Probabilistic ML algorithm

Naïve Bayes and Maximum Likelyhood

Axioms of Probability Theory

- All probabilities between 0 and 1

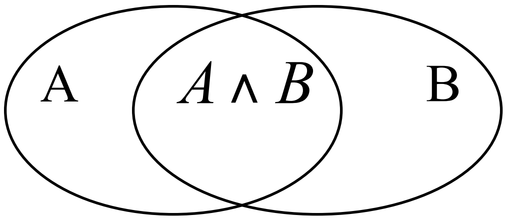
$$0 \le P(A) \le 1$$

- True proposition has probability 1, false has probability 0.

P(true) = 1   P(false) = 0.

- The probability of disjunction is:

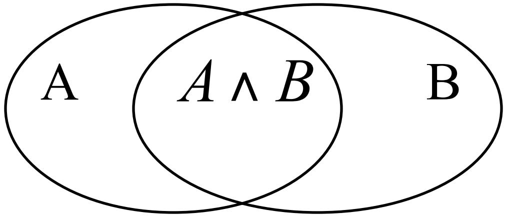
$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$



Conditional Probability

- P(A | B) is the probability of A given B
- Assumes that B is all and only information known.
- Defined by:

$$P(A \mid B) = \frac{P(A \wedge B)}{P(B)}$$



Independence

- A and B are *independent* iff:

$$P(A \mid B) = P(A)$$

$$P(B \mid A) = P(B)$$

These two constraints are logically equivalent

- Therefore, if A and B are independent:

$$P(A \mid B) = \frac{P(A \wedge B)}{P(B)} = P(A)$$

$$P(A \wedge B) = P(A)P(B)$$

Joint Distribution

- The joint probability distribution for a set of random INDEPENDENT variables, $X_1,\ldots,X_d$ gives the probability of every combination of values (an n-dimensional array with K values if all variables are discrete with K values, all K *prob* values must sum to 1):

Pr(shape=circle, color=blue, C=+)

Class=positive

	circle	square
red	0.20	0.02
blue	0.02	0.01

Class=negative

	circle	square
red	0.05	0.30
blue	0.20	0.20

- The probability of all possible conjunctions (assignments of values to some subset of variables) can be calculated by summing the appropriate subset of values from the joint distribution.

$$P(red \wedge circle) = P(red \wedge circle \wedge positive) + P(red \wedge circle \wedge negative) = 0.20 + 0.05 = 0.25$$

$$P(red) = P(red \wedge circle \wedge positive) + P(red \wedge square \wedge positive) + ecc = 0.20 + 0.02 + 0.05 + 0.3 = 0.57$$

- Therefore, all conditional probabilities can also be calculated.

$$P(positive \mid red \wedge circle) = \frac{P(positive \wedge red \wedge circle)}{P(red \wedge circle)} = \frac{0.20}{0.25} = 0.80$$

Probabilistic Classification

- Let Y be the random variable for the class C which takes values $\{y_1,y_2,\ldots y_m\}$ ($|C|=m$ possible classifications for our instances).
- Let X be the random variable describing an instance consisting of a vector of values for d features $<X_1,X_2\ldots X_d>$, let v_{jk} be a possible value for X_j ($\boldsymbol{x}_k$ is an instance in X and v_{jk} is the value of feature X_j for $\boldsymbol{x}_k$) .
- For our classification task, we need to compute:

$P(Y=y_i \mid X=x_k)$ for $i=1\ldots m$

(e.g. $P(Y=positive/\mathbf{x_k}=<blue,circle>)$)

- E.g. the objective is to classify a new unseen $\mathbf{x_k}$ by estimating the probability of each possible classification y_i , **given** the feature values of the instance to be classified $\mathrm{x_k}:<X_1=\mathrm{v_{1k}},X_2=\mathrm{v_{2k2}}\ldots X_d=v_{\mathrm{dk}}>$
- To estimate $P(Y=y_i \mid X=x_k)$ we use a learning set D of pairs $(\mathrm{x_i},C(\mathrm{x_i}))$

Probabilistic Classification (2)

- However, given no other assumptions, this requires a table **giving the probability of each category for each possible instance (combination of feature values) in the instance space,** which is impossible to accurately estimate from a reasonably-sized training set.
- E.g. $\Pr(Y=y_i / X_1=v_{1k}, X_2=v_{2k2} \ldots X_d=v_{dk})$
  - Assuming that Y and all X_i are binary, and we have d features, we need 2^d entries to specify

  $P(Y=1 \mid X=x_k)$ for each of the 2^d possible x_k since:
  - $P(Y=0 \mid X=x_k) = 1 - P(Y=1 \mid X=x_k)$
  - Compared to $2^{d+1} - 1$ entries for the joint distribution $P(Y, X_1, X_2 \ldots X_d)$

