

**CG136:  
Introduction to Computational Linguistics**

**Mark Johnson  
Brown University**

**February 2006**

# Talk overview

---

- Multinomial models

# Language models

---

A *language model* estimates the probability  $P(\mathbf{w})$  of a sentence or text

$$\mathbf{w} = (w_1, \dots, w_m)$$

$$P(w_1, \dots, w_m) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \dots P(w_m|w_1, \dots, w_{m-1})$$

Strictly speaking we also have to predict the *length  $m$  of the sentence*.

We can either:

- model  $P(m)$  directly, or
- add dummy non-word elements (end-markers) to short sentences to make all sentences the same length (works well with  $n > 1$ -gram models)

# *n*-gram language models

---

Repeated conditionalization gives us:

$$P(w_1, \dots, w_m) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \dots P(w_m|w_1, \dots, w_{m-1})$$

An *n*-gram language model assumes that  $w_i$  is *conditionally independent* of  $w_1, \dots, w_{i-n}$  given  $w_{i-n+1}, \dots, w_{i-1}$

$$P(w_i|w_1, \dots, w_{i-1}) = P(w_i|w_{i-n+1}, \dots, w_{i-1})$$

and is *homogeneous* or *time-invariant*, i.e., for all  $i, j$

$$P(w_i|w_{i-n+1}, \dots, w_{i-1}) = P(w_j|w_{j-n+1}, \dots, w_{j-1})$$

This means that an *n*-gram distribution is *completely characterized* by  $P(w_k|w_1, \dots, w_{k-1})$  for  $k = 1, \dots, n$

# Multinomial distributions

---

- A *multinomial distribution* is one where there are  $n$  independent trials, each with  $m$  outcomes.
- The rolls from an unfair die form a multinomial distribution
- A “*monkeys typing randomly*” model of English assumes that texts are generated by a multinomial distribution over English words
- IBM Model 0 modeled the French translations of a single English word by a multinomial distribution (so we had one multinomial distribution for each English word)
- $P(w_n | w_1, \dots, w_{n-1})$  in an  $n$ -gram model is a *conditional multinomial* (i.e., a different multinomial for each context  $w_1, \dots, w_{n-1}$ )

# Estimating multinomials

---

- If the probability of outcome  $k$  on a single trial is  $\theta_k$ , the probability of getting outcomes of  $(n_1, \dots, n_m)$  out of  $n = \sum_{j=1}^m n_j$  trials (where  $n_k$  is the number of times outcome  $k$  appeared) is:

$$P(n_1, \dots, n_m) = \left( \frac{n!}{\prod_{k=1}^m n_k!} \right) \prod_{k=1}^m \theta_k^{n_k}$$

- Given data  $(n_1, \dots, n_m)$ , the MLE  $\hat{\theta}$  is:

$$\begin{aligned} \hat{\theta} &= \operatorname{argmax}_{\theta} P(n_1, \dots, n_m) \\ &= \operatorname{argmax}_{\theta} \prod_{k=1}^m \theta_k^{n_k} \\ &= (n_1/n, \dots, n_m/n) \end{aligned}$$

# MLE estimates of $n$ -gram models

---

Obvious (but bad) method: estimate a separate multinomial for  $P(w_n|w_1, \dots, w_{n-1})$  for each context  $w_1, \dots, w_{n-1}$

$$P(w_n|w_1, \dots, w_{n-1}) = \theta_{w_n|w_1, \dots, w_{n-1}}.$$

Then if  $n_{w_1, \dots, w_k}$  is the number of times the  $k$ -gram  $w_1, \dots, w_k$  occurs in our training data, the MLE is:

$$\hat{\theta}_{w_n|w_1, \dots, w_{n-1}} = \frac{n_{w_1, \dots, w_n}}{n_{w_1, \dots, w_{n-1}}}$$

- What is  $\hat{\theta}_{w_n|w_1, \dots, w_{n-1}}$  when  $n_{w_1, \dots, w_n} = 0$ ?
- What is  $\hat{\theta}_{w_n|w_1, \dots, w_{n-1}}$  when  $n_{w_1, \dots, w_{n-1}} = 0$ ?

