Machine Learning

CSE 6363 (Fall 2019)

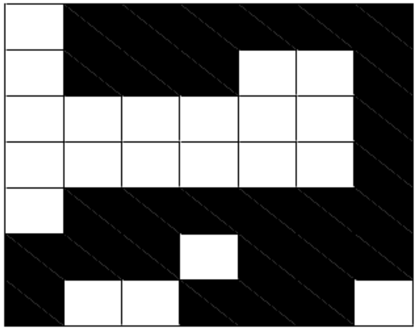
Lecture 6 Decision Theory, Probability Distribution

Dajiang Zhu, Ph.D.

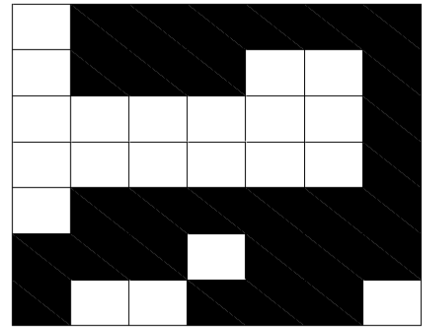
Department of Computer Science and Engineering

Slides of this course (CSE6363) courtesy: Dr. Heng Huang, Dr. Aarti Singh, Dr. **Tibshirani**

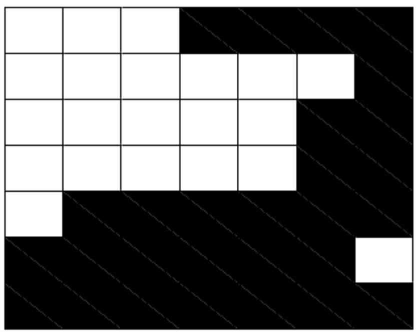
Learning, Biases, Representation



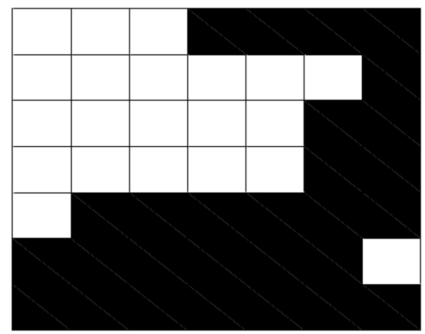
“yes”



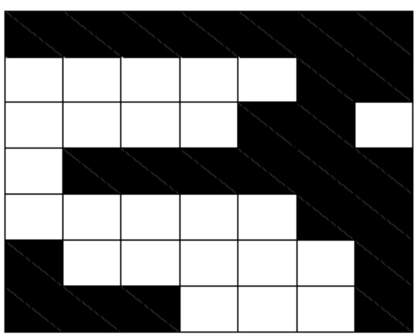
“yes”



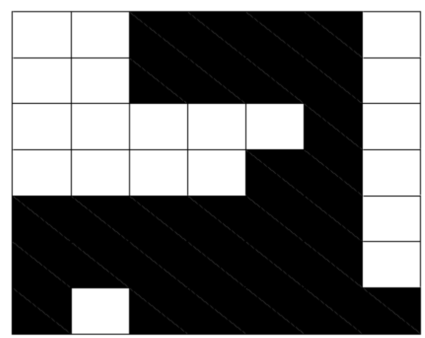
“yes”



“yes”



“no”

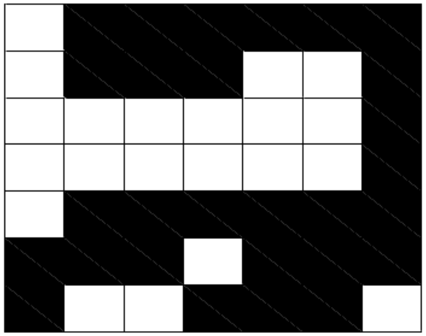


“no”

Ref: Tommi Jaakkola

Representation

- There are many ways of presenting the same information



0111111001110010000000100000001001111110111011111100
111011110001

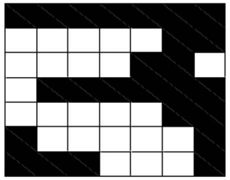
- The choice of representation may determine whether the learning task is very easy or very difficult

Ref: Tommi Jaakkola

Hypothesis Class

- Representation: examples are binary vectors of length $d = 64$

$$\mathbf{x} = [111\ldots0001]^T \;=$$



and labels $y \in \{-1, 1\}$ ("no","yes")

- The mapping from examples to labels is a "linear classifier"

$$\hat{y} = \text{sign}\,(\,\theta \cdot \mathbf{x}\,) = \text{sign}\,(\,\theta_1 x_1 + \ldots + \theta_d x_d\,)$$

where θ is a vector of *parameters* we have to learn from examples.

