Hierarchical Bayesian Languge Model Based on Pitman-Yor Processes

Yee Whye Teh

Probabilistic model of language

n-gram model

P(word i | $\text{word}_{\text{i-n+1}}^{\text{i-1}}$)

Typically, trigram model (n=3)

Utility

e.g., speech, handwriting recognition

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

Challenge

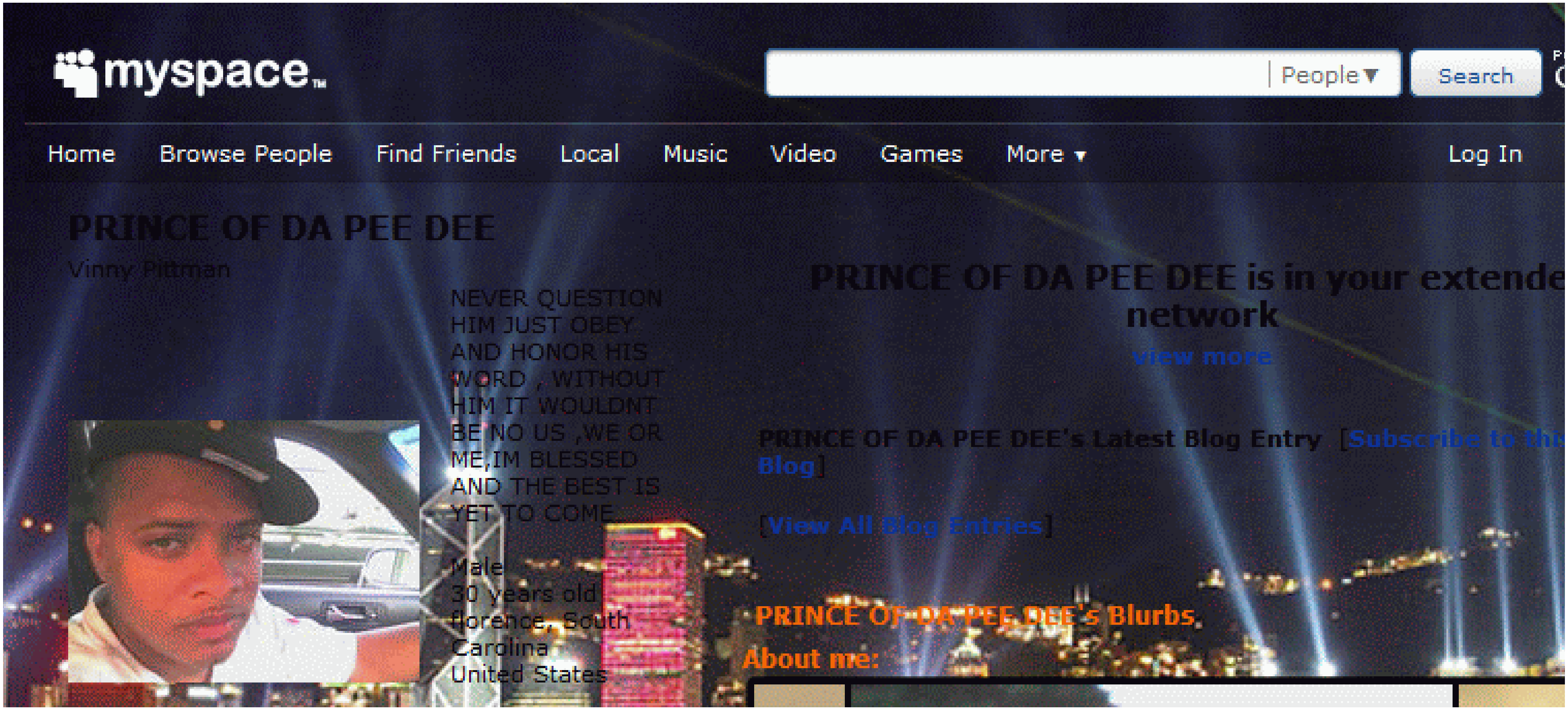
If n too small, doesn't capture regularities of language

If n too large, insufficient data

Past approaches have used heuristics for choosing appropriate n, or by averaging predictions across a range of values of n, or by smoothing hacks of various sorts.

