CS 562: Empirical Methods in Natural Language Processing

Unit 1: Sequence Models

Lecture 5: Probabilities and Estimations

Lecture 6: Weighted Finite-State Machines

Week 3 -- Sep 8 & 10, 2009
Liang Huang (lhuang@isi.edu)
Probabilities

- experiment (e.g., “toss a coin 3 times”)
- basic outcomes \( \Omega \) (e.g., \( \Omega = \{HHH, HHT, HTH, ..., TTT\} \) )
- event: some subset \( A \) of \( \Omega \) (e.g., \( A = \text{“heads twice”} \) )
- probability distribution
  - a function \( p \) from \( \Omega \) to \([0, 1]\)
  - \( \sum_{e \in \Omega} p(e) = 1 \)
- probability of events (marginals)
  - \( p(A) = \sum_{e \in A} p(e) \)
Joint and Conditional Probs

Joint probability of A and B

\[ P(A, B) = P(A \cap B) \]

Conditional probability of A given B

\[ P(A|B) = \frac{P(A, B)}{P(B)} \]

\[ P(B|A) = \frac{P(A, B)}{P(A)} \]

\[ \frac{\text{proportion of } \Omega \text{ taken up by } B}{\text{proportion of } \Omega \text{ taken up by } A} \]

Example:

\( A = \text{“heads twice”} \quad P(A) = \frac{3}{8} \)
\( B = \text{“first flip = heads”} \quad P(B) = \frac{1}{2} \)

\[ P(B|A) = \text{“if you got two heads, what is chance the first flip was heads?”} \]

\[ P(A, B) \neq P(A) + P(B) ! \]
**Multiplication Rule**

\[ P(A, B) = P(A) \cdot P(B | A) \]
\[ = P(B) \cdot P(A | B) \]

*from defn of conditional prob.*

*Chain rule generalization*

\[ P(A_1, \ldots, A_n) = P(A_1) \cdot P(A_2 | A_1) \cdot P(A_3 | A_1, A_2) \cdots P(A_n | A_1, \ldots, A_{n-1}) \]

\[ \frac{P(A_1, A_2, A_3)}{P(A_1, A_2)} \]

\[ \Omega, \quad A_1, A_2, A_3 \]
Independence

• \( P(A, B) = P(A) P(B) \) or \( P(A) = P(A|B) \)

• disjoint events are always dependent! \( P(A,B) = 0 \)
  - unless one of them is “impossible”: \( P(A)=0 \)

• conditional independence:
  \[ P(A, B|C) = P(A|C) P(B|C) \]
  \[ P(A|C) = P (A|B, C) \]

A, B are independent if \( P(A) = P(A|B) \)

\[
\begin{align*}
\text{FRIDAY} & \quad \text{RAIN} \\
P(\text{FRIDAY}) &= 1/7 \\
P(\text{FRIDAY} | \text{RAIN}) &= 1/7 \\
\text{independent}
\end{align*}
\]

\[
\begin{align*}
\text{FRIDAY} & \quad \text{RAIN} \\
P(\text{FRIDAY}) &= 1/7 \\
P(\text{FRIDAY} | \text{RAIN}) &= 0 \\
\text{dependent}
\end{align*}
\]

\[
\begin{align*}
\text{FRIDAY} & \quad \text{RAIN} \\
P(\text{FRIDAY}) &= 1/7 \\
P(\text{FRIDAY} | \text{RAIN}) &= 4/5 \\
\text{dependent}
\end{align*}
\]
Marginalization

- compute marginal probs from joint/conditional probs

\[ P(B) = P(A, B) + P(\bar{A}, B) \]

\[ = P(A) \cdot P(B|A) + P(\bar{A}) \cdot P(B|\bar{A}) \]
Bayes Rules

Bayes Rule

\[
P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B)}
\]

Proof:

\[
P(A, B) = P(A) \cdot P(B|A)
\]

\[
\Uparrow
\]

\[
equate \& \ divide \ by \ P(B)
\]

alternative bayes rule by partition
Most Likely Event

\[
\text{most likely event} = \arg\max_{A_i} P(A_i)
\]

\[
\arg\max_{A_i} P(A_i | B)
\]