Example

- $X:(X_1,X_2..X_4)$, $Xi:\left\{0,1\right\}$ $Y:\left\{0,1\right\}$ (d=4, m=2)
- $x_k:(0,1,0,0)$
- Need to estimate Pr(Y=0/(0,1,0,0))
- If P(Y=0/(0,1,0,0))>(1-P(Y=0/(0,1,0,0))) then class is 0, else class is 1
- Overall, 2^4 estimates are needed for our probabilistic classifier
- For large m and n this is not feasible

Maximum Likelihood learning

- We have a probabilistic model, *M*, of some phenomena. We know exactly the structure of *M* *(e.g. a Gaussian),* but not the values of its probabilistic parameters, Θ (e.g. μ,σ).
- Each "execution" of *M* produces an observation, *x[i]* , according to the (unknown) distribution induced by *M.*

◆ Goal: After observing x[1] ,…, x[n] , estimate the model parameters, Θ, that generated the observed data.

Maximum Likelihood Estimation (MLE)

- The likelihood of the observed data, given the model parameters Θ, is the conditional probability that the model, M, with parameters Θ, produces the observations $x_1, \ldots, x_m$.

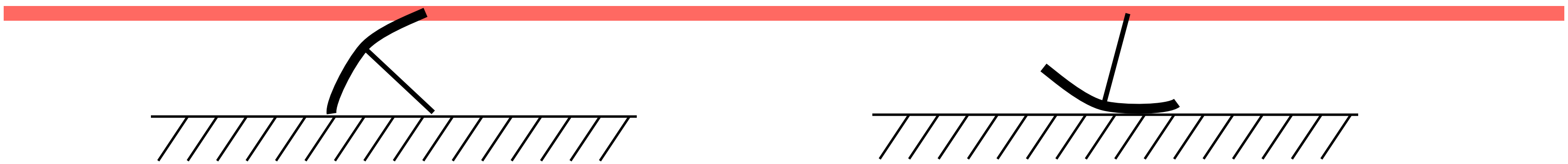
$$L(\Theta) = \Pr(x[1], \ldots, x[n] \mid \Theta, M),$$

- In MLE we seek the model parameters, Θ, that maximize the likelihood.

Maximum Likelihood Estimation (MLE)

- In MLE we seek the model parameters, Θ, that maximize the likelihood.
- The MLE principle is applicable in a wide variety of ML applications, from speech recognition, through natural language processing, to computational biology.
- We will start with the simplest example: Estimating the bias of a thumbtack.

Example: Binomial Experiment



- When tossing the thumbtack, it can land in one of two positions: *Head* ***(H)*** or *Tail* ***(T)***
- ◆We denote by θ the (unknown) probability **P(H)**.

Estimation task:

◆ Given a sequence of toss samples **x1..xm** we want to estimate the probabilities **P(H)=**θ and **P(T) = 1 -** θ

◆**θ is also called the model parameter**

Statistical Parameter Fitting (general definition)

- Consider instances D: $x_1, x_2, \ldots, x_m$ such that
  - The set of values that x can take is known
  - Each is sampled from the same distribution
  - Each sampled independently of the rest

  i.i.d. Samples

- The task is to find a vector of parameters Θ that have generated the given data. This vector parameter Θ can be used to predict future data.

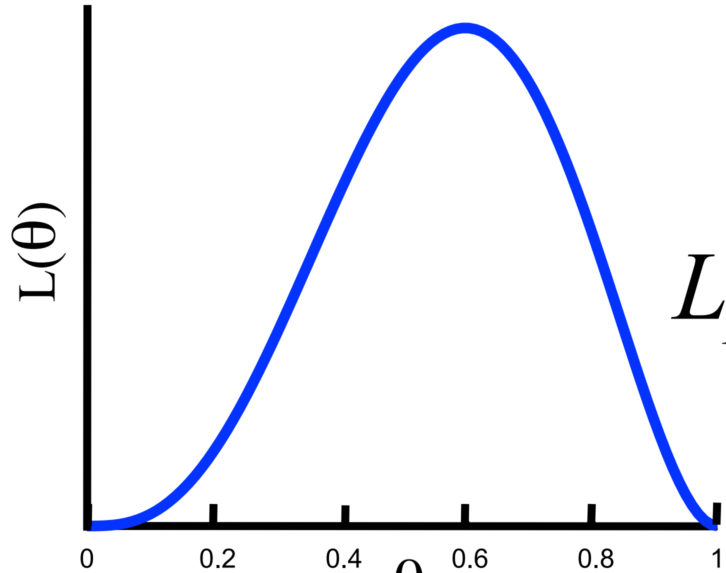
The Likelihood Function

- How good is a particular θ?

