Differential Topic Models

Changyou Chen, Wray Buntine, Nan Ding, Lexing Xie, and Lan Du

Abstract—In applications we may want to *compare* different document collections: they could have shared content but also different and unique aspects in particular collections. This task has been called comparative text mining or cross-collection modeling. We present a *differential topic model* for this application that models both topic differences and similarities. For this we use hierarchical Bayesian nonparametric models. Moreover, we found it was important to properly model power-law phenomena in topic-word distributions and thus we used the full Pitman-Yor process rather than just a Dirichlet process. Furthermore, we propose the transformed Pitman-Yor process (TPYP) to incorporate prior knowledge such as vocabulary variations in different collections into the model. To deal with the non-conjugate issue between model prior and likelihood in the TPYP, we thus propose an efficient sampling algorithm using a data augmentation technique based on the Multinomial theorem. Experimental results show the model discovers interesting aspects of different collections. We also show the proposed MCMC based algorithm achieves a dramatically reduced test perplexity compared to some existing topic models. Finally, we show our model outperforms the state-of-the-art for document classification/ideology prediction on a number of text collections.

Index Terms—Differential Topic Model, Transformed Pitman-Yor process, MCMC, Data Augmentation

1 Introduction

Automatic comparison of different data collections (or multiple corpora) is a broad challenge task that has been called comparative text mining [1], and is important due to the well known phenomenon of information overload. In this paper, we develop a *differential topic model* to address this task, preferring the term over cross-collection topic model [2]. For this, we want to compare topics for document collections where some of these topics capture the *shared* content among collections and others capture the *different* aspects that each collection contains. For example, in text discovery systems analysts may want to:

- compare news coverage for related companies, for instance two big supermarket chains,
- explore news bias across different media empires on key issues, *e.g.*, political leadership challenges;
- contrast reports written by different subject matter experts on an area of strategic national importance, *e.g.*, the purchase of strike fighter aircraft.

A related task is differentiating ideologies or perspectives [3], also approachable from different levels of granularity [4], for instance the sentence level.

The first topic models of this kind were developed by Zhai *et al.* [1] in the framework of PLSI [5], and later modified using an LDA style by Paul [2] and some related approaches [3, 6]. Empirical studies of these approaches were done [3, 7], and they were extended to different tasks, for instance to multi-faceted topic models where the facets are to be discovered or only partially known [8] and using linguistic analysis for additional tasks [9].

The basic idea here is that multiple collections have word usage in common but also word usage that is unique to each collection. By linking the common and unique words through a latent topic, and thus enforcing co-occurrence, the similarities and differences are discovered. The basic approach [2] is simple and fast, for instance ccLDA[1] has speeds similar to LDA.

The initial point of departure for our research is that we should explore the same ideas but in the context of *hierarchical Bayesian modeling*. In the machine learning community, a topic is defined as a collection of related words from the vocabulary [10]. In general, words are samples from a discrete distribution called the topic-word distribution. Rather than maintaining a shared word probability vector for each topic, we make the word probability vectors for specific topics across different collections have a common parent for a prior. Thus topics across collections are matched and apriori expected to share some similarities. Perhaps this more subtle approach can give better results? Those topics that are similar across collections should come from the same parent in the hierarchy. Those topics that have (reduced) similarity but also some differences, should also come from the same parent but have greater prior variation from the parent. The variance parameters for each topic in the hierarchical Bayesian model are a key handle for tuning the model, and their affect is one target for our research.

Most existing hierarchical techniques for modeling topic-word distributions are based on the Dirichlet process (DP) [11–14]. This can often be improved by using the Pitman-Yor process instead because it has

- *C. Chen, W. Buntine and L. Xie are with the Australian National University and National ICT, Australia. E-mail: {Changyou.Chen,Wray.Buntine,Lexing.Xie}@NICTA.com.au*
- *N. Ding is with Google Inc., USA. E-mail: dingnan@google.com*
- *L. Du is with the Macquarie University, Australia. E-mail: dulan520@gmail.com*

1. http://cs.jhu.edu/∼mpaul/downloads/mftm.php

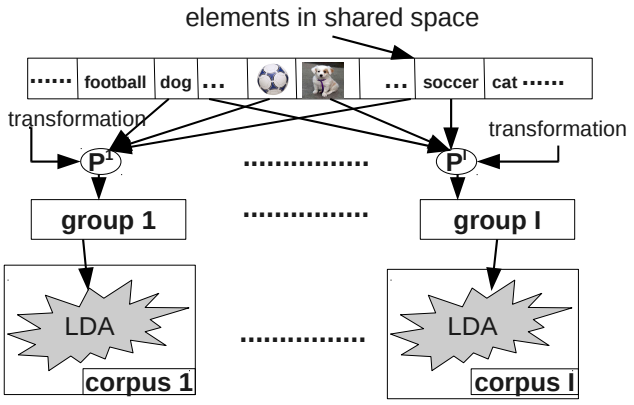


Fig. 1: Differential topic modeling using the TPYP. The top level is an abstract space that generates each sub-space for each group of documents. Each group's vocabulary subspace is formed by taking a transformation from the top abstract space.

been shown that the power-law behaviour of PYP is in line with the Zipf's law for word usage [15]. Models that can capture the power-law can perform better [16]. Therefore, another target for our research is to model the topic-word distribution with a prior having a power-law behaviour.

An important aspect we observed in initial investigations is changes in word use across collections. One can assume that all the collections might share vocabulary, however, the use of specific words might be different across collections. Here, the notion of *different word use* is that different words can be used to express the same meaning, and the same word in different collections may mean different things. For example, "takeover" and "merge" used in Australia and New Zealand stock markets respectively actually mean the same thing so they should share some information. Thus, another target of our research is to encode the information about different word use, but without losing the shared semantics in the topics. In what follows, we will show that our proposed *transformed Pitman-Yor process* (TPYP), which is defined as a *Pitman-Yor process* (PYP) with a transformed based measure, can be used to achieve the goals of power-law behaviour, different degrees of variation amongst topics, and different word use.

The general structure of our differential topic model is given in Figure 1. It consists of several LDA instances run in parallel unified with a hierarchical model of the word-topic distributions that are modelled with the TPYP.

1.1 Overview and Paper Organization

In this paper, we propose a framework to model differences of topic-word distribution among groups of datasets from different sources (each called a "collection" or "group"). The basic idea is to use the TPYP as a prior on the topic-word distribution, so that not only the power-law phenomenon is properly modeled, but also each group has different but correlated base topic-word distributions. The main contributions of this paper are:

- the use of the TPYP in a hierarchical context for differential topic modeling,
- an efficient sampling algorithm with data augmentation and re-parameterization of the TPYP,
- and state of the art results for document/ideology classification.

Experimental work shows significant improvement over baselines and related work:

- Tests on a number of datasets from different sources such as texts from news media and blogs, natural images and handwritten digit images.
- Evaluation with various criteria such as topic alignment, perplexity and opinion prediction/document classification, all show significant improvement compared with state-of-the-art baselines.

For the rest of this paper, we first review some related work on correlated topic modeling in Section 2. We then introduce the basic theories of the hierarchical PYP and TPYP in Section 3. The differential topic model using the transformed Pitman-Yor process (TPYP) is proposed in Section 4. In Section 5 we introduce an efficient algorithm for posterior inference of the TPYP. Experimental results are reported in Section 6.

2 Related Work

The proposed *differential topic model* is an instance of the general *correlated topic model* family, where we try to model different sources of correlation between documents. Correlation in topic models can be considered in two forms: (1) the correlation in *topic distributions*, the correlation between topics; and (2) the correlation in *topic-word distributions*, the correlation between words. Our model falls into the later case. There is considerable research from both perspectives, each with different motivation and algorithms.

For the first case, representative work are on shared and hierarchical topic models. Blei *et al.* proposed the correlated topic model [17], which replaces the Dirichlet prior with a logistic normal distribution. A Gibbs sampling method for this kind of model is described in [18]. Later, Paisley *et al.* extend the logistic normal distribution to a nonparametric setting and also use it for correlated topic modeling [19]. This generalizes the model of [17]. The nested Chinese restaurant process [14] models topic hierarchies by introducing a nested Chinese restaurant process (nCRP) prior on a tree. Documents are generated by drawing a set of words along the path of one branch in the tree, following the nCRP prior. Li *et al.* proposed the Pachinko Allocation model (PAM) [12] to model topic

correlations using a directed acyclic graph. In the four-level PAM, they assume words in the documents are drawn by choosing a super-topic which generates the sub-topic word distributions. Sampling is performed on an extended version of LDA with multiple levels. Du *et al.* developed a series of models exhibiting sharing across segments in a document both hierarchically and sequentially [20, 21] that were very competitive against standard LDA. Note that the above works, while hierarchical, do not consider the problem of topic sharing between groups of datasets, nor do they consider correlations among words in the topic.

On the other hand, there is also work on modeling *topic-word distributions*. Andrzejewski *et al.* [13] use a Dirichlet forest prior for the topic-word matrices so that some must-link and cannot-link constraints between words can be introduced. These constraints are modeled as preferences so the technique is quite general, and in our view should see wider use in the community. While their model is a correlation model rather than a differential model of word use, we could have employed this technique to handle shared semantics. Sato and Nakagawa use the PYP to model word distributions [16], however, they do not consider word correlations for each topic and the topic sharing between groups. Our model is thus a sharing extension of theirs. Furthermore, sparsity constraints are introduced in [22], Markov constraints are introduced in [23] in which priors for the topic-word distribution are defined as Gaussian and encoded with domain knowledge. Petterson *et al.* [24] proposed an extension of LDA using an informative prior instead of the symmetric Dirichlet prior for the topic-word distribution matrices, again without considering the problem of topic sharing between groups. Their technique is comparable in goal to Newman *et al.* [25], and our technique is basically an application of the same approach to the context of hierarchical Bayesian modeling. There are now several useful tools to model correlations in word use, and some we could explore in later work. However, our specific goal was to model differential word use.

