Machine Learning

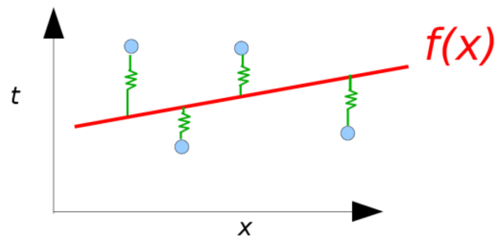
CSE 6363 (Fall 2016)

Lecture 3 Probability Distribution

Heng Huang, Ph.D.

Department of Computer Science and Engineering

Fitting Data with Linear Model (Regression)



- Force exerted by spring is proportional to square of length.
- Rod will settle at position that minimizes

$$\sum_i^N (t_i - f(x_i))^2 \qquad \text{where } N=4$$

- If *f* is affine (linear + constant) function of *x*,

$$f(x) = w_o + w_1 x \quad \text{with parameters } \boldsymbol{w} = (w_{0,} w_1)$$

- Which parameter values give best fit?

$$\boldsymbol{w}_{\textbf{best}} = \underset{\boldsymbol{w}}{\text{argmin}} \sum_i^N (t_i - f(x_i; \boldsymbol{w}))^2$$

- Set derivative with respect to *w* to zero and solve for *w*.

Ref: Chuck Anderson

Fitting Data with Linear Model

$$\boldsymbol{w}_{\textbf{best}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \sum_i^N (t_i - f(x_i;\boldsymbol{w}))^2$$

- Set derivative with respect to *w* to zero and solve for *w*.

$$\frac{d\sum_i^N (t_i - f(x_i;\boldsymbol{w}))^2}{dw_0} = 2\sum_i^N (t_i - f(x_i;\boldsymbol{w}))\frac{(-d\,f(x_i;\boldsymbol{w}))}{dw_0} = 0$$

$$-2\sum_i^N (t_i - f(x_i;\boldsymbol{w})) = 0$$

$$\sum_i^N t_i - N w_0 - w_1 \sum_{i=1}^N x_i = 0$$

$$\frac{d\sum_i^N (t_i - f(x_i;\boldsymbol{w}))^2}{dw_1} = 2\sum_i^N (t_i - f(x_i;\boldsymbol{w}))\frac{(-d\,f(x_i;\boldsymbol{w}))}{dw_1} = 0$$

$$-2\sum_i^N (t_i - f(x_i;\boldsymbol{w}))\,x_i = 0$$

$$\sum_i^N t_i - w_0 \sum_{i=1}^N x_i - w_1 \sum_{i=1}^N x_i^2 = 0$$

- Two simultaneous linear equations to solve for w_0 and w_1.

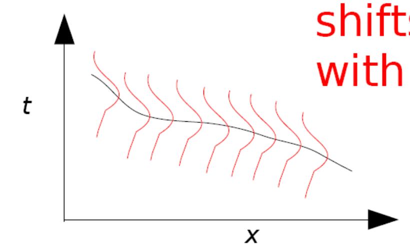
Ref: Chuck Anderson

Fitting Curves with Gaussian Conditional Distribution

- Replace mean μ by some parameterized function of x.

mean of Gaussian shifts with *x*

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



- Now, to find *w* that maximizes this likelihood, set derivative of log likelihood with respect to w equal to zero and solve for *w*.
- Recall that before we

maximized $-\frac{1}{2\sigma^2}\sum_i^N (t_i-\mu)^2$ to get $\mu=\frac{1}{N}\sum_{i=1}^N t_i$

- Now we will

maximize $-\frac{1}{2\sigma^2}\sum_i^N (t_i-y(x_i,\boldsymbol{w}))^2$ or minimize $\frac{1}{2\sigma^2}\sum_i^N (t_i-y(x_i,\boldsymbol{w}))^2$

which is the usual (non-probabilistic) approach of fitting a function to minimize the squared error.

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

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

Posterior probability of weight vector W given training data D

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

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

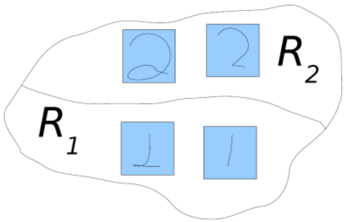
- We get to this by first defining measure of decision accuracy.

