ABSTRACT
One of the important approaches for Knowledge discovery and Data mining is to estimate unobserved variables because latent variables can indicate hidden and specific properties of observed data. The latent factor model assumes that each item in a record has a latent factor; the co-occurrence of items can then be modeled by latent factors. In document modeling, a record indicates a document represented as a “bag of words”, meaning that the order of words is ignored and an item indicates a word. Latent Dirichlet allocation (LDA) has stimulated the use of the Dirichlet distribution over the latent topic distribution of a document. LDA assumes that latent topics, i.e., discrete latent variables, are distributed according to a multinomial distribution whose parameters are generated from the Dirichlet distribution. In an experiment using real data, this model outperformed LDA in document modeling.

Keywords
Topic Model, Latent Dirichlet Allocation, Nonparametric Bayes, Pitman-Yor Process, Power-law

1. INTRODUCTION
Probabilistic models with latent variables have attracted attention in Knowledge discovery and Data mining because of their power and flexibility to model real world phenomena. In this approach, it is necessary to estimate unobserved, namely latent variables. The latent factor model assumes that each item in a record has a latent factor and that the co-occurrence of items can then be modeled by latent factors. The goals of a probabilistic modeling are to find shot descriptions that preserve the statistical relationships of data and predict new occurrences.

In document modeling, a record indicates a document represented as a “bag of words”, meaning that the order of words is ignored and an item indicates a word. The latent factor in fact stands for the topic. Probabilistic latent semantic indexing (PLSI)[1] was one of the first topic models. PLSI has a problem in that it cannot treat new document data that does not coincide with any of the training data. Latent Dirichlet allocation (LDA)[2] generalizes PLSI by applying a Bayesian framework that can avoid over-fitting and can treat new documents on the basis of a prior distribution. LDA assumes that latent topics, i.e., discrete latent variables, are distributed according to a multinomial distribution whose parameters are generated from the Dirichlet distribution. PLSI and LDA model a document’s property that a document has multiple topics. LDA has stimulated the use of the Dirichlet distribution over the latent topic distribution of a document and inspired many other topic models such as LDA-HMM [3], author-topic model [4, 5], entity-topic models [6], correlated topic model [7], hidden topic Markov models [8], dynamic topic model [9, 10], topic models for text and citations [11], topic model for visualization [12], topic models for Hypertext [13], topic models conditioned on arbitrary features [14], syntactic topic models [15], and so on.

LDA also models a word distribution by using the multinomial distribution and parameters of the multinomial distributions follows the Dirichlet distribution. These Dirichlet-multinomial settings cannot capture the power-law phenomenon of a word distribution, which is known as “Zipf’s law” in linguistics. The Power-law distributions are produced by a stochastic process in which frequent outcomes attract probability mass such as “rich-get-richer” process. A major example of a power-law distribution is a distribution of links pointing to web pages. New web pages are more likely to link to already-popular pages that have already a lot of links. A widely used process is a preferential attachment process[1]. One of the statistical properties of natural language is that word frequencies follow a power-law distribution given by

\[ p(n^w = x) \propto x^{-l}, \]

where \( n^w \) is the number of frequencies of words and \( l \) is some constant parameter. This observation is often called Zipf’s law. Fig.1 (a) shows the empirical probability of words in Reuters corpus. The plots appear approximately linear on a log-log plot. This behavior is characteristic of a power-law distribution. Fig.1 (c) indicates that the power-law property of a word distribution is also observed in a document. The Pitman-Yor (PY) process[2] is one of the most adaptive processes for a document modeling due to its exchange-ability property.

