Dirichlet-Multinomial and Naive Bayes

Instructor: Alan Ritter

Last time: Beta-Binomial

- Binary random variable: bent coin

Data Likelihood:

$$P(x_1, x_2, \ldots, x_n | \theta_H) = \theta_H^{\#H}(1 - \theta_H)^{\#T}$$

Last time: Beta-Binomial

- Binary random variable: bent coin

Data Likelihood:

$$P(x_1, x_2, \ldots, x_n | \theta_H) = \theta_H^{\#H}(1-\theta_H)^{\#T}$$

Last time: Beta-Binomial

- Binary random variable: bent coin

Data Likelihood:

$$P(x_1, x_2, \ldots, x_n | \theta_H) = \theta_H^{\#H}(1-\theta_H)^{\#T}$$

Prior (Beta distribution):

$$P(\theta_H | \alpha, \beta) = \frac{1}{B(\alpha, \beta)} \theta_H^{\alpha-1}(1-\theta_H)^{\beta-1}$$

Last time: Beta-Binomial

- Binary random variable: bent coin

Data Likelihood:

$$P(x_1, x_2, \ldots, x_n | \theta_H) = \theta_H^{\#H}(1-\theta_H)^{\#T}$$

Prior (Beta distribution):

$$P(\theta_H|\alpha,\beta) = \frac{1}{B(\alpha,\beta)}\theta_H^{\alpha-1}(1-\theta_H)^{\beta-1}$$

Posterior:

$$P(\theta_H|\alpha,\beta,x_1,\ldots,x_n) = \frac{1}{B(\alpha+\#H,\beta+\#T)}\theta^{\#H+\alpha-1}(1-\theta)^{\#T+\beta-1}$$

Last time: Beta-Binomial

- Binary random variable: bent coin

Maximum Likelihood:

$$\theta^{ML} = \frac{\#H}{\#T + \#H}$$

Last time: Beta-Binomial

- Binary random variable: bent coin

Maximum Likelihood:

$$\theta^{ML} = \frac{\#H}{\#T + \#H}$$

Last time: Beta-Binomial

- Binary random variable: bent coin

Maximum Likelihood:

$$\theta^{ML} = \frac{\#H}{\#T + \#H}$$

Maximum a Posteriori:

$$\theta^{MAP} = \frac{\#H + \alpha - 1}{\#T + \#H + \alpha + \beta - 2}$$

K-Sided Dice

- Weighted
  - (Generalization of Bent Coin)
- Assume an observed sequence of rolls:

1123213213

$$\theta_1 \qquad \theta_2 \qquad \theta_3$$

K-Sided Dice

- Weighted
  - (Generalization of Bent Coin)
- Assume an observed sequence of rolls:

1123213213

$$\theta_1 \qquad \theta_2 \qquad \theta_3$$

$$P(x;\theta) = \theta_x$$

Likelihood

$$P(1123213213|\theta) = \theta_1 \times \theta_1 \times \theta_2 \times \ldots \times \theta_3$$

$$= \theta_1^4 \times \theta_2^3 \times \theta_3^3$$

Likelihood In General

- N Dice Rolls, K possible outcomes:

$$P(D|\theta) = \prod_{k=1}^{K} \theta_k^{N_k}$$

Likelihood In General

- N Dice Rolls, K possible outcomes:

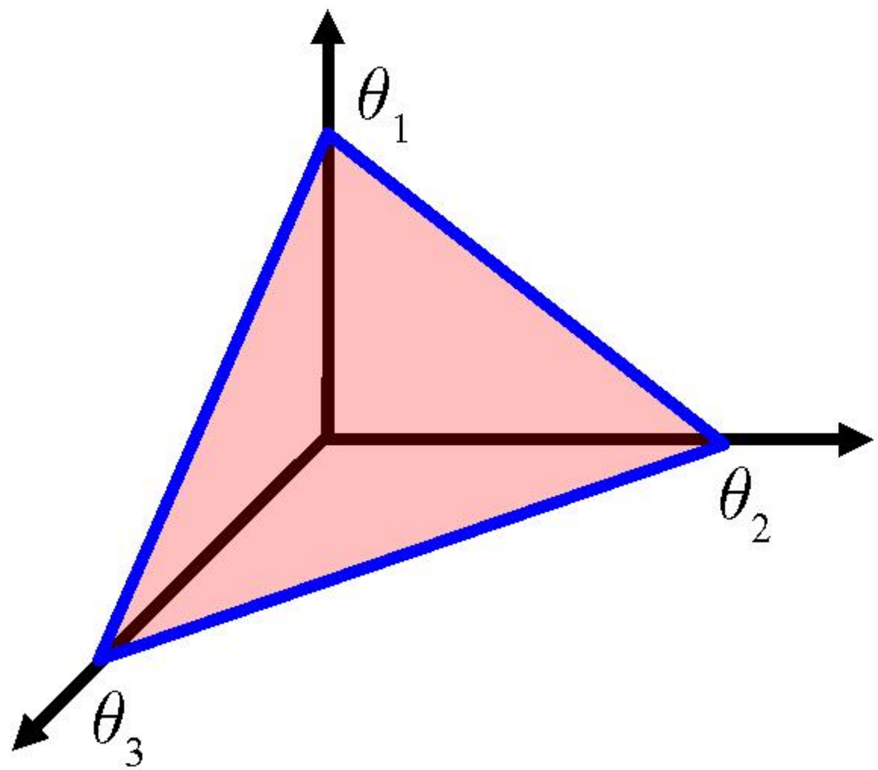
$$P(D|\theta) = \prod_{k=1}^{K} \theta_k^{N_k}$$

