Naïve Bayes Model

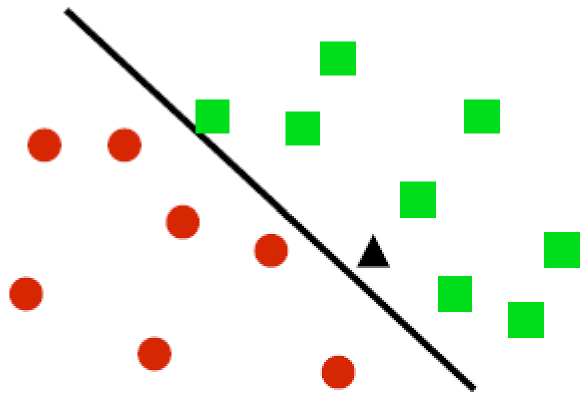
Rui Xia
School of Computer Science & Engineering
Nanjing University of Science & Technology
http://www.nustm.cn/~rxia

Naïve Bayes Models

- A Probabilistic Model
- A Generative Model
- Known as the “Naïve” Assumption
- Suitable for Discrete Distributions
- Widely used in Text Classification, Natural Language Processing and Pattern Recognition

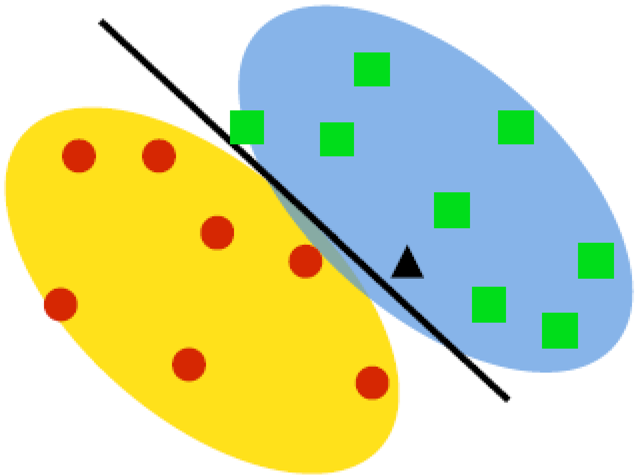
Generative vs. Discriminative

- Discriminative Model



It models the posterior probability of class label given observation $p(y|\boldsymbol{x})$

- Generative Model



It models the joint probability of class label and observation $p(\boldsymbol{x}, y)$, and then use the Bayes rule ($p(y|\boldsymbol{x}) = p(\boldsymbol{x}, y)/p(\boldsymbol{x})$) for prediction.

Naïve Bayes Assumption

- A Mixture Model

Class prior probability

$$p(\boldsymbol{x}, y = c_j) = p(y = c_j)p(\boldsymbol{x}|c_j)$$

Class-conditional probability

- Bag-of-words (BOW) representation

$$\boldsymbol{x} = (\omega_1, \omega_2, \dots, \omega_{|x|})$$

$$p(\boldsymbol{x}|c_j) = p(\omega_1, \omega_2, \dots, \omega_{|\boldsymbol{x}|}|c_j) = \prod_{h=1}^{|\boldsymbol{x}|} p(\omega_h|c_j)$$

Having two event models

Multinomial Event Model

Model Description

- Hypothesis

$$p(y = c_j) = \pi_j$$

$$p(\boldsymbol{x}|c_j) = p([\omega_1, \omega_2, \dots, \omega_{|\boldsymbol{x}|}]|c_j) = \prod_{h=1}^{|\boldsymbol{x}|} p(\omega_h|c_j)$$

$$= \prod_{i=1}^{V} p(t_i|c_j)^{N(t_i,\boldsymbol{x})} = \prod_{i=1}^{V} \theta_{i|j}^{N(t_i,\boldsymbol{x})}$$

- Joint Probability

Model Parameters

$$p(\boldsymbol{x}, y = c_j) = p(c_j)p(\boldsymbol{x}|c_j) = \pi_j \prod_{i=1}^{V} \theta_{i|j}^{N(t_i,\boldsymbol{x})}$$

