Machine Learning

Lecture 17c: Naïve Bayes Classifier for Text Classification

Dr. Beilun Wang

Southeast University
School of Computer Science
and Engineering

$$\underset{k}{\operatorname{argmax}} P(C\_k \mid X) = \underset{k}{\operatorname{argmax}} P(X, C) = \underset{k}{\operatorname{argmax}} P(X \mid C) P(C)$$

Generative Bayes Classifier

Task → Representation → Score Function → Search/Optimization → Models, Parameters

classification → Prob. models p(X|C) → EPE with 0-1 loss → MAP Rule → Many options → Prob. Models' Parameter

$$P(X_1, \cdots, X_p \mid C)$$

Gaussian Naive

$$\hat{P}(X_j \mid C = c_k) = \frac{1}{\sqrt{2\pi}\sigma_{jk}} \exp\left(-\frac{(X_j - \mu_{jk})^2}{2\sigma_{jk}^2}\right)$$

Bernoulli Naïve

$$p(W_i = true \mid c_k) = p_{i,k}$$

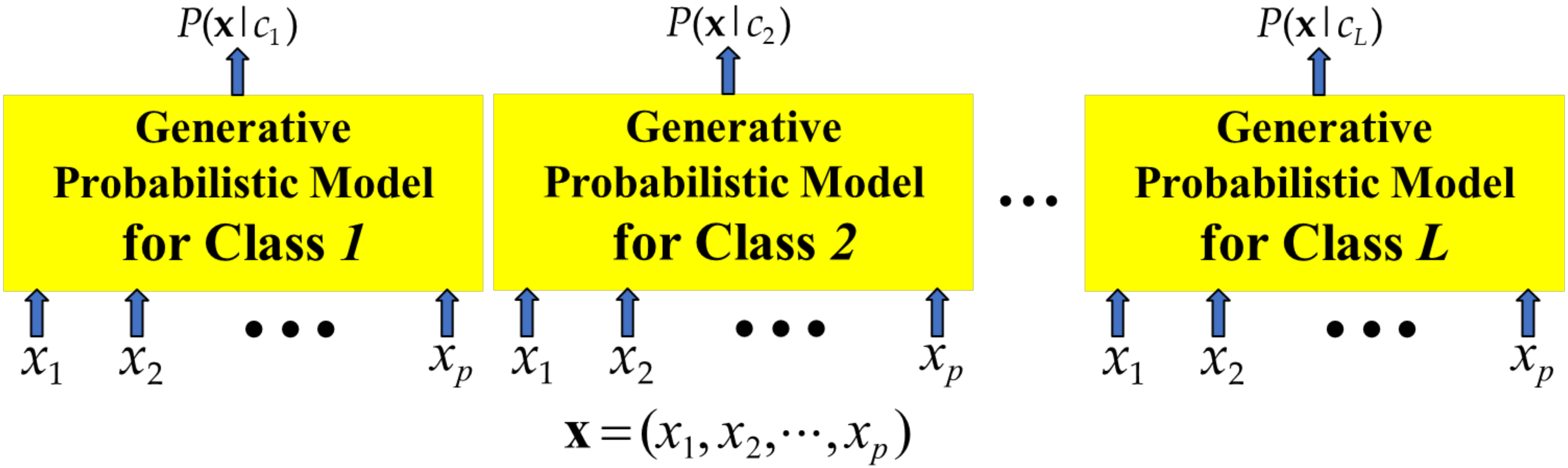
Multinomial

$$P(W_1 = n_1, \ldots, W_v = n_v \mid c_k) = \frac{N!}{n_{1k}! n_{2k}! .. n_{vk}!} \theta_{1k}^{n_{1k}} \theta_{2k}^{n_{2k}} .. \theta_{vk}^{n_{vk}}$$