  It depends on how likely it is to generate the observed data

$$L_D(\theta) = P(D \mid \theta) = \prod_{j=1..m} P(xj \mid \theta)$$

- The likelihood for the sequence H,T, T, H, H is



$$L_D(\theta) = \theta \cdot (1-\theta) \cdot (1-\theta) \cdot \theta \cdot \theta$$

Sufficient Statistics

- To compute the likelihood in the thumbtack example we only require N_H and N_T (the number of heads and the number of tails)

$$L_D(\theta) = \theta^{N_H} \cdot (1-\theta)^{N_T}$$

- N_H and N_T are **sufficient statistics** for the binomial distribution
- A sufficient statistic is a function whose value contains all the information needed to compute any estimate of the parameter

Maximum Likelihood Estimation

MLE Principle:

Choose parameters that maximize the likelihood function

- This is one of the most commonly used estimators in statistics
- Intuitively appealing
- One usually maximizes the log-likelihood function, defined as $l_D(\theta) = \ln L_D(\theta)$

Example: MLE in Binomial Data

$$l_D(\theta) = N_H \log\theta + N_T \log(1-\theta)$$

Taking derivative and equating it to 0 we get

$$\frac{N_H}{\theta} = \frac{N_T}{1-\theta} \Rightarrow \hat{\theta} = \frac{N_H}{N_H + N_T}$$

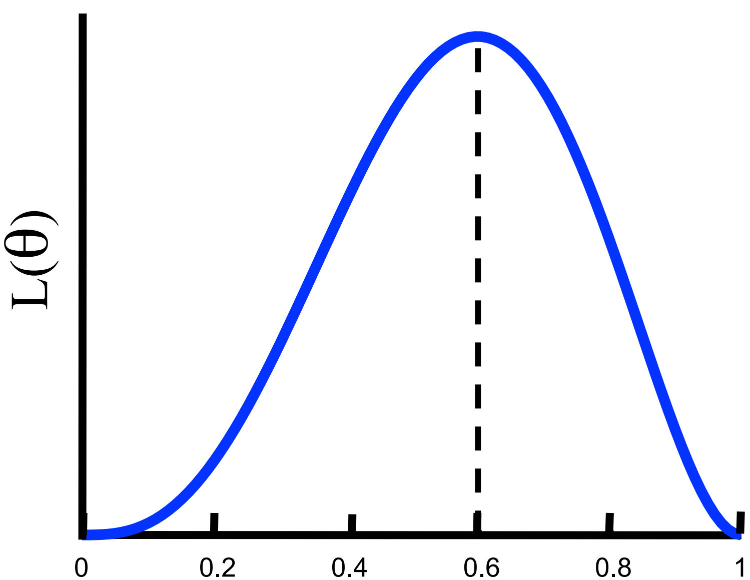
Remember, to maximize minimize a function you need to take the derivative

(which coincides with what one would expect)

Example:

$(N_H, N_T) = (3,2)$

MLE estimate is $3/5 = 0.6$



From Binomial to Multinomial

- Now suppose X can have the values $1,2,\ldots,K$ *(For example a die has K=6 sides)*
- We want to learn the parameters $\theta_1, \theta_2, \ldots, \theta_K$ *(the vector Θ)*

Sufficient statistics:

◆$N_1, N_2, \ldots, N_K$ - the number of times each outcome is observed

$$L_D(\theta) = \prod_{k=1}^{K} \theta_k^{N_k} \quad s.t. \sum_k \theta_k = 1 \ \ and \ \ \theta_k \geq 0 \ \ \forall k$$

$$\hat{\theta}_k = \frac{N_k}{\sum_j Nj}$$

Lagrangian (again)

$$L(\alpha, \Theta) = \sum N^k log\theta_k - \alpha(\sum \theta_k - 1)$$

$$\frac{dL(\alpha, \Theta)}{d\theta_k} = 0$$

$$\frac{N^k}{\theta_k} - \alpha = 0 \implies \theta_k = \frac{N^k}{\alpha}$$

$$\sum \frac{N^k}{\alpha} = 1 \Rightarrow \alpha = \sum N^k$$

$$\hat{\theta}_k = \frac{N_k}{\sum_j Nj}$$

Example: Multinomial

- Let $x_1 x_2 \ldots. x_n$ be a protein sequence
- We want to learn the parameters $\theta_1, \theta_2, \ldots, \theta_{20}$ corresponding to the probabilities of the 20 amino acids
- $N_1, N_2, \ldots, N_{20}$ - the number of times each amino acid is observed in the sequence

Likelihood function:

$$L_D(q) = \prod_{k=1}^{20} \theta_k^{N_k}$$

MLE: $$\theta_k = \frac{N_k}{n} \quad n = \sum_{i=1}^{20} N_i$$

NAIVE BAYES CLASSIFIER

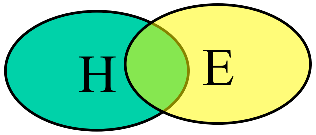
Bayes Theorem

- H=hypothesis
- E= evidence of data

$$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)}$$

Simple proof from definition of conditional probability:

$$P(H \mid E) = \frac{P(H \wedge E)}{P(E)}$$ (Def. cond. prob.)



