Naïve Bayes

LING 572

Fei Xia

Week 3: 1/19-1/21/2010

Outline

- Last week: kNN and DT
- Naïve Bayes in general
- Naïve Bayes for Text Classification

Questions

- Modeling:
  - Why is it called Naïve Bayes?
  - What objective function does it optimize?
  - How many types of model parameters?

- What happens at the training time?

- What happens at the test time?

Modeling

- Given $x=(f_1, \ldots, f_d)$, find
  $c^* = \arg\max_c P(c|x)$
  $= \arg\max_c P(c)\, P(x|c) / P(x)$ 🡐 Bayes' rule
  $= \arg\max_c P(c)\, P(x|c)$

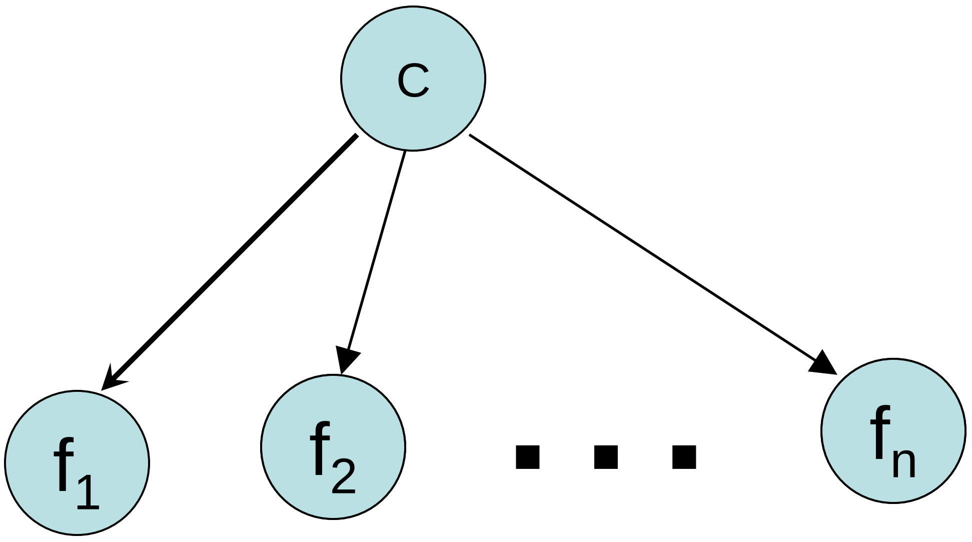
- Conditional independence assumption:

$$P(x \mid c) = P(f_1, f_2, \ldots, f_d \mid c)$$

$$= \prod_k P(f_k \mid c, f_1^{k-1})$$

$$\approx \prod_k P(f_k \mid c)$$ 🡐 "Naïve" assumption

Naïve Bayes Model



Assumption: each f_i is conditionally independent from f_j given C.

Model parameters

- Choose

  $c^* = \arg\max_c P(c) \prod_k P(f_k \mid c)$

- Two types of model parameters:
  - Class prior: $P(c)$
  - Conditional probability: $P(f_k \mid c)$

- The number of model parameters:

  $|C|+|CV|$

Training stage: estimating parameters θ

- Maximum likelihood (ML):

  $\theta^* = \arg\max_\theta P(\text{trainingData} \mid \theta)$

- $P(f_k \mid c_i) = cnt(f_k, c_i) / cnt(c_i)$

- $P(c_i) = cnt(c_i) / \sum_i cnt(c_i)$

Laplace Smoothing (add-one smoothing)

- Pretend you saw outcome one more than you actually did.