Hierarchical Pitman-Yor probabilistic model addresses this challenge via principled approach with explicit assumptions.

Pitman not Pittman

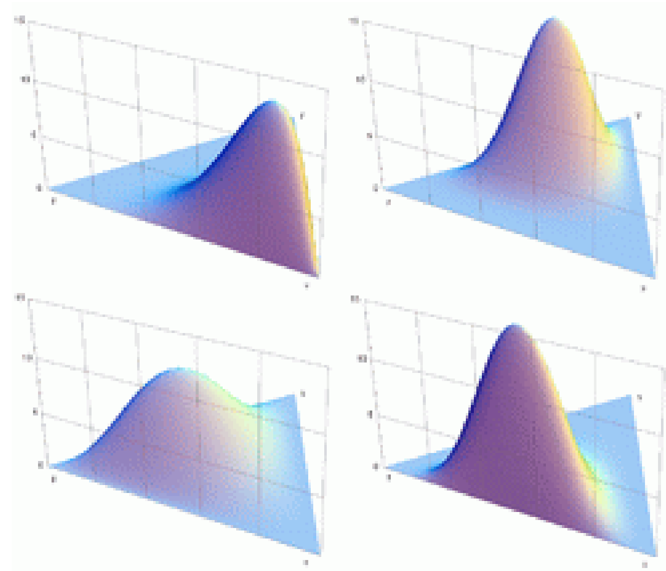


Dirichlet Distribution

Distribution over distributionsf over a finite set of alternatives

$$f(x_1,\ldots,x_{K-1};\alpha_1,\ldots,\alpha_K)=\frac{1}{\mathrm{B}(\alpha)}\prod_{i=1}^{K}x_i^{\alpha_i-1}$$

$$\mathrm{B}(\alpha)=\frac{\prod_{i=1}^{K}\Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)},\qquad \alpha=(\alpha_1,\ldots,\alpha_K).$$



E.g., uncertainty in distribution over a finite set of words

Dirichlet Process

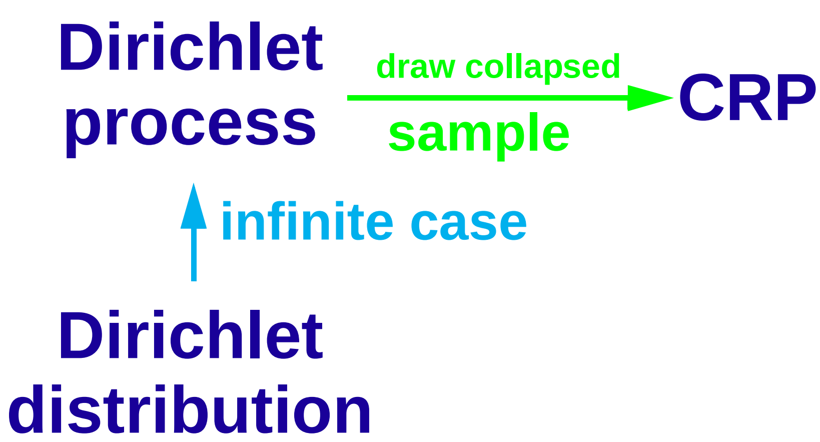
Distribution over a distributions over a countably infinite set of alternatives

