NAÏVE BAYES

David Kauchak
CS159 Fall 2014

Admin

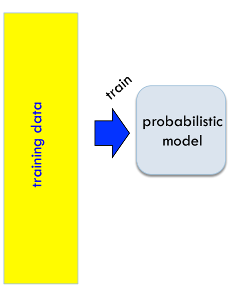
Assignment 7 out soon (due next Friday at 5pm)

Quiz #3 next Tuesday

- Text similarity -> this week (though, light on ML)

Project proposal presentations Tuesday

Probabilistic Modeling



Model the data with a probabilistic model

specifically, learn p(*features, label*)

p(*features, label*) tells us how likely these features and this example are

Basic steps for probabilistic modeling

Step 1: pick a model

Step 2: figure out how to estimate the probabilities for the model

Step 3 (optional): deal with overfitting

Probabilistic models

Which model do we use, i.e. how do we calculate p(*feature, label*)?

How do train the model, i.e. how to we we estimate the probabilities for the model?

How do we deal with overfitting?

Naïve Bayes assumption

$$p(features, label) = p(y)\prod_{j=1}^{m} p(x_j \mid y, x_1, ..., x_{j-1})$$

$$p(x_j \mid y, x_1, x_2, ..., x_{j-1}) = p(x_j \mid y)$$

What does this assume?

Naïve Bayes assumption

$$p(features, label) = p(y)\prod_{j=1}^{m} p(x_j \mid y, x_1, ..., x_{j-1})$$

$$p(x_j \mid y, x_1, x_2, ..., x_{j-1}) = p(x_j \mid y)$$

Assumes feature i is independent of the the other features *given the label*

Naïve Bayes model

$$p(features, label) = p(y)\prod_{j=1}^{m} p(x_j \mid y, x_1, ..., x_{j-1})$$

$$= p(y)\prod_{j=1}^{m} p(x_j \mid y)$$

naïve Bayes assumption

$p(x_i \mid y)$ is the probability of a particular feature value given the label

How do we model this?

- for binary features (e.g., "banana" occurs in the text)
- for discrete features (e.g., "banana" occurs x_i times)
- for real valued features (e.g, the text contains x_i proportion of verbs)

p(x|y)

Binary features (aka, Bernoulli Naïve Bayes) :

$$p(x_j \mid y) = \begin{cases} \theta_j & if\ x_i = 1 \\ 1-\theta_j & otherwise \end{cases}$$

biased coin toss!

Other features types:

Could use a lookup table for each value, but doesn't generalize well

Better, model as a distribution:

- gaussian (i.e. normal) distribution
- poisson distribution
- multinomial distribution (more on this later)
- …

Basic steps for probabilistic modeling

Probabilistic models

Step 1: pick a model

Which model do we use, i.e. how do we calculate p(*feature, label*)?

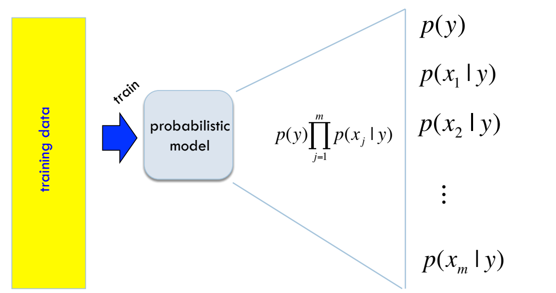
Step 2: figure out how to estimate the probabilities for the model

How do train the model, i.e. how to we we estimate the probabilities for the model?

Step 3 (optional): deal with overfitting

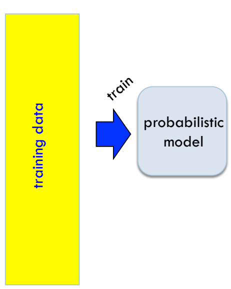
How do we deal with overfitting?

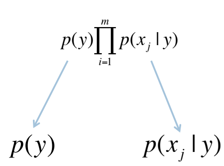
Obtaining probabilities



(m = number of features)

MLE estimation for NB





What are the MLE estimates for these?

