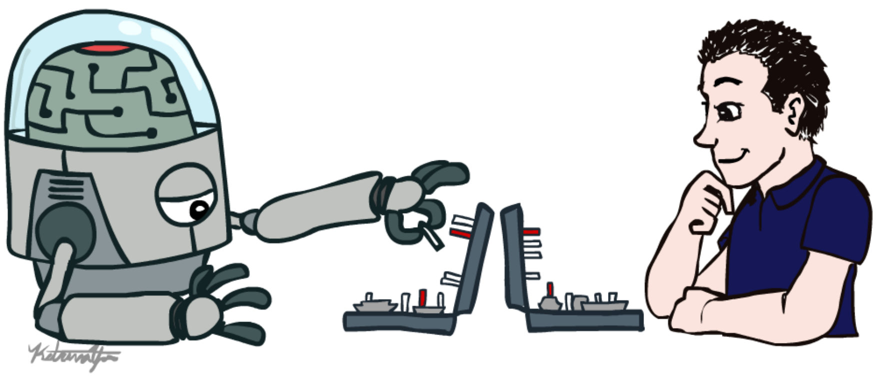
CSE 3521:
Introduction to Artificial Intelligence



[Many slides are adapted from the UC Berkeley. CS188 Intro to AI at UC Berkeley and previous CSE 3521 course at OSU.]

THE OHIO STATE UNIVERSITY

Parameter Estimation

- How to estimate parameters from data?

Maximum Likelihood Principle:

Choose the parameters that maximize the probability of the observed data!

Maximum Likelihood Estimation Recipe

1. Use the log-likelihood
2. Differentiate with respect to the parameters
3. Equate to zero and solve

An Example

- Let's start with the simplest possible case
  - Single observed variable
  - Flipping a bent coin

- We Observe:
  - Sequence of heads or tails
  - HTTTTTHTHT
- Goal:
  - Estimate the probability that the next flip comes up heads

Assumptions

- Fixed parameter θ_H
  - Probability that a flip comes up heads
- Each flip is independent
  - Doesn't affect the outcome of other flips
- (IID) Independent and Identically Distributed

Example

- Let's assume we observe the sequence:
  - HTTTTTHTHT
- What is the **best** value of θ_H?
  - Probability of heads
- Intuition: should be 0.3 (3 out of 10)
- Question: how do we justify this?

Maximum Likelihood Principle

- The value of θ_H which maximizes the probability of the observed data is best!
- Based on our assumptions, the probability of "HTTTTTHTHT" is:

$$P(x_1 = H, x_2 = T, \dots, x_m = T; \theta_H)$$

$$= P(x_1 = H; \theta_H) P(x_2 = T; \theta_H), \dots P(x_m = T; \theta_H)$$

$$= \theta_H \times (1 - \theta_H), \times \dots \times \theta_H$$

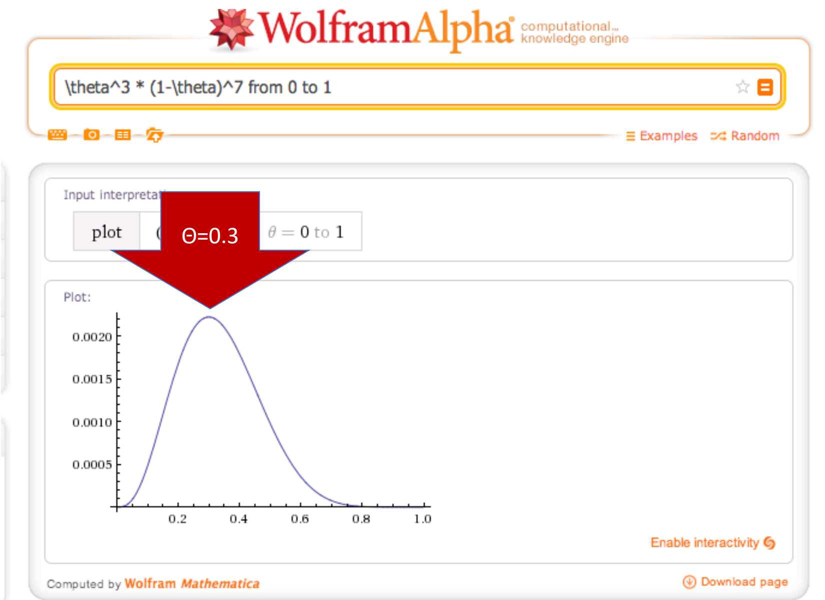
$$= \theta_H^3 \times (1 - \theta_H)^7$$