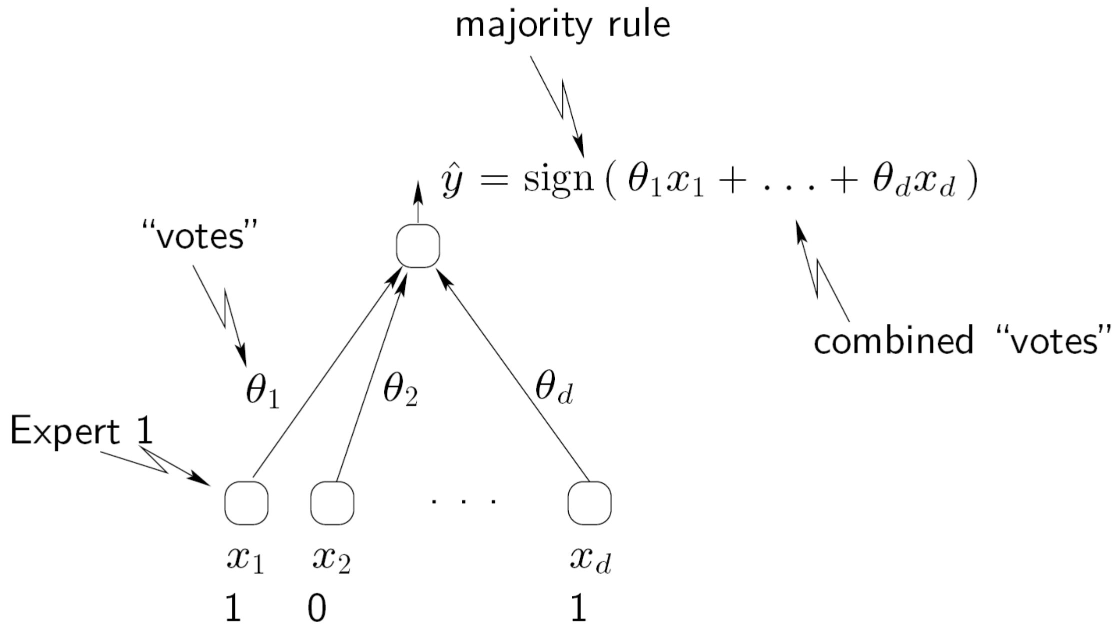
Ref: Tommi Jaakkola

Linear Classifier/Experts

- We can understand the simple linear classifier

$$\hat{y} = \text{sign}\,(\,\theta \cdot \mathbf{x}\,) = \text{sign}\,(\,\theta_1 x_1 + \ldots + \theta_d x_d\,)$$

as a way of combining expert opinion (in this case simple binary features)



Ref: Tommi Jaakkola

Estimation

$\mathbf{x}$	y
0111111001110010000000100000001001111110111011111001110111110001	+1
0001111100000011000001110000011001111110111111001111111100000011	+1
1111111000000110000011000111111000000111100000111110001101111111	−1
...	...

- How do we adjust the parameters θ based on the labeled examples?

$$\hat{y} = \text{sign}\left(\, \theta \cdot \mathbf{x} \,\right)$$

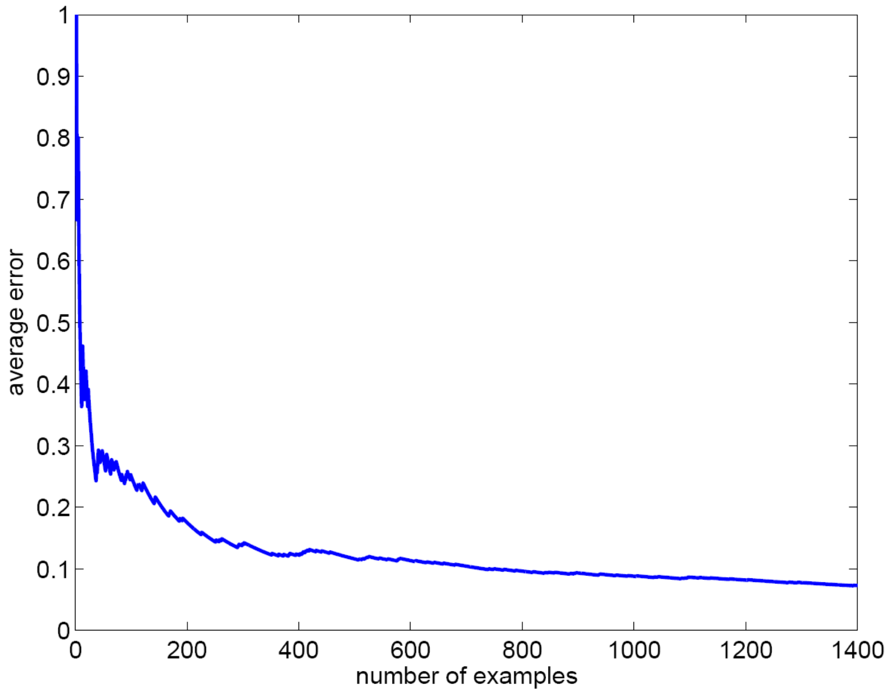
For example, we can simply refine/update the parameters whenever we make a mistake:

$$\theta_i \leftarrow \theta_i + y\, x_i,\;\; i = 1, \ldots, d \quad \text{if prediction was wrong}$$

Ref: Tommi Jaakkola

Evaluation

- Does the simple mistake driven algorithm work?

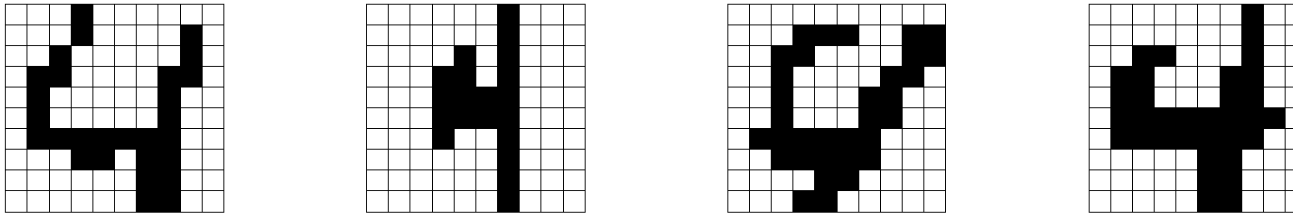


(average classification error as a function of the number of examples and labels seen so far)