Likelihood Function

- (Joint) Likelihood

$$
\begin{aligned}
L(\boldsymbol{\pi}, \boldsymbol{\theta}) &= \log \prod_{k=1}^{N} p(\boldsymbol{x}_k, y_k) \\
&= \log \prod_{k=1}^{N} \sum_{j=1}^{C} I(y_k = c_j) p(y_k = c_j) p(\boldsymbol{x}_k | y_k = c_j) \\
&= \sum_{k=1}^{N} \sum_{j=1}^{C} I(y_k = c_j) \log p(y_k = c_j) p(\boldsymbol{x}_k | y_k = c_j) \\
&= \sum_{k=1}^{N} \sum_{j=1}^{C} I(y_k = c_j) \log \pi_j \prod_{i=1}^{V} \theta_{i|j}^{N(t_i, \boldsymbol{x}_k)} \\
&= \sum_{k=1}^{N} \sum_{j=1}^{C} I(y_k = c_j) \left(\log \pi_j + \sum_{i=1}^{V} N(t_i, \boldsymbol{x}_k) \log \theta_{i|j} \right)
\end{aligned}
$$

Maximum Likelihood Estimation

- MLE Formulation

$$\max_{\pi,\boldsymbol{\theta}} L(\boldsymbol{\pi}, \boldsymbol{\theta})$$

$$s.t. \begin{cases} \sum_{j=1}^{C} \pi_j = 1 \\ \sum_{i=1}^{V} \theta_{i|j} = 1, j = 1, \dots, C \end{cases}$$

- Applying Lagrange multipliers

$$J = L(\boldsymbol{\pi}, \boldsymbol{\theta}) + \alpha\left(1 - \sum_{j=1}^{C} \pi_j\right) + \sum_{j=1}^{C} \beta_j \left(1 - \sum_{i=1}^{V} \theta_{i|j}\right)$$

$$= \sum_{k=1}^{N} \sum_{j=1}^{C} I(y_k = c_j)\left(\log\pi_j + \sum_{i=1}^{V} N(t_i, \boldsymbol{x}_k)\log\boldsymbol{\theta}_{i|j}\right) + \alpha\left(1 - \sum_{j=1}^{C} \pi_j\right) + \sum_{j=1}^{C} \beta_j \left(1 - \sum_{i=1}^{V} \theta_{i|j}\right)$$

Close-form MLE Solution

- Gradient

$$\frac{\partial J}{\partial \pi_j} = \sum_{k=1}^{N} I(y_k = c_j) \frac{1}{\pi_j} - \alpha = 0$$

$$\frac{\partial J}{\partial \theta_{i|j}} = \sum_{k=1}^{N} I(y_k = c_j) \frac{N(t_i, \boldsymbol{x}_k)}{\theta_{i|j}} - \beta_j = 0$$

- MLE Solution

$$\pi_j = \frac{\sum_{k=1}^{N} I(y_k = c_j)}{\sum_{k=1}^{N} \sum_{j'=1}^{C} I(y_k = c_j)} = \frac{N_j}{N}$$

$$\theta_{i|j} = \frac{\sum_{k=1}^{N} I(y_k = c_j) N(t_i, \boldsymbol{x}_k)}{\sum_{k=1}^{N} I(y_k = c_j) \sum_{i'=1}^{V} N(t_{i'}, \boldsymbol{x}_k)}$$

Laplace Smoothing

- In order to prevent from zero probability

$$p(\boldsymbol{x}, y = c_j) = \pi_j \prod_{i=1}^{V} \theta_{i|j}^{N(t_i, \boldsymbol{x})}$$

