Bayesian Class-based Language Models

Yi Su

July 27, 2015

Motivation 1/2

- Why does Random Forest Language Model (RFLM) [Xu and Jelinek, 2004] work?
  - A collection of randomized Decision Tree Language Models
  - Interpolated with *equal* weights
  - ...which looks awfully like a typical day of Bayesian inference by sampling
- Hypothesis: RFLM works because it approximates a Bayesian model
  - The Bayesian model that it approximates should work even better

Motivation 2/2

- Can Latent Dirichlet Allocation (LDA) [Blei et al, 2003] be applied to language modeling?
  - Many have tried [Tam and Schultz, 2005 & 2006; Hsu and Glass 2006; Heidel et al, 2007; Liu and Liu, 2008]
  - But none feels "right"
    - Most of them focus on adaptation
    - The concept of "document" is forced onto the problem of language modeling
- Idea: consider all words following a history as a "document"
  - Then a "topic" is nothing but a word class!
  - LDA simply models a uni-gram distribution with a "mixture of multinomials"
  - Which is exactly the idea behind the class-based language model

What is a Class-based Language Model?

- A large family of language models which use word classes
- Intuition: similar words appear in similar context
- Most of them maintain a *hard-clustering* assumption
  - One word belongs to one class
- Definition

$$P(w \mid h) = P(c(w) \mid h)P(w \mid c(w), h) \tag{1}$$

- Terminology
  - Class model: $P(c \mid h)$
  - Word model: $P(w \mid c, h)$

Finding Classes

- Agglomerative clustering [Brown et al, 1992]
  - Start with one word per class
  - Iteratively combine two classes to increase the log likelihood
  - Until the desired number of classes is reached
- Exchange-based clustering [Martin et al, 1998]
  - Start with an initial assignment of words to classes
  - Iteratively move one word to another class to increase the log likelihood
  - Until a stopping criterion is met
- Random clustering [Emami and Jelinek, 2005]
  - Averaging many CLMs with word classes derived from randomly initialized exchange-based clustering gives better results.

$$P(w \mid h) = \frac{1}{K} \sum_{k=1}^{K} P_k(c_k(w) \mid h) P_k(w \mid c_k(w), h), \quad (2)$$

A Bayesian Formulation of CLMs: First Attempt

- Generative process: for every word w_i and its history h_i
  - Generate c_i:

$$c_i \mid h_i \sim \mathrm{Mult}(G_{h_i}(c)) \tag{3}$$

  - Generate w_i:

$$w_i \mid c_i, h_i \sim \mathrm{Mult}(H_{c_i h_i}(w)) \tag{4}$$

- Predictive probability

$$P(w \mid h) = \sum_{c \in \mathcal{C}} P(c \mid h) P(w \mid c, h) \tag{5}$$

- Bayesian predictive probability

$$P(w \mid h) \;\; = \;\; \sum_{A \in \mathcal{A}} P(A) \sum_{c \in \mathcal{C}} P_A(c \mid h) P_A(w \mid c, h) \tag{6}$$

where A denote a class assignment of all words in the training text

Soft-clustering Class-based Language Models

- Effectively replaced the hard-clustering assumption with *soft-clustering*
  - One word belongs to any class with certain probability
- Inference
  - Model does not subscribe itself to any inference algorithm
  - Inference by sampling

$$\begin{aligned} P(w \mid h) &= \sum_{A \in \mathcal{A}} P(A) \sum_{c \in \mathcal{C}} P_A(c \mid h) P_A(w \mid c, h) \\ &\approx \frac{1}{K} \sum_{k=1}^{K} \sum_{c \in \mathcal{C}} P_{A_k}(c \mid h) P_{A_k}(w \mid c, h) \end{aligned} \quad (7)$$

Sampling

- Collapsed Gibbs sampler, a Markov Chain Monte Carlo method, comes natural to this model
  - For every class assignment c_i of the word w_i
    - Given everything else, compute its distribution by

$$P(c_i = j \mid \mathbf{c}_{\neg i}, \mathbf{w}) \propto P(c_i = j \mid \mathbf{c}_{\neg i}, \mathbf{w}_{\neg i}) \cdot P(w_i \mid c_i = j, \mathbf{c}_{\neg i}, \mathbf{w}_{\neg i}) \quad (8)$$

    - Sample c_i from this newly computed distribution
  - Repeat above until the assignment reaches steady state
- Given a complete class assignment of all words in the training text, both class and word models are just regular *n*-gram language models!
- Two terms on the right hand side can be computed with "leave-one-out" versions of class and word models, respectively
- Taking multiple samples is trivially parallelizable

Hierarchical Pitman-Yor Language Models

- Previous model is still not "Bayesian enough"
  - After sampling, class and word models are both built with Kneser-Ney (KN) smoothing
  - We need a Bayesian version of KN smoothing
- Smoothing is to frequentists as prior is to Bayesians
  - Hierarchical Pitman-Yor (HPY) prior is the Bayesian counterpart of KN smoothing [Teh, 2006]
  - Or KN smoothing is a frequentist approximation of HPY
- Plug and play...