In this paper, we develop a topic model based on the hierarchical Pitman-Yor (HPY) process for modeling a word distribution. The PY process is a stochastic process generalized from the Dirichlet process[16]. The PY process has a concentration parameter \( \gamma \) and a discount parameter \( d \) that control the power-law property. If discount parameter is set to zero, the PY process has the same property as the Dirichlet process. The a discount parameter place emphasis on a new-word generation that induces a long-tail phenomena of a
distribution, which is useful for modeling word-frequency distributions that tend to have many frequency-1 words. The PY process that has a stochastic process called the Chinese restaurant process that is a process of customers’ seating arrangement in a restaurant where the number of customers seated at tables follows a power-law distribution. In this case, the power-law parameter \( l \) in Eq.(1) is equal to 1+d. Because of the power-law property, the Pitman-Yor process applies well natural language processing, in cases morphological structure analysis of words[17], N-gram language modeling [18, 19], a dependency parsing[20], and so forth.

We assume a power-law word distribution not only in a document but also in a topic. A topic in LDA is represented by a word distribution. A word distribution in a specific topic, e.g. search engines, can give high probability to the word specific to the topic, e.g., “Google ⊗” “Yahoo ⊗” and “MSN ⊗”. Like many phenomena of linguistics, a power-law property can be observed in a topic-word distribution. For example, Reuters corpus has labels indicating their document’s topic and each document has multiple labels. Fig.1(c) illustrates the empirical probability of words in documents with label “trade”. Fig.1(c) also provides a power-law property. We models this property by using the hierarchical Pitman-Yor (HPY) process[18, 19].

**Contribution and Remainder.**

We propose a novel topic model using the PY process, called the PY topic model. The PY topic model captures two properties of a document, a power-law word distribution and multiple topics.

The remainder of this paper is organized as follows. Section 2 overviews LDA. Section 3 describes the PY process. Section 4 proposes the PY topic models. Section 5 presents experimental results. Section 6 analyzes extracted latent topics. Section 7 summarizes the paper.

**2. LDA**

In this section, we overview LDA where documents are represented as random mixtures over latent topics and each topic is characterized by a distribution over words. First, we define notation. \( T \) is the number of topics. \( M \) is the number of documents. \( V \) is the number of vocabulary size. \( N_j \) is the number of words in document \( j \). \( w_{j,i} \) denotes the \( i \)-th word in document \( j \). \( z_{j,i} \) denotes the latent topic of word \( w_{j,i} \). \( \text{Multi}() \) is a multinomial distribution. \( \text{Dir}() \) is a Dirichlet distribution. \( \theta_j \) denotes a \( T \)-dimensional probability vector that is the parameters of the multinomial distribution, and represents the topic distribution of document \( j \). \( \phi_v \) is a \( V \) dimensional probability vector where \( \phi_{v,j} \) specifies the probability of generating word \( v \) given topic \( t \). \( \alpha_t \) is the \( T \)-dimensional parameter vector of the Dirichlet distribution over \( \theta_j \) (\( j = 1, \cdots, M \)). \( \beta \) is a parameter of th Dirichlet over \( \phi_t \) (\( t = 1, \cdots, T \)).

LDA assumes the generative process shown in Algorithm ???. The graphical model of LDA is shown in Fig. 2 (a).

The Gibbs sampler is applied given by

\[
p(z_{j,i} = k|w_{j,i} = v, Z^{-j,i}, W^{-j,i}) = \frac{N_{j,k}^{-j,i} + \alpha_k}{N_j^{-j,i} + \alpha_0} \frac{N_{j,k+1}^{-j,i} + \beta}{N_{j+1}^{-j,i} + \alpha_0 + V \beta}\]  

(2)

where \( \alpha_0 = \sum_k \alpha_k \). \( Z \) denotes a set of all latent topic variables, \( Z^{-j,i} = Z \setminus \{z_{j,i}\} \). \( W \) denotes a set of all words, \( W^{-j,i} = \).

**3. PITMAN-YOR PROCESS**

In this section, we explain the Pitman-Yor (PY) process [22, 18, 19] by modeling a document. The PY document model captures the power-law property of the word distribution.
Algorithm 1 Generative process of LDA
1: Draw $\phi_1 \sim \text{Dir}(\phi|\beta)$ ($t = 1, \ldots, T$),
2: for all document $j = 1, \ldots, M$ do
3: Draw $\theta_j \sim \text{Dir}(\theta|\alpha)$,
4: for all word $i = 1, \ldots, N_j$ do
5: Draw topic $z_{j,i} \sim \text{Mult}(z|\theta_j)$,
6: Draw word $w_{j,i} \sim p(w|z_{j,i}, \phi_t)$,
7: end for
8: end for
9: where $\text{Dir}(\theta|\alpha) \propto \prod \theta^\alpha_k - 1$,
$p(w = v|z = t, \phi) = \phi_{t,v}$.