This is the Likelihood Function

Maximum Likelihood Principle

- Probability of "HTTTTTHTHT" as a function of θ_H

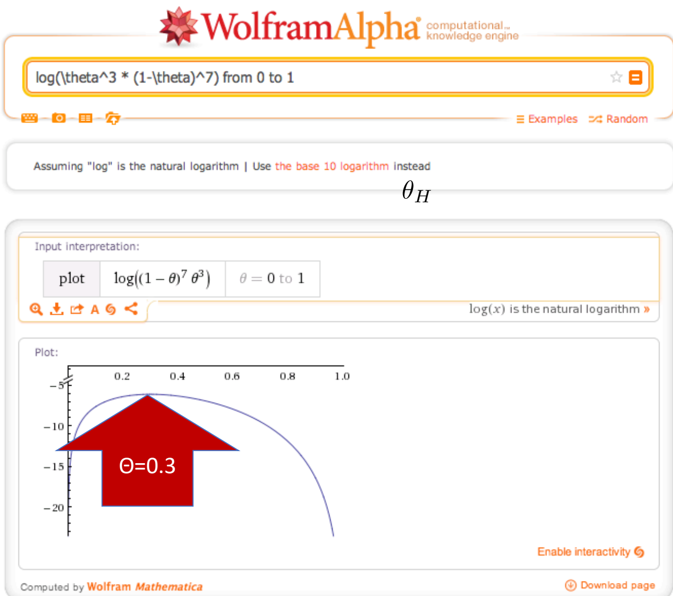
$$\theta_H^3 \times (1-\theta_H)^7$$



Maximum Likelihood Principle

- Probability of “HTTTTTHTHT” as a function of θ_H

$$\log(\theta_H^3 \times (1-\theta_H)^7)$$



Maximum Likelihood value of θ_H

$$\frac{\partial}{\partial \theta_H} \boxed{\log(\theta_H^{\#H}(1-\theta_H)^{\#T})} = 0$$

$$\frac{\partial}{\partial \theta_H} \boxed{\log(\theta_H^{\#H}) + \log((1-\theta_H)^{\#T})} = 0$$

Log Identities

$$\frac{\partial}{\partial \theta_H} \#H \log(\theta_H) + \#T \log(1-\theta_H) = 0$$

Maximum Likelihood value of θ_H

$$\frac{\partial}{\partial \theta_H} \#H \log(\theta_H) + \#T \log(1 - \theta_H) = 0$$

$$\frac{\#H}{\theta_H} - \frac{\#T}{1 - \theta_H} = 0$$

$$\vdots$$

$$\hat{\theta} = \frac{\#H}{\#H + \#T}$$

The problem with Maximum Likelihood

- What if the coin doesn't look very bent?
  - Should be somewhere around 0.5?
- What if we saw 3,000 heads and 7,000 tails?
  - Should this really be the same as 3 out of 10?
- Maximum Likelihood
  - No way to quantify our **uncertainty**.
  - No way to incorporate our prior knowledge!

Q: how to deal with this problem?

Bayesian Parameter Estimation

- Let's just treat θ_H like any other variable
- Put a prior on it!
  - Encode our prior knowledge about possible values of θ_H using a probability distribution
- Now consider two probability distributions:

$$P(x_i|\theta_H) = \begin{cases} \theta_H, & \text{if } x_i = H \\ 1 - \theta_H, & \text{otherwise} \end{cases}$$

Posterior Over θ_H

$$P(\theta | x_1 = H, x_2 = T, \ldots, x_m = T)$$

$$= \frac{P(x_1 = H, x_2 = T, \ldots, x_m = T | \theta) P(\theta)}{P(x_1 = H, x_2 = T, \ldots, x_m = T)}$$

$$= \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

How can we encode prior knowledge?

- Example: The coin doesn't look very bent
  - Assign higher probability to values of θ_H near 0.5
- Solution: The **Beta Distribution**

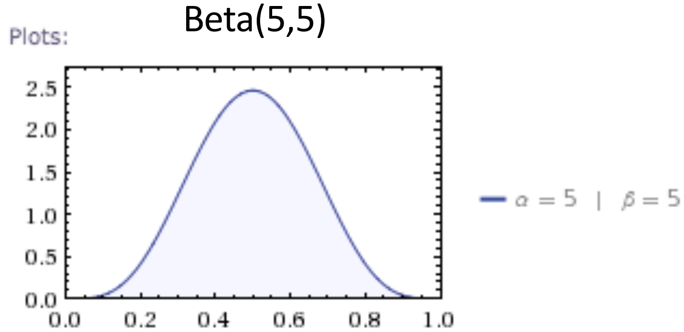
$$P(\theta_H|\alpha,\beta) = \frac{1}{B(\alpha,\beta)}\theta_H^{\alpha-1}(1-\theta_H)^{\beta-1}$$

