Graphical Models

Stephen Clark
(based heavily on slides by Mark Gales)

Lent 2013

Machine Learning for Language Processing: Lecture 3

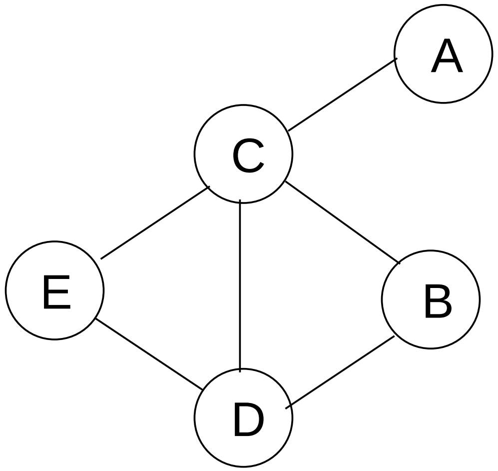
MPhil in Advanced Computer Science

Graphical Models

- Graphical models have their origin in several areas of research
  - a union of graph theory and probability theory
  - framework for representing, reasoning with, and learning complex problems
- Used for for multivariate (multiple variable) probabilistic systems; encompass:
  - language models (Markov Chains);
  - mixture models;
  - factor analysis;
  - hidden Markov models;
  - Kalman filters
- Subsequent lectures will examine forms, training and inference with these systems

Basic Notation

- A graph consists of a collection of nodes and edges
  - Nodes, or vertices, are usually associated with the variables
    distinction between discrete and continuous ignored in this initial discussion
  - Edges connect nodes to one another
- For undirected graphs absence of an edge between nodes indicates conditional independence
  - graph can be considered as representing dependencies in the system



- 5 nodes, $\{A, B, C, D, E\}$, 6 edges
- Various operations on sets of these:
  - $\mathcal{C}_1 = \{A, C\}; \mathcal{C}_2 = \{B, C, D\}; \mathcal{C}_3 = \{C, D, E\}$
  - union: $\mathcal{S} = \mathcal{C}_1 \cup \mathcal{C}_2 = \{A, B, C, D\}$
  - intersection: $\mathcal{S} = \mathcal{C}_1 \cap \mathcal{C}_2 = \{C\}$
  - removal: $\mathcal{C}_1 \setminus \mathcal{S} = \{A\}$

Conditional Independence

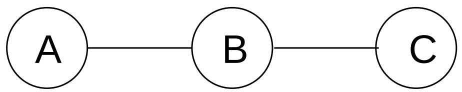
- A fundamental concept in graphical models is conditional independence
  - consider three variables, A, B and C. We can write