Review: Generative BC

$$P(\mathbf{X}|C),$$

$$C = c_1, \cdots, c_L, \mathbf{X} = (X_1, \cdots, X_p)$$



Review: Naïve Bayes Classifier

$$\underset{C}{\operatorname{argmax}} P(C \mid X) = \underset{C}{\operatorname{argmax}} P(X, C) = \underset{C}{\operatorname{argmax}} P(X \mid C) P(C)$$

Naïve Bayes Classifier

$$P(X_1, X_2, \cdots, X_p | C) = P(X_1|C) P(X_2|C) \cdots P(X_p|C)$$

$$c^* = argmax \ \ P(C = c_i | \mathbf{X} = \mathbf{x}) \propto P(\mathbf{X} = \mathbf{x} | C = c_i) P(C = c_i)$$

$$for\ i = 1, 2, \cdots, L$$

Assuming all input attributes are conditionally independent given a specific class label!

Today: Naïve Bayes Classifier for Text

- Dictionary based Vector space representation of text article
- Multivariate Bernoulli vs. Multinomial
- Multivariate Bernoulli naïve Bayes classifier
  - Testing
  - Training With Maximum Likelihood Estimation for estimating parameters
- Multinomial naïve Bayes classifier
  - Testing
  - Training With Maximum Likelihood Estimation for estimating parameters

Text classification Tasks

- Input: document D
- Output: the predicted class C,   c is from $\{c_1,\ldots,c_L\}$

- Text classification examples:
- Classify email as 'Spam ' , ' Other ' .

From: "" <takworlld@hotmail.com>
Subject: real estate is the only way... gem oalvgkay

Anyone can buy real estate with no money down
Stop paying rent TODAY !

Change your life NOW by taking a simple course!
Click Below to order:
http://www.wholesaledaily.com/sales/nmd.htm

→ P (C=spam | D)

Text classification Tasks

- Input: document D
- Output: the predicted class C, c is from $\{c_1,...,c_L\}$

- Text classification examples:
- Classify email as 'Spam' , 'Other ' .
- Classify web pages as 'Student' , 'Faculty' , 'Other '
- Classify news stories into topics 'Sports' , 'Politics' ..
- Classify movie reviews as 'Favorable' , 'Unfavorable' , ' Neutral'
- ... and many more.

Text Categorization/Classification

- Given:
  - A representation of a text document d
    - Issue: how to represent text documents.
    - Usually some type of high-dimensional space – bag of words

- A fixed set of output classes:
  - $C = \{c_1, c_2, \ldots, c_J\}$

The bag of words representation

f(I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun… It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet.)=c

The bag of words representation

$$f\left(\begin{array}{|l|l|}\hline \texttt{great} & 2 \\ \hline \texttt{love} & 2 \\ \hline \texttt{recommend} & 1 \\ \hline \texttt{laugh} & 1 \\ \hline \texttt{happy} & 1 \\ \hline \dots & \dots \\ \hline \end{array}\right)=c$$

Representing text: a list of words → Dictionary

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun… It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet.

word	*frequency*
great	2
love	2
recommend	1
laugh	1
happy	1
...	.

Common refinements: remove stopwords, stemming, collapsing multiple occurrences of words into one….

'Bag of words' representation of text

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun… It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet.

word	*frequency*
great	2
love	2
recommend	1
laugh	1
happy	1
...	.

Bag of word representation: Represent text as a vector of word frequencies.