Similar to our goal, Paul and Girju's topic-aspect model [8] extends Paul's cross-collection topic model [2]. It models different aspects within the dataset by using an extension of the LDA model. Later they combined this model with a random walk model to achieve summarizing contrastive viewpoints in opinionated text [9]. They also extended their topic-aspect model to achieve sparsity in topic distributions in [26]. Other recent related work includes Eisenstein *et al.*'s sparse additive model [6], which models the topic-word distribution by adding a set of base distributions; and Wan *et al.*'s hybrid neural network topic model [27], which incorporates the neural network to learn representative features of the input before topic modeling.

3 Background Theory

In this section we introduce the relevant background theory of the PYP, the basic notion of the TPYP, and how we do hierarchical modeling.

3.1 Modeling Topic-Word Distributions with Pitman-Yor Processes

The Pitman-Yor process and the Dirichlet process [28, 29], as non-parametric Bayesian priors, have become increasingly popular in statistical machine learning with applications found in diverse fields such as topic modeling [11], n-gram language modeling [30, 31], image segmentation [32] and annotation [33], scene learning [34], data compression [35], and relational modeling [36]. The Pitman-Yor process, denoted as $\text{PYP}(a, b, H(\cdot))$, is a random probability measure $\vec{\phi}$ defined as $\vec{\phi} = \sum_{k=1}^{\infty} p_k \delta_{x_k^*}(\cdot)$, where $\vec{p} = (p_1, p_2, ...)$ is a probability vector satisfying $p_k > 0 (\forall k)$ and $\sum_{k=1}^{\infty} p_k = 1$, and is generated through a stick-breaking process [37] or equivalent parameterized with a *discount parameter* a and a *concentration parameter* b, while the samples (atoms) x_i are independently and identically drawn from a *base probability measure* $H(\cdot)$ on space $\mathcal{X}$. We use $\{x_k^*\}$ to denote the unique values among $\{x_i\}$, and these are referred to as *types*.

Each draw from a PYP is a probability distribution with possibly infinitely many types, facilitating the use of the PYP as a prior in modeling topic-word distributions. Thus in topic modeling, the base measure $H(\cdot)$ is a probability distribution over a vocabulary space, samples x_i are words, and p_k is the probability of observing word x_k^* in a topic.

Both the Chinese restaurant process (CRP) [38] and the stick-breaking process [37] are closely related to the PYP, thus can be used in the representation. Here we use the former, and the base probability measure is discrete and finite dimensional, *i.e.*, a probability distribution over a vocabulary.

The notion of Chinese restaurant here has customers entering to be seated at tables, and each table serves a single dish and is labeled with the dish. In our case, each topic is associated with a restaurant. So observed words $\{x_i\}$ are customers in a restaurant. We will not distinguish these terms in this paper and will use them interchangeably, *e.g.*, customers$\triangleq$words. Types $\{x_k^*\}$ in a vocabulary are the dishes served at each table. So observed words in a document that are the same type are spread over tables labeled with that type. The *seating arrangement* is the assignment of observed words to tables, noting that each table can only have words of the one type, though other tables can also have the same type.

A distribution given by a seating arrangement is a word distribution for a topic (usually called a topic-word distribution). A specific seating arrangement can be generated as the follows:

- The first customer x_1 comes into the restaurant, opens a new table, and orders a dish. The type is generated from $H(\cdot)$ over a vocabulary, and the customer x_1 is assigned that type.
- All the subsequent customers come into the restaurant, and choose a table to sit as follows:
  - With probability proportional to $n_t - a$ to join an occupied table t labeled with type x_t^* (thus share the dish x_t^*), where n_t is the number of customers currently siting at table t.
  - With probability proportional to $b + aT$ to open a new table (as done for the first customer), where T is the number of occupied tables in the restaurant. A new type x_{T+1}^* is thus generated for the new table.

3.2 Transformed Pitman-Yor Processes

The above generating process requires that customers (*i.e.*, words) sitting at a table should be morphologically the same as the type attached to the table (*i.e.*, they are the same word). For example, all the customers sitting at a table labeled with a type "dog" should have the same morphological form "dog". This can be an unrealistic assumption if one wants a more semantically oriented model of topic-word distributions. For example, we might expect that words with similar meaning (e.g., "stock" and "equity") can be linked together to obtain more information sharing so that a table labeled with "equity" can have all its customers in the form of "stock", and vice versa. To address this, we propose a modified version of the PYP – the transformed Pitman-Yor process, that brings dependencies among customers/words.

To motivate this, now instead of labeling each table with a type that has the same morphological form as its customers, we consider that the morphological form of a type (of each table) can be different from its customers if the type and customers are related semantically or by stem. For example, word "dog" creates a new table, which can be labeled with one of the following types, "dog", "dogs", "doggie", "puppy", and "pooch". In other words, one can assume "dog" can be represented as a combination of the five types with different weights, i.e., a probability vector over these types. We will show with a data augmentation technique in Section 5 that this is equivalent to defining a transformed base probability measure $H(\cdot)$ on top of the PYP, which is as follows:

$$\text{TPYP}(a, b, P, H(\cdot)) \triangleq \text{PYP}(a, b, (PH)(\cdot)) \ ,$$

where P is a linear measure transformation operator that encodes the transformed probabilities from one word to other words. In topic-word distribution modeling, the base probability measure $H(\cdot)$ is discrete (we can endow it with a Dirichlet prior so $H(\cdot) \triangleq \vec{\phi}^0 \in \triangle_V$, where $\triangle_V$ denotes the V-dimensional simplex space) and P becomes a left-stochastic matrix so that each of its columns sums to one. A similar idea has been applied to the Dirichlet distribution [39]. Here we extend the idea to the full PYP and also develop an effective posterior inference algorithm. Note this is different from the transformed Dirichlet process [34] in that we do the transformation on the base measure while they do it on the components in a mixture model.

The transformed PYP fits our goal well, but we find performing a full Gibbs sample drawing elements $\vec{\phi}^0$ from the base measure $H(\cdot)$ is inefficient and impractical. Therefore we look for an approach that can marginalize out $\vec{\phi}^0$ so that a collapsed Gibbs sampler is feasible. However, it is challenging to do so due to the transformed base measure on the TPYP. This transformation breaks the conjugacy between PYP and the prior on $\vec{\phi}^0$. As a consequence we develop a novel algorithm in Section 5 that uses a data augmentation technique with a re-parameterization for the hierarchical PYP [40] (see Section 3.4) to make the marginalization analytically tractable.

3.3 Hierarchical Pitman-Yor Processes

In a typical hierarchical Bayesian topic model, a discrete probability vector $\vec{\phi}$ of finite dimension[2] V (which is the *topic-word distribution* in this paper) is sampled from some distribution family $F(\tau, \vec{\phi}^0)$, where τ is a parameter set, and $\vec{\phi}^0$ is a base probability vector of finite dimension V. In topic modeling, the Dirichlet distribution is usually used. Others, like the Dirichlet process and the Pitman-Yor process can also be included in this family. The generating process corresponding to this family samples a probability vector $\vec{\phi}$ and then a sequence of data using it. It is defined as follows:

$$\vec{\phi} \sim \ \text{F}(\tau, \vec{\phi}_0); \qquad x_i \sim \ \text{Discrete}_V(\vec{\phi}) \quad \text{for } i = 1, ..., N \ .$$

Suppose that a set of N samples is drawn from a probability distribution $\vec{\phi}$ over a discrete and finite space (In our case, the space is a vocabulary $\{1, 2, \ldots, V\}$). A count vector $\vec{m} = (m_1, \ldots, m_V)$ can be constructed from the N samples, where m_v is the number of times type v appears in the N samples, and $\sum_v m_v = N$. In Dirichlet processes and Pitman-Yor processes [41], using the Chinese restaurant process metaphor described above, an *auxiliary variable* called the *table count* can be introduced. This makes hierarchical modeling, such as with PYPs feasible, because these *table counts* require draws from the base distribution so are essentially customers in a restaurant at the next level up the hierarchy. There is a *table count* t_v for each *customer count* m_v and it represents the number of "tables" over which the m_v "customers" are spread in the restaurant. Thus

2. This can be infinite dimension, we focus on the finite dimension case in this paper for simplicity.

$1 \leq t_v \leq m_v$ and $t_v = 0$ if and only if $m_v = 0$, we denote their total as $t_\cdot = \sum_v t_v$.

When the distribution over probability vectors follows a Pitman-Yor process which has two parameters $a, b \in \tau$ and the base distribution $\vec{\phi}^0$, then $\mathrm{F}(\tau, \vec{\phi}^0) \triangleq \mathrm{PYP}(a, b, \vec{\phi}^0)$. In this case, according to [42], after integrating out $\vec{\phi}^0$, Bayesian analysis yields an augmented marginalised likelihood of

$$p\left(\vec{x}, \vec{t} | \tau, \vec{\phi}^0, \mathrm{PYP}\right) = \frac{(b|a)_{t_\cdot}}{(b)_N} \prod_v S^{m_v}_{t_v,a} \left(\phi^0_v\right)^{t_v}, \quad (1)$$

where $(b|a)_t = \prod_{n=0}^{t-1}(b+na)$ denotes the Pochhammer symbol with increment a, and $(b)_N = (b|1)_N$, and $S^N_{M,a}$ is a generalized Stirling number that is readily tabulated, as presented in [42].