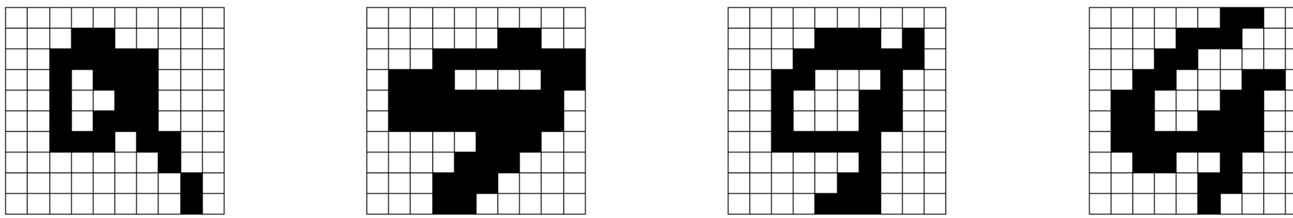
Ref: Tommi Jaakkola

Similar Problem

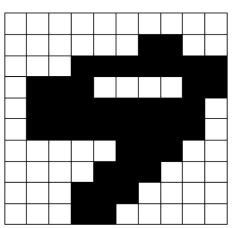
$y = +1$



$y = -1$



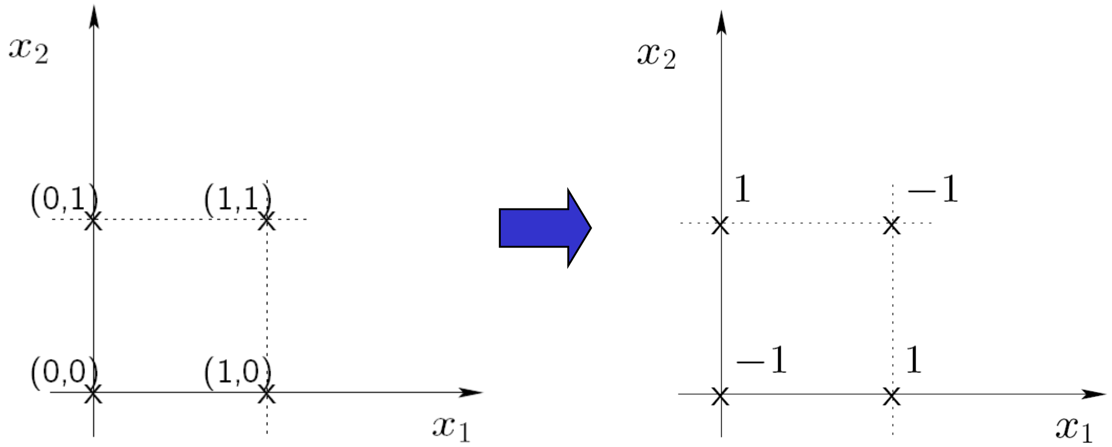
- Representation as a vector:



$$\Rightarrow \quad [0000000000\ 0000001100\ 0001111111\ \ldots\ 0001100000]^T$$

Model Selection

- The simple linear classifier cannot solve all the problems (e.g., XOR)



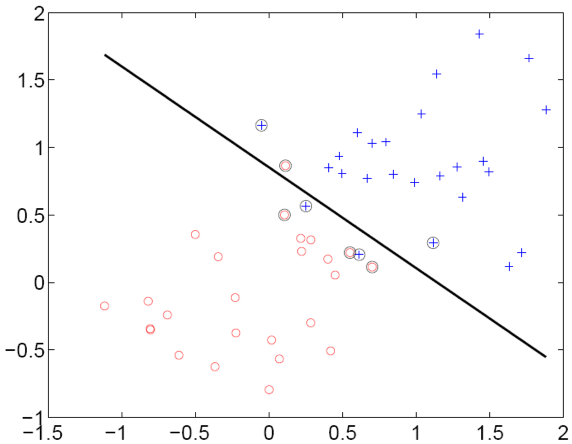
- Can we rethink the approach to do even better?

  We can, for example, add "polynomial experts"

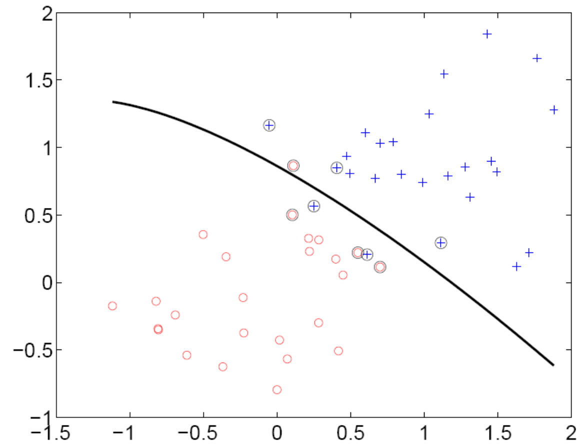
$$\hat{y} = \text{sign}\left(\theta_1 x_1 + \ldots + \theta_d x_d + \theta_{12} x_1 x_2 + \ldots \right)$$

Ref: Tommi Jaakkola

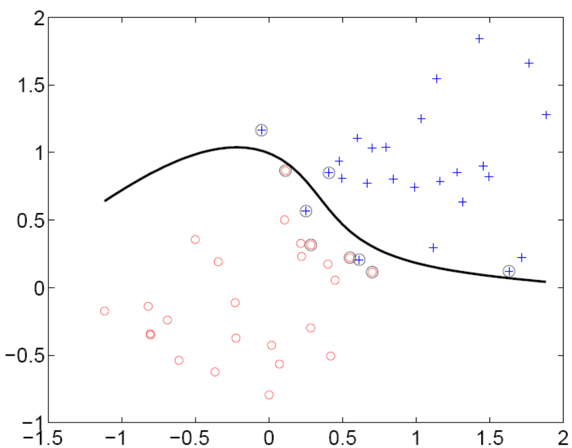
Model Selection (cont.)