# Smoothing and backoff

---

- Because of Zipfian distribution of  $n$ -grams, sparse data is a real issue!
- *Smoothing*: take probability mass from seen events and give it to unseen events

$$\theta_k^* = \frac{n_k + 1}{n + m} \quad (n \text{ tokens, } m \text{ types})$$

- *Backoff*: approximate a complex distribution with a simpler one

$$\theta_{w_n|w_1,\dots,w_{n-1}}^* = \hat{\theta}_{w_{n-1}|w_1,\dots,w_{n-2}} \quad \text{if } n_{w_1,\dots,w_{n-1}} < 5$$

- In practice some combination of both are usually done

# Bayesian estimation

---

- Given some data  $D$  we wish to estimate model parameters  $\theta$
- **Bayes rule:**

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}, \quad \text{where:}$$

- $P(\theta|D)$  is the *posterior* distribution over  $\theta$
- $P(\theta)$  is the *prior* distribution over  $\theta$
- $P(D|\theta)$  is the *likelihood* of the data  $D$  given  $\theta$
- $P(D) = \int P(D|\theta)P(\theta)d\theta$  is a normalization constant required to ensure that  $\int P(\theta|D)d\theta = 1$

# Dirichlet distributions

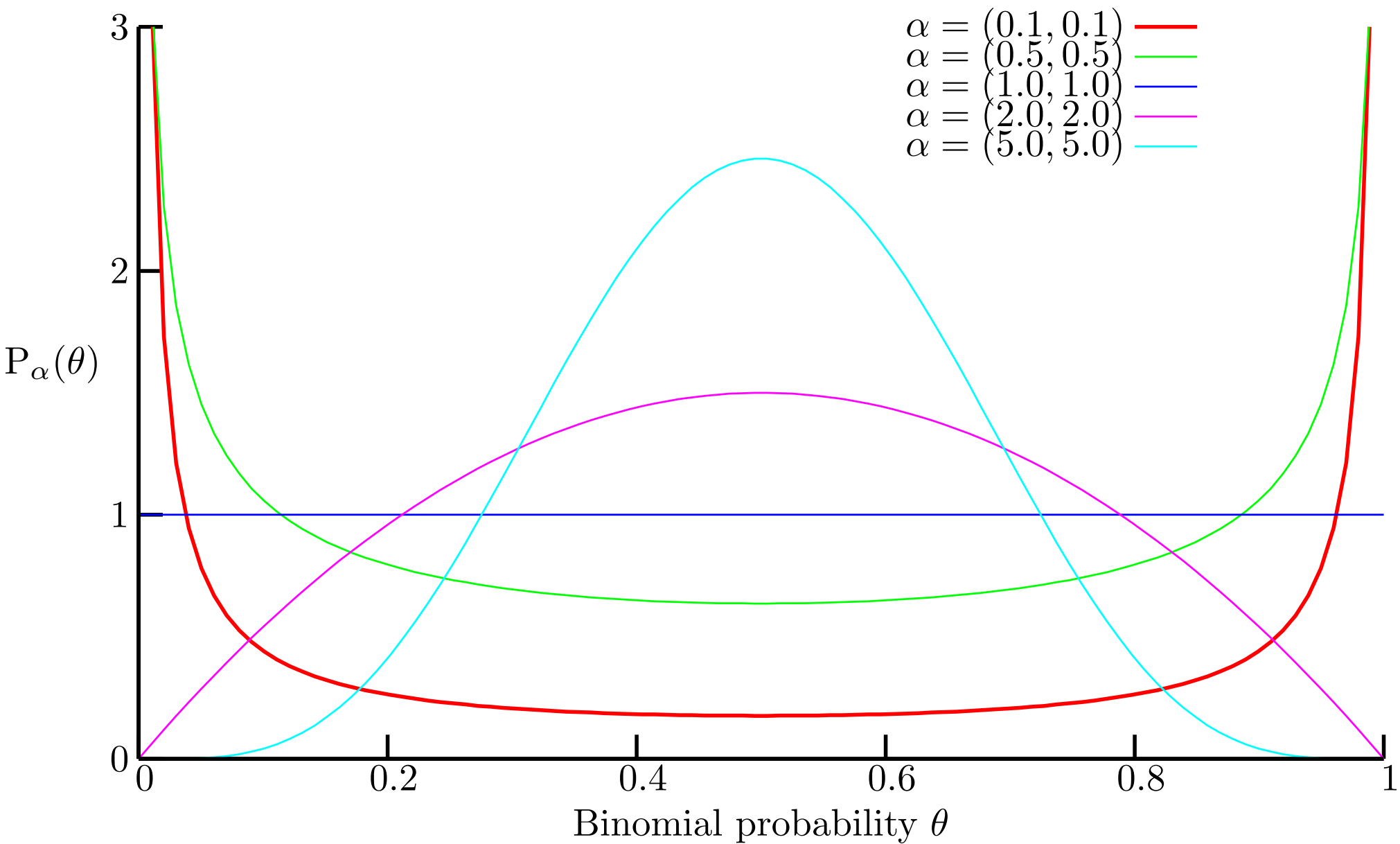
---

- To perform Bayesian inference we need *priors* over models
- *Dirichlet distributions* are probability distributions over multinomial models  $\theta = (\theta_1, \dots, \theta_m)$
- A Dirichlet distribution is specified by parameters  $\alpha = (\alpha_1, \dots, \alpha_m)$ , where each  $\alpha_k > 0$ .