$$P(A,B,C) = P(A)P(B|A)P(C|B,A)$$

  - if C is conditionally independent of A given B, then we can write

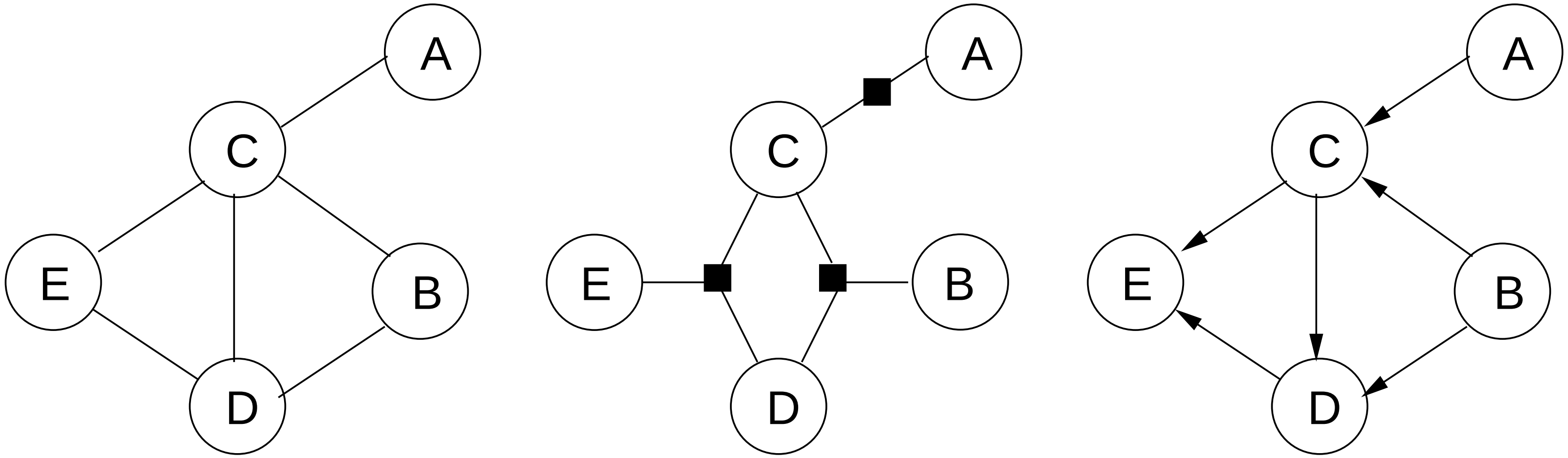
$$P(A,B,C) = P(A)P(B|A)P(C|B)$$

  - the value of A does not affect the distribution of C if B is known.
- Graphically this can be described as



- Conditional independence is important when modelling highly complex systems

Forms of Graphical Model



Undirected Graph　　Factor Graph　　Bayesian Network

- For the undirected graph probability calculation based on

$$P(A,B,C,D,E) = \frac{1}{Z}P(A,C)P(B,C,D)P(C,D,E)$$

where Z is the appropriate normalisation term

– this is the same as the product of the three factors in the factor graph

- This course will concentrate on Bayesian Networks

Bayesian Networks

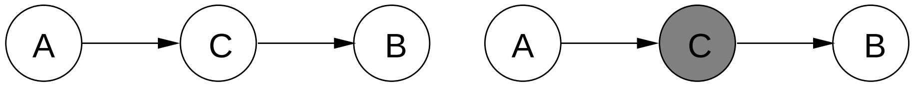
- A specific form of graphical model are Bayesian networks:
  - directed acyclic graphs (DAGs)
  - directed: all connections have arrows associated with them
  - acyclic: following the arrows around it is not possible to complete a loop
- The main problems that need to be addressed are:
  - inference (from observation `it's cloudy` infer probability of `wet grass`)
  - training the models
  - determining the structure of the network (i.e. what is connected to what)
- The first two issues will be addressed in these lectures
  - the final problem is an area of on-going research

Notation

- In general the variables (nodes) may be split into two groups:
  - observed (shaded) variables are the ones we have knowledge about
  - unobserved (unshaded) variables are ones we don't know about and therefore have to infer the probability
- The observed/unobserved variables may differ between training and testing
  - e.g. for supervised training know the class of interest
- We need to find efficient algorithms that allow rapid inference to be made
  - preferably a general scheme that allows inference over any Bayesian network
- First, three basic structures are described in the next slides
  - detail effects of observing one of the variables on the probability

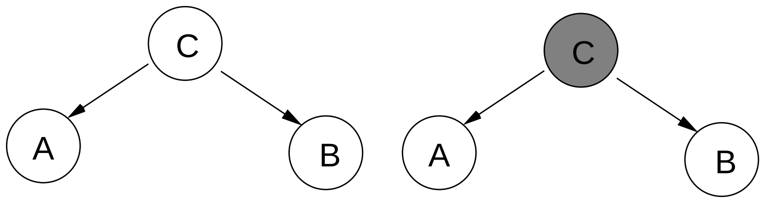
Standard Structures

- Structure 1



  - C not observed: $P(A,B) = \sum_C P(A,B,C) = P(A)\sum_C P(C|A)P(B|C)$
    then A and B are dependent on each other
  - $C = \mathtt{T}$ observed: $P(A,B|C=\mathtt{T}) = P(A)P(B|C=\mathtt{T})$
    A and B are then independent; the path is sometimes called blocked