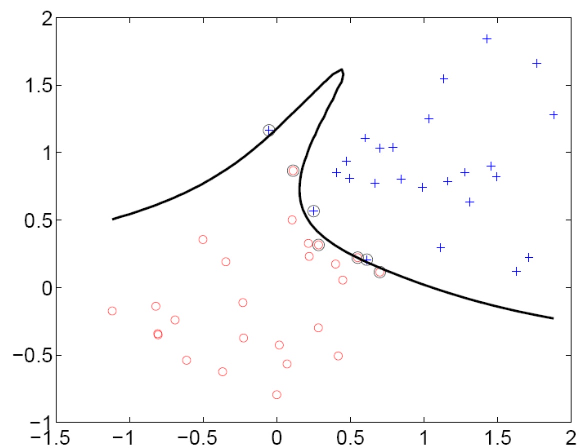
linear



2^{nd} order polynomial



4^{th} order polynomial

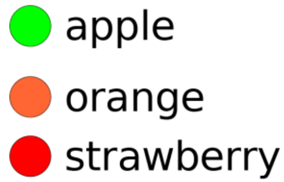


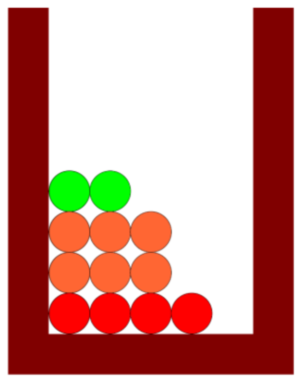
8^{th} order polynomial

Ref: Tommi Jaakkola

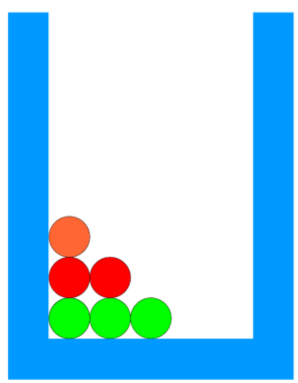
Probability Theory

- Boxes of fruit





	apples	oranges	strawberries
red jar	2	6	4

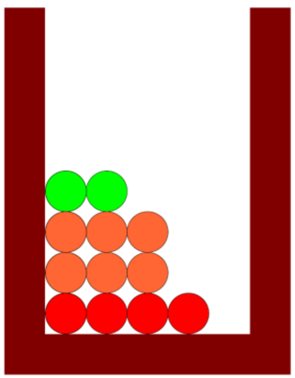


	apples	oranges	strawberries
blue jar	3	1	2

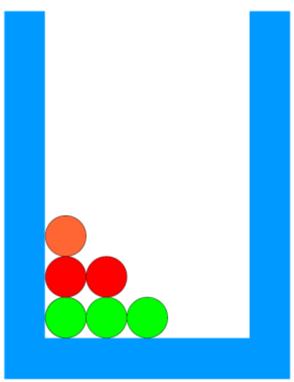
Ref: Chuck Anderson

Probabilities of Fruit from a Given Jar

- apple
- orange
- strawberry



	apples	oranges	strawberries	
red jar	2/12 = 0.167	6/12 = 0.5	4/12 = 0.33	sum = 1.0



	apples	oranges	strawberries	
blue jar	3/6 = 0.5	1/6 = 0.167	2/6 = 0.33	sum = 1.0

Ref: Chuck Anderson

Choose Jar then Draw a Fruit

apple

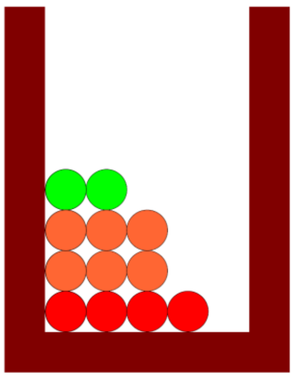
orange

strawberry

Say the probability of choosing a jar is

P(Jar = red) = 0.6
P(Jar = blue) = 0.4

Prob is 0.6



The probability of choosing the red jar **and** drawing an apple out of it is

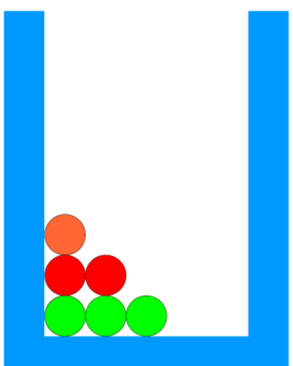
P(Jar=red, Fruit=apple)
= P(Jar=red) P(Fruit=apple|Jar=red)
= 0.6 (0.167) = 0.1

conditional probabiliity

Doing all multiplications results in:

	apples	oranges	strawberries	
red jar (P=0.6)	0.6(0.167) = 0.1	0.6(0.5) = 0.3	0.6(0.33) = 0.2	sum = 0.6

Prob is 0.4



	apples	oranges	strawberries	
blue jar (P=0.4)	0.4(0.5) = 0.2	0.4(0.167) = 0.067	0.4(0.33) = 0.133	sum = 0.4

Joint Probability Table

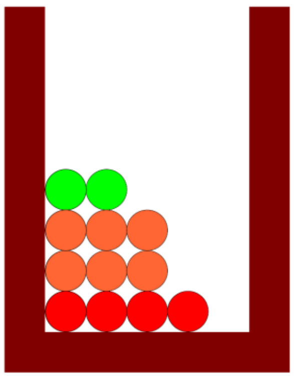
- apple
- orange
- strawberry

Combine in a two-dimensional table to show joint probabilities of two events.

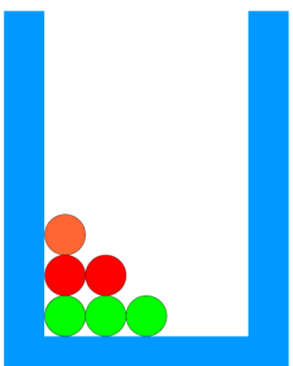
r = red, b = blue

a = apple, o = orange, s = strawberry

Prob is 0.6



Prob is 0.4



		Fruit			
		a	o	s	
Jar	r	0.1	0.3	0.2	$\Sigma = 0.6$
	b	0.2	0.067	0.133	$\Sigma = 0.4$
		$\Sigma = 0.3$	$\Sigma = 0.367$	$\Sigma = 0.333$	$\Sigma = 1.0$

Let J be random variable for Jar, and F be random variable for fruit.

