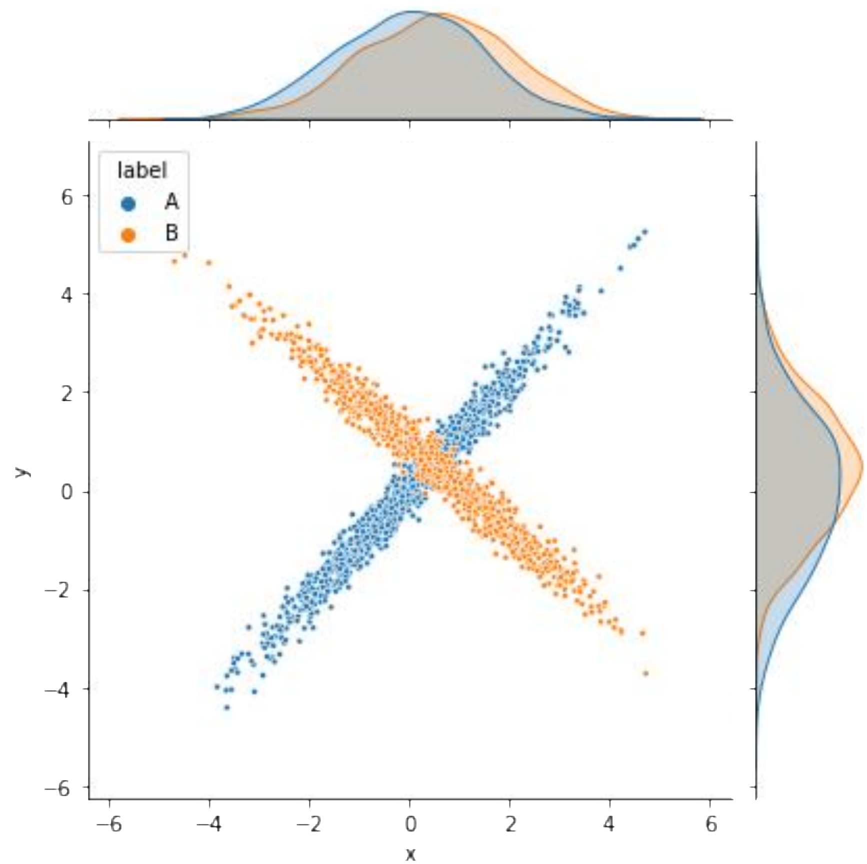
- Full:

$$C(\vec{x}) = \underset{i}{\operatorname{argmax}} P(A_i) P(\vec{x} | A_i)$$

Learn the full multivariate conditional from the data.

Example of Naïve vs Full

- Each class is a bivariate Gaussian
- The difference is that the 'A' class has positive cov and the 'B' class has negative cov
- This is the full Bayes point of view



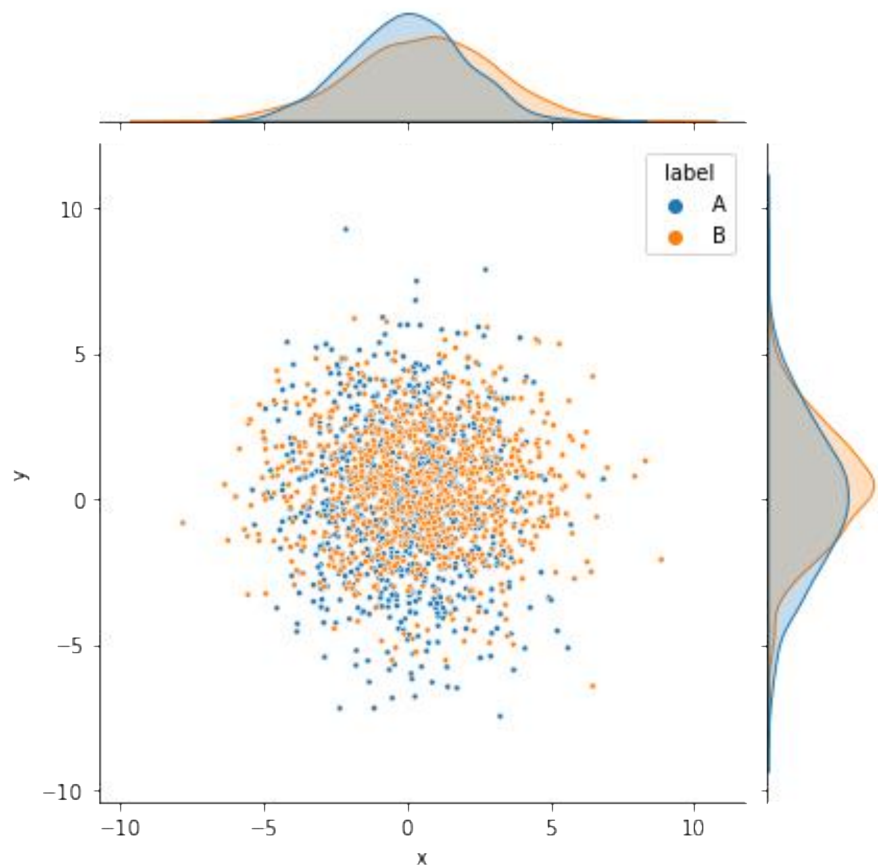
mean_A = [0, 0], cov_A = [[2, 2.2], [2.2, 2.5]]