$$P(E \mid H) = \frac{P(H \wedge E)}{P(H)}$$ (Def. cond. prob.)

$$P(H \wedge E) = P(E \mid H)P(H)$$

QED: $$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)}$$

Bayesian Categorization

For each classification value y_i we have (applying Bayes):

$$P(Y = y_i \mid X = x_k) = \frac{P(Y = y_i)P(X = x_k \mid Y = y_i)}{P(X = x_k)}$$

- P(Y=yi) and P(X=x_k) are called **p** estimated from learning set D sin **complete** and **disjoint**

$$P(A) = \sum_{\forall i} P(B_i)P(A / B_i) \ \ if$$

$$\sum_{\forall i} P(B_i) = 1 \ \ and$$

$$\forall i \neq j \ \ P(B_i \wedge B_j) = P(B_i)P(B_j)$$

$$\sum_{i=1}^{m} P(Y = y_i \mid X = x_k) = \sum_{i=1}^{m} \frac{P(Y = y_i)P(X = x_k \mid Y = y_i)}{P(X = x_k)} = 1$$

$$P(X = x_k) = \sum_{i=1}^{m} P(Y = y_i)P(X = x_k \mid Y = y_i)$$

Complete and Disjoint

- Complete: Y can only assume values in

$$\{y_1, y_2, \ldots y_m\}$$

- Disjoint: $y_1 \cap y_2 \ldots . \cap y_m = \varnothing$
- If a set of categories is complete and disjoint, X is a random variable, and x_k is any of its possible values, then:

$$P(X = xk) = \sum_{i=1..m} P(X = xk / Y = y_i) P(Y = y_i)$$

Bayesian Categorization (cont.)

- To estimate $P(Y=y_i|X=x_k)$ need to know the following parameters:
  - Priors: $P(Y=y_i)$
  - Conditionals: $P(X=x_k \mid Y=y_i)$
  - Note we don't need to estimate $P(X=x_k)$ since the denominator is common to all $P(Y=y_i|X=x_k)$ (therefore it does not change the rank)
  - Therefore the model parameters are:

$$\theta_i^1 = P(Y = y_i);\ \theta_{ki}^2 = P(X = x_k / Y = y_i)$$

MLE for Naive Bayes

$$L(\Theta) = \sum N^i log\theta_i^1 + \sum\sum N^{ki} \log\theta_{ki}^2$$

Subject to: $\sum \theta_i^1 = 1 \quad \theta_i^1 \geq 0; \quad \sum_i \theta_{ki}^2 = 1 \quad \theta_{ki}^2 \geq 0$

Lagrangian:

$$L(\Theta) = \sum N^i log\theta_i^1 + \sum\sum N^{ki} \log\theta_{ki}^2 - \alpha \sum(\theta_i^1 - 1) - \beta \sum(\theta_{ki}^2 - 1)$$

$$\frac{dL(\alpha, \beta, \Theta)}{d\theta_k} = 0$$

Estimation of P(Y=yi)

$$\frac{dL(\alpha,\beta,\Theta)}{d\theta_i^1} = \frac{N^i}{\theta_i^1} - \alpha = 0 \Rightarrow \theta_i^1 = \frac{N^i}{\alpha}$$

$$\sum \frac{N^i}{\alpha} = 1 \Rightarrow \alpha = \sum N^i = |D| = N$$

$$\theta_i^1 = \frac{N^i}{N}$$

Remember N_i = numer of times $Y=y_i$ in the learning set D

Estimating $P(X=x_k/Y=y_i)$

$$L(\Theta) = \sum N^{i} log\theta_i^1 + \sum\sum N^{ki} \log\theta_{ki}^2 - \alpha \sum(\theta_i^1 - 1) - \beta \sum(\theta_{ki}^2 - 1)$$

$$\frac{dL(\alpha,\beta,\Theta)}{d\theta_k} = 0$$

The evidence for N^{ki} is likely to be very small: remember it is the number of times x_k has classification y_i; we only have in D a limited number of instances, and they should have a single classification, therefore for most (likely all) (k,i) we have $N^{ki}=0$ (no evidence) or $N^{ki}=1$ (1 sample)

Estimating $P(X=x_k/Y=y_i)$ (2)