The PY process PY($\gamma, d, G_0$) is a distribution over distributions over a probability space. The PY process has three parameters, a concentration parameter $\gamma$, a discount parameter $d(0 \leq d \leq 1)$ that controls the power-law property of distribution and a base distribution $G_0$ that is understood as a mean of draws from PY($\gamma, d, G_0$). The PY process is a generalization of the Dirichlet process where the discount parameter is regarded as zero in the Dirichlet process. The PY document model has a perspective given by the Chinese restaurant process (CRP) [23]. We consider two kinds of distribution for a document collection: let $G_0(w)$ be a general word distribution, i.e., the base distribution of the whole set of back-off document collections, and $G_j(w)$ be a document-specific word distribution for document $j$.

The generation process for the PY document model is

$$ G_j \sim \text{PY}(\gamma, d, G_0), \quad w_{j,i} \sim G_j. \quad (5) $$

We now provide details on a CRP representation for the PY document model. A CRP representation is composed of four elements, a customer, a table, a dish, and a restaurant. A customer denotes a word in a document, a table a latent variable, and a dish a word type. A restaurant denotes a document. Let $w_{j,1}, w_{j,2}, \ldots$ be a sequence of identical, independent draws from $G_j$, i.e., $\{w_{j,i}\}$ denotes words in document $j$. The sequence, $\{w_{j,i}\}$, represents customers visiting restaurant $j$ corresponding to $G_j$ with an unbounded number of tables. $\{x_{j,i}\}$ denotes seating arrangements of customers. $x_{j,i} = k$ indicates that the $i$-th customer sits in the $k$-th table. $v_{j,k} = v$ denotes that word type $v$ is served at the $k$-th table in restaurant $j$. Namely, if $x_{j,i} = k$ and $v_{j,k} = v$, then $w_{j,i} = v$ that means the $i$-th word in document $j$ is word type $v$. For example, $w_{j,i} = \text{the}^*$ ($x_{j,i} = k$ and $v_{j,k} = \text{the}^*$) indicates the $i$-th customer visiting a restaurant $j$ is eating dish (word) "the". Fig. 3 explains an example of the CRP representation. Note that, the HY document model assumes that a document is represented as a "bag of words", meaning that the order of words is ignored and an item indicates a word.

The CRP assigns a distribution over the seating arrangement of the customers. The sequence generated with CRP can be shown to be exchangeable [23]. When the $i$-th customer $x_{j,i}$ enters restaurant $j$ with $K_j$ occupied tables at which other customers ($x_{j,1}, \ldots, x_{j,i-1}$) have already been seated, the new customer sits at a table under two conditions:

$$ \begin{cases} 
\text{The } k\text{-th occupied table with probability } & \frac{N_{j,k}^* - d}{\gamma + N_{j,k}^*}, \\
\text{A new unoccupied table with probability } & \frac{\gamma + dK_j}{\gamma + N_{j,k}^*}. 
\end{cases} \quad (6) $$

Here, $N_{j,k}^*$ denotes the number of customers sitting at the $k$-th table and $N_{j,v}^* = \sum_k N_{j,k}^*$ indicates the document length $N_j$, $K_j$ denotes the total number of tables in restaurant $j$. If a customer sits at a new table, word $v^{\text{new}}$ is drawn from the base distribution $G_0(v)$ and served at the new table. This means that $w_{j,i}$ is given value $v^{\text{new}}$, which is a term in the document, i.e., this indicates that the $i$-th word in document $j$ is term $v^{\text{new}}$. If the customer sits at the $k$-th table, $w_{j,i}$ is given value $v_k$, which is the word served at the table. If $d$ is not zero, the number of tables increases as many customers enter the restaurant, and this leads to a power-law phenomenon.