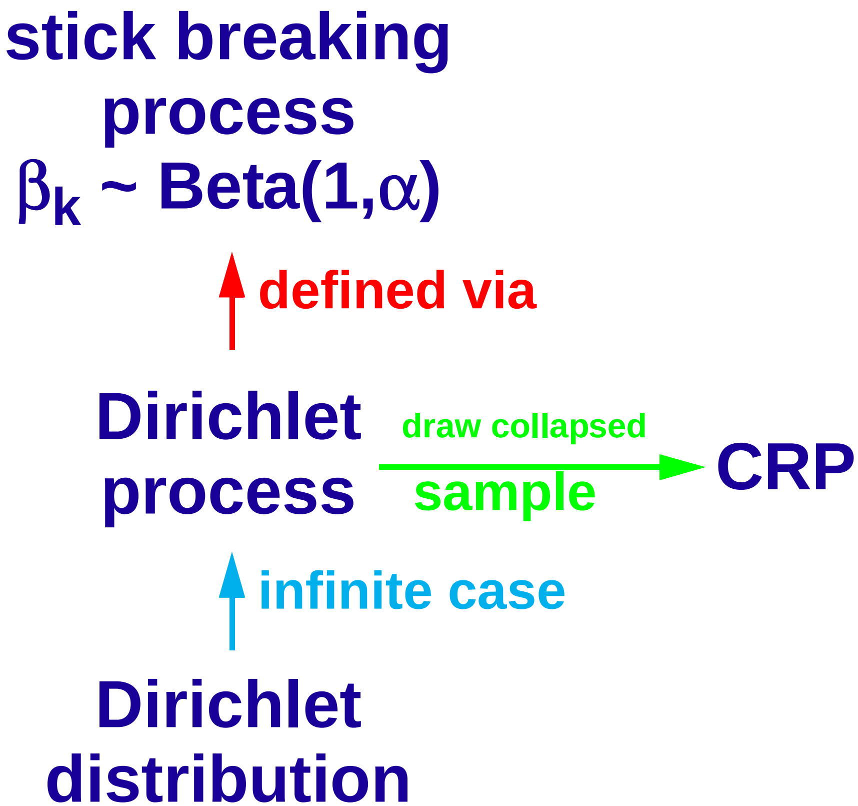
Dirichlet distribution

Dirichlet process

infinite case

Dirichlet distribution


Dirichlet process
draw collapsed
sample
CRP
infinite case
Dirichlet distribution


stick breaking process
$\beta_k \sim \text{Beta}(1,\alpha)$
defined via
Dirichlet process
draw collapsed sample
CRP
infinite case
Dirichlet distribution

stick breaking process

$\beta_k \sim \mathrm{Beta}(1,\alpha)$

defined via

Dirichlet process

draw collapsed sample

CRP

infinite case

Dirichlet distribution

simple analytic form

stick breaking process

$\beta_k \sim \mathrm{Beta}(1-d,\theta+dk)$

defined via

Pitman-Yor process

draw collapsed sample

fancy CRP

infinite case

Pitman-Yor distribution

no known analytic form

Pitman-Yor

Pitman-Yor distribution

Generalization of Dirichlet distribution

No known analytic form for density of PY for finite vocabulary.

Pitman-Yor Process

$G \sim PY(d, \alpha, G_0)$

d: discount parameter, $0 <= d < 1$

α: strength parameter, $\alpha > -d$

Large α = more concentration of probability according to G_0

Large d = more uniform a distribution

DP is special case of PY Process

$PY(0, \alpha, G_0) = DP(\alpha, G_0)$

see stick breaking construction

Understanding G_0

With Gaussian Mixture Model, domain of G_0 is real-valued vector

$G_0(\theta)$ is the mean probability of a Gaussian bump with mean μ and covariance Σ, where θ is vector $(\mu\ \Sigma)$.

With natural language, think of domain of G_0 as all letter strings (potential words)

$G_0(w)$ is the mean probability of “word w”

Drawing Sample Of Words Directly From PY Process Prior

Chinese restaurant process

person 1 comes into restaurant and sits at table 1

person c+1 comes into restaurant and sits at one of the populated tables, k, $k \in \{1, ..., t\}$, with probability ~ c_k (# people at table k)

... or sits at a new table (t+1) with probability ~ α

Generalized version of CRP

person 1 comes into restaurant and sits at table 1

person c+1 comes into restaurant and sits at one of the populated tables, k, $k \in \{1, ..., t\}$, with probability ~ $c_k - d$

... or sits at a new table (t+1) with probability ~ $\alpha + d\,t$

Pitman Yor

Why the restrictions $0 <= d < 1$ and $\alpha > -d$?

Generalized version of CRP

person 1 comes into restaurant and sits at table 1

person c+1 comes into restaurant and sits at one of the populated tables, k, $k \in \{1, ..., t\}$, with probability ~ $c_k - d$

... or sits at a new table (t+1) with probability ~ $\alpha + d\, t$

Pitman Yor

Like the CRP, notion that a specific "meal" is served at each table

- $\theta_k \sim G_0$ is meal for table k
- $\phi_c = \theta_k$ is meal instance served to individual c sitting at table k
- With DP mixture model, "meal" served was parameters of a gaussian bump
- With DP/PY language model, "meal" served is a word

Types vs. tokens

- e.g., The_1 mean dog_1 ate the_2 small dog_2.

