Exploration of Stick-breaking Process to Develop Efficient Algorithms for Modern Data Science

Mrinal Das

Computer Science and Engineering
Indian Institute of Technology Palakkad

Some prevalent challenges in Modern Data Science

- Inherent and natural imbalance in dataset
- Demand of efficient algorithms
  space, speed, energy
- Convergence analysis of learning algorithms
- Preserving Privacy of users
- Fairness and accountability

Some prevalent challenges in Modern Data Science

- Imbalance in dataset
- Demand of efficient algorithms
  space, speed, energy
- Convergence analysis on learning algorithms
- Preserving Privacy of users
- Fairness and accountability

An Avenue with Theoretical Basis

MENU

nature
International journal of science

Review Article | Published: 27 May 2015

Probabilistic machine learning and artificial intelligence

Zoubin Ghahramani

Nature **521**, 452–459 (28 May 2015) | Download Citation

Salient features

- Representing uncertainty
- Flexibility through nonparametrics

In this talk, I will present one such tool:

Stick-Breaking Process

Stick-breaking process

Stick-breaking process is a meachanism to define distributions

- Uncertainty is inherent in data and machine learning
- Probability and distributions are critical tools
- Often we need to define a suitable distribution

Stick-breaking process (SBP)

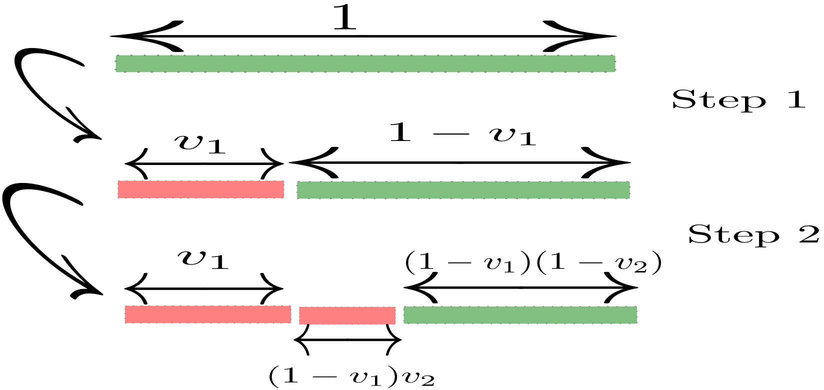
Any G is a SBP prior (Ishwaran and James, 2001) if

$$\mathrm{G} = \sum_{j=1}^{\infty} \theta_j \delta_{\beta_j}$$
$$\theta_1 = v_1, \; \theta_j = v_j \prod_{l=1}^{j-1} (1 - v_l)$$
$$a_j, b_j > 0, \; v_j \sim Beta(a_j, b_j), \;\; \beta_j \sim \mathrm{H}$$

- δ_{β_j}: atomic distribution
- $\{\beta_j\}$: commonly referred as atoms
- H: a distribution generally referred as base measure
- $\{a_j, b_j\}$: hyper-parameters

Illustration of SBP

$\theta_j = v_j \prod_{l=1}^{j-1}(1 - v_l)$



Assign each broken part to $\{\theta_j\}$

Illustration of SBP

P is an SBP if,

Distribution

$$P = \sum_{j=1}^{\infty} \theta_j \delta_{\beta_j}$$

$P(\beta_j) = \theta_j$

components

H

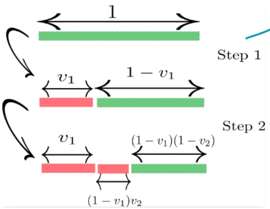
$$\begin{array}{l} \beta_1,\ \beta_2,\ \beta_3,\ \ldots,\beta_k,\ \beta_{k+1},\ \ldots \\ \theta_1,\ \theta_2,\ \theta_3,\ \ldots,\theta_k,\ \theta_{k+1},\ \ldots \end{array}$$

weights

$$\theta_1 = v_1, \theta_j = v_j \prod_{l=1}^{j-1}(1 - v_l)$$

$$v_j \sim Beta(a_j, b_j), \beta_j \sim H$$

$$\forall j, \theta_j \geq 0, \ \ \sum_{j=1}^{\infty} \theta_j = 1$$



Two important special cases of SBP

- Dirichlet process (DP) Ferguson, 1973

  – when $a_j = 1$, and $b_j = \gamma$ for all j, $\mathrm{G} \sim DP(\gamma, \mathrm{H})$

- Pitman-Yor process (PYP) Pitman and Yor, 1997

  – when $a_j = 1 - \lambda$ and $b_j = \gamma + j\lambda$ for all j, $\mathrm{G} \sim PYP(\lambda, \gamma, \mathrm{H})$

See Ishwaran and James, 2001 for more details.

Application of SBP in modeling

$$
\begin{aligned}
P &\sim SBP(a, b, H) \\
\forall i, \quad & \phi_i \sim P \\
& x_i \sim f(\phi_i)
\end{aligned}
$$

- This forms a generative model.
- This is also a hierarchical Bayesian model.