- Naive Bayes assumption:

$$P(Y=y_i \mid X=x_k) = P(Y=y_i)P(X_1^k = v1, X_2^k = v2, \cdots X_n^k = vn \mid Y=y_i) / P(X=x_k) =$$

$$P(Y=y_i)\prod_{j=1}^{d} P(vjk \mid Y=y_i) / P(X=x_k)$$

We assume feature values v_{jk} of different features X_j being statistically independent. v_{jk} is the k-th value of feature j where j=1,2..d and k=1…K_j (if binary features, k=0 or 1)

e.g. P(x(color=blue, shape=circle, dimension=big))= P(color=blue)P(shape=circle)P(dimension=big)

and furthermore

$$\sum_{k \in colors} P(color = k / Y = y_i) = 1$$

Estimating $P(X=x_k/Y=y_i)$ (3)

$$\theta^2_{jki} = P(X_j = v_{jk} \, / Y = y_i)$$

$$L(\Theta) = \sum N^i log\theta^1_i + \sum \prod N^{jki} log\theta^2_{jki}$$

The parameter $P(X=x_k/Y=y_i)$, i.e. the probability that a given instance x_k has a given classification y_i, is replaced with the probability that a given feature **value** v_{jk} of feature X_j has a given classification y_i

j=1…d; k=1..Kj ; i=1..|C|; |D|=N

Estimating $P(X=x_k/Y=y_i)$ (4)

The new MLE problem is therefore:

$$\sum \theta_i^1 = 1 \quad \theta_i^1 \geq 0; \ \sum_k \theta_{jki}^2 = 1 \ \ \theta_{jki}^2 \geq 0$$

$$L(\Theta) = \sum N^i log\theta_i^1 + \sum \prod N^{jki} log\theta_{jki}^2 - \alpha \sum_{i=1..|C|} (\theta_i^1 - 1) - \beta \sum_{k=1...K} (\theta_{jki}^2 - 1)$$

The computation of parameters θ^1 does not change (the derivative is the same)

Also note that

$$\sum \prod N^{jki} log\theta_{jki}^2 = \sum \sum N^{jki} log\theta_{jki}^2$$

Estimating $P(X=x_k/Y=y_i)$ (4)

$$\frac{\partial L(\alpha,\beta,\Theta)}{\partial \theta_{jki}^2} = \frac{N^{jki}}{\theta_{jki}^2} - \beta = 0 \rightarrow \theta_{jki}^2 = \frac{N^{jki}}{\beta}$$

$$\textit{and since} \sum_{k=1..K} \theta_{jki}^2 = 1$$

$$\textit{we obtain } \theta_{jki}^2 = \frac{N^{jki}}{\sum_k \theta_{jki}^2} = \frac{N^{jki}}{N^i}$$

I.e. the θ^2 can be estimated as the ratio between the number of times feature X_j takes value k when $Y=y_i$ and the total number of examples in D for which $Y=y_i$

How do we compute the category of an instance?

$$P(Y = y_i | X = x_k) = P(Y = y_i) \prod_{j=1..d} P(v_{jk} | Y = y_i) / \prod_{j=1..d} P(v_{jk}) =$$