The predictive probability of a new word, given the seating arrangement is

$$ p(w_{j,i+1} = v|\{w_{j,i}\}, \{x_{j,i}\}) = \frac{N_{j,v} - dK_{j,v}}{\gamma + N_j} + \frac{\gamma + dK_j}{\gamma + N_{j,v}} G_0(v), \quad (7) $$

where $N_{j,v}$ denotes the number of customers serving word $v$ that indicates the frequency of word $v$ in document $j$, and $K_{j,v}$ denotes the number of tables serving word $v$ in restaurant $j$.

The discount parameter, $d$, and the number of table, $K_{j,v}$, effect smoothing. In a prediction of a word, frequent words such as "the" and "a" often hurt the performance. However, their frequency is discounted by $dK_{j,v}$ in Eq.(7). The discount parameter, which places more emphasis on new-word (new-table in the CRP) generation than $d = 0$, is useful for modeling word frequency distributions that tend to have many frequency-1 words.

4. PROPOSED MODEL

The basic idea of our model is that a word distribution is generated from the PY process. First, we propose the PY topic model and then, the HPY topic model that is more general model.

4.1 Pitman-yor topic model

The difference between LDA and the PY topic model is how to generate a topic in a document. Although LDA generates a topic in each word in a document, We assume that the PY topic model generates a topic in each table in CRP representation for a document. That is, the number of generated topics is equal to that of words in LDA and that of tables in the PY topic model. Therefore, we introduce latent variable $z_{j,k} = t$ which denotes that topic $t$ is assigned in the $k$-th table in document $j$. Like the PY document model, a
customer sits in a table following Eq.(6) in the PY topic model. The PY topic model assumes the generative process shown in algorithm 2 from an analogy to the PY document model. The graphical model of the PY topic model is shown in Fig. 2 (b). The inference for seating arrangements is given by Algorithm 3 where AddCustomer and RemoveCustomer are described in the Appendix.

Algorithm 2 Generative process of PYTM

1: Draw \( \phi_t \sim Dir(\phi|\beta) \) (\( t = 1, \cdots, T \)).
2: for all document \( j(= 1, \cdots, M) \) do
3: Draw \( \theta_j \sim Dir(\theta|\alpha) \).
4: for all word \( i(= 1, \cdots, N_j) \) do
5: Sit in the \( k \)-th occupied table with proportion to \( N_j^c \).
6: Sit in a new unoccupied table with proportion to \( N_j^v + d_K \).
7: end for
8: end for

Algorithm 3 Inference process of HPYTM

1: for iteations do
2: for all document \( j(= 1, \cdots, M) \) do
3: for all word \( i(= 1, \cdots, N_j) \) do
4: RemoveCustomer(\( w_{j,i} \), document \( j \)).
5: AddCustomer(\( w_{j,i} \), document \( j \)).
6: end for
7: end for
8: Estimate \( \alpha \) by using Eq.(3).
9: end for

The predictive probability of a new, given words, topics and the seating arrangements in documents is

\[
p(w^\text{new} = v| W, Z, X) = \frac{N_{j,v} - d_K + \alpha_j}{\gamma + N_j} + \frac{\gamma + d_K}{\gamma + N_j} \sum_{t=1}^{T} N_{j,t} + \alpha_0 \frac{N_{j,v} + \beta}{N_j + \alpha_0 + V \beta }.
\]

The probability of a topic generating a new Table is given by

\[
p(z_{j,k^\text{new}} = t|x_{j,i} = k^{\text{new}}, Z, W^{j-1}, X^{-j,i}) = \frac{N_{j,t} + \alpha_k}{K_j + \alpha_0} p_t(w_{j,i}).
\]

4.2 Hierarchical Pitman-Yor topic model

We propose a more general model, the hierarchical Pitman-Yor (HPY) topic model that assumes a power-law word distribution not only in a document but also in a topic. The HPT topic model models a power-law property of a topic-specific word distribution by using the hierarchical Pitman-Yor (HPY) process. We replace the generation process of \( \{ \phi_t \} \) in the PY topic model (Algorithm 2 step 2) as follows.

