MACHINE LEARNING AND PATTERN RECOGNITION

Spring 2004, Lecture 2:

Review of Probability and Statistics

Learning and Optimization

Yann LeCun
The Courant Institute,
New York University
http://yann.lecun.com

Review of Probability and Statistics: Definitions

- Random Variable X: a variable that represents a particular measurement/state of the world.
- The probability that X has value x (the result of a drawing, a sampling, or the result of a measurement) is denoted $P(x)$, or sometimes $P(X = x)$.
- The space of outcomes x, can be discrete, or continuous, possibly multidimensional.
- A discrete distribution associates a number $0 \leq P(x) \leq 1$ to each possible outcome x, such that $\sum_x P(x) = 1$.
- A probability Density Function (PDF) associates a positive number $P(x)$ to each point in the space of outcomes (can be larger than 1) such that $\int P(x)dx = 1$.
- The probability that X belongs to a set S is equal to $Prob(X \in S) = \int_{x \in S} P(x)dx$.

Expectations

- Expected value of a function f of a random variable X (a.k.a. the "average value"):

$$\mathcal{E}(f) = \sum_x f(x)P(x)$$

- in the continuous case:

$$\mathcal{E}(f) = \int f(x)P(x)dx$$

- Example 1, the mean of X: $\mathcal{E}(X) = \sum_x xP(x)$
- Example 2, the variance of X: $Var(X) = \mathcal{E}[(X - \mathcal{E}(X))^2] = \sum_x (x - \mathcal{E}(X))^2 P(x)$
- Example 3, the covariance of a multidimensional random variable (dimension N): $Cov(X) = \mathcal{E}(X.X') = \sum_x x.x' P(x)$ $x.x'$ is the outer product of x by itself: $[x.x']_{ij} = x_i x_j$, a symmetric $N \times N$ matrix.

Joint Probability

- Two random variables X and Y (e.g. $X =$ percentage of alcohol in the blood of a person today (continuous), $Y = 1$ if the person is in a car crash, 0 otherwise).
- The joint probability is the function that maps an (x, y) pair to the probability that $X = x$ and $Y = y$ for a person.
- Dependency: Y is more likely to be 1 if X is large, and X is more likely to be large if Y is 1.
- Marginal probabilities:

$$P(x) = \sum_y P(x, y)$$

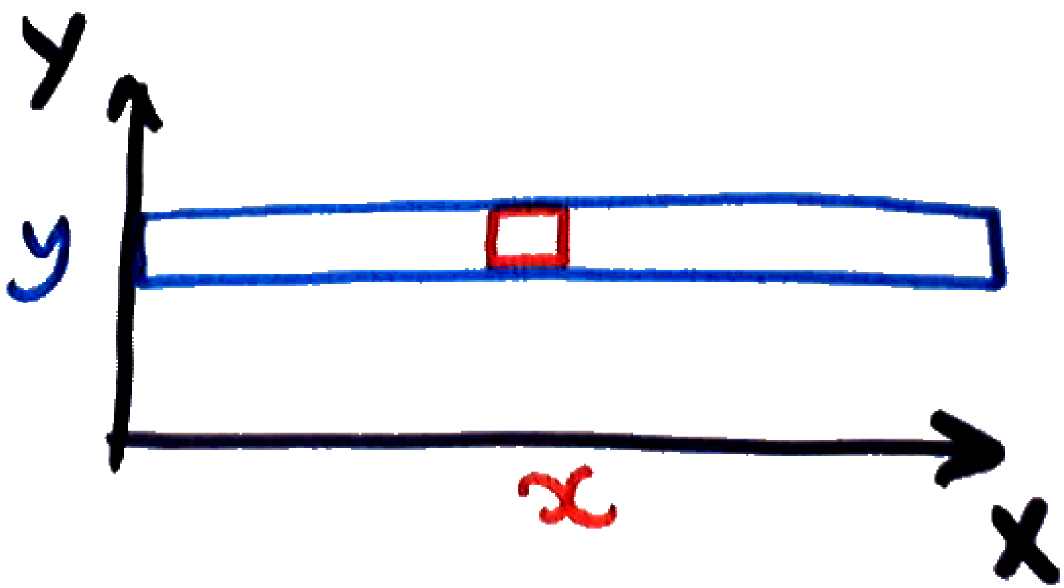
$$P(y) = \int P(x, y) dx$$

Conditional Probability

- Probability that someone was in a car crash knowing that the person was drunk $=$ of all the persons who were drunk, what proportion had a car crash:

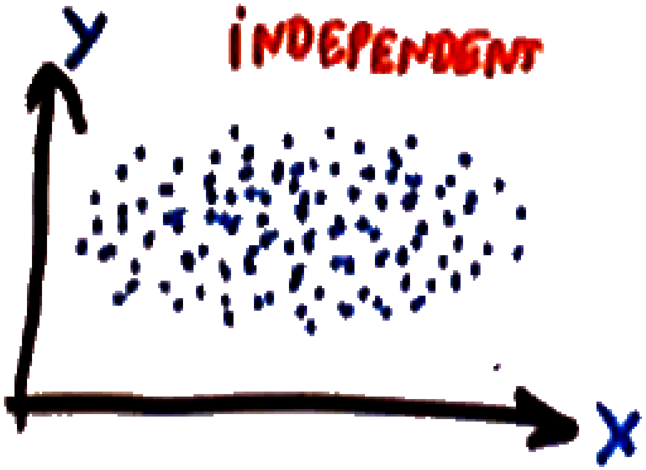
$$P(y|x) = p(x,y)/p(x)$$

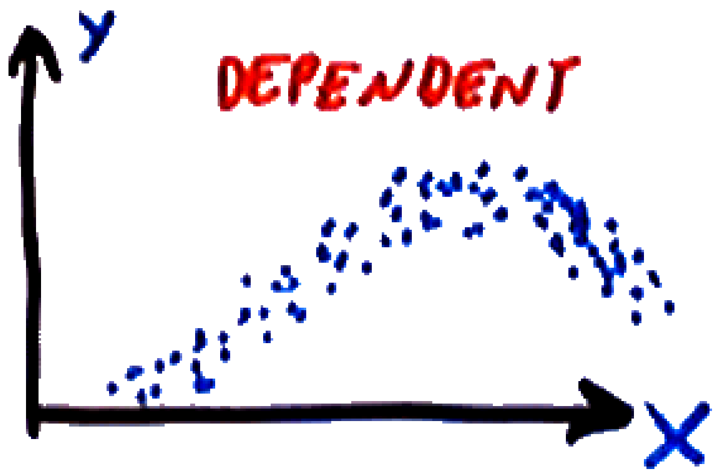
- $P(y|x)$ is read "Probability of y given x.
- Normalization: $\sum_y P(y|x) = 1$



Conditional Independence

Independence: X and Y are *independent* iff $P(x, y) = P(x)P(y)$, in other words $P(x|y) = P(x)$ and $P(y|x) = P(y)$.





Special Distributions: Exponential Family

- A very general family of parameterized distributions.
- $P(x|\omega) = h(x)\exp(\omega' T(x) - A(\omega)) = \frac{1}{Z(\omega)} h(x)\exp(\omega' T(x))$
- ω the “natural” parameter
- $Z(\omega) = \exp(A(\omega))$ is the *partition function*
- $T(x)$ a *sufficient statistic*: all you need to know about x to compute its distribution with a linear combination.

Special Distributions: Gaussian

- For a continuous random variable: $P(x|m,v) = \frac{1}{\sqrt{2\pi v}} \exp(-\frac{1}{2v}(x-m)^2)$
- m is the mean, v is the variance.
- exponential family with
  - $w = [m/v, -1/2v]$
  - $T(x) = [x, x^2]$
  - $Z(w) = \sqrt{v}\exp(m/2v)$
  - $h(x) = 1/\sqrt{2\pi}$

Special Distributions: Multivariate Gaussian

- For a continuous random variable (X, and M are N-dimensional vectors, V is an $N \times N$ matrix):
  $P(X|M,V) = |2\pi V|^{-1/2} \exp(-1/2(X-M)'V^{-1}(X-M))$
- $|2\pi V|$ is the determinant of $2\pi V$.
- exponential family with
  - $w = [V^{-1}M, -1/2V^{-1}]$
  - $T(x) = [X, XX']$
  - $Z(w) = |V|/2 \exp(1/2M'V^{-1}M)$
  - $h(x) = (2\pi)^{-N/2}$
- Important facts: marginals of Gaussians are Gaussians, products of Gaussians are Gaussians, conditionals of Gaussians are Gaussians.