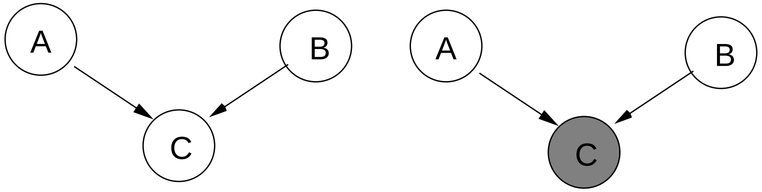
- Structure 2



  - C not observed: $P(A,B) = \sum_C P(A,B,C) = \sum_C P(C)P(A|C)P(B|C)$
    then A and B are dependent on each other
  - $C = \mathtt{T}$ observed: $P(A,B|C=\mathtt{T}) = P(A|C=\mathtt{T})P(B|C=\mathtt{T})$
    A and B are then independent

Standard Structures (cont)

- Structure 3



  - C not observed:

$$P(A,B) = \sum_C P(A,B,C) = P(A)P(B)\sum_C P(C|A,B) = P(A)P(B)$$

    A and B are independent of each other

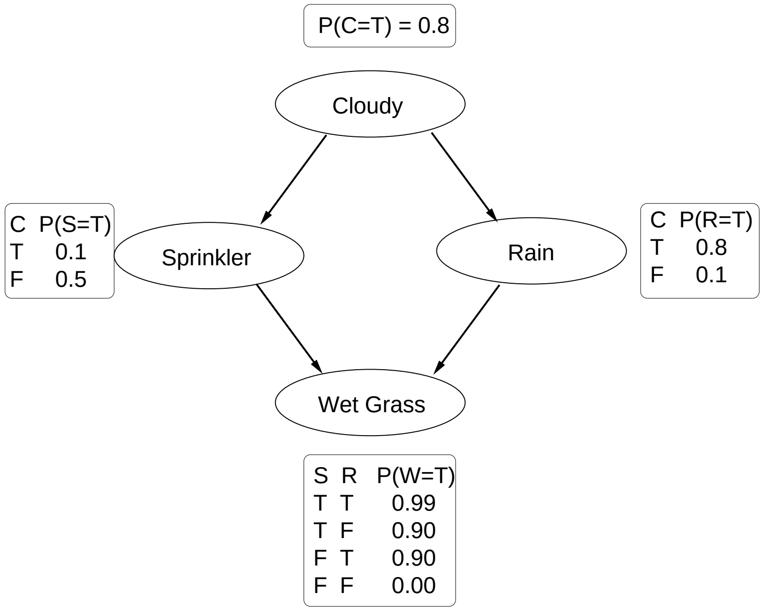
  - $C = \mathtt{T}$ observed:

$$P(A,B|C=\mathtt{T}) = \frac{P(A,B,C=\mathtt{T})}{P(C=\mathtt{T})} = \frac{P(C=\mathtt{T}|A,B)P(A)P(B)}{P(C=\mathtt{T})}$$

    A and B are not independent of each other if C is observed

- Two variables are dependent if a common child is observed - explaining away

Simple Example



- Consider the Bayesian network to left
  - whether the grass is wet, W
  - whether the sprinkler has been used, S
  - whether it has rained, R
  - whether it is cloudy C
- Associated with each node
  - conditional probability table (CPT)
- Yields a set of conditional independence assumptions so that:

$$P(C,S,R,W) = P(C)P(S|C)P(R|C)P(W|S,R)$$

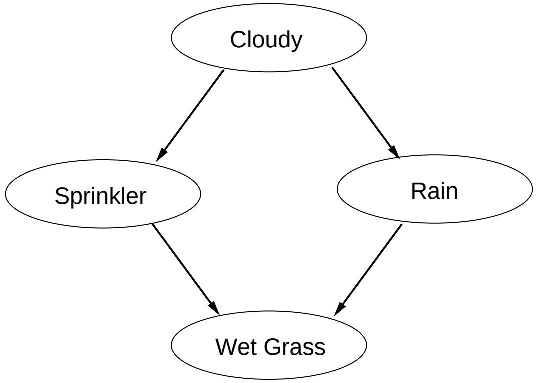
- Possible to use CPTs for inference: Given $C = \mathtt{T}$ what is

$$P(W = \mathtt{T}|C = \mathtt{T}) = \sum_{S=\{\mathtt{T},\mathtt{F}\}} \sum_{R=\{\mathtt{T},\mathtt{F}\}} \frac{P(C = \mathtt{T}, S, R, W = \mathtt{T})}{P(C = \mathtt{T})} = 0.7452$$

