Natural Language Processing (CSE 517): Cotext Models

Noah Smith

University of Washington
nasmith@cs.washington.edu

April 18, 2018

Latent Dirichlet Allocation

(Blei et al., 2003)

Widely used today.

$$p(\boldsymbol{x}) = \int_{\boldsymbol{\gamma}} \sum_{\boldsymbol{z} \in \{1,\ldots,k\}^{\ell}} p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\gamma}) \; d\boldsymbol{\gamma}$$

$$p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\gamma}) = \mathrm{Dir}_{\boldsymbol{\alpha}}(\boldsymbol{\gamma}) \prod_{i=1}^{\ell} \gamma_{z_i} \; \theta_{x_i|z_i}$$

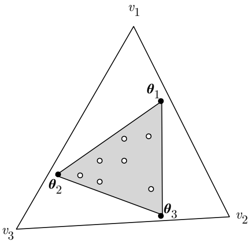
Parameters:

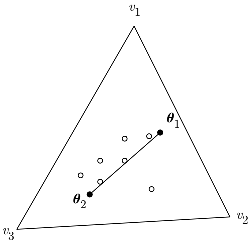
- $\boldsymbol{\alpha} \in \mathbb{R}^k_{>0}$
- $\boldsymbol{\theta}_{*|z} \in \triangle^V$, $\forall z \in \{1, \ldots, k\}$

There is no closed form for the MLE!

Understanding LDA

Models with $k = 3$ (left) and $k = 2$ (right):





- Unigram model estimates one "topic" for the whole corpus.
- PLSA places each document at one point in the topic simplex.
- LDA estimates a posterior distribution in the "topic simplex" for each document (and its vertices).

LDA

Topics discovered by LDA-like models continue to be interesting:

- As a way of interacting with and exploring large corpora without reading them.
  - But this is hard to evaluate!
- As a "pivot" for relating to other variables like author (Rosen-Zvi et al., 2004), geography (Eisenstein et al., 2010), and many more.

LDA is also extremely useful as a pedagogical gateway to Bayesian modeling of text (and other discrete data).

- It's right on the boundary between "easy" and "hard" Bayesian models.

Three Kinds of Cotext

If we consider a word token at a particular position i in text to be the observed value of a random variable X_i, what other random variables are predictive of/related to X_i?

1. the document containing i (a moderate-to-large collection of other words) $\longrightarrow$ topic models
2. the words that occur within a small "window" around i (e.g., x_{i-2}, x_{i-1}, x_{i+1}, x_{i+2}, or maybe the sentence containing i) $\longrightarrow$ distributional semantics
3. a sentence known to be a translation of the one containing i $\longrightarrow$ translation models

Local Contexts: Distributional Semantics

Within NLP, emphasis has shifted from topics to the relationship between $v \in \mathcal{V}$ and more local contexts.

For example: LSI/A, but replace documents with "nearby words." This is a way to recover word vectors that capture distributional similarity.

These models are designed to "guess" a word at position i given a word at a position in $\{i - w, \ldots, i - 1\} \cup \{i + 1, \ldots, i + w\}$.

Sometimes such methods are used to "pre-train" word vectors used in other, richer models (like neural language models).

Word2vec

(Mikolov et al., 2013a,b)

Two models for word vectors designed to be computationally efficient.

- Continuous bag of words (CBOW): $p(v \mid c)$
  - Similar in spirit to the feedforward neural language model we saw before (Bengio et al., 2003)
- Skip-gram: $p(c \mid v)$

It turns out these are closely related to matrix factorization as in LSI/A (Levy and Goldberg, 2014)!

Skip-Gram Model

$$p(C = c \mid X = v) = \frac{1}{Z_v} \exp \mathbf{c}_c^\top \mathbf{v}_v$$

- Two different vectors for each element of $\mathcal{V}$: one when it is "v" ($\mathbf{v}$) and one when it is "c" ($\mathbf{c}$).
- Like the log-bilinear model we saw before, normalization term Z_v is expensive, so approximations are required for efficiency.
- Can expand this to be over the whole sentence or document, or otherwise choose which words "count" as context.