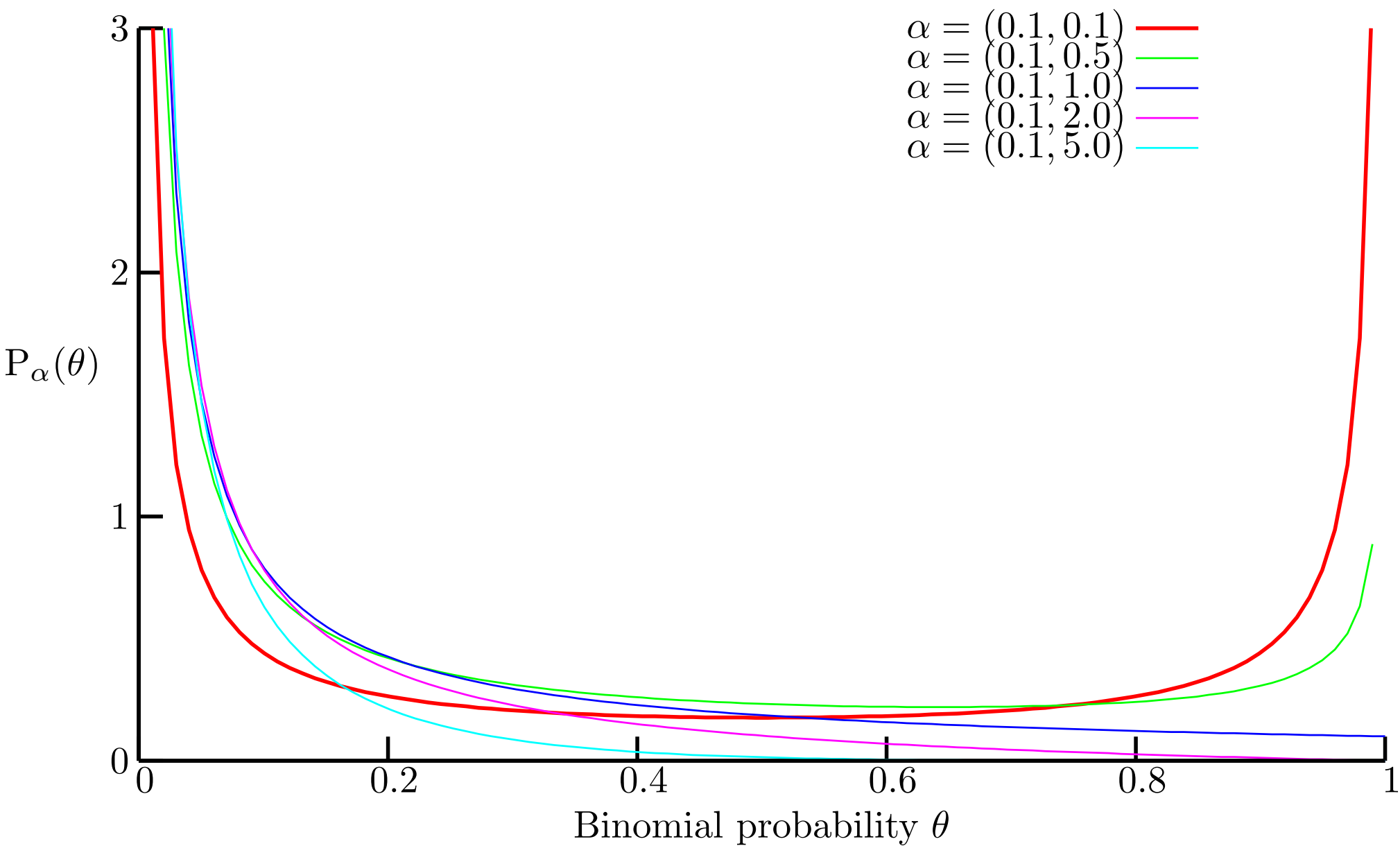
$$P(\theta) = \frac{\Gamma(\sum_{k=1}^m \alpha_k)}{\prod_{k=1}^m \Gamma(\alpha_k)} \prod_{k=1}^m \theta_k^{\alpha_k - 1}$$

- $\Gamma$  generalizes the factorial to non-integer arguments. If  $i$  is a positive integer, then  $\Gamma(i) = (i - 1)!$
- The binomial form (i.e.,  $m = 2$ ) of a Dirichlet distribution is called a *Beta* distribution