- Likelihood is a multivariable function

$$= f(\theta_1, \theta_2, \ldots, \theta_K)$$

3D Probability Simplex

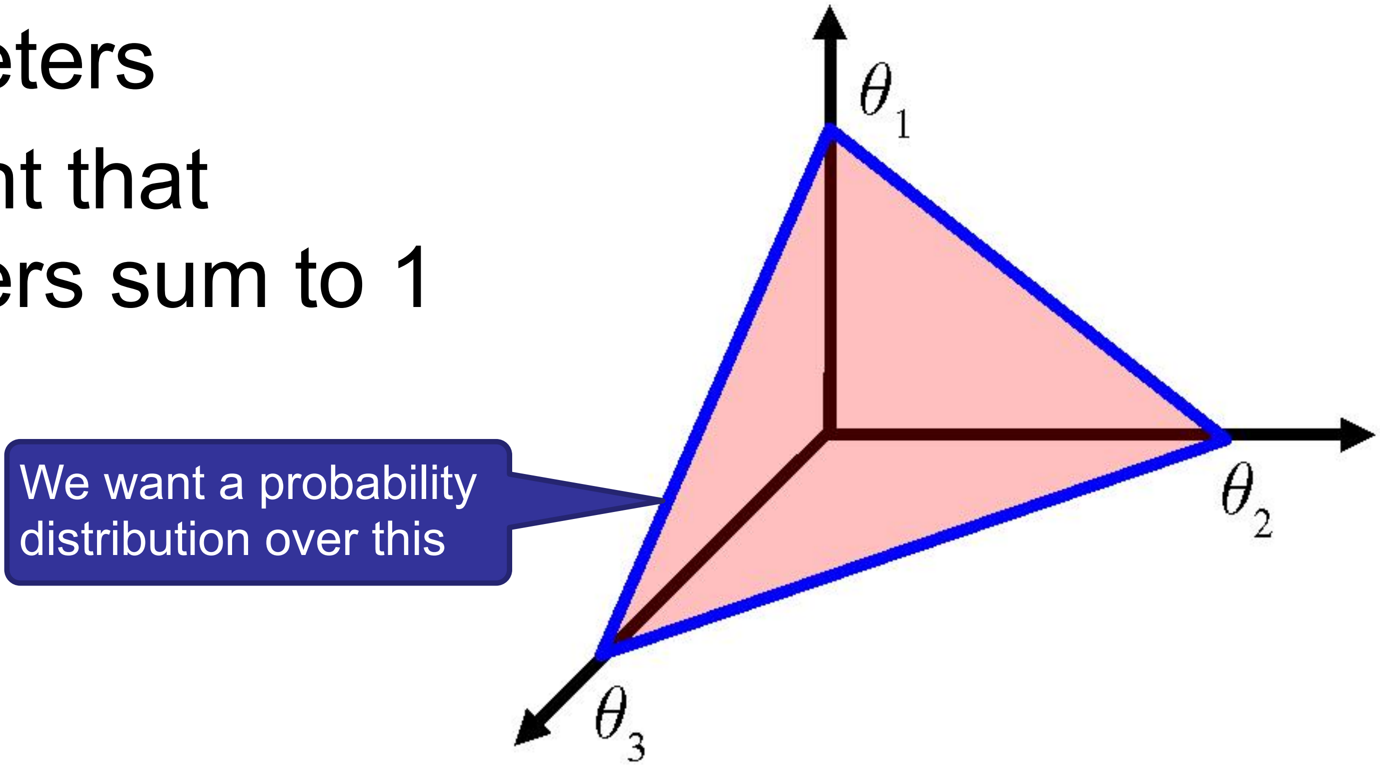
- 3 parameters
- Constraint that parameters sum to 1



$$S_K = \{\theta : 0 \leq \theta_k \leq 1, \sum_{k=1}^{K} \theta_k = 1\}$$

3D Probability Simplex

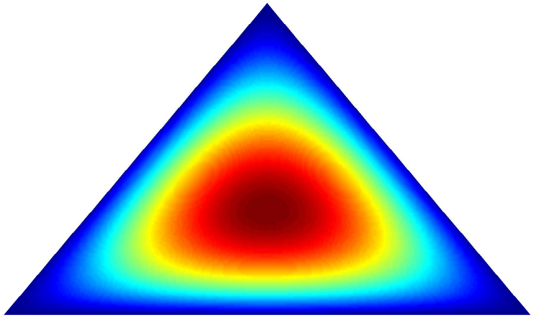
- 3 parameters
- Constraint that parameters sum to 1



$$S_K = \{\theta : 0 \leq \theta_k \leq 1, \sum_{k=1}^{K} \theta_k = 1\}$$

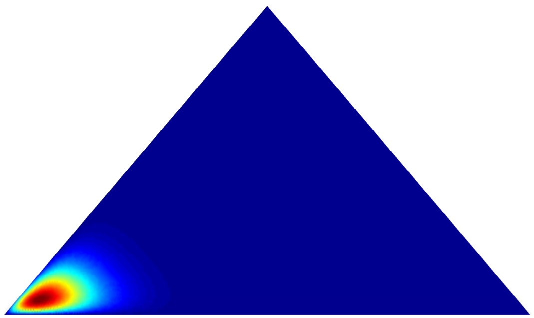
Dirichlet distribution

- Multivariate generalization of Beta distribution
- Conjugate prior to multinomial