Algorithm for drawing word c+1 according to Pitman-Yor

```
if c = 0            /* special case for first word */
   t ← 1            /* t: number of tables */
   θ_t ~ G_0        /* meal associated with table */
   c_t ← 1          /* c_t: count of customers at table t */
   φ_c ← θ_t        /* φ_c: meal associated with customer c */

else
   choose k ∈ {1, ..., t} with probability ~ c_k – d,
          k = t+1 with probability ~ α + d t

   if k = t + 1
      t ← t + 1
      θ_t ~ G_0
      c_t ← 1
   else
      c_k ← c_k + 1
   endif

   φ_c ← θ_k

endif
```

Pitman Yor Produces Power Law Distribution

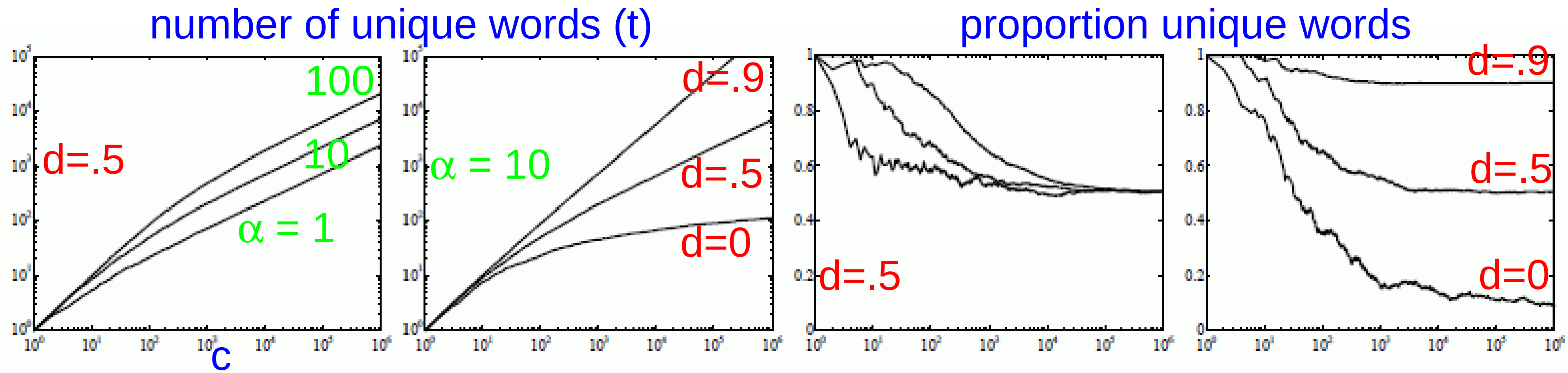


Figure 1: First panel: number of unique words as a function of the number of words drawn on a log-log scale, with $d = .5$ and $\alpha = 1$ (bottom), 10 (middle) and 100 (top). Second panel: same, with $\alpha = 10$ and $d = 0$ (bottom), .5 (middle) and .9 (top). Third panel: proportion of words appearing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

