Naive Bayes

Density Estimation Problem

- $P(y|x) = P(y|x^1,x^2,\ldots,x^d)$ joint (d+1)-dim distribution
- ... actually we cannot estimate this joint
- if each feature has 10 buckets, and we have 100 features (very reasonable assumptions)
- then the joint distribution has 10^{100} cells - impossible

how to get around estimating the joint $P(x^1,x^2,\ldots,x^d|y)$?

- **SOLUTION** : assume feature independence
  - then $P(x^1,x^2,\ldots,x^d|y) = P(x^1|y)*P(x^2|y)*\ldots P(x^d|y)$
  - estimate each feature density, usually easy
  - the independence assumption rarely holds perfectly, but the model kind-of-works if it approx. holds

- it is called **NAIVE BAYES**
  - very easy to implement
  - smoothing often necessary
  - very popular

Naive Bayes

- $P(x1,x2,\ldots,xd|y) = P(x1|y)*P(x2|y)*\ldots P(xd|y)$
- d+1 joint distribution problem => d problems of simple conditional distributions
- each $P(xj|y)$ estimated separately, independent of the other features
  - **assumes features are independent**
  - **assumption doesn't really hold, but Naive Bayes still works in many cases**

how to estimate the simple distributions

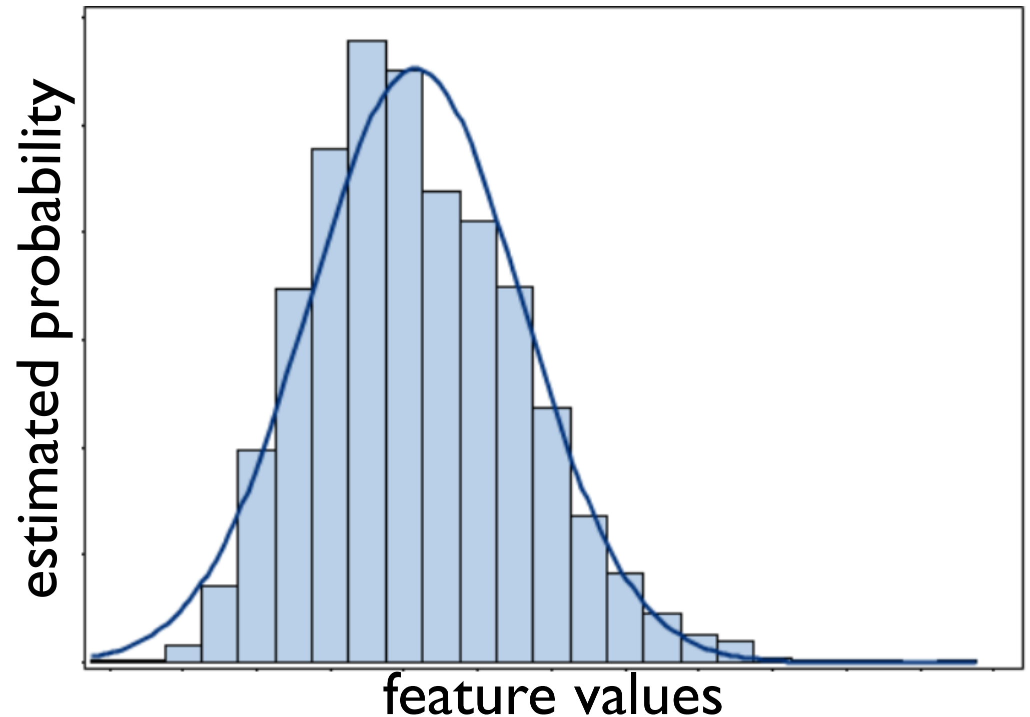
- want to estimate $P(x^j|y)$ = density of feature j values for class y
  - usually easy, since x^j is unidimensional

- **OPTION1-MODEL**: apply an imposed model, calculate Max-Likelihood parameters for the model
  - gaussian (normal), bernoulli, binomial, exponential etc
  - mixture of distributions
  - for many models, there are closed form equation stat give the max-likelihood params

how to estimate the simple distributions

- want to estimate $P(x^j|y)$ = density of feature j values for class y
  - usually easy, since x^j is unidimensional

