A Hierarchical Bayesian Language Model based on Pitman-Yor Processes

Yee Whye Teh
School of Computing,
National University of Singapore,
3 Science Drive 2, Singapore 117543.
tehyw@comp.nus.edu.sg

Abstract

We propose a new hierarchical Bayesian n-gram model of natural languages. Our model makes use of a generalization of the commonly used Dirichlet distributions called Pitman-Yor processes which produce power-law distributions more closely resembling those in natural languages. We show that an approximation to the hierarchical Pitman-Yor language model recovers the exact formulation of interpolated Kneser-Ney, one of the best smoothing methods for n-gram language models. Experiments verify that our model gives cross entropy results superior to interpolated Kneser-Ney and comparable to modified Kneser-Ney.

1 Introduction

Probabilistic language models are used extensively in a variety of linguistic applications, including speech recognition, handwriting recognition, optical character recognition, and machine translation. Most language models fall into the class of n-gram models, which approximate the distribution over sentences using the conditional distribution of each word given a context consisting of only the previous $n-1$ words,

$$P(\text{sentence}) \approx \prod_{i=1}^{T} P(\text{word}_i \,|\, \text{word}_{i-n+1}^{i-1}) \quad (1)$$

with $n = 3$ (trigram models) being typical. Even for such a modest value of n the number of parameters is still tremendous due to the large vocabulary size. As a result direct maximum-likelihood parameter fitting severely overfits to the training data, and smoothing methods are indispensible for proper training of n-gram models.

A large number of smoothing methods have been proposed in the literature (see (Chen and Goodman, 1998; Goodman, 2001; Rosenfeld, 2000) for good overviews). Most methods take a rather ad hoc approach, where n-gram probabilities for various values of n are combined together, using either interpolation or back-off schemes. Though some of these methods are intuitively appealing, the main justification has always been empirical—better perplexities or error rates on test data. Though arguably this should be the only real justification, it only answers the question of *whether* a method performs better, not *how* nor *why* it performs better. This is unavoidable given that most of these methods are not based on internally coherent Bayesian probabilistic models, which have explicitly declared prior assumptions and whose merits can be argued in terms of how closely these fit in with the known properties of natural languages. Bayesian probabilistic models also have additional advantages—it is relatively straightforward to improve these models by incorporating additional knowledge sources and to include them in larger models in a principled manner. Unfortunately the performance of previously proposed Bayesian language models had been dismal compared to other smoothing methods (Nadas, 1984; MacKay and Peto, 1994).

In this paper, we propose a novel language model based on a hierarchical Bayesian model (Gelman et al., 1995) where each hidden variable is distributed according to a Pitman-Yor process, a nonparametric generalization of the Dirichlet distribution that is widely studied in the statistics and probability theory communities (Pitman and Yor, 1997; Ishwaran and James, 2001; Pitman, 2002).

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 985–992,
Sydney, July 2006.

Our model is a direct generalization of the hierarchical Dirichlet language model of (MacKay and Peto, 1994). Inference in our model is however not as straightforward and we propose an efficient Markov chain Monte Carlo sampling scheme.

Pitman-Yor processes produce power-law distributions that more closely resemble those seen in natural languages, and it has been argued that as a result they are more suited to applications in natural language processing (Goldwater et al., 2006). We show experimentally that our hierarchical Pitman-Yor language model does indeed produce results superior to interpolated Kneser-Ney and comparable to modified Kneser-Ney, two of the currently best performing smoothing methods (Chen and Goodman, 1998). In fact we show a stronger result—that interpolated Kneser-Ney can be interpreted as a particular approximate inference scheme in the hierarchical Pitman-Yor language model. Our interpretation is more useful than past interpretations involving marginal constraints (Kneser and Ney, 1995; Chen and Goodman, 1998) or maximum-entropy models (Goodman, 2004) as it can recover the exact formulation of interpolated Kneser-Ney, and actually produces superior results. (Goldwater et al., 2006) has independently noted the correspondence between the hierarchical Pitman-Yor language model and interpolated Kneser-Ney, and conjectured improved performance in the hierarchical Pitman-Yor language model, which we verify here.

Thus the contributions of this paper are threefold: in proposing a langauge model with excellent performance and the accompanying advantages of Bayesian probabilistic models, in proposing a novel and efficient inference scheme for the model, and in establishing the direct correspondence between interpolated Kneser-Ney and the Bayesian approach.

We describe the Pitman-Yor process in Section 2, and propose the hierarchical Pitman-Yor language model in Section 3. In Sections 4 and 5 we give a high level description of our sampling based inference scheme, leaving the details to a technical report (Teh, 2006). We also show how interpolated Kneser-Ney can be interpreted as approximate inference in the model. We show experimental comparisons to interpolated and modified Kneser-Ney, and the hierarchical Dirichlet language model in Section 6 and conclude in Section 7.