3.4 Re-parameterizing the PYP

There has been existing work such as [31, 42] doing posterior inference for the PYP based on the marginalized posterior (1). However, the problem of using MCMC on (1) is that t_w's range $\{0, ..., m_w\}$ is broad and the contributions from individual data x_i seem to have been lost. As a result, MCMC can sometimes be slow. To overcome this, a re-parameterization of the PYP is proposed in [40] where instead of using the *table counts*, another set of auxiliary variables $\{r_i\}_{1:N}$ called *table indicators* are introduced. For each datum x_i, the indicator $r_i = 1$ when it is the "head (creator) of its table" (recall the m_w data are spread over t_w tables, each table has and only has one "head"), and zero otherwise. It can be seen that $t_w = \sum_{i=1}^N 1_{x_i=w} 1_{r_i=1}$. Moreover, if there are t_w tables then there must be exactly t_w heads of table, and it is equally likely as to which data are heads of table, thus the posterior of the model using this set of auxiliary variables is (from (1))

$$p\left(\vec{x}, \vec{r} | \tau, \vec{\phi}^0, \mathrm{PYP}\right) = p\left(\vec{x}, \vec{t} | \tau, \vec{\phi}^0, \mathrm{PYP}\right) \prod_w \binom{m_w}{t_w}^{-1}. \quad (2)$$

As shown in [40], a block Gibbs sampler for (x_i, r_i) is easily derived from (2). Since $\vec{r}$ only appears indirectly through the table counts $\vec{t}$ and it is uniformly distributed conditioned on customers on the same table. We do not need to store the $\vec{r}$, we just resample an r_i when needed according to the proportion t_{x_i}/m_{x_i}. We will follow this representation of the PYP in this paper, not only because it allows more efficient sampling, as is shown in [40], but also because it allows us to do data augmentation for the TPYP more easily, as will be shown below.

4 Modeling with Transformed PYP

We build differential topic models using the popular topic model LDA [10] as a building block, but with the TPYP as the prior for the word-topic distributions.

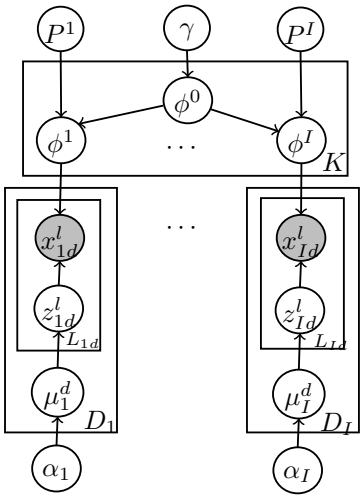


Fig. 2: Graphical model of differential topic modeling.

As illustrated in Figure 1, there are multiple groups of data. Each group consists of a set of data collected from a particular source, *e.g.*, news articles from a particular region. We use a TPYP to model its *topic-word* distributions with a group-specific transformation matrix P^i. Together the groups share a common base measure $\vec{\phi}^0$. Since data from different sources could be quite different, we can think of the common base measure defined on an abstract space, *i.e.*, samples from this space are not necessarily restricted to be words, they could be the index of a synonym set.

Notationally, we use i to denote the group index which ranges over $1..I$, d to denote document index for each group which ranges over $1..D_i$, l to denote word index for each document which ranges over $1..L_{i,d}$, k to denote topic index which ranges over $1..K$, and (w, v) to denote row index and column index of the transformation matrices $\{P_i\}$. Given a vocabulary of size V, the transformed matrices $P^i = (p^i_{wv})_{V\times V}$ are sparse matrices. For each word w, we allow it to be associated with the most similar words so that each row of P_i will only have a few nonzero entries. Note that these matrices provide prior information of how words are correlated and are not learned by the model (see the experimental part for the construction). With these indices and dimensions, data are represented in two sets, which are listed in the following together with some statistics:

- $\mathbf{X}$: the words in documents, x^l_{id} for $i = 1..I, d = 1..D_i, l = 1..L_{i,d}$;
- $\mathbf{Z}$: the latent topic of each word, z^l_{id} for i, d, l.
- m_{ikw}: number of words w in group i for topic k.
- t_{ikw}: the corresponding table count, for use in Equation (1).
- n_{idk}: the number of words for topic k in document d of group i.
- $\mathbf{R}$: the table indicator for each word, r_{idl} for i, d, l.

For simplicity, dots denote marginal sums, *e.g.*, $m_{ik\cdot} = \sum_w m_{ikw}$ and $t_{ik\cdot} = \sum_w t_{ikw}$.

The generating process for our model as illustrated in Figure 2 (right) is then as follows:

$$\begin{aligned}
\vec{\phi}_k^0 &\sim \text{Dirichlet}(\vec{\gamma}) && k = 1..K \\
\vec{\phi}_k^i &\sim \text{TPYP}(a_k, b_k, P^i, \vec{\phi}_k^0) && i = 1..I, k = 1..K \\
\vec{\mu}_i^d &\sim \text{Dirichlet}(\vec{\alpha}_i) && i = 1..I, d = 1..D_i \\
z_{id}^l &\sim \text{Discrete}(\vec{\mu}_i^d) && i = 1..I, d = 1..D_i, l = 1..L_{i,d} \\
x_{id}^l &\sim \text{Discrete}(\vec{\phi}_{z_{id}^l}^i) && i = 1..I, d = 1..D_i, l = 1..L_{i,d}
\end{aligned}$$

Using the Dirichlet-multinomial and PYP-multinomial conjugacy we can easily marginalize out $\vec{\mu}_i^d$ and $\vec{\phi}_k^i$ in the above generative process. Together with the marginal posterior of the PYP in (2), we obtain the following marginal posterior $p(\mathbf{X}, \mathbf{Z}, \mathbf{R}, \vec{\phi}^0 | \vec{a}, \vec{b}, \vec{\alpha}_{1:I}, \vec{\gamma}, \mathbf{P}^{1:I}) =$

$$p(\mathbf{Z}|\vec{\alpha}_{1:I}) p(\vec{\phi}^0|\vec{\gamma}) p(\mathbf{X}, \mathbf{R}|\mathbf{Z}, \vec{a}, \vec{b}, \vec{\phi}^0, \mathbf{P}^{1:I}) \ , \qquad (3)$$

where $p(\mathbf{Z}|\vec{\alpha}_{1:I}) = \prod_{i=1}^I \prod_{d=1}^{D_i} \frac{Beta_K(\vec{\alpha}_i + \vec{n}_{id})}{Beta_K(\vec{\alpha}_i)}$,
$p(\vec{\phi}^0|\vec{\gamma}) = \prod_{k=1}^K \frac{1}{Beta_V(\vec{\gamma}_k)} \prod_{w=1}^V \left(\phi_{kw}^0\right)^{\gamma_k - 1}$,
$p(\mathbf{X}, \mathbf{R}|\mathbf{Z}, \vec{a}, \vec{b}, \vec{\phi}^0, \mathbf{P}^{1:I}) = \prod_{i=1}^I \prod_{k=1}^K \frac{(b_k|a_k)_{t_{ik\cdot}}}{(b_k)_{m_{ik\cdot}}} \prod_{w=1}^V S_{t_{ikw},a_k}^{m_{ikw}} \left(\sum_{v=1}^V p_{w,v}^i \phi_{kv}^0\right)^{t_{ikw}} \binom{m_{ikw}}{t_{ikw}}^{-1}$.

and $Beta_K(\cdot)$ is a function normalizing the K-dimensional Dirichlet.

Note that the above marginal posterior yields poor direct MCMC sampling because of the high-dimensional continuous variable $\vec{\phi}^0$ (in our model it has V dimensions, the vocabulary size). In order to derive an efficient sampler, we should collapse it into the posterior as well. In the following section, we use a data augmentation technique based on the Multinomial theorem by introducing new auxiliary variables that enables us to marginalize out $\vec{\phi}^0$.