- Suppose X has K possible outcomes, and the counts for them are n_1, …, n_{K} , which sum to N.
  - Without smoothing: $P(X=i) = n_i / N$
  - With Laplace smoothing: $P(X=i) = (n_i + 1) / (N+K)$

Testing stage

- MAP (maximum a posteriori) decision rule:

  classify (x)
  $= \text{classify}(f_1, .., f_d)$
  $= \text{argmax}_c\, P(c|x)$
  $= \text{argmax}_c\, P(x|c)\, P(c)$
  $= \text{argmax}_c\, P(c) \prod_k P(f_k \mid c)$

Naïve Bayes for the text classification task

Features

- Features: bag of words (word order information is lost)

- Number of feature types: 1
- Number of features: $|V|$
- Features: w_t, $t \in \{1, 2, \ldots, |V|\}$

Issues

- Is w_t a binary feature?
- Are absent features used for calculating $P(d_i|c_j)$?

Two Naive Bayes Models (McCallum and Nigram, 1998)

- Multi-variate Bernoulli event model

  (a.k.a. binary independence model)
  - All features are binary: the number of times a feature occurs in an instance is ignored.
  - When calculating p(d | c), all features are used, including the absent features.

- Multinomial event model: “unigram LM”

Multi-variate Bernoulli event model

Bernoulli distribution

- Bernoulli trial: a statistical experiment having exactly two mutually exclusive outcomes each with a constant probability of occurrence:
  - Ex: toss a coin

- Bernoulli distribution: has exactly two mutually exclusive outcomes: $P(X=1)=p$ and $P(X=0)=1-p$.

Multi-variate Bernoulli Model

- A document is seen as a collection of |V| independent Bernoulli experiments, one for each word in the vocabulary: does this word appear in the document?

- Another way to look at this: (to be consistent with the general NB model)
  - Each word in the voc corresponds to two features:

    $w_k \text{ and } \bar{w}_k$

  - In any document, either $w_k \text{ or } \bar{w}_k$ is present; that is, it is always the case that exactly |V| features will be present in any document.

Training stage

ML estimate:

$$P(w_t|c_i) = \frac{Cnt(w_t, c_i)}{Cnt(c_i)}$$

$$P(c_i) = \frac{Cnt(c_i)}{\sum_i Cnt(c_i)}$$

With add-one smoothing:

$$P(w_t|c_i) = \frac{1 + Cnt(w_t, c_i)}{2 + Cnt(c_i)}$$

$$P(c_i) = \frac{1 + Cnt(c_i)}{|C| + \sum_i Cnt(c_i)}$$

Notation used in the paper

$$P(w_t|c_j) = \frac{1+Cnt(w_t,c_j)}{2+Cnt(c_j)}$$

Let B_{it} =1 if w_t appears in d_i
= 0 otherwise

$P(c_j \mid d_i)$ = 1 if d_i has the label c_j
= 0 otherwise

$$\hat{\theta}_{w_t|c_j} = \mathrm{P}(w_t|c_j;\theta) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} B_{it}\mathrm{P}(c_j|d_i)}{2 + \sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(c_j|d_i)}.$$

Testing stage

$$classify(d_i) = argmax_c P(c)P(d_i|c)$$

$$P(d_i|c)$$

$$= \prod_k P(f_k|c)$$

$$= \prod_{w_k \in d_i} P(w_k|c) \prod_{w_k \notin d_i} P(\bar{w_k}|c)$$

$$= \prod_{w_k \in d_i} P(w_k|c) \prod_{w_k \notin d_i} (1 - P(w_k|c))$$

Multinomial event model

Multinomial distribution

- Possible outcomes = $\{w_1, w_2, \ldots, w_{|v|}\}$
- A trial for each word position:
  $P(CurWord=w_i)=p_i$ and $\sum_i p_i = 1$
- Let X_i be the number of times that the word w_i is observed in the document.

$$P(X_1 = x_1, ..., X_v = x_v) = p_1^{x_1} ... p_v^{x_v} \frac{n!}{x_1!...x_v!}$$

$$= n! \prod_k \frac{p_k^{x_k}}{x_k!}$$

An example

- Suppose
  - the voc, V, contains only three words: a, b, and c.
  - a document, d_i, contains only 2 word tokens
  - For each position, P(w=a)=p1, P(w=b)=p2 and P(w=c)=p3.

