Natural Language Understanding
Lecture 9: Unsupervised Part-of-Speech Tagging 2

Frank Keller

School of Informatics
University of Edinburgh
keller@inf.ed.ac.uk

Based on slides by Sharon Goldwater

February 12, 2015
1 Introduction
   • Hidden Markov Models
   • Approaches to Parameter Estimation

2 Bayesian Integration
   • Dirichlet Distribution
   • Integration and Smoothing

3 Bayesian HMM
   • Bayesianizing the HMM
   • Gibbs Sampling
   • Evaluation

Reading: Goldwater and Griffiths (2007), Resnik and Hardisty (2009). Also, Jurafsky and Martin (2009: Ch. 6.1–6.5) for background on HMMs and EM.
We will use Hidden Markov Models (HMMs) as our running example for Bayesian modeling.

The parameters of the HMM are $\theta = (\tau, \omega)$. They define:

- $\tau$: the probability distribution over tag-tag transitions;
- $\omega$: the probability distribution over word-tag outputs.

$$P(t, w) = \prod_{i=1}^{n} P(t_i|t_{i-1})P(w_i|t_i)$$
Approaches to Parameter Estimation

Maximum-likelihood estimation:

$$\theta^* = \underset{\theta}{\text{argmax}} \ P(w|\theta)$$

Maximum a posteriori estimation:

$$\theta^* = \underset{\theta}{\text{argmax}} \ P(w|\theta)P(\theta)$$

Bayesian integration:

$$P(w_{n+1}|w) = \int_{\Delta} P(w_{n+1}|\theta)P(\theta|w)d\theta$$

$$P(t|w) = \int_{\Delta} P(t|\theta, w)P(\theta|w)d\theta$$
Choosing the right prior can make integration easier.

This is where the *Dirichlet distribution* comes in. A $K$-dimensional Dirichlet with parameters $\alpha = \alpha_1 \ldots \alpha_K$ is defined as:

$$P(\theta) = \frac{1}{Z} \prod_{j=1}^{K} \theta_j^{\alpha_j-1}$$

We usually only use symmetric Dirichlets, where $\alpha_1 \ldots \alpha_K$ are all equal to $\beta$. We write $\text{Dirichlet}(\beta)$ to mean $\text{Dirichlet}(\beta, \ldots, \beta)$. 
A 2-dimensional symmetric Dirichlet(\(\beta\)) prior over \(\theta = (\theta_1, \theta_2)\):

\[ \beta > 1: \text{prefer uniform distributions} \]
\[ \beta = 1: \text{no preference} \]
\[ \beta < 1: \text{prefer sparse (skewed) distributions} \]

The 2-dim. Multinomial is called Binomial; the 2-dim. Dirichlet is called Beta.
Example: coin factory. What is the prior distribution over $\theta_h$, the probability of flipping heads using a coin from the factory?

- Factory makes weighted coins, but we don’t know the weight. $\beta = 1$: an uninformative prior.
- Factory normally makes fair coins, but occasionally the equipment is misaligned. $\beta > 1$: we think coins are fair (unless we get a lot of evidence to the contrary);
- Someone has tampered with the equipment. $\beta < 1$: we think coins are biased, but don’t know which way. (A little evidence suggests which way, a lot of evidence required to convince us coins are actually fair.)
The Dirichlet distribution is a useful prior for the multinomial distribution because they are *conjugate* distributions.

The posterior distribution has same form as prior distribution:\(^1\)

$$P(\theta \mid \mathbf{w}) \propto P(\mathbf{w} \mid \theta)P(\theta)$$

This makes integration work out nicely. Specifically, in the symmetric case:

$$P(w_{n+1} = k \mid \mathbf{w}) = \frac{n_k + \alpha_k}{n + \sum_{j=1}^{W} \alpha_j} = \frac{n_k + \beta}{n + W \beta}$$

\(^1\)If \(P(\theta)\) is Dirichlet\((\alpha_1 \ldots \alpha_W)\), then \(P(\theta \mid \mathbf{w})\) is Dirichlet\((\alpha_1 + n_1, \ldots, \alpha_W + n_W)\), where is \(n_k\) is the number of times the \(k\)-th lexical item occurs in \(\mathbf{w}\).
Integration and Smoothing

Hence Bayesian integration gives us a distribution that is automatically smoothed:

A Dirichlet priors give us add-$\beta$ smoothing:

$$P(w_{n+1} = k | w) = \frac{n_k + \beta}{n + W\beta}$$

With $\beta = 1$, we get Laplace smoothing ($n_k$ is frequency of word $k$, $n$ is frequency of all words together, $W$ is vocabulary size):

$$P(w_{n+1} = k | w) = \frac{n_k + 1}{n + W}$$

Fancier priors give us fancier smoothing, e.g., using a hierarchical Pitman-Yor process as prior (generalization of Dirichlet) yields Kneser-Ney smoothing (Teh 2006).
To Bayesianize the HMM, we augment it with symmetric Dirichlet priors:

\[
\begin{align*}
t_i | t_{i-1} = t, \tau(t) & \sim \text{Multinomial}(\tau(t)) \\
\omega_i | t_i = t, \omega(t) & \sim \text{Multinomial}(\omega(t)) \\
\tau(t) | \alpha & \sim \text{Dirichlet}(\alpha) \\
\omega(t) | \beta & \sim \text{Dirichlet}(\beta)
\end{align*}
\]