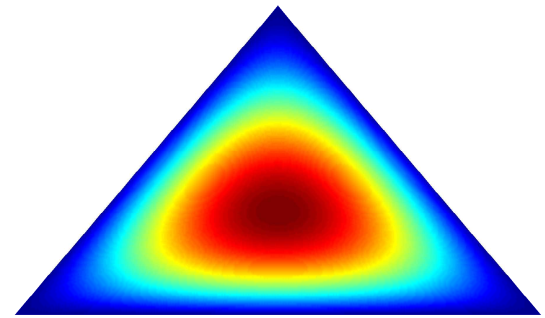


$$\mathrm{Dir}(\theta|\alpha) = \frac{1}{B(\alpha)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1} \mathbb{1}(\theta \in S_K)$$

Dirichlet distribution



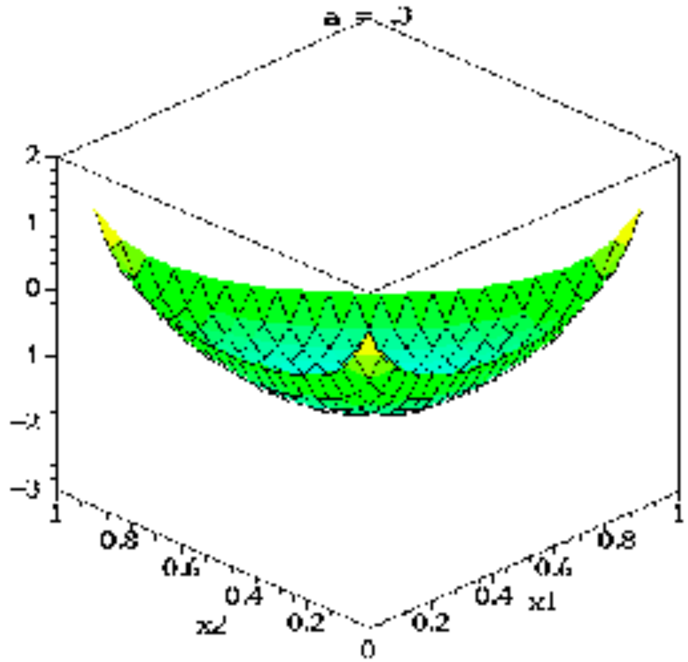
α = <20,2,2>



α = <2,2,2>

$$\mathrm{Dir}(\theta|\alpha) = \frac{1}{B(\alpha)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1} \mathbb{1}(\theta \in S_K)$$

(log) Dirichlet distribution



α = <0.3,0.3,0.3> to <2.0, 2.0, 2.0>

Posterior

$$P(\theta|D) \propto P(D|\theta)P(\theta)$$

Posterior

$$P(\theta|D) \propto P(D|\theta)P(\theta)$$

Posterior

$$P(\theta|D) \propto P(D|\theta)P(\theta)$$

$$\propto \prod_{k=1}^{K} \theta_k^{N_k} \theta_k^{\alpha_k - 1} = \prod_{k=1}^{K} \theta_k^{N_k + \alpha_k - 1}$$

Posterior

$$P(\theta|D) \propto P(D|\theta)P(\theta)$$

$$\propto \prod_{k=1}^{K} \theta_k^{N_k} \theta_k^{\alpha_k - 1} = \prod_{k=1}^{K} \theta_k^{N_k + \alpha_k - 1}$$

$$= \mathrm{Dir}(\theta|\alpha_1 + N_1, \ldots, \alpha_K + N_K)$$

Posterior

$$P(\theta|D) \propto P(D|\theta)P(\theta)$$

$$\propto \prod_{k=1}^{K} \theta_k^{N_k} \theta_k^{\alpha_k - 1} = \prod_{k=1}^{K} \theta_k^{N_k + \alpha_k - 1}$$

$$= \mathrm{Dir}(\theta|\alpha_1 + N_1, \ldots, \alpha_K + N_K)$$

Dirichlet is Conjugate to Multinomial

MAP Point Estimate

$$\theta^{MAP} = \arg\max_{\theta} P(\theta|D)$$

MAP Point Estimate

$$\theta^{MAP} = \arg\max_{\theta} P(\theta|D)$$

$$= \frac{N_k + \alpha_k - 1}{\sum_{k=1}^{K} N_k + \sum_{k=1}^{K} \alpha_k - K}$$