(e.g., if \( A_i \in \{\text{rain, snow, sun}\} \) and
\[
P(\text{rain}) = 0.1 \\
P(\text{snow}) = 0.1 \\
P(\text{sun}) = 0.8
\]

then
\[
\arg\max_{A_i} P(A_i = \text{sun})
\]
\[
\arg\max_{A_i} P(A_i | B) = \arg\max_{A_i} P(A_i) \cdot P(B | A_i)
\]

the right-hand side is often easier to work with.

e.g., let \( B \) be a misspelled character sequence
let \( A_1, \ldots, A_n \) be possible corrections

\[
P(A_i | B) \Rightarrow P(\text{knight} | \text{kmight})
\]

\[
P(A_i) \cdot P(B | A_i) \Rightarrow P(\text{knight}) \cdot P(\text{kmight} | \text{knight})
\]

the "prior" term allows us to bring a lot of background knowledge to bear.
Estimating Probabilities

- how to get probabilities for basic outcomes?
  - do experiments
  - count stuff
- e.g. how often do people start a sentence with “the”?
  - \( P(A) = \frac{\text{(# of sentences like “the ...” in the sample)}}{\text{( # of all sentences in the sample)}} \)
  - \( P(A \mid B) = \frac{\text{(count of A, B)}}{\text{(count of B)}} \)
- we will show that this is Maximum Likelihood Estimation
• what is a MODEL?

• a general theory of how the data is generated,

• along with a set of parameter estimates

• e.g., given this statistics

• we can “guess” it’s generated by a 12-sided die

• along with 11 free parameters $p(1), p(2), \ldots, p(11)$

• alternatively, by two tosses of a single 6-sided die

• along with 5 free parameters $p(1), p(2), \ldots, p(5)$

• which is better given the data? which better explains the data?

$$\text{argmax}_m p(m|d) = \text{argmax}_m p(m) \ p(d|m)$$
Maximum Likelihood Estimation

- always maximize **posterior**: what’s the best m given d?
- when do we use maximum likelihood estimation?
- with uniform **prior**, same as **likelihood** (explains data)
  - \( \text{argmax}_m p(m|d) = \text{argmax}_m p(m) \frac{p(d|m)}{p(d)} \)  
    - Bayes, and p(d)=1
  - \( = \text{argmax}_m p(d|m) \)  
    - when p(m) uniform

Suppose \( d = H H T H \)

- \( m_1 \) coin is unbiased \( \Rightarrow p(H) = \frac{1}{2} \)
  - \( p(d|m) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = 0.066 \)

- \( m_2 \) coin is biased so that \( p(H) = \frac{3}{4} \)
  - \( p(d|m) = \frac{3}{4} \cdot \frac{3}{4} \cdot \frac{1}{4} \cdot \frac{3}{4} = 0.105 \)

- \( m_3 \) coin is biased so that \( p(H) = \frac{9}{10} \)
  - \( p(d|m) = \frac{9}{10} \cdot \frac{9}{10} \cdot \frac{1}{10} \cdot \frac{9}{10} = 0.073 \)
How do we rigorously derive this?

• assuming any \( p_m(H) = \theta \) is possible, what’s the best \( \theta \)?
• e.g.: data is still H, H, T, H.
• \( \text{argmax}_\theta p(d|m; \theta) = \text{argmax}_\theta \theta^3 (1-\theta) \)
  • take derivatives, make it zero: \( \theta = 3/4 \).
  • works in the general case: \( \theta = n / (n+m) \) \hspace{1cm} (n heads, m tails)
• this is why MLE is just counting in the discrete case
What if we have some prior?

- what if we have arbitrary prior
  - like $p(\theta) = \theta (1-\theta)$
- maximum a posteriori estimation (MAP)
- MAP approaches MLE with infinite
- MAP = MLE + smoothing
  - this prior is just “extra two tosses, unbiased”
- you can inject other priors, like “extra 4 tosses, 3 Hs”
Probabilistic Finite-State Machines

- adding probabilities into finite-state acceptors (FSAs)

- FSA: a set of strings; WFSA: a distribution of strings

```
FSA  \xrightarrow{\text{accepts/generates some strings, rejects others.}}

WFSA  P(x)  \xrightarrow{\text{associates a prob. } P(x) \text{ w/each } x. \text{ if } P(x) = 0, \text{ x is rejected.}}

\text{weighted FSA represents set of pairs, e.g.:}
\{<\text{you too, } 10^{-6}>, <\text{you two, } 10^{-7}>, <\text{you to, } 10^{-21}>, <\text{he the, } 0>, <\text{i eat fish, } 10^{-8}> \ldots\}
```
WFSA

• normalization: transitions leaving each state sum up to 1

• defines a distribution over strings?