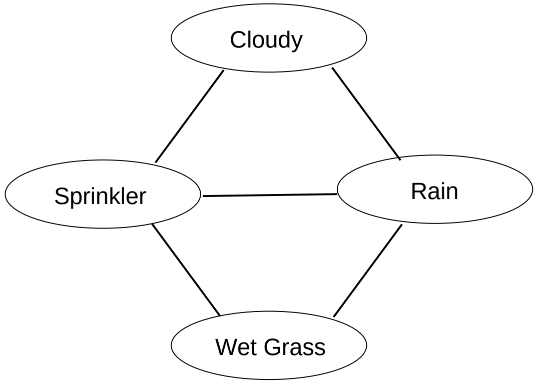
General Inference

- A general approach for inference with BNs is message passing
  - no time in this course for detailed analysis of general case
  - very brief overview here
- Process involves identifying:
  - Cliques $\mathcal{C}$: fully connected (every node is connected to every other node) subset of all the nodes
  - Separators $\mathcal{S}$: the subset of the nodes of a clique that are connected to nodes outside the clique
  - Neighbours $\mathcal{N}$: the set of neighbours for a particular clique
- Thus given the value of the separators for a clique it is conditionally independent of all other variables

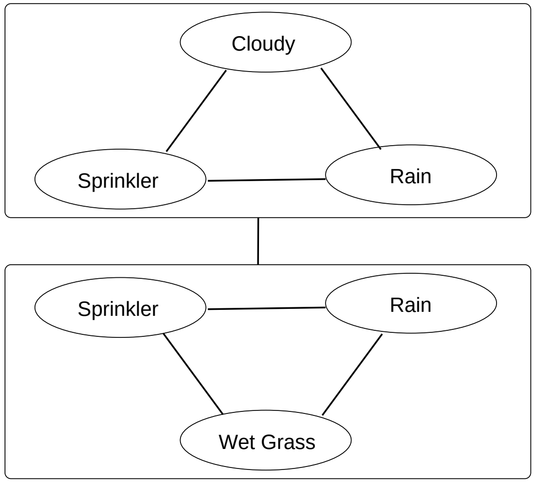
Simple Inference Example



Bayesian Network



Moral Graph



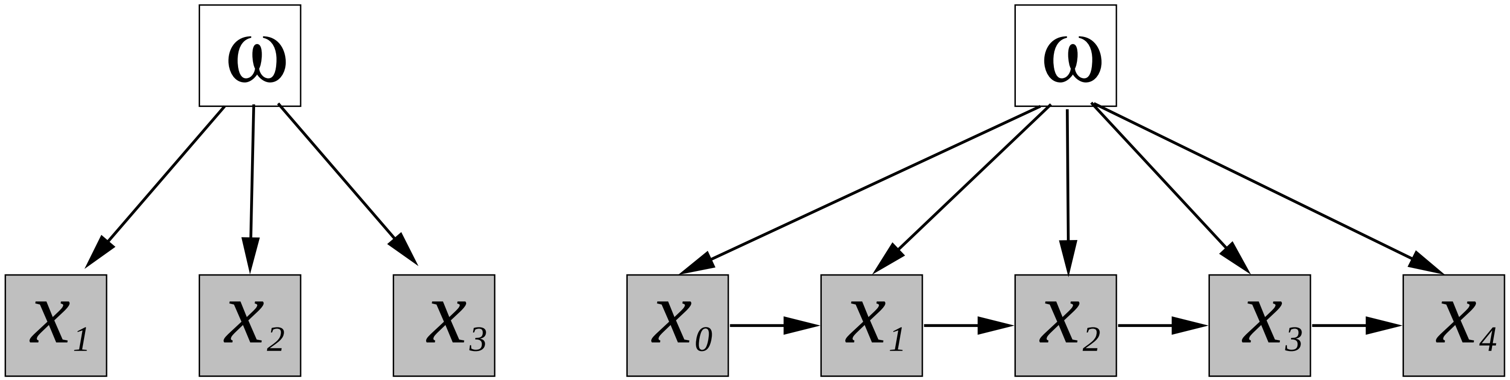
Junction Tree

- Two cliques: $\mathcal{C}_1 = \{C, S, R\}$, $\mathcal{C}_2 = \{S, R, W\}$, one separator: $\mathcal{S}_{12} = \{S, R\}$
  - pass message between cliques: $\phi_{12}(\mathcal{S}_{12}) = \sum_C P(\mathcal{C}_1)$
  - message is: $\phi_{12}(\mathcal{S}_{12}) = P(S|C = \mathtt{T})P(R|C = \mathtt{T})$
  - CPT associated with message to the right

S	R	$P()$
T	T	0.08
T	F	0.02
F	T	0.72
F	F	0.18

Beyond Naive Bayes' Classifier

- Consider classifiers for the class given sequence: x_1, x_2, x_3



$P(\omega_j)\prod_{i=1}^{3} P(x_i|\omega_j)$ $\qquad$ $P(\omega_j)P(x_0|\omega_j)\prod_{i=1}^{4} P(x_i|x_{i-1},\omega_j)$

- Consider the simple generative classifiers above (with joint distribution)
  - naive-Bayes' classifier on left (conditional independent features given class)
  - for the classifier on the right - a bigram model
    * addition of `sequence start` feature x_0 (note $P(x_0|\omega_j) = 1$)