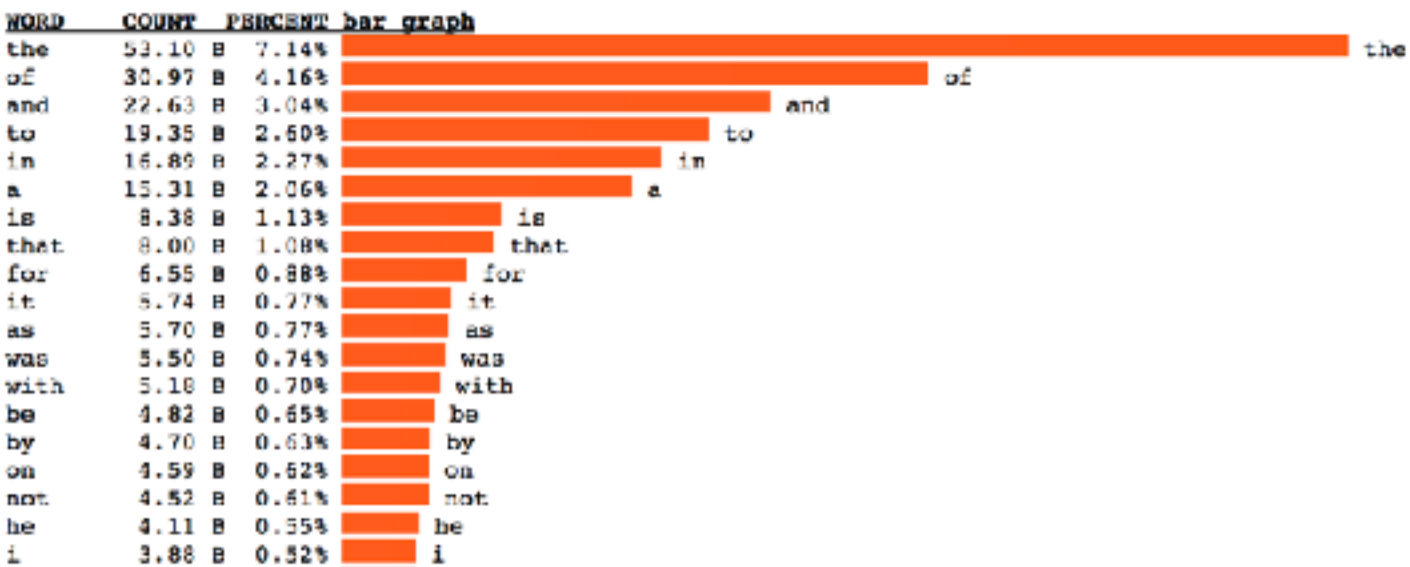
Maximum Likelihood (= uninformative prior)

$$\theta^{MAP} = \arg\max_{\theta} P(D|\theta)$$

$$= \frac{N_k}{\sum_{k=1}^{K} N_k}$$

Parameter Estimation (text)

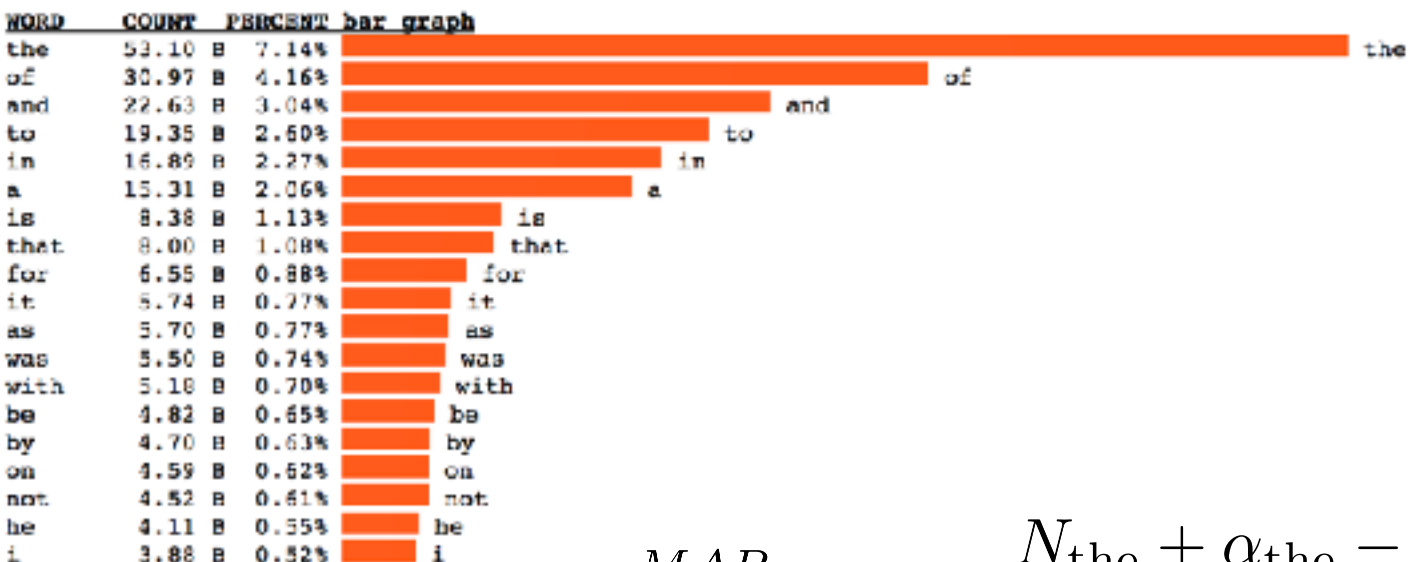
- Count words in Google Books:



WORD	COUNT	PERCENT
the	53.10 B	7.14%
of	30.97 B	4.16%
and	22.63 B	3.04%
to	19.35 B	2.60%
in	16.89 B	2.27%
a	15.31 B	2.06%
is	8.38 B	1.13%
that	8.00 B	1.08%
for	6.55 B	0.88%
it	5.74 B	0.77%
as	5.70 B	0.77%
was	5.50 B	0.74%
with	5.18 B	0.70%
be	4.82 B	0.65%
by	4.70 B	0.63%
on	4.59 B	0.62%
not	4.52 B	0.61%
he	4.11 B	0.55%
i	3.88 B	0.52%

Parameter Estimation (text)

- Count words in Google Books:



$$\theta_{\text{the}}^{MAP} = \frac{N_{\text{the}} + \alpha_{\text{the}} - 1}{\sum_{k=1}^{K} N_k + \sum_{k=1}^{K} \alpha_k - K}$$

Example: Language Modeling

- Q: how do we model the probability of a (text) document?

Example: Language Modeling

- Q: how do we model the probability of a (text) document?
- Assume words are drawn independently (bag of words)

Example: Language Modeling

- Q: how do we model the probability of a (text) document?
- Assume words are drawn independently (bag of words)

$$P(D|\theta) = \prod_{k=1}^{K} \theta_k^{N_k}$$

Example: Language Modeling

- Q: how do we model the probability of a (text) document?
- Assume words are drawn independently (bag of words)

$$P(D|\theta) = \prod_{k=1}^{K} \theta_k^{N_k}$$

Q: What to do about unseen words?

Naïve Bayes Classifier

- Function Approximation:

Generative Model of our Data
(e.g. language model)

$$P(c|X) \propto P(X|c)P(c)$$

Target Output / Class label
(E.g. spam / not spam)

Prior distribution over outputs

Naïve Bayes Classifier

- Function Approximation:

Generative Model of our Data
(e.g. language model)

$$P(c|X) \propto P(X|c)P(c)$$

Target Output / Class label
(E.g. spam / not spam)

This could be any probability distribution (e.g. Gaussian if X is a vector of reals).

Generative Models in General

1. Make up a story about how the data was generated
2. Estimate model parameters from data (or compute sufficient statistics)
3. Apply Bayes' rule to infer a probability distribution over unknown variables on new data.

Naïve Bayes Classifier

- Parameter Estimation

$$\log P(D|\theta) = \sum_{c=1}^{C} N_c \log \pi_c + \sum_{j=1}^{D} \sum_{c=1}^{C} \sum_{i:y_i=c} \log P(x_{ij}|\theta_{jc})$$

Naïve Bayes Classifier

- Parameter Estimation

$$\log P(D|\theta) = \sum_{c=1}^{C} N_c \log \pi_c + \sum_{j=1}^{D} \sum_{c=1}^{C} \sum_{i:y_i=c} \log P(x_{ij}|\theta_{jc})$$

Decomposes, can optimize parameters separately

Naïve Bayes Classifier

- Parameter Estimation

$$P(c) = \pi_c$$

$$\log P(D|\theta) = \sum_{c=1}^{C} N_c \log \pi_c + \sum_{j=1}^{D} \sum_{c=1}^{C} \sum_{i:y_i=c} \log P(x_{ij}|\theta_{jc})$$