- What is the prob that we see “a” once and “b” once in d_i?

An example (cont)

- 9 possible sequences: aa, ab, ac, ba, bb, bc, cc, cb, cc.
- The number of sequences with one “a” and one “b” (ab and ba): $n!/(x_1!...x_v!)$
- The prob of the sequence “ab” is p_1*p_2, so is the prob of the sequence “ba”.
- So the prob of seeing “a” once and “b” once is:

$$n! \prod_k (p_k^{x_k} / x_k!) = 2\ p_1*p_2$$

Multinomial event model

- A document is seen as a sequence of word events, drawn from the vocabulary V.
- N_{it}: the number of times that w_t appears in d_i
- Modeling: multinomial distribution:

$$P(d_i|c_j) = P(|d_i|)|d_i|! \prod_{t=1}^{|V|} \frac{P(w_t|c_j)^{N_{it}}}{N_{it}!}$$

Training stage for multinomial model

Let $P(c_j \mid d_i) = 1$ if d_i has the label c_j
$= 0$ otherwise

$$P(w_t|c_j) = \frac{1+\sum_{i=1}^{|D|} N_{it} P(c_j|d_i)}{|V|+\sum_{s=1}^{|V|}\sum_{i=1}^{|D|} N_{is} P(c_j|d_i)}$$

Compared with the following in the Bernoulli model:

$$\hat{\theta}_{w_t|c_j} = \mathrm{P}(w_t|c_j;\theta) = \frac{1+\sum_{i=1}^{|\mathcal{D}|} B_{it}\mathrm{P}(c_j|d_i)}{2+\sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(c_j|d_i)}.$$

Testing stage

$$classify(d_i) = argmax_c P(c)P(d_i|c)$$

$$P(d_i|c) = P(|d_i|)|d_i|! \prod_{k=1}^{|V|} \frac{P(w_k|c)^{N_{ik}}}{N_{ik}!}$$

$$classify(d_i) = argmax_c P(c) \prod_{k=1}^{|V|} P(w_k|c)^{N_{ik}}$$

Two models

- Multi-variate Bernoulli event model: treat features as binary; each trial corresponds to a word in the voc.
- Multinomial event model: treat features as non-binary; each trial corresponds to a word position in the document.
- Multinomial event model usually beats the Bernoulli event model (McCallum and Nigram, 1998)

Two models (cont)

	Multi-variate Bernoulli	Multinomial
Features	Binary: present or absent	Real-valued: the occurrence
Each trial	Each word in the voc	Each word position in the doc
$P(c_i)$	$\frac{1+Cnt(c_i)}{\lvert C\rvert+\sum_i Cnt(c_i)}$	$\frac{1+Cnt(c_i)}{\lvert C\rvert+\sum_i Cnt(c_i)}$
$P(w_t \mid c_j)$	$\frac{1+Cnt(w_t,c_j)}{2+Cnt(c_j)}$	$\frac{1+\sum_{i=1}^{\lvert D\rvert} N_{it}P(c_j \mid d_i)}{\lvert V\rvert+\sum_{s=1}^{\lvert V\rvert}\sum_{i=1}^{\lvert D\rvert} N_{is}P(c_j \mid d_i)}$
$classify(d_i)$	$P(c)\prod_{w_k\in d_i} P(w_k \mid c)$ $\prod_{w_k\notin d_i}(1-P(w_k \mid c))$	$P(c)\prod_{k=1}^{\lvert V\rvert} P(w_k \mid c)^{N_{ik}}$

Summary of Naïve Bayes

- It makes a strong independence assumption: all the features are conditionally independent given the class.

- It generally works well despite the strong assumption. Why?

- Both training and testing are simple and fast.