Another 'Bag of words' representation of text → Each dictionary word as Boolean

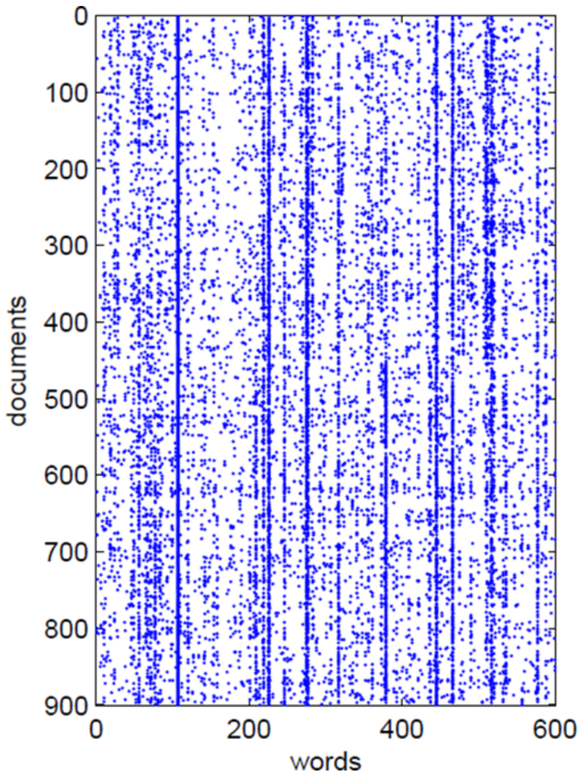
I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun… It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet.

word	*Boolean*
great	Yes
love	Yes
recommend	Yes
laugh	Yes
happy	Yes
hate	No
...	.

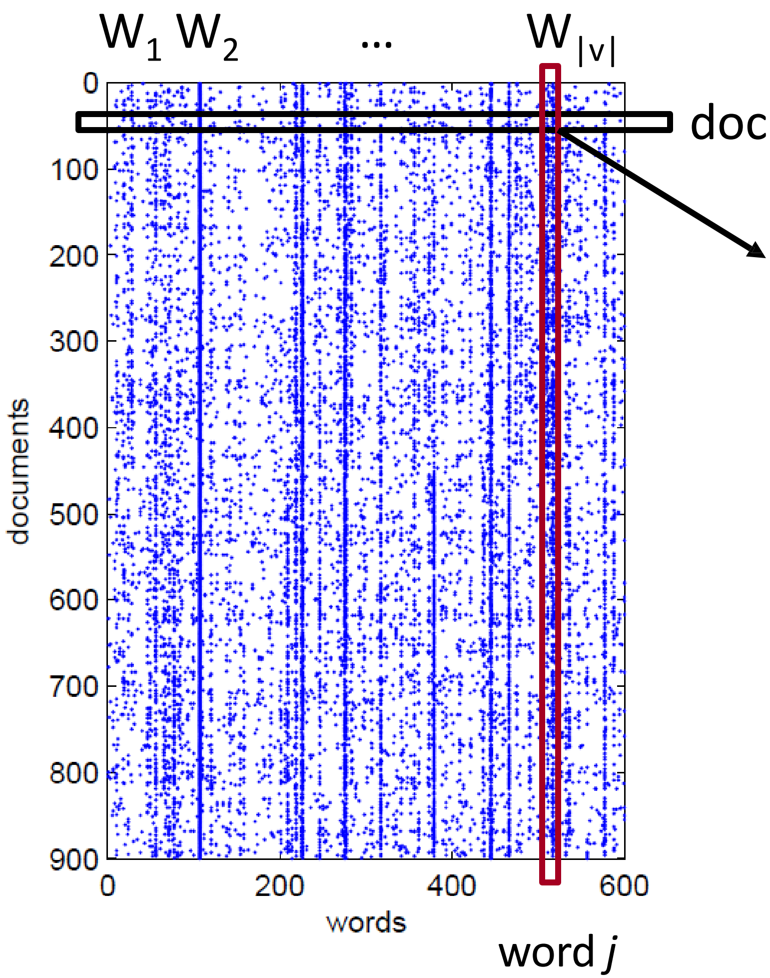
Bag of word representation: Represent text as a vector of Boolean representing if a word *Exists or NOT*.

Bag of words

- What simplifying assumption are we taking?
  - We assume word order is not important.