2 Pitman-Yor Process

Pitman-Yor processes are examples of nonparametric Bayesian models. Here we give a quick description of the Pitman-Yor process in the context of a unigram language model; good tutorials on such models are provided in (Ghahramani, 2005; Jordan, 2005). Let W be a fixed and finite vocabulary of V words. For each word $w \in W$ let $G(w)$ be the (to be estimated) probability of w, and let $G = [G(w)]_{w \in W}$ be the vector of word probabilities. We place a Pitman-Yor process prior on G:

$$G \sim \mathrm{PY}(d, \theta, G_0) \tag{2}$$

where the three parameters are: a discount parameter $0 \leq d < 1$, a strength parameter $\theta > -d$ and a mean vector $G_0 = [G_0(w)]_{w \in W}$. $G_0(w)$ is the a priori probability of word w: before observing any data, we believe word w should occur with probability $G_0(w)$. In practice this is usually set uniformly $G_0(w) = 1/V$ for all $w \in W$. Both θ and d can be understood as controlling the amount of variability around G_0 in different ways. When $d = 0$ the Pitman-Yor process reduces to a Dirichlet distribution with parameters θG_0.

There is in general no known analytic form for the density of $\mathrm{PY}(d, \theta, G_0)$ when the vocabulary is finite. However this need not deter us as we will instead work with the distribution over sequences of words induced by the Pitman-Yor process, which has a nice tractable form and is sufficient for our purpose of language modelling. To be precise, notice that we can treat both G and G_0 as distributions over W, where word $w \in W$ has probability $G(w)$ (respectively $G_0(w)$). Let $x_1, x_2, \ldots$ be a sequence of words drawn independently and identically (i.i.d.) from G. We shall describe the Pitman-Yor process in terms of a generative procedure that produces $x_1, x_2, \ldots$ iteratively with G marginalized out. This can be achieved by relating $x_1, x_2, \ldots$ to a separate sequence of i.i.d. draws $y_1, y_2, \ldots$ from the mean distribution G_0 as follows. The first word x_1 is assigned the value of the first draw y_1 from G_0. Let t be the current number of draws from G_0 (currently $t = 1$), c_k be the number of words assigned the value of draw y_k (currently $c_1 = 1$), and $c_\cdot = \sum_{k=1}^{t} c_k$ be the current number of draws from G. For each subsequent word $x_{c_\cdot+1}$, we either assign it the value of a previous draw y_k with probability $\frac{c_k - d}{\theta + c_\cdot}$ (increment c_k; set $x_{c_\cdot+1} \leftarrow y_k$), or we assign it the value of a new draw from G_0

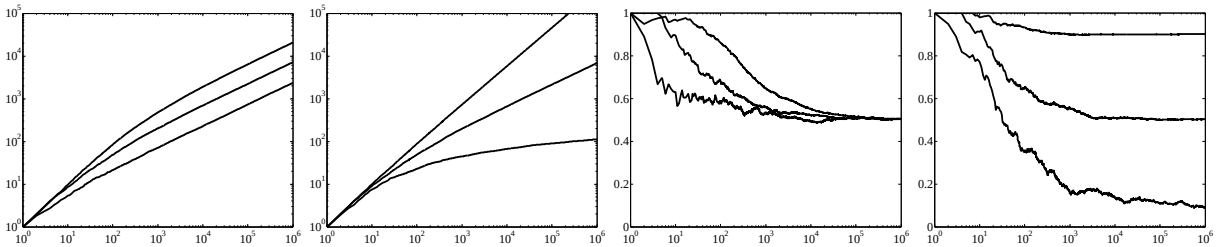

Figure 1: First panel: number of unique words as a function of the number of words drawn on a log-log scale, with $d = .5$ and $\theta = 1$ (bottom), 10 (middle) and 100 (top). Second panel: same, with $\theta = 10$ and $d = 0$ (bottom), .5 (middle) and .9 (top). Third panel: proportion of words appearing only once, as a function of the number of words drawn, with $d = .5$ and $\theta = 1$ (bottom), 10 (middle), 100 (top). Last panel: same, with $\theta = 10$ and $d = 0$ (bottom), .5 (middle) and .9 (top).

with probability $\frac{\theta+dt}{\theta+c.}$ (increment t; set $c_t = 1$; draw $y_t \sim G_0$; set $x_{c.+1} \leftarrow y_t$).

The above generative procedure produces a sequence of words drawn i.i.d. from G, with G marginalized out. It is informative to study the Pitman-Yor process in terms of the behaviour it induces on this sequence of words. Firstly, notice the rich-gets-richer clustering property: the more words have been assigned to a draw from G_0, the more likely subsequent words will be assigned to the draw. Secondly, the more we draw from G_0, the more likely a new word will be assigned to a new draw from G_0. These two effects together produce a power-law distribution where many unique words are observed, most of them rarely. In particular, for a vocabulary of unbounded size and for $d > 0$, the number of unique words scales as $O(\theta T^d)$ where T is the total number of words. For $d = 0$, we have a Dirichlet distribution and the number of unique words grows more slowly at $O(\theta \log T)$.