Decomposes, can optimize parameters separately

Naïve Bayes Classification: Practical Issues

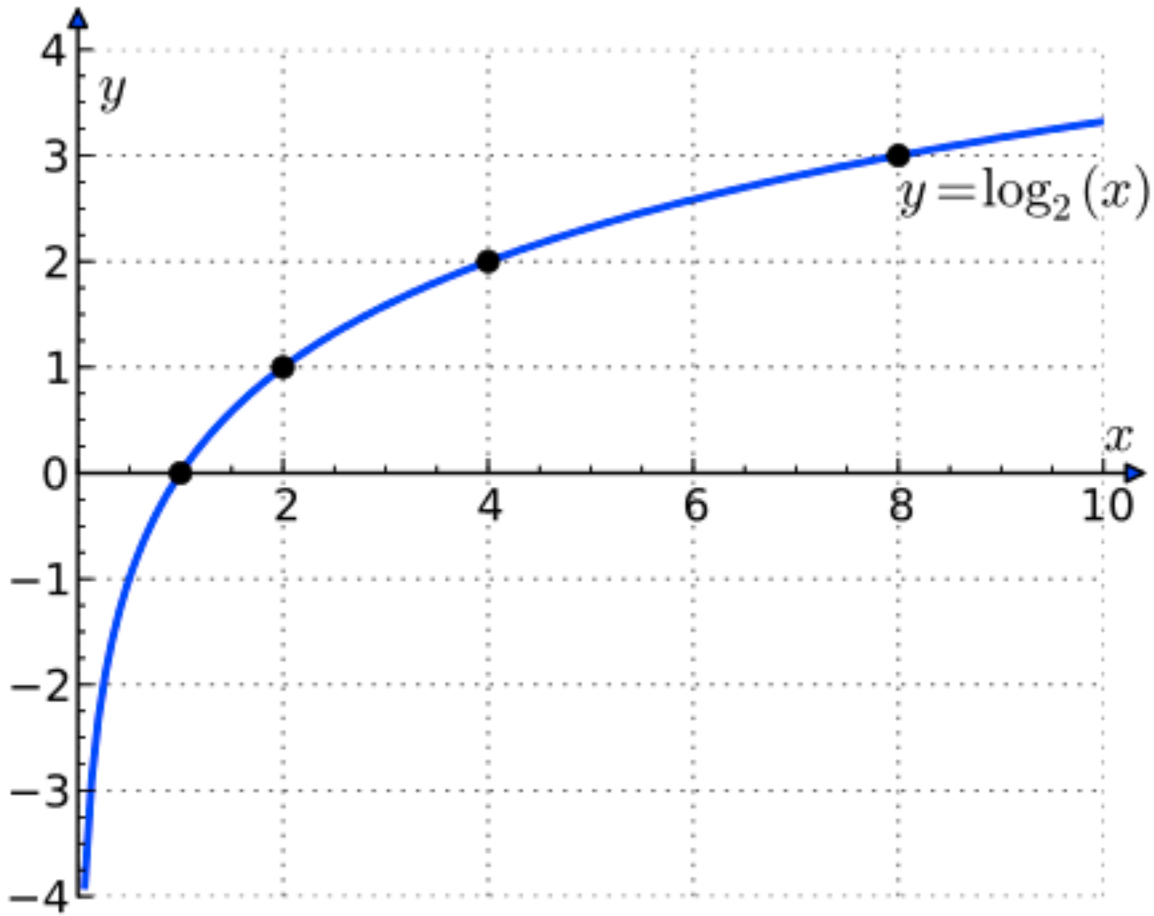
$$
\begin{aligned}
c_{MAP} &= \mathrm{argmax}_c P(c|x_1, \ldots, x_n) \\
&= \mathrm{argmax}_c P(x_1, \ldots, x_n|c)P(c) \\
&= \mathrm{argmax}_c P(c) \prod_{i=1}^{n} P(x_i|c)
\end{aligned}
$$

Naïve Bayes Classification: Practical Issues

$$
\begin{aligned}
c_{MAP} &= \text{argmax}_c P(c|x_1, \ldots, x_n) \\
&= \text{argmax}_c P(x_1, \ldots, x_n|c) P(c) \\
&= \text{argmax}_c P(c) \prod_{i=1}^{n} P(x_i|c)
\end{aligned}
$$

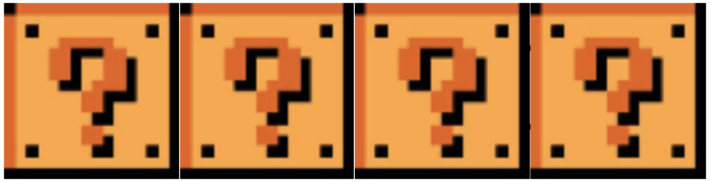
- Multiplying together lots of probabilities
- Probabilities are numbers between 0 and 1
- Q: What could go wrong here?

Working with probabilities in log space



Log Identities (review)

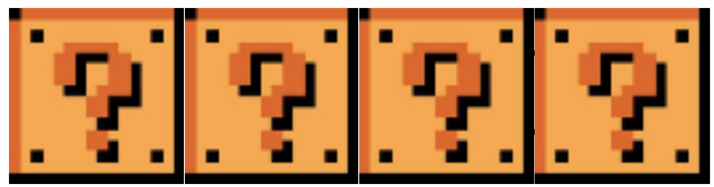
$$\log(a \times b) =$$ 

$$\log(\frac{a}{b}) =$$ 

$$\log(a^n) =$$ 

Log Identities (review)

$$\log(a \times b) = \log(a) + \log(b)$$

$$\log(\frac{a}{b}) =$$ 

$$\log(a^n) =$$ 

Log Identities (review)

$$\log(a \times b) = \log(a) + \log(b)$$

$$\log(\frac{a}{b}) = \log(a) - \log(b)$$

$$\log(a^n) =$$ 

Log Identities (review)

$$\log(a \times b) = \log(a) + \log(b)$$

$$\log(\frac{a}{b}) = \log(a) - \log(b)$$

$$\log(a^n) = n\log(a)$$

Naïve Bayes with Log Probabilities

$$c_{MAP} = \text{argmax}_c P(c|x_1, \ldots, x_n)$$

$$= \text{argmax}_c P(c) \prod_{i=1}^{n} P(x_i|c)$$

$$= \text{argmax}_c \log \left(P(c) \prod_{i=1}^{n} P(x_i|c) \right)$$

$$= \text{argmax}_c \log P(c) + \sum_{i=1}^{n} \log P(x_i|c)$$

Naïve Bayes with Log Probabilities

$$c_{MAP} = \operatorname{argmax}_c \log P(c) + \sum_{i=1}^{n} \log P(x_i|c)$$

Naïve Bayes with Log Probabilities

$$c_{MAP} = \operatorname{argmax}_c \log P(c) + \sum_{i=1}^{n} \log P(x_i|c)$$

- Q: Why don’t we have to worry about floating point underflow anymore?