		Fruit			
		a	o	s	
Jar	r	$P(J=r, F=a)$	$P(J=r, F=o)$	$P(J=r, F=s)$	$P(J=r)$
	b	$P(J=b, F=a)$	$P(J=b, F=o)$	$P(J=b, F=s)$	$P(J=b)$
		$P(F=a)$	$P(F=o)$	$P(F=s)$	1.0

Ref: Chuck Anderson

Joint Probabilities and Bayes Rule

Just saw example of the *product rule*:

P(Fruit=orange, Jar = blue)
= P(Fruit=orange | Jar = blue) P(Jar = blue)

Since P(Fruit=orange, Jar = blue) = P(Jar = blue, Fruit = orange),

P(Jar = blue, Fruit=orange)
= P(Jar = blue | Fruit = orange) P(Fruit = orange).

Setting these equal leads to *Bayes Rule*:

P(Jar = blue | Fruit = orange) P(Fruit = orange)
= P(Fruit=orange | Jar = blue) P(Jar = blue)

so

P(Jar = blue | Fruit = orange)
= P(Fruit=orange | Jar = blue) P(Jar = blue) / P(Fruit = orange)

Ref: Chuck Anderson

Joint Probabilities and Bayes Rule

On the right hand side of Bayes Rule, all terms are given to us except P(Fruit = orange):

$$P(\text{Jar} = \text{blue} \mid \text{Fruit} = \text{orange}) = \frac{P(\text{Fruit}=\text{orange} \mid \text{Jar} = \text{blue})\, P(\text{Jar} = \text{blue})}{P(\text{Fruit} = \text{orange})}$$

We can use the *sum rule* to get this.

$$P(\text{Fruit} = \text{orange}) = \sum_j P(\text{Fruit}=\text{orange}, \text{Jar}=j) = 0.367$$

So, Bayes Rule can be rewritten as

$$P(\text{Jar} = \text{blue} \mid \text{Fruit} = \text{orange})$$

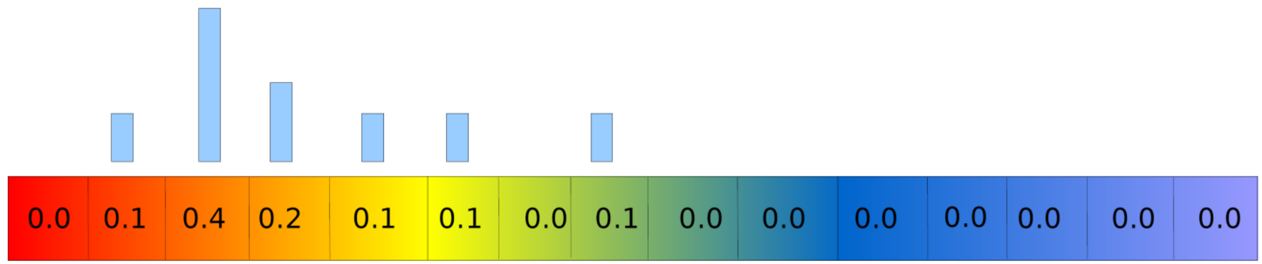
$$= \frac{P(\text{Fruit}=\text{orange} \mid \text{Jar} = \text{blue})\, P(\text{Jar} = \text{blue})}{\sum_j P(\text{Fruit}=\text{orange}, \text{Jar}=j)} = \frac{P(\text{Fruit}=\text{orange} \mid \text{Jar} = \text{blue})\, P(\text{Jar} = \text{blue})}{\sum_j P(\text{Fruit}=\text{orange} \mid \text{Jar}=j)\, P(\text{Jar} = j)}$$

Ref: Chuck Anderson

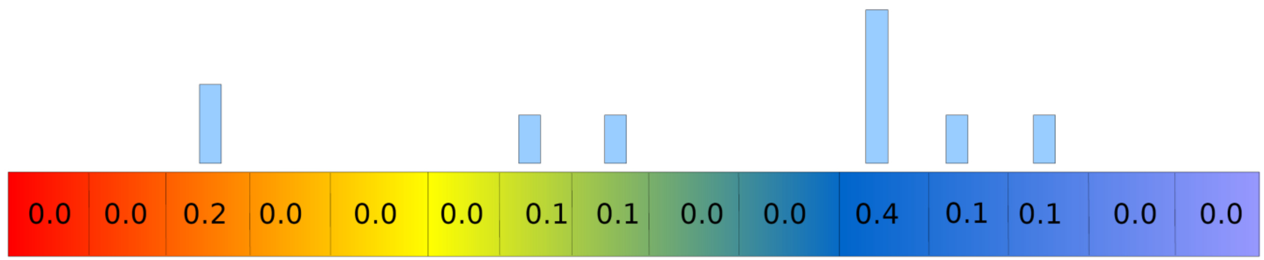
Probability Distributions

- Rather than three colors of fruit, imagine objects of 15 possible colors

Jar 1 contains objects with colors in these proportions



Jar 2 contains objects with colors in these proportions



- Can calculate joint probability table as before.
- But what if we have 100 colors or 1000 colors?
- What if we have an infinite number of colors?