$$B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$$

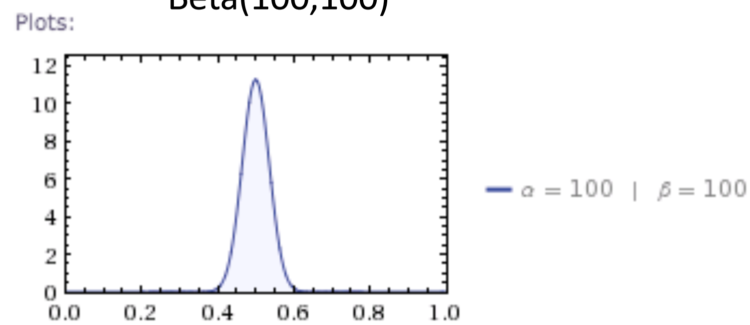
Gamma is a continuous generalization of the Factorial Function

Beta Distribution

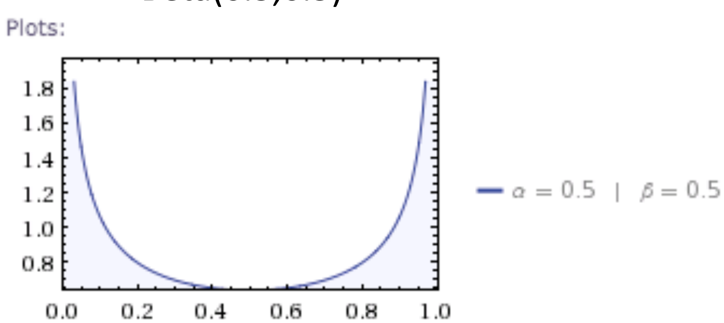
Beta(5,5)



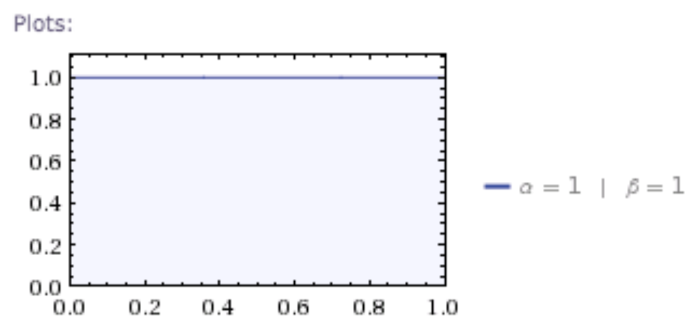
Beta(100,100)



Beta(0.5,0.5)



Beta(1,1)



Marginal Probability over single Toss

$$
\begin{aligned}
P(x_1 = H|\alpha, \beta) \\
&= \int P(x_1 = H|\theta_H)P(\theta_H|\alpha, \beta)d\theta_H \\
&= \int \theta P(\theta_H|\alpha, \beta)d\theta_H \\
&\vdots \\
&= \frac{\alpha}{\alpha + \beta}
\end{aligned}
$$

Beta prior indicates α imaginary heads and β imaginary tails

More than one toss

$$P(\theta_H|x_1,\ldots,x_m) \propto P(x_1,\ldots,x_m|\theta)P(\theta|\alpha,\beta)$$

$$\propto \theta_H^{\#H}(1-\theta_H)^{\#T}\theta_H^{\alpha-1}(1-\theta_H)^{\beta-1}$$

$$= \theta_H^{\#H+\alpha-1}(1-\theta_H)^{\#T+\beta-1}$$

$$= Beta(\#H+\alpha, \#T+\beta)$$

- If the prior is Beta, so is posterior!
- Beta is **conjugate** to the Bernoulli likelihood

Prediction

- Immediate result
  - Can compute the probability over the next toss:

$$P(x_{m+1}|x_1,\ldots,x_m) = \frac{\alpha + \#H}{\alpha + \#H + \beta + \#T}$$

Summary: Maximum Likelihood vs. Bayesian Estimation

- Maximum likelihood: find the "best" $\hat{\theta}_H$
- Bayesian approach:
  - Don't use a point estimate
  - Keep track of our beliefs about θ_H
  - Treat θ_H like a random variable

Modeling Text

- Not a sequence of coin tosses…
- Instead we have a sequence of words
- But we could think of this as a sequence of die rolls
    - Very large die with one word on each side
- **Multinomial** is n-dimensional generalization of Bernoulli
- **Dirichlet** is an n-dimensional generalization of Beta distribution

