Language and Statistics II

Lecture 2: Sequences
Noah Smith
Administrivia

• Course list?
• Lit review proposal due in 12 days
• Assignment 1 posted
• Office hours right after lecture (2602F NSH)
Text Data

• Sequence of symbols (letters, characters, words).
  – Infinite or finite set?

• Let $\Sigma$ be the finite set of symbols (alphabet).

• Can we define a distribution over $\Sigma^*$?
  – Assume we want every string to get some mass.
History-Based Models

- Predict each word from left to right.

\[
p(s^n_1) = \prod_{i=1}^{n} \gamma(s_i \mid s^{i-1}_1)
\]

- Representational power?
- How many parameters?
  \(= (\text{number of histories}) \times |\Sigma|\)
- Probability of sequences not in training data?
History-Based Models
Markov (n-gram) Models

• Predict each word from left to right.

\[
p(s_1^n) = \prod_{i=1}^{n} \gamma(s_i | s_{i-m}^{i-1})
\]

• Independence assumption?
• Representational power?
• How many parameters?
  \( \mathcal{O}(|\Sigma|^{m+1}) \)
• Why does it work?
Why are \( n \)-gram models so great?

- Formalism: understandable
- Features: simple (not too many)
  - Really?
- Model: fully generative
- Algorithms?
  - Probability of a sequence
  - Choosing a sequence from a set
  - Training …
Drawbacks of $n$-gram models

• Data sparseness
• Black art of smoothing
• Is $\Sigma$ really fully known?
Application: $\Sigma^*$ is the **output**

$$score(s_1^n) = p(s_1^n \mid x) \propto p(x \mid s_1^n) \cdot p(s_1^n)$$

**Examples of channel models?**

**Language model is a source model.**
n-gram as a Source Model

- Speech Recognition (Jelinek, 1997)
- Machine Translation (Brown et al., 1993)
- Optical Character Recognition (Kolak and Resnik, 2002)
- Spelling Correction (Kernighan, Church, & Gale, 1990)
- Punctuation Restoration (Beeferman, Berger, & Lafferty, 1998)

(This list is not exhaustive!)
n-grams and Lattices

• Suppose we have a weighted lattice (output from the channel model).
• Problem 1: how many paths?
Counting Paths in a Lattice

\[
\text{paths}_\text{to}(v) = \sum_{(u,v) \in E} \text{paths}_\text{to}(u)
\]
\textit{n-grams and Lattices}

- Suppose we have a weighted lattice (output from the channel model).
- Problem 2: Best path?
Best Path in a Lattice

\[
\text{best}(v) = \max_{(u,v) \in E} \text{weight}(u,v) \times \text{best}(u)
\]
n-grams and Lattices

• Suppose we have a weighted lattice (output from the channel model).
• Problem 3: Best path, factoring in n-gram source model?
Best Path in a Lattice, including unigram model

\[
\text{best}(v) = \max_{(u,v,s) \in E} \text{weight}(u,v) \times \text{best}(u) \times \gamma(s)
\]
Best Path in a Lattice, including *bigram* model

$$\text{best}(v; s) = \max_{(u,v,s) \in E, s' \in \Sigma} \text{weight}(u,v) \times \text{best}(u; s') \times \gamma(s',s)$$
Application: \( \Sigma^* \) is the input

\[
\text{score}(x) = p(x \mid s^n_1) \propto p(s^n_1 \mid x) \cdot p(x)
\]

Language model for each \( x \).

Examples of source models?
$n$-gram as a Channel Model

- Text categorization
- Language identification
- Topic segmentation
- Information retrieval (Ponte and Croft, 1998; Berger and Lafferty, 1999)
- Sentence compression (Knight and Marcu, 2002)
- Question $\rightarrow$ Search query (Radev, Qi, Zheng, et al., 2001)

(This list is not exhaustive!)
Improving $n$-gram Models

“Improving” in what sense?

Faster algorithms?  Unlikely!
Better fit to unseen data?
   Smoothing …
Improving $n$-gram Models

“Improving” in what sense?

Faster algorithms? Unlikely!
Better fit to unseen data? Unlikely!
(Smoothing research appears to be at a plateau)
Better suited to tasks? Maybe …
Make use of domain knowledge?
Improving $n$-gram Models

1. Word classes (Brown et al., 1990)

$$p(s^n_1) = \prod_{i=1}^{n} \eta(s_i \mid c_i) \cdot \gamma(c_i \mid c_{i-1}^{i-1})$$

$c_i = \text{class}(s_i)$

class : $\Sigma \rightarrow \Lambda$

Classes are a partition on $\Sigma$; must be chosen.
Improving \textit{n}-gram Models

2. Hidden Markov models

\[ p(c_1^n, s_1^n) = \prod_{i=1}^{n} \eta(s_i | c_i) \cdot \gamma(c_i | c_{i-m}) \]

\[ p(s_1^n) = \sum_{c_1^n \in \Lambda^n} \prod_{i=1}^{n} \eta(s_i | c_i) \cdot \gamma(c_i | c_{i-m}) \]

Classes are a \textbf{hidden random variable}. 
Hidden Markov Model

\[ \gamma(c_i | c_{i-2}, c_{i-1}) \]

\[ \eta(s_i | c_i) \]

*N*-gram model over states (trigram shown)

More expressive model over words!
What HMMs Can Do (that $n$-gram models can’t)

- Some words behave similarly
  - *Color, color, colour, hue*
  - (Hard classes give us this, too!)
- Some words are ambiguous
  - *John colors$_V$ the picture$_N$*
  - *Many colors$_V$ make a rainbow*
  - *Picture$_V$ a man walking on the shore*
- Long distance dependencies (some)
  - ¿Bastante caliente?
- Constraints like “only one verb”
- Parameters: $|\Sigma||\Lambda| + |\Lambda|^m$
NL Applications of HMMs

• Part-of-speech tagging
  (Church, 1988; Brants, 2000)
• Text chunking/shallow parsing (I-O-B tags)
• Named entity recognition
  (Bikel, Schwartz, Weischedel, 1999)
• Word alignment
  (Vogel, Ney, and Tillman, 1996)
I-O-B Trick

• Shallow “bracketing” structure models from HMMs

Ronald Reagan was an American president.