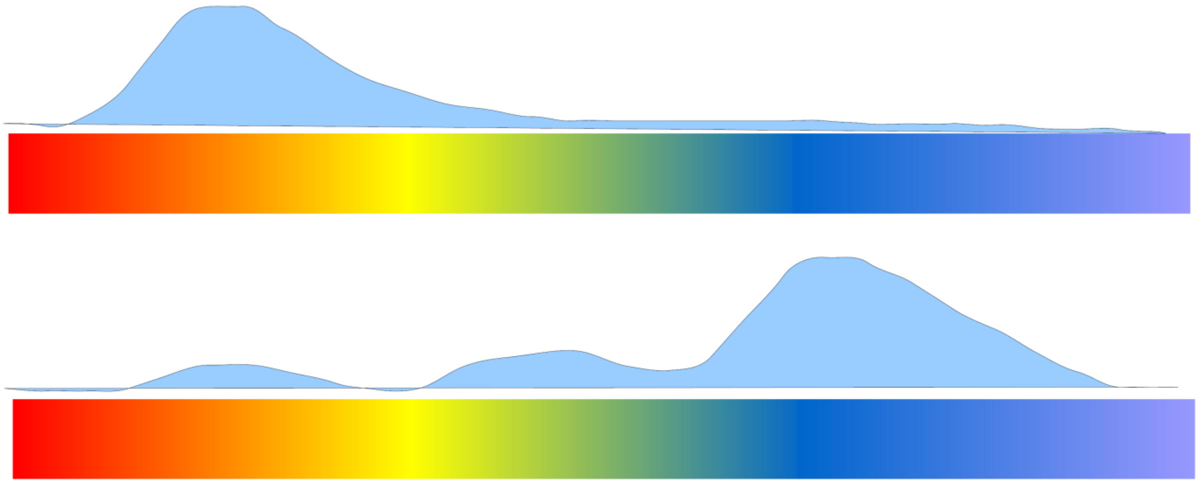
Ref: Chuck Anderson

Distributions Over Continuous Values

- Probability of a color is a function over the continuous spectrum.

Jar 1 contains objects with colors in these proportions

Jar 2 contains objects with colors in these proportions



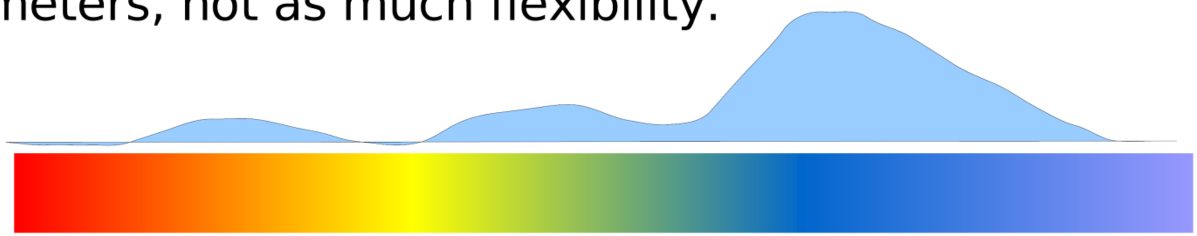
- But what function is this? Would require 1,000s of parameters to specify general function.
- Instead, let's use rather simple functions controlled by a few parameters.
- Common example: Gaussian (Normal) distribution

Ref: Chuck Anderson

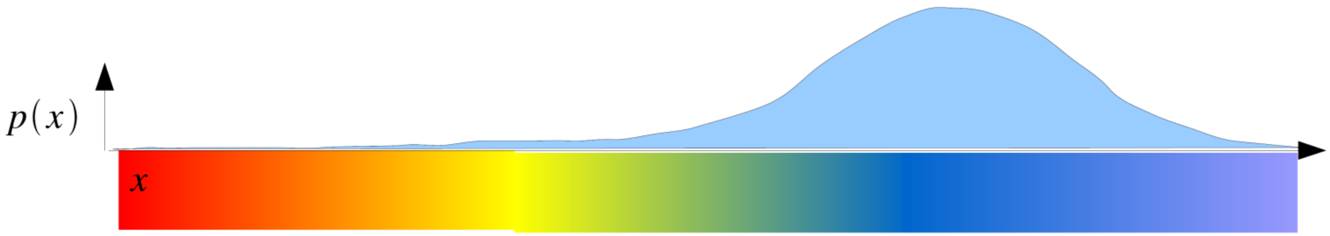
Gaussian Distribution

- With few parameters, not as much flexibility.

Jar 2 distribution



becomes



because

argument parameters

$$p(x;\mu,\sigma)=\frac{1}{(2\pi\sigma^2)^{1/2}}\exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$

- Easy to estimate parameters.

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

Ref: Chuck Anderson

Gaussian Distribution

- Where do these expressions come from?

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

- Maximize the likelihood of the data.
  - Likelihood of data is product of probabilities of each sample x_i

$$p(X|\mu,\sigma) = \Pi_{i=1}^{N}\left[\frac{1}{(2\pi\sigma^2)^{1/2}}\exp\left(-\frac{1}{2\sigma^2}(x_i-\mu)^2\right)\right]$$

  - Maximize this by maximizing its logarithm.

$$\ln p(X|\mu,\sigma) = -\frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi)$$

  - Set its derivative with respect to μ to zero and solve for μ.
  - Set its derivative with respect to σ to zero and solve for σ.

Ref: Chuck Anderson

Bayes Theorem

joint probability

conditional probability

$$p(D)\,p(W\mid D) = p(D,W) = p(W)\,p(D\mid W)$$

Prior probability of weight vector W

Probability of observed data given W

$$\boxed{p(W\mid D)} = \frac{\boxed{p(W)}\ \boxed{p(D\mid W)}}{\boxed{p(D)}}$$

Posterior probability of weight vector W given training data D

$$\boxed{\int_W p(W)\,p(D\mid W)}$$

Why we maximize sums of log probs

- We want to maximize the product of the probabilities of the outputs on the training cases
  - Assume the output errors on different training cases, c, are independent.

$$p(D \mid W) = \prod_c p(d_c \mid W)$$

- Because the log function is monotonic, it does not change where the maxima are. So we can maximize sums of log probabilities

$$\log p(D \mid W) = \sum_c \log p(d_c \mid W)$$

An even cheaper trick

- Suppose we completely ignore the prior over weight vectors
  – This is equivalent to giving all possible weight vectors the same prior probability density.
- Then all we have to do is to maximize:

$$\log p(D \mid W) = \sum_{c} \log p(D_c \mid W)$$

- This is called maximum likelihood learning. It is very widely used for fitting models in statistics.

Decision Theory
Probabilities and Bayes' Theorem

- Classify images as the correct digit.
  - Given *p(Image=i | Digit = d)*, *p(Image = i)*, and *p(Digit = d)*.
  - Calculate

$$p(Digit=d|Image=i)=\frac{p(Image=i|Digit=d)\,p(Digit=d)}{p(Image=i)}$$

- or, more generally

$$p(Class=k|X=x)=\frac{p(X=x|Class=k)\,p(Class=k)}{p(X=x)}$$

- or, more concisely

$$p(C_k|x)=\frac{p(x|C_k)\,p(C_k)}{p(x)}$$

- To classify *x*,

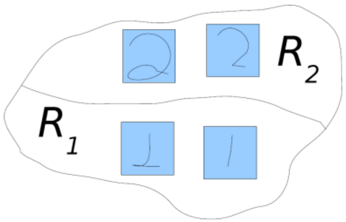
$$\underset{C_k}{\operatorname{argmax}}\, p(C_k|x) \qquad \text{for example, } \underset{C_k}{\operatorname{argmax}}\, p(Digit=d|Image=i)$$

- We get to this by first defining measure of decision accuracy.

Ref: Chuck Anderson

Decision Theory
Decision Regions and Measures of Accuracy

- Decision regions



$$p(\text{mistake}) = p(x \in R_1, Digit=2) + p(x \in R_2, Digit=1)$$
$$= p(x \in R_1, C_2) + p(x \in R_2, C_1)$$

- If x is discrete $= \sum_{x \in R_1} p(x, C_2) + \sum_{x \in R_2} p(x, C_1)$
- If x is continuous $= \int_{R_1} p(x, C_2)\, dx + \int_{R_2} p(x, C_1)\, dx$

- Make assignment of x to R_k to minimize p(mistake), or to maximize p(correct)

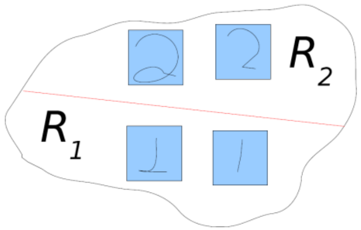
$$p(\text{correct}) = p(x \in R_1, C_1) + p(x \in R_2, C_2)$$
$$= \sum_{k=1}^{K} p(x \in R_k, C_k) = \sum_{k=1}^{K} \sum_{x \in R_k} p(x, C_k)$$
$$= \sum_{k=1}^{K} \int_{R_k} p(x, C_k)$$

Ref: Chuck Anderson

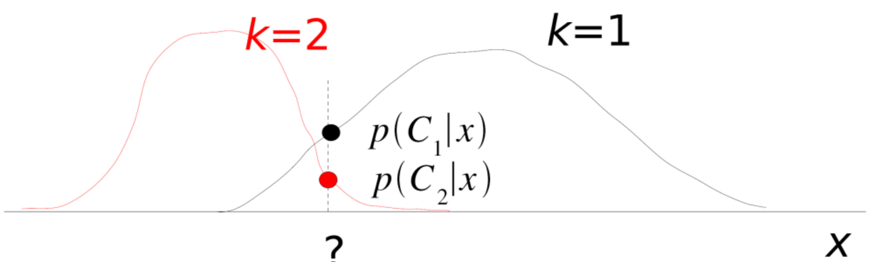
Decision Theory

Decision Regions and Measures of Accuracy

- which, by Bayes' theorem $\sum_{k=1}^{K} \int_{R_k} p(x, C_k) = \sum_{k=1}^{K} \int_{R_k} p(C_k|x)\, p(x)$
- Maximize by constructing R_1, ... R_k as best you can.
- If separating by straight lines



- Each x is assigned to an R_i.
- Since $p(x)$ is same for all k, regions resulting from maximizing p(correct) same as regions resulting from maximizing $\sum_{k=1}^{K} \int_{R_k} p(C_k|x)$
- If x is one-dimensional, and we model $p(C_k \mid x)$ as Gaussian, then

