Naïve Bayes

Advanced Statistical Methods in NLP
Ling 572
January 17, 2012

Roadmap

- Motivation:
  - Example: Spam Tagging
- Naïve Bayes Model
  - Bayesian structure
  - Naïve Assumption
  - Training & Decoding
  - Variants
  - Issues

Bayesian Spam Filtering

- Task: Given an email message, classify it as ‘spam’ or ‘not spam’
  - Form of automatic text categorization

- Features?

Doc1

Western Union Money Transfer office29@yahoo.com.ph
One Bishops Square Akpakpa E1 6AO, Cotonou
Benin Republic
Website: http://www.westernunion.com/ info/selectCountry.asP
Phone: +229 99388639

Attention Beneficiary,

This to inform you that the federal ministry of finance Benin Republic has started releasing scam victim compensation fund mandated by United Nation Organization through our office.

I am contacting you because our agent have sent you the first payment of $5,000 for your compensation funds total amount of $500 000 USD (Five hundred thousand united state dollar)

We need your urgent response so that we shall release your payment information to you.

You can call our office hot line for urgent attention(+22999388639)

Doc2

- Hello! my dear. How are you today and your family? I hope all is good,
  kindly pay Attention and understand my aim of communicating you today
  through this Letter, My names is Saif al-Islam al-Gaddafi the Son of
  former Libyan President. i was born on 1972 in Tripoli Libya,By Gaddafi's
  second wive.
  I want you to help me clear this fund in your name which i deposited in
  Europe please i would like this money to be transferred into your account
  before they find it.
  the amount is 20.300,000 million GBP British Pounds sterling through a

Doc4

- from: acl@aclweb.org
- REMINDER:

  If you have not received a PIN number to vote in the elections and have not already contacted us, please contact either Drago Radev (radev@umich.edu) or Priscilla Rasmussen (acl@aclweb.org) right away.

  Everyone who has not received a pin but who has contacted us already will get a new pin over the weekend.

  Anyone who still wants to join for 2011 needs to do this by Monday (November 7th) in order to be eligible to vote.

  And, if you do have your PIN number and have not voted yet, remember every vote counts!

What are good features?

Possible Features

- Words!

Possible Features

- Words!
  - Feature for each word

Possible Features

- Words!
  - Feature for each word
    - Binary: presence/absence
    - Integer: occurrence count
    - Particular word types: money/sex/: [Vv].*gr.*

Possible Features

- Words!
  - Feature for each word
    - Binary: presence/absence
    - Integer: occurrence count
    - Particular word types: money/sex/: [Vv].*gr.*
- Errors:
  - Spelling, grammar

Possible Features

- Words!
  - Feature for each word
    - Binary: presence/absence
    - Integer: occurrence count
    - Particular word types: money/sex/: [Vv].*gr.*
- Errors:
  - Spelling, grammar
- Images

Possible Features

- Words!
  - Feature for each word
    - Binary: presence/absence
    - Integer: occurrence count
    - Particular word types: money/sex/: [Vv].*gr.*
- Errors:
  - Spelling, grammar
- Images
- Header info

Classification w/Features

- Probabilistic framework:
  - Higher probability of spam given features than !spam
  - P(spam|f) > P(!spam|f)

Classification w/Features

- Probabilistic framework:
  - Higher probability of spam given features than !spam
  - P(spam|f) > P(!spam|f)
- Combining features?

Classification w/Features

- Probabilistic framework:
  - Higher probability of spam given features than !spam
  - P(spam|f) > P(!spam|f)
- Combining features?
  - Could use existing models
    - Decision trees, nearest neighbor

Classification w/Features

- Probabilistic framework:
  - Higher probability of spam given features than !spam
  - P(spam|f) > P(!spam|f)
- Combining features?
  - Could use existing models
    - Decision trees, nearest neighbor
  - Alternative:
    - Consider association of each feature with spam/!spam
    - Combine

ML Questions: Reminder

- Modeling:
  - What is the model structure?
    - Why is it called ‘Naïve Bayes’?
    - What assumptions does it make?

ML Questions: Reminder

- Modeling:
  - What is the model structure?
    - Why is it called ‘Naïve Bayes’?
  - What assumptions does it make?
  - What types of parameters are learned?
    - How many parameters?

ML Questions: Reminder

- Modeling:
  - What is the model structure?
    - Why is it called ‘Naïve Bayes’?
  - What assumptions does it make?
  - What types of parameters are learned?
    - How many parameters?
- Training:
  - How are model parameters learned from data?

