D-LDA: A Topic Modeling Approach without Constraint Generation for Semi-Defined Classification

Fuzhen Zhuang[1,2], Ping Luo[3], Zhiyong Shen[3], Qing He[1], Yuhong Xiong[4], Zhongzhi Shi[1]
[1]The Key Laboratory of Intelligent Information Processing,
Institute of Computing Technology, CAS, {zhuangfz, heq, shizz}@ics.ict.ac.cn
[2] Graduate University of Chinese Academy of Sciences
[3] Hewlett Packard Labs China, {ping.luo, zhiyongs}@hp.com
[4] Innovation Works, yhxiong@yahoo.com

***Abstract*—We study what we call *semi-defined classification*, which deals with the categorization tasks where the taxonomy of the data is not well defined in advance. It is motivated by the real-world applications, where the unlabeled data may also come from some other *unknown classes* besides the *known classes* for the labeled data. Given the unlabeled data, our goal is to not only identify the instances belonging to the known classes, but also cluster the remaining data into other meaningful groups. It differs from traditional semi-supervised clustering in the sense that in semi-supervised clustering the supervision knowledge is too far from being representative of a target classification, while in semi-defined classification the labeled data may be enough to supervise the learning on the known classes.**

In this paper we propose the model of Double-latent-layered LDA (D-LDA for short) for this problem. Compared with LDA with only one latent variable y for word topics, D-LDA contains another latent variable z for (known and unknown) document classes. With this double latent layers consisting of y and z and the dependency between them, D-LDA directly injects the class labels into z to supervise the exploiting of word topics in y. Thus, the semi-supervised learning in D-LDA does not need the generation of pairwise constraints, which is required in most of the previous semi-supervised clustering approaches. We present the experimental results on ten different data sets for semi-defined classification. Our results are either comparable to (on one data sets), or significantly better (on the other nine data set) than the six compared methods, including the state-of-the-art semi-supervised clustering methods.

***Keywords*-Semi-defined classification, Topic modeling, Gibbs Sampling, Semi-supervised clustering;**

I. Introduction

Real-world classification tasks often encounter the problem that the taxonomy of the data cannot be well defined at the beginning. Given a group of unlabeled data, the domain experts may not know all the data classes. Usually, they are only familiar with a subset of all the data classes, and also agree that the unlabeled data may stem from some other meaningful clusters. For example, to build a news portal for one of the Fortune 500 companies we want to classify the everyday news about this company into some classes. It is easy to list some corporation news classes, e.g. "*product-related*", "*financial report, business and industry analysis*", "*stock review*", "*merger and acquisition related*" and so on. However, we cannot exhaustively list all the news classes which are interesting to news readers. Actually, after some deep investigation we know that the news about a big company may include some other classes, such as "*business expansion and new investment*", "*partnership and alliance with other companies*", "*charity, donation and citizenship*" etc. Another example is to recognize the relationships between employees and products in the enterprize internal Web pages. This task can be converted as a classification problem for any employee-product pair, however, neither we do not know all types of employee-product relationships which exist in the corpus.

In these two examples we are given the labeled instances from certain pre-defined classes and the unlabeled data, and aim to not only identify the instances from the known classes but also exploit new meaningful data clusters. Thus, it is a combination of supervised classification and unsupervised clustering. It is worth mentioning that semi-defined classification can be applied in an iterative manner in order to build the full version of the data taxonomy. Specifically, in each iteration round the model outputs not only the instances in the pre-defined classes but also the new data clusters. Then, the domain expert can judge whether certain new cluster is meaningful enough to get a new data label for the next round of the iteration. If so, we add the new labeled data into the labeled data set for the next round of computation. After several rounds, all the unlabeled data will be classified into the full-fledged data classes.

If we generate the *must-link* constraints for any pair of instances in the same pre-defined class and the *cannot-link* constraints for any pair of instances in two of the pre-defined classes, the problem of semi-defined classification can be considered as a semi-supervised clustering task with hard constraints. However, it also own its distinct characteristics:

● In traditional semi-supervised clustering the supervision knowledge is too far from being representative of a target classification, thus supervised learning might not be possible. However, in semi-defined classification we may have enough supervision information, in terms of the labeled data, for the

1550-4786/10 $26.00 © 2010 IEEE
DOI 10.1109/ICDM.2010.13

supervised learning over the known classes of the data.

• The constraints in traditional semi-supervised clustering may cover the instances from all the data clusters, however, those in semi-defined classification only involve with the data from only the known classes.

• Traditional semi-supervised clustering does not require that the clustering result is consistent to all the constraints. However, for semi-defined classification this must be held since we need to map each known class label to certain data cluster.

Altogether, the differences between traditional semi-supervised clustering and semi-defined classification are summarized in Table I.

Table I
SEMI-SUPERVISED CLUSTERING VS. SEMI-DEFINED CLASSIFICATION

	Semi-supervised clustering	Semi-defined classification
size of constraints	small	big
coverage of constraints	any data cluster	only the known data classes
consistency of constraints	not required	required

In this paper we propose the topic modeling approaches to address semi-defined classification. The essential difference between our method and the previous semi-supervised clustering methods is that we use the data labels directly instead of converting them into pairwise constraints. Semi-supervised clustering is shown to work well when the size of document corpus is small compared to the dimensionality of the feature space [3] (under this situation the limited supervision in the form of constraints can effectively avoid to be stuck in local optima). However, it is still unknown whether these previous methods work well for semi-defined classification where the sizes of data collection and constraints are both big.

Compared with the well known topic modeling approaches such as PLSA [8] and LDA [6], our model has two latent variables y and z, corresponding to word topics and (known and unknown) class labels respectively. Thus, we call the proposed model *Double-latent-layered* LDA (D-LDA for short). For D-LDA, we first give its process to generate document corpus in a un-supervised way, and then derive the method of parameter learning by Gibbs Sampling. Next, we show that with the new variable z, representing document classes, it is natural to incorporate the labels of some documents into the learning of D-LDA. Specifically, when z is for a word from a labeled document it is set to the corresponding class label; when z is for a word from an unlabeled document it is sampled by the update function in Gibbs Sampling. In this sense the variable of z is actually semi-latent. Meanwhile, the variable of y still keeps the flexibility in exploiting word topics, which may be more meaningful for classification with the supervision from the labeled data. Therefore, after Gibbs Sampling any instance d can be assigned to $z_d = \arg\max_z p(z|d)$ where z can be any class label. If z_d is a pre-defined class d is classified into a known class. Otherwise, it is grouped into a new class.

To validate our model we construct the ten different data sets for semi-defined classification. On each data set we also use different ratio to sample the labeled instances from a fixed subset of the data classes. Our results are either comparable to (on one data sets), or significantly better (on the other nine data sets) than the six compared methods, including the state-of-the-art semi-supervised clustering methods. More interestingly, we analytically and empirically show how the model prior affects the effectiveness of the proposed model.