Mean_B = [0.5, 0.5], cov_B = [[2.5, -2.2], [-2.2, 2]]

* The marginals (in all related slides) are extrapolated using kde (Pandas)

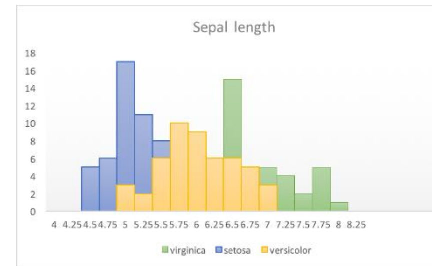
Example of Naïve vs Full

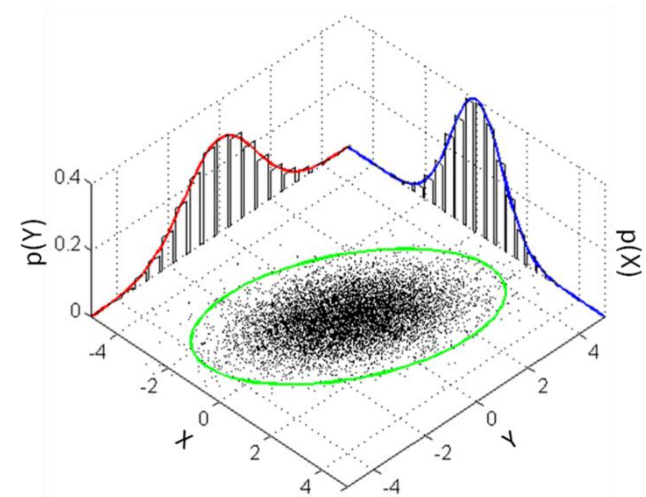
- We now use the same marginals but we also assume conditional independence
- Then recreate the data
- This gives us a visualization of the Naïve Bayes point of view



How to estimate probabilities and densities? (for MAP or for other applications)

- Approach 0: sampling, data
- Approach 1: histograms
  - Problem: do we have sufficiently many samples (especially in high dimensions)?
- Approach 2: parametric (e.g. Gaussian)
  - Advantages: robust models, compact storage, interpretation
  - Problem: are there any valid parametric model assumptions?
- Approach 3: Naïve Bayes
  - Resolves the complexity of estimating high dimensional densities
  - Also resolves similar issues for discrete spaces.
  - Problem: based on simplifying assumptions that are not necessarily true (but … see epilogue of this lecture)