d is asymptotic proportion of words appearing only once

note d=0 ⇔ asymptotically no single-token words

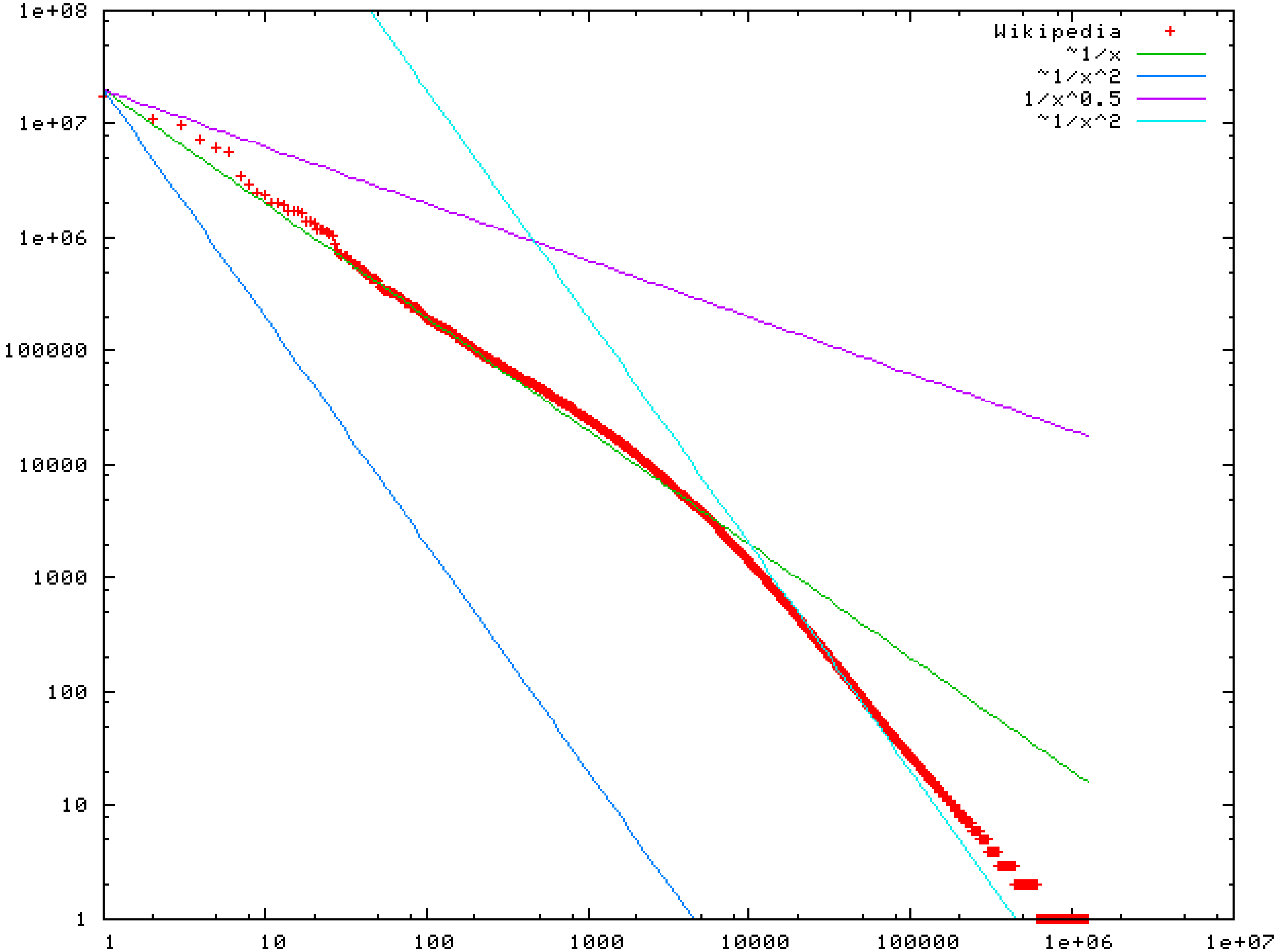
Number of unique words as a function of c

- $d > 0$: $O(\alpha c^d)$
- $d = 0$: $O(\alpha \log c)$

Is This The Same As Zipf's Law?

Rank words by frequency

Plot log frequency vs. log word index



Hierarchical Pitman-Yor (or Dirichlet) Process

Suppose you want to model where people hang out in a town (hot spots)

- Not known in advance how many locations need to be modeled

Some spots are generally popular, others not so much.

But individuals also have preferences that deviate from the population preference.

- E.g., bars are popular, but not for individuals who don't drink.

Need to model distribution over hot spots at level of both population and individual.

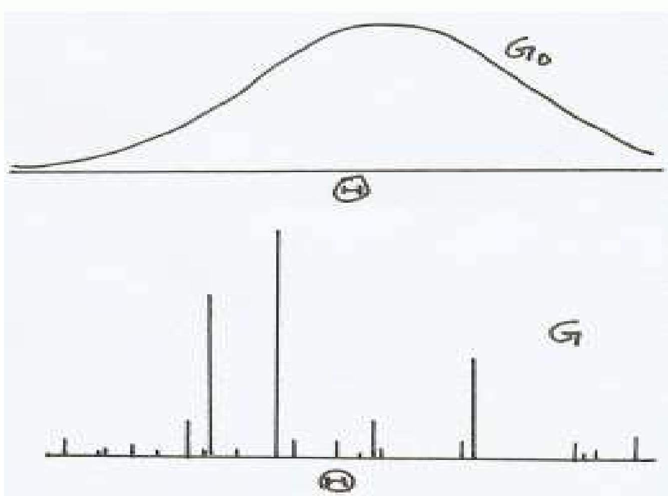
Hierarchical Pitman-Yor (or Dirichlet) Process