Outline. The remainder of this paper is organized as follows. In Section II we formulate the semi-defined classification problem and give some preliminaries. The model of D-LDA and its solutions by Gibbs Sampling, are detailed in Section III. Section IV presents the systematic experiments to validate our algorithms, followed by the related work in Section V. Finally, we give the conclusions and future works in Sections VI.

II. PROBLEM FORMULATION AND PRELIMINARIES

A. Problem formulation

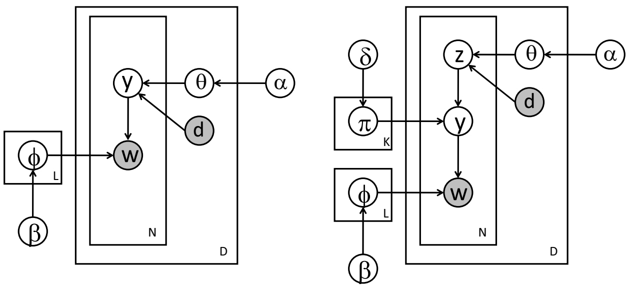
Formally, the problem of semi-defined classification can be stated as follows: given 1) the labeled data set $\mathcal{D}_l = \{(d_1, l_1), \cdots, (d_n, l_n)\}$ from the known classes $\mathcal{K}$ ($l_i \in \mathcal{K}$, $i \in \{1, \cdots, n\}$), 2) the unlabeled data set $\mathcal{D}_u = \{d_{n+1}, \cdots, d_{n+m}\}$ which includes the instances from both the known classes $\mathcal{K}$ and the some other unknown classes $\mathcal{U}$, we aim to produce a function $h : \mathcal{D} \rightarrow \mathcal{K} \cup \mathcal{U}$ that maps any object $d \in \mathcal{D}$ to its class label $l \in \mathcal{K} \cup \mathcal{U}$. Specifically, if $d_i (i \in \{n+1, \cdots, n+m\})$ comes from the the known classes $\mathcal{K}$ we aim to identify its true class label; meanwhile, we aim to group the instances not belonging to the known classes $\mathcal{K}$ into clusters.

Note also that although in this paper we mostly focus on the text data in semi-defined classification our model is generic to any dyadic data.

B. Preliminaries on topic modeling

PLSA [8] is a statistical model to analyze co-occurrence data by a mixture decomposition. Specifically, given the word-document co-occurrence matrix $\boldsymbol{O}$ whose element $O_{w,d}$ represents the frequency of word w appearing in document d, PLSA models $\boldsymbol{O}$ by using a mixture model with latent topics. All the parameters can be obtained by the EM solution to the maximum likelihood problem. LDA [6] with the parameter priors in the form of Dirichlet distribution gives the full generative process of document corpus.

Both PLSA and LDA contain only one latent variable, while the proposed model will contain two latent variables, corresponding to word topics and document classes respectively. Next we will only detail our extension to LDA for



(a) (b)

Figure 1. The graphical models of LDA and D-LDA

this requirement. And the extension to PLSA can be derived similarly, thus is omitted due to the space limitation.

III. Double-Latent-Layered LDA

The graphical model of D-LDA is shown in Figure 1(b), and its sibling of LDA is shown in Figure 1(a). D-LDA includes two latent variables, i.e. z for document classes and y for word topics. Compared with LDA the new variable z is added on the top of y. Thus, in D-LDA the distribution over word topics is directly dependent on the document class, while in LDA the distribution over word topics is generated directly for each document. More detailed, D-LDA assumes the following generative process for the document corpus:

1) draw multinomial distribution $\pi_{kl} = p(y = l|z = k)$ over word topics for each document class z from a Dirichlet prior δ; draw multinomial distribution $\phi_{lv} = p(w = v|y = l)$ over words for each word topic y from a Dirichlet prior β.
2) for each document d draw a multinomial distribution θ_d from a Dirichlet prior α; then each of the N words w_n
    a) choose a document class $z_n \sim \theta_d$;
    b) choose a word topic y_n from $p(y_n|z_n, \pi)$, the multinomial probability over word topics conditioned on the document class z_n;
    c) choose a word w_n from $p(w_n|y_n, \phi)$, a multinomial probability over words conditioned on the word topic y_n.

This way we incorporate the document class into this model. Thus, the class labels on some of the instances may supervise the generative process of document corpus. Meanwhile, the variable of y still keeps the flexibility in exploiting word topics, which could be more meaningful since they are conditioned on the document classes.

A. Gibbs Sampling for D-LDA

In this subsection we detail how to derive the parameters in D-LDA by Gibbs Sampling. If all the tokens in a corpus are flatted we get two vectors $\mathbf{w}$ and $\mathbf{d}$ where $w_i = v$ indicates that the word value of the i-th token is $v \in \{1, 2, \cdots, V\}$ and $d_i = d$ means that the document ID of the i-th token is $d \in \{1, 2, \cdots, D\}$. We aim to compute the posterior distribution of hidden variables given the input variables $\mathbf{w}, \mathbf{d}, \alpha, \beta, \delta$:

$$\begin{aligned} & p(\mathbf{y}, \mathbf{z}, \theta, \pi, \phi|\mathbf{w}, \mathbf{d}, \alpha, \beta, \delta) \\ &= \frac{p(\mathbf{y}, \mathbf{z}, \theta, \pi, \phi, \mathbf{w}, \mathbf{d}|\alpha, \beta, \delta)}{p(\mathbf{w}, \mathbf{d}|\alpha, \beta, \delta)}. \end{aligned} \tag{1}$$

Note that we use the symmetric Dirichlet prior α, β, δ in this work, and it is easy to use un-symmetric Dirichlet prior in this model.

Using Gibbs Sampling it is achieved via the random sampling of $\mathbf{z}, \mathbf{y}$ according to the update equation:

$$\mathcal{U} = p(z_i, y_i|\mathbf{z}_{-i}, \mathbf{y}_{-i}, \mathbf{w}, \mathbf{d}, \alpha, \beta, \delta), \tag{2}$$

where the subscript $-i$ denote the indices excluding i. By the detailed derivation in the appendix we get

$$\begin{aligned} \mathcal{U} = & \frac{O_{dk}^{(-i)} + \alpha}{\sum_{k=1}^{K}(O_{dk}^{(-i)} + \alpha)} \\ & \times \frac{O_{kl}^{(-i)} + \delta}{\sum_{l=1}^{L}(O_{kl}^{(-i)} + \delta)} \times \frac{O_{lv}^{(-i)} + \beta}{\sum_{v=1}^{V}(O_{lv}^{(-i)} + \beta)}, \end{aligned} \tag{3}$$

where $O_{dk}^{(-i)}$ denotes the occurrences of $(d_i = d \wedge z_i = k)$, $O_{kl}^{(-i)}$ denotes the occurrences of $(z_i = k \wedge y_i = l)$, $O_{lv}^{(-i)}$ denotes the occurrences of $(y_i = l \wedge w_i = v)$, all these counts should exclude the current one.