Bag of words representation



e.g., X (i,j) = Frequency of word j in document *i*

A collection of documents

	X_1	X_2	X_3	C
s_1				
s_2				
s_3				
s_4				
s_5				
s_6				

Unknown Words

- How to handle words in the test corpus that did not occur in the training data, i.e. ***out of vocabulary*** (OOV) words?
- Train a model that includes an explicit symbol for an unknown word (<UNK>).
  - Choose a vocabulary in advance and replace other (i.e. not in vocabulary) words in the corpus with
    - <UNK>.
  - Very often, <UNK> also used to replace rare words

Today: Naïve Bayes Classifier for Text

- Dictionary based Vector space representation of text article
- Multivariate Bernoulli vs. Multinomial
- Multivariate Bernoulli naïve Bayes classifier
  - Testing
  - Training With Maximum Likelihood Estimation for estimating parameters
- Multinomial naïve Bayes classifier
  - Testing
  - Training With Maximum Likelihood Estimation for estimating parameters

‘Bag of words’→what probability model?

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun… It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet.

word	
great	.
love	.
recommend	.
laugh	.
happy	.
...	.

$$c^* = argmaxP(\mathbf{X} = \mathbf{x}|C = c_i)P(C = c_i)$$

$$\Pr(D = d \mid C = c_i)$$

?

‘Bag of words’→what probability model?