To simplify things, we will present a bigram version of the Bayesian HMM; Goldwater and Griffiths (2007) use trigrams.
If we integrate out the parameters $\theta = (\tau, \omega)$, we get:

$$P(t_{n+1}|t, \alpha) = \frac{n(t_n, t_{n+1}) + \alpha}{n(t_n) + T\alpha}$$

$$P(w_{n+1}|t_{n+1}, t, w, \beta) = \frac{n(t_{n+1}, w_{n+1}) + \beta}{n(t_{n+1}) + W_{t_{n+1}}\beta}$$

with $T$ possible tags and $W_t$ possible words with tag $t$.

We can use these distributions to find $P(t|w)$ using a method called *Gibbs sampling*. 
Gibbs Sampling

Like EM, a general technique (algorithm family):

- used in lots of Bayesian models (e.g., topic models);
- specific versions for specific models.

Gibbs sampling is an example for a \textit{randomized algorithm}:

- guaranteed to converge;
- after convergence, each iteration produces a sample from the posterior distribution of interest (here, $P(t|w)$).
Gibbs Sampling

1. Initialize hidden variables (tags) at random:

   ![Diagram of hidden variables (tags)]

   - $t_1 ightarrow t_2 ightarrow t_3$
   - $W_1 ightarrow W_2 ightarrow W_3$

2. On each iteration, sample a new value for each tag from its distribution conditioned on the current values of all other variables (both tags and words).

   Basically, pretend all the other tags are correct, treat them as training data, and sample a new value for current tag.
Gibbs Sampling: One Iteration

Sample from:

\[ P(t_1 \mid t_2, t_3, w, \alpha, \beta) \]

\[ P(t_2 \mid t_1, t_3, w, \alpha, \beta) \]

\[ P(t_3 \mid t_1, t_2, w, \alpha, \beta) \]

Result:
Computing Conditional Distributions

By Bayes’ rule, we have:

\[
P(t_1|t_2, t_3, w_1, w_2, w_3, \alpha, \beta) \\ \propto P(w_1|t_1, t_2, t_3, w_2, w_3, \alpha, \beta)P(t_1|t_2, t_3, w_2, w_3, \alpha, \beta) \\ = P(w_1|t_1, t_2, t_3, w_2, w_3, \beta)P(t_1|t_2, t_3, \alpha)
\]

Now pretend that \(t_i\) and \(w_i\) are the last tag/word in the sequence (assume exchangability), and use our formulas from earlier to compute the two factors:

\[
P(w_{n+1}|t_{n+1}, t, w, \beta) = \frac{n(t_{n+1}, w_{n+1}) + \beta}{n(t_{n+1}) + Wt_{n+1}\beta}
\]

\[
P(t_{n+1}|t, \alpha) = \frac{n(t_n, t_{n+1}) + \alpha}{n(t_n) + T\alpha}
\]

- use a dictionary that lists possible tags for each word:
  - run: NN, VB, VBN
- the dictionary is actually derived from WSJ corpus;
- train and test on the unlabeled corpus (24,000 words of WSJ):
  - 53.6% of word tokens have multiple possible tags. Average number of tags per token = 2.3.
Goldwater and Griffiths (2007) evaluate tagging accuracy against the gold-standard WSJ tags and compare to:

- HMM with maximum-likelihood estimation using EM (MLHMM);
- Conditional Random Field with contrastive estimation (CRF/CE).

They also experiment with reducing/eliminating dictionary information.
Integrating over parameters is useful in itself, even with uninformative priors ($\alpha = \beta = 1$);

better priors can help even more, though not quite to state of the art.
Evaluation: Syntactic Clustering

*Syntactic clustering:* input are the words only, no dictionary is used:

- collapse 45 treebank tags onto smaller set of 17;
- hyperparameters ($\alpha$, $\beta$) are inferred automatically using Metropolis-Hastings sampler;
- standard accuracy measure requires labeled classes, so measure accuracy using best matching of classes.
MLHMM groups instances of the same lexical item together;
BHMM clusters are more coherent, more variable in size. Errors are often sensible (e.g., separating common nouns/proper nouns, confusing determiners/adjectives, prepositions/participles).
Results

BHMM transition matrix is sparse, MLHMM is not.
Summary

- Using the Dirichlet distribution for priors lets us easily integrate out parameters;
- this improves performance by averaging out uncertainty.;
- it also allows us to use priors to favor sparse solutions, as they occur in language data;
- Gibbs sampling used for inference:
  - resamples each variable using conditional distribution;
  - converges to samples from posterior distribution;
- similar techniques can be found in most Bayesian models (e.g., Latent Dirichlet Allocation, later in the course).


