Toward More Accurate Bug Triaging with Topic Modeling: Technical Report2

Xin Xia[1,2], Ying Ding[1], David Lo[1], Jafar M. Al-Kofahi[3],
Tien N. Nguyen[3], and Xinyu Wang[2]
[1]School of Information Systems, Singapore Management University, Singapore
[2]College of Computer Science and Technology, Zhejiang University, China
[3]Electrical and Computer Engineering Department, Iowa State University, USA
xxkidd@zju.edu.cn, {ying.ding.2011, davidlo}@smu.edu.sg,
{jafar, tien}@iastate.edu, and wangxinyu@zju.edu.cn

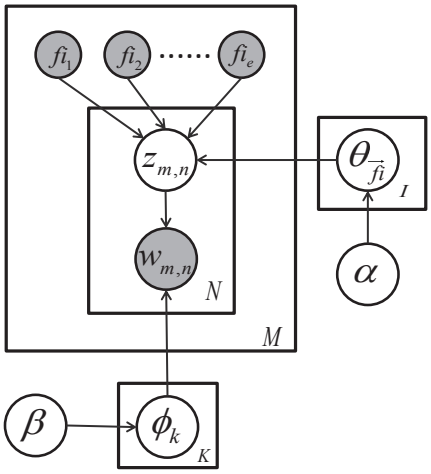

Figure 1: The Graphical Model of Multi-feature Topic Model (MTM)

In this technical report, we describe the detail steps to derive the Gibbs sampling function of MTM. We first describe the generative process of MTM by using more formal description. Next, we present the training phase of MTM, and the formula derivation. Finally, we present the inference of topics for a new bug report.

1. GENERATIVE PROCESS

Our proposed *multi-feature topic model (MTM)* is a generative model that considers the features of a bug report. The general graphical model of *MTM* is shown in Figure 1. The structure is similar to LDA with a few exceptions: first, we now have additional observed variables which are the features of bug reports such as the products and components of the reports. Also, rather than having a topic distribution per document, we now have a topic distribution per feature combination (denoted as θ in the graphical model).

Various features (e.g., product, component, reporter, version, platform, etc.) can be considered in *MTM*. We denote the features as fi_1, fi_2, $\cdots$ fi_e. In this work, we only use an instance of *MTM* with two features: product and component – we set fi_1 to be the product, and fi_2 to be the component. The generative process of *MTM* is as follows:

1. For a feature combination $\overrightarrow{fi} = (fi_1, fi_2, \cdots, fi_e)$ appearing in the corpus, sample the feature-topic distribution $\theta_{\overrightarrow{fi}}$ from $\mathbf{Dir}(\alpha)$;

2. For each topic $t = 1, 2, \cdots, K$, sample the topic-word distribution ϕ_t from $\mathbf{Dir}(\beta)$;

3. For each document (i.e., bug report) $m = 1, 2, \cdots, M$, with feature combination $\overrightarrow{fi}^m$:
   For each word w_{mn} in document m:
   Sample a topic z_{mn} from $\mathbf{Mult}(\theta_{\overrightarrow{fi}^m})$;
   Sample a word w_{mn} from $\mathbf{Mult}(\phi_{z_{mn}})$.

2. TRAINING PHASE

Table 1 summarizes the notations used in the remaining part of this section. Our goal is to generate the topics (i.e., z_{mn}) for all the words in the M documents (i.e., bug reports). Gibbs sampling is often used to recover the topics given a textual corpus and a graphical model [1]. Gibbs sampling is a Markov Chain Monte Carlo method, which is widely used in parameter estimation. In Gibbs sampling, we perform many iterations. For each iteration, we perform many steps. At each step, we estimate the value of one latent variable (e.g., the topic of a word in a document (i.e., bug report)) while keeping the values of all other latent variables constant (e.g., the topics of all other words in the documents). The process continues until a certain number of iterations have been performed or the process converges (i.e., there is no more change to the values assigned to the latent variables).