ML Questions: Reminder

- Modeling:
  - What is the model structure?
    - Why is it called 'Naïve Bayes'?
  - What assumptions does it make?
  - What types of parameters are learned?
    - How many parameters?
- Training:
  - How are model parameters learned from data?
- Decoding:
  - How is model used to classify new data?

Probabilistic Model

- Given an instance x with features $f_1 \ldots f_{k,}$
  - Find the class with highest probability

Probabilistic Model

- Given an instance x with features $f_1 \ldots f_{k,}$
  - Find the class with highest probability
  - Formally, $x = (f_1, f_2, \ldots, f_k)$
    - Find $c^* = \text{argmax}_c \, P(c|x)$

Probabilistic Model

- Given an instance x with features $f_1 \ldots f_{k,}$
  - Find the class with highest probability
  - Formally, $x = (f_1, f_2, \ldots, f_k)$
    - Find $c^* = \text{argmax}_c \, P(c|x)$
  - Applying Bayes' Rule:
    - $c^* = \text{argmax}_c \, P(x|c)P(c)/P(x)$

Probabilistic Model

- Given an instance x with features $f_1 \ldots f_{k,}$
  - Find the class with highest probability
  - Formally, $x = (f_1, f_2, \ldots, f_k)$
    - Find $c^* = \text{argmax}_c \, P(c|x)$
  - Applying Bayes' Rule:
    - $c^* = \text{argmax}_c \, P(x|c)P(c)/P(x)$
  - Maximizing:
    - $c^* = \text{argmax}_c \, P(x|c)P(c)$

Naïve Bayes Model

- So far just Bayes' Rule
- Key question: How do we handle/combine features?

Naïve Bayes Model

- So far just Bayes' Rule
- Key question: How do we handle/combine features?
- Consider just P(x|c)
  - P(x|c)

Naïve Bayes Model

- So far just Bayes' Rule
- Key question: How do we handle/combine features?
- Consider just P(x|c)
  - $P(x|c) = P(f_1,f_2,\ldots,f_k|c)$

Naïve Bayes Model

- So far just Bayes' Rule
- Key question: How do we handle/combine features?
- Consider just P(x|c)
  - $P(x|c) = P(f_1, f_2, \ldots, f_k|c)$

$$= \prod_j P(f_j | c, f_1^{j-1})$$

  - Can we simplify?

Naïve Bayes Model

- So far just Bayes' Rule
- Key question: How do we handle/combine features?
- Consider just P(x|c)
  - P(x|c) = P(f_1,f_2,...,f_k|c)

$$= \prod_j P(f_j \mid c, f_1^{j-1})$$

  - Can we simplify? (Remember ngrams)

Naïve Bayes Model

- So far just Bayes' Rule
- Key question: How do we handle/combine features?
- Consider just P(x|c)
  - $P(x|c) = P(f_1, f_2, \ldots, f_k|c)$

$$= \prod_j P(f_j | c, f_1^{j-1})$$

  - Can we simplify? (Remember ngrams)
    - Assume conditional independence

Naïve Bayes Model

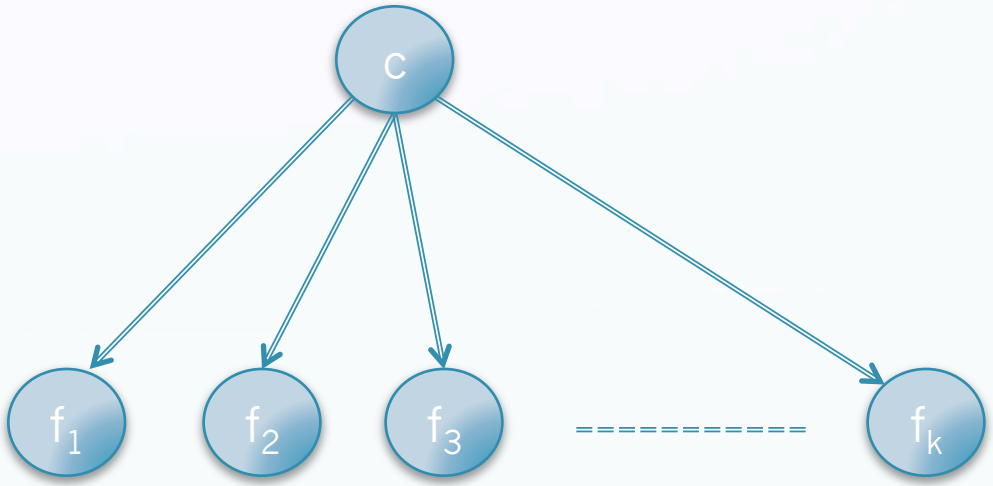
- So far just Bayes' Rule
- Key question: How do we handle/combine features?
- Consider just P(x|c)
  - $P(x|c) = P(f_1, f_2, \ldots, f_k|c)$