Maximum likelihood estimates

$$p(y) = \frac{count(y)}{n}$$

$\frac{\text{number of examples with label}}{\text{total number of examples}}$

$$p(x_j|y) = \frac{count(x_j, y)}{count(y)}$$

$\frac{\text{number of examples with the label with feature}}{\text{number of examples with label}}$

What does training a NB model then involve?
How difficult is this to calculate?

Text classification

$$p(y) = \frac{count(y)}{n}$$

$$p(w_j \mid y) = \frac{count(w_j, y)}{count(y)}$$

Unigram features:
w_j, whether or not word w_j occurs in the text

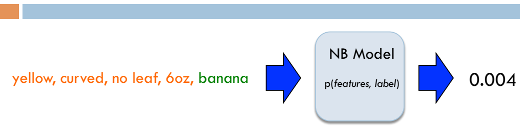
What are these counts for text classification with unigram features?

text classification

$$p(y) = \frac{count(y)}{n} \qquad \frac{\text{number of texts with label}}{\text{total number of texts}}$$

$$p(w_j \mid y) = \frac{count(w_j, y)}{count(y)} \qquad \frac{\text{number of texts with the label with word } w_j}{\text{number of texts with label}}$$

Naïve Bayes classification

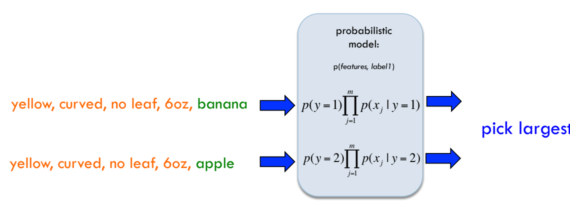


$$p(y)\prod_{j=1}^{m} p(x_j \mid y)$$

Given an unlabeled example: yellow, curved, no leaf, 6oz predict the label

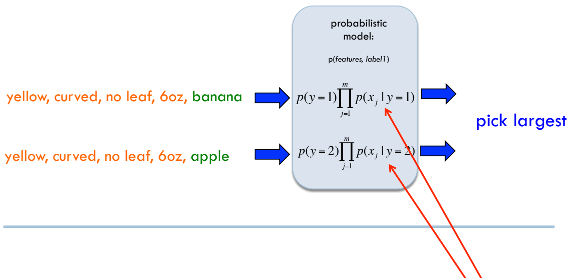
How do we use a probabilistic model for classification/prediction?

NB classification



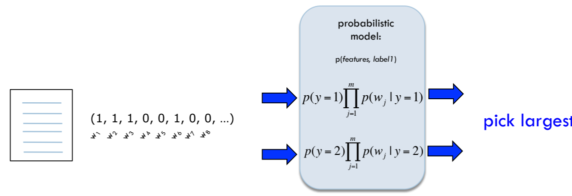
label = $\text{argmax}_{y \in labels}\, p(y)\prod_{j=1}^{m} p(x_j \mid y)$

NB classification



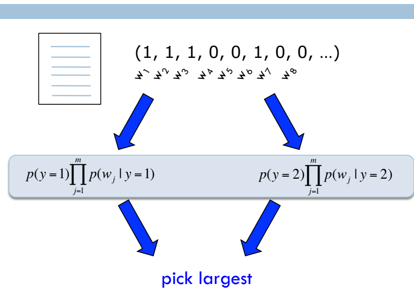
Notice that each label has it's own separate set of parameters, i.e. $p(x_j|y)$

Bernoulli NB for text classification



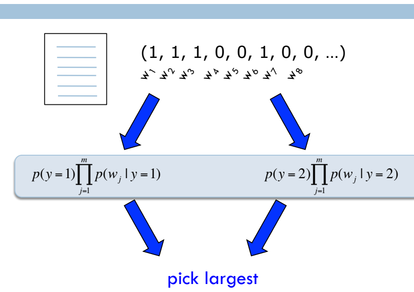
How good is this model for *text* classification?

Bernoulli NB for text classification



For text classification, what is this computation?
Does it make sense?

Bernoulli NB for text classification



Each word that occurs, contributes $p(w_j|y)$
Each word that does NOT occur, contributes $1-p(w_j|y)$

Generative Story

To classify with a model, we're given an example and we obtain the probability

We can also ask how a given model would *generate* an example

This is the "generative story" for a model

Looking at the generative story can help understand the model

We also can use generative stories to help develop a model

Bernoulli NB generative story

$$p(y)\prod_{j=1}^{m} p(x_j \mid y)$$

What is the generative story for the NB model?

