CS 343: Artificial Intelligence Probabilistic Reasoning and Naïve Bayes

Raymond J. Mooney

University of Texas at Austin

1

Need for Probabilistic Reasoning

- Most everyday reasoning is based on uncertain evidence and inferences.
- Classical logic, which only allows conclusions to be strictly true or strictly false, does not account for this uncertainty or the need to weigh and combine conflicting evidence.
- Straightforward application of probability theory is impractical since the large number of probability parameters required are rarely, if ever, available.
- Therefore, early expert systems employed fairly *ad hoc* methods for reasoning under uncertainty and for combining evidence.
- Recently, methods more rigorously founded in probability theory that attempt to decrease the amount of conditional probabilities required have flourished.

2

Axioms of Probability Theory

- All probabilities between 0 and 1

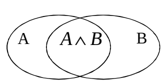
$$0 \leq P(A) \leq 1$$

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

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

- The probability of disjunction is:

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

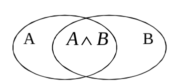


3

Conditional Probability

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

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



4

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:

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

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

5

Classification (Categorization)

- Given:
  - A description of an instance, $x \in X$, where X is the ***instance language*** or ***instance space***.
  - A fixed set of categories: $C=\{c_1, c_2, \ldots c_n\}$
- Determine:
  - The category of x: $c(x) \in C$, where $c(x)$ is a categorization function whose domain is X and whose range is C.
  - If $c(x)$ is a binary function $C=\{0,1\}$ ({true,false}, {positive, negative}) then it is called a ***concept***.

6

Learning for Categorization

- A training example is an instance $x \in X$, paired with its correct category $c(x)$: $<x, c(x)>$ for an unknown categorization function, c.
- Given a set of training examples, D.
- Find a hypothesized categorization function, $h(x)$, such that:

$$\forall < x, c(x) > \in D : h(x) = c(x)$$

Consistency

7

Sample Category Learning Problem

- Instance language: <size, color, shape>
  - size $\in$ {small, medium, large}
  - color $\in$ {red, blue, green}
  - shape $\in$ {square, circle, triangle}
- C = {positive, negative}
- D:

Example	Size	Color	Shape	Category
1	small	red	circle	positive
2	large	red	circle	positive
3	small	red	triangle	negative
4	large	blue	circle	negative

8

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.

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

9

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($Y=y_i \mid X=x_k$) for $i=1\ldots m$
- 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($Y,X_1,X_2\ldots X_n$)

10

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

11

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.

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

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

12

Bayesian Categorization (cont.)

- Need to know:
  - Priors: P(Y=y_i)
  - Conditionals: P(X=x_k | 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 | Y=y_i).
- Still need to make some sort of independence assumptions about the features to make learning tractable.

13

Generative Probabilistic Models

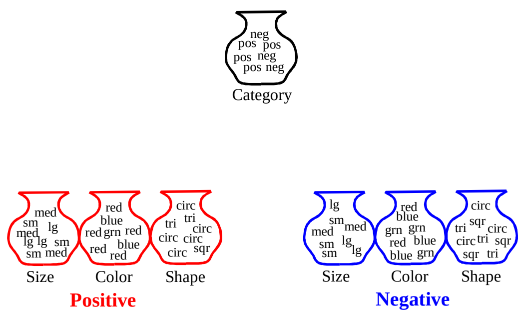
- Assume a simple (usually unrealistic) probabilistic method by which the data was generated.
- For categorization, each category has a different parameterized generative model that characterizes that category.
- **Training**: Use the data for each category to estimate the parameters of the generative 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 kth 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.

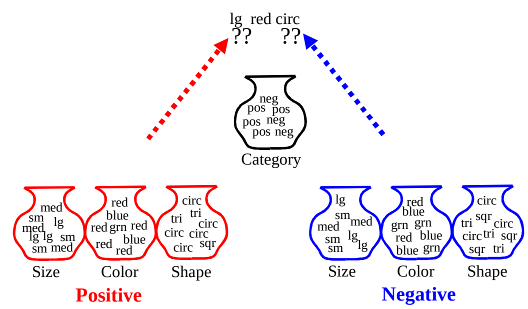
14

Naïve Bayes Generative Model



15

Naïve Bayes Inference Problem



16

Naïve Bayesian Categorization

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

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

- Therefore, we then only need to know P(X_i | 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=true | Y=true) and P(X_i=true | Y=false) for each X_i
  - P(X_i=false | Y) = 1 – P(X_i=true | Y)
- Compared to specifying 2^n parameters without any independence assumptions.

17

Naïve Bayes Categrization 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>

18

Naïve Bayes Categorization 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

19

Naïve Bayes Diagnosis Example

- C = {allergy, cold, well}
- e_1 = sneeze; e_2 = cough; e_3 = fever
- E = {sneeze, cough, ¬fever}

Prob	Well	Cold	Allergy
P(c_i)	0.9	0.05	0.05
P(sneeze|c_i)	0.1	0.9	0.9
P(cough|c_i)	0.1	0.8	0.7
P(fever|c_i)	0.01	0.7	0.4

20

Naïve Bayes Diagnosis Example (cont.)