Figure 1 demonstrates the power-law behaviour of the Pitman-Yor process and how this depends on d and θ. In the first two panels we show the average number of unique words among 10 sequences of T words drawn from G, as a function of T, for various values of θ and d. We see that θ controls the overall number of unique words, while d controls the asymptotic growth of the number of unique words. In the last two panels, we show the proportion of words appearing only once among the unique words; this gives an indication of the proportion of words that occur rarely. We see that the asymptotic behaviour depends on d but not on θ, with larger d's producing more rare words.

This procedure for generating words drawn from G is often referred to as the Chinese restaurant process (Pitman, 2002). The metaphor is as follows. Consider a sequence of customers (corresponding to the words draws from G) visiting a Chinese restaurant with an unbounded number of tables (corresponding to the draws from G_0), each of which can accommodate an unbounded number of customers. The first customer sits at the first table, and each subsequent customer either joins an already occupied table (assign the word to the corresponding draw from G_0), or sits at a new table (assign the word to a new draw from G_0).

3 Hierarchical Pitman-Yor Language Models

We describe an n-gram language model based on a hierarchical extension of the Pitman-Yor process. An n-gram language model defines probabilities over the current word given various contexts consisting of up to $n-1$ words. Given a context $\mathbf{u}$, let $G_{\mathbf{u}}(w)$ be the probability of the current word taking on value w. We use a Pitman-Yor process as the prior for $G_{\mathbf{u}}[G_{\mathbf{u}}(w)]_{w \in W}$, in particular,

$$G_{\mathbf{u}} \sim \mathrm{PY}(d_{|\mathbf{u}|}, \theta_{|\mathbf{u}|}, G_{\pi(\mathbf{u})}) \qquad (3)$$

where $\pi(\mathbf{u})$ is the suffix of $\mathbf{u}$ consisting of all but the earliest word. The strength and discount parameters are functions of the length $|\mathbf{u}|$ of the context, while the mean vector is $G_{\pi(\mathbf{u})}$, the vector of probabilities of the current word given all but the earliest word in the context. Since we do not know $G_{\pi(\mathbf{u})}$ either, We recursively place a prior over $G_{\pi(\mathbf{u})}$ using (3), but now with parameters $\theta_{|\pi(\mathbf{u})|}, d_{|\pi(\mathbf{u})|}$ and mean vector $G_{\pi(\pi(\mathbf{u}))}$ instead. This is repeated until we get to $G_\emptyset$, the vector of probabilities over the current word given the

empty context $\emptyset$. Finally we place a prior on $G_\emptyset$:

$$G_\emptyset \sim \mathrm{PY}(d_0, \theta_0, G_0) \tag{4}$$

where G_0 is the global mean vector, given a uniform value of $G_0(w) = 1/V$ for all $w \in W$. Finally, we place a uniform prior on the discount parameters and a $\mathrm{Gamma}(1,1)$ prior on the strength parameters. The total number of parameters in the model is $2n$.

The structure of the prior is that of a suffix tree of depth n, where each node corresponds to a context consisting of up to $n-1$ words, and each child corresponds to adding a different word to the beginning of the context. This choice of the prior structure expresses our belief that words appearing earlier in a context have (a priori) the least importance in modelling the probability of the current word, which is why they are dropped first at successively higher levels of the model.

4 Hierarchical Chinese Restaurant Processes

We describe a generative procedure analogous to the Chinese restaurant process of Section 2 for drawing words from the hierarchical Pitman-Yor language model with all $G_{\mathbf{u}}$'s marginalized out. This gives us an alternative representation of the hierarchical Pitman-Yor language model that is amenable to efficient inference using Markov chain Monte Carlo sampling and easy computation of the predictive probabilities for test words. The correspondence between interpolated Kneser-Ney and the hierarchical Pitman-Yor language model is also apparent in this representation.

Again we may treat each $G_{\mathbf{u}}$ as a distribution over the current word. The basic observation is that since $G_{\mathbf{u}}$ is Pitman-Yor process distributed, we can draw words from it using the Chinese restaurant process given in Section 2. Further, the only operation we need of its parent distribution $G_{\pi(\mathbf{u})}$ is to draw words from it too. Since $G_{\pi(\mathbf{u})}$ is itself distributed according to a Pitman-Yor process, we can use another Chinese restaurant process to draw words from that. This is recursively applied until we need draws from the global mean distribution G_0, which is easy since it is just uniform. We refer to this as the hierarchical Chinese restaurant process.