    * addition of `sequence end` feature x_{d+1} (variable length sequence)
- Decision now based on a more complex model
  - this is the approach used for generating (class-specific) language models

Language Modelling

- In order to use Bayes' decision rule need to be able to have the prior of a class
  - many speech and language processing this is the sentence probability $P(\boldsymbol{w})$
  - examples include speech recognition, machine translation

$$P(\boldsymbol{w}) = P(w_0, w_1, \ldots w_k, w_{K+1}) = \prod_{k=1}^{K+1} P(w_k | w_0, \ldots w_{k-2}, w_{k-1})$$

  - K words in sentence $w_1, \ldots, w_k$
  - w_0 is the sentence start marker and w_{K+1} is sentence end marker.
  - require word by word probabilities of partial strings given a history

- Can be class-specific - topic classification (select topic τ given text $\boldsymbol{w}$)

$$\hat{\tau} = \underset{\tau}{\operatorname{argmax}} \left\{ P(\tau|\boldsymbol{w}) \right\} = \underset{\tau}{\operatorname{argmax}} \left\{ P(\boldsymbol{w}|\tau) P(\tau) \right\}$$

N-Gram Language Models

- Consider a task with a vocabulary of V words (LVCSR $65K+$)
  - 10-word sentences yield (in theory) V^{10} probabilities to compute
  - not every sequence is valid but number still vast for LVCSR systems

Need to partition histories into appropriate equivalence classes

- Assume words conditionally independent given previous $N-1$ words: $N=2$

$$P(\texttt{bank}|\texttt{I},\texttt{robbed},\texttt{the}) \approx P(\texttt{bank}|\texttt{I},\texttt{fished},\texttt{from},\texttt{the}) \approx P(\texttt{bank}|\texttt{the})$$

  - simple form of equivalence mappings - a bigram language model

$$P(\boldsymbol{w}) = \prod_{k=1}^{K+1} P(w_k|w_0,\ldots w_{k-2},w_{k-1}) \approx \prod_{k=1}^{K+1} P(w_k|w_{k-1})$$