What if we want to calculate posterior log-probabilities?

$$P(c|x_1,\ldots,x_n) = \frac{P(c)\prod_{i=1}^{n} P(x_i|c)}{\sum_{c'} P(c')\prod_{i=1}^{n} P(x_i|c')}$$

What if we want to calculate posterior log-probabilities?

$$P(c|x_1,\ldots,x_n) = \frac{P(c)\prod_{i=1}^{n} P(x_i|c)}{\sum_{c'} P(c')\prod_{i=1}^{n} P(x_i|c')}$$

$$\log P(c|x_1,\ldots,x_n) = \log \frac{P(c)\prod_{i=1}^{n} P(x_i|c)}{\sum_{c'} P(c')\prod_{i=1}^{n} P(x_i|c')}$$

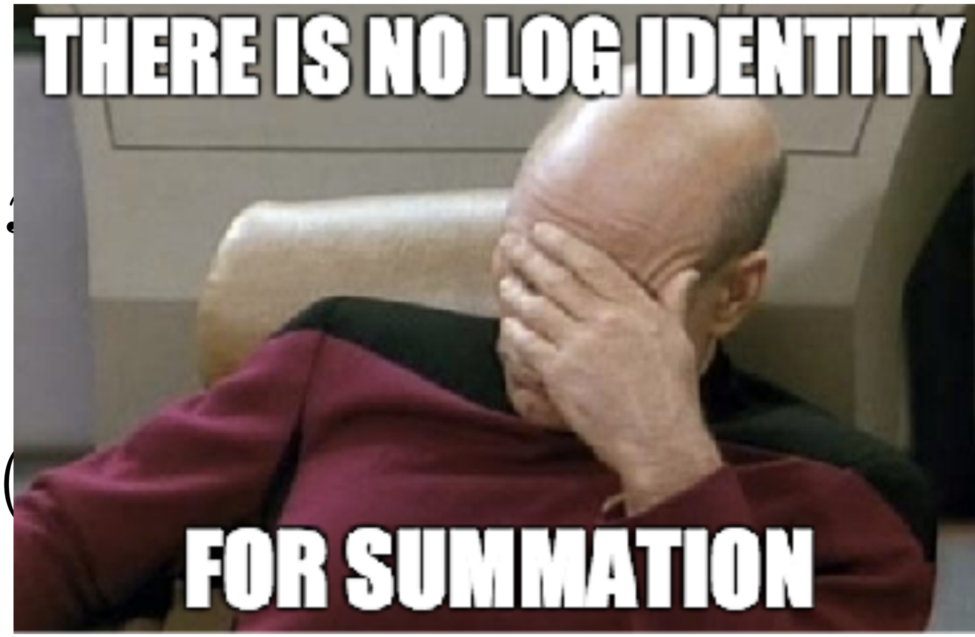
What if we want to calculate posterior log-probabilities?

$$P(c|x_1, \ldots, x_n) = \frac{P(c) \prod_{i=1}^{n} P(x_i|c)}{\sum_{c'} P(c') \prod_{i=1}^{n} P(x_i|c')}$$

$$\log P(c|x_1, \ldots, x_n) = \log \frac{P(c) \prod_{i=1}^{n} P(x_i|c)}{\sum_{c'} P(c') \prod_{i=1}^{n} P(x_i|c')}$$

$$= \log P(c) + \sum_{i=1}^{n} P(x_i|c) - \log \left[\sum_{c'} P(c') \prod_{i=1}^{n} P(x_i|c') \right]$$

What if we want to calculate posterior log-probabilities?



$P(c|$ … $\prod_{i=1}^{n} P(x_i|c)$ / … $\prod_{i=1}^{n} P(x_i|c')$

$\log P($ … $\prod_{i=1}^{n} P(x_i|c)$ / … $c') \prod_{i=1}^{n} P(x_i|c')$

$$= \log P(c) + \sum_{i=1}^{n} P(x_i|c) - \log \left[\sum_{c'} P(c') \prod_{i=1}^{n} P(x_i|c') \right]$$

Log Exp Sum Trick: motivation

- We have: a bunch of log probabilities.
  - log(p1), log(p2), log(p3), … log(pn)
- We want: log(p1 + p2 + p3 + … pn)
- We could convert back from log space, sum then take the log.
  - If the probabilities are very small, this will result in floating point underflow

Log Exp Sum Trick:

$$\log[\sum_i \exp(x_i)] = x_{max} + \log[\sum_i \exp(x_i - x_{max})]$$

Another issue: Smoothing

$$\hat{P}(w_i|c) = \frac{\text{count}(w,c) + 1}{\sum_{w' \in V} \text{count(w',c)} + |V|}$$

Another issue: Smoothing

$$\hat{P}(w_i|c) = \frac{\text{count}(w,c) + \alpha}{\sum_{w' \in V} \text{count}(w',c) + \alpha|V|}$$

Another issue: Smoothing

Alpha doesn't necessarily need to be 1 (hyperparmeter)

$$\hat{P}(w_i|c) = \frac{\text{count}(w,c) + \alpha}{\sum_{w' \in V} \text{count}(w',c) + \alpha|V|}$$

Another issue: Smoothing

Can think of alpha as a "pseudocount".
Imaginary number of times this word has been seen.

$$\hat{P}(w_i|c) = \frac{\text{count}(w,c) + \alpha}{\sum_{w' \in V} \text{count}(w',c) + \alpha|V|}$$

Another issue: Smoothing

$$\hat{P}(w_i|c) = \frac{\text{count}(w,c) + \alpha}{\sum_{w' \in V} \text{count}(w',c) + \alpha|V|}$$

Another issue: Smoothing

$$\hat{P}(w_i|c) = \frac{\text{count}(w,c) + \alpha}{\sum_{w' \in V} \text{count}(w',c) + \alpha|V|}$$

- Q: What if alpha = 0?