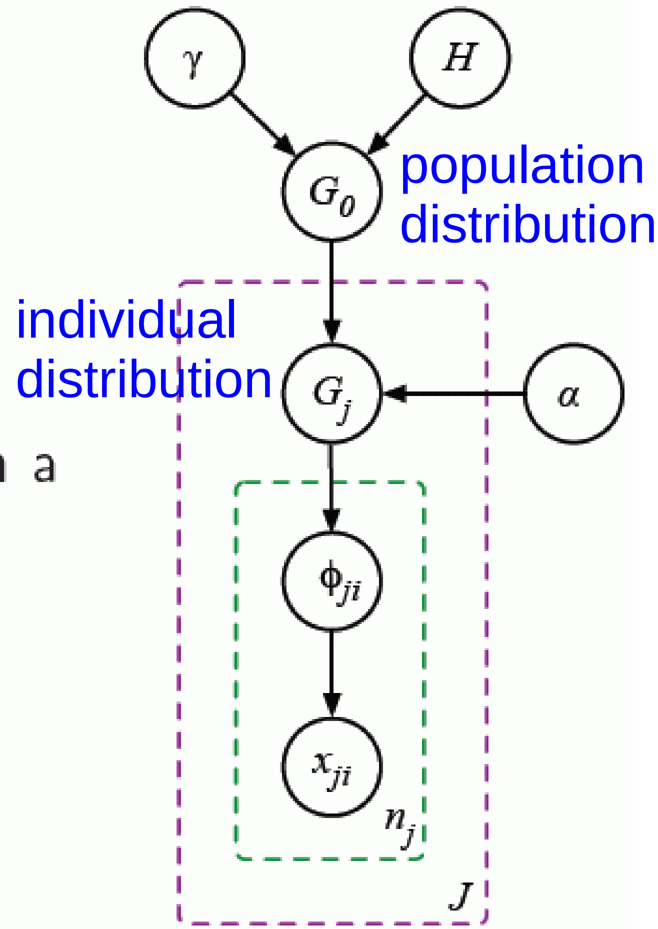
In an HDP there is a common DP:

$$G_0|H,\gamma \sim \mathrm{DP}(\cdot|H,\gamma)$$

Which forms the base measure for a draw from a DP within each group

$$G_j|G_0,\alpha \sim \mathrm{DP}(\cdot|G_0,\alpha)$$





From Two-Level Hot-Spot Model To Multilevel Word Model: Capturing Levels Of Generality

P(hot spot)

P(hot spot | individual)

$P(word_i)$

$P(word_i|word_{i-1})$

$P(word_i|word_{i-1},word_{i-2})$

Intuition

Use the more general distribution as the basis for the more context-specific distribution

$P(word_i \mid word_{i-1})$ is specialized version of $P(word_i)$

$P(word_i \mid word_{i-1},word_{i-2})$ is specialized version of $P(word_i \mid word_{i-1})$

Formally

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

suffix of u consisting of all but the earliest word

context (preceding n–1 words)

Defines G_u recursively, anchored with

$$G_{\emptyset} \sim \mathrm{PY}(d_0, \alpha_0, G_0)$$

Sampling With Hierarchical CRP

Imagine a different restaurant for every possible context u

e.g., “supreme”, “states supreme”, “united states supreme”, “united states of”

... each with its own set of tables and own population of diners

The meals served at the popular tables at restaurant u will tend to be same as meals at popular tables at restaurant π(u)

Consider restaurants (contexts) **u** and π(**u**)

e.g., **u** = “the united” and π(**u**) = “united”

and some past (in generative model) assignments of tables based on

‘united states’
‘united we stand’
‘united under god’
‘the united states’

Notation

$c_{\mathbf{u}.k}$

- number of diners in restaurant **u** assigned to table *k*
- number of tokens of the word indexed by *k* appearing in context **u**
- But we also need to represent the meal served at a table ...