5 Posterior Inference

Now we describe the posterior inference algorithm for our model. To better illustrate the intuition, we simplify our notation. Let us first consider when $K = 1$ and $I = 1$ in (3), so that we drop out the indexes i and k, resulting in $p(\mathbf{X}, \mathbf{R}, \vec{\phi}^0 | \mathbf{Z}, \vec{a}, \vec{b}, \mathbf{P}, \vec{\gamma}) =$

$$\frac{1}{Beta_V(\vec{\gamma})} \prod_{w=1}^V \left(\phi_w^0\right)^{\gamma-1} \frac{(b|a)_{t\cdot}}{(b)_{m\cdot}} \prod_{w=1}^V \frac{S_{t_w,a}^{m_w}}{\binom{m_w}{t_w}} \left(\sum_{v=1}^V p_{w,v} \phi_v^0\right)^{t_w} \qquad (4)$$

The idea of our algorithm is to notice that the summation terms in $\prod_w \left(\sum_v p_{w,v}\phi_v^0\right)^{t_w}$ can be turned into products by introducing column indexes v_{wt} for $t = 1...t_w$ as auxiliary variables. To illustrate this, suppose $t_w = 2$,

$$p(..., t_w, ...) = ... \left(\sum_v p_{w,v}\phi_v^0\right)^{t_w} ... =$$
$$... \left(\sum_{v_{w1}} p_{w,v_{w1}}\phi_{v_{w1}}^0\right)\left(\sum_{v_{w2}} p_{w,v_{w2}}\phi_{v_{w2}}^0\right) ... \xrightarrow{\text{augmenting}}$$

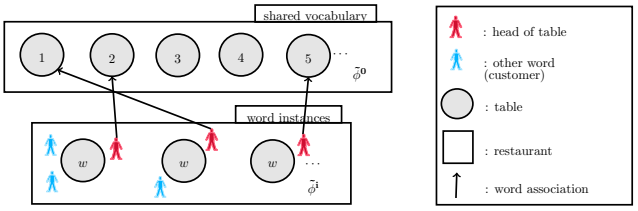


Fig. 3: Illustrates the latent variables associated with the $m_{ikw} = 6$ words of index w with topic k in collection i: each table has a single "head of table" marked in red and there are $t_{ikw} = 3$ in total. The head must choose a single word in the abstract space to associate with, its entry is in $\vec{v}_{ikw}$.

$$p(..., t_w, v_{w1}, v_{w2}, ...) = ... \ p_{w,v_{w1}} \phi_{v_{w1}}^0 p_{w,v_{w2}} \phi_{v_{w2}}^0 \ ... \qquad (5)$$

The last line augments the probability with the two separate auxiliary variables v_{w1}, v_{w2}, and note the augmentation is *reversible* by a marginalisation step, see Appendix B for the proof and detail deviation of these variables applying to our full model. Using this trick (with auxiliary variables $\{v_{wt}\}$), $p(\mathbf{X}, \mathbf{R}, \vec{\phi}^0 | \mathbf{Z}, \vec{a}, \vec{b}, \mathbf{P}, \vec{\gamma})$ can be augmented into a product form proportional to:

$$\prod_{w=1}^V \left(\phi_w^0\right)^{\gamma-1} \cdot \frac{(b|a)_{t\cdot}}{(b)_{m\cdot}} \prod_{w=1}^V S_{t_w,a}^{m_w} \binom{m_w}{t_w}^{-1} \prod_{t=1}^{t_w} p_{w,v_{wt}} \phi_{v_{wt}}^0$$

It is clear that the conjugacy of $\vec{\phi}^0$ with its Dirichlet prior is obtained so that it can be integrated out.

Now apply this data augmentation trick to the full model (3), and we have the set of auxiliary variables as $\{v_{ikwt}\}$[3], each associated with word w in topic k, table t and group i. By carefully inspecting the role of $\vec{v}_{ikwt}$, we can see that:

Remark Using the Chinese restaurant metaphor for the TPYP in Section 3.2, v_{ikwt} denotes the *type* marked on table t by customer w for topic k in group i.

The above observation explains why we define TPYP as a PYP with a transformed base measure, *i.e.*, word w in topic k of group i is associated with word v_{ikwt} in the global vocabulary, and v_{ikwt} itself is random. We call $\vec{v}_{ikw}$ with dimension t_{ikw} the *word associations*, and the full set denoted as $\mathbf{V}$. Note that now $\mathbf{V}$ is closely connected to the table indicators $\mathbf{R}$: each word x_{dl}^i has a table indicator r_{dl}^i to say if it is "head of its table". If the table indicator is 1, it creates the table and marks it with a *type*, which is the word association $\vec{v}_{ikw}$ (please refer to (5)). Otherwise it has none. This situation is represented in Figure 3.

Now defined an auxiliary statistic $\tilde{q}_{kwv}^i$ as the number of word associations taking on a particular value v: $\tilde{q}_{kwv}^i = \sum_{t=1}^{t_{ikw}} 1_{v_{ikwt}=v}$. The $\tilde{q}_{kwv}^i$ can be interpreted as a statistic giving how relevant the word w in group i and topic k is with respect to the global word v.

3. We put the indexes i and k back into v in the following.

Interesting results learnt about this in experiments are shown in Section 6.2.6. Now marginalising out the $\vec{\phi}^0$ yields a collapsed posterior with these statistics:

$$
\begin{aligned}
& p\left(\mathbf{X}, \mathbf{Z}, \mathbf{V}, \mathbf{R} | \vec{a}, \vec{b}, \vec{\alpha}_{1:I}, \vec{\gamma}, \mathbf{P}^{1:I}\right) \qquad (6) \\
= & \prod_{i=1}^{I} \prod_{w=1}^{V} \prod_{v=1}^{V} \left(p_{wv}^{i}\right)^{\sum_k \tilde{q}_{kwv}^{i}} \prod_{i=1}^{I} \prod_{d=1}^{D_i} \frac{Beta_K\left(\vec{\alpha}_i + \vec{n}_{id}\right)}{Beta_K\left(\vec{\alpha}_i\right)} \\
& \prod_{k=1}^{K} \left\{ \prod_{i=1}^{I} \frac{(b_k|a_k)_{t_{ik\cdot}}}{(b_k)_{m_{ik\cdot}}} \frac{\prod_v \Gamma\left(\gamma_v + \sum_i \sum_w \tilde{q}_{kwv}^{i}\right)}{\Gamma\left(\sum_v \gamma_v + \sum_i \sum_w t_{ikw}\right)} \right\} \\
& \prod_{k=1}^{K} \left\{ \frac{1}{Beta\left(\vec{\gamma}\right)} \prod_{i=1}^{I} \prod_{w=1}^{V} S_{t_{ikw}, a_k}^{m_{ikw}} \binom{m_{ikw}}{t_{ikw}}^{-1} \right\} .
\end{aligned}
$$

Note that the square product $\prod_{w=1}^{V} \prod_{v=1}^{V}$ is only computed for elements on the sparse matrices $\mathbf{P}^i$, thus computational complexity is bilinear in V and the level of sparsity, *i.e.*, $O(S\tilde{W}_i)$ where $\tilde{W}_i$ is #types in group i and S is #words associated with each vocabulary word (10 in our construction).

Based on this representation, the corresponding Gibbs sampling algorithm samples latent variables for the word x_{dl}^i sequentially for each group i, document d, and word l. The $(z_{dl}^i, r_{dl}^i, v_{ikwt})$ are sampled as a single block (though v_{ikwt} is ignored when $r_{dl}^i = 0$). This step is carried out as follows: first remove counts from the statistics using Algorithm 1, and then sample a new topic, table indicator and potentially a word association using Algorithm 2. The sampling step given in Algorithm 2 compiles the proportionality of Equation (6). Note that the table indicators $\mathbf{R}$ are not stored, but the table counts $\mathbf{T}$ and the word associations $\mathbf{V}$ are stored.

Algorithm 1 Decrement word $x_{d,l}^i$

1: $k = z_{dl}^i, w = x_{dl}^i$
2: Resample the table indicator by generating a Bernoulli random variable:
$$r_{idl} \sim \text{Bernoulli}\left(\frac{t_{ikw}}{m_{ikw}}\right) .$$
3: Decrease the corresponding statistics m_{ikw}, n_{idk}.
4: **if** $r_{idl} \equiv 1$ **then**
5: decrease the corresponding table count t_{ikw},
6: sample $t \sim \text{Uniform}(1, \cdots, t_{ikw} + 1)$,
7: remove t-th element from the list $\vec{v}_{ikw}$, and
8: decrease $q_{kwv_t}^i$.
9: **end if**

5.1 Handling Hyperparameters

For parameters a_k, b_k, one way is to introduce Gamma, Beta, and Bernoulli variables to sample both, as was done by Teh [30]. However, this requires recording the number of customers on each table and could be expensive. The other way is to fix $a_k > 0$ and use an adaptive rejection sampler to sample b_k's, as was done by Du *et al.* [20]. We implemented both methods and used the second in these experiments as it produced better training likelihoods. For the Dirichlet parameters $\vec{\gamma}$ and $\vec{\alpha}_i$, we consider the symmetric case and optimize them using the Newton-Raphson method [43]. They tend to have focused posteriors and thus optimization is quite adequate.

Algorithm 2 Sample word $x_{d,l}^i$

1: For each k, calculate the following proportionalities:
- $p(z_{idl} = k, r_{idl} = 0 | \text{others})$:
$$\propto \frac{\alpha_{ik} + n_{idk}}{b_k + m_{ik\cdot}} \frac{m_{ikw} - t_{ikw} + 1}{m_{ikw} + 1} \frac{S_{t_{ikw}, a_k}^{m_{ikw}+1}}{S_{t_{ikw}, a_k}^{m_{ikw}}} .$$
- $p(z_{idl} = k, r_{idl} = 1, v_{ikwt} = v | \text{others})$:
$$\propto \quad p_{wv}^{i} \left(\alpha_{ik} + n_{idk}\right) \frac{b_k + a_k t_{ik\cdot}}{b_k + m_{ik\cdot}} \frac{t_{ikw}+1}{m_{ikw}+1} \frac{\gamma_v + \sum_i \sum_w q_{kwv}^{i}}{\sum_{v'} \gamma_{v'} + \sum_i \sum_w t_{ikw}} \frac{S_{t_{ikw}+1, a_k}^{m_{ikw}+1}}{S_{t_{ikw}, a_k}^{m_{ikw}}} .$$
2: Jointly sample z_{idl}, r_{idl} and v_{ikwt} according to these probabilities.
3: Increase the counts of the statistics m_{ikw} and n_{idk}.
4: **if** $r_{idl} \equiv 1$ **then**
5: increase the counts of the statistics t_{ikw} and q_{kwv}^i, and add v to the list $\vec{v}_{ikw}$.
6: **end if**

5.2 Variational Inference

In addition to the Gibbs sampling algorithm developed above, another possibility for posterior inference is to use variational inference technique [44]. We developed two hybrid Gibbs and variational algorithms for the model. See Appendix A for details of the development and Appendix C for the corresponding comparisons. The main technique is to use the Jessen's inequality to upper bound the power term in the likelihood (3).