Another issue: Smoothing

$$\hat{P}(w_i|c) = \frac{\text{count}(w,c) + \alpha}{\sum_{w' \in V} \text{count}(w',c) + \alpha|V|}$$

- Q: What if alpha = 0?
- Q: what if alpha = 0.000001?

Another issue: Smoothing

$$\hat{P}(w_i|c) = \frac{\text{count}(w,c) + \alpha}{\sum_{w' \in V} \text{count}(w',c) + \alpha|V|}$$

- Q: What if alpha = 0?
- Q: what if alpha = 0.000001?
- Q: what happens as alpha gets very large?

Overfitting

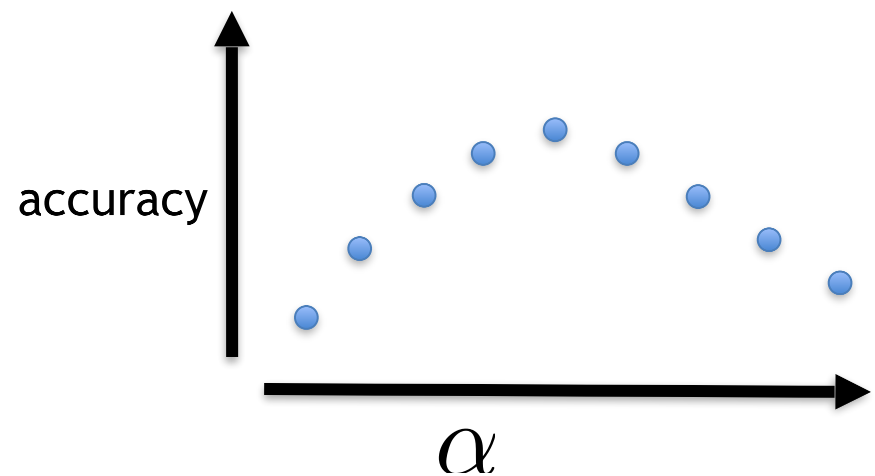
- Model cares too much about the training data
- How to check for overfitting?
  - Training vs. test accuracy
- Pseudocount parameter combats overfitting

Q: how to pick Alpha?

- Split train vs. Test
- Try a bunch of different values
- Pick the value of alpha that performs best
- What values to try? Grid search
  - (10^-2,10^-1,...,10^2)

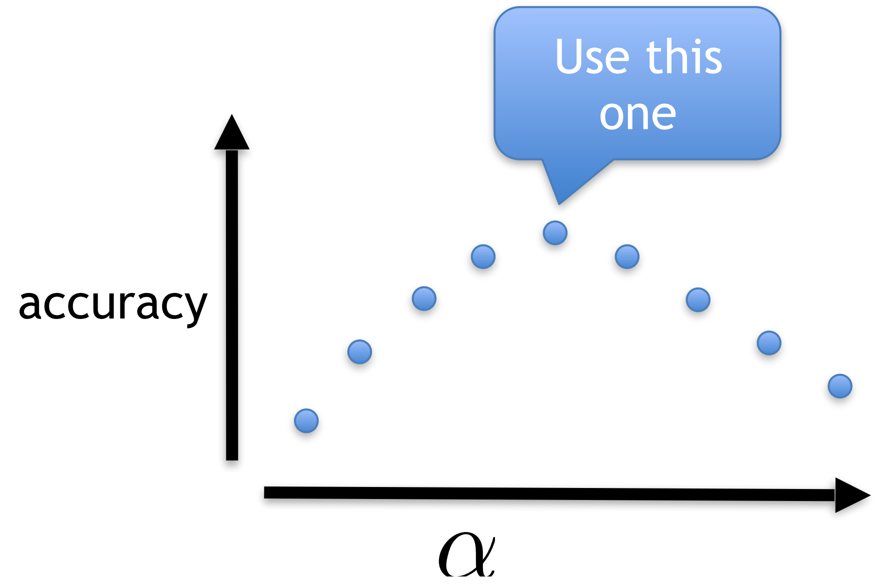
Q: how to pick Alpha?

- Split train vs. Test
- Try a bunch of different values
- Pick the value of alpha that performs best
- What values to try? Grid search
  - (10^-2,10^-1,…,10^2)



Q: how to pick Alpha?

- Split train vs. Test
- Try a bunch of different values
- Pick the value of alpha that performs best
- What values to try? Grid search
  - (10^-2,10^-1,…,10^2)



Data Splitting

- Train vs. Test

- Better:
  - Train (used for fitting model **parameters**)
  - Dev (used for tuning **hyperparameters**)
  - Test (reserve for final evaluation)

- Cross-validation