Let us introduce some notations. For each context $\mathbf{u}$ we have a sequence of words $x_{\mathbf{u}1}, x_{\mathbf{u}2}, \ldots$ drawn i.i.d. from $G_{\mathbf{u}}$ and another sequence of words $y_{\mathbf{u}1}, y_{\mathbf{u}2}, \ldots$ drawn i.i.d. from the parent distribution $G_{\pi(\mathbf{u})}$. We use l to index draws from $G_{\mathbf{u}}$ and k to index the draws from $G_{\pi(\mathbf{u})}$. Define $t_{\mathbf{u}wk} = 1$ if $y_{\mathbf{u}k}$ takes on value w, and $t_{\mathbf{u}wk} = 0$ otherwise. Each word $x_{\mathbf{u}l}$ is assigned to one of the draws $y_{\mathbf{u}k}$ from $G_{\pi(\mathbf{u})}$. If $y_{\mathbf{u}k}$ takes on value w define $c_{\mathbf{u}wk}$ as the number of words $x_{\mathbf{u}l}$ drawn from $G_{\mathbf{u}}$ assigned to $y_{\mathbf{u}k}$, otherwise let $c_{\mathbf{u}wk} = 0$. Finally we denote marginal counts by dots. For example, $c_{\mathbf{u}\cdot k}$ is the number of $x_{\mathbf{u}l}$'s assigned the value of $y_{\mathbf{u}k}$, $c_{\mathbf{u}w\cdot}$ is the number of $x_{\mathbf{u}l}$'s with value w, and $t_{\mathbf{u}\cdot\cdot}$ is the current number of draws $y_{\mathbf{u}k}$ from $G_{\pi(\mathbf{u})}$. Notice that we have the following relationships among the $c_{\mathbf{u}w\cdot}$'s and $t_{\mathbf{u}w\cdot}$:

$$\begin{cases} t_{\mathbf{u}w\cdot} = 0 & \text{if } c_{\mathbf{u}w\cdot} = 0; \\ 1 \le t_{\mathbf{u}w\cdot} \le c_{\mathbf{u}w\cdot} & \text{if } c_{\mathbf{u}w\cdot} > 0; \end{cases} \tag{5}$$

$$c_{\mathbf{u}w\cdot} = \sum_{\mathbf{u}':\pi(\mathbf{u}')=\mathbf{u}} t_{\mathbf{u}'w\cdot} \tag{6}$$

Pseudo-code for drawing words using the hierarchical Chinese restaurant process is given as a recursive function DrawWord($\mathbf{u}$), while pseudo-code for computing the probability that the next word drawn from $G_{\mathbf{u}}$ will be w is given in WordProb($\mathbf{u}$,w). The counts are initialized at all $c_{\mathbf{u}wk} = t_{\mathbf{u}wk} = 0$.

Function DrawWord($\mathbf{u}$):
Returns a new word drawn from $G_{\mathbf{u}}$.
If $\mathbf{u} = 0$, return $w \in W$ with probability $G_0(w)$.
Else with probabilities proportional to:
$c_{\mathbf{u}wk} - d_{|\mathbf{u}|} t_{\mathbf{u}wk}$: assign the new word to $y_{\mathbf{u}k}$.
Increment $c_{\mathbf{u}wk}$; return w.
$\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|} t_{\mathbf{u}\cdot\cdot}$: assign the new word to a new
draw $y_{\mathbf{u}k^{\text{new}}}$ from $G_{\pi(\mathbf{u})}$.
Let $w \leftarrow$ DrawWord($\pi(\mathbf{u})$);
set $t_{\mathbf{u}wk^{\text{new}}} = c_{\mathbf{u}wk^{\text{new}}} = 1$; return w.

Function WordProb($\mathbf{u}$,w):
Returns the probability that the next word after context $\mathbf{u}$ will be w.
If $\mathbf{u} = 0$, return $G_0(w)$. Else return

$$\frac{c_{\mathbf{u}w\cdot} - d_{|\mathbf{u}|} t_{\mathbf{u}w\cdot}}{\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} + \frac{\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|} t_{\mathbf{u}\cdot\cdot}}{\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} \text{WordProb}(\pi(\mathbf{u}),w).$$

Notice the self-reinforcing property of the hierarchical Pitman-Yor language model: the more a word w has been drawn in context $\mathbf{u}$, the more likely will we draw w again in context $\mathbf{u}$. In fact word w will be reinforced for other contexts that share a common suffix with $\mathbf{u}$, with the probability of drawing w increasing as the length of the

common suffix increases. This is because w will be more likely under the context of the common suffix as well.