Word Vector Evaluations

See `http://wordvectors.org` for a suite of examples.

Several popular methods for *intrinsic* evaluations:

Word Vector Evaluations

See `http://wordvectors.org` for a suite of examples.

Several popular methods for *intrinsic* evaluations:

- Do (cosine) similarities of pairs of words' vectors correlate with judgments of similarity by humans?

Word Vector Evaluations

See `http://wordvectors.org` for a suite of examples.

Several popular methods for *intrinsic* evaluations:

- Do (cosine) similarities of pairs of words' vectors correlate with judgments of similarity by humans?
- TOEFL-like synonym tests, e.g., *rug* $\stackrel{?}{\rightarrow}$ $\{$*sofa*, *ottoman*, *carpet*, *hallway*$\}$

Word Vector Evaluations

See `http://wordvectors.org` for a suite of examples.

Several popular methods for *intrinsic* evaluations:

- Do (cosine) similarities of pairs of words' vectors correlate with judgments of similarity by humans?
- TOEFL-like synonym tests, e.g., *rug* $\stackrel{?}{\rightarrow}$ {*sofa*, *ottoman*, *carpet*, *hallway*}
- Syntactic analogies, e.g., "*walking* is to *walked* as *eating* is to what?" Solved via:

$$\min_{v \in \mathcal{V}} \cos\left(\mathbf{v}_v, \mathbf{v}_{\mathit{walking}} - \mathbf{v}_{\mathit{walked}} + \mathbf{v}_{\mathit{eating}}\right)$$

Word Vector Evaluations

See http://wordvectors.org for a suite of examples.

Several popular methods for *intrinsic* evaluations:

- Do (cosine) similarities of pairs of words' vectors correlate with judgments of similarity by humans?
- TOEFL-like synonym tests, e.g., *rug* $\stackrel{?}{\rightarrow}$ {*sofa*, *ottoman*, *carpet*, *hallway*}
- Syntactic analogies, e.g., "*walking* is to *walked* as *eating* is to what?" Solved via:

$$\min_{v \in \mathcal{V}} \cos\left(\mathbf{v}_v, \mathbf{v}_{\textit{walking}} - \mathbf{v}_{\textit{walked}} + \mathbf{v}_{\textit{eating}}\right)$$

Also: *extrinsic* evaluations on NLP tasks that can use word vectors (e.g., sentiment analysis).

An Older Approach to Word Representation

Recall the class-based bigram model:

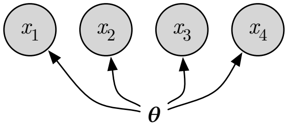
$$
\begin{aligned}
p(x_i \mid x_{i-1}) &= p(x_i \mid z_i) \cdot p(z_i \mid z_{i-1}) \\
&= \theta_{x_i \mid z_i} \cdot \gamma_{z_i \mid z_{i-1}} \\
p(\boldsymbol{x}, \boldsymbol{z}) &= \pi_{z_0} \prod_{i=1}^{\ell} \theta_{x_i \mid z_i} \cdot \gamma_{z_i \mid z_{i-1}}
\end{aligned}
$$

This is like a topic model where topic distributions are **bigram** distributed!

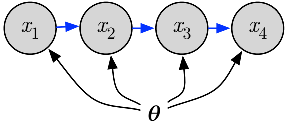
If we treat each z as latent—like in a topic model—we get to something very famous, called the **hidden Markov model** (HMM).

Comparing Five Models

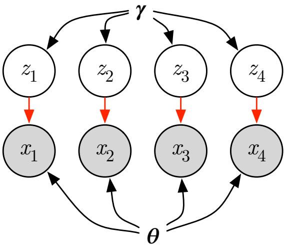
unigram



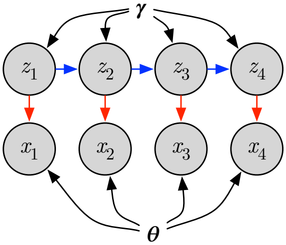
bigram
(Markov)



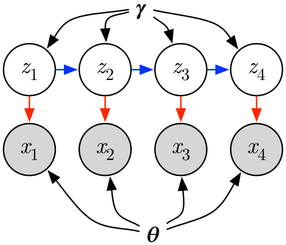
PLSA
and LDA
(topics)



class-based
bigram



hidden
Markov
model



Brown Clustering

There is a whole lot more to say about HMMs, which we'll save for later.