$c_{\mathbf{u}w.}$

- number of diners in restaurant **u** assigned to a table with meal *w*
- number of tokens of word w appearing in context **u**
- NOTE: unlike DPMM, this count is an *observation*

$c_{\mathbf{u}wk}$

- number of diners in restaurant **u** assigned to table *k* which has meal *w*
- in context **u**, number of tokens of word *w*, when word *w* is indexed by *k*

$\mathbf{t}_{\mathbf{u}\cdot\cdot}$

- formerly just t: number of occupied tables in restaurant **u**
- number of word types appearing in context **u**

$\mathbf{t}_{\mathbf{u}w\cdot}$

- 1 if restaurant **u** serves meal *w*, 0 otherwise
- 1 if word *w* appears in context **u**

$\mathbf{t}_{\mathbf{u}wk}$

- 1 if restaurant **u** has meal *w* assigned to table *k*, 0 otherwise

Redundant indexing of c and t by both *k* and *w* necessary to allow for reference either by

table (for generative model) or

meal (for probability computation)

Algorithm for drawing word w in context u

Function w = DrawWord($\mathbf{u}$)

if $\mathbf{u} = 0$
  $w \sim G_0$

else
  choose k $\in$ {1, ..., $t_{\mathbf{u}..}$} with probability ~ $c_{\mathbf{u}.k} - d_{|\mathbf{u}|}$,
    k = $t_{\mathbf{u}..}$+1 with probability ~ $\alpha_{|\mathbf{u}|} + d_{|\mathbf{u}|}\, t_{\mathbf{u}..}$

  if k = $t_{\mathbf{u}..} + 1$
    $\theta_{\mathbf{u}k} \leftarrow$ DrawWord(π($\mathbf{u}$))   <- new table drawn or restaurant is empty
    $w \leftarrow \theta_{\mathbf{u}k}$
    $c_{\mathbf{u}wk} \leftarrow 1$
    $t_{\mathbf{u}wk} \leftarrow 1$ COOL RECURSION!
  else
    $w \leftarrow \theta_{\mathbf{u}k}$
    $c_{\mathbf{u}wk} \leftarrow c_{\mathbf{u}wk} + 1$
endif

Inference

Given training data, $\mathcal{D}$, compute predictive posterior over words w for a given context, u:

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

$\Theta = \{\alpha_m, d_m: 0 <= m <= n-1\}$
$\mathcal{S}$ = seating arrangement

Because of structured relationship among the contexts, this probability depends on more than the empirical count.

e.g., p("of" | "united states") will be influenced by p("of" | "states"), p("of"), p("of" | "altered states"), etc.

Inference

Given training data, $\mathcal{D}$, compute predictive posterior over words w for a given context, u:

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

$\Theta = \{\alpha_m, d_m: 0 <= m <= n-1\}$

$\mathcal{S}$ = seating arrangement

Compute $P(w|u,\mathcal{S},\Theta)$ with...

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

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

Inference

Given training data, $\mathcal{D}$, compute predictive posterior over words w for a given context, u:

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

$\Theta = \{\alpha_m, d_m: 0 <= m <= n-1\}$

$\mathcal{S}$ = seating arrangement

Compute P(w|u,$\mathcal{S}$,Θ) with...

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

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

Estimate P($\mathcal{S}$,Θ|$\mathcal{D}$) with MCMC

Gibbs Sampling of Seat Assignments

For a given individual (l) in a given restaurant (u), there are only a few possible choices for the seat assignment (k_{ul})

- If the individual's meal is already assigned to a table, they can sit at one of those tables.
- A new table can be created that serves the individual's meal.

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

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

Sampling of Parameters Θ

- "Parameters are sampled using an auxiliary variable sampler as detailed in Teh (2006)"

Implementation

$G_0(w) = 1/V$ for all w

$d_i \sim$ Uniform(0,1)

$\alpha_i \sim$ Gamma(1,1)

Results

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.

IKN, MKN = traditional methods in language; HPYLM = hierarchical Pitman-Yor with Gibbs sampling; HPYCV= hierarchical Pitman-Yor with some cross validation hack to determine parameters; HDLM = hierarchical Dirichlet language model