Different Bayes Classifiers

- MAP $\Rightarrow \arg\max_i P(\mathrm{A_i} \mid \mathbf{x}) = \arg\max_i \dfrac{P(\mathbf{x} \mid \mathrm{A_i}) P(\mathrm{A_i})}{\sum_{j=1}^{k} P(\mathbf{x} \mid \mathrm{A_j}) P(\mathrm{A_j})}$
- Dropping $P(\mathbf{x}) \Rightarrow \arg\max_i \{P(\mathbf{x} \mid \mathrm{A_i}) P(\mathrm{A_i})\}$
- ML - Assuming $P(A_i) = P(A_j) \Rightarrow \arg\max_i \{P(\mathbf{x} \mid \mathrm{A_i})\}$
- Using log probability $\Rightarrow \arg\max_i \{\ln P(\mathbf{x} \mid \mathrm{A_i}) + \ln P(\mathrm{A_i})\}$
- Naïve Bayes - assuming $P(\vec{\mathbf{x}} \mid \mathrm{A_i}) = \prod_j P(x_j \mid \mathrm{A_i}) \Rightarrow$

$$\arg\max_i \{P(\mathrm{A_i}) \prod_j P(x_j \mid \mathrm{A_i})\}$$

Summary

- We are interested in minimizing the overall risk in classification, under a probabilistic model set-up.
- The Bayes decision theory approach, under a 0/1 cost model, states that you should choose the action (classification) that minimizes the probability of error.
- This translates to maximizing the posterior probability MAP: $\underset{i}{\operatorname{argmax}}\, P(A_i|\vec{x})$
- Since we do not know this posterior probability we use Bayes Rule …
- This generalizes to other cost functions

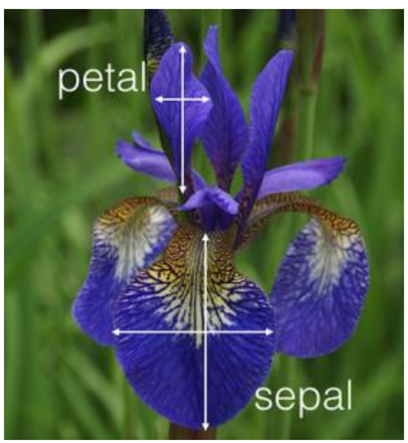
Summary – cont

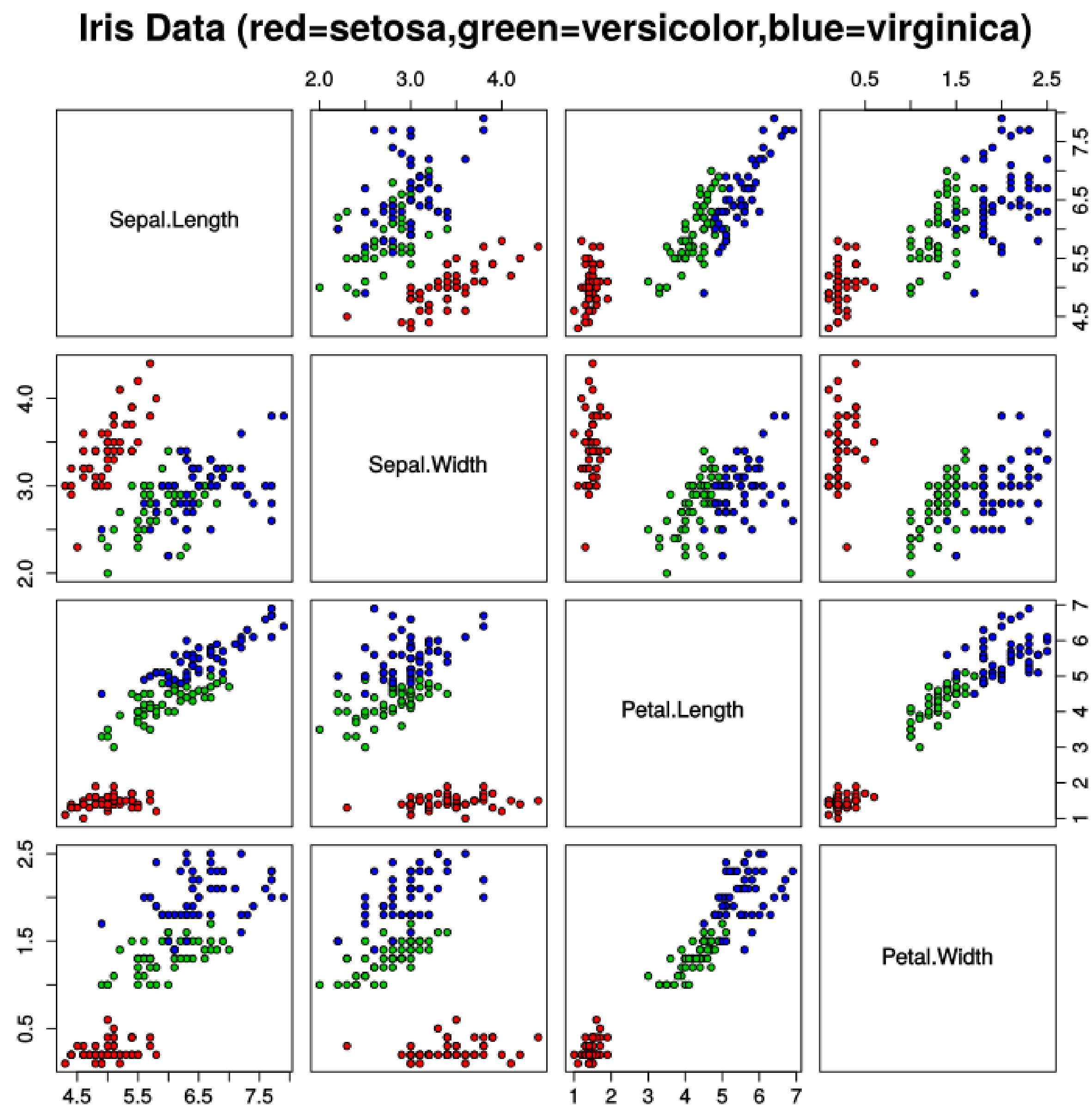
- We can use data to estimate class distributions and class conditional feature distributions for all classes
- A Gaussian estimate is often useful
- Estimates of probability densities (MLE) are useful in the context of classification and in other learning contexts
- Multivariate Gaussians and the covariance matrix
- Conditional independence
- Naïve Bayes Classification – uses conditional independence assumptions.
- Additional topics (next week):
  - Estimates in finite distributions and Laplace smoothing
  - The effect of the cost function
  - Multivar Gaussians, GMMs, EM