$$= \prod_j P(f_j | c, f_1^{j-1})$$

  - Can we simplify? (Remember ngrams)
  - Assume conditional independence

$$= \prod_j P(f_j | c)$$

Naïve Bayes Model



- Naïve Bayes Model assumes features f are independent of each other, given the class C

Model Parameters

- Our model:
  - $c^* = \text{argmax}_c \, P(c) \Pi_j P(f_j | c)$

Model Parameters

- Our model:
  - $c^* = \text{argmax}_c \, P(c) \Pi_j P(f_j | c)$

- How many types of parameters?

Model Parameters

- Our model:
  - $c^* = \text{argmax}_c \, P(c) \Pi_j P(f_j | c)$

- How many types of parameters?
  - Two:

Model Parameters

- Our model:
  - $c^* = \text{argmax}_c \, P(c) \Pi_j P(f_j | c)$

- How many types of parameters?
  - Two:
    - P(C): Class priors
    - $P(f_j | c)$: Class conditional feature probabilities

Model Parameters

- Our model:
  - $c^* = \text{argmax}_c \, P(c) \Pi_j P(f_j | c)$
- How many types of parameters?
  - Two:
    - P(C): Class priors
    - $P(f_j | c)$: Class conditional feature probabilities
- How many features in total?

Model Parameters

- Our model:
  - $c^* = \text{argmax}_c \, P(c) \Pi_j P(f_j | c)$
- How many types of parameters?
  - Two:
    - P(C): Class priors
    - $P(f_j | c)$: Class conditional feature probabilities
- How many features in total?
  - |priors|+|conditional probabilities|

Model Parameters

- Our model:
  - $c^* = \text{argmax}_c \, P(c) \Pi_j P(f_j | c)$
- How many types of parameters?
  - Two:
    - P(C): Class priors
    - $P(f_j | c)$: Class conditional feature probabilities
- How many features in total?
  - |priors|+|conditional probabilities|
  - |C| + |FC|
  - |C|+|VC|, if features are words in vocabulary V

Training

- Parameter estimation from labeled training data
  - Maximum likelihood estimation

Training

- Parameter estimation from labeled training data
  - Maximum likelihood estimation

- Class priors:

Training

- Parameter estimation from labeled training data
  - Maximum likelihood estimation

- Class priors:

$$P(c_i) = \frac{Count(c_i)}{\sum_j Count(c_j)}$$

Training

- Parameter estimation from labeled training data
  - Maximum likelihood estimation

- Class priors: $$P(c_i) = \frac{Count(c_i)}{\sum_j Count(c_j)}$$

- Conditional probabilities

Training

- Parameter estimation from labeled training data
  - Maximum likelihood estimation

- Class priors: $$P(c_i) = \frac{Count(c_i)}{\sum_j Count(c_j)}$$

- Conditional probabilities

$$P(f_j \mid c_i) = \frac{Count(f_j, c_i)}{Count(c_i)}$$

Training

- MLE issues?

Training

- MLE issues?
  - What's the probability of a feature not seen with a c_i?

Training

- MLE issues?
  - What's the probability of a feature not seen with a c_i?
    - 0
    - What happens then?

Training

- MLE issues?
  - What's the probability of a feature not seen with a c_i?
    - 0
    - What happens then?
- Solutions?

Training

- MLE issues?
  - What's the probability of a feature not seen with a c_i?
    - 0
    - What happens then?

- Solutions?
  - Smoothing
    - Laplace smoothing, Good-Turing, Witten-Bell
    - Interpolation, Backoff....

What are Zero Counts?

- Some of those zeros are really zeros…
  - Things that really can't or shouldn't happen.

What are Zero Counts?

- Some of those zeros are really zeros...
  - Things that really can't or shouldn't happen.
- On the other hand, some of them are just rare events.
  - If the training corpus had been a little bigger they would have had a count (probably a count of 1!).

What are Zero Counts?