- $\{x_i\}_{i=1}^{n}$ forms the dataset.
- P defines the distribution at the global level
- ϕ_i is the local variable for data point x_i

Application of SBP in modeling

$$P \sim SBP(a, b, H)$$

$$[P = \sum_{j=1}^{\infty} \theta_j \delta_{\beta_j}, \quad \beta_j \sim H]$$

$$\forall i, \qquad \phi_i \sim P$$

$$[\phi_i \in \{\beta_j\}]$$

$$x_i \sim f(\phi_i)$$

$$[x_i \sim f(\beta_j), \text{ if } \phi_i = \beta_j]$$

Application of SBP in modeling

$$P \sim SBP(a, b, H)$$

$$[P = \sum_{j=1}^{\infty} \theta_j \delta_{\mu_j}, \quad \mu_j \sim N(0, 1)]$$

$$\forall i, \qquad \phi_i \sim P$$

$$[\phi_i \in \{\mu_j\}]$$

$$x_i \sim N(\phi_i, \sigma^2)$$

$$[x_i \sim N(\mu_j, \sigma^2), \text{ if } \phi_i = \mu_j]$$

We get Gaussian Mixture model.

SBP as Bayesian Nonparametric prior

- SBP provides a prior for the model
- Given dataset, we apply Bayes rule and get posterior estimate of the model
  – inference
- SBP allows to learn number of parameters

$$p(\mathcal{M}|\mathcal{D}) \propto p(\mathcal{D}|\mathcal{M})p(\mathcal{M})$$

$$\frac{p(\mathcal{M}_1|\mathcal{D})}{p(\mathcal{M}_2|\mathcal{D})} = \frac{p(\mathcal{D}|\mathcal{M}_1)p(\mathcal{M}_1)}{p(\mathcal{D}|\mathcal{M}_2)p(\mathcal{M}_2)}$$

SBP as a tool for Bayesian model selection

$\mathcal{M}_1 = \delta_{\beta_1}$: model with one parameter set

$\mathcal{M}_2 = \theta_2\delta_{\beta_1} + \theta_2\delta_{\beta_2}$: model with two parameters set

$\mathcal{M}_3 = \theta_2\delta_{\beta_1} + \theta_2\delta_{\beta_2} + \theta_3\delta_{\beta_3}$: model with three parameters set

$\mathcal{M}_J = \sum_{j=1}^{J} \theta_j\delta_{\beta_j}$: model with J parameters set

$$p(\mathcal{M}_k|\mathcal{D}) \propto p(\mathcal{D}|\mathcal{M}_k)p(\mathcal{M}_k)$$

Converge to a model with appropriate complexity.

A brief history of SBP

- Sethuraman, 1994 first proposed stick-breaking construction for DP
- Pitman and Yor, 1997 generalized DP from many aspects – proposed Pitman-Yor process (PYP)
- (Ishwaran and James, 2001) proposed SBP to unify many BNP priors such as DP, PYP

- SBP inference turned out to be hard in general
- SBP forms of DP and PYP are less used
- SBP remained highly unexplored

Interesting property of SBP

- SBP provides a constructive method to define a.s. discrete probability measures

  – potential solution to many unsolved problems where other priors do not apply

This talk explores and extends constructive framework of SBP to address some prevalent tasks of recent interest.

Challenges with inference

- Inference becomes non-standard due to SBP

  – Predictive probability functions (PPFs) do not exist for SBP (Pitman, 1996)

  PPFs are useful tool in MCMC inference

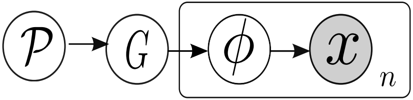
MCMC Inference for SBP

- SBP found to be equivalent to generalized Dirichlet distribution (GDD) (Connor and Mosimann, 1969)
- GDD is a generalized version of Dirichlet distribution
- GDD is conjugate to multinomial distribution
- We utilize this relationship to derive an efficient collapsed Gibbs sampling inference

Memory Efficient Learning using Stick-breaking process

BNP models and large scale learning

A wide class of Bayesian NonParametric (BNP) models



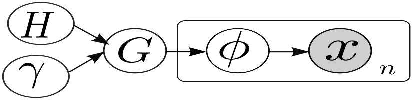
- $\mathcal{P}$ is a BNP prior
  – e.g. Dirichlet process (DP), Pitman-Yor process (PYP)

Question ?

- How can we use such BNP models
  – to analyse corpora of million documents
  – using sequential processing

Dirichlet process mixture model (DPMM)

A corner stone of BNP methods



- $\{x_i\}_1^n$ are observations
- $G \sim DP(\gamma, H)$

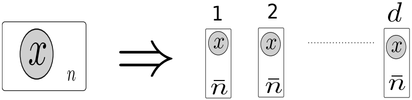
- MCMC is generally accurate and widely used for inference

★ For large n, existing MCMC methods do not apply (Wang and Blei 2012, Williamson et. al. 2013)

Sequential inference for large *n*

One can consider sequential Monte Carlo (Doucet et al, 2001)

- Split observations into mini-batches
- Process mini-batches sequentially



Standard technique – Particle filtering (PF) (Fearnhead, 2004)

- Needs to maintain multiple configurations of $O(n)$
- Can be implemented only in distributed setup for large n (Wang and Blei 2012, Williamson et. al. 2013)