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

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

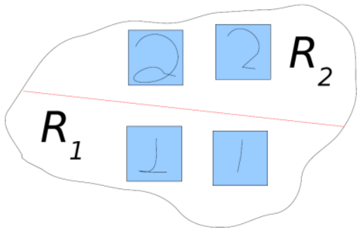
$$p(\text{correct}) = p(x \in R_1, C_1) + p(x \in R_2, C_2)$$
$$= \sum_{k=1}^{K} p(x \in R_k, C_k) = \sum_{k=1}^{K} \sum_{x \in R_k} p(x, C_k)$$
$$= \sum_{k=1}^{K} \int_{R_k} p(x, C_k)$$

Ref: Chuck Anderson

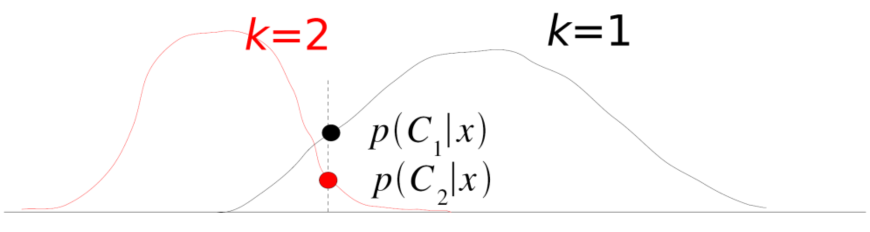
Decision Theory

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

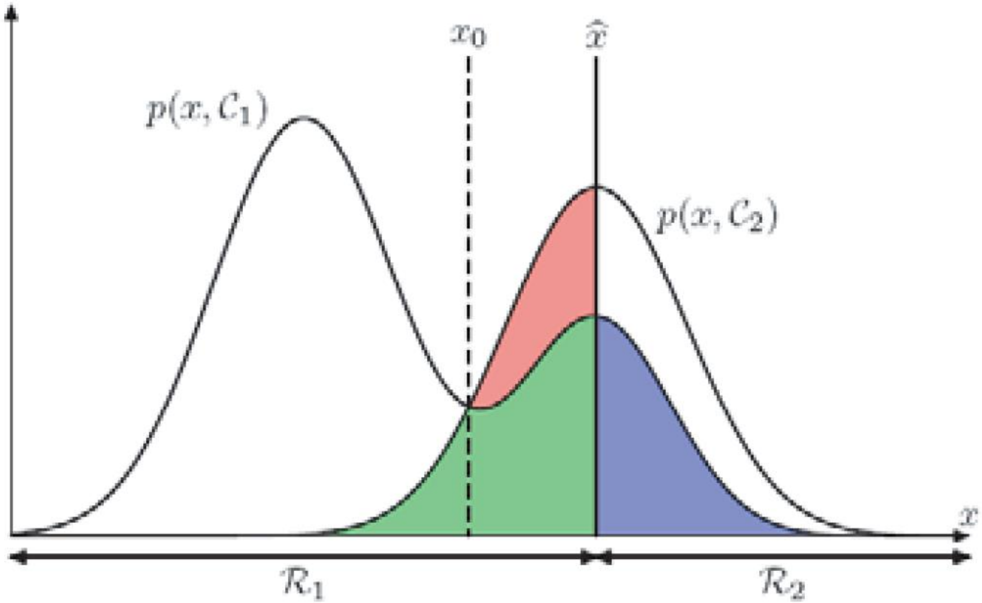


Class 1, because $p(C_1|x) > p(C_2|x)$

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

- or, utility matrix

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

Ref: Chuck Anderson

Decision Theory

Measures of Accuracy

- Given an *x*, we pick column but do not know true class. We will know probability of true class.
- 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	no damage
True	damage	0	10,000
	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
    PMF: $P(X = y) = Px(y)$

- 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

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

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

Joint and Conditional Entropy

Entropy

$$H(X) = -E[logP(X)]$$

$$H(X,Y) = -E_{X,Y}[logP(X,Y)]$$

this is really just entropy with a shift of perspective, in which the pair X, Y is the new random variable. Because $P(X,Y) = P(X)P(Y|X)$

$$\begin{aligned} H(X,Y) &= -E_{X,Y}[logP(X)P(Y|X))] \\ &= -E_{X,Y}[log(P(X) + log(P(Y|X)] \\ &= H(X) - E_{X,Y}[log(P(Y|X)] \end{aligned}$$

so we dignify the last term with the name *conditional entropy* and the notation $H(Y|X)$.

Ref: Chris Brew

Mutual Information

We just derived a chain rule for entropy:

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$$

from which it follows that

$$H(X) - H(X|Y) = H(Y) - H(Y|X)$$

and we call this quantity $I(X;Y)$. If Y is informative about X then $H(X|Y) < H(X)$. In the limit Y determines X and $H(X|Y) = 0$.

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(X) - (H(X,Y) - H(Y)) \\ &= H(X) + H(Y) - H(X,Y) \\ &= -E_x[logP(x)] - E_y[log(P(y))] + E_{x,y}[logP(x,y)] \\ &= E\left[log\frac{P(x,y)}{P(x)P(y)}\right] \end{aligned}$$

Ref: Chris Brew