• or a distribution over paths?

• => also induces a distribution over strings
Max / Sum Probs

- in a WFSA, which string $x$ has the greatest $p(x)$?
  - graph search (shortest path) problem
    - Dijkstra;
    - or Viterbi if the FSA is acyclic
- does it work for NFA?
  - best path much easier than best string
  - you can determinize it (with exponential cost!)
  - popular work-around: $n$-best list crunching

Edsger Dijkstra (1930-2002)
“GOTO considered harmful”

Dr. Andrew Viterbi
(b. 1932)
Viterbi Alg. (1967)
CMDA, Qualcomm

CS 562 - Lec 5-6: Probs & Ws
A Note on Two Problems in Connexion with Graphs

By
E. W. DIJKSTRA

We consider \( n \) points (nodes), some or all pairs of which are connected by a branch; the length of each branch is given. We restrict ourselves to the case where at least one path exists between any two nodes. We now consider two problems.

**Problem 1.** Construct the tree of minimum total length between the \( n \) nodes. (A tree is a graph with one and only one path between every two nodes.)

In the course of the construction that we present here, the branches are subdivided into three sets:

I. the branches definitely assigned to the tree under construction (they will form a subtree);
II. the branches from which the next branch to be added to set I, will be selected;
III. the remaining branches (rejected or not yet considered).

that's min. spanning tree!
Jarnik\textsuperscript{(1930)} - Prim\textsuperscript{(1957)} - Dijkstra\textsuperscript{(1959)}

Edsger Dijkstra
(1930-2002)
“GOTO considered harmful”
the data for at most $n$ branches, viz. the branches in sets I and II and the branch under consideration in step 2.

**Problem 2.** Find the path of minimum total length between two given nodes $P$ and $Q$.

We use the fact that, if $R$ is a node on the minimal path from $P$ to $Q$, knowledge of the latter implies the knowledge of the minimal path from $P$ to $R$. In the solution presented, the minimal paths from $P$ to the other nodes are constructed in order of increasing length until $Q$ is reached.

In the course of the solution the nodes are subdivided into three sets:

A. the nodes for which the path of minimum length from $P$ is known; nodes will be added to this set in order of increasing minimum path length from node $P$;

B. the nodes from which the next node to be added to set A will be selected; this set comprises all those nodes that are connected to at least one node of set A but do not yet belong to A themselves;

C. the remaining nodes.
Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm

ANDREW J. VITERBI, SENIOR MEMBER, IEEE

Abstract—The probability of error in decoding an optimal convolutional code transmitted over a memoryless channel is bounded from above and below as a function of the constraint length of the code. For all but pathological channels the bounds are asymptotically (exponentially) tight for rates above $R_0$, the computational cutoff rate of sequential decoding. As a function of constraint length the performance of optimal convolutional codes is shown to be superior to that of block codes of the same length, the relative improvement increasing with rate. The upper bound is obtained for a specific probabilistic nonsequential decoding algorithm which is shown to be asymptotically optimum for rates above $R_0$ and whose performance bears certain similarities to that of sequential decoding algorithms.

I. SUMMARY OF RESULTS

Since Elias\textsuperscript{[1]} first proposed the use of convolutional (tree) codes for the discrete memoryless channel, it has been conjectured that the performance of this class of codes is potentially superior to that of block codes of the same length. The first quantitative verification of this conjecture was due to Yudkin\textsuperscript{[2]} who obtained...
Sum Probs

- what is $P(x)$ for some particular $x$?
  - for DFA, just follow $x$
  - for NFA,
    - get a subgraph (by composition), then sum ??
      - acyclic $\Rightarrow$ Viterbi
      - cyclic $\Rightarrow$ compute strongly connected components
        - SCC-DAG cluster graph (cyclic locally, acyclic globally)
        - do infinite sum (matrix inversion) locally, Viterbi globally
  - refer to extra readings on course website
WFSTs

- FST: a relation over strings (a set of string pairs)
- WFST: a probabilistic relation over strings (a set of \(<s, t, p>: \text{strings pair } <s, t> \text{ with probability } p\>)