Existing MCMC requires huge space

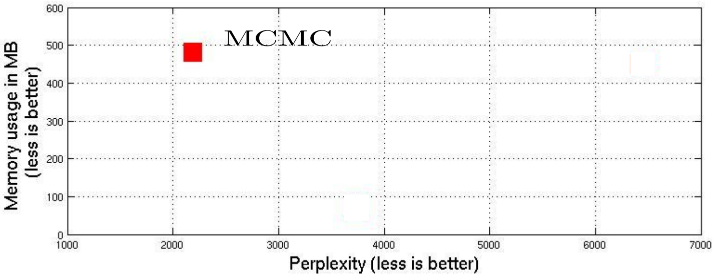
State of the art in sequential inference for large n

- Truncation-free stochastic variational inference (TSVI) (Wang and Blei, 2012)

★ No MCMC method exists to compete with TSVI in scale (Wang and Blei, 2012)

Memory vs perplexity for MCMC

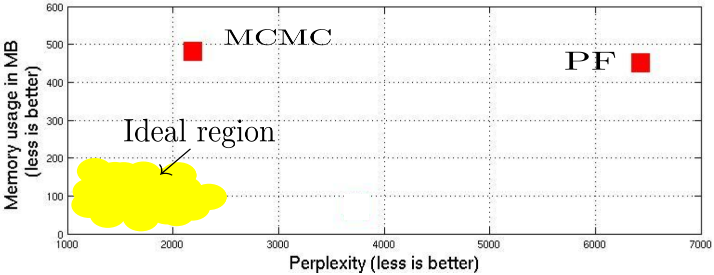
Existing MCMC method is accurate (Neal 2000)
but consumes high memory $O(n)$



– dataset NIPS (1500 documents)

Resorting to sequential Monte Carlo

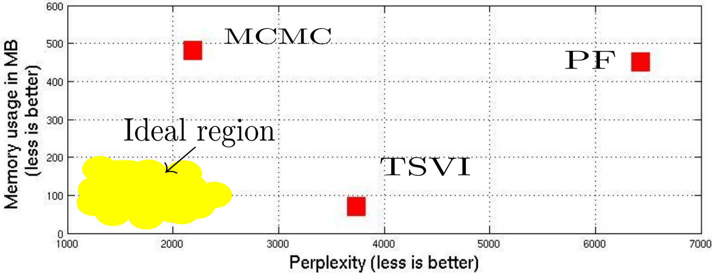
Particle filtering is state of the art (Fearnhead, 2004)
Not sufficiently reduces memory requirement



Ideal region – low perplexity, low memory.

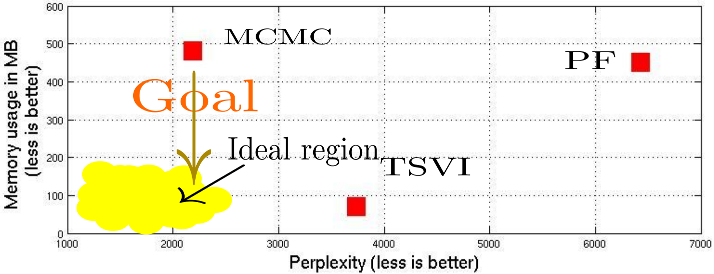
Stochastic variational inference is effective

TSVI consumes much less memory (Wang and Blei, 2012) but not as accurate as MCMC



Ideal region – low perplexity, low memory.

Goal



- Can we reduce memory requirement of MCMC approach – to compete with TSVI in space and MCMC in accuracy ?

Route

A Bayesian approach to process mini-batches sequentially

such that

- the technique will generally apply to BNP models
- the resultant inference problem can be solved using MCMC

★ We propose to use a BNP prior over mini-batches

Challenges

Need a suitable prior over mini-batches

- Mini-batches arrive in sequence – non-exchangeable
- Memory requirement must not grow linearly

Approach

OSBP – a novel BNP prior

- Proposed ordered stick-breaking process (OSBP) extending stick-breaking process (SBP)
- the first attempt to put a prior over mini-batches
- generally applies to BNP models

SUMO – SeqUential MCMC inference through OSBP

- reduces memory requirement of MCMC significantly
- competitive with TSVI in space and outperforms in accuracy

Ordered Stick-Breaking Process

Ordered stick-breaking process

$$
\begin{aligned}
\mathrm{G}_t \,\Big|\, \mathrm{G}_{1:t-1}, (\rho_j), \Gamma \quad &\sim \quad \sum_{j=1}^{k_{t-1}} \rho_j \delta_{\mathrm{Q}_j} \;+\; \alpha_{k_{t-1}} \Gamma \\
\rho_1 = v_1,\ \ \forall j > 1,\ \rho_j &= v_j \textstyle\prod_{l=1}^{j-1} (1 - v_l) \\
v_j | \mu_j, \nu_j &\sim Beta\,(\mu_j \nu_j, (1 - \mu_j)\nu_j) \\
\alpha_{k_{t-1}} &= 1 - \textstyle\sum_{j=1}^{k_{t-1}} \rho_j, \quad \mathrm{G}_1 \sim \Gamma
\end{aligned} \tag{1}
$$