- Some of those zeros are really zeros...
  - Things that really can't or shouldn't happen.
- On the other hand, some of them are just rare events.
  - If the training corpus had been a little bigger they would have had a count (probably a count of 1!).
- Zipf's Law (long tail phenomenon):
  - A small number of events occur with high frequency
  - A large number of events occur with low frequency
  - You can quickly collect statistics on the high frequency events
  - You might have to wait an arbitrarily long time to get valid statistics on low frequency events

Laplace Smoothing

- Add 1 to all counts (aka Add-one Smoothing)

Laplace Smoothing

- Add 1 to all counts (aka Add-one Smoothing)
- V: size of vocabulary; N: size of corpus
- Unigram: P_{MLE}

Laplace Smoothing

- Add 1 to all counts (aka Add-one Smoothing)
- V: size of vocabulary; N: size of corpus
- Unigram: P_{MLE}: $P(w_i) = C(w_i)/N$
- $P_{Laplace}(w_i) =$

Laplace Smoothing

- Add 1 to all counts (aka Add-one Smoothing)
- V: size of vocabulary; N: size of corpus
- Unigram: P_{MLE}: $P(w_i) = C(w_i)/N$
- $$P_{Laplace}(w_i) = \frac{C(w_i)+1}{N+V}$$

Decoding

- Maximum a posteriori (MAP) decision rule

Decoding

- Maximum a posteriori (MAP) decision rule

- Given our model, and an instance $x=\{f_1,f_2,....,f_k\}$
- Classify(x)=
- $= \text{Classify}(f_1,f_2,...,f_k)$
- =

Decoding

- Maximum a posteriori (MAP) decision rule

- Given our model, and an instance $x=\{f_1,f_2,....,f_k\}$
- Classify(x)=
- $= \text{Classify}(f_1,f_2,...,f_k)$
- $= \text{argmax}_c\ P(c|x)$
- =

Decoding

- Maximum a posteriori (MAP) decision rule

- Given our model, and an instance $x=\{f_1,f_2,....,f_k\}$
- Classify(x)=
- $= \text{Classify}(f_1,f_2,...,f_k)$
- $= \text{argmax}_c\ P(c|x)$
- $= \text{argmax}_c\ P(x|c)P(c)$
- $= \text{argmax}_c\ \Pi_j P(f_j|c)P(c)$

Naïve Bayes Spam Tagging

- Pantel & Lin
- Represent message x as set of features
  - Features: $f_1, f_2, ..., f_n$

Naïve Bayes Spam Tagging

- Pantel & Lin
- Represent message x as set of features
  - Features: $f_1, f_2, ..., f_n$
- Features:
  - Words
    - Alternatively (skip) n-gram sequences
    - Stemmed (?)

Naïve Bayes Spam Tagging II

- Estimating term conditional probabilities
  - Add-delta smoothing

Naïve Bayes Spam Tagging II

- Estimating term conditional probabilities
  - Add-delta smoothing
- Selecting good features:
  - Exclude terms s.t.

Naïve Bayes Spam Tagging II

- Estimating term conditional probabilities
  - Add-delta smoothing

- Selecting good features:
  - Exclude terms s.t.
    - N(W|Spam)+N(W|NotSpam)<4
    - 0.45 <=P(W|Spam)/(P(W|Spam)+P(W|NotSpam))<=0.55

Experimentation (Pantel & Lin)

- Training: 160 spam, 466 non-spam
- Test: 277 spam, 346 non-spam

- 230,449 training words; 60434 spam
  - 12228 terms; filtering reduces to 3848

Results (PL)

- False positives: 1.16%
- False negatives: 8.3%
- Overall error: 4.33%

- Simple approach, effective

Naïve Bayes for Text Classification

Naïve Bayes and Text Classification

- Naïve Bayes model applied for many TC applications

Naïve Bayes and Text Classification

- Naïve Bayes model applied for many TC applications
  - Spam tagging
    - Many widespread systems (e.g., SpamAssassin)

Naïve Bayes and Text Classification

- Naïve Bayes model applied for many TC applications
  - Spam tagging
    - Many widespread systems (e.g., SpamAssassin)
  - Authorship classification

Naïve Bayes and Text Classification

- Naïve Bayes model applied for many TC applications
  - Spam tagging
    - Many widespread systems (e.g., SpamAssassin)
  - Authorship classification
  - General document categorization
    - e.g. McCallum & Nigam paper
      - WebKB, Reuters, 20 newsgroups,....

Features

- General text classification features:
  - Words
  - Referred to as ‘bag of words’ models
    - No word order information

Features

- General text classification features:
  - Words
  - Referred to as ‘bag of words’ models
    - No word order information

- # features:

Features

- General text classification features:
  - Words
  - Referred to as ‘bag of words’ models
    - No word order information

- # features: $|V|$
- Features: w_t, t in $\{1,..,|V|\}$

Questions

- How do we model word features?