The hierarchical Chinese restaurant process is equivalent to the hierarchical Pitman-Yor language model insofar as the distribution induced on words drawn from them are exactly equal. However, the probability vectors $G_{\mathbf{u}}$'s have been marginalized out in the procedure, replaced instead by the assignments of words $x_{\mathbf{u}l}$ to draws $y_{\mathbf{u}k}$ from the parent distribution, i.e. the seating arrangement of customers around tables in the Chinese restaurant process corresponding to $G_{\mathbf{u}}$. In the next section we derive tractable inference schemes for the hierarchical Pitman-Yor language model based on these seating arrangements.

5 Inference Schemes

In this section we give a high level description of a Markov chain Monte Carlo sampling based inference scheme for the hierarchical Pitman-Yor language model. Further details can be obtained at (Teh, 2006). We also relate interpolated Kneser-Ney to the hierarchical Pitman-Yor language model.

Our training data $\mathcal{D}$ consists of the number of occurrences $c_{\mathbf{u}w\cdot}$ of each word w after each context $\mathbf{u}$ of length exactly $n-1$. This corresponds to observing word w drawn $c_{\mathbf{u}w\cdot}$ times from $G_{\mathbf{u}}$. Given the training data $\mathcal{D}$, we are interested in the posterior distribution over the latent vectors $\mathcal{G} = \{G_{\mathbf{v}} : \text{all contexts } \mathbf{v}\}$ and parameters $\mathbf{\Theta} = \{\theta_m, d_m : 0 \leq m \leq n-1\}$:

$$p(\mathcal{G}, \mathbf{\Theta}|\mathcal{D}) = p(\mathcal{G}, \mathbf{\Theta}, \mathcal{D})/p(\mathcal{D}) \tag{7}$$

As mentioned previously, the hierarchical Chinese restaurant process marginalizes out each $G_{\mathbf{u}}$, replacing it with the seating arrangement in the corresponding restaurant, which we shall denote by $S_{\mathbf{u}}$. Let $\mathcal{S} = \{S_{\mathbf{v}} : \text{all contexts } \mathbf{v}\}$. We are thus interested in the equivalent posterior over seating arrangements instead:

$$p(\mathcal{S}, \mathbf{\Theta}|\mathcal{D}) = p(\mathcal{S}, \mathbf{\Theta}, \mathcal{D})/p(\mathcal{D}) \tag{8}$$

The most important quantities we need for language modelling are the predictive probabilities: what is the probability of a test word w after a context $\mathbf{u}$? This is given by

$$p(w|\mathbf{u}, \mathcal{D}) = \int p(w|\mathbf{u}, \mathcal{S}, \mathbf{\Theta}) p(\mathcal{S}, \mathbf{\Theta}|\mathcal{D})\, d(\mathcal{S}, \mathbf{\Theta}) \tag{9}$$

where the first probability on the right is the predictive probability under a particular setting of seating arrangements $\mathcal{S}$ and parameters $\mathbf{\Theta}$, and the overall predictive probability is obtained by averaging this with respect to the posterior over $\mathcal{S}$ and $\mathbf{\Theta}$ (second probability on right). We approximate the integral with samples $\{\mathcal{S}^{(i)}, \mathbf{\Theta}^{(i)}\}_{i=1}^{I}$ drawn from $p(\mathcal{S}, \mathbf{\Theta}|\mathcal{D})$:

$$p(w|\mathbf{u}, \mathcal{D}) \approx \sum_{i=1}^{I} p(w|\mathbf{u}, \mathcal{S}^{(i)}, \mathbf{\Theta}^{(i)}) \tag{10}$$

while $p(w|\mathbf{u}, \mathcal{S}, \mathbf{\Theta})$ is given by the function WordProb($\mathbf{u}$,w):

$$p(w \mid 0, \mathcal{S}, \mathbf{\Theta}) = 1/V \tag{11}$$

$$p(w \mid \mathbf{u}, \mathcal{S}, \mathbf{\Theta}) = \frac{c_{\mathbf{u}w\cdot} - d_{|\mathbf{u}|} t_{\mathbf{u}w\cdot}}{\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} + \frac{\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|} t_{\mathbf{u}\cdot\cdot}}{\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} p(w \mid \pi(\mathbf{u}), \mathcal{S}, \mathbf{\Theta}) \tag{12}$$

where the counts are obtained from the seating arrangement $\mathcal{S}_{\mathbf{u}}$ in the Chinese restaurant process corresponding to $G_{\mathbf{u}}$.