By (3) we can sample the two variables z_i, y_i simultaneously. Actually we have to compute $(L \times K)$ (where L and K are the numbers of word topics and document classes respectively) values for one sampling. We can also sample these two variables separately in the two steps as follows:

$$\begin{aligned} & p(z_i|\mathbf{z}_{-i}, \mathbf{y}, \mathbf{w}, \mathbf{d}, \alpha, \beta, \delta) = \\ & \frac{O_{dk}^{(-i)} + \alpha}{\sum_{k=1}^{K}(O_{dk}^{(-i)} + \alpha)} \times \frac{O_{kl}^{(-i)} + \delta}{\sum_{l=1}^{L}(O_{kl}^{(-i)} + \delta)}, \end{aligned} \tag{4}$$

$$\begin{aligned} & p(y_i|\mathbf{z}, \mathbf{y}_{-i}, \mathbf{w}, \mathbf{d}, \alpha, \beta, \delta) = \\ & \frac{O_{kl}^{(-i)} + \delta}{\sum_{l=1}^{L}(O_{kl}^{(-i)} + \delta)} \times \frac{O_{lv}^{(-i)} + \beta}{\sum_{v=1}^{V}(O_{lv}^{(-i)} + \beta)}. \end{aligned} \tag{5}$$

(4) and (5) can be derived similarly to the derivation of (3). Since sampling by (4) and (5) only need the computing of $(L+K)$ values it is more efficiently than the process in (3). After some initial experiments we find that sampling these two variables separately does not sacrifice the performance. Thus, our experiments adopt this more efficient one.

After the Gibbs Sampling process all the parameters in the model can be obtained as follows,

$$\theta_{dk} = \frac{O_{dk}^{(-i)} + \alpha}{\sum_{k=1}^{K}(O_{dk}^{(-i)} + \alpha)},$$

$$\pi_{kl} = \frac{O_{kl}^{(-i)} + \delta}{\sum_{l=1}^{L}(O_{kl}^{(-i)} + \delta)},$$

$$\phi_{lv} = \frac{O_{lv}^{(-i)} + \beta}{\sum_{v=1}^{V}(O_{lv}^{(-i)} + \beta)}.$$

B. Incorporating label information into D-LDA

It is easy to incorporate the class label of any $d \in \mathcal{D}_l$ into D-LDA. Specifically, if the class label of a document is known we just set the corresponding z to the label rather than sample it by the update function. If the class label of a document is unknown we still have to sample its z by the update function. In this sense the variable of z is actually *semi-latent* , and it helps to inject the supervision into the Gibbs Sampling process.

After the process of Gibbs Sampling converges, each unlabeled document d is assigned to the document class

$$z_d = \arg\max_k O_{dk}. \tag{6}$$

When $z_d \in \mathcal{K}$, d is actually classified into one of the known classes. When $z_d \in \mathcal{U}$, d is grouped into one of the unknown classes. The whole procedure of D-LDA for semi-defined classification is depicted in Algorithm 1.

Algorithm 1 D-LDA for Semi-defined Classification

```
Input: the data set $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$, from which we
       can obtain the two vector $\mathbf{w}$, $\mathbf{d}$; the number of
       iterations $T$ and the number of word topics $L$, the
       number of document classes $K = |\mathcal{K}| + |\mathcal{U}|$; the
       priors $\alpha, \beta, \delta$.
Outputthe class label for each document $d \in \mathcal{D}_u$.
procedure:
    1) $\mathbf{y}$ is initialized randomly, and $\mathbf{z}$ is set as fol-
       lows: when $d \in \mathcal{D}_l$, we set the corresponding
       $z$ to its true class label, otherwise we initialize
       it randomly.
    2) set $t := 1$
    3) while $t < T$
       a) for all $i$
          • Compute the counts $O_{dk}^{(-i)}$, $O_{kl}^{(-i)}$,
            $O_{lv}^{(-i)}$;
          • Update $z_i$ according to Equation (4),
            when $d \in \mathcal{D}_u$;
          • Update $y_i$ according to Equation (5);
          end for
       b) $t := t + 1$;
       end while
    4) Each document $d \in \mathcal{D}_u$ is assigned to the
       class label $z_d$ by Equation (6).
```

C. Discussion on the model prior δ

δ is the Dirichlet prior from which the multinomial distribution over all the word topics $\mathbf{y}$ for each document class z, denoted by $p(\mathbf{y}|z)$, is drawn. The smaller δ is, the more skewed the generated multinomial distribution is. In other words the small δ has the bias towards sparsity, and tend to pick the distributions favoring just a few word topics. Thus, the distributions generated by a smaller δ are more different from each other.

Therefore, if we know that in the given corpus the data distribution of a document class is greatly different from those of the other classes, we should select a small δ so that the generated distributions ($p(\mathbf{y}|z)$ for each z) are different from each other. On the contrary, if we know that in the given corpus the data distribution of a document class is similar to those of the other classes, we should select a big δ. In Section IV-D5 we will show that the experimental results coincide with this analysis.

IV. Experiments

In this section, we provide systemic experiments to show the superiority of our model D-LDA over the compared methods (Sections IV-D1 and IV-D2), and empirically analyze how some key factors, including the amount of labeled instances (Section IV-D3), the degree of data sparsity (Section IV-D4), and the model priors (Section IV-D5), affect the effectiveness of the model.

A. Data preparation

20Newsgroup: *20Newsgroup* is a collection of approximately 20,000 newsgroup documents, which is partitioned evenly cross 20 different newsgroups, each of which corresponds to a unique subcategory. These subcategories are further grouped into certain top categories. For example, the four subcategories *sci.crypt*, *sci.electronics*, *sci.med* and *sci.space* belong to the top category *sci*. We conduct our experiments on four top categories, and eight data sets are constructed to validate our algorithms, including four easy tasks (denoted as Easy1, Easy2, Easy3 and Easy4) and four difficult tasks (denoted as difficult1, difficult2, difficult3 and difficult4). The description of these eight tasks is detailed in Table II. To construct an easy task four subcategories from different top categories are selected (e.g., the data set of Easy1 consists of the four subcategories of *comp.graphics*, *rec.autos*, *sci.crypt* and *talk.politics.guns*). Since the four subcategories come from the four different top categories, the degree of the data distribution difference among the classes is great. Thus, it is easier to partition these data. On the other side, to construct a difficult task four subcategories from the same top category are selected. Obviously, the subcategories from the same top category have similar topics, thus they much more difficult to partition. The threshold of document frequency with value of 15 is used to cut down the number of word features.