- Should we model absence of features directly?
  - In addition to presence

Naïve Bayes Models in Detail

- (McCallum & Nigam, 1998)
- Alternate models for Naïve Bayes Text Classification

Naïve Bayes Models in Detail

- (McCallum & Nigam, 1998)
- Alternate models for Naïve Bayes Text Classification
  - Multivariate Bernoulli event model
    - Binary independence model
      - Features treated as binary – counts ignored

Naïve Bayes Models in Detail

- (McCallum & Nigam, 1998)
- Alternate models for Naïve Bayes Text Classification

  - Multivariate Bernoulli event model
    - Binary independence model
      - Features treated as binary – counts ignored
  - Multinomial event model
    - Unigram language model

Multivariate Bernoulli Event Model

- Bernoulli distribution:
  - Bernoulli trial:
    - statistical experiment with:
      - exactly two mutually exclusive outcomes
      - with constant probability of occurrence

Multivariate Bernoulli Event Model

- Bernoulli distribution:
  - Bernoulli trial:
    - statistical experiment with:
      - exactly two mutually exclusive outcomes
      - with constant probability of occurrence
    - Typical example is a coin flip
      - Heads, tails
      - P(X=heads)= p: P(X=tails)=$1-p$

Multivariate Bernoulli Event Text Model

- Each document:
  - Result of |V| independent Bernoulli trials
  - I.e. for each word in vocabulary,
    - does the word appear in the document?

$\bar{w}_t$

$\bar{w}_t$

Multivariate Bernoulli Event Text Model

- Each document:
  - Result of |V| independent Bernoulli trials
  - I.e. for each word in vocabulary,
    - does the word appear in the document?

- From general Naïve Bayes perspective
  - Each word corresponds to two variables, w_t and $\bar{w}_t$
  - In each doc, either w_t or $\bar{w}_t$ appears
    - Always have |V| elements in a document

Training

- MLE:

$$P(w_t \mid c_j) = \frac{Count(w_t, c_j)}{Count(c_j)}$$

$$P(c_j) = \frac{Count(c_j)}{\sum_k Count(c_k)}$$

- Laplace smoothed:

Training

- MLE:

$$P(w_t \mid c_j) = \frac{Count(w_t, c_j)}{Count(c_j)}$$

$$P(c_j) = \frac{Count(c_j)}{\sum_k Count(c_k)}$$

- Laplace smoothed:

$$P(w_t \mid c_j) = \frac{1 + Count(w_t, c_j)}{2 + Count(c_j)}$$

Training

- MLE:

$$P(w_t \mid c_j) = \frac{Count(w_t, c_j)}{Count(c_j)}$$

$$P(c_j) = \frac{Count(c_j)}{\sum_k Count(c_k)}$$

- Laplace smoothed:

$$P(w_t \mid c_j) = \frac{1 + Count(w_t, c_j)}{2 + Count(c_j)}$$

$$P(c_j) = \frac{1 + Count(c_j)}{|C| + \sum_k Count(c_k)}$$

Notation

$$P(w_t \mid c_j) = \frac{1 + Count(w_t, c_j)}{2 + Count(c_j)}$$

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

$P(c_j \mid d_i)$ = 1 if document d_i is of class c_j
= 0 otherwise

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

Testing

- MAP decision rule:

- classify(d) = $\text{argmax}_c P(c)P(d|c)$

Testing

- MAP decision rule:

- classify(d) = $\text{argmax}_c\, P(c)P(d|c)$
- = $P(c)\ \Pi_k P(f_k|c)$

Testing

- MAP decision rule:

- classify(d) = argmax_c P(c)P(d|c)
- = P(c) Π_kP(f_k|c)
- = P(c) $\left(\prod_{w_k \in d} P(w_k \mid c) \prod_{w_k \notin d} P(\bar{w}_k \mid c)\right)$

Testing

- MAP decision rule:

- classify(d) = argmax_c P(c)P(d|c)
- = P(c) Π_kP(f_k|c)
- = P(c) $\left(\prod_{w_k \in d} P(w_k \mid c) \prod_{w_k \notin d} P(\bar{w}_k \mid c)\right)$
- = P(c)$\left(\prod_{w_k \in d} P(w_k \mid c) \prod_{w_k \notin d} (1 - P(\bar{w}_k \mid c))\right)$