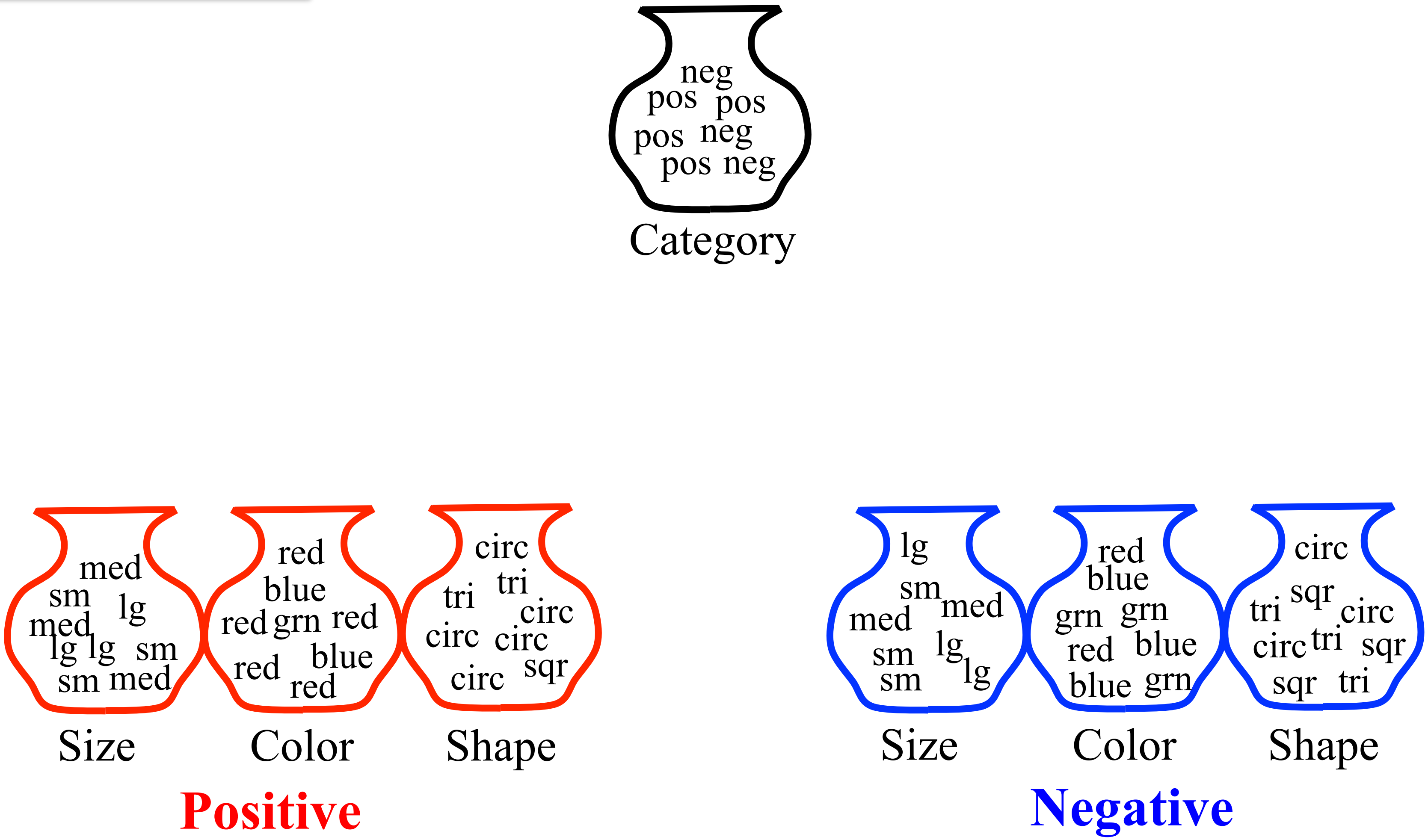
$$\theta_i^1 \prod_{j=1..d} \theta_{ikj}^2 / \prod_{j=1..d} P(v_{jk})$$

$$y_i = \underset{k}{\operatorname{argmax}}(\theta_i^1 \prod_{j=1..d} \theta_{ikj}^2)$$

Note that since the denominator is common to all conditional probabilities, it does not affect the argmax