ODP and Amazon: These two data sets are collected by Yin et al. [16], which are originally used for web object classification by exploiting social tags. The ODP

Table II
THE TOP CATEGORIES AND THEIR SUBCATEGORIES IN 20NEWSGROUPS

Tasks	Subcategories from Top Categories
Easy1	***comp.graphics***, ***rec.autos*** *sci.crypt*, *talk.politics.guns*
Easy2	***comp.os.ms-windows.misc***, ***rec.motorcycles*** *sci.electronics*, *talk.politics.mideast*
Easy3	***comp.sys.ibm.pc.hardware***, ***rec.sport.baseball*** *sci.med*, *talk.politics.misc*
Easy4	***comp.sys.mac.hardware***, ***rec.sport.hockey*** *sci.space*, *talk.religion.misc*
Difficult1	***comp.graphics***, ***comp.os.ms-windows.misc*** *comp.sys.ibm.pc.hardware*, *comp.sys.mac.hardware*
Difficult2	***rec.autos***, ***rec.motorcycles*** *rec.sport.baseball*, *rec.sport.hockey*
Difficult3	***sci.crypt***, ***sci.electronics*** *sci.med*, *sci.space*
Difficult4	***talk.politics.guns***, ***talk.politics.mideast*** *talk.politics.misc*, *talk.religion.misc*

data contain 5536 web pages from 8 categories, and the Amazon data include 6155 products information from the same 8 categories. These data are detailed in Table 1 in[16]. Since the features on each object (web page in ODP and product in Amazon) are the social tags on it, these data are extremely sparse. Specifically, the average numbers of tag words on the objects from ODP and Amazon are 25.76 and 36.75 respectively. These numbers are much smaller than that (more than 160) in 20Newsgroup.

To generate the semi-defined classification tasks, for each data set we randomly select k (e.g., $k = 2$ in our experiments) classes as the known classes (In the experiments, the known classes are marked with bold in Table II, and for the tasks odp and amazon, the known classes are Books and Electronic), and the left classes are used as unknown classes. For the k known classes, we randomly sample a subset of the data as the labeled instances, and the sampling ratio r ranges from 0.1 to 0.6 with an interval 0.05. For each ratio r we sample the labeled instances three times and the average results over these samplings are reported.

B. Baseline methods and implementation details

We compare our proposed model D-LDA for semi-defined classification with the following three types of methods:

• Semi-supervised clustering with constraints. We generate the pairwise constrains from labeled data to perform semi-supervised clustering. Let n_i be the number of labeled instances in class i, then we can generate $\sum_{i=1}^{k} n_i(n_i-1)/2$ *must-link* constraints and $\sum_{i=1}^{k-1}\sum_{j=i+1}^{k} n_i \times n_j$ *cannot-link* constraints. Obviously, the number of constraints increases in square order of the number of labeled instances. The semi-supervised clustering algorithms of MPCKMeans [4] and SS-Kernel-Kmeans [10] are considered for this comparison. Specifically, two variants of SS-Kernel-Kmeans with the linear kernel and exponential kernel, denoted as SSKK_Linear and SSKK_Exp respectively, are evaluated. The parameters of SS-Kernel-Kmeans are carefully tuned in the preliminary experiments.

• Two-step method. In this method, first we train a classifier (with the prediction confidence) on the labeled data from the known classes, and then use it to predict all the unlabeled data. If the prediction confidence over an unlabeled instance is bigger than a user-specified threshold τ it is assigned to the known class label. Next, we cluster the data with the prediction confidence lower than τ by an unsupervised clustering algorithm. In our experiments we use Logistic Regression [9] for the first step and CLUTO[1] for the seconde step. We carefully tune the threshold τ from 0.8 to 0.98 with an interval 0.02, and the best and average values are recorded, denoted as TwoStep$_{max}$ and TwoStep$_{mean}$, respectively.

• Un-supervised clustering. We also evaluate two unsupervised clustering methods in our experiments. One is the algorithm of MPCKMeans without any constraint, denoted by MPCKMeans_0. Another is CLUTO, the same with the one used in the second step of the Two-step method.

The parameters in D-LDA are set as follows for all the data sets. The number of document classes is set to the true class number, the number of word topics to 128, the iteration number to 2000, and the hyper parameters $\alpha = 0.2$, $\delta = 0.4$, $\beta = 0.01$. These parameters are tuned by some initial experiments.

C. Evaluation metrics

We evaluate all these methods in terms of the clustering effectiveness over all the data and the classification accuracy on the data from the known classes.

We adopt two popular metrics *normalized mutual information* (*NMI*) and *Pairwise F-measure* (*PF* for short) for clustering evaluation. *NMI* [7] measures how closely the clustering algorithm could reconstruct the underlying label distribution in the data. If L is the random variable denoting the underlying class labels on the data, and P is the random variable denoting the cluster assignments, then *NMI* measure is defined as:

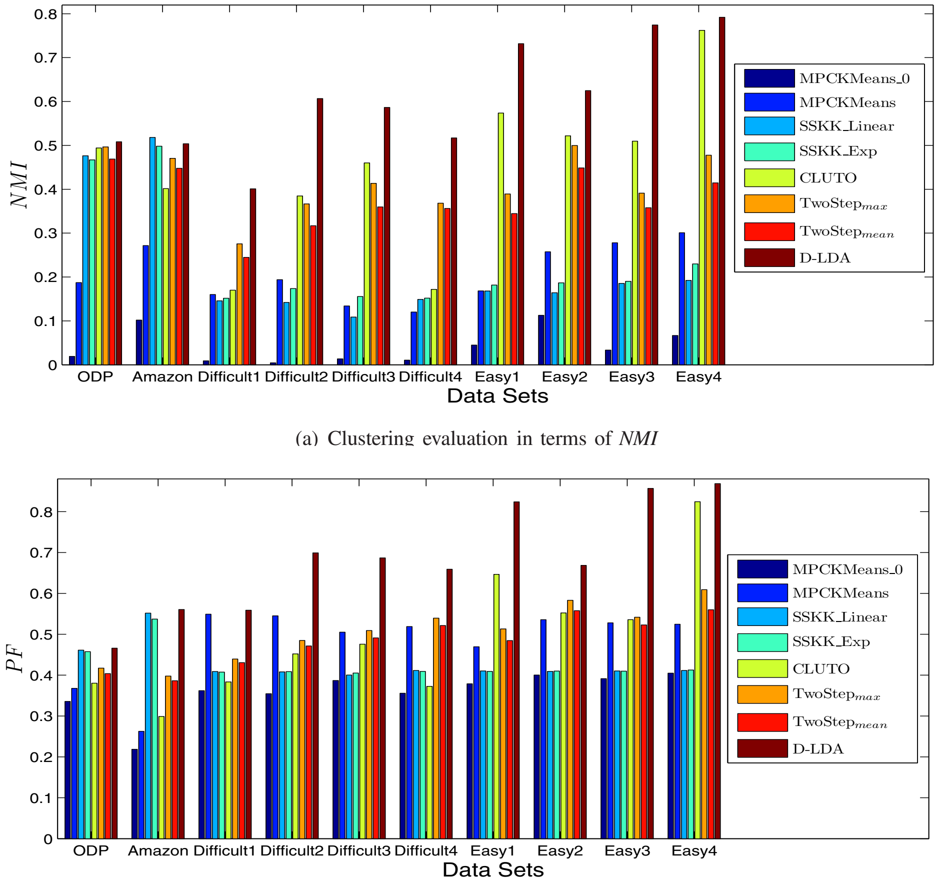
$$NMI = \frac{I(P,L)}{(H(P)+H(L))/2}, \tag{7}$$