N-Gram Language Models

- The simple bigram can be extended to general N-grams

$$P(\boldsymbol{w}) = \prod_{k=1}^{K+1} P(w_k|w_0, \dots w_{k-2}, w_{k-1}) \approx \prod_{k=1}^{K+1} P(w_k|w_{k-N+1}, \dots, w_{k-1})$$

- Number of model parameters scales with the size if N (consider $V = 65K$):
  - unigram (N=1): $65K^1 = 6.5 \times 10^4$
  - bigram (N=2): $65K^2 = 4.225 \times 10^9$
  - trigram (N=3): $65K^3 = 2.746 \times 10^{14}$
  - 4-gram (N=4): $65K^4 = 1.785 \times 10^{19}$

  Web comprises about 20 billion pages - not enough data!

- Long-span models should be more accurate, but large numbers of parameters

A central problem is how to get robust estimates and long-spans?

Modelling Shakespeare

- Jurafsky & Martin: N-gram trained on the complete works of Shakespeare

Unigram

- Every enter now severally so, let
- Will rash been and by I the me loves gentle me not slavish page, the and hour; ill let

Bigram

- What means, sir. I confess she? then all sorts, he is trim, captain.
- The world shall- my lord!

Trigram

- Indeed the duke; and had a very good friend.
- Sweet prince, Fallstaff shall die. Harry of Monmouth's grave.

4-gram

- It cannot be but so.
- Enter Leonato's brother Antonio, and the rest, but seek the weary beds of people sick.

Assessing Language Models

- Often use entropy, H, or perplexity, PP, to assess the LM

$$H = -\sum_{w \in \mathcal{V}} P(w)\log_2(P(w)), \quad PP = 2^H; \ \mathcal{V} \text{ is the set of all possible events}$$

  - difficult when incorporating word history into LMs
  - not useful to assess how well specific text is modelled with a given LM

- Quality of a LM is usually measures by the test-set perplexity
  - compute the average value of the sentence log-probability (LP)

$$LP = \lim_{K\to\infty} -\frac{1}{K+1}\sum_{k=1}^{K+1}\log_2 P\left(w_k|w_0\ldots w_{k-2}w_{k-1}\right)$$

- In practice LP must be estimated from a (finite-sized) portion of test text
  - this is a (finite-set) estimate for the entropy
  - the test-set perplexity, PP, can be found as $PP = 2^{LP}$

Language Model Estimation

- Simplest approach to estimating N-grams is to count occurrences

$$\hat{P}(w_k|w_i, w_j) = \frac{f(w_i, w_j, w_k)}{\sum_{k=1}^{V} f(w_i, w_j, w_k)} = \frac{f(w_i, w_j, w_k)}{f(w_i, w_j)}$$

  $f(a, b, c, \ldots) =$ number of times that the word sequence ($event$) "a b c . . . " occurs in the training data

- This is the maximum likelihood estimate
  - excellent model of the training ...
  - many possible events will not be seen, zero counts - zero probability
  - rare events, $f(w_i, w_j)$ is small, estimates unreliable

- Two solutions discussed here:
  - discounting allocating some "counts" to unseen events
  - backing-off for rare events reduce the size of N

Maximum Likelihood Training - Example

- As an example take University telephone numbers. Let's assume that
  1. All telephone numbers are 6 digits long
  2. All numbers start (equally likely) with "33", "74" or "76"
  3. All other digits are equally likely

What is the resultant perplexity rates for various N-grams?

- Experiment using 10,000 or 100 numbers to train (ML), 1000 to test.
  - Perplexity numbers are given below (11 tokens including sentence end):