6 Experiments

We tested our models on a variety of datasets, including six text datasets, one natural image dataset and one handwritten digit dataset. We will first give some illustrations in the next section.

6.1 Illustrations

First, by thresholding the variance parameter (represented as b_k in the definition of PYP, the larger, the more similar the topic pair is), our model automatically aligns some topics between groups, while also leaves some other topics effectively unaligned. Figure 4 shows an example of different issues discovered by our model from two blog media, *Daily Kos* and *Right Wing News*. This dataset is described later, and denoted BD. For the paired topics in the lower left of the figure, seemingly on *policy issues*,

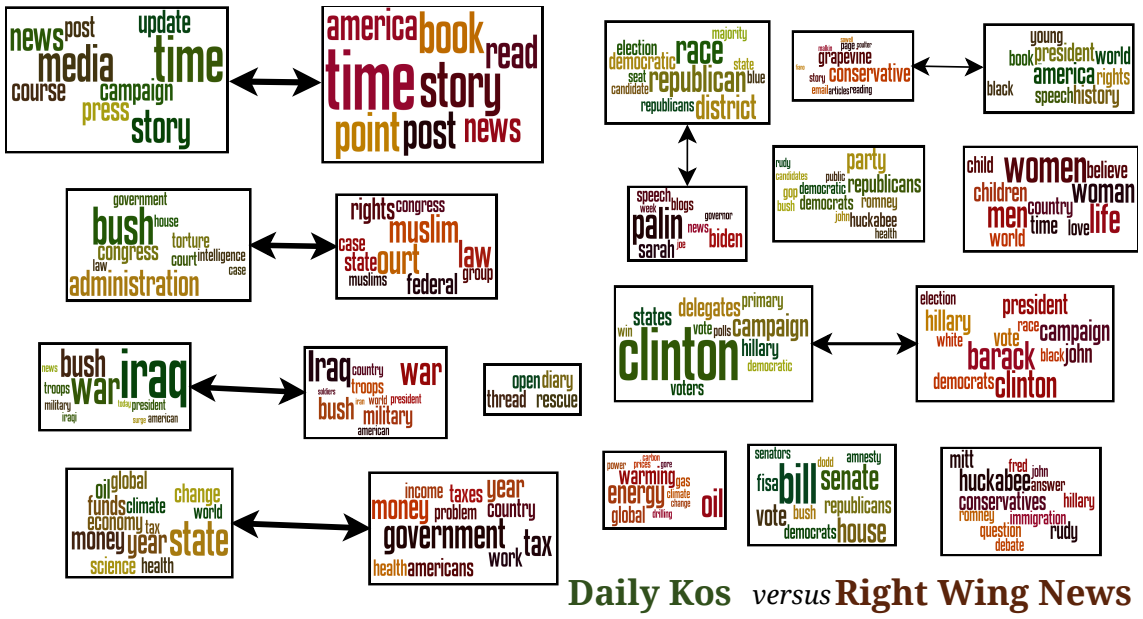


Fig. 4: An example of topic differences between the blogs of *Daily Kos* (green boxes, size proportional to the frequency of the topic), Democrats, and *Right Wing News* (red boxes), Republicans, best viewed in color. The arcs represent the similarity strength of topic pairs. Word sizes are proportional to their frequencies.

the Democrat group is concerned with global issues, the economy, and climate and change, while the Republican group emphasizes government, income, Americans and taxes. It is also interesting to see the Republicans discussing issues to do with family and life (right, second from the top) and energy and oil (middle, bottom) whereas the Democrats have no comparable topics.

Second, note that our differential topic model defines a hierarchical structure on topic-word distributions. Table 1 shows an example of the topic hierarchy learned on a Reuters News dataset GENT consisting of 6 groups, which is described below. It is interesting to see that for the general topic with concentration $b = 4958$, the six children topics across the different regions are almost identical. For the "movie star" topic with concentration $b = 103$, which means the children topics vary across the different regions, we can see the regional focus for movie stars: Cannes in Europe, the Oscars in the USA, and the movie "Evita" in South America. A figure showing more interesting topic pairs can be found in Appendix C. Note we have also tried the TAM model [8] for these two illustrations but found much less interpretable topics, thus we do not show the results here.

6.2 Topic Modeling in Text

6.2.1 Datasets

For these experiments, we extracted three datasets from the *Reuters RCV1* collection[4] about disasters, entertainment and politics, the Reuters categories GDIS, GENT and GPOL respectively. Sentences were parsed with the C&C Parser[5], then lemmatised and function words discarded. The lemmas were then readily used with the transformation matrices below. To divide the three Reuters datasets into groups, we split them into 6 groups according to their location, *i.e.*, Middle Asia, Africa, South America, North America, Europe, East Asia and Oceania. Articles from multiple regions are multiply included. GDIS has a vocabulary of size 39534, with 1508, 443, 1315, 1833, 1580, 2418 documents in each group. The GENT dataset has 43990 words in the vocabulary, 308, 78, 285, 1413, 348, 1694 documents in each group. While the biggest GPOL dataset has 109586 words in the vocabulary and 8464, 3227, 4033, 14593, 5517, 9339 documents in each group. A typical document is 200–400 words.

We also used the political blog data from [45], but only used the 9560 main blog entries by "Carpetbagger Report", "Daily Kos", "Matthew Yglesias", "Red State" and "Right Wing News", removing comments. This had already been segmented and tokenised so we discarded words appearing less than 5 times or more than 9,500 times in total. Remaining was a vocabulary of size 18038 with 1201, 2599, 1828, 2485 and 1447 blogs entries respectively in the five blogs. This dataset is denoted as BD.

Finally we crawled and parsed the abstracts for the *Journal of Machine Learning Research* volumes 1–11, the *International Machine Learning Conference* years 2007–2011, and *IEEE Trans. of PAMI* 2006-2011. Simple

4. Reuters Corpus, Volume 1, English language, 1996-08-20 to 1997-08-19 (Release date 2000-11-03).

5. http://svn.ask.it.usyd.edu.au/trac/candc

TABLE 1: Two topic hierarchies for GENT dataset. The left most column of each topic is the master topic, while the others correspond to topics in the six region. Values of b reflects the similarity of the region topics.

Global	Middle East	Africa	South America	Europe	USA	East Asia&Oceania
	Topic_1: $b = 103$					
stars	film	government	film	film	best	film
movie	films	national	movie	festival	film	films
star	bombay	circumcision	Evita	films	actor	festival
rupees	Somalia	Madonna	best	best	actress	Cannes
venice	cinema	practice	Peron	director	Oscar	best
president	India	circumcised	Argentine	Cannes	awards	director
	script	Inkatha	director	actor	won	shine
	Topic_4: $b = 4958$					
years	years	years	years	years	years	years
life	life	life	time	time	life	life
time	time	time	life	life	time	time
world	work	world	work	world	world	world
work	book	work	world	book	show	show
book	world	book	book	made	made	made
	home	young	year	work	work	work

tokenisation was done (splitting on spaces and punctuation, case ignored, leaving contiguous letters and numbers) to create words. Stop-words were discarded as well as words appearing less than 5 times or more than 2900 times. This resulted in a vocabulary of size 4660 with 818, 765 and 1108 documents respectively. This is denoted as MLJ.

Note that we further tokenised and also leammatised the GDIS, GENT and GPOL datasets, thus we have in total 8 datasets for the experiments. In the following we use postfix "*cc*" to denote the datasets with tokenisation and postfix "*ccp*" to denote those with lemmatisation assisted by the C&C Parser.

6.2.2 Transformation Matrices

We constructed two different transformation matrices. First, we ran a sliding window of size 20 along the full text of entries in the Wikipedia of December 2011 (discarding tables, category, list and disambiguation pages)[6]. Co-occurrence statistics were then computed and only the top 10 pairs were kept for each word in order to introduce sparsity for the transformation matrices, and a uniform probability given to the 10 or less alternatives. This matrix is labeled co. Second, Ted Pedersen's Perl package WordNet::Similarity::vector was used to compute the geometric mean of similarity between word lemmas, and those less than 0.2 were discarded. This matrix is labeled wn.

Specifically, for each word w in the local vocabulary of group i, we looked for the 10 most related words $(v_1, \cdots, v_{10})$ from the global vocabulary, we then filled in the entries $\{(w, v_1), \cdots, (w, v_{10})\}$ of the transformation matrix P^i with their word correlation values. It can be seen that by doing this, each group would statistically focus on its local vocabulary but can also enjoy the global information sharing. Note we built the transformation matrices on the training sets for fair comparison.

6. Using the wex2link and linkCoco programs in https://forge.nicta.com.au/projects/dca-bags.

6.2.3 Measuring Perplexity

Perplexity was measured on a test set, 20% of the original data sets, and was done using the standard dictionary hold-out method (50% of document words were held out when estimating topic probabilities) [46] known to be unbiased. The results are presented as the average over six runs for each dataset with different initializations.