Bayes' Rules

- From the definition of conditional probabilities $P(x, y) = P(x|y)P(y)$.
- Therefore $P(x, y) = P(x|y)P(y) = P(y|x)P(x)$.
- Hence

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

- Or equivalently:

$$P(x|y) = \frac{P(y|x)P(x)}{\sum_{x'} P(y|x')P(x')}$$

- This is a convenient way of reversing conditional probabilities.

More General Forms of Bayes' Rules

- Chain rule (any ordering works):

$$P(x, y, z) = P(x|y, z)P(y|z)P(z) = P(z|y, x)P(y|x)P(x) = ...$$

- In general: $P(x_1 ... x_n) = \prod_i P(x_i|x_1 ... x_{i-1})$ for any ordering $1..n$.
- Conditional Bayes inversion:

$$P(x|y, z) = \frac{P(y|x, z)P(x, z)}{P(y, z)}$$

- Chain rule and maginalization in one fell swoop (feels like a matrix-vector or matrix-matrix product):

$$P(y) = \int_x P(y|x)P(x)$$

$$P(y|z) = \int_x P(y|x)P(x|z)$$

Probabilistic Models: Bayes Decision Theory

A common (but according to some, flawed) way of building a classifier is to estimate the density function for each class $P(X|C1)$ and $P(X|C2)$. When a new input comes in, compute the **posterior probability** of the class conditioned on the input using Bayes rule:

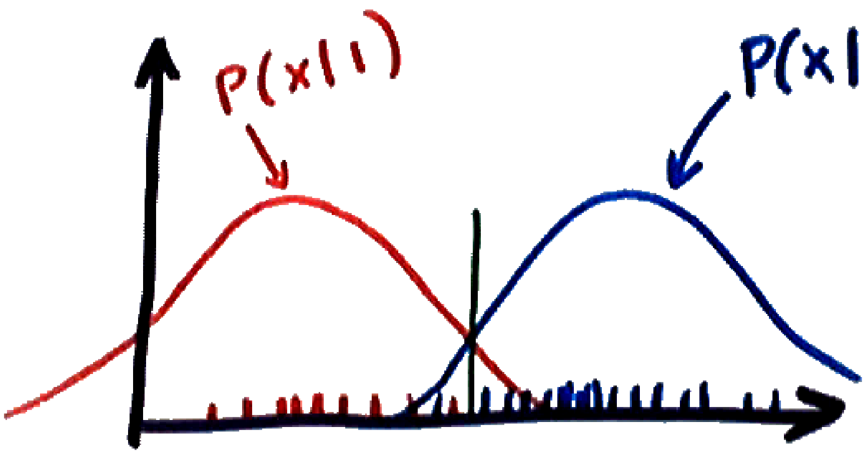
$$P(C1|X) = \frac{P(X|C1)P(C1)}{P(X)}$$

This can be rewritten as:

$$P(C1|X) = \frac{P(X|C1)P(C1)}{\sum_C P(X|C)P(C)}$$

The same can be done for class $C2$. Then, pick the class that has the largest posterior probability for the given X.

Minimum Bayes Error Rate



The area of the intersection between the two curves (assuming those curves are the real ones, not just estimates) is the **Minimum Bayes Error Rate**. Inputs that fall into that region are always classified wrong by the Bayes decision rule.