We use Gibbs sampling to obtain the posterior samples $\{\mathcal{S}, \mathbf{\Theta}\}$ (Neal, 1993). Gibbs sampling keeps track of the current state of each variable of interest in the model, and iteratively resamples the state of each variable given the current states of all other variables. It can be shown that the states of variables will converge to the required samples from the posterior distribution after a sufficient number of iterations. Specifically for the hierarchical Pitman-Yor language model, the variables consist of, for each $\mathbf{u}$ and each word $x_{\mathbf{u}l}$ drawn from $G_{\mathbf{u}}$, the index $k_{\mathbf{u}l}$ of the draw from $G_{\pi(\mathbf{u})}$ assigned $x_{\mathbf{u}l}$. In the Chinese restaurant metaphor, this is the index of the table which the lth customer sat at in the restaurant corresponding to $G_{\mathbf{u}}$. If $x_{\mathbf{u}l}$ has value w, it can only be assigned to draws from $G_{\pi(\mathbf{u})}$ that has value w as well. This can either be a preexisting draw with value w, or it can be a new draw taking on value w. The relevant probabilities are given in the functions DrawWord($\mathbf{u}$) and WordProb($\mathbf{u}$,w), where we treat $x_{\mathbf{u}l}$ as the last word drawn from $G_{\mathbf{u}}$. This gives:

$$p(k_{\mathbf{u}l} = k|\mathcal{S}^{-\mathbf{u}l}, \mathbf{\Theta}) \propto \frac{\max(0, c^{-\mathbf{u}l}_{\mathbf{u}x_{\mathbf{u}l}k} - d)}{\theta + c^{-\mathbf{u}l}_{\mathbf{u}\cdot\cdot}} \tag{13}$$

$$p(k_{\mathbf{u}l} = k^{\text{new}} \text{ with } y_{\mathbf{u}k^{\text{new}}} = x_{\mathbf{u}l}|\mathcal{S}^{-\mathbf{u}l}, \mathbf{\Theta}) \propto \frac{\theta + d t^{-\mathbf{u}l}_{\mathbf{u}\cdot\cdot}}{\theta + c^{-\mathbf{u}l}_{\mathbf{u}\cdot\cdot}} p(x_{\mathbf{u}l}|\pi(\mathbf{u}), \mathcal{S}^{-\mathbf{u}l}, \mathbf{\Theta}) \tag{14}$$

where the superscript $-\mathbf{u}l$ means the corresponding set of variables or counts with $x_{\mathbf{u}l}$ excluded. The parameters $\mathbf{\Theta}$ are sampled using an auxiliary variable sampler as detailed in (Teh, 2006). The overall sampling scheme for an n-gram hierarchical Pitman-Yor language model takes $O(nT)$ time and requires $O(M)$ space per iteration, where T is the number of words in the training set, and M is the number of unique n-grams. During test time, the computational cost is $O(nI)$, since the predictive probabilities (12) require $O(n)$ time to calculate for each of I samples.

The hierarchical Pitman-Yor language model produces discounts that grow gradually as a function of n-gram counts. Notice that although each Pitman-Yor process $G_{\mathbf{u}}$ only has one discount *parameter*, the predictive probabilities (12) produce different discount *values* since $t_{\mathbf{u}w\cdot}$ can take on different values for different words w. In fact $t_{\mathbf{u}w\cdot}$ will on average be larger if $c_{\mathbf{u}w\cdot}$ is larger; averaged over the posterior, the actual amount of discount will grow slowly as the count $c_{\mathbf{u}w\cdot}$ grows. This is shown in Figure 2 (left), where we see that the growth of discounts is sublinear.

The correspondence to interpolated Kneser-Ney is now straightforward. If we restrict $t_{\mathbf{u}w\cdot}$ to be at most 1, that is,

$$t_{\mathbf{u}w\cdot} = \min(1, c_{\mathbf{u}w\cdot}) \tag{15}$$

$$c_{\mathbf{u}w\cdot} = \sum_{\mathbf{u}':\pi(\mathbf{u}')=\mathbf{u}} t_{\mathbf{u}'w\cdot} \tag{16}$$

we will get the same discount value so long as $c_{\mathbf{u}w\cdot} > 0$, i.e. absolute discounting. Further supposing that the strength parameters are all $\theta_{|\mathbf{u}|} = 0$, the predictive probabilities (12) now directly reduces to the predictive probabilities given by interpolated Kneser-Ney. Thus we can interpret interpolated Kneser-Ney as the approximate inference scheme (15,16) in the hierarchical Pitman-Yor language model.

Modified Kneser-Ney uses the same values for the counts as in (15,16), but uses a different valued discount for each value of $c_{\mathbf{u}w\cdot}$ up to a maximum of $c^{(\max)}$. Since the discounts in a hierarchical Pitman-Yor language model are limited to between 0 and 1, we see that modified Kneser-Ney is not an approximation of the hierarchical Pitman-Yor language model.

6 Experimental Results

We performed experiments on the hierarchical Pitman-Yor language model on a 16 million word corpus derived from APNews. This is the same dataset as in (Bengio et al., 2003). The training, validation and test sets consist of about 14 million, 1 million and 1 million words respectively, while the vocabulary size is 17964. For trigrams with $n = 3$, we varied the training set size between approximately 2 million and 14 million words by six equal increments, while we also experimented with $n = 2$ and 4 on the full 14 million word training set. We compared the hierarchical Pitman-Yor language model trained using the proposed Gibbs sampler (HPYLM) against interpolated Kneser-Ney (IKN), modified Kneser-Ney (MKN) with maximum discount cut-off $c^{(\max)} = 3$ as recommended in (Chen and Goodman, 1998), and the hierarchical Dirichlet language model (HDLM).