Summary of Naïve Bayes (cont)

- Strengths:
  - Simplicity (conceptual)
  - Efficiency at training
  - Efficiency at testing time
  - Handling multi-class
  - Scalability
  - Output topN

- Weakness:
  - Theoretical validity: the independency assumption
  - Predication accuracy: not as good as MaxEnt etc.

Hw3

Hw3

- Q1: run the NB learner in Mallet
- Q2-Q3: build a Multi-variate Bernoulli NB learner
- Q4: build a Multinomial NB learner
- Q5: get the results with binary features
- Q6: Conclusions from the experiments

Q2

- build_NB1.sh training_data test_data prior_delta cond_prob_delta model_file sys_output > acc

- prior_delta: delta for calculating P(c).

- cond_prob_delta: delta for calculating P(f|c).

Model file

c1 P(c1) log P(c1) ## log is all 10-based

….

f1 c1 P(f1|c1) log P(f1|c1)

f2 c1 P(f2|c1) log P(f2|c1)

…

f1 c2 P(f1|c2) log P(f1|c2)

f2 c2 P(f2|c2) log P(f2|c2)

…

Sys_output

instanceName trueClass c_1 p_1 c_2 p_2 …

(c_i, p_i) should be sorted by the value of p_i.

$$p_i = P(c_i|x) = \frac{P(x|c_i)P(c_i)}{P(x)}$$

$$P(x) = \sum_i P(c_i, x) = \sum_i P(x|c_i)P(c_i)$$

The issue of underflow

$$p_i = P(c_i|x) = \frac{P(x|c_i)P(c_i)}{P(x)}$$

$logP(x, c_1)$ is -200, $logP(x, c_2)$ is -201.

$logP(x, c_3)$ is -202.

What is p_i?

$$p_1 = \frac{1}{1+10^{-1}+10^{-2}} = 100/111 = 0.901$$

$$p_2 = \frac{10^{-1}}{1+10^{-1}+10^{-2}} = 10/111 = 0.09$$

$$p_3 = \frac{10^{-2}}{1+10^{-1}+10^{-2}} = 1/111 = 0.009$$

Efficiency issue: Ex 1

Log $P(c) \prod_{k=1}^{|V|} P(w_k|c)^{N_{ik}}$

$$= logP(c) + \sum_{k=1}^{|V|} log(P(w_k|c)^{N_{ik}}$$

$$= logP(c) + \sum_{k=1}^{|V|} N_{ik}\ log\ P(w_k|c)$$

Efficiency: Ex #2

$$P(d_i, c)$$

$$= P(c)(\prod_{w_k \in d_i} P(w_k|c))(\prod_{w_k \notin d_i}(1 - P(w_k|c)))$$

$$= P(c) \prod_{w_k \in d_i} \frac{P(w_k|c)}{1 - P(w_k|c)} \prod_{w_k}(1 - P(w_k|c))$$

Efficiency: Ex #3

Multinomial model:

Let $P(c_j \mid d_i) = 1$ if d_i has the label c_j
$= 0$ otherwise

$$P(w_t|c_j) = \frac{1+\sum_{i=1}^{|D|} N_{it} P(c_j|d_i)}{|V|+\sum_{s=1}^{|V|}\sum_{i=1}^{|D|} N_{is} P(c_j|d_i)}$$

$$\text{Complexity: } O(|V|^2 * |C| * |D|)$$

$Z(c_j) = 0$ for every c_j;

for each d_i
  Let c_j be the class label of d_i
  for each w_t that is present in d_i
    Let N_{it} be the number of times w_t appears in d_i
    $cnt(w_t, c_j) \ += N_{it}$
    $Z(c_j) \ += N_{it}$

for each c_j
  for each w_t
    $P(w_t|c_j) = \frac{1+cnt(w_t,c_j)}{|V|+Z(c_j)}$

Complexity: $O(|V| * |C| + |D| * avg(feat/doc))$