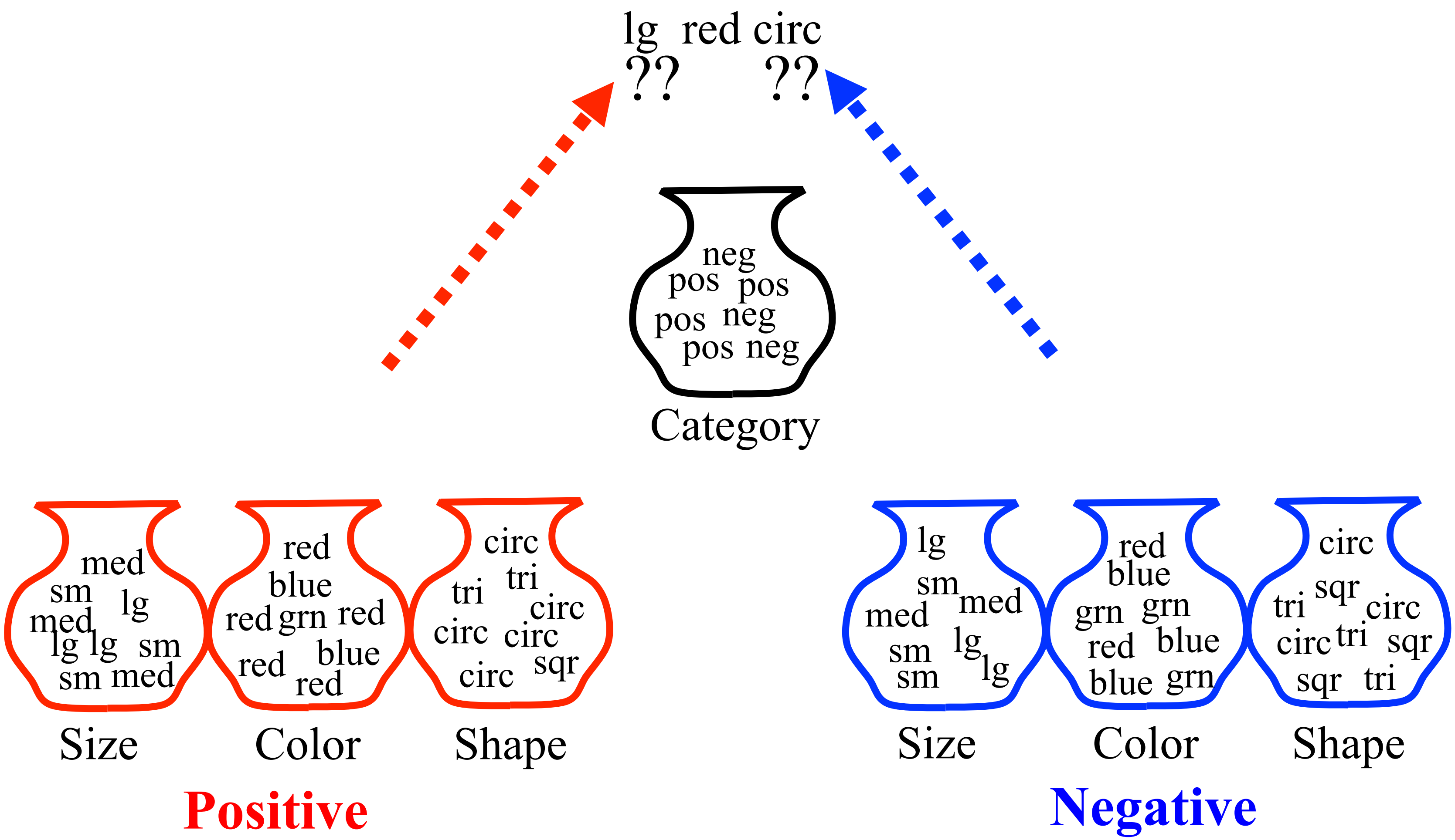
Naïve Bayes Generative Model

K=3,|C|=2,d=3



Naïve Bayes Inference Problem

I estimate on the learning set the probability of extracting lg, red, circ from the red or blue urns.

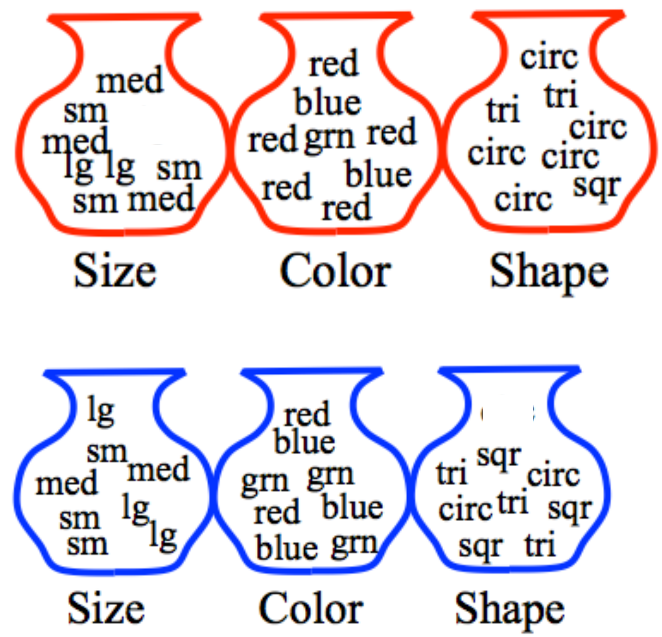


HOW?

Naïve Bayes Example

$$\theta_{size,small,positive} = P(size = small/C = positive)$$

	Probability	Y=positive	Y=negative
	P(*Y*)	0.5	0.5
Shape	P(small | *Y*)	3/8	3/8
	P(medium | *Y*)	3/8	2/8
	P(large | *Y*)	2/8	3/8
Color	P(red | *Y*)	5/8	2/8
	P(blue | *Y*)	2/8	3/8
	P(green | *Y*)	1/8	3/8
Size	P(square | *Y*)	1/8	3/8
	P(triangle | *Y*)	2/8	3/8
	P(circle | *Y*)	5/8	2/8



Training set

Have 3 small out of 8 instances in red “size” urn then P(size=small/pos)=3/8=0,375 (round 4)

Naïve Bayes Example

Probability	positive	negative
P(*Y*)	0.5	0.5
P(medium | *Y*)	3/8	2/8
P(red | *Y*)	5/8	2/8
P(circle | *Y*)	5/8	2/8

$$y_i = \operatorname*{argmax}_k \left(\theta_i^1 \prod_{j=1..d} \theta_{ikj}^2\right)$$

Test Instance:
X:<medium ,red, circle>

P(positive | *X*) = P(positive)*P(medium | positive)*P(red | positive)*P(circle | positive)/P(X)

0.5 * 3/8 * 5/8 * 5/8

= 0,073

P(negative | *X*) = P(negative)*P(medium | negative)*P(red | negative)*P(circle | negative) / P(*X*)