6.2.4 Implementation

We compared our model with a number of baselines, which are listed below. All algorithms except ccLDA are implemented in C, have been extensively tested, and reviewed by multiple coders. The models we compare are (see Appendix for more model comparison including the variational inference):

- TI: the full Gibbs table indicator sampler for the TPYP.
- TII: a degenerated TI with identity transformation matrix I.
- CS: the collapsed Gibbs sampler for the HPYP [47].
- SS: a variant of the CRP based algorithm, originally the sampling by direct assignment algorithm proposed for the HDP [11].
- PYP: use PYP as the prior for the topic-word distributions for each group separately [16].
- ccLDA: cross-collection topic models [2].
- LDA: plain LDA [10] trained on each group.

Note only the first algorithm deals with non-identity transformation matrices, thus have incorporated word correlation information via the transformation matrices, while the others do not. Since we construct the transformation matrix in two ways, we will use subscripts 'co' and 'wn' to denote the algorithms using the matrices constructed from Wikipedia and WordNet, respectively. All the algorithms were run using

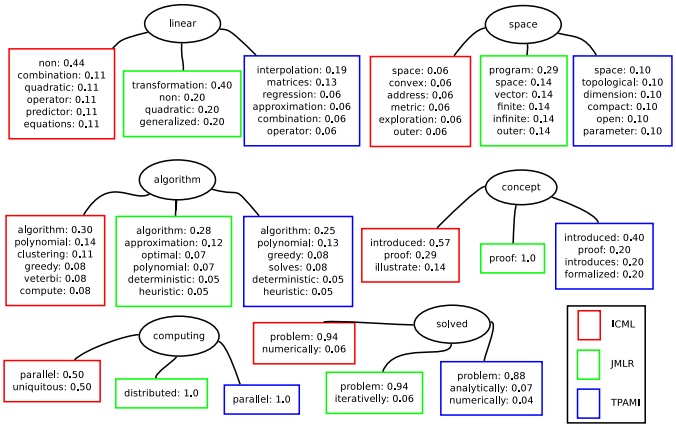


Fig. 5: Some word association structures on MLJ dataset. The words in the eclipses are from the global vocabulary, each corresponds to a set of words (in the colored boxes) in each group, represented by the statistics $\{\tilde{q}^i_{kwv}\}$ in (6), the numbers following the words represent the strength of the correlations in range $[0, 1]$. Best viewed in color.

2000 Gibbs/variational cycles as burn in, which was adequate for convergence in the experiments, and 100 samples were collected for the perplexity calculation. The hyperpameters were also sampled, but with the *discount parameter* a set to 0.7, known to perform well for topic-word distribution modeling in text.

6.2.5 Result: Topic Alignment

First, due to the nature of our model, it does automatic alignment of topic, performing this task as well as a standard baseline. See Appendix C for details.

6.2.6 Result: Word Associations

In our inference algorithm we have introduced the auxiliary variables called *word associations*, and defined an auxiliary statistic $\tilde{q}^i_{kwv}$ derived from *word associations*. From the definition, we can think of $\tilde{q}^i_{kwv}$ as how relevant word w in group i for topic k is to the word v in the global vocabulary, the larger, the more relevant. Fig. 5 shows an example of these *word associations* trained on the MLJ dataset and picked from a subset of the words within one topic. It is interesting to see that we can also tell the topic difference based on these relations. For example, for the word "computing", words associated in ICML are "parallel" and "ubiquitous", while in JMLR and TPAMI, they focus on "distributed" and "parallel", respectively.

6.2.7 Result: Perplexity Comparison

We first compare the 7 Gibbs sampling based algorithms described in Section 6.2 on the 5 datasets, the variational based methods are not shown here because of their bad performance. We use the transformation matrices constructed from Wikipedia for our model. The results are shown in Figure 6. The main observations are:

- TI performs significantly better than other algorithms. This means semantic information is important, and can be neatly dealt with by the proposed TPYP.
- TII is consistently better than the other sampler for the PYP, *e.g.*, CS and SS, which shows the superiority of our table indicator sampling.
- ccLDA is worse than TII (thus TI) in most cases, and generally better than the other methods, except in the BD and GPOL datasets where it performs poorly.
- In the MLJ dataset TI is slightly worse than TII and ccLDA. This might because on the very specific subject domain of machine learning, the transformation matrices did not help.

6.2.8 Result: Full Comparison

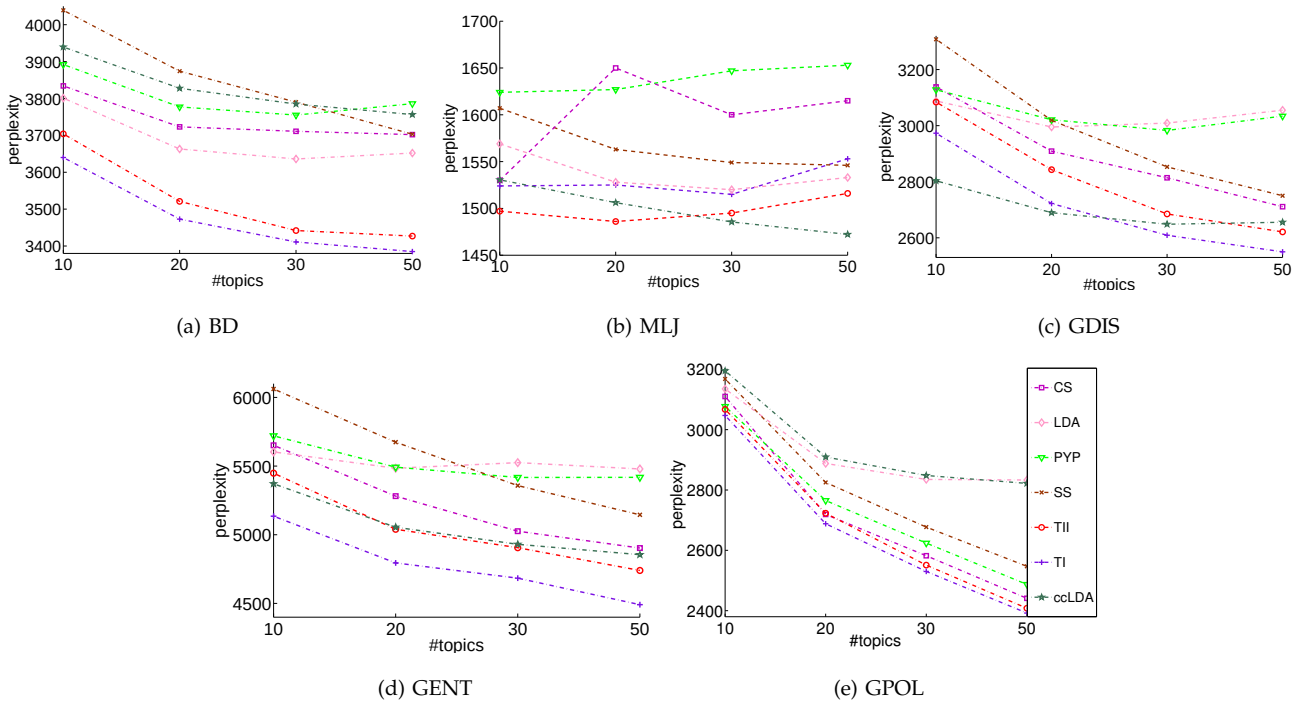
This section shows the performance of different models under different experimental settings, *e.g.*, different hyperparameters, different transformation matrices and datasets with different preprocessing, *etc.*. The following summarises these results.

First, we claimed that the Pitman-Yor process should be better as a model of word probability vectors than the Dirichlet process. For this series of experiments we fix discount $a = 0.70$ as the approximate value known to perform well in topic-word distribution modeling. The claim are confirmed by Figure 7(a). Second, we claimed that the new table indicator sampler should perform better than the original sampler used by Teh *et al.* [11]. This is confirmed in Figure 7(b). Third, we expect that by introducing semantic information into the model, TI should perform better than the plain PYP–TII. This is confirmed by Figure 7(c).

Finally, a summary of all the algorithms and datasets with different transformation matrix settings is shown in Appendix D.

6.3 Topic Modeling in Natural Images

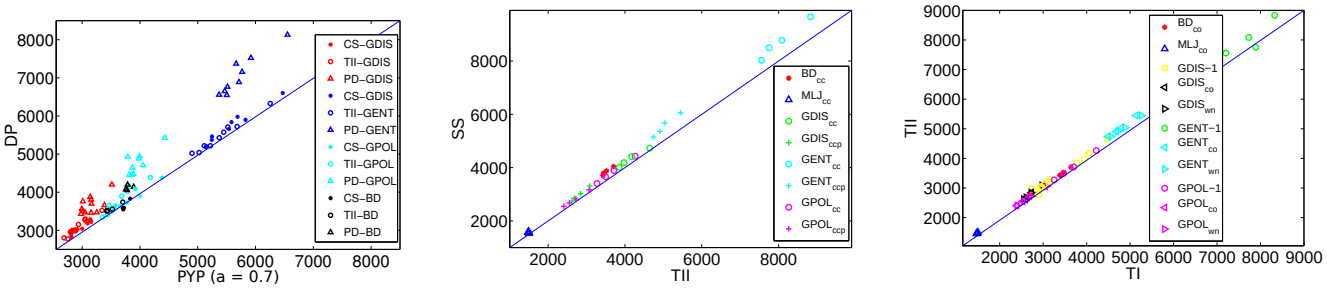
We also carried out a pilot evaluation of differential topic modeling on image datasets. Two example pairs of contrasting image collections are taken from ImageNet [48], *i.e.*, one for *mango* versus *pineapple* and the other *bike* versus *car*. We turned each image into bag-of-words representation by using the densely sampled bag-of-visual-words [48] to describe 300 images from each collection, where 128-dimensional SIFT descriptors are extracted from evenly spaced image patches and then quantized into 1000 visual words with K-means. Refer to [48] for detailed descriptions. We ran TII with 20 topics in this experiment since it was found to yield well-aligned topics. Other settings have similar results. Figure 8 shows one example aligned topic for *bikes versus cars*. Each topic is illustrated as



(a) BD (b) MLJ (c) GDIS

(d) GENT (e) GPOL

Fig. 6: Perplexities versus #topics on the five datasets, best viewed in color.