$$\Pr(D \mid C = c) = \ ?$$

Two Previous models

$$\Pr(W_1 = true, W_2 = \ false..., W_k = true \mid C = c)$$

$$\Pr(W_1 = n_1, W_2 = n_2, ..., W_k = n_k \mid C = c)$$

Naïve Probabilistic Models of text documents

$$\Pr(D \mid C = c) = \ ?$$

Two Previous models

$$\Pr(W_1 = true, W_2 = false \ldots, W_k = true \mid C = c)$$

Multivariate Bernoulli Distribution

$$\Pr(W_1 = n_1, W_2 = n_2, \ldots, W_k = n_k \mid C = c)$$

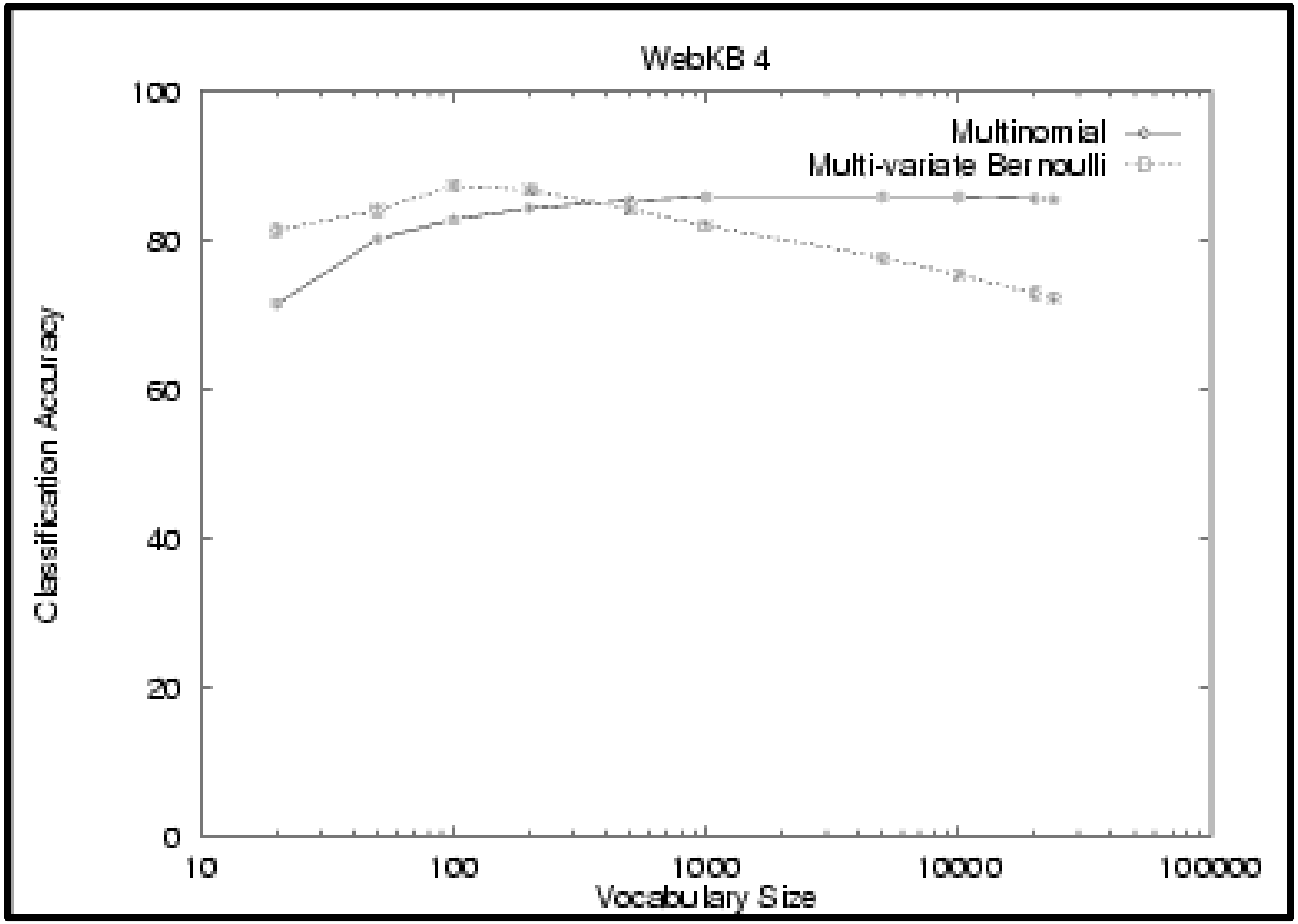
Multinomial Distribution

- Multinomial vs Multivariate Bernoulli?
- Multinomial model is almost always more effective in text applications!

Experiment: Multinomial vs multivariate Bernoulli

- M&N (1998) did some experiments to see which is better
- Determine if a university web page is {student, faculty, other_staff}
- Train on ~5,000 hand-labeled web pages
- – Cornell, Washington, U.Texas, Wisconsin
- Crawl and classify a new site (CMU)

Multinomial vs. multivariate Bernoulli



Today: Naïve Bayes Classifier for Text

- Dictionary based Vector space representation of text article
- Multivariate Bernoulli vs. Multinomial
- Multivariate Bernoulli naïve Bayes classifier
  - Testing
  - Training With Maximum Likelihood Estimation for estimating parameters
- Multinomial naïve Bayes classifier
  - Testing
  - Training With Maximum Likelihood Estimation for estimating parameters

Model 1: Multivariate Bernoulli

- Model 1: Multivariate Bernoulli
  - For each word in a dictionary, feature X_w
  - X_w = true in document *d* if *w* appears in *d*

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun… It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet.

word	*Boolean*
great	Yes
love	Yes
recommend	Yes
laugh	Yes
happy	Yes
hate	No
...	.

Model 1: Multivariate Bernoulli

- Model 1: Multivariate Bernoulli
  - For each word in a dictionary, feature X_w
  - X_w = true in document *d* if *w* appears in *d*