- Laplace Smoothing

$$\theta_{i|j} = \frac{\sum_{k=1}^{N} I(y_k = c_j)\, N(t_i, \boldsymbol{x}_k)}{\sum_{i'=1}^{V} \sum_{k=1}^{N} I(y_k = c_j) N(t_{i'}, \boldsymbol{x}_k)} \qquad \pi_j = \frac{\sum_{k=1}^{N} I(y_k = c_j)}{\sum_{j'=1}^{C} \sum_{k=1}^{N} I(y_k = c_j)}$$

$$\Downarrow \qquad\qquad\qquad \Downarrow$$

$$\theta_{i|j} = \frac{\sum_{k=1}^{N} I(y_k = c_j)\, N(t_i, \boldsymbol{x}_k) + 1}{\sum_{i'=1}^{V} \sum_{k=1}^{N} I(y_k = c_j) N(t_{i'}, \boldsymbol{x}_k) + V} \qquad \pi_j = \frac{\sum_{k=1}^{N} I(y_k = c_j) + 1}{\sum_{j'=1}^{C} \sum_{k=1}^{N} I(y_k = c_j) + C}$$

Multi-variate Bernoulli Event Model

Model Description

- Hypothesis

$$p(y = c_j) = \pi_j$$

$$p(\boldsymbol{x}|y = c_j) = p(t_1, t_2, \dots, t_V|c_j)$$
$$= \prod_{i=1}^{V}\left[I(t_i \epsilon \boldsymbol{x})p(t_i|c_j) + I(t_i \notin \boldsymbol{x})\left(1 - p(t_i|c_j)\right)\right]$$
$$= \prod_{i=1}^{V}\left[I(t_i \epsilon \boldsymbol{x})\mu_{i|j} + I(t_i \notin \boldsymbol{x})(1 - \mu_{i|j})\right]$$

- Joint Probability

Model Parameters

$$p(\boldsymbol{x}, c_j) = \pi_j \prod_{i=1}^{V}\left[I(t_i \epsilon \boldsymbol{x})\mu_{i|j} + I(t_i \notin \boldsymbol{x})(1 - \mu_{i|j})\right]$$

Likelihood Function

- (Joint) Likelihood

$$
\begin{aligned}
L(\boldsymbol{\pi}, \boldsymbol{\mu}) &= \log \prod_{k=1}^{N} p(\boldsymbol{x}_k, y_k) \\
&= \sum_{k=1}^{N} \log \sum_{j=1}^{C} I\left(y_k = c_j\right) p(\boldsymbol{x}_k, y_k) \\
&= \sum_{k=1}^{N} \sum_{j=1}^{C} I\left(y_k = c_j\right) \log p(c_j) \prod_{i=1}^{V} I(t_i \epsilon \boldsymbol{x}) p\left(t_i \middle| c_j\right) + I(t_i \notin \boldsymbol{x}) \left(1 - p\left(t_i \middle| c_j\right)\right) \\
&= \sum_{k=1}^{N} \sum_{j=1}^{C} I\left(y_k = c_j\right) \left(\log \pi_j + \sum_{i=1}^{V} I(t_i \epsilon \boldsymbol{x}_k) \log \mu_{i|j} + I(t_i \notin \boldsymbol{x}_k) \log(1 - \mu_{i|j}) \right)
\end{aligned}
$$

Maximum Likelihood Estimation

- MLE Formulation

$$\max_{\pi,\mu} L(\boldsymbol{\pi}, \boldsymbol{\mu})$$

$$s.t. \sum_{j=1}^{C} \pi_j = 1$$

- Applying Lagrange multipliers

$$J = L(\boldsymbol{\pi}, \boldsymbol{\mu}) + \alpha\left(1 - \sum_{j=1}^{C} \pi_j\right)$$

$$= \sum_{k=1}^{N} \sum_{j=1}^{C} I\left(y_k = c_j\right)\left(\log\pi_j + \sum_{i=1}^{V} I(t_i \epsilon \boldsymbol{x}_k) \log\mu_{i|j} + I(t_i \notin \boldsymbol{x})\log(1 - \mu_{i|j})\right) + \alpha\left(1 - \sum_{j=1}^{C} \pi_j\right)$$

Close-form MLE Solution

- Gradient

$$\frac{\partial J}{\partial \pi_j} = \sum_{k=1}^{N} I(y_k = c_j)\frac{1}{\pi_j} - \alpha = 0$$

$$\frac{\partial J}{\partial \mu_{i|j}} = \sum_{k=1}^{N} I(y_k = c_j)\left(\frac{I(t_i \epsilon \boldsymbol{x}_k)}{\mu_{i|j}} - \frac{I(t_i \notin \boldsymbol{x}_k)}{1-\mu_{i|j}}\right) = 0, \forall j = 1, \dots, C.$$