where $I(X;Y) = H(X) - H(X|Y)$ is the mutual information between the random variables X and Y, $H(X)$ is the Shannon entropy of X, and $H(X|Y)$ is the conditional entropy of X given Y.

We follow the definition of *PF* in [2], which is the harmonic mean of pairwise precision and recall.

$$Pre = \frac{\#PairsCorrectlyPredictedInSameCluster}{\#TotalPairsPredictedInSameCluster}, \tag{8}$$

[1] The code from http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download, and with the parameter of ”-clmethod=direct -crfun=i1”.



(a) Clustering evaluation in terms of *NMI*

(b) Clustering evaluation in terms of *PF-measure*

Figure 2. Clustering evaluation over all the algorithms

$$Rec = \frac{\#PairsCorrectlyPredictedInSameCluster}{\#TotalPairsActuallyInSameCluster}, \tag{9}$$

$$PF = \frac{2 \times Pre \times Rec}{Pre + Rec}. \tag{10}$$

To evaluate the classification accuracy on the data from the known classes, we use the standard $F1$ measure. For each known class we calculate $F1_i$ as follows,

$$F1_i = \frac{2 \times Precison_i \times Recall_i}{Precison_i + Recall_i}, \quad i \in \{1, \cdots, k\}, \tag{11}$$

where $Precison_i$ and $Recall_i$ are the precision and recall on the i-th known class. Then,

$$F1 = \sum_i F1_i/k, \tag{12}$$

where k is the number of known classes.

D. Experimental results

We list all the comparison results in this subsection. For each data set, we have eleven sampling ratios and average the results over these ratios.

1) Clustering results: The clustering evaluation results over the ten tasks (eight from 20newsgroup, one on ODP and one on Amazon) are shown in Figure 2. The measures in Figures 2(a) and Figure 2(b) are *NMI* and *PF*, respectively. We have the following observations on these results:

1) it is clear that D-LDA significantly outperforms all the compared methods on the ten data sets, except that SSKK_Linear is slightly better than D-LDA on the task of Amazon.

2) We also find that the clustering results on the four easy tasks (Easy1, Easy2, Easy3 and Easy4) are much better than those on four difficult ones (Difficult1, Difficult2, Difficult3 and Difficult4). This observation coincides with our intension in generating these data sets.

2) Classification results on the known classes: Since it is not easy to map a document cluster output by the unsupervised and semi-supervised clustering methods to certain known class, we do not evaluate their classification accuracy on the documents from the known classes. Thus, we only compare D-LDA with the Two-step method in terms of classification accuracy. As shown in Figure 3, it is clear that D-LDA is significantly better than TwoStep$_{max}$ and

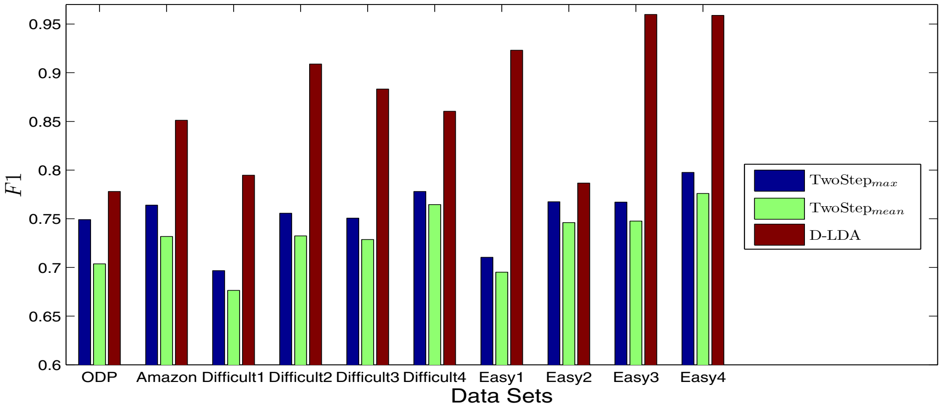


Figure 3. Classification results on the known classes in terms of $F1$

TwoStep$_{mean}$ in terms of $F1$.

3) The Effect of sampling ratio for the labeled instances: We also investigate the effect of sampling ratio r for the labeled instances on the performance of D-LDA. For each data set eleven sampling ratios are selected from the range [0.1, 0.6] with interval 0.05. We show these results over the eight tasks from *20Newsgroup* in Figure 4. Figure 4(a), Figure 4(b) and Figure 4(c) show the values of the measures *NMI*, $PF - measure$ and $F1$, respectively.

From these figures we find: the tendency that the more amount of the labeled data improves the effectiveness of D-LDA is more clear over the four difficult tasks than that over the easy ones. The reason may be the fact that for the easy tasks only a small amount of the labeled instances can reach its improvement limit, while for the difficult tasks more labeled instances continue to increase the effectivenes of D-LDA.

4) The effect of data sparsity: Then, we check how th data sparsity relates to the improvement of D-LDA. Here the degree of data sparsity can be measured by the averag number of the words in a document from the corpus. Th smaller this number is, the greater the degree of data sparsit is. Figure 5 shows the results of the improvement of D-LD over all the other methods (namely the difference betwee the two values), and the data sets are sorted by the decreasin order of data sparsity. As mentioned in Section IV-A, the tw data sets of ODP and Amazon are the most sparse ones, thu are ranked near the coordinate origin.

From Figure 5 we can observe the tendency that the less the degree of data sparsity is, the more improvement D-LDA achieves. The reason may come from that the topic modeling method performs better when the word-document co-occurrence is enough to find more meaningful topics. Thus, when the document in a corpus contains more words averagely D-LDA may achieve more improvement.

