CS5489 - Machine Learning

Lecture 2a - Bayes Classifier

Prof. Antoni B. Chan

Dept. of Computer Science, City University of Hong Kong

Outline

1. Bayes Classification and Generative Models
2. Parameter Estimation
3. Bayesian Decision Rule

Classification Examples

- Given an email, predict whether it is spam or not spam.
    - **Email 1:**

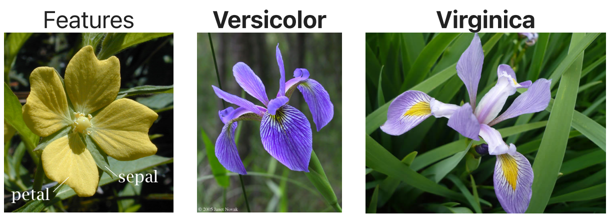
> There was a guy at the gas station who told me that if I knew Mandarin

and Python I could get a job with the FBI.

- **Email 2:**

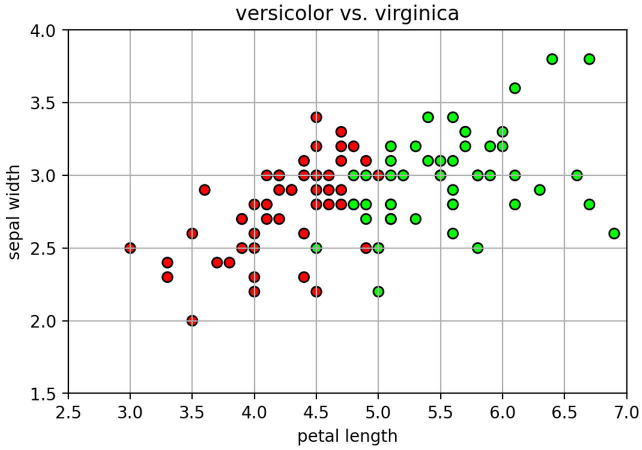
> A home based business opportunity is knocking at your door. Don't be rude and let this chance go by. You can earn a great income and find your financial life transformed. Learn more Here. To Your Success. Work From Home Finder Experts

- Classification Examples
    - Given the *petal length* and *sepal width*, predict the type of iris flower.



In [3]:
```
irisfig
```

Out[3]:



General Classification Problem

- Observation $\mathbf{x}$ (i.e., features)
  - typically a real vector, $\mathbf{x} \in \mathbb{R}^d$.
  - **Example**: a 2-dim vector containing the petal length and sepal width.
    - $\mathbf{x} = \begin{bmatrix} \text{petal length} \\ \text{sepal width} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$
- Class y
  - takes values from a set of possible class labels $\mathcal{Y}$.
  - **Example:** $\mathcal{Y} = \{\text{"versicolor"}, \text{"virginica"}\}$.
    - or equivalently as numbers, $\mathcal{Y} = \{1, 2\}$.
- **Goal**: given an observed features $\mathbf{x}$, predict its class y.

Probabilistic model

- To build a classifier we need to model the relationship between observations and classes.
- Model *how* the data is generated using probability distributions.
  - called a **generative model**.
  - build our assumptions about the world into the model.
- Generative model
  - 1. The world has objects of various classes.
  - 2. The observer measures features/observations from the objects.
  - 3. Each class of objects has a particular probability distribution of features.

- Need to define probability models for:
  1. the classes
  2. the features for each class

Class model

- Set of possible classes are $\mathcal{Y}$
  - For example, $\mathcal{Y} = \{\text{"versicolor"}, \text{"virginica"}\}$.
    - or more generally, $\mathcal{Y} = \{1, 2\}$.
- In the world, the frequency that class y occurs is given by the probability distribution $p(y)$.
  - $p(y)$ is called the **prior distribution**.

- **Example**: Bernoulli class distribution
  - $p(y = 1) = 0.4$

- $p(y=2) = 0.6$
- "In the world of iris flowers, there are 40% that are Class 1 (versicolor) and 60% that are Class 2 (virginica)"
- distribution: $p(y) = \pi^{1(y=1)}(1-\pi)^{1(y=2)}$
  - π is the parameter (e.g., 0.4)
  - *Indicator function*: $1(q) = \begin{cases} 1, & q \text{ is true} \\ 0, & \text{otherwise} \end{cases}$

Learn from our data

- How to get the parameter $p(y=1) = \pi$ for our model?
  - Assume we have collected some data, $\mathcal{D} = \{y_1, \cdots, y_N\}$.
- **Maximum Likelihood Estimation (MLE)**
  - find the parameter that maximizes the likelihood (log-likelihood) of observing the data.
  - $\pi^* = \mathrm{argmax}_\pi \sum_{i=1}^N \log p(y_i)$
    - sum over the log-likelihoods of each sample (assumes samples are independent)

- if $y=1$, then the log-likelihood is $\log(\pi)$, and if $y=2$ the log-likelihood is $\log(1-\pi)$.
- Sum over each sample:

$$\ell(\pi) = \sum_i 1(y_i=1)\log\pi + 1(y_i=2)\log(1-\pi)$$

  - Then, $\ell(\pi) = N_1 \log\pi + N_2\log(1-\pi)$
    - where $N_1 = \sum_i 1(y_i=1)$ = Number of 1's observed.
    - and $N_2 = \sum_i 1(y_i=2)$ = Number of 2's observed.

- Now solve for π by maximizing $\ell(\pi)$.

  - Take derivative and set to 0 to find the maximum.

  $$\begin{aligned}\frac{d \ell(\pi)}{d\pi} = \frac{N_1}{\pi} - \frac{N_2}{1-\pi} = 0 \ N_1(1-\pi) - N_2 \pi = 0 \ N_1 - (N_1+N_2) \pi = 0 \

\Rightarrow \pi = \frac{N_1}{N_1+N_2}\end{aligned}$$

- $p(y=1) = \frac{\text{number of examples of Class 1}}{\text{total number of examples}}$
- $p(y=2) = \frac{\text{number of examples of Class 2}}{\text{total number of examples}}$

In [4]:
```
N1 = count_nonzero(y==1)  # number of Class 1 examples
N2 = count_nonzero(y==2)  # number of Class 2 examples
N  = len(y)               # total
py = [double(N1)/N, double(N2)/N] # note: avoids integer division!
print(py)
```

```
[0.5, 0.5]
```

Observation model

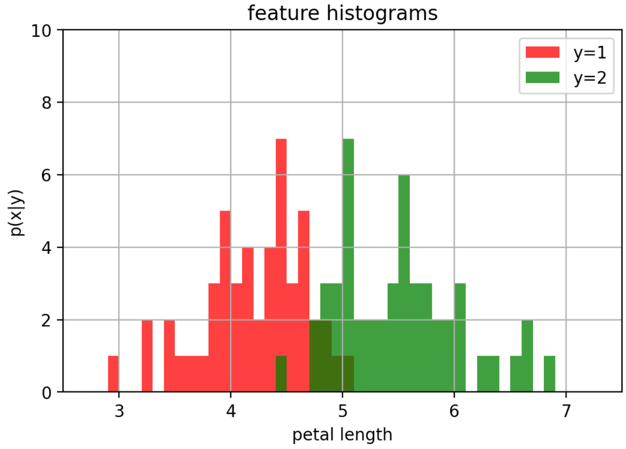
- We measure/observe a feature x
  - the value of the feature x *depends* on the class.
- The observation is drawn according to the distribution $p(x|y)$.
  - $p(x|y)$ is called the **class conditional distribution**
    - "probability of observing a particular feature value x *given* the object is class y"
  - Each class has its own class conditional:
    - $p(x|y=1)$ = distribution of features when its class 1
    - $p(x|y=2)$ = distribution of features when its class 2

Learn from the data

- Histograms for feature "petal length" for each class

In [6]:
```
ccdhist
```

Out[6]:



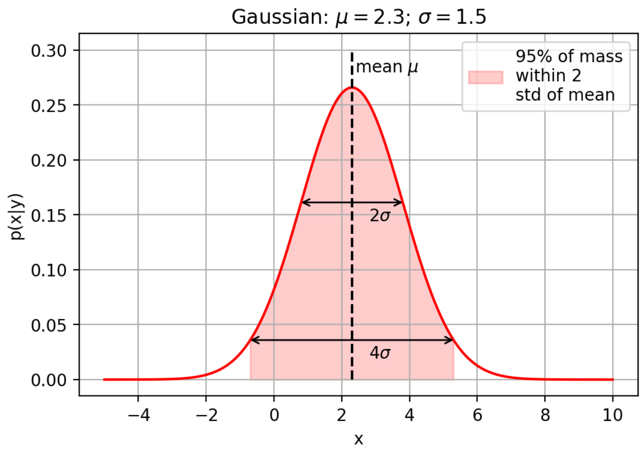
- **Problem:** looks a little bit noisy.
- **Solution:** assume a probability model for the class conditional $p(x|y)$

Gaussian distribution (normal distribution)

- Each class is modeled as a separate Gaussian distribution of the feature value
  - $p(x|y=c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{1}{2\sigma_c^2}(x-\mu_c)^2}$
  - Each class has its own mean and variance parameters (μ_c, σ_c^2).

In [8]:
```
gfig
```

Out[8]:



MLE for Gaussian

- Set the parameters (μ, σ^2) to maximize the log-likelihood of the samples for that class.
  - Let $\{x_i\}_{i=1}^N$ be the observed features for class 1:

$$(\hat{\mu}, \hat{\sigma}^2) = \underset{\mu,\sigma^2}{\operatorname{argmax}} \sum_{i=1}^N \log p(x_i|y_i = 1)$$

  - Then, the objective is $\ell(\mu) = \sum_{i=1}^N -\frac{1}{2\sigma^2}(x_i - \mu)^2 - \frac{1}{2}\log 2\pi\sigma^2$

- take derivative and set to 0

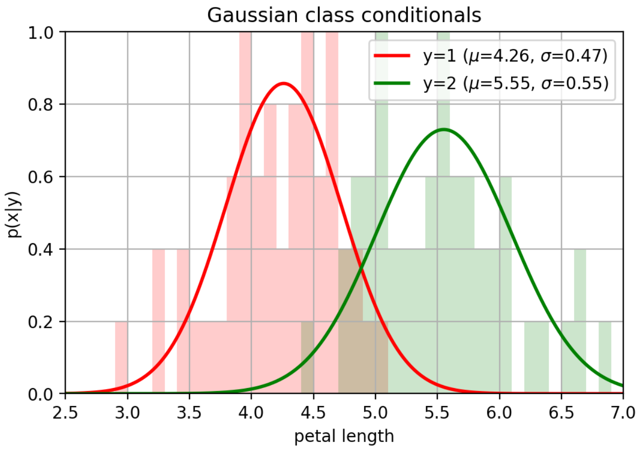
$$\sum_{i=1}^{N} \frac{1}{\sigma^2}(x_i - \mu) = 0$$
$$\sum_{i=1}^{N} x_i - N\mu = 0$$
$$\Rightarrow \mu = \frac{1}{N}\sum_i x_i$$

- Solution:
  - sample mean: $\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i$
  - sample variance:

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{\mu})^2$$

In [11]:
```
gcd
```

Out[11]:



Bayesian Decision Rule

- The Bayesian decision rule (BDR) makes the optimal decisions on problems involving probability (uncertainty).
  - minimizes the *probability of making a prediction error*.
- **Bayes Classifier**
  - Given observation x, pick the class c with the *largest posterior probability*, $p(y = c|x)$.
    - Probability of the class given observed x.
  - **Example:**
    - if $p(y = 1|x) > p(y = 2|x)$, then choose Class 1
    - if $p(y = 1|x) < p(y = 2|x)$, then choose Class 2

- Problem: we don't have $p(y|x)$!
  - we only have $p(y)$ and $p(x|y)$.

Bayes' Rule

- The posterior probability can be calculated using Bayes' rule:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

- The denominator is the probability of feature x, regardless of its class.
  - $p(x) = \sum_{y \in \mathcal{Y}} p(x|y)p(y)$
- The denominator makes $p(y|x)$ sum to 1.

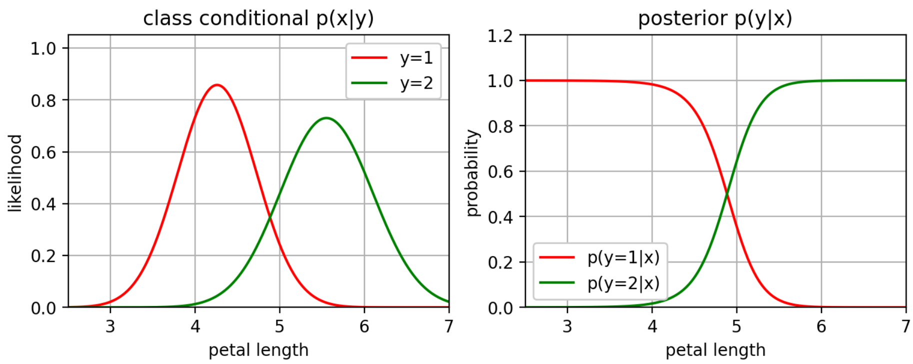
- Bayes' rule:

$$p(y = 1|x) = \frac{p(x|y = 1)p(y = 1)}{p(x|y = 1)p(y = 1) + p(x|y = 2)p(y = 2)}$$

- **Example**:

In [13]:
```
iris1dpost
```

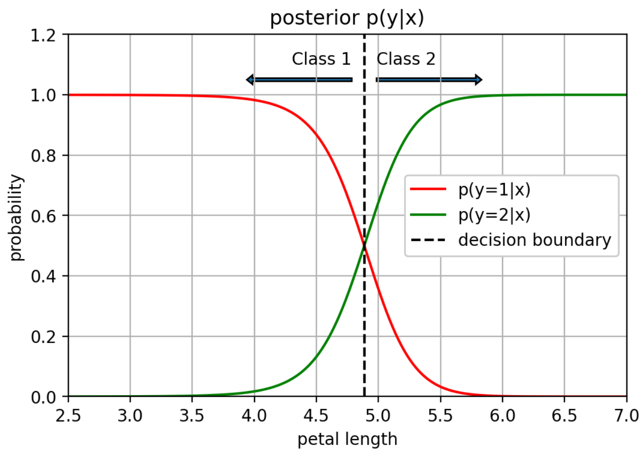
Out[13]:



- The *decision boundary* is where the two posterior probabilites are equal
  - $p(y = 1|x) = p(y = 2|x)$

In [15]:
```
iris1dpost2
```

Out[15]:



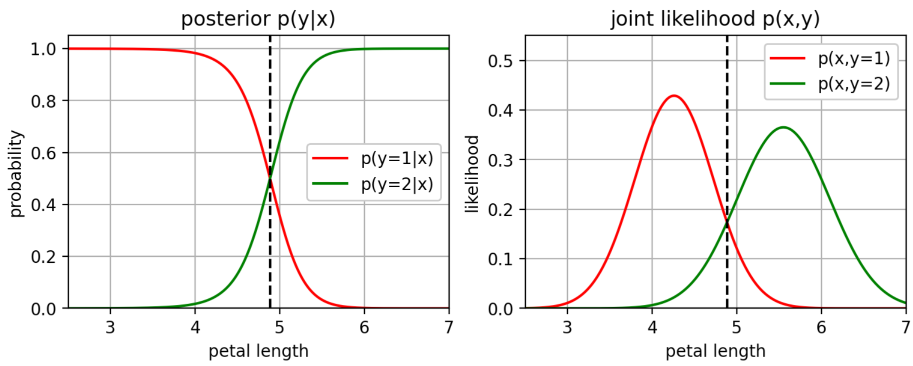
Bayes rule revisited

- Bayes' rule: $p(y|x) = \frac{p(x|y)p(y)}{p(x)}$
- Note that the denominator is the same for each class y.
  - hence, we can compare just the numerators $p(x|y)p(y)$.
  - This also called the *joint likelihood* of the observation and class
    - $p(x, y) = p(x|y)p(y)$

- **Example:**
  - BDR using joint likelihoods:
    - if $p(x|y = 1)p(y = 1) > p(x|y = 2)p(y = 2)$, then choose Class 1
    - otherwise, choose Class 2

In [17]:
```
iris1djoint
```

Out[17]:



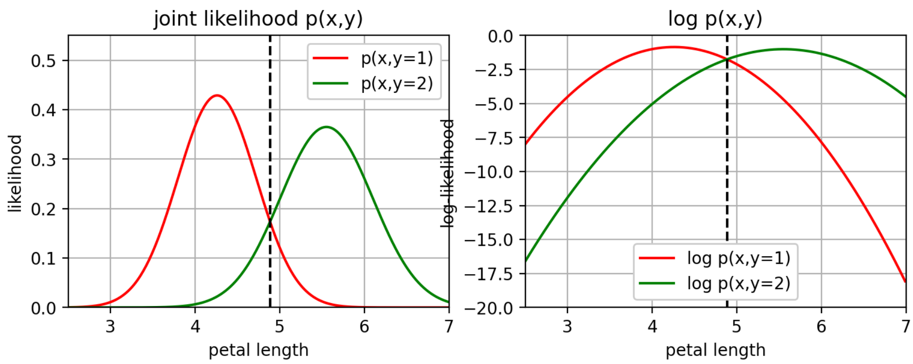
- Can also apply a monotonic increasing function (like $\log$) and do the comparison.
  - Using log likelihoods:

$$\log p(x|y=1) + \log p(y=1) > \log p(x|y=2) + \log p(y=2)$$

  - This is more numerically stable when the likelihoods are small.

In [19]:
```
iris1dLL
```

Out[19]:



Bayes Classifier Summary

- **Training:**
  1. Collect training data from each class.
  2. For each class c, estimate the class conditional densities $p(x|y=c)$:
  3. select a form of the distribution (e.g. Gaussian).
  4. estimate its parameters with MLE.
  5. Estimate the class priors $p(y)$ using MLE.
- **Classification:**
  1. Given a new sample x^*, calculate the likelihood $p(x^*|y=c)$ for each class c.
  2. Pick the class c with largest posterior probability $p(y=c|x^*)$.
  - (equivalently, use $p(x^*|y=c)p(y=c)$ or $\log p(x^*|y=c) + \log p(y=c)$)