Epilogue on the Naïve assumption

Fisher's Iris Data

Which features are (close to) conditionally independent given the class?





Correct Vs. Practical

- Often the naïve assumption is violated (example: petal width and height):

$$\hat{P}(x_1, x_2 \ldots x_n \mid A_j) \neq \prod_i \hat{P}(x_i \mid A_j)$$

- However, in practice, this estimator works surprisingly well.
- Note that, in actuality, we do not need this assumption to be true.
  We just need the following to be true:

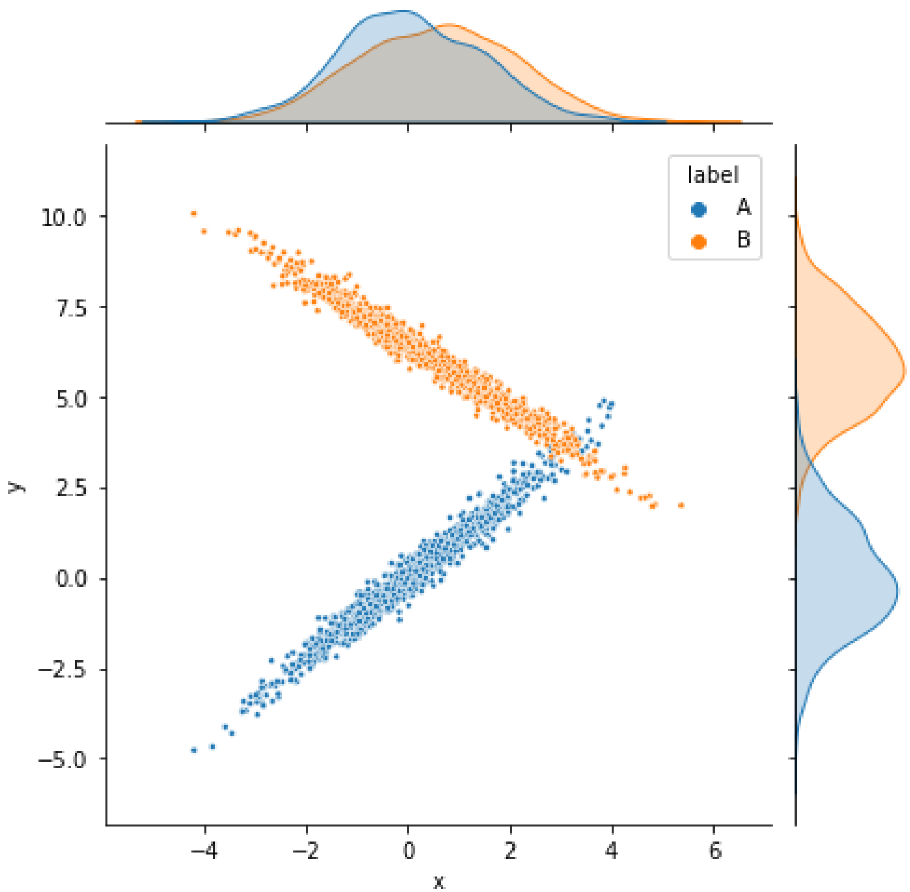
$$\boxed{\arg\max_j \hat{P}(A_j) \prod_i \hat{P}(x_i \mid A_j) = \arg\max_j \hat{P}(A_j) \hat{P}(x_1 \ldots, x_n \mid A_j)}$$