Bernoulli NB generative story

$$p(y)\prod_{j=1}^{m} p(x_j \mid y)$$

1. Pick a label according to p(y)
   - roll a biased, num_labels-sided die
2. For each feature:
   - Flip a *biased* coin:
     - if heads, include the feature
     - if tails, don't include the feature

What does this mean for text classification, assuming unigram features?

Bernoulli NB generative story

$$p(y)\prod_{j=1}^{m} p(w_j \mid y)$$

1. Pick a label according to p(y)
   - roll a biased, num_labels-sided die
2. For each word in your vocabulary:
   - Flip a *biased* coin:
     - if heads, include the word in the text
     - if tails, don't include the word

Bernoulli NB

$$p(y)\prod_{j=1}^{m} p(x_j \mid y)$$

Pros/cons?

Bernoulli NB

Pros

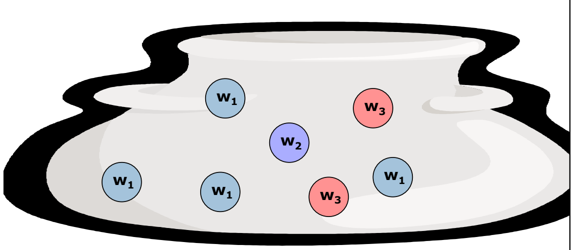
- Easy to implement
- Fast!
- Can be done on large data sets

Cons

- Naïve Bayes assumption is generally not true
- Performance isn't as good as more complicated models
- For text classification (and other sparse feature domains) the $p(x_i=0 \mid y)$ can be problematice