(a) PYP versus DP (b) Table indicators versus Direct sampling (c) TPYP versus PYP

Fig. 7: Comparison of test perplexity for different algorithms and datasets. Each point corresponds to one parameter setting. "A-B"in the legend indicates the algorithm "A" and data set "B", while subscripts "ccp" and "cc" mean the dataset with and without tokenisation preprocessing, "wn" and "co" mean the 2 kinds of transformation matrices. Postscript "-1" in (c) means the original datasets without stop word removal.

the average of image patches that belong to the top 49 visual words (top). We can see from Figure 8 that our model captures the shared structure in different categories, *i.e.*, patches with horizontal structures (top), found on both bikes and cars (bottom). Rather than using the perplexity measure, we validated the ability of `TII` to identify objects in the different groups, since these groups tended to have similar background, but remarkably different foreground objects. We measured the ratio of the number of visual words in the same topic that fell within the object bounding boxes and those outside, we called this ration *localization ratio* for short. Figure 8 (right) shows the results in comparison with LDA. We can see that `TII` has higher localization ratios than LDA, especially over the first few most discriminating topics.

6.4 Topic Modeling in Handwritten Digits

To further illustrate our model with images, we tested it on the `BinaryAlphaDigs` dataset[7]. It contains binary 20×16 digits of "0"–"9" and capital "A"–"Z" and there are 39 examples for each class. The images are represented by binary matrices; each pixel is regarded as one word in the vocabulary, a pixel with value "1" indicates existing of this word in the corresponding image, while "0" indicates absent. Furthermore, we divided the whole dataset into 36 groups, each corresponded to one class. Different from the above

7. http://www.cs.toronto.edu/~roweis/data.html

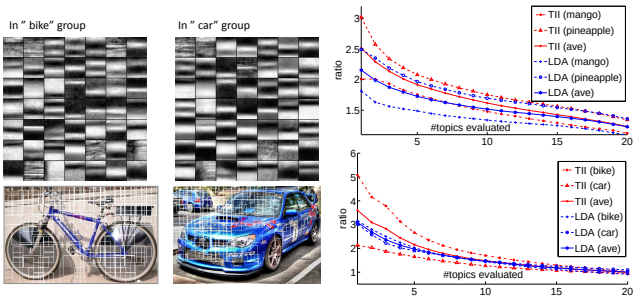


Fig. 8: Results on image datasets. Left: an example topic. Top: average of patches in its top 49 visual words; bottom: the locations of these patches on the images. Right: object localization. x-axis: #topics considered; y-axis: average *localization ratio* over topics (larger is better).Dash line: scores on groups; Solid: average scores. Best viewed in color.

experiment, in this setting, each word indicates the existence or absent of a pixel in the corresponding location, thus a topic can be visualized using an image, with pixel values equal to the weights of the corresponding words in this topic. Because it was not straightforward to construct a suitable transformation matrix for this dataset, we used the version TII for testing. Instead of showing the perplexities (which we found comparative for all the models), we show the specific topics learned by our model and LDA in Fig. 9. We can see from the figure that TII manages to learn the sharing structures among all the characters while each of them varies smoothly between groups. In particular, we see that the first topic in TII represents the shapes of different characters, while the other are shared variations among all characters. This is not observed in the topics learned by LDA, which seems to be some random patches.

6.5 Document Classification

We further evaluated our model in the task of document classification. The most popular technique for this task currently is the *support vector machine* (SVM) [49], which usually achieves the state-of-the-art. Some probabilistic models such as [6] can achieve comparable performance with SVM in particular datasets. Therefore, we compare our model TI with SVM on the *BD*, *Blog* and *MLJ* datasets. *BD* and *MLJ* are the same dataset used in Section 6.2 and *Blog* contains six political blogs about the U.S. presidential election [6]. After training our model, we did the classification by computing the marginal likelihoods of the testing documents, where we used the shared global *topic-word* distributions $\vec{\phi}_k^0$ to estimate the topic distribution $\vec{\theta}_d$ for each testing document d (by running a standard LDA inference with topic-word distributions fixed as $\vec{\phi}_k^0$), we then simply assigned the testing document to group $c = \arg\max_i \sum_{\ell=1}^{N_d} \sum_{k=1}^{K} \theta_{dk} \phi_{k\ell}^i$, where N_d is the number of words in d. We find that TI benefits from the transformation matrices, and tends to have more stable accuracies when the number of topic is small. We thus set the number of topics to be 5. Moreover, we observed fast convergence of the testing accuracy for TI (usually within 50 iterations), thus we reported the results obtained between 50 and 200 iterations. Other hyperparameters were set as in previous experiments. For SVM, we represented each document as a *td-idf* vector [50] and used the *libSVM* implementation [51] with linear and RBF kernels, where we did a 5-fold cross validation to select the optimal parameters using the provided function. Finally, we also compared our model with the SVM with features learned from LDA. We followed [6] in partitioning the dataset into training and testing sets for the *Blog* dataset. For the other two, we randomly took 80% of the whole dataset for training and the rest for testing. Table 2 shows the results of classification accuracy. The result for SVM_L is comparable to that in [6] where they report obtaining 69.6% with SAGE. We can see from the results that TI significantly outperforms SVM and SAGE, demonstrating the differential ability of our model. On the other hand, TII with identity transformation matrices fails to compete with SVM in most cases. Furthermore, we observed worse performance of the SVM with LDA features than the simple SVM with sparse *tf-idf* features, indicating the simple LDA model might not be a good one for classification tasks.

TABLE 2: Classification accuracies on the three datasets. The second row in the "TI" entry represents the highest accuracies obtained during the runs. SVM_L means SVM classifier with linear kernel, while SVM_R means SVM with RBF kernel. LDA+SVM means SVM classifier with LDA features.

Datasets	BD	Blog	MLJ
TI	$\mathbf{81.58}\% \pm 0.9\%$ (84.48%)	$\mathbf{73.54}\% \pm 0.8\%$ (75.17%)	$\mathbf{80.98}\% \pm 0.8\%$ (83.00)%
TII	$45.03\% \pm 1.2\%$ (47.18%)	$71.52\% \pm 2.1\%$ (75.80%)	$42.40\% \pm 3.0\%$ (50.01)%
SVM_L	78.35%	69.59%	71.92%
SVM_R	78.35%	70.40%	71.91%
LDA + SVM	65.13%	70.63%	69.27%

7 Conclusion

We developed a hierarchical topic model for differential analysis to be applied to comparable data collections as a means to understand similarities and differences. The Poisson Dirichlet Process (PYP) was used to manage a hierarchy of topics across collections, rather than using the "shared and distinct" word vectors of earlier work. The variance parameters of the PYP then can control the level of sharing across collections and also allow unpaired topics. Moreover, we proposed the Transformed PYP (TPYP), a type

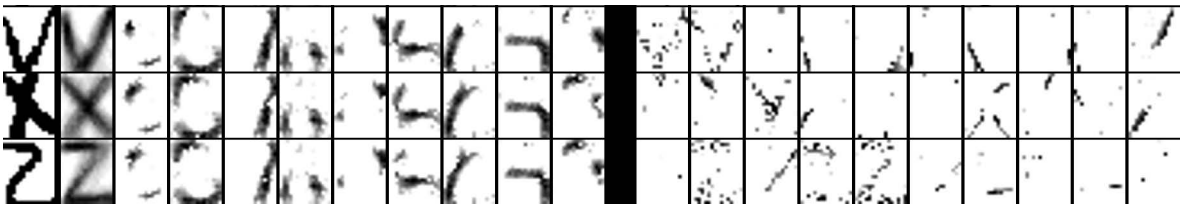


Fig. 9: 10 topics for the three groups "V", "X" and "Z" from TII (left) and LDA (right). The first column contains random samples for three groups, the others are the corresponding 10 topics. The second column of TII topics reveals different structures among the characters while the other columns represent shared structures.

of PYP with transformed based measures, and developed an efficient inference algorithm to deal with the non-conjugacy of the model using an auxiliary variable trick and a table indicator representation for the hierarchical PYP.

Experimental results on both text and images show significant improvement compared to existing algorithms in terms of test perplexity, and illustrative examples demonstrate the application. Finally, we have show our model outperforms the state-of-the-art for some document classification tasks.

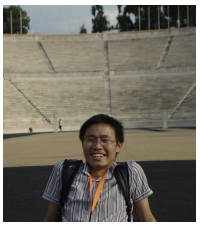
ACKNOWLEDGMENT

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Center of Excellence program. Lan Du was supported under Australian Research Council's Discovery Projects funding scheme (DP110102506 and DP110102593).