5) The effect of model prior δ: Finally, we will show how the prior δ, which generates the multinomial distributions over word topics for each document class, affects the effectiveness of D-LDA. Here, we select the two data sets of Difficult1 and Easy1. For each data set we randomly sample 60% of the labeled instances from the known classes. Fixing all the other parameters, we evaluate D-LDA under the 15 δ values, namely $\{0.02, 0.04, 0.08, 0.2, 0.4, 0.8, 2, 4, 8, 16, 32, 64, 128, 256, 1024\}$. The results are shown in Figure 6. It is clear that the difficult task, in which the data distribution of a document class is greatly different from those of the other classes, favors the smaller values of δ. On the contrary, the easy task favors the bigger values of δ. These experimental results empirically prove the correctness of our analysis in Section III-C. Therefore, it provides the guidance on how to select the prior δ according the prior knowledge on the data.

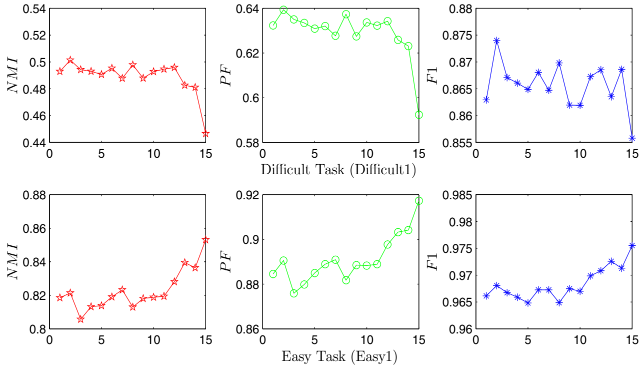
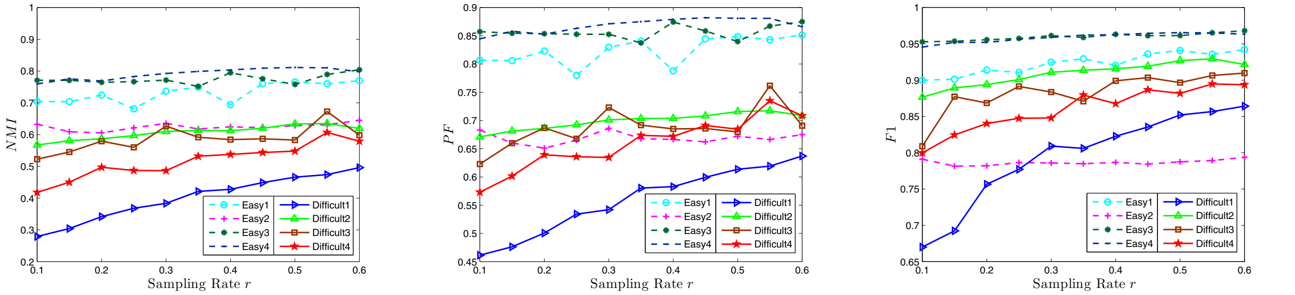


Figure 6. Clustering evaluation vs. the prior δ

E. Experiment summary

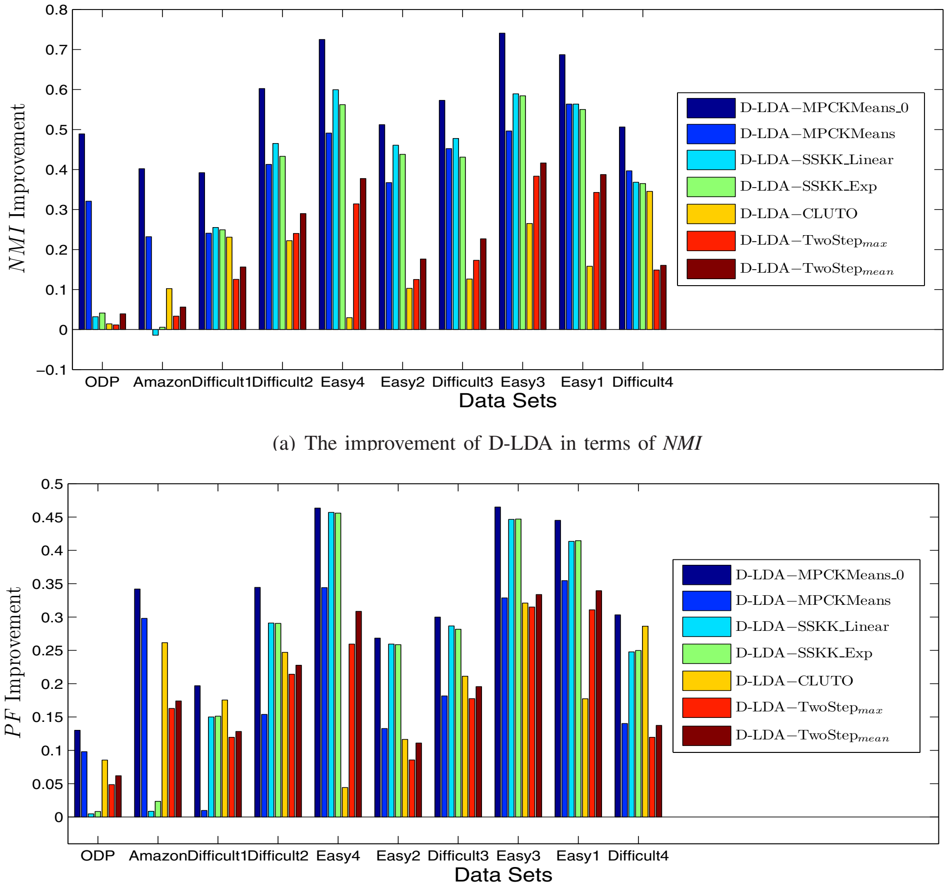
To highlight the achievements from these experiments we summarize them as follows.

- We compare D-LDA with the other six methods on the ten data sets for semi-defined classification. Our results are either comparable to (on one data set), or significantly better (on the other nine data sets) than the compared methods, in



(a) The performance of D-LDA vs. sampling ratio r on *NMI* (b) The performance of D-LDA vs. sampling ratio r on $PF - measure$ (c) The performance of D-LDA vs. sampling ratio r on $F1$

Figure 4. The performance of D-LDA vs. sampling ratio r



(a) The improvement of D-LDA in terms of *NMI*

(b) The improvement of D-LDA in terms of $PF - measure$

Figure 5. The improvement of D-LDA compared with all the other methods

terms of both clustering over all the data and classification over the data from the known classes.

• We also empirically analyze how the amount of labeled instances, the degree of data sparsity, and the model priors affect the effectiveness of D-LDA. More interestingly, we show how the prior knowledge on the degree of the data distribution difference among the document classes can be injected into the prior parameter δ to improve the model effectiveness.

V. Related works

In this section we introduce some previous works closely related with our work, including semi-supervised clustering, semi-supervised classification and topic modeling.