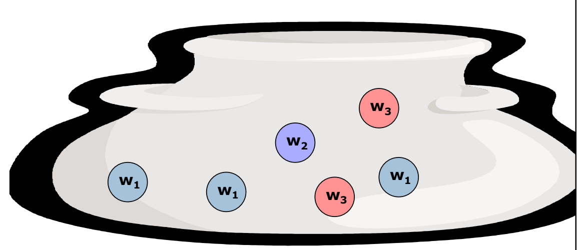
Another generative story

Randomly draw words from a "bag of words" until document length is reached



Draw words from a fixed distribution

Selected: w_1

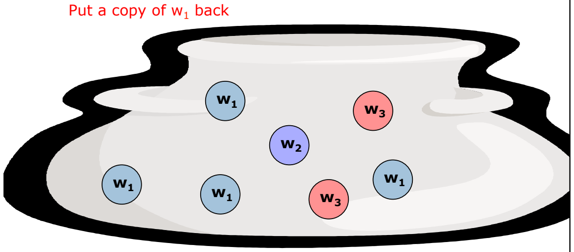


Draw words from a fixed distribution

Selected: w_1

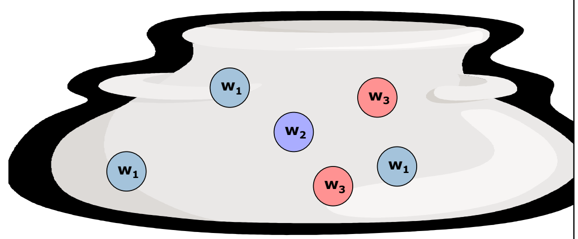
Put a copy of w_1 back

sampling with replacement



Draw words from a fixed distribution

Selected: w_1 w_1

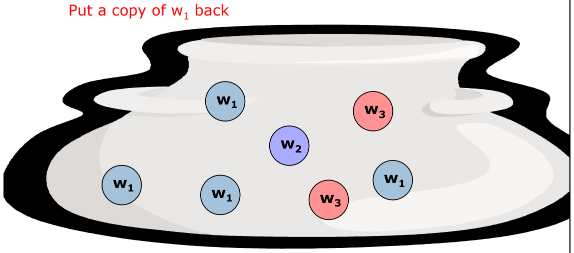


Draw words from a fixed distribution

Selected: w_1 w_1

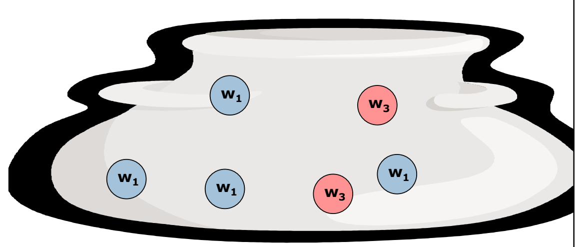
Put a copy of w_1 back

sampling with replacement



Draw words from a fixed distribution

Selected: w_1 w_1 w_2

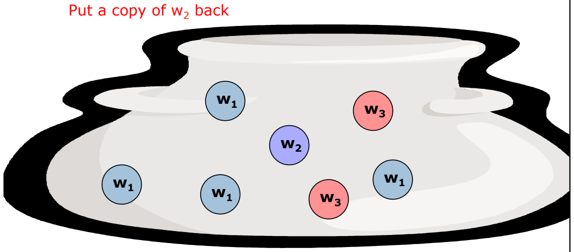


Draw words from a fixed distribution

Selected: w_1 w_1 w_2

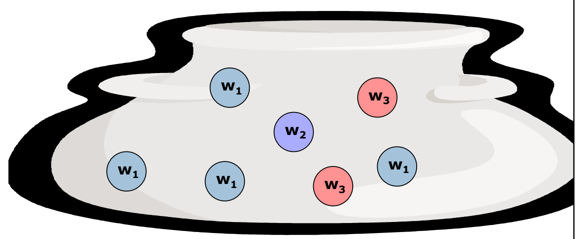
Put a copy of w_2 back

sampling with replacement



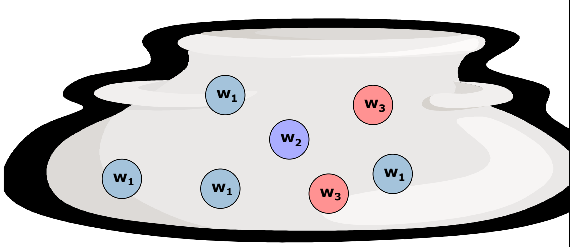
Draw words from a fixed distribution

Selected: w_1 w_1 w_2 …



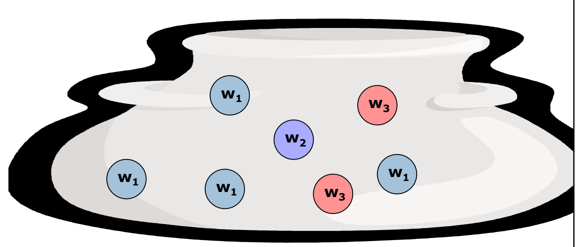
Draw words from a fixed distribution

Is this a NB model, i.e. does it assume each individual word occurrence is independent?



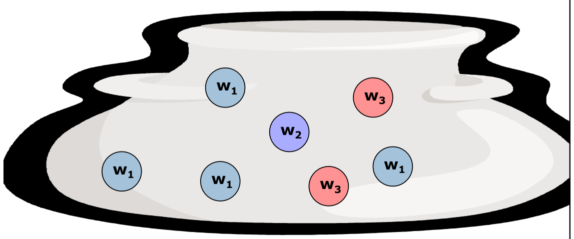
Draw words from a fixed distribution

Yes! Doesn't matter what words were drawn previously, still the same probability of getting any particular word



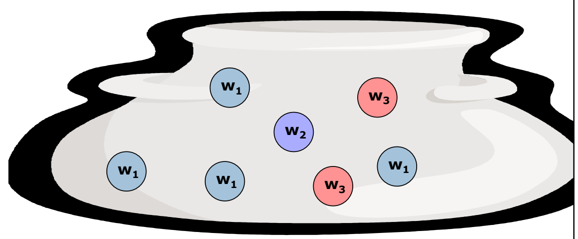
Draw words from a fixed distribution

Does this model handle multiple word occurrences?



Draw words from a fixed distribution

Selected: w_1 w_1 w_2 …



NB generative story

Bernoulli NB

1. Pick a label according to p(y)
   - roll a biased, num_labels-sided die
2. For each word in your vocabulary:
   - Flip a biased coin:
     - if heads, include the word in the text
     - if tails, don't include the word

Multinomial NB

1. Pick a label according to p(y)
   - roll a biased, num_labels-sided die
2. Keep drawing words from *p(words|y)* until text length has been reached.