At each step of Gibbs sampling, we need to calculate the probability of assigning topic k to each word w_{mn} based

on the topic assignments of all other words and all other variables of the graphical model:

$$p(z_{mn} = k|\boldsymbol{Z}_{-mn}, \boldsymbol{W}, \boldsymbol{FI}, \alpha, \beta)$$

Here, $\boldsymbol{Z}$ denotes the topic assignment of all words made so far, $\boldsymbol{Z}_{-mn}$ denotes the topic assignment of all words excluding the n^{th} word in the m^{th} document, $\boldsymbol{W}$ denotes all the words in the bug report collections, and $\boldsymbol{FI}$ denotes the feature information collection of all reports (i.e., feature combinations of each bug report in the collection). We refer to this probability as the *Gibbs sampling function*. The derivation of the Gibbs sampling functions used in the training phase (i.e., Equation (12)) is as follows:

The Gibbs sampling function for the training phase can be calculated by Bayes theorem as follows:

$$\begin{aligned} & p(z_{mn} = k|\boldsymbol{Z}_{-mn}, \boldsymbol{W}, \boldsymbol{FI}, \alpha, \beta) \\ = & \frac{p(z_{mn}=k, \boldsymbol{Z}_{-mn}, \boldsymbol{W}|\boldsymbol{FI}, \alpha, \beta)}{p(\boldsymbol{Z}_{-mn}, \boldsymbol{W}|\boldsymbol{FI}, \alpha, \beta)} \\ = & \frac{p(\boldsymbol{Z}, \boldsymbol{W}|\boldsymbol{FI}, \alpha, \beta)}{p(\boldsymbol{Z}_{-mn}, \boldsymbol{W}_{-mn}|\boldsymbol{FI}, \alpha, \beta) p(w_{mn}|\boldsymbol{Z}_{-mn}, \boldsymbol{W}_{-mn}, \boldsymbol{FI}, \alpha, \beta)} \\ \propto & \frac{p(\boldsymbol{Z}, \boldsymbol{W}|\boldsymbol{FI}, \alpha, \beta)}{p(\boldsymbol{Z}_{-mn}, \boldsymbol{W}_{-mn}|\boldsymbol{FI}, \alpha, \beta)} \end{aligned} \quad (1)$$

Since $p(w_{mn}|\boldsymbol{Z}_{-mn}, \boldsymbol{W}_{-mn}, \boldsymbol{FI}, \alpha, \beta)$ is not related to the value of z_{mn}, we simplify it. Based on our graphical model, the characteristic joint distribution of multi-feature topic model is:

$$p(\boldsymbol{Z}, \boldsymbol{W}|\alpha, \beta, \boldsymbol{FI}) = p(\boldsymbol{W}|\boldsymbol{Z}, \beta) p(\boldsymbol{Z}|\alpha, \boldsymbol{FI}) \quad (2)$$

The first term at the right part of the above equation is the same as the first term of the characteristic joint distribution of LDA (c.f. [1]), which is:

$$p(\boldsymbol{W}|\boldsymbol{Z}, \beta) = \prod_{k=1}^{K} \frac{\mathbf{Del}(C_{k\cdot}^{KV} + \beta)}{\mathbf{Del}(\beta)} \quad (3)$$

where $C_{k\cdot}^{KV}$ is the k^{th} row of matrix C^{KV}, which is a vector representing the number of times each word is assigned to topic k. **Del** is the Dirichlet Delta function, c.f., [1], which can be calculated as:

$$\mathbf{Del}(\beta) = \frac{\prod_{w=1}^{V} \Gamma(\beta)}{\Gamma(\beta V)} \quad (4)$$

Similarly, $p(\boldsymbol{Z}|\alpha, \boldsymbol{FI})$ is derived as follows,

$$\begin{aligned} p(\boldsymbol{Z}|\alpha, \boldsymbol{FI}) &= \int p(\boldsymbol{Z}, \theta|\alpha, \boldsymbol{FI}) \, d\theta \\ &= \int p(\boldsymbol{Z}|\theta, \boldsymbol{FI}) p(\theta|\alpha, \boldsymbol{FI}) \, d\theta \end{aligned} \quad (5)$$

and

$$\begin{aligned} p(\theta|\alpha, \boldsymbol{FI}) &= \prod_{\overrightarrow{fi}} p(\theta_{\overrightarrow{fi}}|\alpha) \\ &= \prod_{\overrightarrow{fi}} \frac{1}{\mathbf{Del}(\alpha)} \times \prod_{k=1}^{K} \theta_{\overrightarrow{fi},k}^{a-1} \end{aligned} \quad (6)$$

$$\begin{aligned} p(\boldsymbol{Z}|\theta, \boldsymbol{FI}) &= \prod_{m=1}^{M} \prod_{n=1}^{L_m} \prod_{k=1}^{K} \theta_{\overrightarrow{fi}^m,k}^{I(z_{mn}=k)} \\ &= \prod_{m=1}^{M} \prod_{k=1}^{K} \theta_{\overrightarrow{fi}^m,k}^{C(m,k)} \end{aligned} \quad (7)$$

