

PRIORS

David Kauchak  
CS159 Fall 2014

## Admin

Assignment 7 due Friday at 5pm

Project proposals due Thursday at 11:59pm

Grading update

## Maximum likelihood estimation

Intuitive

Sets the probabilities so as to maximize the probability of the training data

### Problems?

- Overfitting!
- Amount of data
  - particularly problematic for rare events
- Is our training data representative

## 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  $p(\text{feature}, \text{label})$ ?

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

How do we deal with overfitting?

## Priors

Coin1 data: 3 Heads and 1 Tail  
 Coin2 data: 30 Heads and 10 tails  
 Coin3 data: 2 Tails  
 Coin4 data: 497 Heads and 503 tails

If someone asked you what the probability of heads was for each of these coins, what would you say?

## Training revisited

From a probability standpoint, MLE training is selecting the  $\Theta$  that maximizes:

$$p(\theta \mid data)$$

i.e.

$$\operatorname{argmax}_{\theta} p(\theta \mid data)$$

We pick the most likely model parameters given the data

## Estimating revisited

We can incorporate a prior belief in what the probabilities might be!

To do this, we need to break down our probability

$$p(\theta \mid data) = ?$$

(Hint: Bayes rule)

## Estimating revisited

What are each of these probabilities?

$$p(\theta \mid data) = \frac{p(data \mid \theta)p(\theta)}{p(data)}$$

### Priors

likelihood of the data under the model

probability of different parameters, call the **prior**

$$p(\theta | data) = \frac{p(data | \theta)p(\theta)}{p(data)}$$

probability of seeing the data (regardless of model)

### Priors

$$\theta = \operatorname{argmax}_{\theta} \frac{p(data | \theta)p(\theta)}{p(data)}$$

Does  $p(data)$  matter for the argmax?

### Priors

likelihood of the data under the model

probability of different parameters, call the **prior**

$$\theta = \operatorname{argmax}_{\theta} p(data | \theta)p(\theta)$$

What does MLE assume for a prior on the model parameters?

### Priors

likelihood of the data under the model

probability of different parameters, call the **prior**

$$\theta = \operatorname{argmax}_{\theta} p(data | \theta)p(\theta)$$

- Assumes a **uniform prior**, i.e. all  $\Theta$  are equally likely!
- Relies solely on the likelihood

### A better approach

$$\theta = \operatorname{argmax}_{\theta} p(\text{data} | \theta) p(\theta)$$

$\swarrow$   $\nwarrow$

$\text{likelihood}(\text{data}) = \prod_{i=1}^n p_{\theta}(x_i)$

We can use any distribution we'd like  
 This allows us to impart addition **bias**  
 into the model

### Another view on the prior

Remember, the max is the same if we take the log:

$$\theta = \operatorname{argmax}_{\theta} \log(p(\text{data} | \theta)) + \log(p(\theta))$$

$\swarrow$   $\nwarrow$

$\text{log-likelihood} = \sum_{i=1}^n \log(p(x_i))$

We can use any distribution we'd like  
 This allows us to impart addition **bias**  
 into the model

### Prior for NB

$$\theta = \operatorname{argmax}_{\theta} \log(p(\text{data} | \theta)) + \log(p(\theta))$$

Uniform prior

Dirichlet prior

$\lambda = 0$  → increasing

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

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

### Prior: another view

$$p(\text{features, label}) = p(\text{label}) \prod_{j=1}^m \theta_j^{c_j}$$

$$\text{MLE: } p(w_j | y) = \theta_j = \frac{\text{count}(w_j, y)}{\sum_{k=1}^m \text{count}(w_k, y)}$$

What happens to our likelihood if, for one of the labels, we never saw a particular feature?

Goes to 0!

## Prior: another view

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



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

Adding a prior avoids this!

## Smoothing

training data

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



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

for each label, pretend like we've seen each feature/word occur in  $\lambda$  additional examples

Sometimes this is also called **smoothing** because it is seen as smoothing or interpolating between the MLE and some other distribution