# Dirichlet distributions (2)



# Dirichlet distributions (3)



# Posterior distribution with Dirichlet Prior

---

- The Dirichlet distribution is called a *conjugate prior* for the multinomial because the posterior distribution is also Dirichlet

$$\begin{aligned} P(\theta|D) &\propto P(D|\theta)P(\theta) && \text{(Bayes rule)} \\ &\propto \left( \prod_{k=1}^m \theta_k^{n_k} \right) \left( \prod_{k=1}^m \theta_k^{\alpha_k-1} \right) \\ &= \prod_{k=1}^m \theta_k^{n_k+\alpha_k-1}, && \text{so} \\ P(\theta|D) &= \frac{\Gamma(\sum_{k=1}^m n_k + \alpha_k)}{\prod_{k=1}^m \Gamma(n_k + \alpha_k)} \prod_{k=1}^m \theta_k^{n_k+\alpha_k-1} \end{aligned}$$

- That is, the  $\alpha_k$  in the prior behave like “pseudo-data”

# The expected value $E[\theta]$

---

- Bayesians prefer to use  $P(\theta|D)$  if possible, but often isn't practical
- The *expected value* of a quantity  $X$  is its “weighted average” value

$$E[X] = \int x P(x) dx$$

- The expected value of  $\theta$  given data  $D$  and a Dirichlet prior  $\alpha$  is:

$$\begin{aligned} E[\theta] &= \int_{\sum_k \theta_k = 1} \theta P(\theta|D) d\theta \\ &= \frac{n_k + \alpha_k}{\sum_{k=1}^m n_k + \alpha_k} \end{aligned}$$

- Thus the Bayesian expected value  $E[\theta]$  is like the MLE, except that it has “pseudo-data”  $\alpha$

# Bayes estimates of $n$ -gram multinomials

---

- The *expected value* of  $\theta$  given data  $D = (n_1, \dots, n_m)$ , where  $n_k$  is number of times word type  $k$  appears in training corpus, and Dirichlet prior  $\alpha = (\alpha_1, \dots, \alpha_m)$  is

$$E[\theta_k] = \frac{n_k + \alpha_k}{\sum_{k=1}^m n_k + \alpha_k}$$

- $\alpha_k = 1$  is a *flat prior*
  - in practice this gives far too much probability to unseen bigrams
  - and the actual distribution of bigram probabilities is skewed towards 0, so we need  $\alpha_k \ll 1$

# Using held-out data

---

- Split data into a (larger) *training* corpus and (smaller) *held-out* corpus
- Idea: The probability  $\theta_k$  of word  $k$  depends only on its frequency in the training corpus  $n_k$
- All words  $k$  that have training corpus frequency  $n_k$  will get the same  $\theta_k$
- Estimate  $\theta_k$  from *average frequency in held-out corpus of all words with training frequency  $n_k$*

# Held-out estimate of $\theta$

---

- $n'_k$  is number of times word  $k$  appears in *heldout data*, and  $n' = \sum_{k=1}^m n'_k$  is total number of words in heldout data

$\hat{\theta}'_k = n'_k/n'$  MLE  $\theta$  for word  $k$  based on heldout frequencies

$S_i = \{k : n_k = i\}$  set of words  $k$  that appear  $i$  times in train

- So the *average  $\theta'_k$  of words in  $S_i$*  is:

$$\frac{1}{|S_i|} \sum_{k \in S_i} \hat{\theta}'_k = \frac{n'_{S_i}}{|S_i|n'}$$

where  $n'_{S_i} = \sum_{k \in S_i} n'_k$  is the number of times any word from  $S_i$  appears in the heldout corpus

# Entropy

---

- The *entropy* of a probability distribution is the average amount of information (in bits) needed to identify a sample  $X \sim P$

$$\begin{aligned} H(X) &= -E[\log_2 P(x)] \\ &= -\sum_x \log_2(P(x))P(x) \end{aligned}$$

- The *cross-entropy* is the amount of information needed to identify  $X \sim P$  when guessing using probability distribution  $Q$

$$\begin{aligned} H(X, Q) &= -E[\log_2 Q(x)] \\ &= -\sum_x \log_2(Q(x))P(x) \end{aligned}$$

- In general  $H(X, Q) \geq H(X)$ , and  $H(X, Q) = H(X)$  iff  $Q = P$

# Homework for next Tuesday

---

- Please read Chapter 7 for next week
- Programs are available on CS machines in `~mj/cg136/programs`.  
The Hansard files are in `~mj/cg136/hansard-data`
- For the current version of IBM Model 1, what kinds of alignment errors does it make? Hint: does frequency matter?
- Modify your Model 1 code to replace the MLE estimate  $P(f|e) = \hat{\theta}_{f|e}$  in the EM algorithm with the Bayesian estimate  $P(f|e) = E[\theta_{f|e}]$ .
  - What effect does the Dirichlet parameter  $\alpha$  have on alignment accuracy?