where L_m is the length of m^{th} document and $I(\cdot)$ is an indicator function which returns 1 when the condition is true, $C(m,k)$ is document-topic count, i.e., the number of times topic k is assigned to the words in the m^{th} document. Combining the equations (5), (6) and (7), we derive:

$$\begin{aligned} p(\boldsymbol{Z}|\alpha, \boldsymbol{FI}) &= \int \prod_{\overrightarrow{fi}} \frac{1}{\mathbf{Del}(\alpha)} \prod_{k=1}^{K} \theta_{\overrightarrow{fi},k}^{C_{\overrightarrow{fi},k}^{IK} + \alpha - 1} d\theta \\ &= \prod_{\overrightarrow{fi}} \frac{\mathbf{Del}(C_{\overrightarrow{fi},\cdot}^{IK} + \alpha)}{\mathbf{Del}(\alpha)} \end{aligned} \quad (8)$$

In the above equation, C^{IK} is the feature-topic count matrix (see Table 1), and $C_{\overrightarrow{fi},k}^{IK}$ is the number of times words inside document with feature information $\overrightarrow{fi} = (fi_1, \cdots, fi_e)$ is assigned to topic k, and $C_{\overrightarrow{fi},\cdot}^{IK}$ is a vector of number of times each topic appears in documents with feature information $\overrightarrow{fi}$. From equations (3) and (8), we derive:

$$\begin{aligned} p(\boldsymbol{Z}, \boldsymbol{W}|\alpha, \beta, \boldsymbol{FI}) &= \prod_{k=1}^{K} \frac{\mathbf{Del}(C_{k\cdot}^{KV} + \beta)}{\mathbf{Del}(\beta)} \times \\ & \prod_{\overrightarrow{fi}} \frac{\mathbf{Del}(C_{\overrightarrow{fi},\cdot}^{IK} + \alpha)}{\mathbf{Del}(\alpha)} \end{aligned} \quad (9)$$

Similarly, we can get the denominator of Equation (1):

$$\begin{aligned} & p(\boldsymbol{Z}_{-mn}, \boldsymbol{W}_{-mn}|\alpha, \beta, \boldsymbol{FI}) = \\ & \prod_{k=1}^{K} \frac{\mathbf{Del}(C_{k\cdot,-mn}^{KV} + \beta)}{\mathbf{Del}(\beta)} \times \prod_{\overrightarrow{fi}} \frac{\mathbf{Del}(C_{\overrightarrow{fi},\cdot,-mn}^{IK} + \alpha)}{\mathbf{Del}(\alpha)} \end{aligned} \quad (10)$$

The Gamma function $\Gamma(a)$ has an important property:

$$\Gamma(a) = a \times \Gamma(a-1) \qquad a > 1 \quad (11)$$

Based on this, we can get the final Gibbs sampling function:

$$\begin{aligned} p(z_{mn} = k|\boldsymbol{Z}_{-mn}, \boldsymbol{W}, \boldsymbol{FI}, \alpha, \beta) \propto & \frac{C_{kv,-mn}^{KV} + \beta}{\sum_{v'} C_{kv',-mn}^{KV} + V\beta} \\ & \times \frac{C_{\overrightarrow{fi},k,-mn}^{IK} + \alpha}{\sum_{k'} C_{\overrightarrow{fi},k',-mn}^{IK} + K\alpha} \end{aligned} \quad (12)$$

Here, $C_{kv,-mn}^{KV}$ is the number of times word v is assigned to topic k excluding w_{mn} (i.e., the n^{th} word in the m^{th} document). $C_{\overrightarrow{fi},k,-mn}^{IK}$ is the number of times topic k is assigned to a word inside a document with feature combination $\overrightarrow{fi}$ excluding w_{mn}. In other words,