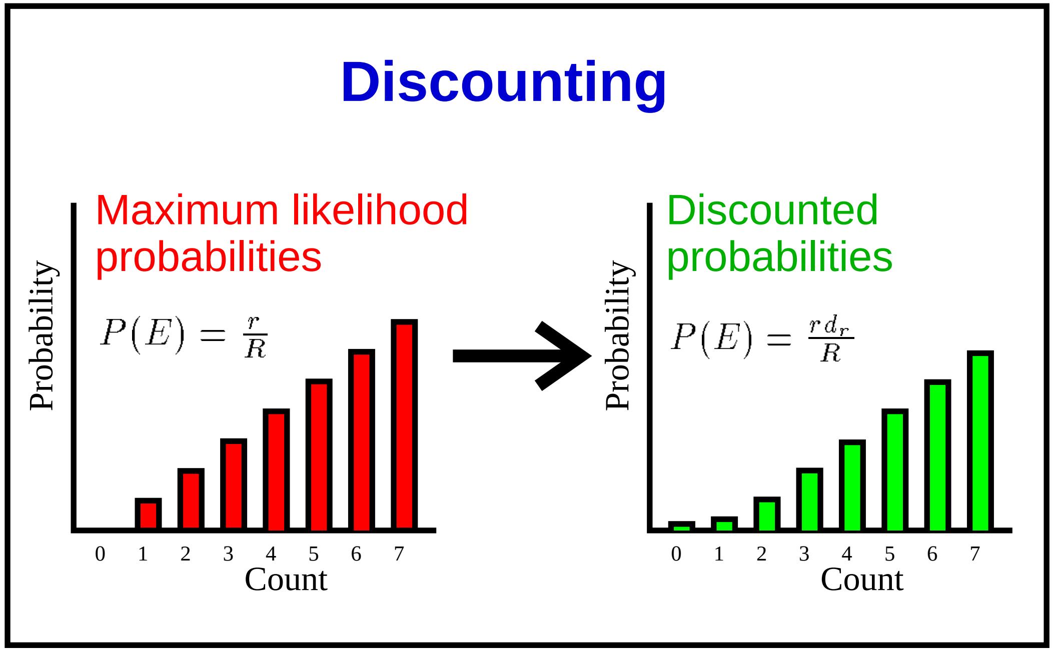
Language Model	10000		100	
	Train	Test	Train	Test
equal	11.00	11.00	11.00	11.00
unigram	10.04	10.01	10.04	10.04
bigram	7.12	7.13	6.56	∞

Discounting

- Need to reallocate some counts to unseen events

- Must satisfy (valid PMF)

$$\sum_{k=1}^{V} \hat{P}(w_k|w_i, w_j) = 1$$



- General form of discounting

$$\hat{P}(\omega_k|\omega_i, \omega_j) \quad = \quad d(f(\omega_i, \omega_j, \omega_k))\frac{f(\omega_i, \omega_j, \omega_k)}{f(\omega_i, \omega_j)}$$

  – need to decide form of $d(f(\omega_i, \omega_j, \omega_k))$ (and ensure sum-to-one constraint)

Forms of Discounting

- Notation: r=count for an event, n_r=number of N-grams with count r
- Various forms of discounting (Knesser-Ney also popular)
  - Absolute discounting: subtract constant from each count

    $$d(r) = (r - b)/r$$

    Typically $b = n_1/(n_1 + 2n_2)$ - often applied to all counts
  - Linear discounting:

    $$d(r) = 1 - (n_1/T_c)$$

    where T_c is the total number of events - often applied to all counts.
  - Good-Turing discounting: ("mass" observed once $= n_1$, observed $r = rn_r$)

    $$r^* = (r+1)n_{r+1}/n_r; \text{ probability estimates based on } r^*$$

    unobserved same "mass" as observed once; once same "mass" as twice etc

Backing-Off

- An alternative to using discounting is to use lower N-grams for rare events
  - lower-order N-gram will yield more reliable estimates
  - for the example of a bigram

$$\hat{P}(w_j|w_i) = \begin{cases} d(f(w_i, w_j))\frac{f(w_i,w_j)}{f(w_i)} & f(w_i, w_j) > C \\ \alpha(w_i)\hat{P}(w_j) & \text{otherwise} \end{cases}$$

  $\alpha(w_i)$ is the back-off weight, it is chosen to ensure that $\sum_{j=1}^{V} \hat{P}(w_j|w_i) = 1$

- C is the N-gram cut-off point (can be set for each value of N)
  - value of C also controls the size of the resulting language model
- Note that the back-off weight is computed separately for each history and uses the $N-1$'th order N-gram count.