For the various variants of Kneser-Ney, we first determined the parameters by conjugate gradient descent in the cross-entropy on the validation set. At the optimal values, we folded the validation set into the training set to obtain the final n-gram probability estimates. This procedure is as recommended in (Chen and Goodman, 1998), and takes approximately 10 minutes on the full training set with $n = 3$ on a 1.4 Ghz PIII. For HPYLM we inferred the posterior distribution over the latent variables and parameters given both the training and validation sets using the proposed Gibbs sampler. Since the posterior is well-behaved and the sampler converges quickly, we only used 125 iterations for burn-in, and 175 iterations to collect posterior samples. On the full training set with $n = 3$ this took about 1.5 hours.

Perplexities on the test set are given in Table 1. As expected, HDLM gives the worst performance, while HPYLM performs better than IKN. Perhaps surprisingly HPYLM performs slightly worse than MKN. We believe this is because HPYLM is not a perfect model for languages and as a result posterior estimates of the parameters are not optimized for predictive performance. On the other hand parameters in the Kneser-Ney variants are optimized using cross-validation, so are given optimal values for prediction. To validate this conjecture, we also experimented with HPYCV, a hierarchical Pitman-Yor language model where the parameters are obtained by fitting them in a slight generalization of IKN where the strength param-

T	n	IKN	MKN	HPYLM	HPYCV	HDLM
2e6	3	148.8	**144.1**	145.7	144.3	191.2
4e6	3	137.1	**132.7**	134.3	**132.7**	172.7
6e6	3	130.6	126.7	127.9	**126.4**	162.3
8e6	3	125.9	122.3	123.2	**121.9**	154.7
10e6	3	122.0	118.6	119.4	**118.2**	148.7
12e6	3	119.0	115.8	116.5	**115.4**	144.0
14e6	3	116.7	113.6	114.3	**113.2**	140.5
14e6	2	169.9	**169.2**	169.6	169.3	180.6
14e6	4	106.1	102.4	103.8	**101.9**	136.6

Table 1: Perplexities of various methods and for various sizes of training set T and length of n-grams.

eters $\theta_{|\mathbf{u}|}$'s are allowed to be positive and optimized over along with the discount parameters using cross-validation. Seating arrangements are Gibbs sampled as in Section 5 with the parameter values fixed. We find that HPYCV performs better than MKN (except marginally worse on small problems), and has best performance overall. Note that the parameter values in HPYCV are still not the optimal ones since they are obtained by cross-validation using IKN, an approximation to a hierarchical Pitman-Yor language model. Unfortunately cross-validation using a hierarchical Pitman-Yor language model inferred using Gibbs sampling is currently too costly to be practical.

In Figure 2 (right) we broke down the contributions to the cross-entropies in terms of how many times each word appears in the test set. We see that most of the differences between the methods appear as differences among rare words, with the contribution of more common words being negligible. HPYLM performs worse than MKN on words that occurred only once (on average) and better on other words, while HPYCV is reversed and performs better than MKN on words that occurred only once or twice and worse on other words.

7 Discussion

We have described using a hierarchical Pitman-Yor process as a language model and shown that it gives performance superior to state-of-the-art methods. In addition, we have shown that the state-of-the-art method of interpolated Kneser-Ney can be interpreted as approximate inference in the hierarchical Pitman-Yor language model.

In the future we plan to study in more detail the differences between our model and the variants of Kneser-Ney, to consider other approximate inference schemes, and to test the model on larger data sets and on speech recognition. The hierarchical Pitman-Yor language model is a fully Bayesian model, thus we can also reap other benefits of the paradigm, including having a coherent probabilistic model, ease of improvements by building in prior knowledge, and ease in using as part of more complex models; we plan to look into these possible improvements and extensions.

The hierarchical Dirichlet language model of (MacKay and Peto, 1994) was an inspiration for our work. Though (MacKay and Peto, 1994) had the right intuition to look at smoothing techniques as the outcome of hierarchical Bayesian models, the use of the Dirichlet distribution as a prior was shown to lead to non-competitive cross-entropy results. Our model is a nontrivial but direct generalization of the hierarchical Dirichlet language model that gives state-of-the-art performance. We have shown that with a suitable choice of priors (namely the Pitman-Yor process), Bayesian methods can be competitive with the best smoothing techniques.