  - what is p representing?

```plaintext
{ <they can fish, PRO AUX V, 0.99>,
  <they can fish, PRO V N, 0.01>,
  <they can fish, DT DT DT DT, 0.00>,
  <they sing, PRO V, 1.00>,
  ...
}
```

```
WFST

P(ade | acdet) = 0.9 \cdot 0.1 \cdot 0.9 \cdot 0.9 \cdot 0.1 = 0.00729
```

*CS 562 - Lec 5-6: Probs & WFSTs*
Edit Distance as WFST

- this is simplified edit distance
- real edit distance as an example of WFST, but not PFST

```
P(ade | acdet) = 0.9 * 0.1 * 0.9 * 0.9 * 0.1 = 0.00729
```

```
costs:
replacement: 1
insertion: 2
deletion: 2
```

WFST: real edit distance
Normalization

- if transitions leaving each state and each input symbol sum up to 1, then...
  - WFST defines conditional prob $p(y|x)$ for $x \Rightarrow y$
  - what if we want to define a joint prob $p(x, y)$ for $x \Rightarrow y$?
  - what if we want $p(x \mid y)$?

$P(ade \mid acdet) = 0.9 \cdot 0.1 \cdot 0.9 \cdot 0.9 \cdot 0.1 = 0.00729$
Questions of WFSTs

• given \( x, y \), what is \( p(y|x) \) ?
• for a given \( x \), what’s the \( y \) that maximizes \( p(y|x) \) ?
• for a given \( y \), what’s the \( x \) that maximizes \( p(y|x) \) ?
• for a given \( x \), supply all output \( y \) w/ respective \( p(y|x) \)
• for a given \( y \), supply all input \( x \) w/ respective \( p(y|x) \)

\[
P(ade \mid acdet) = 0.9 \cdot 0.1 \cdot 0.9 \cdot 0.9 \cdot 0.1 = 0.00729
\]
Answer: Composition

\[ p(z \mid x) = p(y \mid x) \cdot p(z \mid y) \]

- \[ = \sum_y p(y \mid x) \cdot p(z \mid y) \] have to sum up \( y \)

- how to build a composed WFST \( C \) out of WFSTs \( A, B \)?
  - again, like intersection
  - sum up the weights
Example

\[
P(bt|dr) = 0.06 \cdot 0.5 = 0.03
\]
Another Example
Given x, supply all output y
Given $x$, $y$, what’s $p(y|x)$
Given $x$, what's max $p(y|x)$
Part-of-Speech Tagging Again

Before Probabilities

\[
\text{they} \rightarrow \text{Can} \rightarrow \text{fish} \rightarrow 0 + \text{they:PRO} \cdot \text{can:V} \cdot \text{can:aux} \cdot \text{fish:n} \cdot \text{fish:v} = \text{PRO} \rightarrow \text{V} \rightarrow 0
\]

\text{no preferences}

After Probabilities

\[
\text{they:PRO/1} \cdot \text{can:v/0.01} \cdot \text{can:n/0.14} \cdot \text{can:aux/0.85} \cdot \text{fish:n/0.8} \cdot \text{fish:v/0.2} = \text{PRO/1} \rightarrow \text{V/0.01} \rightarrow 0
\]

\text{needs tweaking:}

\[
\text{can:aux/0.85} \rightarrow 0 \rightarrow \text{fish:v/1.0}
\]
Noisy-Channel Model
Noisy-Channel Model

Best path is
\[ \text{argmax } P(t \ldots t | w \ldots w) \]

Composed

Each try sequence scored by
\[ P(w \ldots w | t \ldots t) \]

Composed

Each try sequence
\[ \text{scored by } P(t \ldots t) \cdot P(w \ldots w | t \ldots t) \]
Applications of Noisy-Channel

| Application                  | Input            | Output                  | $p(i)$                      | $p(o|i)$                  |
|------------------------------|-------------------|-------------------------|-----------------------------|---------------------------|
| Machine Translation          | $L_1$ word sequences | $L_2$ word sequences | $p(L_1)$ in a language model | translation model         |
| Optical Character Recognition (OCR) | actual text       | text with mistakes      | prob of language text       | model of OCR errors       |
| Part Of Speech (POS) tagging | POS tag sequences | English words           | prob of POS sequences       |                           |
| Speech recognition           | word sequences    | speech signal           | prob of word sequences      | acoustic model            |