Multinomial

- Rather than one parameter, we have a vector

$$\theta =< \theta_1, \ldots, \theta_V > \qquad s.t. \sum_i \theta_i = 1$$

- Likelihood Function:

$$P(w_1 = \text{"the"}, \ldots, w_n = \text{"dog"} | \theta) = \prod_{i=1}^{K} \theta_i^{\#i}$$

Dirichlet

- Generalizes the Beta distribution from 2 to K dimensions
- Conjugate to Multinomial

$$P(\vec{\theta}|\vec{\alpha}) = Dir(\vec{\theta}|\vec{\alpha})$$

$$= \frac{1}{\Delta(\vec{\alpha})} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}$$

$$\Delta(\vec{\alpha}) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^{K} \alpha_k)}$$

Example: Text Classification

- Problem: Spam Email classification
    - We have a bunch of email (e.g. 10,000 emails) labeled as spam and non-spam
    - Goal: given a new email, predict whether it is spam or not
    - How can we tell the difference?
        - Look at the words in the emails
        - Viagra, ATTENTION, free

$$P(\text{Spam} = \text{true} | \text{Free}, \text{viagra}, \text{act}, \text{now!}) = ?$$

Naïve Bayes Text Classifier

$$P(\text{Spam} = \text{true} | \text{Free, viagra, act, now!})$$

$$= \alpha P(\text{Free, viagra, act, now!} | \text{Spam} = \text{true}) P(\text{Spam} = \text{true})$$

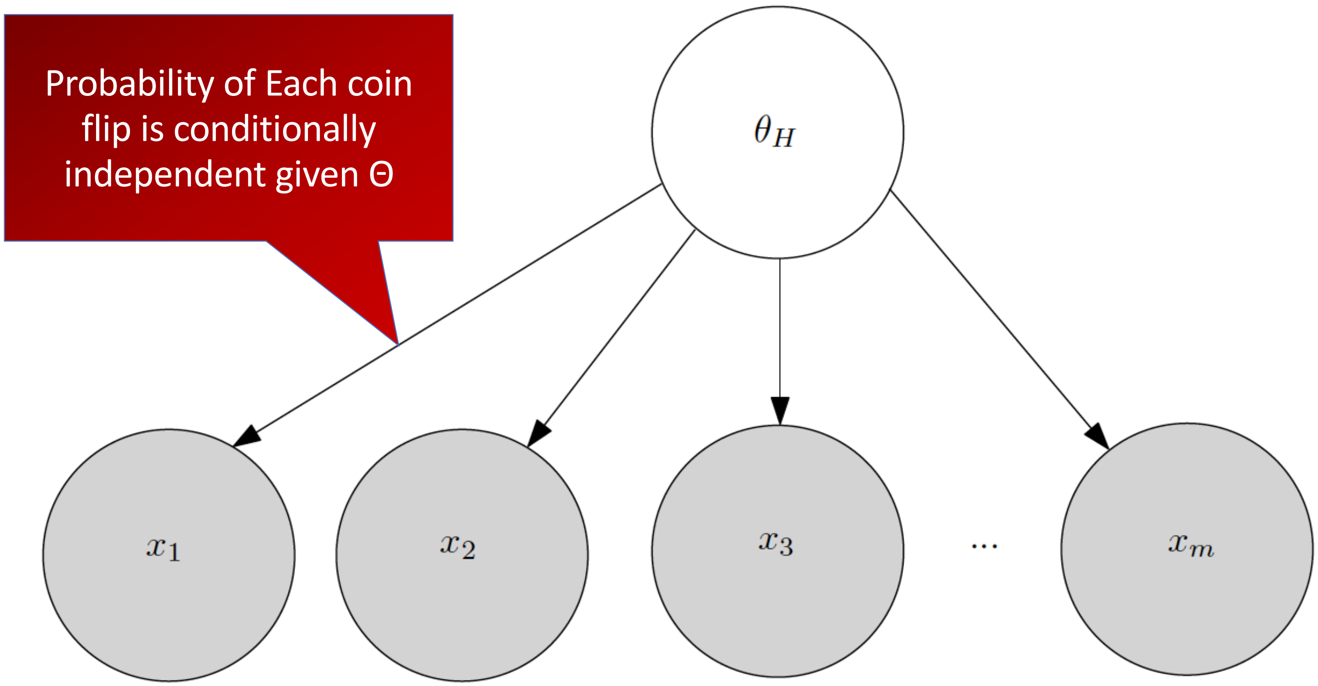
$$= \frac{P(\text{Free}|\text{spam}) P(\text{viagra}|\text{spam}) \ldots P(\text{spam})}{P(\text{free,viagra,act,now})}$$

