CS 391L: Machine Learning: Bayesian Learning: Naïve Bayes

Raymond J. Mooney
University of Texas at Austin

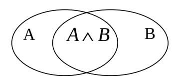
Axioms of Probability Theory

- 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:

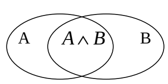
  $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$



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 variables, $X_1,\ldots,X_n$ gives the probability of every combination of values (an n-dimensional array with v^n values if all variables are discrete with v values, all v^n values must sum to 1): P($X_1,\ldots,X_n$)

positive

	circle	square
red	0.20	0.02
blue	0.02	0.01

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.

$$P(red \wedge circle) = 0.20 + 0.05 = 0.25$$
$$P(red) = 0.20 + 0.02 + 0.05 + 0.3 = 0.57$$

- 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 which takes values $\{y_1,y_2,\ldots y_m\}$.
- Let X be the random variable describing an instance consisting of a vector of values for n features $<X_1,X_2\ldots X_n>$, let x_k be a possible value for X and x_{ij} a possible value for X_i.
- For classification, we need to compute P(Y=y_i | X=x_k) for i=1…m
- However, given no other assumptions, this requires a table giving the probability of each category for each possible instance in the instance space, which is impossible to accurately estimate from a reasonably-sized training set.
  - Assuming Y and all X_i are binary, we need 2^n entries to specify P(Y=pos | X=x_k) for each of the 2^n possible x_k's since P(Y=neg | X=x_k) = 1 – P(Y=pos | X=x_k)
  - Compared to 2^{n+1} – 1 entries for the joint distribution P(Y,X_1,X_2…X_n)

Bayes Theorem

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

Simple proof from definition of conditional probability:

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

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

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

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

Bayesian Categorization

- Determine category of x_k by determining for each y_i

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

- $P(X=x_k)$ can be determined since categories are complete and disjoint.

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

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

Bayesian Categorization (cont.)

- Need to know:
  - Priors: $P(Y=y_i)$
  - Conditionals: $P(X=x_k \mid Y=y_i)$
- $P(Y=y_i)$ are easily estimated from data.
  - If n_i of the examples in D are in y_i then $P(Y=y_i) = n_i / |D|$
- Too many possible instances (e.g. 2^n for binary features) to estimate all $P(X=x_k \mid Y=y_i)$.
- Still need to make some sort of independence assumptions about the features to make learning tractable.

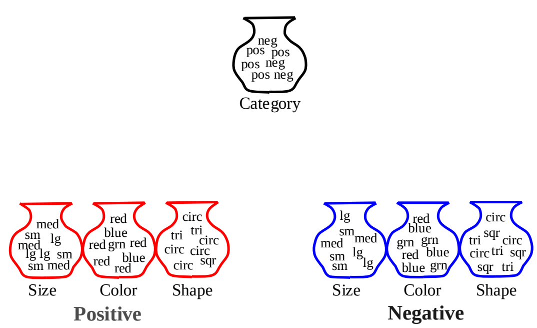
Generative Probabilistic Models

- Assume a simple (usually unrealistic) probabilistic method by which the data was generated.
- For categorization, each category has a different parameterized generative model that characterizes that category.
- **Training**: Use the data for each category to estimate the parameters of the generative model for that category.
  - **Maximum Likelihood Estimation (MLE):** Set parameters to maximize the probability that the model produced the given training data.
  - If M_λ denotes a model with parameter values λ and D_k is the training data for the *k*th class, find model parameters for class *k* (λ_k) that maximize the likelihood of D_k:

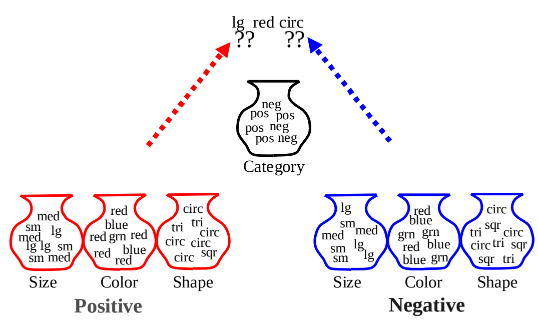
$$\lambda_k = \underset{\lambda}{\operatorname{argmax}}\, P(D_k \mid M_\lambda)$$

- **Testing**: Use Bayesian analysis to determine the category model that most likely generated a specific test instance.