Recall SBP: $\mathrm{P} = \sum_{j=1}^{\infty} \rho_j \delta_{\mathrm{Q}_j}$

- $(\mathrm{Q}_1, \ldots, \mathrm{Q}_{k_{t-1}})$: k_{t-1} atoms after time $t - 1$
- $\mathrm{G}_t \in \{\mathrm{Q}_1, \ldots, \mathrm{Q}_{k_{t-1}}, \mathrm{Q}_{k_{t-1}+1}\}$, $\mathrm{Q}_{k_{t-1}+1}$: next new atom
- Atoms $(\mathrm{Q}_1, \mathrm{Q}_2, \ldots)$ appear in order
- We denote, $\mathrm{G}_1, \mathrm{G}_2, \ldots \sim \mathrm{OSBP}(\boldsymbol{\mu}, \boldsymbol{\nu}, \Gamma)$

Atoms appearing in order

- OSBP is an extension of SBP for atoms appearing in order

Chinese restaurant process

Customers arriving in a restaurant and sitting on tables

- Tables are numbered
- Customers are also numbered

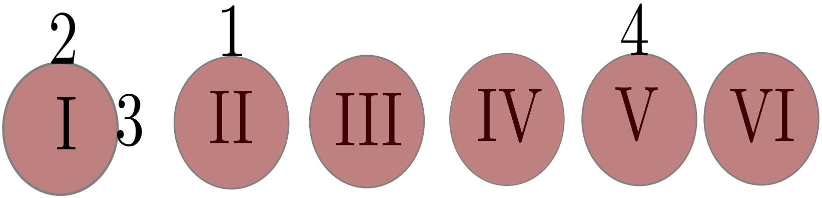
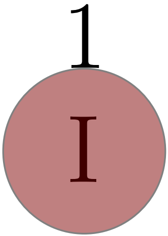


Table numbers do not matter – exchangeability

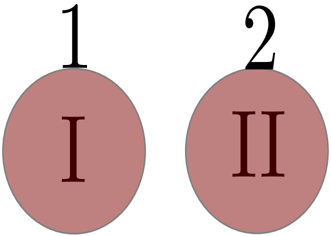
Appearance in order

- Customer 1 has to sit on table I



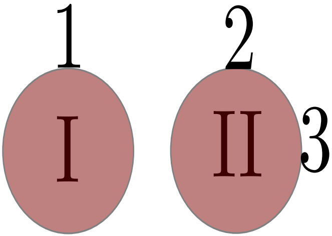
Appearance in order

- Customer 2 has to sit on table I or II



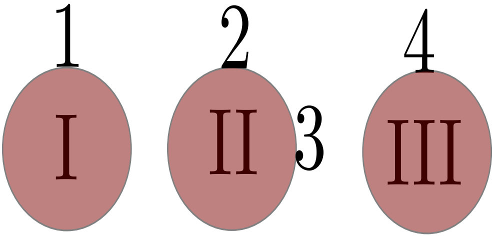
Appearance in order

- Customer 3 has to sit on table I, II or III

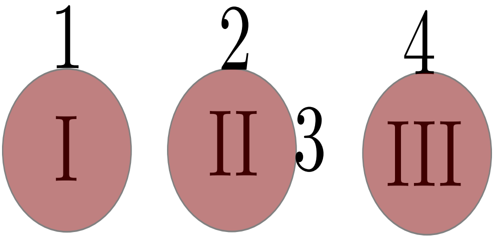


Appearance in order

- Customer 4 has to sit on table I, II or III



Appearance in order



- Tables must be occupied maintaining order

- Customers are samples – arriving in sequence i.e. ordered
- Tables are atoms – must be ordered by appearance

- Atoms appearing in order
  – a novel concept in stick-breaking

Asymptotic nature of OSBP

Theorem 1

If $\mathsf{P}_1 = \Gamma$, $\mathsf{P}_t = \sum_{j=1}^{k_{t-1}} \rho_j \delta_{\mathsf{Q}_j} + \alpha_{k_{t-1}} \Gamma$ for $t > 1$ and $\mathsf{P}^* = \sum_{j=1}^{\infty} \rho_j \delta_{\mathsf{Q}_j}$ such that $\sum_{j=1}^{\infty} \rho_j = 1$, where (ρ_j), (Q_j), α_{k_t} and Γ as defined in Eq. (1) with parameter $\boldsymbol{\mu}, \boldsymbol{\nu}$, then $\lim_{t\to\infty} \mathsf{P}_t = \mathsf{P}^*$ a.s.

- $\sum_{j=1}^{k_{t-1}} \rho_j \delta_{\mathsf{Q}_j} + \alpha_{k_{t-1}} \Gamma \to \sum_{j=1}^{\infty} \rho_j \delta_{\mathsf{Q}_j}$ as $t \to \infty$
- Recall SBP: $\mathsf{P} = \sum_{j=1}^{\infty} \rho_j \delta_{\mathsf{Q}_j}$

★ OSBP asymptotically behaves like SBP

Probability of adding atoms decreases

Theorem 2

For α_{k_t} as defined in Eq. (1) with parameters $\boldsymbol{\mu}, \boldsymbol{\nu}$, and any $\epsilon \in (0, 1)$, if $\mu_j > 1/2$ for all j, then $\alpha_k \leq \epsilon$ whenever $k \geq \frac{2}{\log 2} \log \frac{1}{\epsilon}$ with probability more than $1 - \epsilon$.