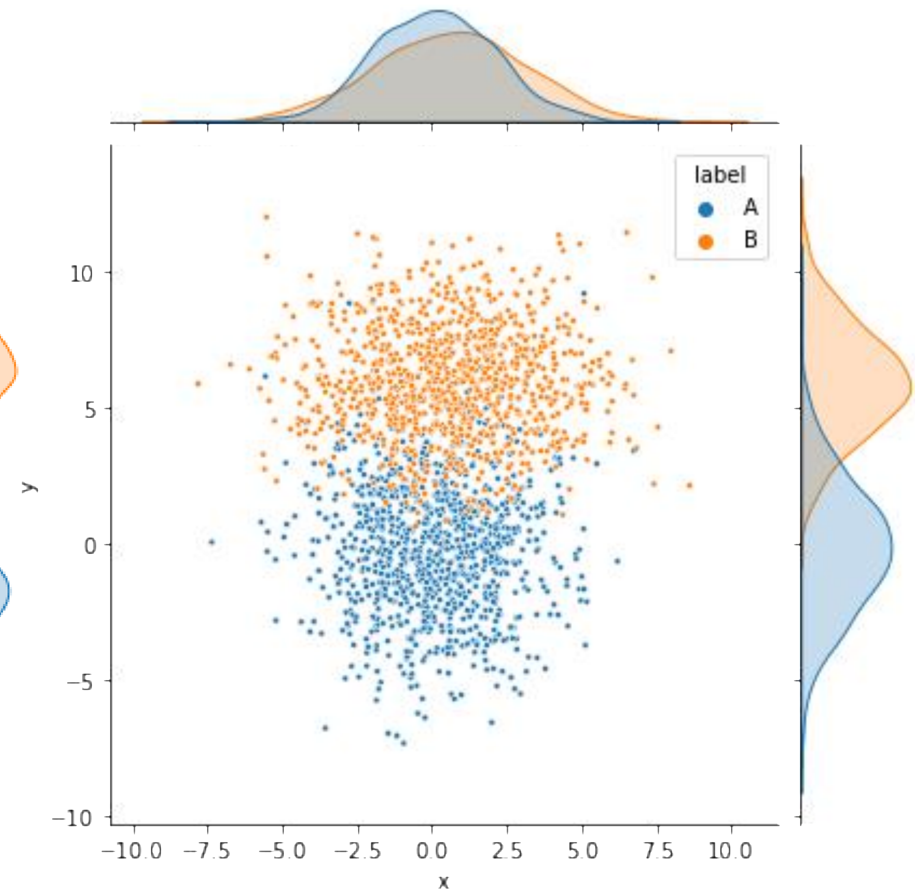
Naïve Bayes

Full Bayes

A variation on the example from S58

- Shifting away the mean of the 'B' class





mean_A = [0, 0], cov_A = [[2, 2.2], [2.2, 2.5]]

Mean_B = [0.5, 6], cov_B = [[2.5, -2.2], [-2.2, 2]]