Probabilities

Bernoulli NB

1. Pick a label according to p(y)
   - roll a biased, num_labels-sided die
2. For each word in your vocabulary:
   - Flip a biased coin:
     - if heads, include the word in the text
     - if tails, don't include the word

$$p(y)\prod_{j=1}^{m} p(x_j \mid y)$$

(1, 1, 1, 0, 0, 1, 0, 0, …)
w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8

Multinomial NB

1. Pick a label according to p(y)
   - roll a biased, num_labels-sided die
2. Keep drawing words from *p(words|y)* until document length has been reached

?

(4, 1, 2, 0, 0, 7, 0, 0, …)
w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8

A digression: rolling dice

What's the probability of getting a 3 for a single roll of this dice?

1/6

A digression: rolling dice

What is the probability distribution over possible single rolls?

1/6	1/6	1/6	1/6	1/6	1/6
1	2	3	4	5	6

A digression: rolling dice

What if I told you 1 was twice as likely as the others?

2/7	1/7	1/7	1/7	1/7	1/7
1	2	3	4	5	6

A digression: rolling dice

What if I rolled 400 times and got the following number?

1: 100
2: 50
3: 50
4: 100
5: 50
6: 50

1/4	1/8	1/8	1/4	1/8	1/8
1	2	3	4	5	6

A digression: rolling dice

1. What it the probability of rolling a 1 and a 5 (in any order)?
2. Two 1s and a 5 (in any order)?
3. Five 1s and two 5s (in any order)?

1/4	1/8	1/8	1/4	1/8	1/8
1	2	3	4	5	6

A digression: rolling dice

1. What it the probability of rolling a 1 and a 5 (in any order)?

   (1/4 * 1/8) * 2 = 1/16

   prob. of those two rolls

   number of ways that can happen (1,5 and 5,1)

2. Two 1s and a 5 (in any order)?

   $((1/4)^2 * 1/8) * 3 = 3/128$

3. Five 1s and two 5s (in any order)?

   $((1/4)^5 * (1/8)^3) * 21 = 21/524{,}288 = 0.00004$

General formula?

1/4	1/8	1/8	1/4	1/8	1/8
1	2	3	4	5	6

Multinomial distribution

Multinomial distribution: independent draws over *m* possible categories

If we have frequency counts x1, x2, …, xm over each of the categories, the probability is:

$$p(x_1, x_2, \ldots, x_m \mid \theta_1, \theta_2, \ldots, \theta_m) = \frac{n!}{\prod_{j=1}^{m} x_j!} \prod_{j=1}^{m} \theta_j^{x_j}$$

number of different ways to get those counts

probability of particular counts

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	
1	2	3	4	5	6	…

Multinomial distribution

$$p(x_1, x_2, \ldots, x_m \mid \theta_1, \theta_2, \ldots, \theta_m) = \frac{n!}{\prod_{j=1}^{m} x_j!} \prod_{j=1}^{m} \theta_j^{x_j}$$

What are θ_j?

Are there any constraints on the values that they can take?

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	
1	2	3	4	5	6	…

Multinomial distribution

$$p(x_1, x_2, \ldots, x_m \mid \theta_1, \theta_2, \ldots, \theta_m) = \frac{n!}{\prod_{j=1}^{m} x_j!} \prod_{j=1}^{m} \theta_j^{x_j}$$

θ_j: probability of rolling "j"

$$\theta_j \geq 0$$

$$\sum_{j=1}^{m} \theta_j = 1$$

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	
1	2	3	4	5	6	…

Back to words…

Why the digression?

$$p(x_1, x_2, \ldots, x_m \mid \theta_1, \theta_2, \ldots, \theta_m) = \frac{n!}{\prod_{j=1}^{m} x_j!} \prod_{j=1}^{m} \theta_j^{x_j}$$

Drawing words from a bag is the same as rolling a die!

number of sides = number of words in the vocabulary

Back to words…

Why the digression?