Brown et al. (1992) focused on the case where each $v \in \mathcal{V}$ is constrained to belong to only one cluster, $\mathsf{cl}(v)$.

They developed a greedy way to cluster words hierarchically.

Brown Clustering: Sketch of the Algorithm

Given: k (the desired number of clusters)

- Initially, every word v belongs to its own cluster.
- Repeat $V - k$ times:
  - Find the pairwise merge that gives the greatest value for $p(\boldsymbol{x}_{1:n}, \boldsymbol{z}_{1:n})$.

It turns out this is equivalent to PMI for adjacent cluster values!

This is very expensive; Brown et al. (1992) and others (later) introduced tricks for efficiency. See Liang (2005) and Stratos et al. (2014), for example.

Added Bonus to Brown Clusters

If you keep track of every merge, you have a *hierarchical* clustering.

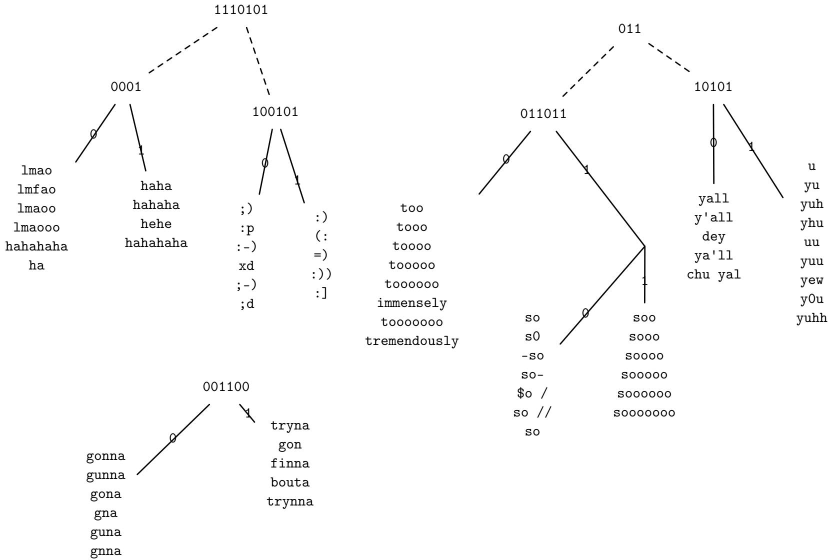
Each cluster is a binary tree with words at the leaves and internal nodes corresponding to merges.

Indexing the merge-pairs by 0 and 1 gives a bit-string for each word; prefixes of each word's bit string correspond to the hierarchical clusters it belongs to.

These can be seen as word embedings!

Brown Clusters from 56,000,000 Tweets

http://www.cs.cmu.edu/~ark/TweetNLP/cluster_viewer.html



Three Kinds of Cotext

If we consider a word token at a particular position i in text to be the observed value of a random variable X_i, what other random variables are predictive of/related to X_i?

1. the document containing i (a moderate-to-large collection of other words) $\longrightarrow$ topic models
2. the words that occur within a small "window" around i (e.g., x_{i-2}, x_{i-1}, x_{i+1}, x_{i+2}, or maybe the sentence containing i) $\longrightarrow$ distributional semantics
3. a sentence known to be a translation of the one containing i $\longrightarrow$ translation models

Bitext

Let $\boldsymbol{f}$ and $\boldsymbol{e}$ be two sequences in $\mathcal{V}^\dagger$ (French) and $\bar{\mathcal{V}}^\dagger$ (English), respectively.

We're going to define $p(\boldsymbol{F} \mid \boldsymbol{e})$, the probability over French translations of English sentence $\boldsymbol{e}$.

In a noisy channel machine translation system, we could use this together with source/language model $p(\boldsymbol{e})$ to "decode" $\boldsymbol{f}$ into an English translation.

Where does the data to estimate this come from?