Table 1: Symbols Associated with Multi-feature Topic Model

Notation	Type	Description
M	scalar	Numbers of documents (i.e., bug reports) in the document collection.
K	scalar	Numbers of topics.
V	scalar	Number of unique terms in the documents.
e	scalar	Number of features.
α	scalar	Dirichlet prior, hyperparameter for the topic distribution for each feature combination.
β	scalar	Dirichlet prior, hyperparameter for the word distribution for each topic.
E	vector	Different feature combinations of bug reports in the document collection.
I	scalar	Number of feature combinations.
$\overrightarrow{fi}$	vector	Vector representation of a feature combination, i.e., $\overrightarrow{fi} \in E$, and $\overrightarrow{fi} = (fi_1, fi_2, \cdots fi_e)$
$\boldsymbol{Z}$	vector	Topic assignment of all words.
$\boldsymbol{FI}$	vector	Feature combinations of bug reports in the document collection.
$\boldsymbol{W}$	vector	All words in the bug reports in the document collection.
ϕ_k	vector	Word distribution for topic k.
$\theta_{\overrightarrow{fi}}$	vector	Topic distribution for feature combination $\overrightarrow{fi}$.
$\overrightarrow{fi}^m$	vector	Feature combination of the m^{th} bug report, i.e., $\overrightarrow{fi}^m = (fi_1^m, \cdots, fi_e^m)$.
z_{mn}	scalar	Topic of the n^{th} word in the m^{th} bug report.
w_{mn}	scalar	n^{th} word in the m^{th} bug report.
C^{KV}	$K \times V$ matrix	Topic-word count matrix, i.e., a matrix containing the number of times various topics are assigned to various words in the document collection. Also, C_{kv}^{KV} is the number of times word v in the vocabulary is assigned to topic k.
C^{IK}	$I \times K$ matrix	Feature-topic count matrix, i.e., a matrix containing the number of times various topics are assigned to various feature combinations. A topic is assigned to a feature combination if it is assigned to a word inside a document with that feature combination. Also, $C_{\overrightarrow{fi},k}^{IK}$ is the number of times topic k is assigned to a word inside a document with feature combination $\overrightarrow{fi}$.

$$C_{kv,-mn}^{KV} = \begin{cases} C_{kv}^{KV} - 1 & \text{if } w_{mn} = v \text{ and } z_{mn} = k \\ C_{kv}^{KV} & \text{otherwise} \end{cases} \tag{13}$$

$$C_{\overrightarrow{fi},k,-mn}^{IK} = \begin{cases} C_{\overrightarrow{fi},k}^{IK} - 1 & \text{if } \overrightarrow{fi}^m = \overrightarrow{fi} \text{ and } z_{mn} = k \\ C_{\overrightarrow{fi},k}^{IK} & \text{otherwise} \end{cases} \tag{14}$$

In the above equation, C_{kv}^{KV} denotes the number of times word v is assigned to topic k. $\overrightarrow{fi}^m$ denotes the feature combination of the m^{th} document, i.e., $\overrightarrow{fi}^m = (fi_1^m, \cdots, fi_e^m)$. $C_{\overrightarrow{fi},k}^{IK}$ denotes the number of times topic k is assigned to a word inside a document with feature combination $\overrightarrow{fi}$.

Algorithm 1 The Training Phase of MTM.

Input: Document collection $\boldsymbol{C}$, Feature information collection of all reports $\boldsymbol{FI}$, Hyperparameters α, β, Topic number K, Maximum number of iteration $maxIters$.
Output: Topic assignment for all of the words in $\boldsymbol{C}$: $\boldsymbol{Z}$
Initialization:
1: **for** $m \leftarrow 1, \cdots, M$ **do**
2: **for** $n \leftarrow 1, \cdots, L_m$ **do**
3: Sample topic z_{mn} from **Mult**$(1/K, \cdots, 1/K)$
4: **end for**
5: **end for**
6: Initialize C^{KV} and C^{IK} matrices
Sampling:
7: **repeat**
8: **for** $m \leftarrow 1, \cdots, M$ **do**
9: **for** $n \leftarrow 1, \cdots, L_m$ **do**
10: Randomly assign a topic to w_{mn} according to Equation (12)
11: **end for**
12: **end for**
13: Update C^{KV} and C^{IK} matrices
14: **until** Converge or iterated more than $maxIters$ steps

The Gibbs sampling algorithm for the training phase of MTM is shown in Algorithm 1. In the algorithm, we have two matrices C^{KV} and C^{IK} which are updated at each Gibbs sampling step. C^{KV} is a topic-word count matrix , i.e., a matrix containing the number of times various topics are assigned to various words in the document collection $\boldsymbol{C}$, and C^{IK} is a feature-topic count matrix, i.e., a matrix containing the number of times various topics are assigned to various feature combinations. A topic is assigned to a feature combination if it is assigned to a word inside a document with that feature combination. In the initialization part (Lines 1 to 6), we first randomly assign a topic to each word in our corpus and update the two matrices according to this random assignment. In the sampling part (Lines 7 to 13), we perform the following steps iteratively:

1. For each word w_{mn}, we assign a topic randomly to it according to the probability computed using Equation (12). (Line 10).

2. After going through all words in our corpus, we update the two matrices (i.e., C^{KV} and C^{IK}) according to the new topic assignments (Line 13).