- Probability of adding new atom exponentially decreases

★ Impacts directly on memory footprint of SUMO

Predictive probability functions (PPFs)

PPFs are defined as below (Pitman, 1996)

$$\pi_j = p(z_t = j | z_{1:t-1}, \Theta), \ 1 \leq j \leq k,$$
$$\sigma_k = p(z_t = k + 1 | z_{1:t-1}, \Theta)$$

- PPFs are useful for truncation-free inference

  $z_t | z_{1:t-1} \sim \sum_{j=1}^{k} \pi_j \delta_j + \sigma_k \delta_{k+1}$

- k can grow un-boundedly – true BNP spirit

  – example: Chinese restaurant process for DP

Truncation-free inference

Theorem 3

- Predictive probability functions (PPFs) exist for OSBP.

★ PPFs allow truncation-free MCMC inference for OSBP based models

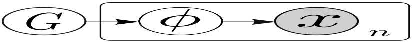
- SBP in general does not allow truncation-free inference

  – except DP and Pitman-Yor process Pitman, 1996

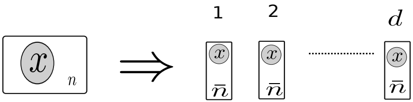
Seq**U**ential **M**CMC inference through **O**SBP (SUMO)

Sequential inference through mini-batches

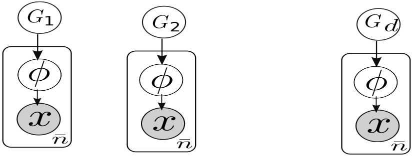
DPMM: $G \sim DP(\gamma, H)$



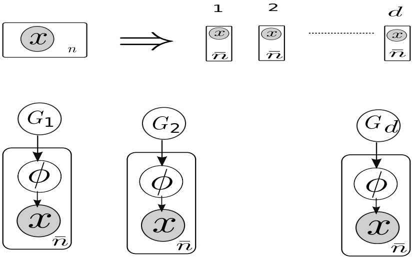
Step 1: split observations into mini-batches



Step 2: process mini-batches using DPMM, each $G_j \sim DP(\gamma, H)$



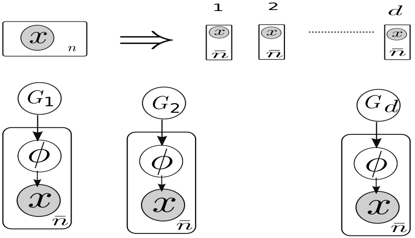
Limitations of the existing methods



- Independent $(G_1, G_2, \ldots)$ cant approximate full posterior
- Need to share information across $(G_1, G_2, \ldots)$
- Stochastic variational inference does that effectively

★ No equivalent mechanism exists in MCMC family

Our Bayesian approach



- Need to share information across $(G_1, G_2, \ldots)$

Consider Bayesian approach to put a prior over mini-batches

OSBP as a prior over mini-batches

★ Consider Bayesian approach to put a prior over mini-batches
– we can use discrete probability measures e.g. SBP

Notice that

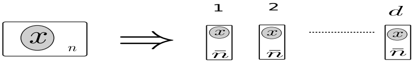
- Mini-batches arrive in a pre-defined order
- $\Rightarrow (G_1, G_2, \ldots)$ a sequence in order
- Creates appearance in order effect

★ OSBP comes into play here

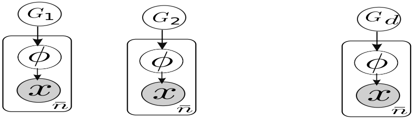
SeqUential MCMC inference through OSBP (SUMO)

Recipe:

Step 1: split observations into mini-batches



Step 2: process mini-batches using DPMM, each $\mathsf{G}_j \sim DP(\gamma, \mathsf{H})$

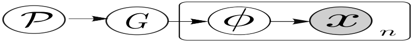


$\mathsf{G}_1, \mathsf{G}_2, \ldots \sim OSBP(\mu, \nu, DP(\gamma, \mathsf{H}))$

★ The resultant MCMC inference $\Rightarrow$ SUMO

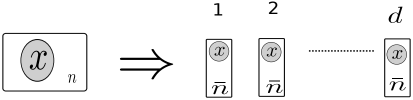
SUMO applies to a general class of BNP models

$\mathcal{P}$ a BNP prior e.g. DP, PYP, HDP, SBP

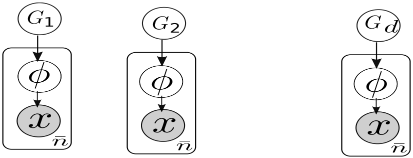


Recipe:

Step 1: split observations into mini-batches



Step 2: process mini-batches with $G_1, G_2, \ldots \sim OSBP(\mu, \nu, \mathcal{P})$