By making independence assumptions we can better estimate these probabilities from data

Naïve Bayes Text Classifier

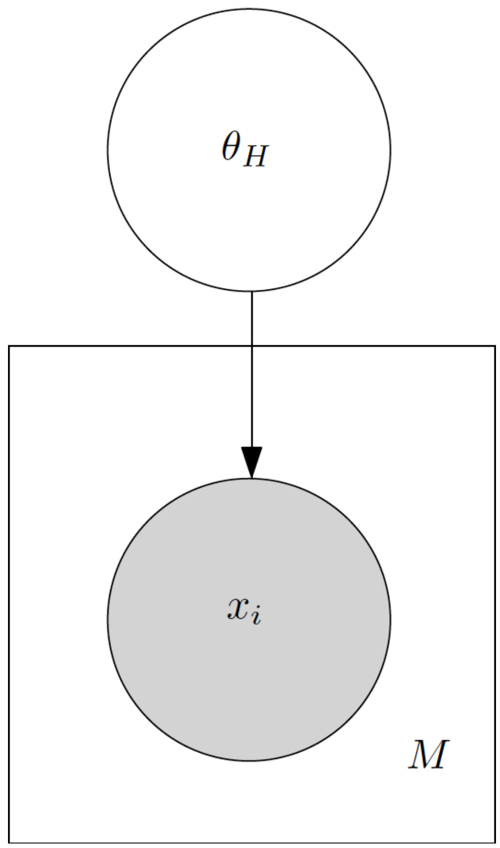
- Simplest possible classifier
- Assumption: probability of each word is conditionally independent given class memberships.
- Simple application of Bayes Rule

Bent Coin Bayesian Network



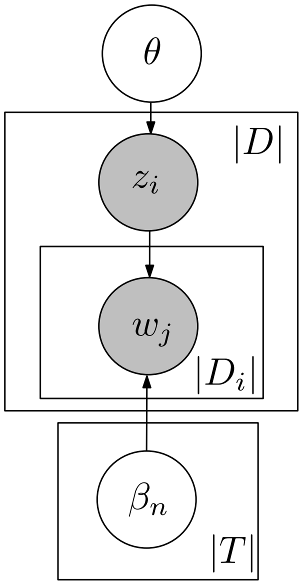
$$\boxed{P(x_1, x_2, \ldots, x_m | \theta_H) = P(x_1|\theta_H) P(x_2|\theta_H) \ldots P(x_m|\theta_H)}$$

Bent Coin Bayesian Network (Plate Notation)



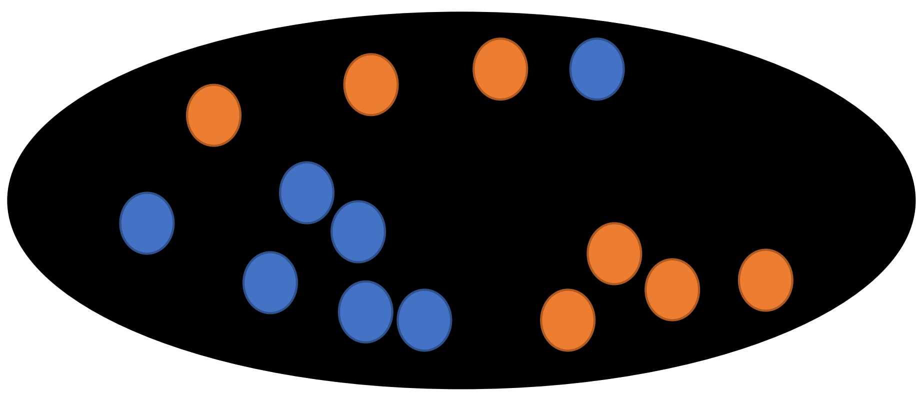
Naïve Bayes Model For Text Classification

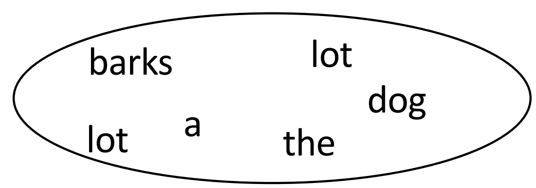
- Data is a set of “documents”
- Z variables are categories
- Z’s Observed during learning
- Hidden at test time.
- Learning from training data:
  - Estimate parameters (θ,β) using fully-observed data
- Prediction on test data:
  - Compute P(Z|w1,…wn) using Bayes’ rule



BOW Example (Q-A take home)

- 2 boxes, 1 box full of blue balls and the other with red balls
- RRBRRRRBBBB





- The dog barks a lot lot