REFERENCES

[1] C. Zhai, A. Velivelli, and B. Yu, "A cross-collection mixture model for comparative text mining," in *SIGKDD*, 2004.
[2] M. Paul, "Cross-collection topic models: Automatically comparing and contrasting text," Master's thesis, Univ. of Illinois at Urbana-Champaign, 2009.
[3] A. Ahmed and E. Xing, "Staying informed: Supervised and semi-supervised multi-view topical analysis of ideological perspective," in *EMNLP*, 2010.
[4] W.-H. Lin, T. Wilson, J. Wiebe, and A. Hauptmann, "Which side are you on?: identifying perspectives at the document and sentence levels," in *CoNLL*, 2006.
[5] T. Hofmann, "Probabilistic latent semantic indexing," in *SIGR*, 1999.
[6] J. Eisenstein, A. Ahmed, and E. Xing, "Sparse additive generative models of text," in *ICML*, 2011.
[7] M. Paul and R. Girju, "Cross-cultural analysis of blogs and forums with mixed-collection topic models," in *EMNLP*, 2009.
[8] ——, "A two-dimensional topic-aspect model for discovering multi-faceted topics," in *AAAI*, 2010.
[9] M. Paul, C. Zhai, and R. Girju, "Summarizing contrastive viewpoints in opinionated text," in *EMNLP*, 2010.
[10] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
[11] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet processes," *Journal of the ASA*, vol. 101, no. 476, pp. 1566–1581, 2006.
[12] W. Li and A. Mccallum, "Pachinko allocation: DAG-structured mixture models of topic correlations," in *ICML*, 2006.
[13] D. Andrzejewski, X. Zhu, and M. Craven, "Incorporating domain knowledge into topic modeling via Dirichlet Forest priors," in *ICML*, 2009.
[14] D. M. Blei, T. L. Griffiths, and M. I. Jordan, "The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies," *J. ACM*, vol. 57, no. 2, pp. 1–30, 2010.
[15] S. Goldwater, T. Griffiths, and M. Johnson, "Producing power-law distributions and damping word frequencies with two-stage language models," *J. Mach. Learn. Res.*, vol. 12, pp. 2335–2382, 2011.
[16] I. Sato and H. Nakagawa, "Topic models with power-law using Pitman-Yor process," in *SIGKDD*, 2010.
[17] D. M. Blei and J. D. Lafferty, "A correlated topic model of science," *Ann. Appl. Stat.*, vol. 1, no. 1, pp. 17–35, 2007.
[18] D. Mimno, H. Wallach, and A. McCallum, "Gibbs sampling for logistic normal topic models with graph-based priors," NIPS Workshop on Analyzing Graphs, Tech. Rep., 2008.
[19] J. Paisley, C. Wang, and D. Blei, "The discrete infinite logistic normal distribution," *Bayesian Analysis*, vol. 7, no. 2, pp. 235–272, 2012.
[20] L. Du, W. Buntine, H. Jin, and C. Chen, "Sequential latent Dirichlet allocation," *KAIS*, vol. 31, no. 3, pp. 475–503, 2012.
[21] L. Du, W. Buntine, and H. Jin, "Modelling sequential text with an adaptive topic model," in *EMNLP*, 2012.
[22] C. Wang and D. M. Blei, "Decoupling sparsity and smoothness in the discrete hierarchical Dirichlet process," in *NIPS*, 2009.
[23] C. Wang, B. Thiesson, C. Meek, and D. M. Blei, "Markov topic models," in *AISTATS*, 2009.
[24] J. Petterson, A. Smola, T. Caetano, W. Buntine, and S. Narayanamurthy, "Word features for latent Dirichlet allocation," in *NIPS*, 2010.
[25] D. Newman, E. Bonilla, and W. Buntine, "Improving topic coherence with regularized topic models," in *NIPS*, 2011.
[26] M. Paul and M. Dredze, "Factorial LDA: Sparse multi-dimensional text models," in *NIPS*, 2012.
[27] L. Wan, L. Zhu, and R. Fergus, "A hybrid neural network-latent topic model," in *AISTATS*, 2012.
[28] T. Ferguson, "A Bayesian analysis of some nonparametric problems," *The Anna. of Stat.*, vol. 1, pp. 209–230, 1973.
[29] Y. W. Teh and M. I. Jordan, *Hierarchical Bayesian Nonparametric Models with Applications*. Cambridge, UK: Cambridge University Press, 2010.

[30] Y. W. Teh, "A hierarchical Bayesian language model based on Pitman-Yor processes," in *ACL*, 2006.
[31] ——, "A Bayesian interpretation of interpolated Kneser-Ney," National University of Singapore, Tech. Rep. TRA2/06, 2006.
[32] P. Orbanz and J. M. Buhmann, "Nonparametric Bayesian image segmentation," *IJCV*, vol. 77, pp. 25–45, 2007.
[33] L. Du, L. Ren, D. B. Dunson, and L. Carin, "A Bayesian model for simultaneous image clustering, annotation and object segmentation," in *NIPS*, 2009.
[34] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky, "Describing visual scenes using transformed Dirichlet processes," in *NIPS*, 2005.
[35] F. Wood, C. Archambeau, J. Gasthaus, L. James, and Y. Teh, "A stochastic memoizer for sequence data," in *ICML*, 2009.
[36] Z. Xu, V. Tresp, K. Yu, and H. P. Kriegel, "Infinite hidden relational models," in *UAI*, 2006.
[37] H. Ishwaran and L. F. James, "Gibbs sampling methods for stick-breaking priors," *Journal of ASA*, vol. 96, no. 453, pp. 161–173, 2001.
[38] D. Aldous, "Exchangeability and related topics," *École d'Été de Probabilités de Saint-Flour XIII*, pp. 1–198, 1983.
[39] B. A. Frigyik, M. R. Gupta, and Y. Chen, "Shadow Dirichlet for restricted probability modeling," in *NIPS*, 2010.
[40] C. Chen, L. Du, and W. Buntine, "Sampling table configurations for the hierarchical Poisson-Dirichlet process," in *ECML*, 2011.
[41] C. Robert and G. Casella, *Monte Carlo statistical methods*. Springer, 2004, second edition.
[42] W. Buntine and M. Hutter, "A Bayesian view of the Poisson-Dirichlet process," NICTA and ANU, Australia, Tech. Rep. arXiv:1007.0296, 2012.
[43] T. P. Minka, "Estimating a Dirichlet distribution," MIT, Tech. Rep., 2000.
[44] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, pp. 183–233, 1999.
[45] T. Yano, W. Cohen, and N. Smith, "Predicting response to political blog posts with topic models," in *North American ACL-HLT*, 2009.
[46] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *UAI*, 2004.
[47] L. Du, W. Buntine, and H. Jin, "A segmented topic model based on the two-parameter Poisson-Dirichlet process," *Machine Learning*, vol. 81, pp. 5–19, 2010.
[48] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.
[49] C. Cortes and V. N. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
[50] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.
[51] C. C. Chang and C. J. Lin, "Libsvm – a library for support vector machines," National Taiwan University, Tech. Rep., 2013. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm/



Changyou Chen received his B.S. and M.S. degree in 2007 and 2010 respectively, both from School of Computer Science, Fudan University, Shanghai, China. He is now a PhD candidate at the College of Engineering and Computer Science, the Australian National University supervised by Dr. Wray Buntine. He is in the Machine Learning Research Group at NICTA in Canberra. His current research interests include Dependent Normalized Random Measures, Dependent Poisson-Kingman Processes, MCMC for Hierarchical Pitman-Yor Processes and Topic Models.

Wray Buntine joined NICTA in Canberra Australia in April 2007. He was previously at the Helsinki Institute for Information Technology from 2002. He is known for his theoretical and applied work in document and text analysis, data mining and machine learning, and probabilistic methods. He is currently Principle Researcher working on applying probabilistic and non-parametric methods to tasks such as text analysis. In 2009 he was programme co-chair of ECML-PKDD in Bled, Slovenia, and was programme co-chair of ACML in Singapore in 2012. He reviews for conferences such as ACML, ECIR, SIGIR, ECML-PKDD, ICML, UAI, and KDD.

Nan Ding is now a software engineer at Google Inc., USA. He completed his Ph.D. from the Department of Computer Science, Purdue University in May 2013. He received his B.E. Degree from the Department of Electronic Engineering, Tsinghua University in July 2008, and M.S. Degree from the Department of Computer Science, Purdue University in December 2010. His research is in graphical models, nonparametric Bayesian, approximate inference, convex analysis and optimization.

Lexing Xie is Senior Lecturer in Computer Science at the Australian National University. She received the B.S. degree from Tsinghua University, Beijing, China, in 2000, and the M.S. and Ph.D. degrees in Electrical Engineering from Columbia University, in 2002 and 2005, respectively. She was with the IBM T.J. Watson Research Center, Hawthorne, NY from 2005 to 2010. Her recent research interests are in multimedia, machine learning and social media analysis. Dr. Xie's research has won six conference paper awards. She plays an active role in editorial and organizational roles in the multimedia community.

Lan Du is a Research Fellow in the Department of Computing at Macquarie University, Sydney. He received the B.Sc. degree from the Flinders University of South Australia, in 2006, and the first class honours degree in Information Technology and Ph.D. degree in Computer Science in 2007 and 2012 respectively. Both are from the Australian National University. He was also affiliated with the Machine Learning Resarch Group in NICTA at Canberra from 2008 to 2011. His research interests are in statistical modelling and learning for document and text analysis.