- MLE Solution

$$\pi_j = \frac{\sum_{k=1}^{N} I(y_k = c_j)}{\sum_{k=1}^{N} \sum_{j'=1}^{C} I(y_k = c_{j'})} = \frac{N_j}{N}$$

$$\mu_{i|j} = \frac{\sum_{k=1}^{N} I(y_k = c_j)\, I(t_i \epsilon \boldsymbol{x}_k)}{\sum_{k=1}^{N} I(y_k = c_j)}$$

Laplace Smoothing

- In order to prevent from zero probability

$$p(\boldsymbol{x}, c_j) = \pi_j \prod_{i=1}^{V} [I(t_i \epsilon \boldsymbol{x}) \mu_{i|j} + I(t_i \notin \boldsymbol{x})(1 - \mu_{i|j})]$$

- Laplace Smoothing

$$\mu_{i|j} = \frac{\sum_{k=1}^{N} I(y_k = c_j) I(t_i \epsilon x_k)}{\sum_{k=1}^{N} I(y_k = c_j)} \qquad \pi_j = \frac{\sum_{k=1}^{N} I(y_k = c_j)}{\sum_{j'=1}^{C} \sum_{k=1}^{N} I(y_k = c_j)}$$

$$\Downarrow \qquad \Downarrow$$

$$\mu_{i|j} = \frac{\sum_{k=1}^{N} I(y_k = c_j) I(t_i \epsilon x_k) + 1}{\sum_{k=1}^{N} I(y_k = c_j) + 2} \qquad \pi_j = \frac{\sum_{k=1}^{N} I(y_k = c_j) + 1}{\sum_{j'=1}^{C} \sum_{k=1}^{N} I(y_k = c_j) + C}$$

Text Classification as An Example

Data sets

- Training data

ID	Text	Label
$d_{tr}1$	Chinese Beijing Chinese	C
$d_{tr}2$	Chinese Chinese Shanghai	C
$d_{tr}3$	Chinese Macao	C
$d_{tr}4$	Tokyo Japan Chinese	J

- Test data

ID	Text
$d_{te}1$	Chinese Chinese Chinese Tokyo Japan
$d_{te}2$	Tokyo Tokyo Japan Shanghai

- Class labels

$c1 = C;$

$c2 = J$

- Feature vector

$t1 = $ Beijing

$t2 = $ Chinese

$t3 = $ Japan

$t4 = $ Macao

$t5 = $ Shanghai

$t6 = $ Tokyo

Multinomial Naïve Bayes

- Training

		Doc	t1	t2	t3	t4	t5	t6
Term Frequency	c1	3	1	5	0	1	1	0
	c2	1	0	1	1	0	0	1
Probability	c1	3/4	2/14	(5+1)/(1+5+1+1+6)=6/14	1/14	2/14	2/14	1/14
	c2	1/4	1/9	(1+1)/(1+1+1+6)=2/9	2/9	1/9	1/9	2/9

- Prediction

	Un-normalized	Normalized
$P(c1\|d_{te}1)$	(3/4)*(6/14)^3*(1/14)*(1/14)=0.0030121	0.689757
$P(c2\|d_{te}1)$	(1/4)*(2/9)^3*(2/9)*(2/9)=0.0013548	0.310243
$P(c1\|d_{te}2)$	(3/4)*(1/14)^2*(1/14)*(2/14)	0.113547
$P(c2\|d_{te}2)$	(1/4)*(2/9)^2*(2/9)*(1/9)	0.886453

Multi-variate Bernoulli Naïve Bayes

- Training

		Doc	t1	t2	t3	t4	t5	t6
Document Frequency	c1	3	1	3	0	1	1	0
	c2	1	0	1	1	0	0	1
Probability	c1	3/4	2/5	(3+1)/(3+2)=4/5	1/5	2/5	2/5	1/5
	c2	1/4	1/3	(1+1)/(1+2)=2/3	2/3	1/3	1/3	2/3