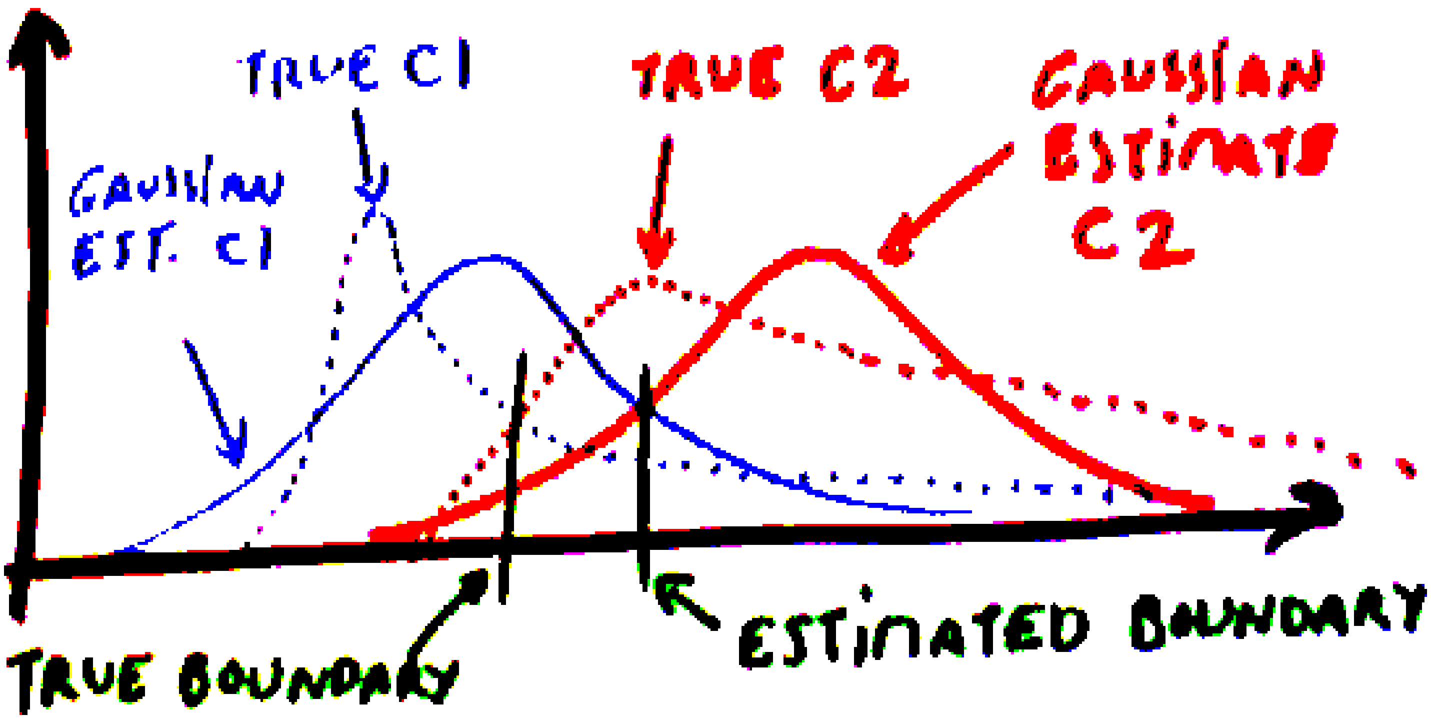
CAUTION: in practice we **never** know the "real" distributions, so we can never really compute the Bayes error rate, except in datasets that we cook up artificially by sampling from known distributions.

In real life *there is no such thing as "the distribution from which the data is sampled"*, we are just given a finite number of samples, period.

Assuming that our samples are drawn independently from some distribution is a convenient (sometimes necessary) hypothesis, but we must keep in mind that it's *wrong*.