OSBP on DPMM converges to DPMM

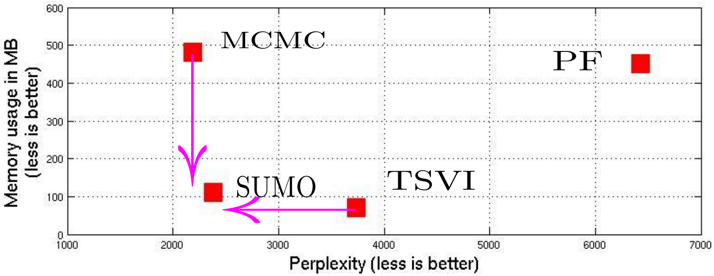
Theorem 4

For any $t \in \mathbf{N}$, each x_{ti} sampled using OSBP based model has marginal distribution same as x_i sampled with DPMM model with $\mathrm{G} \sim DP(c_t, \mathrm{H})$, where $c_t = \sum_{j=1}^{k_{t-1}} \gamma_j + (1-\mu)^{k_{t-1}}\gamma$. Furthermore, with probability greater than $1-\epsilon$ for t when $k_t \geq k \geq \frac{2}{\log 2} \log \frac{1}{\epsilon}$ and any $\epsilon > 0$, each x_{ti} in OSBP based model has marginal distribution same as x_i in DPMM with $\mathrm{G} \sim DP(\sum_{j=1}^{k} \gamma_j, \mathrm{H})$. Also, for $t \to \infty$, each x_{ti} in OSBP based model has marginal distribution same as x_i in DPMM with $\mathrm{G} \sim DP(\gamma, \mathrm{H})$.

- OSBP makes loss-less approximation of DPMM asymptotically

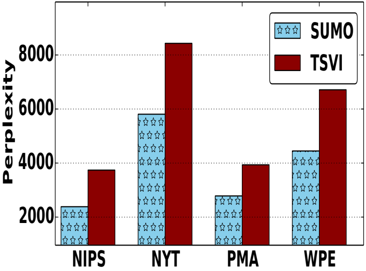
SUMO can achieve our goal

- Reduces memory requirement of existing MCMC
- Maintains accuracy of existing MCMC



– dataset NIPS

Comparison with state of the art



Dataset	Documents	Tokens
NIPS	1500	1.9 M
NYT	300 K	100 M
PMA	8.2 M	730 M
WPE	1 M	296 M

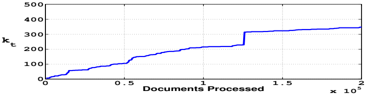
- ★ SUMO outperforms TSVI on all datasets (average 33%)
- ★ Existing MCMC, and PF both do not fit in memory

Experiments are run on a 3 GB RAM system

Memory requirement for SUMO

Grows with k_t

- k_t does not increase much – Theorem 2
- k_t stops increasing soon – Theorem 1



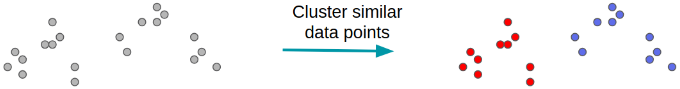
– dataset NYT

Largest run-time memory footprint among all datasets

- SUMO: 1.8 GB
- TSVI: 1.1 GB

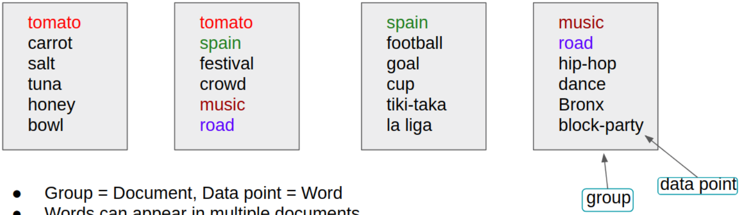
Modeling Rare Statistics using Stick-breaking process

Clustering



Hard to Guess the Number of Clusters

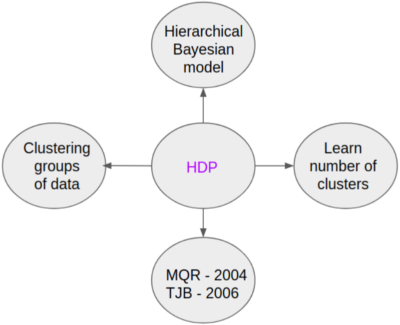
Clustering of Group of Observations



- Group = Document, Data point = Word
- Words can appear in multiple documents

1. Cluster words across documents.
2. Number of clusters is not known.
3. Hierarchical Dirichlet Process is useful.

Hierarchical Dirichlet Process



A Brief History of HDP:

- Dirichlet Process (DP) is the cornerstone
- Ferguson developed DP in 1973
- A statistical tool extending Dirichlet distribution for unlimited complexity
- HDP extends DP for grouped data
- HDP has 2 major versions
    - Muller et al 2004, Journal of Royal Statistical Society (nearly 150 citations)
    - Teh et al 2006, Journal of American Statistical Association (nearly 3500 citations)

Popularity of Clusters are Forced to be Correlated

Observations: $\{\{x_{si}\}_{i=1}^{n_s}\}_{s=1}^{S}$

Hierarchical Bayesian model using HDP

$$\forall s,\ \forall i \quad x_{si} \sim f(\phi_{si}), \quad \phi_{si} \sim \mathrm{P}_s,$$
$$\mathrm{P}_s = \sum_{j=1}^{\infty} \theta_{sj}\delta_{\beta_{0j}}, \quad \forall j \quad \beta_{0j} \sim \mathrm{H}$$