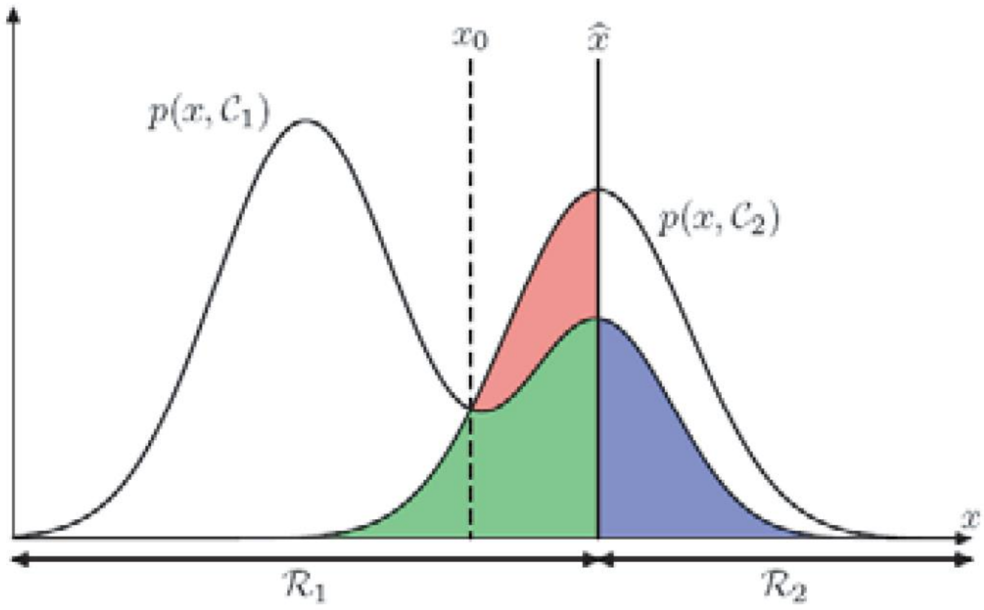


Class 1, because $p(C_1|x) > p(C_2|x)$

Ref: Chuck Anderson

Example

- e.g., the optimal decision threshold is when $\hat{x} = x_0$



Ref: Chuck Anderson

Decision Theory

Decision Regions and Measures of Accuracy

- So far we assumed each missclassification is equally bad, or each correct classification is equally good.
- But, predictions of whether or not a particular space shuttle launch given all known current conditions will result in damage from loose tiles are not equally risky.
  - incorrect prediction of damage (no launch) is better than incorrect prediction of no damage (launch, damage)!
- Define a loss matrix L_{kj}

		Predicted	
		damage	no damage
True	damage	0	10,000
	no damage	10	0

- or, utility matrix

		Predicted	
		damage	no damage
True	damage	100	-10000
	no damage	-10	100

Ref: Chuck Anderson

Decision Theory
Measures of Accuracy

- If we classify x as Class j our loss will be $\sum_{k=1}^{K} L_{kj}\, p(x, C_k)$
- Call this **expected value of loss**, given x classified as Class j.
- Given all x's and a classification scheme that partitions the x space into regions $R_1,\ldots R_k$, the overall expected loss is

$$E[L] = \sum_{k=1}^{K} \sum_{j=1}^{K} \int_{R_j} L_{kj}\, p(x, C_k)\, dx$$

- By Bayes' theorem, to minimize $E[L]$ we would assign x to Class j that minimizes

$$\sum_{k=1}^{K} L_{kj}\, p(C_k | x)$$

- Would classify current shuttle condition as "damage" much more often than "no damage" because of

		Predicted: damage	Predicted: no damage
True	damage	0	10,000
True	no damage	10	0

Ref: Chuck Anderson

Decision Theory

Three ways of making classification decision

- Generative model
  - Learn class-conditional probability (generative model) $p(x|C_k)$
  - Use Bayes' Theorem to convert to $p(C_k|x)$
  - Use decision theory to minimize loss
- Discriminative model
  - Learn posterior class probability (discriminative model) $p(C_k|x)$
  - Use decision theory to minimize loss
- Discriminant function
  - Learn discriminant function *f(x)* that calculates a class directly. Probabilities not involved.
- Advantages and disadvantages of each, in Section 1.5.4.

Ref: Chuck Anderson

Information Theory

- Useful to have measure *h(x)* of how much information is provided by an event, *x*. We would like it to reflect how "surprising" the event is, so it should be related monotonically to probability *p(x)*.
- If two events *x* and *y* are unrelated, total information gained should be sum of each

Ref: Chuck Anderson

Information Content of A Random Variable

- Random variable X
  - Outcome of a random experiment
  - Discrete R.V. takes on values from a finite set of possible outcomes
- How much information is contained in the event X = y?
  - Will the sun rise today?
    - Revealing the outcome of this experiment provides no information
  - Will the Maverick win the NBA championship?
    - Since this is unlikely, revealing yes provides more information than revealing no
- Events that are less likely contain more information than likely events

Ref: Eytan Modiano

Entropy

The **entropy** of a random variable X with a probability mass function $p(x)$ is defined by

$$H(X) = -\sum_{x} p(x) \log_2 p(x).$$

The entropy is measured in bits and is a measure of the average uncertainty in the random variable. It is the number of bits on the average required to describe the random variable.

We write $\log x := \log_2 x$ in the sequel.

Ref: Patric Ostergard

Example: Variable with Uniform Distribution

Consider a random variable with uniform distribution over 32 $(= 2^5)$ outcomes. Obviously, 5-bit strings suffice to identify an outcome. The entropy is

$$H(X) = -\sum_{i=1}^{32} p(i) \log p(i) = -\sum_{i=1}^{32} \frac{1}{32} \log \frac{1}{32} = \log 32 = 5 \text{ bits},$$

which agrees with the number of bits needed to describe X.

Ref: Patric Ostergard

Example: Variable with Nonuniform

Assume that the probabilities of winning for eight horses taking part in a horse race are $\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}\}$. The entropy of this distribution (that is, of the horse race) is then

$$H(X) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{4}\log\frac{1}{4} - \frac{1}{8}\log\frac{1}{8} - \frac{1}{16}\log\frac{1}{16} - 4\frac{1}{64}\log\frac{1}{64} = 2 \text{ bits.}$$

To send a message indicating the winner of the race, one can send the index of the winning horse $(000, \ldots, 111)$; this requires 3 bits for any of the horses. But there is another (better) description.

Ref: Patric Ostergard

Example: Variable with Nonuniform

As the win probabilities are not uniform, it makes sense to use shorter descriptions for the more probably horses, and longer descriptions for the less probable ones. For example, the following strings can be used to represent the eight horses:

$$0, 10, 110, 1110, 111100, 111101, 111110, 111111.$$

The average description length is then 2 bits (=entropy).

- ▷ The entropy gives a lower bound for the average description length.

Ref: Patric Ostergard