3. This process repeats many times until convergence or a predefined number of iterations has been reached (Line 14).

3. INFERENCE PHASE

In the inference phase, we infer the topic distribution of a new bug report m_{new}. We fix the topic assignments of words in the training bug reports, and use Gibbs sampling to iterate through the terms in the new bug report enough number of times to get their topic assignments. At each step of Gibbs sampling, we need to calculate the probability of assigning topic k to each word in m_{new}. In the inference

part, we estimate the topic assignments of words in a new document. For a new document m_{new} with feature combination fi^{new}, we can estimate the topic assignment of its n^{th} word. The joint probability of words and topics for m_{new} is:

$$p(\boldsymbol{Z}^{new}, \boldsymbol{W}^{new}, \boldsymbol{Z}, \boldsymbol{W} | \alpha, \beta, \overrightarrow{fi}^{new}, \boldsymbol{FI}) = p(\boldsymbol{W}^{new}, \boldsymbol{W} | \boldsymbol{Z}^{new}, \boldsymbol{Z}, \beta) p(\boldsymbol{Z}^{new}, \boldsymbol{Z} | \alpha, \overrightarrow{fi}^{new}, \boldsymbol{FI}) \quad (15)$$

Similar to the derivation of the Gibbs sampling function for the training part (i.e., Equation (12)), we can derive Equation (16). This probability is given by the following:

Algorithm 2 The Inference Phase of MTM.

Input: Words of training data $\boldsymbol{W}$, Topic assignments of words in training data $\boldsymbol{Z}$, Feature information collection of the training data $\boldsymbol{FI}$, Features of the new document $\overrightarrow{fi}^{new}$, Words in the new document (i.e., bug report) $\boldsymbol{W}^{new}$, Maximum number of iteration $maxIters$.
Output: Topic assignment of all the words in the new bug report: $\boldsymbol{Z}^{new}$
Initialization:
1: **for** $n \leftarrow 1, 2, \cdots, L_{new}$ **do**
2: Sample topic z_n from **Mult**$(1/K, \cdots, 1/K)$.
3: **end for**
4: Initialize C'^{KV} and C'^{K} matrices
Sampling:
5: **repeat**
6: **for** $n \leftarrow 1, 2, \cdots, L_{new}$ **do**
7: Randomly assign a topic to n^{th} word according to Equation (16)
8: **end for**
9: Update C'^{KV} and C'^{K} matrices
10: **until** Converge or iterated more than $maxIters$ steps

$$p(z_n^{new} = k | \boldsymbol{Z}_{-n}^{new}, \boldsymbol{Z}, \boldsymbol{W}^{new}, \boldsymbol{W}, \overrightarrow{fi}^{new}, \boldsymbol{FI}, \alpha, \beta) \propto \frac{C_{kv}^{KV} + C'^{KV}_{kv,-n} + \beta}{\sum_{v'} [C_{kv'}^{KV} + C'^{KV}_{kv',-n}] + V\beta} \times \frac{C^{IK}_{\overrightarrow{fi}^{new},k} + C'^{K}_{k,-n} + \alpha}{\sum_{k'} [C^{IK}_{\overrightarrow{fi}^{new},k'} + C'^{K}_{k',-n}] + K\alpha} \quad (16)$$

In the above equation, $\boldsymbol{Z}_{-n}^{new}$ denotes the current topic assignment of all words excluding the n^{th} word in the new bug report. $\boldsymbol{W}^{new}$ denotes all words in the new bug report. C'^{KV} and C'^{K} represent the topic-word count matrix and feature-topic count matrix for the new bug report m_{new}. $C'^{KV}_{kv,-n}$ denotes the number of times word v is assigned to topic k in the new bug report excluding the n^{th} word, $C'^{K}_{k,-n}$ is the number of times topic k is assigned to the new bug report excluding the n^{th} word.

Algorithm 2 presents the inference phase of MTM by using Gibbs sampling. In the algorithm, we have two matrices C'^{KV} and C'^{K} which are updated at each Gibbs sampling step. In the initialization part (Lines 1 to 4), we first randomly assign a topic to each word in the new bug report and update the two matrices according to this random assignment. In the sampling part (Lines 5 to 10), we perform the following steps iteratively:

1. For the n^{th} word, we assign a topic randomly to it according to the probability computed using equation (16) (Line 7).
2. After going through all words in the new bug report, we update the two matrices (i.e., C'^{KV} and C'^{K}) according to the new topic assignments (Line 9).
3. This process repeats many times until convergence or a predefined number of iterations has been reached (Line 10).

4. REFERENCES

[1] G. Heinrich, "Parameter estimation for text analysis," *Web: http://www.arbylon.net/publications/text-est.pdf*, 2005.