IBM Model 2

(Brown et al., 1993)

Let ℓ and m be the (known) lengths of $\boldsymbol{e}$ and $\boldsymbol{f}$.

Latent variable $\boldsymbol{a} = \langle a_1, \ldots, a_m \rangle$, each a_i ranging over $\{0, \ldots, \ell\}$ (positions in $\boldsymbol{e}$).

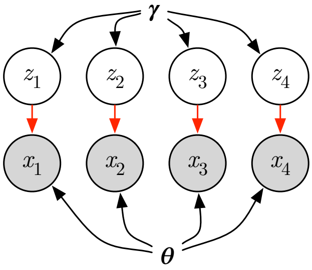
- E.g., $a_4 = 3$ means that f_4 is "aligned" to e_3.

$$p(\boldsymbol{f} \mid \boldsymbol{e}, m) = \sum_{\boldsymbol{a} \in \{0,\ldots,n\}^m} p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m)$$

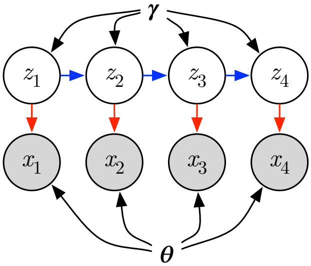
$$\begin{aligned} p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) &= \prod_{i=1}^{m} p(a_i \mid i, \ell, m) \cdot p(f_i \mid e_{a_i}) \\ &= \delta_{a_i \mid i, \ell, m} \cdot \theta_{f_i \mid e_{a_i}} \end{aligned}$$

IBM Model 2, Depicted

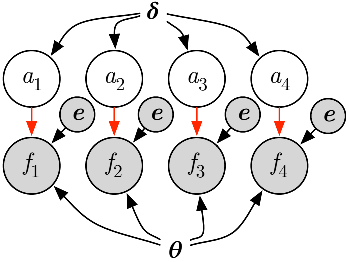
PLSA
and LDA
(topics)



hidden
Markov
model



IBM 2



Parameter Estimation

Use EM!

E step: calculate posteriors over all a_i, and then soft counts (left as an exercise: what soft counts do you need?)

M step: use relative frequency estimation from soft counts to get $\boldsymbol{\delta}$ and $\boldsymbol{\theta}$

Variations

- IBM Model 1 is the same, but fixes $\delta_{j|i,\ell,m} = \frac{1}{\ell+1}$.
  - Log-likelihood is convex!
  - Often used to initialize IBM Model 2.
- Dyer et al. (2013) introduced a new parameterization:

$$\delta_{j|i,\ell,m} \propto \exp -\lambda \left| \frac{i}{m} - \frac{j}{\ell} \right|$$

  (This is called `fast_align`.)
- IBM Models 3–5 (Brown et al., 1993) introduced increasingly more powerful ideas, such as "fertility" and "distortion."

Wow! That was a lot of models!

We covered:

- Topic models: LSI/A, PLSA, LDA
- Distributional semantics models: Skip-gram, Brown clustering
- Translation models: IBM 1 and 2

All of them are probabilistic models that capture patterns of cooccurrence between words and cotext.

They do *not* have: morphology (word-guts), syntax (sentence structure), or translation dictionaries . . .

References I

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155, 2003. URL `http://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf`.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Peter F. Brown, Peter V. Desouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.

Michael Collins. Statistical machine translation: IBM models 1 and 2, 2011. URL `http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/ibm12.pdf`.

Chris Dyer, Victor Chahuneau, and Noah A Smith. A simple, fast, and effective reparameterization of IBM Model 2. In *Proc. of NAACL*, 2013.

Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric P. Xing. A latent variable model for geographic lexical variation. In *Proc. of EMNLP*, 2010.

Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *NIPS*, 2014.

Percy Liang. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology, 2005.

References II

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of ICLR*, 2013a. URL `http://arxiv.org/pdf/1301.3781.pdf`.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013b. URL `http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf`.

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proc. of UAI*, 2004.

Karl Stratos, Do-kyum Kim, Michael Collins, and Daniel Hsu. A spectral algorithm for learning class-based n-gram models of natural language. In *Proc. of UAI*, 2014.