\[
\phi_t \sim PY(\gamma_1, d_1, \phi_0) \quad (t = 1, \cdots, T), \quad \phi_0 \sim PY(\gamma_0, d_0, U),
\]

where \( \phi_0 \) is a base word distribution in whole corpus, \( U \) denotes a uniform distribution in which the probability of all words is assigned according to the size of the vocabulary \( V \), i.e., \( U(v) = 1/V \). The relationship of \( \phi_0 \) and \( \phi_t \) just looks like that of (a) and (b) in Fig.1. Like a word distributions of each document, \( \phi_t(t = 0, 1, \cdots, T) \) can be also represented as the CRP.

The predictive probability of a new word, given words, topics and the seating arrangements in documents is recursively given by

\[
p(w^\text{new} = v| W, Z, X) = \frac{N_{j,v} - d_K + \alpha_j}{\gamma + N_j} + \frac{\gamma + d_K}{\gamma + N_j} \sum_{t=1}^{T} N_{j,t} + \alpha_0 \frac{N_{j,v} + \beta}{N_j + \alpha_0 + V \beta }.
\]

The probability of a topic generating a new Table is given by

\[
p(z_{j,k^\text{new}} = t|x_{j,i} = k^{\text{new}}, Z, W, X^{-j,i}) = \frac{N_{j,t} + \alpha_k}{K_j + \alpha_0} p_t(w_{j,i}).
\]

Note that \( \hat{N}_{j,t} \) indicates the number of tables in which a seated word is generated from topic \( t \).

5. EXPERIMENTS

6. KNOWLEDGE DISCOVERY

7. CONCLUSION

Acknowledgements

This research was funded in part by a MEXT Grant-in-Aid for Scientific Research on Priority Areas “i-explosion” in Japan.

8. REFERENCES


Black dots indicate customers. A customer sits in a table proportional to the number of customers that have already sat. A customer also can sit in a new table in some probability. For example, a customer sits in a table serving word “the” proportional to $\frac{2-d}{\alpha+6}$ and a new table proportional to $\frac{2-d}{\alpha+6}$. If a customer sits in a new table, a word is generated from a base distribution $G_0$ and served in the table. Multiple tables can serve a word type that has already served in other tables, e.g. a new table can serve word “the”. Since frequent words tend to have many tables, the total frequency is discounted corresponding to the number of table and parameter $d$.

A customer sits in a table proportional to the number of customers that have already sat as well as the PY document model. If a customer sits in a new table, a topic is generated from the topic distribution as well as LDA and a word is served in the table from a topic-specific word distribution. The PY topic model has properties of a power-law and multiple topics.


Algorithm 4 Function WordProbability(word t, topic i)
1: if PY topic model
2: \[ N_{i,v} + \beta \]
3: else if HPY topic model
4: \[ N_{t,v} + V \beta \]
5: \[ \text{WordProbability}(\text{word } v, \text{topic } i) \]
6: \[ \text{add a customer at the } k\text{-th table serving word } v \text{ in topic } j \]
7: \[ \text{add a customer at a new table and call AddCustomer(word } v, \text{topic } i) \]
8: \[ \text{end if} \]
9: \[ \text{end if} \]

Algorithm 5 Function AddCustomer(word v, document j)
1: Draw topic t from for a new table using Eq.(9) in PYTM and Eq.(15) in HPYTM, and set \( z_{j,k,v} = t \).
2: If the \( t\)-th table serving word \( v \) becomes unoccupied, remove a customer from the \( k\)-th table to the new table.
3: If the \( k\)-th table is created from the \( j\)-th document as \( N_{j,v} = 0 \), remove a customer from the \( k\)-th table to the new table.
4: If the \( k\)-th table serving word \( v \) is occupied, remove the table from document \( j \) and call RemoveCustomer(word \( v \), topic \( i \)) if \( v \) in topic \( i \).
5: \[ \text{end if} \]
6: \[ \text{end if} \]