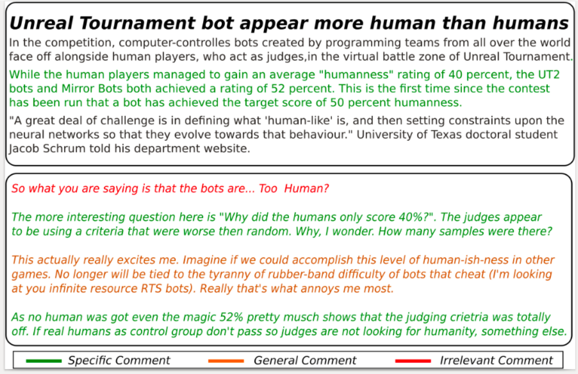
2 standard formulations for HDP

$$\text{MQR-HDP} \quad \forall s,\ \mathrm{P}_s = \epsilon \mathrm{G}_0 + (1-\epsilon)\mathrm{G}_s;$$
$$\text{TJB-HDP} \quad \forall s,\ \mathrm{P}_s \sim DP(\gamma, \mathrm{G}_0);$$

global

Correlation across groups is evident from above. (see the paper for a proof)

Popularity of Clusters Vary Across Groups



Unreal Tournament bot appear more human than humans

In the competition, computer-controlles bots created by programming teams from all over the world face off alongside human players, who act as judges,in the virtual battle zone of Unreal Tournament.

While the human players managed to gain an average "humanness" rating of 40 percent, the UT2 bots and Mirror Bots both achieved a rating of 52 percent. This is the first time since the contest has been run that a bot has achieved the target score of 50 percent humanness.

"A great deal of challenge is in defining what 'human-like' is, and then setting constraints upon the neural networks so that they evolve towards that behaviour." University of Texas doctoral student Jacob Schrum told his department website.

So what you are saying is that the bots are... Too Human?

The more interesting question here is "Why did the humans only score 40%?". The judges appear to be using a criteria that were worse then random. Why, I wonder. How many samples were there?

This actually really excites me. Imagine if we could accomplish this level of human-ish-ness in other games. No longer will be tied to the tyranny of rubber-band difficulty of bots that cheat (I'm looking at you infinite resource RTS bots). Really that's what annoys me most.

As no human was got even the magic 52% pretty musch shows that the judging crietria was totally off. If real humans as control group don't pass so judges are not looking for humanity, something else.

Specific Comment *General Comment* *Irrelevant Comment*

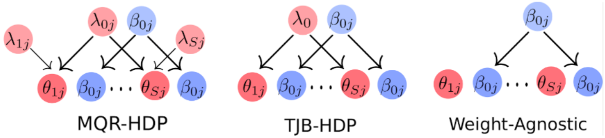
Weight-Agnosticism in Hierarchical Bayesian Models

$$\boxed{p\left(\prod_{s\in\mathcal{S}}\theta_{sj}|\Theta\right) = \prod_{s\in\mathcal{S}} p\left(\theta_{sj}|\Theta\right)}$$

θ_{sj} : weight over j^{th} component in s^{th} group

β_{0j} : j^{th} component (global)

λ_{0j} : global weight over j^{th} component



Weight-Agnostic Hierarchical Stick-Breaking Process

exploit-explore stick-breaking

$$\phi_{sn} \sim \mathrm{P}_{sn}, \ \boxed{\mathrm{P}_{sn} = \sum\nolimits_{j=1}^{k_{sn}} \theta_{sj}\delta_{\beta_{sj}} + \alpha_{sn}\Gamma}$$

$$\theta_{s1} = v_{s1}, \ \forall j > 1, \ \theta_{sj} = v_{sj}\prod\nolimits_{l=1}^{j-1}(1 - v_{sl})$$

$$\alpha_{sn} = 1 - \sum\nolimits_{j=1}^{k_{sn}} \theta_{sj}, \quad v_{sj} \sim Beta(a_j, b_j)$$

$$\Gamma_{sn}(\beta) \propto \begin{cases} 1 & \beta \in G_{sn} \\ 0 & \beta \in A_{sn} \\ \zeta & \beta \sim \mathrm{H} \end{cases}$$

$$\forall s, A_{sn} = \{\beta_{s1}, \ldots, \beta_{sk_{sn}}\}, G_{sn} = (\cup_{l \neq s} A_{ln}, \ l \in [S]$$

Relatively-diffuse probability measure

- zero mass on exploited components
- non-zero mass on components exploited in other groups
- all components in other groups are equally likely

Weight-Agnostic Hierarchical Stick-Breaking Process

$$\phi_{sn} \sim \mathrm{P}_{sn}, \boxed{\mathrm{P}_{sn} = \sum_{j=1}^{k_{sn}} \theta_{sj}\delta_{\beta_{sj}} + \alpha_{sn}\Gamma}$$

$$\theta_{s1} = v_{s1},\ \forall j > 1,\ \theta_{sj} = v_{sj}\prod_{l=1}^{j-1}(1 - v_{sl})$$

$$\alpha_{sn} = 1 - \sum_{j=1}^{k_{sn}} \theta_{sj}, \quad v_{sj} \sim Beta(a_j, b_j)$$