Semi-supervised clustering exploits a small amount of knowledge available to help partition unlabeled data into groups. Researchers have proposed many works in the past decade, such as [1], [2], [3], [11], [12], [13], [14], [15]. Basu et at. [3] developed a probabilistic framework for semi-supervised clustering based on Hidden Markov Random Fields. This framework provides a general form that combines constraints and distance learning where different distances can be used. Lu et al. [12] combined pairwise constraints with spectral clustering for semi-supervised clustering learning, in which the affinity information is propagated through pairwise constraints. Wang et al. [15] imposed the pairwise constraints into matrix factorization. Kulis et al. [10] formulated a framework that unified the vector-based and graph-based approaches for semi-supervised clustering. In which they proved an equivalence between a special case of the HMRF-based semi-supervised clustering objective and the kernel k-means objective function by constructing an appropriate kernel. All these methods consider the pairwise constraints, a more general form of supervision, to guide the clustering. Since these methods allows the violation of the constraints it is not easy to map the resultant data clusters to the known classes which is required by semi-defined classification. Although we can increase the values of cost for constraint violation it is still hard to select the right costs. Instead of using constraints as supervision we directly leverage label information in our model. The experiments show that our model significantly outperforms them for semi-defined classification.

Semi-supervised classification makes use of a large amount of unlabeled data together with a small set of labeled data, to build better classifiers [17]. Different from semi-defined classification, the semi-supervised classification learning needs some labeled data from each class. It indicates that the data taxonomy is well defined in semi-supervised classification. Thus, semi-supervised classification methods cannot be used for semi-defined classification.

Topic modeling provides simple ways to analyze large volumes of dyadic data. Most topic models, such as PLSA [8] and LDA [6], are unsupervised. They can be viewed as the methods of co-clustering for both features and instances. Blei and McAuliffe [5] proposed supervised LDA which can predict response values for new instances. However, this model is only for supervised learning. Our model D-LDA uses two separate latent variables for document clusters and word clusters. The variable for document clusters gives the interface to inject the label information into the model, while the variable for word clusters maintains the flexibility in exploiting meaningful word topics.

VI. Conclusions and future Works

In this paper we formulate an interesting problem of semi-defined classification, where the training data are from some known classes and the test data include not only the ones from the known classes but also some instances from the other unknown classes. We aim to simultaneously identify the instances from the known classes and group the left test data into some meaningful clusters. Methods to semi-defined classification are very helpful under the situation that we cannot give the full version of the data taxonomy and want to exploit other meaningful data classes except the known ones. Along this line we propose the topic model of D-LDA for this task. The experiments on the tasks built from the real-world data sets validate the effectiveness of this model.

With the proposed model of D-LDA, the building of the full data taxonomy for a corpus becomes an interactive and explorative process. After each round of semi-defined classification, human judgement is required to check whether it is necessary to give a new class label to certain new data cluster and then select the representative points in the new class as the new labeled data for the next round of semi-defined classification. How do we select these instances for labeling in order that the next round of computing is more effective? This task, the combination of semi-defined classification and active learning, will be our promising future work.

VII. Acknowledgments

The authors Fuzhen Zhuang, Qing He and Zhongzhi Shi are Supported by the National Natural Science Foundation of China (No. 60933004, 60975039, 61035003, 60903141, 61072085), National Basic Research Priorities Programme (No.2007CB311004) and National Science and Technology Support Plan (No.2006BAC08B06).

References

[1] S. Basu, A. Banerjee, and R. Mooney. Semi-supervised clustering by seeding. In *Proc. of the 19th ICML*, 2002.

[2] S. Basu, A. Banerjee, and R. J. Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of SDM*, 2004.

[3] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semisupervised clustering. In *Proc. of the 10th ACM SIGKDD*, 2004.

[4] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proc. of the 21th ICML*, 2004.

[5] D. M. Blei and J. D. McAuliffe. Sepervised topic models. In *Proc. of the 21th NIPS*, 2007.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 2003.

[7] T. M. Cover and J. A. Thomas. Elements of information theory. *Wiley-Interscience*, 1991.

[8] T. Hofmann. Probabilistic latent semantic analysis. In *Proc. of the 15th UAI*, 1999.

[9] D. Hosmer and S. Lemeshow. *Applied Logistic Regression*. Wiley, New York, 2000.

[10] B. Kulis, S. Basu, I. Dhillon, and R. Mooney. Semi-supervised graph clustering: A kernel approach. *Journal of Machine Learning*, 2008.

[11] H. Lee, J. Yoo, and S. J. Choi. Semi-supervised nonnegative matrix factorization. *IEEE Signal Processing Letters, Vol 17 (1)*, 2010.

[12] Z. D. Lu and M. A. Carreira-Perpinan. Constrained spectral clustering through affinity propagation. In *Proc. of CVPR*, 2008.

[13] W. Tang, H. Xiong, S. Zhong, and J. Wu. Enhancing semi-supervised clustering: A feature projection perspective. In *Proc. of the 13th ACM SIGKDD*, 2007.

[14] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proc. of the 18th ICML*, 2001.

[15] F. Wang, T. Li, and C. S. Zhang. Semi-supervised clustering via matrix factorization. In *Proceedings of SDM*, 2008.

[16] Z. J. Yin, R. Li, Q. Z. Mei, and J. W. Han. Exploring social tagging graph for web object classification. In *Proc. of the 15th ACM SIGKDD*, 2009.

[17] X. J. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. of the 20th ICML*, 2003.

APPENDIX

Step1. expand joint distribution and integrate the parameters:: The joint distribution given α, β, δ can be written as follow according to Figure 1(b):

$$\begin{aligned} & p(\mathbf{y}, \mathbf{z}, \mathbf{w}, \mathbf{d}, \theta, \phi, \pi | \alpha, \beta, \delta) \\ & = p(\theta|\alpha)p(\pi|\delta)p(\phi|\beta)p(\mathbf{y}|\mathbf{z}, \pi)p(\mathbf{w}|\mathbf{y}, \phi)p(\mathbf{z}|\mathbf{d}, \theta)p(\mathbf{d}) \end{aligned} \tag{13}$$

Integrate each side of (13) over θ, π, ϕ we have:

$$\begin{aligned} & p(\mathbf{y}, \mathbf{z}, \mathbf{w}, \mathbf{d} | \alpha, \beta, \delta) \\ & = p(\mathbf{d}) \cdot \int p(\theta|\alpha)p(\mathbf{z}|\mathbf{d}, \theta)\ \mathrm{d}\theta \cdot \int p(\pi|\delta)p(\mathbf{y}|\mathbf{z}, \pi)\mathrm{d}\pi \\ & \cdot \int p(\phi|\beta)p(\mathbf{w}|\mathbf{y}, \phi)\ \mathrm{d}\phi \\ & = p(\mathbf{d}) \cdot p(\mathbf{z}|\mathbf{d}, \alpha) \cdot p(\mathbf{y}|\mathbf{z}, \delta) \cdot p(\mathbf{w}|\mathbf{y}, \beta) \end{aligned} \tag{14}$$