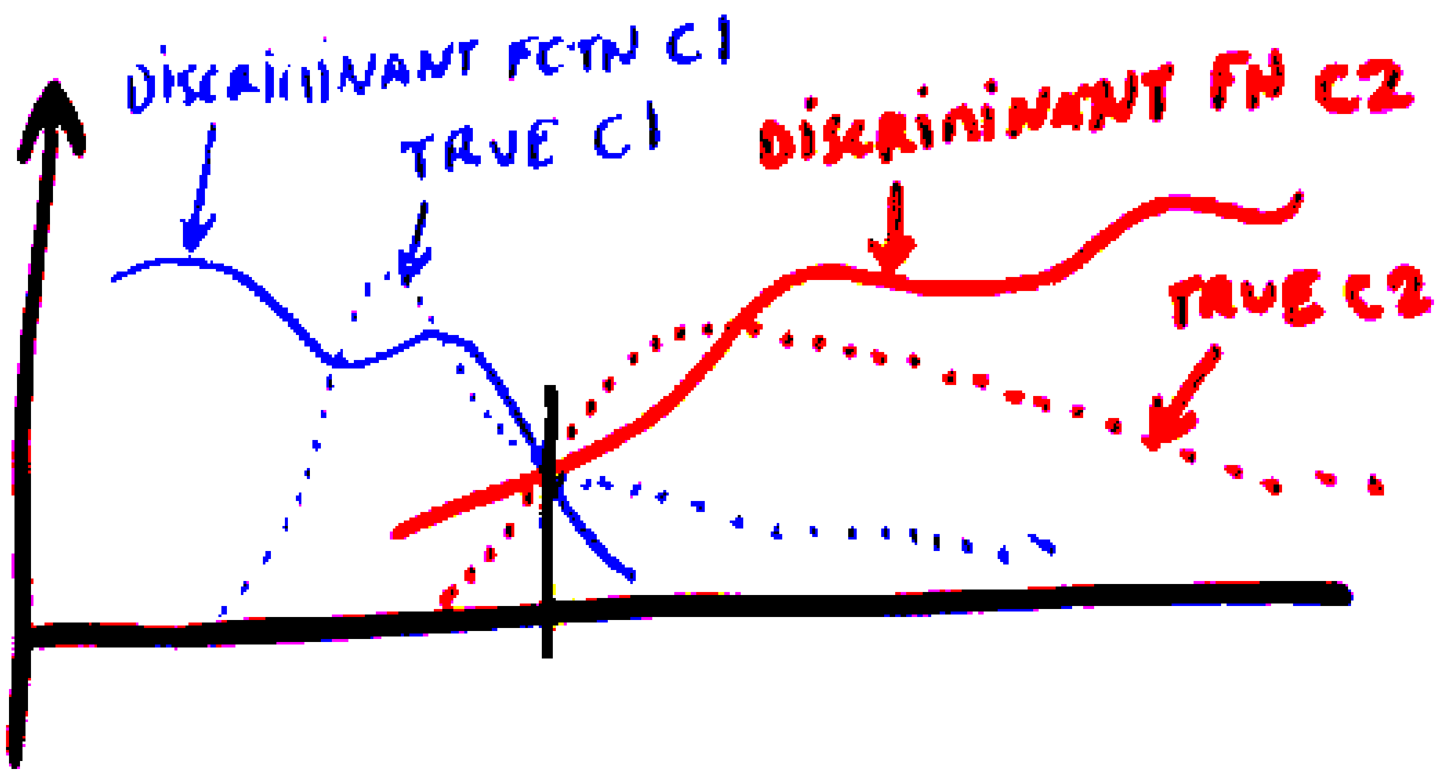
Generative Classifiers, Flawed?

A common criticism of Bayesian classifiers and other **generative** models is that they require us to solve a much more complicated problem than we have to. We are asked to solve several density estimation problems over the whole space just to come up with a decision boundary.



Discriminative Classifiers

Discriminative classifiers (such as the Perceptron) do not attempt to estimate the class densities, but simply try to find an suitable boundary (or simpy try to estimate the class posterior probabilities without going through the class densities).
This is a considerably easier problem than estimating densities over the whole space.



Naive Bayes Classifier

- The Naive Bayes classifiers is a very simple (but way suboptimal) linear classifier. It assumes independence of the input variables.
- Simple setting: two class classification problem
- Probability that X belong to class $C1$:

$$P(C1|X) = P(X|C1)P(C1)/P(X)$$

  Where $P(X)$ is simply $P(X|C1)P(C1) + P(X|C2)P(C2)$.
- Let's assume that the input variables x_i are independent, we can factorize $P(X|C1)$ as a product $\prod_i P(x_i|C1)$:

$$P(C1|X) = \frac{\prod_i P(x_i|C1)P(C1)}{P(X)}$$

Naive Bayes Classifier

- Estimating the terms $P(x_i|C1) = P(x_i, C1)/P(C1)$ is simply performed by counting of how many times the i-th input variable takes the value x_i when the sample category is $C1$, and dividing by the number of samples of class $C1$.
- To classify, we can drop the constant term $P(X)$ (which does not change from class to class). Taking logs we can write:

$$logP(C1|X) = logP(C1) + \sum_i \log[P(x_i|C1)]$$

If the variables x_i are binary (1 or 0) we can write this as

$$logP(C1|X) = logP(C1) + \sum_i (1 - x_i) \log[P(x_i = 0|C1)] + xi \log[P(x_i = 1|C1)]$$