0.5 * 2/8 * 2/8 * 2/8

=0.0078

P(positive/X)>P(negative/X) ➔ positive

Naive summary

Classify any new datum instance $\mathbf{x_k}=(x_1, \ldots x_n)$ *as:*

$$y_{Naive\ Bayes} = \underset{i}{\operatorname{argmax}} P(y_i)P(\mathbf{x} \mid y_i) = \underset{i}{\operatorname{argmax}} P(y_i) \prod_{j=1..d} P(v_{jk} \mid y_i)$$

- To do this based on training examples, estimate the parameters from the training examples in D:
  - For each target value of the classification variable (hypothesis) y_i

    $$\hat{P}(Y = y_j) := \textbf{estimate } P(y_i)$$

  - For each attribute value a_t of each datum instance

    $$\hat{P}(x_j = v_{jk} \mid Y = yi) := \textbf{estimate } P(v_{jk} \mid y_i)$$

Estimating Probabilities

- Normally, as in previous example, probabilities are estimated based on observed frequencies in the training data.
- If D contains N_i examples in category y_i, and N_{jki} of these N_i examples have the k-th value for feature X_j, v_{jk}, then:

$$P(X_j = v_{jk} \mid Y = y_i) = \frac{N_{jki}}{N_i}$$

- However, estimating such probabilities from small training sets is error-prone.
- If due only to chance, a rare feature, X_j, is always false in the training data, $\forall y_k$:P(X_j=true | Y=y_i) = 0.
- If X_j=true then occurs in a test example, X, the result is that $\forall$y$_k$: P(X | Y=y$_i$) = 0 and $\forall$y$_i$: P(Y=y$_i$ | X) = 0

Probability Estimation Example

Ex	Size	Color	Shape	Category
1	small	red	circle	positive
2	large	red	circle	positive
3	small	red	triangle	negitive
4	large	blue	circle	negitive

Probability	positive	negative
P(*Y*)	0.5	0.5
P(small | *Y*)	0.5	0.5
P(medium | *Y*)	0.0	0.0
P(large | *Y*)	0.5	0.5
P(red | *Y*)	1.0	0.5
P(blue | *Y*)	0.0	0.5
P(green | *Y*)	0.0	0.0
P(square | *Y*)	0.0	0.0
P(triangle | *Y*)	0.0	0.5
P(circle | *Y*)	1.0	0.5

Test Instance *X*:
<medium, red, circle>

P(positive | *X*) = 0.5 * 0.0 * 1.0 * 1.0 / P(X) = 0

P(negative | *X*) = 0.5 * 0.0 * 0.5 * 0.5 / P(X) = 0

Smoothing

- To account for estimation from small samples, probability estimates are adjusted or *smoothed*.
- Laplace smoothing using an *m*-estimate assumes that each feature is given a prior probability, *p*, that is assumed to have been previously observed in a "virtual" sample of size *m*.

$$P(Xj = v_{jk} \mid Y = y_i) = \frac{N_{jki} + mp}{N_i + m}$$

- For binary features, *p* is simply assumed to be 0.5.

Laplace Smothing Example

- Assume training set contains 10 positive examples:
  - 4: small
  - 0: medium
  - 6: large
- Estimate parameters as follows (if m=1, p=1/3)
  - P(small | positive) = (4 + 1/3) / (10 + 1) = 0.394
  - P(medium | positive) = (0 + 1/3) / (10 + 1) = 0.03
  - P(large | positive) = (6 + 1/3) / (10 + 1) = 0.576
  - P(small or medium or large | positive) = 1.0

Continuous Attributes

- If X_i is a **continuous** feature rather than a discrete one, need another way to calculate $P(X_j \mid Y)$.
- Assume that X_j has a Gaussian distribution whose mean and variance depends on Y.
- During training, for each combination of a continuous feature X_j and a class value for Y, y_i, estimate a mean, μ_{ji} , and standard deviation σ_{ji} based on the values of feature X_j in class y_i in the training data. μ_{ji} is the mean value of Xj observed over instances for which $Y = y_i$ in D
- **During testing**, estimate $P(X_j \mid Y{=}y_i)$ for a given example, using the Gaussian distribution defined by μ_{ji} and σ_{ji} .

$$P(Xj = v_{jk} \mid Y = y_i) = \frac{1}{\sigma_{ji}\sqrt{2\pi}} \exp\left(\frac{-(X_i - \mu_{ji})^2}{2\sigma_{ji}^2} \right)$$

Comments on Naïve Bayes

- Tends to work well despite strong assumption of conditional independence.
- Experiments show it to be quite competitive with other classification methods on standard UCI datasets.
- Although it does not produce accurate probability estimates when its independence assumptions are violated, it may still pick the correct maximum-probability class in many cases.
  - Able to learn conjunctive concepts in any case
- Does not perform any search of the hypothesis space. Directly constructs a hypothesis from parameter estimates that are easily calculated from the training data.
  - Strong bias
- Not guarantee consistency with training data.
- Typically handles noise well since it does not even focus on completely fitting the training data.