A Bayesian Formulation of CLMs: Second Attempt

- For every word w_i and its history $h_i = w_{i-1} \cdots w_{i-n+1}$:
  - Generate c_i:

$$\begin{aligned} G_\phi(c) &\sim \mathrm{PY}(d_0, \theta_0, G_0(c)) \\ G_{w_{i-1}}(c) &\sim \mathrm{PY}(d_1, \theta_1, G_\phi(c)) \\ &\cdots \\ c_i \mid h_i &\sim \mathrm{Mult}(G_{h_i}(c)), \end{aligned} \tag{9}$$

  - Generate w_i:

$$\begin{aligned} H_\phi(w) &\sim \mathrm{PY}(e_0, \mu_0, H_0(w)) \\ H_{c_i}(w) &\sim \mathrm{PY}(e_1, \mu_1, H_\phi(w)) \\ H_{c_i w_{i-1}}(w) &\sim \mathrm{PY}(e_2, \mu_2, H_{c_i}(w)) \\ &\cdots \\ w_i \mid c_i, h_i &\sim \mathrm{Mult}(H_{c_i h_i}(w)). \end{aligned} \tag{10}$$

Soft-clustering Class-based Hierarchical Pitman-Yor LMs

- For completeness,
  - Choose parameters d_j and θ_j for $j \in \{0, 1, \cdots, n-1\}$:

$$\begin{aligned} d_j &\sim \text{Beta}(1,1) \\ \theta_j &\sim \text{Gamma}(1,1) \end{aligned} \tag{11}$$

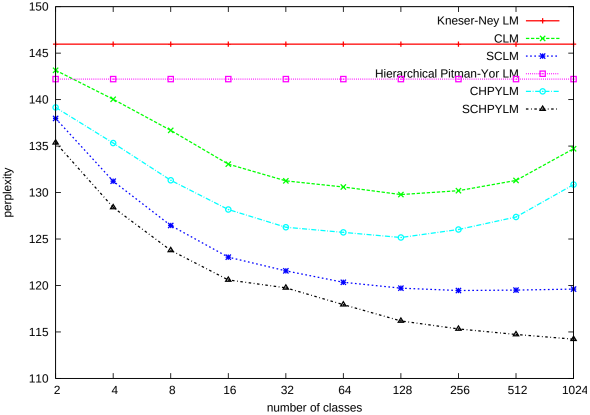
  - Choose parameters e_j and μ_j for $j \in \{0, 1, \cdots, n\}$:

$$\begin{aligned} e_j &\sim \text{Beta}(1,1) \\ \mu_j &\sim \text{Gamma}(1,1) \end{aligned} \tag{12}$$

- Inference by sampling
  - Even the sampler is plug and play
    - Sample class assignment given class and word models
    - Sample class model given class assignment and word model
    - Sample word model given class assignment and class model

Perplexity

- Setup
  - 1M words Wall Street Journal (WSJ), 10K vocabulary, 3-grams
  - 100 samples per experiment, 800 iterations of burn-in

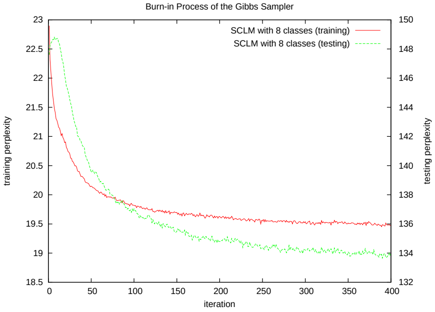


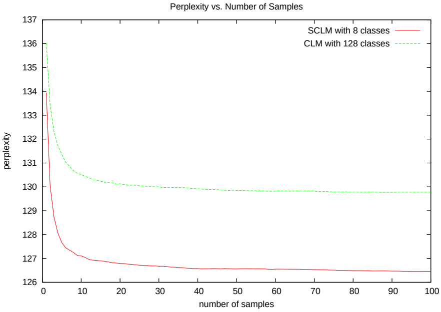
Word Error Rate

- Setup
  - Lattice-rescoring
  - IBM 2004 Rich Transcription Conversational Telephony Speech system
  - 20M words Fisher data, 30K vocabulary, 4-grams
  - Interpolation with a big LM built from many other sources

Model	**w/o Interp.**	**w/ Interp.**
Kneser-Ney	14.4	13.5
CLM	14.2	13.4
SCLM	13.7	13.3
Hier. Pitman-Yor	14.1	13.4
CHPYLM	13.7	13.2
SCHPYLM	13.5	13.1

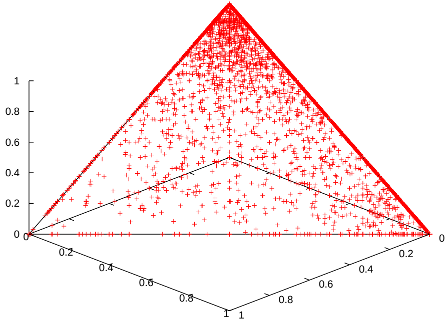
Burn-in and Mixing

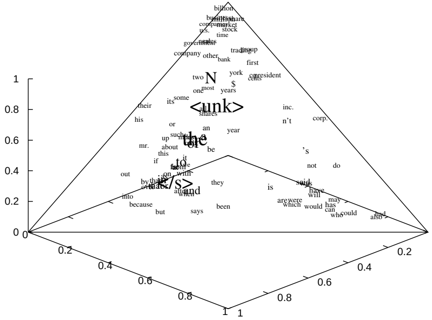




Probabilistic Word Embeddings

$$P(c \mid w) = \frac{\sum_{h \in \mathcal{L}} P(c \mid h) P(w \mid c, h) P(h)}{P(w)} \tag{13}$$





Conclusions

- We proposed
  - Two Bayesian class-based LMs which naturally support soft-clustering
  - A simple collapsed Gibbs sampler for inference
- Great performance in perplexity and word error rate
  - 22% perplexity reduction on WSJ
  - 6% WER reduction on IBM RT-04 CTS
- Model averaging is a powerful idea
  - Either frequentist (RFLM, random clustering) or Bayesian (our models)
  - Good ideas converge ☺

Thank you!