- Prediction

	Un-normalized	Normalized
$P(c1\|d_{te}1)$	(3/4)*(1-2/5)*4/5*1/5*(1-2/5)* (1-2/5)* 1/5=0.005184	0.1911
$P(c2\|d_{te}1)$	(1/4)*(1-1/3)*2/3*2/3*(1-1/3)* (1-1/3)*2/3=0.02195	0.8089
$P(c1\|d_{te}2)$	(3/4)*(1-2/5)*(1-3/5)*1/5*(1-2/5)*2/5*1/5=0.001728	0.2395
$P(c2\|d_{te}2)$	(1/4)*(1-1/3)*(1-2/3)*2/3*(1-1/3)*1/3*2/3=0.005487	0.7605

Xia-NB Software

- Functions
  - Written in C++
  - Support multinomial and multi-variate Bernoulli event model
  - Laplace smoothing
  - Uniform data format like SVM-light/LibSVM
  - Fast running with sparse representation

- Download

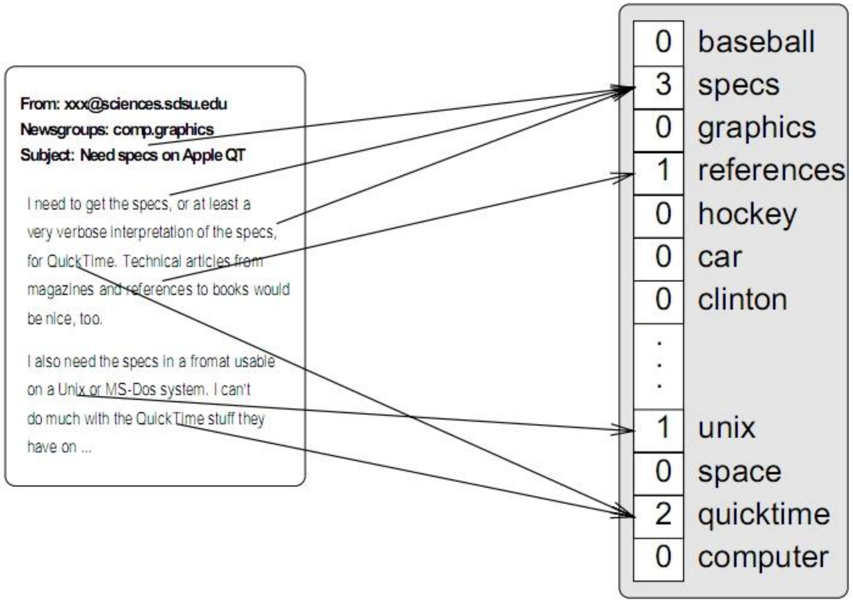
  https://github.com/NUSTM/XIA-NB

Practice 7: Naïve Bayes for Text Classification

- Implement naïve Bayes algorithm with
  - Multinomial event model
  - Multi-variate Bernoulli model
- Running the algorithm on the Tsinghua text classification data set (http://www.nustm.cn/member/rxia/ml/data/Tsinghua.zip) and report the classification accuracy.
- Implement softmax regression based on the Bag-of-words (BOW) representation and two kinds of term weighting methods (term frequency and presences).
- Compare the naïve Bayes and softmax regression, from the perspective of model and results.

Text Representation (in softmax regression)

- Vector Space Model (VSM)
  also called Bag-of-words (BOW) model



Vocabulary $[t_1, t_2, \cdots, t_i, \cdots, t_V] =$
[baseball, specs, graphics, …, quicktime, computer]

- Term Weighting Methods
  - BOOL (presence)

$$\omega_{ki} = \begin{cases} 1, & if\ t_i\ exists\ in\ \boldsymbol{d}_k \\ 0, & otherwise \end{cases}$$

  - Term frequency (TF)

$$\omega_{ki} = tf_{ki}$$

  - Inverse document frequency (IDF)

$$\omega_i = \log \frac{N}{df_i}$$

  - TF-IDF

$$\omega_{ki} = tf_{ki} \cdot \log \frac{N}{df_i}$$

Questions?