Algorithm 6 Function AddCustomer(word v, topic i)
1: if PY topic model
2: \[ \text{add a customer at the } k\text{-th table serving word } v \text{ in topic } j \]
3: else if HPY topic model
4: \[ \text{add a customer at a new table and call AddCustomer(word } v, \text{topic } i) \]
5: \[ \text{end if} \]
6: \[ \text{end if} \]

Algorithm 7 Function RemoveCustomer(word v, document j)
1: \[ \text{remove a customer from the } k\text{-th table serving word } v \text{ in document } j \]
2: \[ \text{remove the table from document } j \text{ and call RemoveCustomer(word } v, \text{topic } i) \]
3: \[ \text{end if} \]
4: \[ \text{end if} \]

Algorithm 8 Function RemoveCustomer(word v, topic i)
1: \[ \text{remove a customer from the } k\text{-th table serving word } v \text{ in document } j \]
2: \[ \text{remove the table from document } j \text{ and call RemoveCustomer(word } v, \text{topic } i) \]
3: \[ \text{end if} \]
4: \[ \text{end if} \]

A word distribution of each document, topic, whole corpus can be regarded as a restaurant in the CRP representation of our models. The seating arrangements of customers in a restaurant are sampled by running the two function alternately, AddCustomer and RemoveCustomer. AddCustomer adds the \( i\)-th customer into restaurant \( j \) shown in Algorithm 5 and 6. RemoveCustomer moves a customer using menu \( t \) from restaurant \( v \) shown in Algorithm 7 and 8. \( N_{j,v,k} \) and \( N^c_{j,v,k} \) indicate the number of customers at the \( k\)-th table serving word type \( v \) in document \( j \) and topic \( t \), respectively. WordProbability(word \( v \), topic \( i \)) indicates the probability that word \( v \) is observed in topic \( t \). Note that \( N_{j,t} \) in our models indicates the number of tables in which a seated word is generated from topic \( t \) in document \( j \), not the number of words generated from topic \( t \) in document \( j \).

The parameters \( \alpha \) and \( d \) of the PYPY topic model can be estimated by auxiliary variable sampling [18, 19, 24]. Those of the HPY topic model are estimated in a similar way.

First, auxiliary variables \( x_j, y_{jk}, \) and \( z_{jk,v} \) are sampled for each

APPENDIX

A. GIBBS SAMPLER FOR PROPOSED MODELS


document restaurant $j = 1, \cdots, M$.

\begin{align}
    x_j & \sim \text{Beta}(\tilde{\alpha} + 1, N_j - 1) \ (j = 0, 1, 2, \cdots, M), \quad (16) \\
    y_{jk} & \sim \text{Bern}(\frac{\tilde{\alpha}}{\tilde{\alpha} + dk}) \ (k = 1, 2, \cdots, K_j - 1), \quad (17) \\
    z_{jvki} & \sim \text{Bern}(\frac{i - 1}{i - d}) \ (i = 1, 2, \cdots, n_{jvki} - 1), \quad (18)
\end{align}

Next, given the auxiliary variables, the parameters are sampled.

\begin{align}
    d & \sim \text{Beta}(\tilde{a}_d, \tilde{b}_d), \quad (19) \\
    \alpha & \sim \text{Gamma}(a_\alpha + \sum_{j=1}^{M} \sum_{k=1}^{t_j} y_{jk}, b_\alpha - \sum_{j=1}^{M} \log x_j), \quad (20) \\
    \tilde{a}_d & = a_d + \sum_{j=1}^{M} \sum_{k=1}^{t_j - 1} (1 - y_{jk}), \quad (21) \\
    \tilde{b}_d & = b_d + \sum_{j=1}^{M} \sum_{v,k} n_{jvki} \geq 2 \sum_{i=1}^{n_{jvki} - 1} (1 - z_{jvki}), \quad (22)
\end{align}

where $a_d, b_d, a_\alpha$, and $b_\alpha$ are hyper parameters. We set all hyper parameters to 1.