Naïve Bayes Generative Model



Naïve Bayes Inference Problem



Naïve Bayesian Categorization

- If we assume features of an instance are independent **given the category** (***conditionally independent***).

$$P(X \mid Y) = P(X_1, X_2, \cdots X_n \mid Y) = \prod_{i=1}^{n} P(X_i \mid Y)$$

- Therefore, we then only need to know $P(X_i \mid Y)$ for each possible pair of a feature-value and a category.
- If Y and all X_i and binary, this requires specifying only $2n$ parameters:
  - $P(X_i\text{=true} \mid Y\text{=true})$ and $P(X_i\text{=true} \mid Y\text{=false})$ for each X_i
  - $P(X_i\text{=false} \mid Y) = 1 - P(X_i\text{=true} \mid Y)$
- Compared to specifying 2^n parameters without any independence assumptions.

13

Naïve Bayes Example

Probability	positive	negative
P(*Y*)	0.5	0.5
P(small | *Y*)	0.4	0.4
P(medium | *Y*)	0.1	0.2
P(large | *Y*)	0.5	0.4
P(red | *Y*)	0.9	0.3
P(blue | *Y*)	0.05	0.3
P(green | *Y*)	0.05	0.4
P(square | *Y*)	0.05	0.4
P(triangle | *Y*)	0.05	0.3
P(circle | *Y*)	0.9	0.3

Test Instance:
<medium ,red, circle>

14

Naïve Bayes Example

Probability	positive	negative
P(*Y*)	0.5	0.5
P(medium | *Y*)	0.1	0.2
P(red | *Y*)	0.9	0.3
P(circle | *Y*)	0.9	0.3

Test Instance:
<medium ,red, circle>

P(positive | *X*) = P(positive)*P(medium | positive)*P(red | positive)*P(circle | positive) / P(*X*)
0.5 * 0.1 * 0.9 * 0.9
= 0.0405 / P(*X*) = 0.0405 / 0.0495 = 0.8181

P(negative | *X*) = P(negative)*P(medium | negative)*P(red | negative)*P(circle | negative) / P(*X*)
0.5 * 0.2 * 0.3 * 0.3
= 0.009 / P(*X*) = 0.009 / 0.0495 = 0.1818

P(positive | *X*) + P(negative | *X*) = 0.0405 / P(*X*) + 0.009 / P(*X*) = 1

P(*X*) = (0.0405 + 0.009) = 0.0495

15

Estimating Probabilities

- Normally, probabilities are estimated based on observed frequencies in the training data.
- If D contains n_k examples in category y_k, and n_{ijk} of these n_k examples have the jth value for feature X_i, x_{ij}, then:

$$P(X_i = x_{ij} \mid Y = y_k) = \frac{n_{ijk}}{n_k}$$

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

16

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

17

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.

$$P(X_i = x_{ij} \mid Y = y_k) = \frac{n_{ijk} + mp}{n_k + m}$$

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

18

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

19

Continuous Attributes

- If X_i is a continuous feature rather than a discrete one, need another way to calculate P(X_i | Y).
- Assume that X_i has a Gaussian distribution whose mean and variance depends on Y.
- During training, for each combination of a continuous feature X_i and a class value for Y, y_k, estimate a mean, μ_{ik} , and standard deviation σ_{ik} based on the values of feature X_i in class y_k in the training data.
- During testing, estimate P(X_i | Y=y_k) for a given example, using the Gaussian distribution defined by μ_{ik} and σ_{ik} .

$$P(X_i \mid Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} \exp\left(\frac{-(X_i - \mu_{ik})^2}{2\sigma_{ik}^2} \right)$$

20

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.

21