Relatively-diffuse probability measure

- zero mass on exploited components
- non-zero mass on components exploited in other groups
- all components in other groups are equally likely

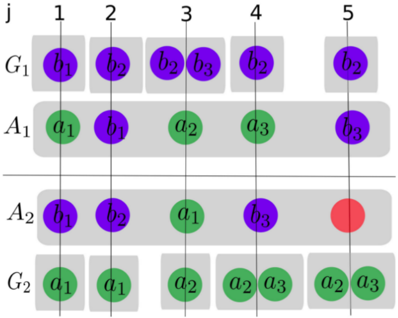
no dominance from other groups

Implication:

- No component is repeated in a group
- Components are shared across groups with non-zero probability

sharing of component is happening

Illustration of WAS



- j denoting integers : time stamps
- G : global components
- A : local components
- a in green circles :
  - components originated in group-1
- b in blue circles :
  - components originated in group-2
- red circle : next component in group-2
  - either a_2, a_3 or new

Weight-agnosticism of WAS

Theorem

Let (θ_{sj}) is as defined in WAS, and $(\beta_{0j},\ \beta_{0j} \sim \mathrm{H})$ denotes the set of shared atoms. If σ_s is the permutation function for s-th group such that weight of β_{0j} in that group is $\theta_{s\sigma_s(j)}$, then for any $\beta_{0j} \in (\cap_{s\in\mathcal{S}} A_{sn})$, where $\mathcal{S} \subseteq [S]$ following holds.

$$p\Big(\prod_{s\in\mathcal{S}} \theta_{s\sigma_s(j)}|\boldsymbol{a},\boldsymbol{b}\Big) = \prod_{s\in\mathcal{S}} p\Big(\theta_{s\sigma_s(j)}|\boldsymbol{a},\boldsymbol{b}\Big)$$

WAS is weight-agnostic hierarchical Bayesian model

Asymptotic convergence of WAS

Theorem

Let P_{sn}, (θ_{sj}) *are as defined in WAS, and* $(\beta_{0j},\ \beta_{0j} \sim \mathrm{H})$ *denotes the set of shared atoms with* σ_s *permutation function for s-th group such that weight of* β_{0j} *in that group is* $\theta_{s\sigma_s(j)}$. *If* $\mathrm{P}^*_s = \sum_{j=1}^{\infty} \theta_{s\sigma_s(j)} \delta_{\beta_{0j}}$, *such that and* $\theta_{s\sigma_s(j)} \geq 0,\ \sum_{j=1}^{\infty} \theta_{s\sigma_s(j)} = 1$; *then* $\lim_{n\to\infty} \mathrm{P}_{sn} = \mathrm{P}^*_s$ *a.s.* $\forall s$.

Mixing distribution for each group asymptotically converges to standard SBP

Empirical evaluation of Modeling and Quality

WAS can learn distribution from data

Datasets:

- NIPS-2005 Proceedings
- Obama-Speech
- BerkeleyDB
- JHotDraw

Baselines:

- Stick-breaking process (truncated version)
- Hierarchical Dirichlet Process (HDP)
- IBP compound Dirichlet process
  - IBP: Indian buffet process

Evaluation metrics:

- Perplexity: goodness-of-fit for *unseen* data
- Topic coherence: quality of clustering

TABLE I
COMPARISON OF SBP, HDP, ICD AND WAS USING *perplexity* (LESS IS BETTER).

Dataset	SBP	HDP	ICD	WAS
BerkeleyDB	60	92	86	**51**
JHotDraw	81	107	94	**72**
NIPS-05	402	435	418	**400**
Obama-speech	582	1300	1102	**412**
Average	281	483.5	425	**234**

TABLE II
COMPARISON OF SBP, HDP, ICD AND WAS USING *topic coherence* (GREATER IS BETTER).

Dataset	SBP	HDP	ICD	WAS
BerkeleyDB	-27.9	-39.6	-19.1	**-18.2**
JHotDraw	-28.2	-82.2	-46.2	**-23.2**
NIPS-05	-37.7	-49.7	-44.6	**-35.7**
Obama-speech	-52.5	-68.2	-70.9	**-42.5**
Average	-36.6	-59.9	-45.2	**-30.15**

WAS can find clusters of good quality

Empirical evaluation of Learnability and Retrieval

Dataset:

- ArsTechnica
- ArsTechnica labelled
  - Blogs with their comments
  - 500 articles labelled

Evaluation

- Comparison with truncated version
- Information retrieval

Information Retrieval Task:

- Find out specific comments for each blog

Specific Comments

- Comments related to specific parts of an article

WAS can learn model complexity

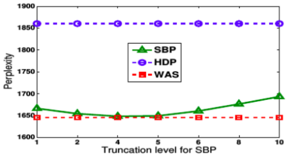


Fig. 4. Comparison of SBP, HDP and WAS on AT-Science dataset.

TABLE III
COMPARISON OF SBP, HDP AND WAS ON RETRIEVAL TASK USING *precision*, *recall* AND *F1* (GREATER IS BETTER).

Model	Precision	Recall	F1
WAS	**0.62**	**0.61**	**0.62**
HDP	0.28	0.25	0.26
SBP	0.60	0.61	0.60

WAS can detect specific comments

Thank You