- Naive Bayes assumption:
  - Given the document's class label,
  - appearance of one word in the document tells us nothing about chances that another word appears

$$\Pr(W_1 = true, W_2 = false \ldots, W_k = true \mid C = c)$$

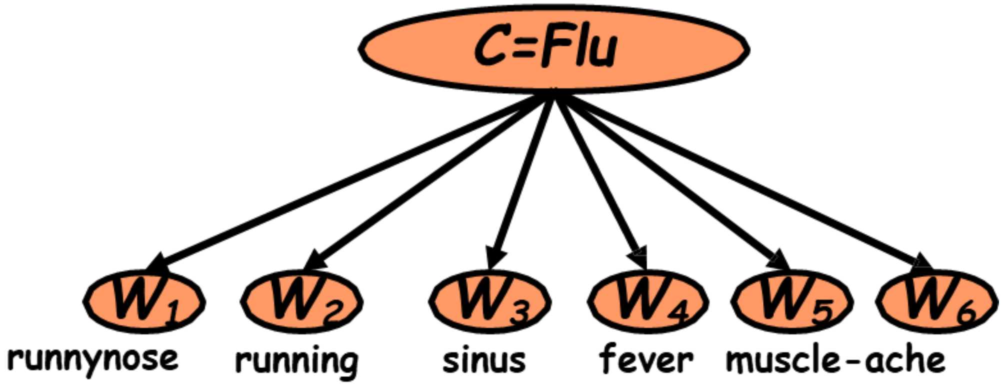
Model 1: Multivariate Bernoulli

- Naïve Bayes Classifier

word	*Boolean*
great	Yes
love	Yes
recommend	Yes
laugh	Yes
happy	Yes
hate	No
...	.

- **Conditional Independence Assumption:** Features (word presence) are independent of each other given the class variable:
- Multivariate Bernoulli model is appropriate for binary feature variables

Model 1: Multivariate Bernoulli



this is naïve

$$\Pr(W_1 = true, W_2 = false, \ldots, W_k = true \mid C = c)$$
$$= P(W_1 = true \mid C) \bullet P(W_2 = false \mid C) \bullet \cdots \bullet P(W_k = true \mid C)$$

Parameter estimation

- Multivariate Bernoulli model:

$$\hat{P}(w_i = true \mid c_j) = \text{fraction of documents of label } c_j \text{ in which word } w\_i \text{ appears}$$

- Smoothing to Avoid Overfitting

Underflow Prevention: log space

- Multiplying lots of probabilities, which are between 0 and 1, 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.

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} \left\{ \log P(c_j) + \sum_{i \in dictionary} \log P(x_i \mid c_j) \right\}$$

- Note that model is now just max of sum of weights…

Today: Naïve Bayes Classifier for Text

- Dictionary based Vector space representation of text article
- Multivariate Bernoulli vs. Multinomial
- Multivariate Bernoulli naïve Bayes classifier
  - Testing
  - Training With Maximum Likelihood Estimation for estimating parameters
- Multinomial naïve Bayes classifier
  - Testing
  - Training With Maximum Likelihood Estimation for estimating parameters

Model 2: Multinomial Naïve Bayes

- 'Bag of words' representation of text

word	*frequency*
great	2
love	2
recommend	1
laugh	1
happy	1
...	.

$$\Pr(W_1 = n_1, \ldots, W_k = n_k \mid C = c)$$

- Can be represented as a multinomial distribution.
- Words = like colored balls, there are K possible type of them (i.e. from a dictionary of K words)
- A Document = contains N words, each word occurs n_i times (like a bag of N colored balls)

Review: Multinomial distribution

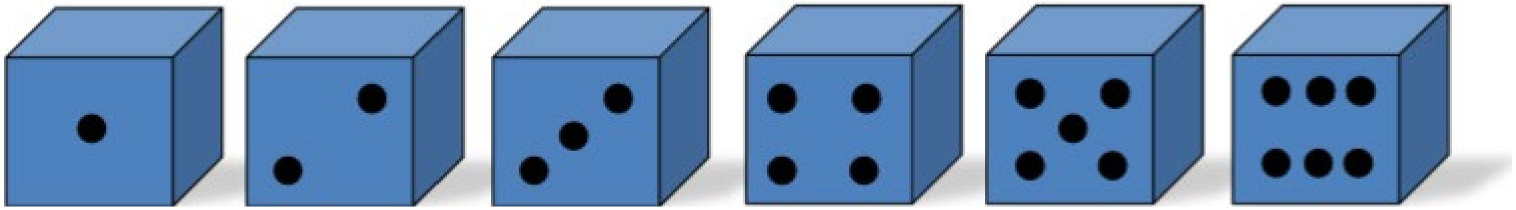
- The multinomial distribution is a generalization of the binomial distribution.