Step2. the update function in the form of the joint distributions:: Using Bayes rule, the right side of (2) is

$$\mathcal{U} = \frac{p(z_i, y_i, \mathbf{z}_{-i}, \mathbf{y}_{-i}, \mathbf{w}, \mathbf{d} | \alpha, \beta, \delta)}{p(\mathbf{z}_{-i}, \mathbf{y}_{-i}, \mathbf{w}, \mathbf{d} | \alpha, \beta, \delta)} \tag{15}$$

The denominator could be written as:

$$p(\mathbf{z}_{-i}, \mathbf{y}_{-i}, \mathbf{w}_{-i}, \mathbf{d}_{-i} | w_i, d_i, \alpha, \beta, \delta) \cdot p(w_i, d_i) \tag{16}$$

Since w_i, d_i are independent to $\mathbf{z}_{-i}, \mathbf{y}_{-i}$ and $p(w_i, d_i)$ is constant, we have:

$$\mathcal{U} \propto \frac{p(\mathbf{z}, \mathbf{y}, \mathbf{w}, \mathbf{d} | \alpha, \beta, \delta)}{p(\mathbf{z}_{-i}, \mathbf{y}_{-i}, \mathbf{w}_{-i}, \mathbf{d}_{-i} | \alpha, \beta, \delta)} \tag{17}$$

Step3. expand the joint distributions in the update equation:: Note that $p(\mathbf{z}_{-i}|\mathbf{d}_{-i}, \alpha) = p(\mathbf{z}_{-i}|\mathbf{d}, \alpha)$, $p(\mathbf{y}_{-i}|\mathbf{z}_{-i}, \delta) = p(\mathbf{y}_{-i}|\mathbf{z}, \delta)$ and $p(\mathbf{w}_{-i}|\mathbf{y}_{-i}, \beta) = p(\mathbf{w}_{-i}|\mathbf{y}, \beta)$. Substituting (14) into (17) we have:

$$\begin{aligned} \mathcal{U} & \propto \frac{p(\mathbf{z}|\mathbf{d}, \alpha) \cdot p(\mathbf{y}|\mathbf{z}, \delta) \cdot p(\mathbf{w}|\mathbf{y}, \beta)}{p(\mathbf{z}_{-i}|\mathbf{d}, \alpha) \cdot p(\mathbf{y}_{-i}|\mathbf{z}, \delta) \cdot p(\mathbf{w}_{-i}|\mathbf{y}, \beta)} \\ & = \frac{p(z_i, \mathbf{z}_{-i}|\mathbf{d}, \alpha)}{p(\mathbf{z}_{-i}|\mathbf{d}, \alpha)} \cdot \frac{p(y_i, \mathbf{y}_{-i}|\mathbf{z}, \delta)}{p(\mathbf{y}_{-i}|\mathbf{z}, \delta)} \cdot \frac{p(w_i, \mathbf{w}_{-i}|\mathbf{y}, \beta)}{p(\mathbf{w}_{-i}|\mathbf{y}, \beta)} \end{aligned} \tag{18}$$

Using the Bayes rule again, we have:

$$\mathcal{U} \propto \underbrace{p(z_i|\mathbf{z}_{-i}, \mathbf{d}, \alpha)}_{\text{i}} \cdot \underbrace{p(y_i|\mathbf{y}_{-i}, \mathbf{z}, \delta)}_{\text{ii}} \cdot \underbrace{p(w_i|\mathbf{w}_{-i}, \mathbf{y}, \beta)}_{\text{iii}} \tag{19}$$

Step4. applying the properties of Dirichlet distribution:: Writing (19.i) back into integration form and for each document d we have:

$$\begin{aligned} & p(z_i = k|\mathbf{z}_{-i}, d_i = d, \alpha) = \int p(z_i = k|d_i = d, \theta_{dk}) \cdot \\ & p(\theta_{dk}|\mathbf{z}_{-i}, d_i = d, \alpha)\mathrm{d}\theta_{dk} \end{aligned} \tag{20}$$

where $p(z_i = k|d_i = d, \theta_{dk})$ is just θ_{dk} and $p(\theta_{dk}|\mathbf{z}_{-i}, d_i = d, \alpha)$ is posterior distribution of θ_{dk}. We use O_{dk} denotes the occurrences of $(d_i = d \wedge z_i = k)$, $O_{d(\cdot)}$ for the vector $(O_{d1}, O_{d2}, ..., O_{dK})$ and $O_{dk}^{(-i)}$ means the count should exclude the current one. Since $O_{d(\cdot)} \sim \mathrm{Mult}(\theta_{d(\cdot)})$ and the prior of θ_d is $\mathrm{Diri}(\alpha)$, the posterior distribution of θ_{dk} given the observation $(\mathbf{z}_{-i}, d_i = d)$ is $\mathrm{Diri}(\theta_{dk}; O_{d(\cdot)}^{(-i)} + \alpha)$. Then according to the conjugate relationship between Dirichlet distribution and multinomial distribution,

$$p(z_i = k|\mathbf{z}_{-i}, d_i = d, \alpha) = \int \theta_{dk} \cdot \mathrm{Diri}(\theta_{dk}; O_{d(\cdot)}^{(-i)} + \alpha)\mathrm{d}\theta_{dk} \tag{21}$$

We see that (21) is just the expectation of the posterior of θ_{dk}. Then ,

$$p(z_i = k|\mathbf{z}_{-i}, d_i = d, \alpha) = \frac{O_{dk}^{(-i)} + \alpha}{\sum_{k=1}^{K}(O_{dk}^{(-i)} + \alpha)} \tag{22}$$

Analogously for (19.ii) and (19.iii) we have:

$$p(y_i = l|\mathbf{y}_{-i}, z_i = k, \delta) = \frac{O_{kl}^{(-i)} + \delta}{\sum_{l=1}^{L}(O_{kl}^{(-i)} + \delta)}, \tag{23}$$

$$p(w_i = v|\mathbf{w}_{-i}, y_i = l, \beta) = \frac{O_{lv}^{(-i)} + \beta}{\sum_{v=1}^{V}(O_{lv}^{(-i)} + \beta)}, \tag{24}$$

where O_{kl} and O_{lv} is the occurrences of $(z_i = k \wedge y_i = l)$ and $(y_i = l \wedge w_i = v)$ and the superscript $\bullet^{(-1)}$ still means the counts should exclude the current one. Substituting them into (19) we have the result in (3).