Probability	Well	Cold	Allergy
P(c_i)	0.9	0.05	0.05
P(sneeze | c_i)	0.1	0.9	0.9
P(cough | c_i)	0.1	0.8	0.7
P(fever | c_i)	0.01	0.7	0.4

E={sneeze, cough, ¬fever}

P(well | E) = (0.9)(0.1)(0.1)(0.99)/P(E)=0.0089/P(E)
P(cold | E) = (0.05)(0.9)(0.8)(0.3)/P(E)=0.01/P(E)
P(allergy | E) = (0.05)(0.9)(0.7)(0.6)/P(E)=0.019/P(E)

Most probable category: allergy
P(E) = 0.0089 + 0.01 + 0.019 = 0.0379
P(well | E) = 0.23
P(cold | E) = 0.26
P(allergy | E) = 0.50

21

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=true | Y=y_k) = 0.
- If X_i=true then occurs in a test example, X, the result is that $\forall y_k$: P(X | Y=y_k) = 0 and $\forall y_k$: P(Y=y_k | X) = 0

22

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

23

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.

24

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

25

Text Categorization Applications

- Web pages
  - Recommending
  - Yahoo-like classification
- Newsgroup/Blog Messages
  - Recommending
  - spam filtering
  - Sentiment analysis for marketing
- News articles
  - Personalized newspaper
- Email messages
  - Routing
  - Prioritizing
  - Folderizing
  - spam filtering
  - Advertising on Gmail

26

Text Categorization Methods

- Most common representation of a document is a "bag of words," i.e. set of words with their frequencies, word order is ignored.
- Gives a high-dimensional vector representation (one feature for each word).
- Vectors are sparse since most words are rare.
  - Zipf's law and heavy-tailed distributions

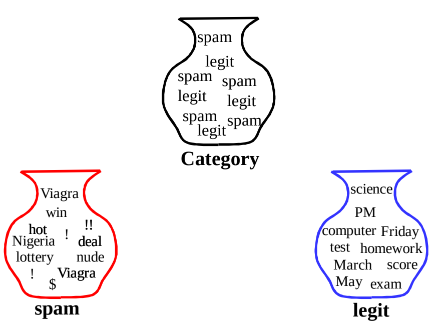
27

Naïve Bayes for Text

- Modeled as generating a bag of words for a document in a given category by repeatedly sampling with replacement from a vocabulary $V = \{w_1, w_2, \ldots w_m\}$ based on the probabilities $P(w_j \mid c_i)$.
- Smooth probability estimates with Laplace m-estimates assuming a uniform distribution over all words ($p = 1/|V|$) and $m = |V|$
  - Equivalent to a virtual sample of seeing each word in each category exactly once.

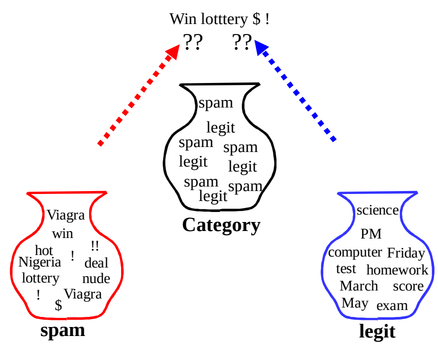
28

Naïve Bayes Generative Model for Text



29

Naïve Bayes Text Classification



30

Text Naïve Bayes Algorithm (Train)

Let V be the vocabulary of all words in the documents in D
For each category $c_i \in C$
  Let D_i be the subset of documents in D in category c_i
  $P(c_i) = |D_i| / |D|$
  Let T_i be the concatenation of all the documents in D_i
  Let n_i be the total number of word occurrences in T_i
  For each word $w_j \in V$
    Let n_{ij} be the number of occurrences of w_j in T_i
    Let $P(w_j \mid c_i) = (n_{ij} + 1) / (n_i + |V|)$

31

Text Naïve Bayes Algorithm (Test)

Given a test document X
Let n be the number of word occurrences in X
Return the category:

$$\underset{c_i \in C}{\operatorname{argmax}}\, P(c_i) \prod_{i=1}^{n} P(a_i \mid c_i)$$

where a_i is the word occurring the ith position in X

32

Underflow Prevention

- Multiplying lots of probabilities, which are between 0 and 1 by definition, can result in floating-point underflow.
- Since $\log(xy) = \log(x) + \log(y)$, it is better to perform all computations by summing logs of probabilities rather than multiplying probabilities.
- Class with highest final un-normalized log probability score is still the most probable.

33

Comments on Naïve Bayes

- Makes probabilistic inference tractable by making a strong assumption of conditional independence.
- Tends to work fairly well despite this strong assumption.
- Experiments show it to be quite competitive with other classification methods on standard datasets.
- Particularly popular for text categorization, e.g. spam filtering.

34