- The binomial distribution counts successes of an event (for example, heads in coin tosses).

- The parameters:
  - *N* (number of trials)
  - p (the probability of success of the event)

Review: Multinomial distribution

- The multinomial counts the number of a set of events (for example, how many times each side of a die comes up in a set of rolls).
  - The parameters:
  - N (number of trials)
  - $\theta_1,\ldots,\theta_k$ (the probability of success for each category)



Multinomial Distribution for Text Classification

- $W_1, W_2, ..W_k$ are variables

Number of possible orderings of N balls

$$P(W_1 = n_1, \ldots, W_k = n_k \mid c, N, \theta_{1,c}, .., \theta_{k,c}) = \frac{N!}{n_1! n_2! .. n_k!} \theta_{1,c}^{\;n_1} \theta_{2,c}^{\;n_2} .. \theta_{k,c}^{\;n_k}$$

Label invariant

$$\sum_{i=1}^{k} n_i = N \qquad \sum_{i=1}^{k} \theta_{i,c} = 1$$

Model 2: Multinomial Naïve Bayes

- 'Bag of words' – Testing Stage

word	frequency
great	2
love	2
recommend	1
laugh	1
happy	1
...	.

$$\underset{c}{\operatorname{argmax}} P(W_1 = n_1, \ldots, W_k = n_k, c)$$

$$= \underset{c}{\operatorname{argmax}} \{ p(c) * \theta_{1,c}{}^{n_1} \theta_{2,c}{}^{n_2} .. \theta_{k,c}{}^{n_k} \}$$

Today: Naïve Bayes Classifier for Text

- Dictionary based Vector space representation of text article
- Multivariate Bernoulli vs. Multinomial
- Multivariate Bernoulli naïve Bayes classifier
  - Testing
  - Training With Maximum Likelihood Estimation for estimating parameters
- Multinomial naïve Bayes classifier
  - Testing
  - Training With Maximum Likelihood Estimation for estimating parameters

Deriving the Maximum Likelihood Estimate for multinomial distribution

LIKELIHOOD: $$\underset{\theta_1,\dots,\theta_k}{\operatorname{argmax}}\ P(d_1,\dots,d_T \mid \theta_1,\dots,\theta_k)$$

function of θ $$= \underset{\theta_1,\dots,\theta_k}{\arg\max} \prod_{t=1}^{T} P(d_t \mid \theta_1,\dots,\theta_k)$$

$$= \underset{\theta_1,\dots,\theta_k}{\operatorname{argmax}} \prod_{t=1}^{T} \frac{N_{dt}!}{n_{1,d_t}!\,n_{2,d_t}!\dots n_{k,d_t}!} \theta_1^{n_{1,d_t}} \theta_2^{n_{2,d_t}} \dots \theta_k^{n_{k,d_t}}$$

$$= \underset{\theta_1,\dots,\theta_k}{\operatorname{argmax}} \prod_{t=1}^{T} \theta_1^{n_{1,d_t}} \theta_2^{n_{2,d_t}} \dots \theta_k^{n_{k,d_t}}$$

$$s.t. \sum_{i=1}^{k} \theta_i = 1$$

Deriving the Maximum Likelihood Estimate for multinomial distribution

$$\arg\max_{\theta_1,\ldots,\theta_k} \log(L(\theta)) \qquad s.t. \sum_{i=1}^{k} \theta_i = 1$$

$$= \arg\max_{\theta_1,\ldots,\theta_k} \log\left(\prod_{t=1}^{T} \theta_1^{n_{1,d_t}} \theta_2^{n_{2,d_t}} \ldots \theta_k^{n_{k,d_t}}\right)$$

$$= \arg\max_{\theta_1,\ldots,\theta_k} \sum_{t=1,\ldots T} n_{1,d_t} \log(\theta_1) + \sum_{t=1,\ldots T} n_{2,d_t} \log(\theta_2) + \ldots + \sum_{t=1,\ldots T} n_{k,d_t} \log(\theta_k)$$