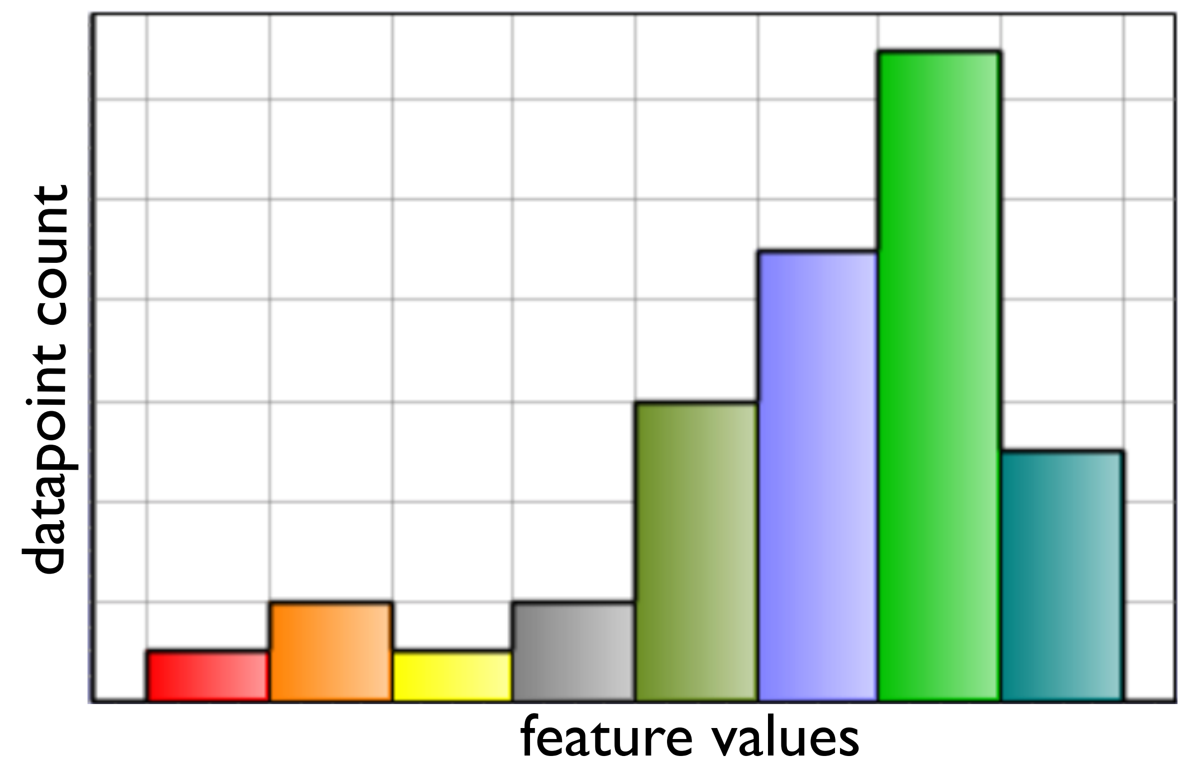
- **OPTION1-MODEL**: apply an imposed model, calculate Max-Likelihood parameters for the model



how to estimate the simple distributions

- want to estimate $P(x^j|y)$ = density of feature j values for class y
  - usually easy, since x^j is unidimensional

- **OPTION2-HISTOGRAM**: bucket/cluster/bin and count feature value in each bucket/bin



Naive Bayes problem 1: constant feature

- if x^j is constant, some estimates could be unusable
  - example: the variance of the gaussian fit is 0, and the probability of a single value is 1

- **solution: CONTROL THE PARAMETERS** (like variance) to not allow values close to zero
  - if $\Sigma < \varepsilon$ then $\Sigma = \varepsilon$

- **solution : SMOOTHING**
  - generally a good idea for all probability estimates

- **solution: FEATURE SELECTION**
  - discussed later in the course

Naive Bayes Problem 2: “zero probability”

- in the case of histograms (bins), estimate of zero probability is quite possible
  - when there are many bins, and not so many data points

- especially true for text documents, when features are word occurrences
  - there are many words, and most of them do not appear in most documents
  - probability estimate by count often gives 0 probability

- solution : **SMOOTHING** the estimate

Smoothing: Laplace

- N possibilities / cases
- $t_1, t_2, t_3, \ldots, t_N$ observed counts for each case
- $M = t_1 + t_2 + t_3 + \ldots + t_N$ number of observations
- direct estimate $P(i) = t_i / M$
- Laplace estimate $P(i) = (t_i + 1) / (M+N)$
  - note that Laplace P(i) still sum to 1

Smoothing: Foreground and Background

- N possibilities / cases
- t1, t2, t3, ... , tN observed counts for each case
- M = t1 + t2 + t3 + ... + tN number of observations
- direct (foreground) estimate P(i) = ti / M

- Background estimate in a larger setting
  - each experiment j has Nj, Mj, tij etc
- $Q(i) = (\Sigma j\ tij) / (\Sigma j\ Mj)$ background probability
  - note that Laplace P(i) still sum to 1

- smoothed estimate $Prob(i) = \lambda P(i) + (1-\lambda)Q(i)$
  - note that smoothed estimates still sum to 1

Naive Bayes overview

- Training
  - $P(x|y) = P(x^1,x^2,\ldots,x^d|y) = P(x^1|y)*P(x^2|y)*\ldots P(x^d|y)$
  - estimate separately each $P(x^j|y)$ from training
  - store the model

- Testing
  - for datapoint x apply the estimates to compute $P(x|y) = P(x^1,x^2,\ldots,x^d|y) = P(x^1|y)*P(x^2|y)*\ldots P(x^d|y)$
  - use Bayes Rule $P(y|x) = P(x|y)*\ P(y)\ /\ P(x)$
  - predict y that maximizes $P(x|y)*\ P(y)$