Naive Bayes Classifier

- regrouping the terms:

$$logP(C1|X) = logP(C1) + \sum_i \log[P(x_i = 0|C1)] +$$

$$\sum_i (\log[P(x_i = 1|C1)] - \log[P(x_i = 0|C1)])x_i$$

This is just like a linear classifier of the form $W_0 + W'X$ with funny weights and biases. Naive Bayes classifiers rarely work well compared to discriminative linear classifiers.

Estimating Probabilities

- Estimating probabilities cannot be performed without a **model**, a set of independence hypotheses, and a well defined set of measurements.
- Since those choices are somewhat arbitrary, there is no such thing as "The Probability" of a real event, there are only estimates conditioned upon arbitrary assumptions.
- Example: I toss a fair coin, here is the result: 1101110011101111010000000110100...
- Now, predict the next toss.
- Method 0 [charming na iveté]: you told me it was a fair coin, so 0 and 1 are equiprobable.
- Method 1 [independent draws]: I assume that the draws are independent (the next bit does not directly depend upon the previous bits). I Just compute the empirical ratio of 1 and 0 and predict accordingly.

Estimating Probabilities

- 11011100111011110100000000110100...
- Method 2 [extra measurements]: If I use my secret super-duper measurement device, I can get a glimpse of the state of the universe within cubic kilometer around you (including your brain). With that, I can predict which side the coin will fall on with quasi-certainty (except for quantum interactions with the rest of the universe). Each bit now depends on $10^{1}00$ known bits (and an even larger number of unknown but largely irrelevant bit) through a horribly complicated function.
- Method 3 [internal structure/dependencies]: I know you cooked up this example. Those bits would not have something to do with the decimals of π by any chance?

Depending on your hypotheses and assumptions, your probability estimate may be very different from mine.

Probabilistic Linear Classification: Logistic Regression

- We want to classify vectors into two classes $C1$ and $C2$.
- We assume that the quantity $\log \frac{P(C1|X,W)}{P(C2|X,W)}$ is parameterized as a linear combination of the inputs (W is the parameter vector):

$$\log \frac{P(C1|X,W)}{P(C2|X,W)} = W'X$$

- since we only have two classes, we can write $P(C2|X,W) = 1 - P(C1|W,X)$
- hence

$$\frac{P(C1|X,W)}{1 - P(C1|X,W)} = \exp(W'X)$$

- solving for $P(C1|X,W)$, we get:

$$P(C1|X,W) = \sigma(-W'X) = \frac{1}{1 + \exp(-W'X)}$$

σ is called the logistic function.

Estimating a Logistic Regression

- How do we compute the W that best approximates the desired distribution of $P(C|X)$?
- We measure the "distance" between the desired distribution (which is given by the samples) and the proposed distribution.
- A good dissimilarity measure between two discrete distributions P and Q is the **Kullback-Leibler Divergence**:

$$KL(Q,P) = -\sum_x Q(x)\log(P(x)/Q(x))$$

- in our case:

$$L(W) = -\sum_i y^i \log(P(C1|X^i)) + (1-y^i)\log(1-P(C1|X^i))$$

where y^i is 1 if sampl X^i is of class 1, and 0 if it is of class 2.

Estimating a Logistic Regression

- Logistic regression objective function:

$$L(W) = -\sum_i y^i \log(P(C1|X^i)) + (1 - y^i) \log(1 - P(C1|X^i))$$

  where y^i is 1 if sampl X^i is of class 1, and 0 if it is of class 2.
- We can minimize $L(W)$ by gradient descent:

$$W \leftarrow W - \eta \frac{\partial L(W)}{\partial W}$$

  with

$$\frac{\partial L(W)}{\partial W} = \sum_i (y^i - \sigma(W' X^i)) X^i$$

- This looks a lot like the Perceptron learning rule