Constrained optimization MLE estimator

$$\theta_i = \frac{\sum_{t=1,\ldots T} n_{i,d_t}}{\sum_{t=1,\ldots T} n_{1,d_t} + \sum_{t=1,\ldots T} n_{2,d_t} + \ldots + \sum_{t=1,\ldots T} n_{k,d_t}} = \frac{\sum_{t=1,\ldots T} n_{i,d_t}}{\sum_{t=1,\ldots T} N_{d_t}}$$

→ i.e. We can create a mega-document by concatenating all documents d_1 to d_T

→ Use relative frequency of w_i in mega-document

Deriving the Maximum Likelihood Estimate for multinomial Bayes Classifier

LIKELIHOOD: $$\underset{\theta_{1,Cj},..,\theta_{k,Cj}}{\operatorname{argmax}} P(d_1,\ldots,d_T|\Theta)$$

$$\underset{\theta_1,..,\theta_k}{\operatorname{argmax}} P(d_1,\ldots,d_T|\theta_1,\ldots,\theta_k)$$

Parameter estimation

- Multinomial model:

$$\hat{P}(X_i = w_i \mid c_j) = $$ fraction of times in which each dictionary word *w* appears across all documents of class c_j

- Can create a mega-document for class j by concatenating all documents on this class,
- Use frequency of w in mega-document

Multinomial: Learning Algorithm for parameter estimation with MLE

- From training corpus, extract *Vocabulary*
- Calculate required $P(c_j)$ and $P(w_k \mid c_j)$ terms
  - For each c_j in C do
    - $docs_j \leftarrow$ subset of documents for which the target class is c_j

$$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$

Multinomial: Learning Algorithm for parameter estimation with MLE

$$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$

- $Text_j \leftarrow$ is length n_j and is a single document containing all $docs_j$
- for each word w_k in *Vocabulary*
  - $n_{k,j} \leftarrow$ number of occurrences of w_k in $Text_j$; n_j *is length of* $Text_j$
  - $$P(w_k|c_j) \leftarrow \frac{n_{k,j}+\alpha}{n_j + \alpha\,|Vocabulary|} \qquad e.g., \alpha = 1$$

Relative frequency of word *w_k* appears across all documents of class c_j

Naive Bayes is Not So Naive

- Naïve Bayes: First and Second place in KDD-CUP 97 competition, among 16 (then) state of the art algorithms
  - Goal: Financial services industry direct mail response prediction model: Predict if the recipient of mail will actually respond to the advertisement – 750,000 records.
- Robust to Irrelevant Features
  - Irrelevant Features cancel each other without affecting results
  - Instead Decision Trees can heavily suffer from this.
- Very good in domains with many equally important features
  - Decision Trees suffer from fragmentation in such cases – especially if little data
- A good dependable baseline for text classification (but not the best)!
- Optimal if the Independence Assumptions hold: If assumed independence is correct, then it is the Bayes Optimal Classifier for problem
- Very Fast: Learning with one pass of counting over the data; testing linear in the number of attributes, and document collection size
- Low Storage requirements

References

- https://qiyanjun.github.io/2019f-UVA-CS6316-MachineLearning/
- Prof. Andrew Moore's review tutorial
- Prof. Ke Chen NB slides
- Prof. Carlos Guestrin recitation slides
- Prof. Raymond J. Mooney and Jimmy Lin's slides about language model
- Prof. Manning' textCat tutorial

Thanks for listening