The hierarchical Pitman-Yor process is a natural generalization of the recently proposed hierarchical Dirichlet process (Teh et al., 2006). The hierarchical Dirichlet process was proposed to solve a different problem—that of clustering, and it is interesting to note that such a direct generalization leads us to a good language model. Both the hierarchical Dirichlet process and the hierarchical Pitman-Yor process are examples of Bayesian nonparametric processes. These have recently received much attention in the statistics and machine learning communities because they can relax previously strong assumptions on the parametric forms of Bayesian models yet retain computational efficiency, and because of the elegant way in which they handle the issues of model selection and structure learning in graphical models.

Acknowledgement

I wish to thank the Lee Kuan Yew Endowment Fund for funding, Joshua Goodman for answering many questions regarding interpolated Kneser-Ney and smoothing techniques, John Blitzer and Yoshua Bengio for help with datasets, Anoop Sarkar for interesting discussion, and Hal Daume III, Min Yen Kan and the anonymous reviewers for

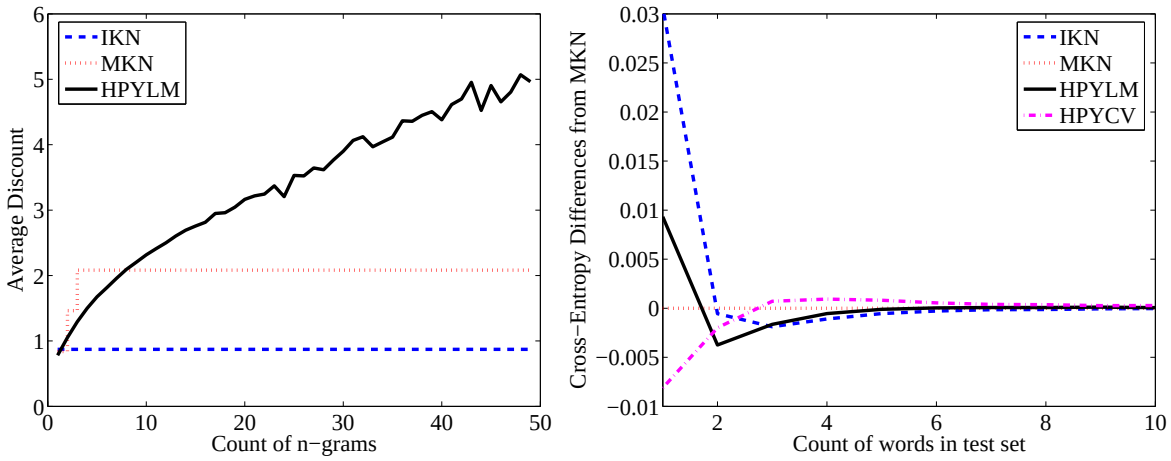

Figure 2: Left: Average discounts as a function of n-gram counts in IKN (bottom line), MKN (middle step function), and HPYLM (top curve). Right: Break down of cross-entropy on test set as a function of the number of occurrences of test words. Plotted is the sum over test words which occurred c times of cross-entropies of IKN, MKN, HPYLM and HPYCV, where c is as given on the x-axis and MKN is used as a baseline. Lower is better. Both panels are for the full training set and $n = 3$.

helpful comments.

References

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

S.F. Chen and J.T Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University.

A. Gelman, J. Carlin, H. Stern, and D. Rubin. 1995. *Bayesian data analysis*. Chapman & Hall, London.

Z. Ghahramani. 2005. Nonparametric Bayesian methods. Tutorial presentation at the UAI Conference.

S. Goldwater, T.L. Griffiths, and M. Johnson. 2006. Interpolating between types and tokens by estimating power-law generators. In *Advances in Neural Information Processing Systems*, volume 18.

J.T. Goodman. 2001. A bit of progress in language modeling. Technical Report MSR-TR-2001-72, Microsoft Research.

J.T. Goodman. 2004. Exponential priors for maximum entropy models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

H. Ishwaran and L.F. James. 2001. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173.

M.I. Jordan. 2005. Dirichlet processes, Chinese restaurant processes and all that. Tutorial presentation at the NIPS Conference.

R. Kneser and H. Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1.

D.J.C. MacKay and L.C.B. Peto. 1994. A hierarchical Dirichlet language model. *Natural Language Engineering*.

A. Nadas. 1984. Estimation of probabilities in the language model of the IBM speach recognition system. *IEEE Transaction on Acoustics, Speech and Signal Processing*, 32(4):859–861.

R.M. Neal. 1993. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto.

J. Pitman and M. Yor. 1997. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25:855–900.

J. Pitman. 2002. Combinatorial stochastic processes. Technical Report 621, Department of Statistics, University of California at Berkeley. Lecture notes for St. Flour Summer School.

R. Rosenfeld. 2000. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8).

Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. 2006. Hierarchical Dirichlet processes. *To appear in Journal of the American Statistical Association*.

Y. W. Teh. 2006. A Bayesian interpretation of interpolated Kneser-Ney. Technical Report TRA2/06, School of Computing, National University of Singapore.