$$p(x_1, x_2, \ldots, x_m \mid \theta_1, \theta_2, \ldots, \theta_m) = \frac{n!}{\prod_{j=1}^{m} x_j!} \prod_{j=1}^{m} \theta_j^{x_j}$$

$$p(features, label) = p(y) \frac{n!}{\prod_{j=1}^{m} x_j!} \prod_{j=1}^{m} (\theta_y)_j^{x_j}$$

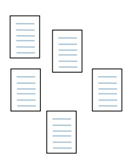
θ_j for class y

Basic steps for probabilistic modeling

Model each class as a multinomial:

$$p(features, label) = p(y) \frac{n!}{\prod_{j=1}^{m} x_j!} \prod_{j=1}^{m} (\theta_y)_j^{x_j}$$

Step 2: figure out how to estimate the probabilities for the model



How do we train the model, i.e. estimate θ_j for each class?

A digression: rolling dice

What if I rolled 400 times and got the following number?

1: 100
2: 50
3: 50
4: 100
5: 50
6: 50

1/4	1/8	1/8	1/4	1/8	1/8
1	2	3	4	5	6

Training a multinomial

$label_1$

$label_2$

1/4	1/8	1/8	1/4	1/8	1/8
1	2	3	4	5	6

Training a multinomial

$label_1$

For each label, y:

w1: 100 times
w2: 50 times
w3: 10 times
w4: …

$$\theta_j = \frac{count(w_j, y)}{\sum_{k=1}^{m} count(w_k, y)}$$

$$= \frac{\text{number of times word } w_j \text{ occurs in label y docs}}{\text{total number of words in label y docs}}$$

1/4	1/8	1/8	1/4	1/8	1/8
1	2	3	4	5	6

Classifying with a multinomial

(10, 2, 6, 0, 0, 1, 0, 0, …)

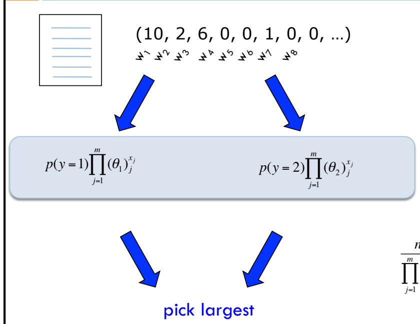
w1 w2 w3 w4 w5 w6 w7 w8

$$p(y=1)\frac{n!}{\prod_{j=1}^{m} x_j!}\prod_{j=1}^{m}(\theta_1)_j^{x_j} \qquad p(y=2)\frac{n!}{\prod_{j=1}^{m} x_j!}\prod_{j=1}^{m}(\theta_2)_j^{x_j}$$

Any way I can make this simpler?

pick largest

Classifying with a multinomial



$\frac{n!}{\prod_{j=1}^{m} x_m!}$ Is a constant!

Multinomial finalized

Training:

- Calculate p(label)
- For each label, calculate θ s

$$\theta_j = \frac{count(w_j, y)}{\sum_{k=1}^{m} count(w_k, y)}$$

Classification:

- Get word counts
- For each label you had in training, calculate:

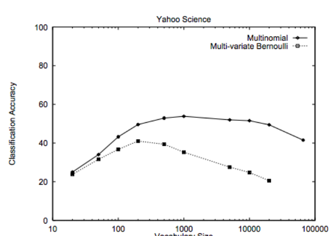
$$p(y)\prod_{j=1}^{m}\theta_j^{x_j}$$

and pick the largest

Multinomial vs. Bernoulli?

Handles word frequency

Given enough data, tends to performs better

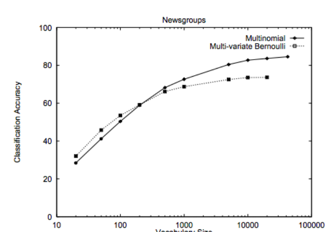


http://www.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf

Multinomial vs. Bernoulli?

Handles word frequency

Given enough data, tends to performs better

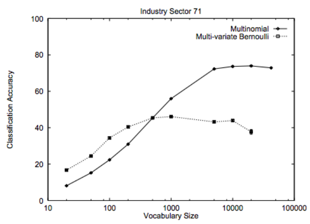


http://www.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf

Multinomial vs. Bernoulli?

Handles word frequency

Given enough data, tends to performs better



http://www.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf