Modeling Statement Context to Surface even Rare Diffused Topics Automatically

Suparna Bhattacharya, Mrinal Kanti Das, Chiranjib Bhattacharyya, K. Gopinath

August 14, 2012

Abstract

Statistical topic models infer topics from statistical information contained in a dataset. Existing topic models can detect topics if they are present *prominently* in some files or scattered widely across the files. However, they fail if a topic is diffused across a small percentage of files or in other words if a topic is neither prominent inside any file nor diffused *widely* across files. In this work we explore the problem of detecting such *rare diffused* topics. We observe that the local context of lines in a file play a key role in surfacing these topics. We introduce various mechanisms to control a topic model's sensitivity towards local context. We propose CSTM (*Context Sensitive Topic Model*), a new model that is capable of discovering *prominent*, *widely diffused* as well as *rare diffused* topics by leveraging the context of individual lines within each file.

Rare diffused topics are quite common in software code, particularly in framework based software. We evaluate our model on surfacing software *concerns* automatically at the fine granularity of *individual program statements*. CSTM achieves a statement level concern assignment accuracy that agrees 70% of the time with typical programmer interpretation (as measured using systematically gathered feedback from 35 programmers for four Java applications). The ability to discover statement level concerns paves the way for a new class of automated analyses correlating latent concerns with program properties that vary at statement granularity. As a novel application, we demonstrate a completely unsupervised automatic summarization of byte-code execution profiles in terms of latent concerns.

1 Introduction

Topic models are widely used in various applications, where a topic is a distribution over the features of a dataset. In case of text a feature is word in the simplest form of representation. Despite its mathematical nature topics are found to correlate with human knowledge to some extent in grouping similar words. However, topics are inferred from the statistical information of a dataset, for example co-occurence of words in a collection of documents (corpus). A topic can be said to be present if most probable words corresponding to that topic is present. A topic can be present dominantly in one or more files (*prominent*), or it may not be present dominantly in a single file but is scattered widely across the files (*widely diffused*). Latent Dirichlet Allocation (LDA) [1] is the basic and the first of its kind to detect prominent topics, whereas MGLDA [2] is designed to detect widely diffused topics. But both of these models fail if a topic is neither dominant in a file nor scattered globally, that is if a topic is diffused across a small number of files (*rare diffused*). In this paper we investigate statistical probabilistic models to detect rare diffused topics. We observe that local context plays a key role in finding such rare topics, however the model should be very sensitive to the context. We devise a novel probabilistic model called *Context Sensitive Topic Model* (CSTM). We argue that rare diffused topics appear naturally in many places, for example in software source code when some routines are defined in library (which is unavailable) and used repeatedly in source files. Further, unless a routine performs an extremely common task, it is unlikely to be referred by a very large number of files; instead the specialized functionality manifests as a rare diffused topic.

Large complex software applications are routinely built using redeployable components and frameworks. While these layers ease the development process and enable highly flexible solutions, understanding performance properties of such software (e.g. for diagnosing inefficiencies, predicting resource consumption, finding runtime bloat, optimizing energy use) requires significant expertise and effort, even with state-of-the-art tools. An understanding of the domain or intent of the application can provide more insight than tools that report costs in terms of low level artifacts (such as methods and components). However, such information may not be available in practice when using unfamiliar code.

Motivating application: Automatic performance summarization based on latent concern discovery A promising alternative is to devise new automated performance summarization techniques that discover latent program concerns (in source code) and compute their runtime resource usage. Latent concerns reflect underlying intent and are not tied to assumptions about the specific nature of concerns. These could be features, non-functional requirements, design idioms, implementation mechanisms or other conceptual considerations that can impact the implementation of a program [3]. Summarizing dynamic properties such as runtime resource usage in terms of latent concerns can provide a novel perspective for performance understanding. For example, it could be used to aid the estimation of energy expended due to software bloat [4] arising from incidental concerns in large framework-based applications.

Statistical topic models One way to automatically summarize concerns in source code for approximate analysis is to use statistical topic models, such as Latent Dirichlet Allocation (LDA). The advantage of this approach over other concern identification and location techniques (whether generative or query-based [5],) is that it works without any additional input, knowledge about concern domains or other assumptions about concern characteristics. This makes it possible to extract a somewhat representative set of latent concerns from a large unfamiliar software code base in a fully automatic manner unbiased by apriori notions about the specific nature of concerns.

Technical challenge and scope Despite the possibilities that it can open up, we find that the use of statistical topic models as an input for applications of this nature is problematic as it raises demanding requirements that challenge the current state of the art in modeling concerns using LDA and its variants. For instance:

1. Interesting cross cutting concerns (in terms of resource usage) may not always have a prominent presence in any source module. This is especially true in framework based code where all underlying module sources may not be available. Existing models cannot usually surface these *rare* concerns as topics, as the statistical contribution of modular content typically dominates over statement level information (Section 3).

2. Accurate statement level granularity of concern assignment is required for concern-wise attribution of dynamic properties like runtime resource usage and bloat (because different statements contribute differently to resource usage). LDA is known to behave poorly on small documents with statistically insignificant textual content, such as individual source code statements seen in isolation.

While point 2 can be covered to a certain extent by suitably engineering a solution that combines existing methods from topic modeling research, addressing point 1 requires new model extensions. We find that even applying specialized models such as MG-LDA [2] to software fails to address these challenges (Section 3). Hence we propose a single extended model called CSTM that jointly addresses challenges 1 and 2. Our key contribution is to explicitly control the trade-off between modular and contextual contribution of concerns so that the model is more sensitive to contextual information contained in the neighborhood of a statement; this helps surface even diffused concerns at statement granularity. LDA and MG-LDA can be obtained as special cases of our more flexible model.

Evaluation methodology challenge To evaluate the effectiveness of our model, we collect measures of the diversity of concerns found for four Java applications and design synthetic experiments using a well-understood application, BerkeleyDB, to expose differences in the sensitivity of alternative models to diffused concerns. This is not enough, as we also need to confirm that the increased sensitivity obtained using our extensions does not hurt accuracy. However, concerns are subjective and represent human interpreted concepts; thus there is no one correct assignment for a given program (Section 2). This makes it difficult to quantitatively measure the accuracy of the models we investigate. Further, our main goal is not to find specific concerns or perform a specific predictive task, but to surface a representation or summarization that serves an exploratory purpose (such as performance understanding). Thus, neither standard information retrieval/concern location metrics like precision and recall for a specific concern nor intrinsic measures like held-out likelihood, are suitable for evaluating the effectiveness of the models for this purpose. Instead, we opt for a human (expert[1]) evaluation approach along the

[1] programmers familiar with Java

lines recommended in [6] for quantitatively judging interpretability of concern topics and their statement level assignments.

Contributions

- We devise a probabilistic model, CSTM (Context Sensitive Rare Topic Model), capable of *automatic discovery* of prominent, diffused and also rare concerns at *statement level granularity* without any human input or apriori knowledge (Section 4). The model assigns a mixture of concerns to each statement via a probabilistic inference procedure that is sensitive to both the surrounding statements (local context) in which the statement occurs and the containing module.

- We conduct a systematic evaluation of the sensitivity of diffused concern detection, interpretability of statement level concern assignments and the diversity of concerns found by our CSTM model and LDA-CS, an adaptation of LDA with an inference procedure for statement level assignment of concerns. Our detailed evaluation (Section 7) includes a programmer interpretability study where we compile 540 responses on word intrusion and topic relevance tasks by 35 programmers from different organizations.

- We illustrate a novel application of the model: computing cumulative byte-code profile summaries in terms of latent concerns (Section 6). This paves the way for a new class of automated analysis correlating latent concerns with program properties that vary at statement granularity.

2 Problem Definition

2.1 Topics

A topic is a distribution over the words, for example if the k^{th} topic is denoted by β_k and words indices are denoted by i, then $\sum_{i=1}^{V} \beta_{ki} = 1$, where V is the number of words. This implies that β_{kj} is the probability of picking word indexed by j for topic k. Two important caveats to be noted here is as follows. A topic is thus not linked to any semantic or inutitive knowledge and two topics can be very close. In other words topics are inferred from statistical information of a dataset, however despite this non-intuitive mathematical representation topics are found to correlate with human intuition to some extent [6].

Topic being a distribution over the words, many words may have non-zero probabilities, however in most of the cases except few words the probabilities are negligibly small, and therefore topic is sometimes referred as soft clusters of words and in many cases is represented by the most probable few words. A topic can be said to be present if the most probable representative words corresponding to that topic is present.

Prominent Topics We refer to those topics as prominent topics which are present dominantly in one or more files.

Widely Diffused Topics Widely diffused topics are not dominant in any single file, however they are present in a considerable number of files.

Rare Diffused Topics Rare diffused topics are those topics which are neither “prominent” in a single file nor “widely diffused” across many files. Rare diffused topics are typically present in small quantities (diffused) in a small number of files.

2.2 Concerns

To support the exploratory context of performance understanding, we take a very broad view of what constitutes a software concern:

Concern : Any consideration that can impact the implementation of a software program. According to [3] it could represent "anything a stakeholder may want to consider as a conceptual unit, including features, nonfunctional requirements, design idioms, and implementation mechanisms". Concerns can exist at many conceptual levels and do not fit neatly within a single dominant decomposition.

In this work we model a concern using a topic, and we have used topic and concern interchangbly.

Concern Localization : The process of locating the structural or syntactic program units (modules or statements) that implement a given concern. Some program concerns may be *modular* (i.e. implemented by a single file or module) while others may be *cross-cutting*, i.e. dispersed (scattered) across several code modules and interspersed (tangled) with other concerns.

Diffused Concern : A cross-cutting concern that does not have a prominent presence in any available source module.

Rare Concern : A diffused concern which is present in small number of files.

2.3 Problem statement

In this chapter, we are interested in the automatic discovery and statement level localization of latent concerns from unfamiliar source code, with the ability to distinguish statements that implement different concerns even if they appear in consecutive lines of code within the same module. The model must work without any apriori knowledge or human input and should be able to surface diverse concerns including diffused concerns.

If $\mathcal{P}$ is a software project which consists of M modules, $\mathcal{P} = \{D_1, \ldots, D_M\}$, where a module consists of N statements, $D_j = \{S_{j1}, \ldots, S_{jN}\}$. Using this notation, we precisely define our problem below.

Objective: Find $\mathbf{f}$ such that,

$$\mathbf{f} : \mathcal{P} \to (\mathcal{C}^m, \mathcal{C}^c, \mathbf{y})$$

where $\mathcal{C}^m$ is the set of modular concerns and $\mathcal{C}^c$ is the set of cross-cutting concerns. $\mathcal{C}^m$ captures the prominent concerns and $\mathcal{C}^c$ captures rare and diffused concerns. $\mathbf{y} = \{\mathbf{y}_{11}, \ldots, \mathbf{y}_{MN}\}$ where $\mathbf{y}_{ij} \in$ the power set of $\mathcal{C}^m \cup \mathcal{C}^c$, is the mixture of concerns assigned to S_{ij}, statement j in module i. This captures the observation that a statement can reflect multiple concerns.

2.4 Evaluation criteria

Given the highly subjective nature of concerns, there is no one true representation of concern assignments for an application. We select the following criteria for evaluating the effectiveness of different models:

Criteria: Even if a model throws up several incoherent concern topics, we will consider it to be effective (for an exploratory purpose) as long as it can *surface diverse concerns (including diffused ones) and their relevant statements as interpreted by a human programmer*. More precisely,

Let $\mathbf{I}$ be the set of valid interpretations $\mathbf{f^A}$ of the representation of actual program concerns $\mathbf{A}^m$ and $\mathbf{A}^c$ and their statement-wise assignment $\mathbf{y^A} = \{y^A{}_{11}, \ldots, y^A{}_{MN}\}$, according to human judgment.

$$I = \{\mathbf{f^A} : \mathcal{P} \to \left(\mathbf{A}^m, \mathbf{A}^c, \mathbf{y^A}\right)\}$$

Then, the effectiveness of the model can be assessed in terms of the following criteria.

1. $\exists \mathbf{f^A} \in \mathbf{I}$ for which $|\mathcal{C}^m \cap \mathbf{A}^m|$ and $|\mathcal{C}^c \cap \mathbf{A}^c|$ is high (Interpretability of $\mathcal{C}^m$ and $\mathcal{C}^c$)
2. $\exists \mathbf{f^A} \in \mathbf{I}$ for which divergence among the concerns in $\mathcal{C}^m \cap \mathbf{A}^m$ and $\mathcal{C}^c \cap \mathbf{A}^c$ is high (Diversity of $\mathcal{C}^m$ and $\mathcal{C}^c$)
3. $\exists \mathbf{f^A} \in \mathbf{I}$ for which overlap between $\mathbf{y}$ and $\mathbf{y^A}$ is high (Interpretability of $\mathbf{y}$)

It is, however, unrealistic that $\mathbf{I}$ can be determined in practice. Hence it is difficult to create test datasets for evaluating the above criteria. However, a systematic evaluation is still possible, if we make the more reasonable assumption that humans with programming or application domain knowledge can judge whether a given concern $c \in C^m \cup C^c$ or whether a particular concern assignment $\mathbf{y_{ij}}$ is likely to be consistent with some valid interpretation $\mathbf{f_A} \in \mathbf{I}$. In this scenario, we propose the following evaluation methodology:

- Create a list of concerns $\mathbf{a}$ which are known to be present under some particular interpretation $\mathbf{f^A}$ for a test project, and check if $\mathbf{a} \in \mathcal{C}^m \cup \mathcal{C}^c$ (criterion 1).
- Measure divergence of $\mathcal{C}^m \cup \mathcal{C}^c$ (criterion 2).
- Design a programmer interpretation study to quantify whether samples from $\mathcal{C}^m$, $\mathcal{C}^c$ and y are consistent with some $\mathbf{f^A} \in \mathbf{I}$ as judged by several humans with programming domain knowledge (criteria 1 and 3).

In order to test for detection of diffused concerns we design two more tests:

- Inject a diffused external concern $\mathbf{w}$ into the test project, and check if $\mathbf{w} \in \mathcal{C}^c$.
- Prune source files in the test project where a known concern $\mathbf{w}$ is prominent, so that it becomes diffused, and check if $\mathbf{w} \in \mathcal{C}^c$.

3 Assessing existing models

Several researchers have observed that the notion of a concern in software is very similar to that of a topic in documents – this has led to the adoption of topic models such as LDA for mining software concerns from source code [7, 8, 9, 10]. Baldi et. al. proposed that the concept of concerns and topics should be unified by definition – according to them, "a concern is a latent topic" [8]. Hence, concerns are modeled exactly as topics, i.e. as multinomial distributions over words present in source code text, which can be estimated by running the LDA inference algorithm on a collection of (suitably pre-processed) source code files. The document-wise (concern) topic proportions inferred by the model reflect the concern assignments, the proportions in which concerns are manifested in each source file (or method).

In this section, we discuss our experiences in applying state-of-the-art statistical topic models to our problem, automatic latent concern discovery at statement granularity, a much finer granularity than has been attempted by prior work.

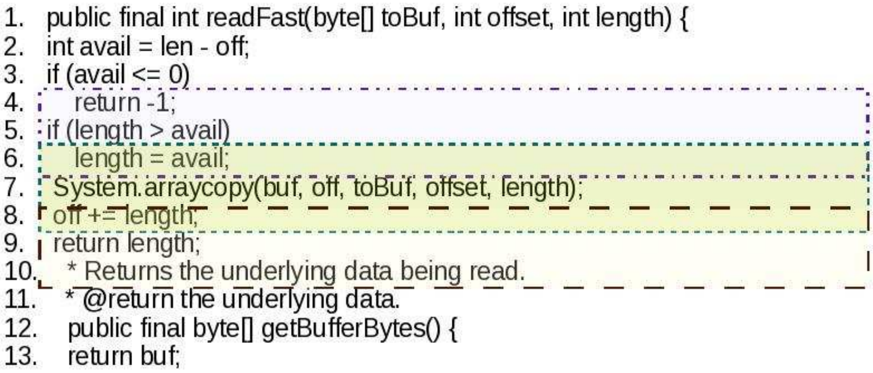
3.1 FINDING 1: LDA cannot detect diffused concerns

Although LDA enables us to discover concerns automatically from software code [8], we find that LDA based models have two major drawbacks when it comes to finding concerns in individual code statements.

First, these models do not localize the concerns at a very fine level of granularity such as code statements. One can try to treat each statement as a single entity and apply LDA based topic modeling but it is difficult to understand the usage of a statement in isolation without looking at the surrounding statements and containing module for context. We mitigate this issue by adopting a two step approach in LDA-CS, our adaptation of the prevalent LDA based methodology. We estimate concern topics by treating each source file as a document and just modify the subsequent inference procedure to treat each statement as a separate document to assign concerns to statements.

Second, we find that LDA works well in locating prominent concerns, where a prominent concern refers to a concern (either modular or cross-cutting) that has a prominent presence in at least one source module. However LDA is unable to detect diffused concerns. We confirmed this by conducting a controlled experiment that introduces a diffused concern by injecting few (5) statements corresponding to a foreign concern into files belonging to a project (Please refer to section 7.2 for full details). We observed that LDA could not detect the concern even when the concern was introduced in all files with more than 100 lines (which covered approximately 50% of the total number of files).

To address this issue, we explored the use of an alternative model that can estimate topics at a finer location granularity than an entire document.



```
1.  public final int readFast(byte[] toBuf, int offset, int length) {
2.  int avail = len - off;
3.  if (avail <= 0)
4.      return -1;
5.  if (length > avail)
6.      length = avail;
7.  System.arraycopy(buf, off, toBuf, offset, length);
8.  off += length;
9.  return length;
10.    * Returns the underlying data being read.
11.    * @return the underlying data.
12.    public final byte[] getBufferBytes() {
13.    return buf;
```

Figure 1: Example of 3 contexts, each as a set of 3 consecutive statements in a sliding window mechanism. Statement 8 or statement 9 alone may not be clearly identified as a `copy` or `buffering` concern, but along with statement 7 it becomes more apparent.

3.2 FINDING 2: MG-LDA is ineffective for source code

Similar challenges have been considered by a specialized model called MG-LDA [2], originally developed for extracting ratable aspects [2] from online user reviews. In addition to topics which have a global presence in some files (called global topics), MG-LDA models topics which only occur across small text fragments in many files (called local topics). Hence we decided to investigate whether MG-LDA could be applied to detect diffused concerns in software.

However, our controlled experiment showed that even MG-LDA fails to detect the diffused concern. This happens even when the concern is present in 50% of the files, i.e. in all files with more than 100 lines. On a closer analysis, our experience with MG-LDA on software datasets uncovered two major issues which make MG-LDA unsuitable for our problem:

First, we find that a concern can only be detected as a local topic by MG-LDA, if it is present in very large percentage of files. Unlike ratable aspects in online review data which are present widely across reviews and hence appropriately modelled as local topics by MG-LDA, cross-cutting concerns are restricted to a smaller percentage of files, which makes it difficult for MG-LDA to detect these concerns.

Second, many concerns in software packages have a typical dual presence - *these concerns are used across several files (a cross-cutting presence) and defined in a separate file (a prominent modular presence).* As the cross-cutting presence of these concerns is very weak statistically, the modular presence makes these concerns appear as global topics in MG-LDA instead of local topics. As they are detected as global topics, MG-LDA now fails to localize these concerns effectively at the statement level.

In the next section we describe a new statistical topic model specifically designed to address these challenges.

4 Context Sensitive Topic Model (CSTM)

We now propose a novel statistical model, called a context sensitive rare topic model(CSTM), which without assuming any human input is not only able to discover prominent, diffused and rare topics/concerns, but is also able to localize them at a statement level.

Our model assumes that spatially co-located statements may give us a *context* to understand the underlying topic/concern of a statement. Hence, CSTM does not treat a statement alone, but models it in a context of neighborhood statements. We precisely define a context as a set of T contiguous statements in a file. As there is no physical boundary between contexts, we build a sliding window of contexts (Figure 1), where the first context window in a file contains only the first line of the file and each line belongs to multiple overlapping context

[2] e.g. location, comfort, food, cleanliness and pricing could be ratable aspects in reviews of hotels

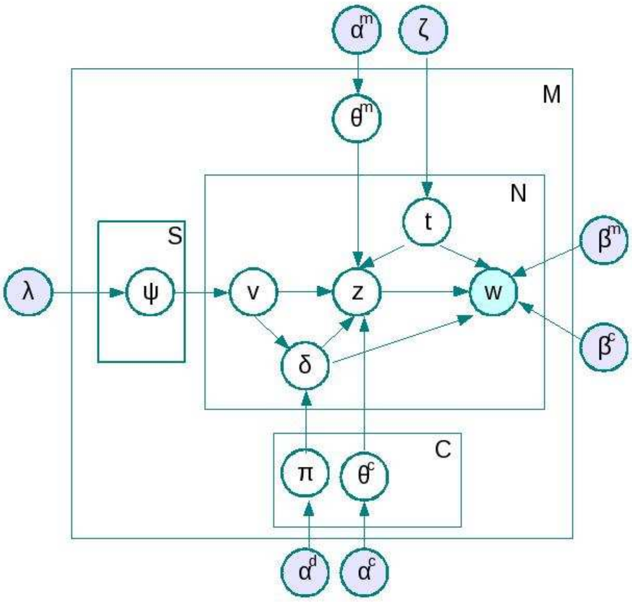


Figure 2: Graphical model representation of CSTM. The shaded circles represent inputs and outputs of the model. The rectangles marked with M, C, S, N imply iteration over M modules, C contexts, S statements and N words. A directed arrow means that the "from" node influences the "to" node. w is the only observed variable here representing words in software code, while z represents the concerns. The suffixes "m" and "c" in the parameters distinguish modules and contexts respectively.

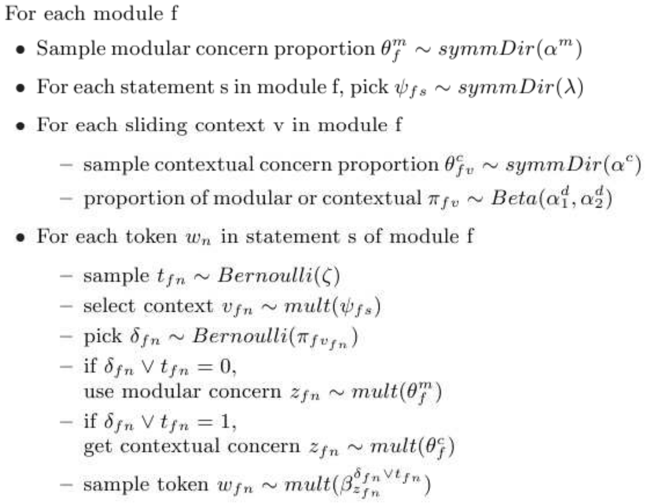
windows. Thus, if there are S number of statements in a file, there are $T + S - 1$ number of context windows in that file. Finally, each context is modeled as a distribution over the concerns.

CSTM assumes two levels of abstraction, one at the file level and other at the context level. Thus, in addition to topics that manifest at the context level which we call contexual topics, there are file level topics which we call a modular topics. Modular topics mainly captures *prominent* topics whereas contextual topics capture *diffused* and *rare* topics.

A statement contributes to both the modular topics and the contextual topics in a proportionate manner. CSTM provides a configuration parameter to allow control over this proportion to adjust the context sensitivity of the model. If a topic is rare so that it does not appear at a file level, it will remain undetected by models like LDA but can be detected by CSTM, even if it is confined to the locality of few statements and present in only a relatively small proportion of files.

The details of the model are described in terms of a graphical model representation (Figure 2) as well as a generative process (Figure 3). We find experimentally that an asymmetric prior over π alone is not sufficient to detect rare topics because the posterior inference of π strongly depends on the data and little on the prior emphasis. Our model is equipped with an external influence through t, which can help increase bias towards contextual concerns more effectively. ζ is the Bernoulli parameter of this external control variable t. β^m and β^c are parameter matrices of sizes $K^m \times V$ and $K^c \times V$ respectively; the matrices represent the *concern topic-word distributions* of the K^m modular and K^c contextual topics $\mathcal{C}^m$ and $\mathcal{C}^c$ found by the model (where V is the total number of distinct words and K^m and K^c are specified by the user). β^m_{ij} is probability of picking word j given that z is the i^{th} modular topic and β^c_{ij} is probability of picking word j given that z is the i^{th} contextual topic. A topic or concern can be described by the most probable words in the distribution.

Modeling context sensitivity Using the parameter ζ it is possible to introduce external control over the level of intensity towards discovering topics at the context level. δ from data and t from the external configuration setting together control the level of context sensitivity of the model. So, we need a function $\mathcal{F}(\delta, t) \in \{0, 1\}$ such



For each module f

- Sample modular concern proportion $\theta_f^m \sim symmDir(\alpha^m)$
- For each statement s in module f, pick $\psi_{fs} \sim symmDir(\lambda)$
- For each sliding context v in module f
    - sample contextual concern proportion $\theta_{fv}^c \sim symmDir(\alpha^c)$
    - proportion of modular or contextual $\pi_{fv} \sim Beta(\alpha_1^d, \alpha_2^d)$
- For each token w_n in statement s of module f
    - sample $t_{fn} \sim Bernoulli(\zeta)$
    - select context $v_{fn} \sim mult(\psi_{fs})$
    - pick $\delta_{fn} \sim Bernoulli(\pi_{fv_{fn}})$
    - if $\delta_{fn} \vee t_{fn} = 0$,
      use modular concern $z_{fn} \sim mult(\theta_f^m)$
    - if $\delta_{fn} \vee t_{fn} = 1$,
      get contextual concern $z_{fn} \sim mult(\theta_f^c)$
    - sample token $w_{fn} \sim mult(\beta_{z_{fn}}^{\delta_{fn} \vee t_{fn}})$

Figure 3: Generative process of CSTM model for each file f

$\mathcal{F}$	ζ	**Effect**	**Existing Models**	**Topic coverage**
δ OR t	0	ignores t (user)	MG-LDA	prominent & diffused
δ OR t	> 0.5	more contextual	-	prominent & diffused & rare
δ OR t	1	always contextual	-	diffused & rare
δ AND t	1	ignores t (user)	MG-LDA	prominent & diffused
δ AND t	< 0.5	more modular	-	prominent & diffused
δ AND t	0	always modular	LDA	prominent

Table 1: Choice of $\mathcal{F}, \zeta$ and its implications. If $\mathcal{F}$ is 1, contextual topics are selected, else modular topics are selected.

that if $\mathcal{F}$ is 1 then CSTM will choose contextual topics, otherwise modular topics. Varying ζ, and using various possible definitions of $\mathcal{F}$, we introduce a lot of flexibility in the model. We highlight some interesting special cases in Table 1.

Inferring concerns: We have used the *variational inference* EM method to infer the concern or topics and document wise topic proportions based on the joint distribution of the model described by the generative process in Fig 3. The detailed inference procedure is described in the section 5.

Localizing Topics to Statements We could assign topics or concerns to statements in two ways. The naive way is to treat each statement as a module and infer the posterior distribution over the concerns for each statement. Instead, our model enables us to utilize the context and obtain topic proportions at both module and context level. Using posterior estimation of θ^m and θ^c we can deduce u_{ijk}^c and u_{ijk}^m, the proportions of the k^{th} contextual or modular topic respectively for statement S_{ij}. We concatenate the modular and contextual topic proportions to get u_{ij}, and then from u_{ij}, we assign topics which have a high contribution to the statement as follows:

$$\mathbf{y}_{ij} = \{k \mid u_{ijk} > \text{threshold}\} \tag{1}$$

4.1 Boosting diversity among topics

We have observed that if the specified number of topics in the model is increased to a large number with the intent of locating other topics, in many cases, instead of detecting new topics, topic models repeats topics with slight variations, while many topics remain un-surfaced.

Following [11], we introduce an asymmetric Dirichlet prior on statement-topic distribution. In addition, we have tried to increase the gap between the concerns using a novel mechanism. After the estimation of the concerns are done, we update the topics as follows.

$$\beta^c_{ij} = \beta^c_{ij} \prod_{l \neq i} (1 - \beta^c_{lj})$$

If a word has high probability in any topic, then this will reduce its probability in other topics, whereas if a word has low probability in almost all the topics, it will increase the probability in one of the topics. Thus, we increase the diversity in detecting topics which in turn helps surface relatively rare topics more effectively. However, the divergence is achieved at the price of coherence of concern topics.

5 Inference

In this section we present the inference of our model CSTM using variational inference [12]. The likelihood of the dataset following the model as given in Figure 3 is as below.

$$\begin{aligned} & p(D|\alpha^c, \alpha^m, \alpha^d, \lambda, \zeta, \beta^c, \beta^m) \\ = & \int_{\theta^c} \int_{\theta^m} \int_{\pi} \sum_z \sum_\delta \sum_t p(w|z, v, t, \beta^c, \beta^m) p(z|v, \delta, t, \theta^c, \theta^m) \\ & p(t|\zeta) p(v|\psi) p(\psi|\lambda) p(\theta^c|\alpha^c) p(\theta^m|\alpha^m) p(\pi|\alpha^d) \end{aligned} \tag{2}$$

Using the maximum likelihood method we can find out the values of the parameters. However, due to coupling between several variables the problem of computing $p(D)$ is intractable. There are two most popular alternative methods available in this kind of problems based on variational inference and sampling. In this paper we propose a variational inference mechanism to solve the model. We define a variational distribution q as below.

$$\begin{aligned} q(z^c, z^m, \delta, t, v, pi, \psi) \quad = \quad & \prod_{s=1}^{S} q(\psi_s|\epsilon_s) q(\theta^c|\gamma^c) q(\theta^m|\gamma^m) \\ \prod_{n=1}^{N} [q(z^c_n|\phi^c_n) \quad & q(z^m_n|\phi^m_n,) \quad q(v_n|\sigma_n) q(t_n \rho_n) q(\delta_n|\xi_n)] \end{aligned} \tag{3}$$

where ϵ_s(Dirichlet), γ^c(Dirichlet), γ^m(Dirichlet), ϕ^c_n(multinomial), ϕ^m_n(multinomial) σ_n(multinomial) ρ_n(Bernoulli) and ξ_n(Bernoulli) are the variational parameters.

If we denote the latent variables by L, then

$$\frac{p(D, L)}{q(L)} = \frac{p(L|D)}{q(L)} p(D) \tag{4}$$

After taking expectaion of the log of the above expression with respect to q, we get

$$\begin{aligned} \log p(D) &= E_q[\log \frac{p(D, L)}{q(L)}] - E_q[\log \frac{p(L|D)}{q(L)}] \\ &= \mathcal{L} + KL(q(L)\|p(L|D)) \end{aligned} \tag{5}$$

$KL(q(L)\|p(L|D))$ is the Kullback-Leibler divergence between the variational distribution over the latent variables and the conditional distribution of the latent variables given the observation. As KL-divergence is always non-negative $\mathcal{L}$ gives a lower bound on the log-likelihood of the observation. However equality holds, when $q(L) = p(L|D)$. If the form of the distribution of q is defined using the conjugacy property in the model, optimizing $\mathcal{L}$ with respect to the variational parameters will give us q, which will be equal to $p(L|D)$. Using that q, we can obtain the model parameters, by maximizing $\mathcal{L}$ with respect to the model parameters.

Now,

$$\begin{aligned}
\mathcal{L} &= E_q[\log p(w|z,\delta,t) + \log p(z|v,\delta,t,\theta) + \log p(\delta|\pi,v) + \log p(t) \\
&\quad + \log p(\pi) + \log p(v|\psi) + \log p(\psi) + \log p(\theta^l) + \log p(\theta^g)] \\
&- E_q[\log q(z) + \log q(\delta) + \log q(t) + \log q(\psi) + \log q(v) + \log q(\pi) \\
&\quad + \log q(\theta^g) + \log q(\theta^l)] \qquad (6)
\end{aligned}$$

$$\begin{aligned}
\log p(w) &\geq \mathcal{L} \\
\mathcal{L} &= E_q[\log p(w,t,z,\delta,v,\psi,\pi,\theta^g,\theta^l)] - E_q[\log q(t,z,\delta,v,\psi,\pi,\theta^g,\theta^l)] \qquad (7)
\end{aligned}$$

Let $\mathcal{L}_p = E_q[\log p(w,t,z,\delta,v,\psi,\pi,\theta^g,\theta^l)]$ and $\mathcal{L}_q = E_q[\log q(t,z,\delta,v,\psi,\pi,\theta^g,\theta^l)]$

For detailed derivation please see Appendix at the end. We put the expected terms in the next section followed by the update rules.

5.1 Terms in $\mathcal{L}_p$

$$\begin{aligned}
E_q[\log p(\pi|\alpha^m)] &= \sum_{i=1}^{T+S-1} [\ \log\Gamma(\sum_{k=0}^{1}\alpha_k^m) - \sum_{k=0}^{1}\log\Gamma(\alpha_k^m) + \sum_{k=0}^{1}(\alpha_k^m - 1)(\Psi(\gamma_{ik}^m) - \Psi(\sum_{j=0}^{1}\gamma_{ij}^m))\ ] \\
E_q[\log p(\theta^g|\alpha^g)] &= \log\Gamma(\sum_{i=1}^{K^g}\alpha_i^g) - \sum_{i=1}^{K^g}\log\Gamma(\alpha_i^g) + \sum_{i=1}^{K^g}(\alpha_i^g - 1)(\Psi(\gamma_i^g) - \Psi(\sum_{j=1}^{K^g}\gamma_j^g)) \\
E_q[\log p(\theta^l|\alpha^l)] &= \sum_{i=1}^{T+S-1} [\ \log\Gamma(\sum_{j=1}^{K^l}\alpha_j^l) - \sum_{k=1}^{K^l}\log\Gamma(\alpha_k^l) + \sum_{k=1}^{K^l}(\alpha_k^l - 1)(\Psi(\gamma_{ik}^l) - \Psi(\sum_{j=1}^{K^l}\gamma_{ij}^l))\ ] \\
E_q[\log p(\psi|\lambda)] &= \sum_{i=1}^{S}[\ \log\Gamma(\sum_{j=1}^{T+S-1}\lambda_j) - \sum_{k=1}^{T+S-1}\log\Gamma(\lambda_k)\ + \sum_{k=1}^{T+S-1}(\lambda_k - 1)(\Psi(\epsilon_{ik}) - \Psi(\sum_{j=1}^{T+S-1}\epsilon_{ij}))\ ] \\
E_q[\log p(v|\psi)] &= \sum_{n=1}^{N}\sum_{i=1}^{T+S-1}\sigma_{ni}(\Psi(\epsilon_{ni}) - \Psi(\sum_{j=1}^{T+S-1}\epsilon_{nj})) \\
E_q[\log p(\delta|\pi,v)] &= \sum_{n=1}^{N}\sum_{k=0}^{1}\sum_{i=1}^{T+S-1}\xi_{nk}\sigma_{ni}(\Psi(\gamma_{ik}^m) - \Psi(\sum_{j=0}^{1}\gamma_{ij}^m)) \\
E_q[\log p(z|t,v,\delta,\theta)] &= \sum_{n=1}^{N}[\ \sum_{k=1}^{K^g}\phi_{nk}^g\xi_{n0}\rho_{n0}(\Psi(\gamma_k^g) - \Psi(\sum_{j=1}^{K^g}\gamma_j^g))\ + \sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1}\phi_{nk}^l\sigma_{nj}\xi_{n1}\rho_{n0}(\Psi(\gamma_{jk}^l) - \Psi(\sum_{h=1}^{K^l}\gamma_{jh}^l))\ ] \\
&+ \sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1}\phi_{nk}^l\sigma_{nj}\xi_{n1}\rho_{n1}(\Psi(\gamma_{jk}^l) - \Psi(\sum_{h=1}^{K^l}\gamma_{jh}^l))\ ]\ + \sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1}\phi_{nk}^l\sigma_{nj}\xi_{n0}\rho_{n1}(\Psi(\gamma_{jk}^l) - \Psi(\sum_{h=1}^{K^l}\gamma_{jh}^l)) \\
E_q[\log p(w|t,z,\delta,\beta)] &= \sum_{n=1}^{N}[\ \sum_{k=1}^{K^g}\sum_{j=1}^{V}\phi_{nk}^g\xi_{n0}\rho_{n0}\log\beta_{kj}^g I_{w_n=j}\ + \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi_{nk}^l\xi_{n1}\rho_{n0}\log\beta_{kj}^l I_{w_n=j}\ ] \\
&+ \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi_{nk}^l\xi_{n1}\rho_{n1}\log\beta_{kj}^l I_{w_n=j}\ + \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi_{nk}^l\xi_{n0}\rho_{n1}\log\beta_{kj}^l I_{w_n=j}\ ] \\
E_q[\log p(t)] &= \sum_{n=1}^{N}\sum_{k=0}^{1}\rho_{nk}\log\zeta_k
\end{aligned}$$

(8

5.2 Terms in $\mathcal{L}_q$

$$\begin{aligned}
E_q[\log q(\theta^g)] &= \sum_{i=1}^{K^g}(\gamma_i^g - 1)(\Psi(\gamma_i^g) - \Psi(\sum_{j=1}^{K^g}\gamma_j^g)) + \log\Gamma(\sum_{i=1}^{K^g}\gamma_i^g) - \sum_{i=1}^{K^g}\log\Gamma(\gamma_i^g) \\
E_q[\log q(\theta^l)] &= \sum_{i=1}^{T+S-1}[\,\sum_{k=1}^{K^l}(\gamma_{ik}^l - 1)(\Psi(\gamma_{ik}^l) - \Psi(\sum_{j=1}^{K^l}\gamma_{ij}^l)) + \log\Gamma(\sum_{j=1}^{K^l}\gamma_{ij}^l) - \sum_{k=1}^{K^l}\log\Gamma(\gamma_{ik}^l)\,] \\
E_q[\log q(\pi)] &= \sum_{i=1}^{T+S-1}[\,\sum_{k=0}^{1}(\gamma_{ik}^m - 1)(\Psi(\gamma_{ik}^m) - \Psi(\sum_{j=0}^{1}\gamma_{ij}^m)) + \log\Gamma(\sum_{k=0}^{1}\gamma_{ik}^m) - \sum_{j=0}^{1}\log\Gamma(\gamma_{ij}^m)\,] \\
E_q[\log q(\psi)] &= \sum_{i=1}^{S}[\sum_{k=1}^{T+S-1}(\epsilon_{ik} - 1)(\Psi(\epsilon_{ik}) - \Psi(\sum_{j=1}^{T+S-1}\epsilon_{ij})) + \log\Gamma(\sum_{j=1}^{T+S-1}\epsilon_{ij}) - \sum_{k=1}^{T+S-1}\log\Gamma(\epsilon_{ik})] \\
E_q[\log q(z|\delta = g)] &= \sum_{n=1}^{N}\sum_{i=1}^{K^g}\phi_{ni}^g\log\phi_{ni}^g \\
E_q[\log q(z|\delta = l)] &= \sum_{n=1}^{N}\sum_{i=1}^{K^l}\phi_{ni}^l\log\phi_{ni}^l \\
E_q[\log q(\delta|\xi)] &= \sum_{n=1}^{N}\xi_n\log\xi_n + (1 - \xi_n)\log(1 - \xi_n) \\
E_q[\log q(v|\sigma)] &= \sum_{n=1}^{N}\sum_{i=1}^{T+S-1}\sigma_{ni}\log\sigma_{ni} \\
E_q[\log q(t)] &= \sum_{n=1}^{N}\sum_{k=0}^{1}\rho_{nk}\log\rho_{nk}
\end{aligned} \tag{9}$$

5.3 Update Rules

5.3.1 Update Rules in E step

$$
\begin{aligned}
\phi^g_{nk} &\propto \beta^{g\ (\xi_{n0}\rho_{n0})}_{kj} \exp((\xi_{n0}\rho_{n0})(\Psi(\gamma^g_k) - \Psi(\sum_{j=1}^{K^g} \gamma^g_j))) \\
\phi^l_{nk} &\propto (\beta^{l\ (\xi_{n0}\rho_{n1}+\xi_{n1}\rho_{n1}+\xi_{n1}\rho_{n0})}_{kg} I_{w_n=g}) \exp(\sum_{j=1}^{T+S-1} \sigma_{nj}(\xi_{n0}\rho_{n1} + \xi_{n1}\rho_{n1} + \xi_{n1}\rho_{n0})(\Psi(\gamma^l_{jk}) - \Psi(\sum_{h=1}^{K^l} \gamma^l_{jh})) \\
\sigma_{ni} &\propto \exp(\sum_{h=0}^{1} \xi_{nh}(\Psi(\gamma^m_{ih}) - \Psi(\sum_{j=0}^{1} \gamma^m_{ij})) + \sum_{k=1}^{K^l} \phi^l_{nk}(\xi_{n0}\rho_{n1} + \xi_{n1}\rho_{n1} + \xi_{n1}\rho_{n0})(\Psi(\gamma^l_{ik}) - \Psi(\sum_{h=1}^{K^l} \gamma^l_{ih})) + \Psi(\epsilon_{ni}) - \Psi(\sum_{j=1}^{T+S-1} \\
\epsilon_{ik} &= \lambda_k + \sum_{n=1}^{N} \sigma_{s_n k} I_{s_n=i} \\
\xi_{n0} &\propto \exp(\sum_{k=1}^{K^g} \phi^g_{nk}\rho_{n0}(\Psi(\gamma^g_k) - \Psi(\sum_{j=1}^{K^g} \gamma^g_j)) + \sum_{k=1}^{K^l} \sum_{j=1}^{T+S-1} \phi^l_{nk}\sigma_{nj}\rho_{n1}(\Psi(\gamma^l_{jk}) - \Psi(\sum_{h=1}^{K^l} \gamma^l_{jh})) \\
&+ \sum_{k=1}^{K^g} \sum_{j=1}^{V} \phi^g_{nk}\rho_{n0} \log \beta^g_{kj} I_{w_n=j} + \sum_{k=1}^{K^l} \sum_{j=1}^{V} \phi^l_{nk}\rho_{n1} \log \beta^l_{kj} I_{w_n=j}) \\
\xi_{n1} &\propto \exp(\sum_{n=1}^{N} [\sum_{k=1}^{K^l} \sum_{j=1}^{T+S-1} \phi^l_{nk}\sigma_{nj}\rho_{n0}(\Psi(\gamma^l_{jk}) - \Psi(\sum_{h=1}^{K^l} \gamma^l_{jh})) \\
&+ \sum_{k=1}^{K^l} \sum_{j=1}^{T+S-1} \phi^l_{nk}\sigma_{nj}\rho_{n1}(\Psi(\gamma^l_{jk}) - \Psi(\sum_{h=1}^{K^l} \gamma^l_{jh})) + \sum_{k=1}^{K^l} \sum_{j=1}^{T+S-1} \phi^l_{nk}\sigma_{nj}\xi_{n0}\rho_{n1}(\Psi(\gamma^l_{jk}) - \Psi(\sum_{h=1}^{K^l} \gamma^l_{jh})) \\
&+ \sum_{k=1}^{K^l} \sum_{j=1}^{V} \phi^l_{nk}\xi_{n1}\rho_{n0} \log \beta^l_{kj} I_{w_n=j} \\
&+ \sum_{k=1}^{K^l} \sum_{j=1}^{V} \phi^l_{nk}\xi_{n1}\rho_{n1} \log \beta^l_{kj} I_{w_n=j} + \sum_{k=1}^{K^l} \sum_{j=1}^{V} \phi^l_{nk}\xi_{n0}\rho_{n1} \log \beta^l_{kj} I_{w_n=j}]) \\
\rho_{n0} &\propto \exp(\log \zeta_0 + \sum_{k=1}^{K^g} \phi^g_{nk}\xi_{n0}(\Psi(\gamma^g_k) - \Psi(\sum_{j=1}^{K^g} \gamma^g_j)) + \sum_{k=1}^{K^g} \sum_{j=1}^{V} \phi^g_{nk}\xi_{n0} \log \beta^g_{kj} I_{w_n=j}) \\
\rho_{n1} &\propto \exp(\log \zeta_1 + \sum_{k=1}^{K^g} \phi^g_{nk}\xi_{n0}(\Psi(\gamma^g_k) - \Psi(\sum_{j=1}^{K^g} \gamma^g_j)) + \sum_{k=1}^{K^g} \sum_{j=1}^{V} \phi^g_{nk}\xi_{n0} \log \beta^g_{kj} I_{w_n=j} \\
&\quad \sum_{k=1}^{K^l} \sum_{j=1}^{T+S-1} \phi^l_{nk}\sigma_{nj}\xi_{n1}(\Psi(\gamma^l_{jk}) - \Psi(\sum_{h=1}^{K^l} \gamma^l_{jh})) + \sum_{k=1}^{K^l} \sum_{j=1}^{V} \phi^l_{nk}\xi_{n1} \log \beta^l_{kj} I_{w_n=j}) \\
\gamma^g_i &= \alpha^g_i + \sum_{n=1}^{N} \phi^g_{ni}(\xi_{n0}\rho_{n0} + \xi_{n0}\rho_{n1}) \\
\gamma^l_{ik} &= \alpha^l_k + \sum_{n=1}^{N} \phi^l_{nk}\sigma_{ni}(\xi_{n1}\rho_{n1} + \xi_{n1}\rho_{n0}) \\
\gamma^m_{ik} &= \alpha^m_k + \sum_{n=1}^{N} \xi_{nk}\sigma_{ni}
\end{aligned}
$$

5.3.2 Update Rules for Concerns

$$\begin{aligned}
\beta^g_{ij} &\propto \sum_{d=1}^{M}\sum_{n=1}^{N} \phi^g_{dni}\xi_{dn0}\rho_{dn0}w^j_{dn} \\
\beta^l_{ij} &\propto \sum_{d=1}^{M}\sum_{n=1}^{N} \phi^l_{dni}(\xi_{dn1}\rho_{dn0} + \xi_{dn1}\rho_{dn1} + \xi_{dn0}\rho_{dn1})w^j_{dn}
\end{aligned} \tag{11}$$

6 Attributing resource usage to latent concerns

In this section, we illustrate an example of a novel application enabled by statement level concern discovery: the ability to correlate program properties that vary at statement granularity, such as its runtime resource usage, with automatically discovered latent concerns.

By jointly post-processing the output of an existing profiler and the results of our model, we can estimate the relative runtime resource consumption of latent concerns for an application or automatically discover concerns that are resource intensive (and hence potential candidates for optimization). We combine statement level resource consumption statistics with concern proportions assigned by our model to generate a concern-wise performance summary instead of summaries in terms of syntactic source code modules (e.g. methods, component packages etc). This can provide an interesting view of program performance behavior in terms of underlying functional intent, as opposed to low level implementation modules.

Since concerns can be implemented using other concerns, a richer form of summarization than a flat profile is useful. For example traditional method-wise profiling often incorporates calling context information. A calling context tree (CCT) profile (as generated by a bytecode profiler like JP2 [13]) can be converted into a *concern context tree* profile by mapping each level in the CCT to the concern proportions assigned to the corresponding statement. This can then be used to generate various summary views. For example, the cumulative bytecode execution cost attributed to a concern includes the cumulative resource usage of statements belonging to the concern and the methods invoked by those statements.

Let $R(S_{ij})$ be the resource usage (e.g. bytecodes executed) of statement S_{ij} (j^{th} statement of i^{th} module).

Attributing costs in accordance with concern proportions when computing flat profiles is relatively straightforward.

Estimated flat resource usage (bytecode execution) cost of the k^{th} concern

$$R_k = \sum_{ij} u_{ijk} * R(S_{ij}) \tag{12}$$

Accounting for statement-wise concern proportions when computing the cumulative resource usage of a concern is more tricky. A CCT node and each of its descendants may be assigned different concern proportions. If a CCT node is assigned only to the k^{th} concern, then the entire cumulative cost of the node should be attributed to the k^{th} concern. On the other hand, if the node is not assigned the k^{th} concern, then we should recursively proceed to apply the same logic to its child nodes. Thus, with probability u_{ijk}, we assign the cumulative cost of S_{ij} to the k^{th} concern and with probability $1 - u_{ijk}$ we examine its child nodes. The formula is more simply expressed in a bottom up fashion in terms of ancestor relationships by noting that the cost of a node S_{ij} should only be attributed to the k^{th} concern if S_{ij} or any ancestor node in its call chain is assigned to the k^{th} concern.

Estimated cumulative resource usage (bytecode execution) cost of the k^{th} concern

$$R^{cum}_k = \sum_{ij} R(S_{ij})(1 - \prod_{S_{pq}\in\hat{S}_{ij}} (1 - u_{pqk})) \tag{13}$$

where $\hat{S}_{ij} = (S_{ij} \cup ancestor(S_{ij}))$

These estimates are approximate, given the statistical nature of the model and statement level concern assignments.

Concern label	Concern topic words	Runtime resource usage % bytecodes executed	
		lusearch	luindex
SEARCH	hits searcher score search docs	15%	0%
QUERY	query parse phrase queries multi	39%	0%
WRITE (INDEX)	write flush optimize characters reopen	3%	46%
STEMMING	stemmer stopwords snowball zip net word letter hyphenation pattern character	1%	3%
TOKEN BUFFER	arraycopy begin end buffersize bufpos	3%	0%
EXPLAIN	weight explanation score expl val	45%	0%
TIMING	date time tools resolution cal	2.6%	0%
READER	read input offset seek pos	23%	16%

Table 2: Byte-code execution summaries computed for sample Apache lucene concerns found by CSTM. Results are shown for two benchmarks DaCapo lusearch and DaCapo luindex. It reports cumulative bytecode execution cost attributed to a concern as a percentage of total bytecodes executed by the program. Shaded rows are examples of diffused concerns and they are undetected by LDA-CS.

Table 2 shows results from computing such a concern-wise bytecode execution summary for two benchmarks from the DaCapo suite [14], `lusearch` and `luindex`, which are both based on Apache lucene. It reports the cumulative bytecode execution cost attributed to sample concerns discovered in Apache lucene by our model CSTM. We list the top 5 words of a concern topic and assign a label to the concern for ease of interpretation. *The entire process of generating the summary is fully automatic* (except the choice of labels for concern topics of interest).

Note the differences in the profile for the two benchmarks.

The `SEARCH` and `QUERY` related concerns, including `EXPLAIN`, have a high bytecode execution cost when running `lusearch`, but are hardly exercised when running `luindex`. On the other hand the `WRITE` concern contributes to a significant percentage of bytecodes executed when running `luindex`. Some other concerns such as `READER` affect the execution cost of both benchmarks. As per the DaCapo benchmark descriptions, `luindex` uses lucene to index a set of documents while `lusearch` uses lucene to perform a text search of keywords over a corpus of data. Thus the results are intuitive. The `TIMING` concern is used for timing search queries (e.g. to timeout queries that might be taking too long), hence relevant for `lusearch`. The `STEMMING` concern is used when indexing words and also when parsing queries.

Some *key concerns that account for the resource usage differences, e.g.* `WRITE` *and* `TOKEN BUFFER` *are diffused concerns*. LDA was unable to detect these concerns. This highlights the importance of modeling statement context.

7 Empirical Evaluation

In this section, we present findings from an empirical evaluation of the models. We use our models to analyze four different Java applications (Table 3). We evaluate the results using experiments that expose differences in concern detection sensitivity and coverage (diversity) of the models, and a human programmer evaluation study of the interpretability of statement level concern assignments.

Parameter settings: The model parameters are selected uniformly for all the tests and models. In the experiments reported here we specified 100 concerns for each application. When using CSTM these were divided into 50 modular concerns and 50 cross-cutting contextual concerns. We have used 3 as the size of the context window, that is a sentence can belong to 3 context windows (previous, current, next). We set $\zeta = 0.9$ and specified $\mathcal{F}$ as δ OR t.

Regarding setting up the other parameters α^c, α^d, α^m and λ, note that, these parameters are estimated in our model as described earlier and in Appendix. To set up the initial value of the parameters we have followed grid search by varying them from 0.01 to 1 in multiples of 10, and manually inspected the concerns detected by

Application	**Description**	**Files**	**Lines**	**Vocabulary**
BerkeleyDB	Embedded database engine	238	38954	2733
Apache lucene	Text search	958	114228	7869
SPECjbb2005	Server-side Java benchmark	63	9723	1444
DaCapo BLOAT	Bytecode-Level optimizer	188	36843	2553

Table 3: Four Java applications of various scales and domains used in our experiments.

MG-LDA on BerkeleyDB if it detects major known concerns. The values we have used across the projects are – $\alpha^c, \alpha^d, \alpha^m$ as 0.01, and $\lambda = 0.1$. The same set of values have been used in CSTM too. Following [11] we have used asymmetric prior over the concern proportions, where we have used $\alpha_i^c = (10^i) * \alpha^c$ and $\alpha_i^m = (10^i) * \alpha^m$, and values of α^c, α^m being 0.001. [11] have nice discussions on parameters of topic models, which is planned to be explored for CSTM in future.

Pre-processing: We consider only the textual part of the source code, any syntactical elements have been removed. We have not used any linguistic tools like stemmer or parts-of-speech tagger, but only removed a set of standard English stop words[3] excluding few Java specific words like get, set etc. We have also removed Java key-words[4]. The full list of stop words have been listed in the Appendix. Tokens like StringCopy have been split into two words String and Copy based on the position of a capital face inside a token, and all uses of capital face have been converted to small face.

7.1 Key evaluation criteria

Our evaluation is designed to assess the selected models according to the following criteria:

1. **Concern detection sensitivity:** *Can the method surface diffused concerns?* (Section 7.2).
2. **Concern coverage diversity:** *Does the method surface a diverse set of concerns ?* (Section 7.3).
3. **Interpretability of concern assignments**: *Does the method assign concerns to relevant statements with a meaningful interpretation ?* (Section 7.4).

7.2 Concern detection sensitivity

Criterion: *Can the method surface diffused concerns?*

In these experiments we use a single application, BerkeleyDB for which a set of known concerns are available from a published manual analysis [15]. Most of these identified concerns have both a modular component (e.g. a key class or interface) and cross-cutting statements. To expose differences in model sensitivity to diffused concerns we design the following tests:

7.2.1 Inject a foreign diffused concern

In this test we insert a few (5) statements corresponding to a foreign concern (we used graphics/color related 44 statements from JHotDraw as the foreign concern) at random positions into some randomly chosen BerkeleyDB source files. The algorithm is described in Figure 7.2.1.

We run our models on this modified source dataset and check whether the foreign concern is detected. We observe that CSTM is able to detect the concern when the number of altered files is only 10% of total number of files, whereas LDA fails to detect the concern even when the number of modified files includes all files with more than 100 lines (which covers around 50% of the total number of files) (Table 4).

[3] *http : //en.wikipedia.org/wiki/Stop_words*

[4] *http : //en.wikipedia.org/wiki/List_of_Java_keywords*

```
• $c = 0$
• For each file $f$ in source, if number of statements in $f$ is more than $thr_f$ and $c < thr_c$.
    – $flag = 0$
    – pick a random number $q$, if $q < thr_q$ skip this file.
    – Set target intrusion list $I_t$ to empty and $i = 0$.
    – For each statement $l$ in source intrusion list $I_s$
        ∗ Pick a rand number $r$
        ∗ If $r > thr_r$
            · include the statement in $I_t$
            · $i = i + 1$
        ∗ End if $i > thr_i$
    – For each statement $f_s$ in $f$ if $flag$ is 0.
        ∗ Pick a rand number $p$
        ∗ If $p > thr_p$, insert $I_t$ in $f$ after $f_s$, $flag = 1$
    – $c = c + 1$
```

Figure 4: Algorithm to inject foreign concern into source files. The file to be injected by foreign concern is random. The intruding statements are randomly chosen, even the number of them random. The position to insert the intruding statements are also random. By controlling thr_c (100 in our case) we can limit number of injected files. We need thr_f to be little high (500 in our case) so that the foreign concern does not become a prominent concern. thr_r and thr_r allows to randomly choose the intruding statement and position of injection respectively (both are 0.9 in our case), whereas thr_i limits the maximum number of foreign statements in a file (5 in our case).

7.2.2 Prune modules where a known concern is prominent

In this test, we choose a couple of known BerkeleyDB cross-cutting concerns and remove the main source files where these concerns are prominent. Now only the diffused cross-cutting statements corresponding to these concerns remain in source tree. We run our models on this pruned source dataset and check whether these two concerns are reflected in the concerns found by these models. The two concerns we chose for this experiment are ”Trace” and ”Memory Budget”.

Table 5 lists the most likely words of the relevant concerns found by the models on running these experiments. LDA on the pruned tree does not find any topics which reflect the test concerns. CSTM is able to detect both these concerns, despite the weakend and diffused presence.

7.3 Concern coverage diversity

Criterion: *Does the method surface a diverse set of concerns?*

The previous experiments focused on the sensitivity of the model with respect to surfacing potentially non-trivial and interesting concerns with a diffused presence in the source. In the next set of experiments we compare models in terms of diversity of concerns found (a representative summary should cover a broad set of concerns).

7.3.1 Topic diversity measurement

We measure concern topic diversity quantitatively for both LDA-CS and CSTM in terms of the Jenson Shannon divergence (JSD) [16] of the concern topic-word distributions of the 100 concerns found. We used the generalized definition of JSD for more than two distributions, which computes the total divergence to the average of these distributions.

Statements of foreign concern “graphics”
public class HSVColorSpace extends ColorSpace
public static HSVColorSpace getInstance()
instance = new HSVColorSpace();
super(ColorSpace.TYPEHSV, 3);
public class HSVHarmonicColorWheelImageProducer extends PolarColorWheelImageProducer

Concern	**CSTM topic**	**LDA topic**
Graphics	color, space, hsv, instance, harmonic, model, wheel, image	NONE

Table 4: Example statements corresponding to foreign concern “graphics” injected into BerkeleyDB (top). The concern is detected by CSTM, but LDA fails to detect it due to its weak presence (bottom).

Concern	**CSTM topic**	**LDA topic**
Trace	param level util cleaner trace	NONE
Memory Budget	memory delete ret match budget	NONE

Table 5: Context sensitivity experiment results: Pruned BerkeleyDB test concern topics found (5 most likely words of relevant topics are listed)

$$JSD(\beta_1, \beta_2, \ldots, \beta_K) = H(\sum_{i}^{K} \pi_i \beta_i) - \sum_{i}^{K} \pi_i H(\beta_i)$$

where $H(\beta_i)$ is the Shannon entropy for distribution β_i and we choose the weights $\pi_1 = \pi_2 = \ldots = \pi_K = \frac{1}{K}$. As the number of topics specified for both models is the same, a higher value of JSD indicates a more diverse set of topics. From our results, we observe that CSTM outperforms LDA-CS in all the four applications in terms of topic diversity (Figure 5).

7.3.2 Coverage of known BerkeleyDB concerns

In order to obtain a qualitative confidence in the ability of the model(s) to surface representative concerns, we also assess whether topics found by the models cover known concerns in BerkeleyDB. We make this assessment based on whether words from a known concern(feature) name appear in the top 10 words of one or more topics. Both LDA-CS and CSTM exhibit a good coverage of these concerns (Figure 6). As most of these known concerns have a modular component, LDA-CS is able to detect them. We observe that CSTM surfaces all the known concerns found by LDA-CS plus a few additional concerns, e.g. ChunkedNIO (a diffused concern). Table 6 lists examples of the concern topics surfaced by CSTM.

Concern name	**CSTM Concern topic (most probable words)**
Evictor	evict nodes scan target bytes evictor renewed iter scanned eviction
Transactions	txn xid transaction active txns nxa prepare aborts commits commit
Latch	latch thread shared owner waiters held access exclusive stats latches
Statistics	bin count stats obsolete progress removed notify accumulator names dcl
ChunkedNIO	closed nio channel channels libraries communications job chunked log lock
Checksum	checksum user pre future adler implementation cksum validator anticipate assume

Table 6: Examples of BerkeleyDB concerns found by CSTM with 10 most likely words

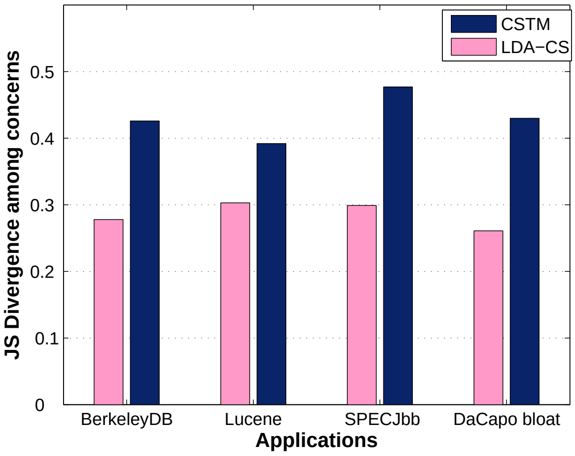


Figure 5: Jenson Shannon Divergence among concerns detected by CSTM and LDA-CS.

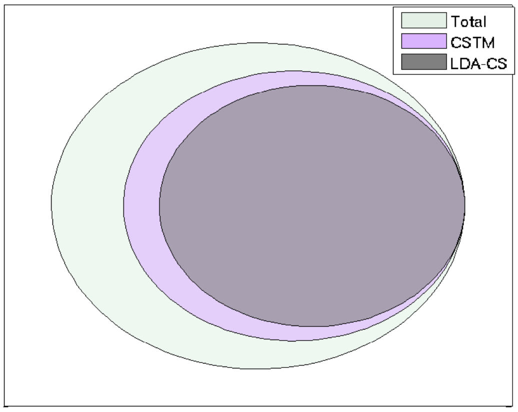


Figure 6: Out of 23 known concerns in BerkeleyDB, CSTM detects 19, and LDA-CS detects 17. All concerns detected by LDA-CS are detected by CSTM.

7.4 Programmer interpretability (human evaluation) study:

Criterion: *Does the method assign concerns to relevant statements with a meaningful interpretation ?*

Assessing the quality of the latent concern structure and the accuracy of fine grained assignments surfaced by these models involves a subjective judgment that requires expertise possessed by experienced programmers. We conducted a human evaluation study by designing multiple choice questions to quantify "programmer" interpretability of concerns assignments, in terms of metrics for word intrusion and statement topic mapping relevance, using a suitably modified version of the methodology recommended in [6].

In order to explain the rationale behind our chosen methodology for evaluating this criteria, we next discuss some of the alternatives we considered from the state-of-the-art in topic model evaluation.

7.4.1 Evaluating Topic Models: Choice of Methodology

Devising a suitable evaluation methodology to assess the outcome of topic modeling is non-trivial. The latent variables represent distributions over words – they are called “topics” only to reflect anecdotal experience that words co-occuring in a “topic” with a high probability are semantically connected to a common theme. This internal representation is difficult to validate directly because of the lack of ground truth to compare against. Instead, different evaluation criteria have been adopted in topic modeling literature [1, 6, 17, 18, 19], depending on the intended purpose, such as:

- Measures based on held-out likelihood (or perplexity [5]) that quantify how well the model learned from a corpus predicts the statistical characteristics of unseen documents or unseen parts of partly seen documents (document completion).
- Secondary measures that evaluate the use of the model for an external task independent of the topic space, such as information retrieval. For example, one such measure could be the performance of a classifier that uses the topics as features with topic proportions representing feature vectors of sampled documents. Here, the latent space inferred by the topic model represents a dimensionality reduction of the feature space of the document collection.
- Qualitative assessment that illustrates whether topics inferred are semantically meaningful. For example, many topic modeling papers present samples of topics found by the model – usually the ten most likely words in each topic are displayed to enable readers to judge the quality for themselves. The semantic content attributed to topics is important when the output is intended for human understanding, e.g. consider the automatic categorization, summarization and annotation of a large corpus of articles by their themes.
- Measures based on human evaluation that quantify the interpretability of the internal representation of the model, i.e. the extent to which both the topics themselves and the document-wise topic assignments are semantically meaningful. For example, Chang et al introduced the use of *word intrusion* and *topic intrusion* tests [6] as a human evaluation methodology – a practical approach that goes beyond a purely qualitative assessment to obtain a quantitative comparison of the interpretability of different models.

For the work described in this chapter, we use the fourth approach above. We employ a variation of the methodology [6] proposed by Chang et al [6] for assessing model interpretability. Interpretability of the latent structure of topics learnt is important when topic models are employed for an exploratory purpose as in our intended application – performance understanding of software concerns. Although objective evaluation measures based on held-out likelihood provide the most easily generalizable techniques to assess topic models, it has been observed that models that perform better on these measures may result in less interpretable topics [6]. There has been some effort to identify alternate objective metrics that could be used as a better proxy for judging topic interpretability e.g. point-wise mutual information [18] or other indicators of semantic coherence [20] of words in a topic. When this works well, it could save the need for a human evaluation. However, in our experimental trials we had mixed success in using or devising semantic coherence metrics as a consistent predictor of interpretability software code topics, especially across topics identified by different models and over different document collections.

7.4.2 Word intrusion and Topic intrusion measures

Chang et al. [6] describe two types of human evaluation tests which can be used to measure interpretability of the latent space discovered by a topic model.

The *word intrusion* test measures the conceptual coherence of the inferred topics according to human interpretation. The experiment involves inserting an intruder word at a random position in the list of top five words of a topic and testing whether participants are able to detect the odd word out (i.e. whether subjects agree with the model on the intruder word). The intruder word is selected so that it has a low probability in the topic being

[5] the *perplexity* of a held-out test set ($exp\{-\frac{\sum_{d=1}^{M}\sum_{n=1}^{N_d} log(p(w_{d,n}))}{\sum_{d=1}^{M} N_d}\}$) is monotonically decreasing in the likelihood of the test data; a lower perplexity score indicates better generalization performance. [1]

[6] we describe the exact methodology in more detail in our evaluation section

evaluated, but has a high probability in some other topic (so that it is not ruled out purely because it is a rare word in the corpus).

The *topic intrusion* test measures whether the topics assigned to a document by the model agree with human judgement. The experiment involves inserting an irrelevant intruder topic at a random position in the list of the three highest probability topics assigned a document and testing whether participants are able to detect the least relevant topic from the list when shown snippets from the document. Each topic in this list is shown as a set of the most likely words corresponding to that topic. The intruder topic is selected randomly from the other topics in the model which have a low probability in the document.

7.4.3 Adapted methodology

We make the following main adaptations to the methodology to make it suitable for assessing statement level concern assignments:

- We use statement topic mapping relevance instead of topic intrusion for assessing concern assignments. Unlike a text document, a code statement is unlikely to be assigned to more than one or two topics. Hence, asking respondents to choose the most relevant topic is more appropriate than detecting the least relevant topic. The percentage of responses that agree with the model provides a measure of statement topic mapping relevance.

- Instead of judging the concern assignment of a statement in isolation, we provide a snippet of about 5 code statements that are assigned to the same concern (preferably from the same file, but potentially non-contiguous lines) so that the participant has some context.

- We add optional fields in the questionnaire for users to fill in a label or name to topic word groups and statement groups. Unlike natural language text, topics for concerns can include terse or obscure words and program or domain specific terms that are difficult to interpret without application knowledge. Hence detecting intruder words can be a problem even for experienced programmers. In such cases, the labels of a topic or its matching statement group acts as a secondary indication of the topic's interpretability. When the labels assigned by different programmers are consistent we can conclude that the topic is interpretable even when an intruder word cannot be detected.

We illustrate a few sample questions from the study in the Appendix.

7.4.4 Study results

35 programmers participated in our study, including experienced Java programmers from multiple software development organizations as well as 12 computer science research students with a strong programming background. The questions were divided into questionnaires containing a set of simple tasks for determining word intrusion and statement topic mapping relevance. Different questionnaires were created covering samples of topics surfaced across the four applications on running the two model variations[7], LDA-CS and CSTM. The question sets were generated using automated scripts applied to the model outputs.

	CSTM	**LDA-CS**
No. of responses	171	85
No. of responses matching model	124	58
% matching responses	72.5%	68.2%

Table 7: Interpretability of statement level concern assignments (most relevant topic)

A total of 540 individual task responses were collected (a single programmer had the option of answering one or two orthogonal questionnaires, a total of 20 questions at the most), 345 responses for CSTM and 195 responses for LDA-CS. We circulated a larger number of copies of the questionnaires created based on CSTM output to focus more attention on evaluating the newer model.

[7] to contain the scale of expert effort required, we limit the number of concern topics sampled for evaluation to 40 per model, i.e. 10 from each application

	CSTM	**LDA-CS**
No. of responses	174	110
No. of detections of intruding word	71	33
% agreement with model on intruding word	40.8%	30%
Topic interpretability (% responses that indicate concern topics to be interpretable)	72.8%	63%

Table 8: Concern topic interpretability results: ability to detect intrusion words or assign consistent labels to concern topic words or its relevant statements

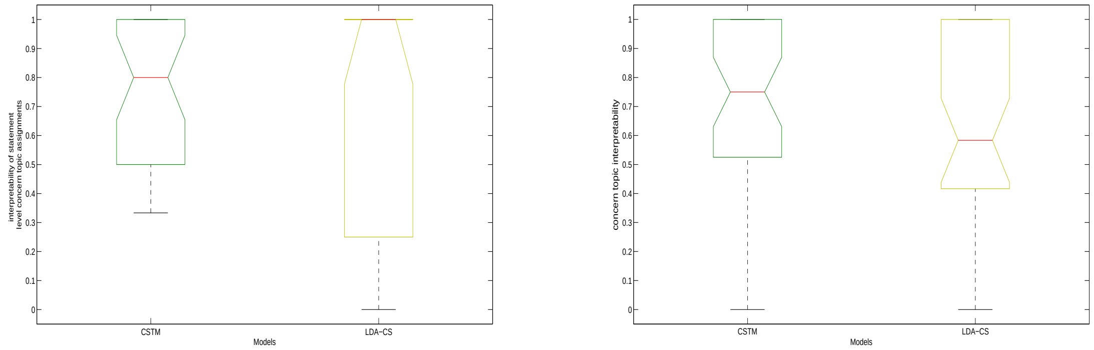


Figure 7: Notched boxplots of the *interpretability of statement level concern (topic) assignments* (left) and *concern (topic) interpretability* (right) for CSTM and LDA-CS. These plots illustrate the variation in interpretability across topics, complementing the results in Tables 7 and 8 which showed the averages aggregated over all responses. The interpretability value (Y-axis) computed for each topic represents the fraction of subjects whose responses indicate that they perceive that topic to be interpretable.

Table 7 and Table 8 summarize the consolidated results. We observe that statement level concern assignments are interpretable to a similar extent for both LDA-CS and CSTM, with about 70% responses that match the model. The word intrusion score, i.e. percentage agreement with the model in word intrusion detection, is comparatively low for the same topics even though we find that in many cases programmers were able to assign labels to the topic words or to the corresponding statement sets. To compute concern topic interpretability, we use the labeling consistency score (percentage of consistent labels) for a concern topic when the topic has a word intrusion score lower than 60% and at least 50% of its labels are consistent. Otherwise we rely on the topic's word intrusion score. We observe that LDA-CS and CSTM exhibit 60-70% topic interpretability for the sampled concern topics used in the study.

Researchers have previously pointed out that not all the topics inferred for a given document collection are interpretable [20]. Rather topic models produce a mix of topics, some of which are coherent and others less so – the higher the number of topics specified[8] for a given collection, the wider the mix. The notched boxplots in Fig 7 complement Tables 7 and 8, by providing a perspective of how the interpretability results vary across the sampled topics for LDA-CS and CSTM. The overlap between the notches indicates that differences in the medians may not be statistically significant. We also notice a lower variation in the CSTM results compared to LDA-CS for the sampled topics.

[8] increasing the number of topics increases the resolution of the model to help unearth specialized topics

Criteria	Evaluation question	LDA-CS	CSTM
Concern detection sensitivity (Table 4, 5)	Can the method surface diffused concerns?	No	Yes
Concern coverage diversity (Figure 5, 6)	Does the method surface a diverse set of concerns?	Good, but only for prominent concerns	Better than LDA-CS
Interpretability of concern assignments (Table 7, 8, Fig 7)	Does the method assign concerns to relevant statements with a meaningful interpretation?	About 60-70% of the time	As good as LDA-CS

Table 9: Evaluation summary

7.5 Summary

Table 9 summarizes our evaluation findings along the dimensions of concern detection sensitivity, concern diversity and interpretability of statement level concern assignments. CSTM exhibits better concern detection sensitivity and diversity and both LDA-CS and CSTM show 60-70% agreement with programmer interpretation in their statement level concern assignments.

8 Discussion

LDA	CSTM
`if (PreviousMaxWarehouses == 0)`	`if (PreviousMaxWarehouses == 0)`
`    MaxWarehouses = numberofwarehouses;`	`    MaxWarehouses = numberofwarehouses;`
`else`	`else`
`    ++MaxWarehouses;`	`    ++MaxWarehouses;`
`String msg = Loading Warehouse MaxWarehouses`	`String msg = Loading Warehouse MaxWarehouses`
`System.out.println(msg);`	`System.out.println(msg);`
`JBButil.getLog().info(msg);`	`JBButil.getLog().info(msg);`
`// Item Table must be loaded first since warehouses`	`//Item Table must be loaded first since warehouses`
`if (PreviousMaxWarehouses == 0) {`	`if (PreviousMaxWarehouses == 0) {`
`    loadItemTable();`	`    loadItemTable();`

Table 10: Lines from SPECjbb2005/spec/jbb/Company.java, colored based on concerns

A major challenge in statistical modeling is that there is no control over the concerns being detected. If we merely increase the number of topics, the same concerns can repeat multiple times, whereas many important concerns remain undetected. We have focused on this issue by introducing external control and boosting divergence among concerns to be detected. This clearly shows an improvement over state-of-the-art statistical models in detecting statement level concerns (Table 10).

Analysis time and scalability CSTM is a more complex model than LDA, and hence requires an increased analysis time for large applications. However, CSTM can be made scalable using online mechanisms [21].

Semantic Coherence Although our model can find meaningful concerns, it is well-known [20] that not all topics found by a statistical model may be meaningful or coherent. This can be addressed to an extent by filtering out topics which exhibit a relatively low semantic coherence [20].

Future work This paper barely scratches the surface in terms of potential applications combining information on latent concerns with program properties. For example, we have experimented with using the model to help highlight concerns responsible for high object churn, a common form of runtime bloat in Java applications. Concerns and their resource usage proportions could be included as features in models for estimating performance / power consumption and for mining resource intensive concern usage patterns.

9 Related Work

Statistical topic models: Latent Dirichlet Allocation (LDA) [1] is a well known topic model which has been successfully applied in various fields. [7, 8] introduced the use of LDA for analysing and mining software code to discover and model program concerns as latent topics without any apriori knowledge or expert input. Several variations of these techniques [22], [23] and delta-LDA [24] have been used for addressing software maintenance tasks such as estimating semantic coupling metrics, statistical debugging and software evolution, besides program comprehension and reverse engineering problems. To the best of our knowledge, we are the first to explore the possibility of using such models for performing automated summarization of runtime resource usage or other program properties in terms of latent concerns.

There has been some recent work on improving LDA to handle sparseness of short documents, by aggregating short documents into a larger text [25, 26] or incorporating large scale external data [27]. [28] proposed a model for sentence based summarization. However, none of these address detection of diffused concerns.

Other concern analysis techniques: There is a large body of existing literature on concern identification and location besides topic models. A variety of techniques ranging from formal concept analysis, exploiting program topology [3], information retrieval, graph mining, program slicing [29, 30] and dynamic analysis [31] have been employed in this context. Solutions that combine multiple approaches [32, 33, 34] and exploit multiple sources of information have also been used to improve the quality of results. However, unlike statistical topic models, most of these techniques assume that some information about the desired concern is provided to begin with, such as feature names, structural attributes, search patterns, bug reports, testcases or execution traces. Hence we find them less suitable for purposes of unsupervised automated performance summarization. Also, a statement may contribute to multiple concerns, in which case we need to attribute properties such as statement execution cost proportionately to these concerns. A statistical topic model provides a natural probabilistic framework to infer these proportions in terms of the probabilities assigned to different concern topics.

10 Conclusions

The ability to automatically discover representative *latent* concerns at the level of individual code statements can enable a new class of automated analyses. We have found that LDA and even specialized models such as MG-LDA, which have been applied successfully in other domains, fail to detect rare topics that occur only at a statement level in few files without a prominent presence in any module. Based on insights gained from these experiences, we presented a new statistical model variation that addresses these challenges along with a systematic evaluation methodology which confirms the effectiveness of our model and an application of the model in automated summarization of bytecode execution profiles. We observe that diffused concerns along with rare concerns can indeed account for a significant differences in resource usage under different execution scenarios. Our work is an important step towards the invention of sophisticated analysis tools (e.g for estimating software bloat) that combine information about underlying intent (as represented by latent concerns) with dynamic or static properties of programs.

References

[1] D. M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. pages 993–1022. Journal of Machine Learning Research, 2003.

[2] Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. WWW, 2008.

[3] M. P. Robillard. Topology analysis of software dependencies. Number 4. TOSEM, Aug. 2008.

[4] Suparna Bhattacharya, Karthick Rajamani, K Gopinath, and Manish Gupta. Software bloat and wasted joules: Is modularity a hurdle to green software? *IEEE Computer*, September 2011.

[5] Marius Marin, Arie Van Deursen, and Leon Moonen. Identifying crosscutting concerns using fan-in analysis. *TOSEM*, 17, December 2007.

[6] Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, 2009.

[7] Erik Linstead, Paul Rigor, Sushil Bajracharya, Cristina Lopes, and Pierre Baldi. Mining concepts from code with probabilistic topic models. ASE, 2007.

[8] Pierre F. Baldi, Cristina V. Lopes, Erik J. Linstead, and Sushil K. Bajracharya. A theory of aspects as latent topics. In *OOPSLA*, 2008.

[9] Girish Maskeri, Santonu Sarkar, and Kenneth Heafield. Mining business topics in source code using latent dirichlet allocation. In *Proceedings of the 1st India software engineering conference*, ISEC '08, pages 113–120, New York, NY, USA, 2008. ACM.

[10] Trevor Savage, Bogdan Dit, Malcom Gethers, and Denys Poshyvanyk. Topicxp: Exploring topics in source code using latent dirichlet allocation. In *Proceedings of the 2010 IEEE International Conference on Software Maintenance*, ICSM '10, pages 1–6, Washington, DC, USA, 2010. IEEE Computer Society.

[11] Hanna Wallach, David Mimno, and Andrew McCallum. Rethinking lda: Why priors matter. In *NIPS*, 2009.

[12] M. Wainwright and M. I. Jordan. A variational principle for graphical models. In *New Directions in Statistical Signal Processing*, chapter 11. MIT Press, 2005.

[13] Walter Binder, Jarle Hulaas, Philippe Moret, and Alex Villaz. Platform independent profiling in a virtual execution environment. SPE, 2009.

[14] S. M. Blackburn and et. al. The DaCapo benchmarks: Java benchmarking development and analysis. In *OOPSLA*, 2006.

[15] Christian Kastner, Sven Apel, and Don Batory. A case study implementing features using aspectj. SPLC, 2007.

[16] J. Lin. Divergence measures based on the shannon entropy. *Information Theory, IEEE Transactions on*, 37(1):145 –151, jan 1991.

[17] Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1105–1112, New York, NY, USA, 2009. ACM.

[18] David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. Evaluating topic models for digital libraries. In *Proceedings of the 10th annual joint conference on Digital libraries*, JCDL '10, pages 215–224, New York, NY, USA, 2010. ACM.

[19] Stephen W. Thomas. Mining software repositories using topic models. In *Proceedings of the 33rd International Conference on Software Engineering*, ICSE '11, pages 1138–1139, New York, NY, USA, 2011. ACM.

[20] David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *EMNLP*, 2011.

[21] Matthew Hoffman, David Blei, and Francis Bach. Online learning for latent dirichlet allocation. NIPS, 2010.

[22] Hazeline U. Asuncion, Arthur U. Asuncion, and Richard N. Taylor. Software traceability with topic modeling. ICSE, 2010.

[23] Stacy K. Lukins, Nicholas A. Kraft, and Letha H. Etzkorn. Bug localization using latent dirichlet allocation. Information and Software Technology, 2010.

[24] D. Andrzejewski, A. Mulhern, B. Liblit, and X Zhu. Statistical debugging using latent topic models. ECML, 2007.

[25] Liangjie Hong and Brian D. Davison. Empirical Study of Topic Modeling in Twitter. SOMA, 2010.

[26] J. Weng, E. P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. WSDM, 2010.

[27] X. H. Phan, L. M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. WWW, 2008.

[28] Chang Ying-Lang and Chien Jen-Tzung. Latent Dirichlet learning for document summarization. ICASSP, 2009.

[29] M. Harman, N. Gold, R. Hierons, and D. Binkley. Code extraction algorithms which unify slicing and concept assignment. In *WCRE*, 2002.

[30] D. Binkley, G. Gold, M. Harman, Z. Li, and K. Mahdavi. An empirical study of the relationship between the concepts expressed in source code and dependence. volume 81. J. Syst. Software, 2008.

[31] B. Cornelissen, A. Zaidman, A. Van Deursen, L. Moonen, and R. Koschke. A systematic survey of program comprehension through dynamic analysis. volume 99, pages 684–702. TSE, Apr. 2009.

[32] Meghan Revelle, Bogdan Dit, and Denys Poshyvanyk. Using data fusion and web mining to support feature location in softwareusing data fusion and web mining to support feature location in software. ICPC, 2010.

[33] W. Zhao, L. Zhang, Y. Liu, J. Sun, and F. Yang. Sniafl: Towards a static non-interactive approach to feature location. volume 15. TOSEM, April 2006.

[34] T. Savage, M. Revelle, and D. Poshyvanyk. Flat3: Feature location and textual tracing tool. ICSE, 2010.

A Computing Likelihood

$$\begin{aligned}
E_q[\log p(\theta^g|\alpha^g)] &= \int q(\theta^g)\log p(\theta^g|\alpha^g)d\theta^g \\
&= \int q(\theta^g)\{\sum_{i=1}^{K^g}(\alpha_i^g-1)\log\theta_i^g + \log\Gamma(\sum_{i=1}^{K^g}\alpha_i^g) - \sum_{i=1}^{K^g}\log\Gamma(\alpha_i^g)\}d\theta^g \\
&= \log\Gamma(\sum_{i=1}^{K^g}\alpha_i^g) - \sum_{i=1}^{K^g}\log\Gamma(\alpha_i^g)\} + \sum_{i=1}^{K^g}(\alpha_i^g-1)\{\int q(\theta^g)\log(\theta_i^g)d\theta^g\} \\
&= \log\Gamma(\sum_{i=1}^{K^g}\alpha_i^g) - \sum_{i=1}^{K^g}\log\Gamma(\alpha_i^g) + \sum_{i=1}^{K^g}(\alpha_i^g-1)E_q[\log\theta_i^g] \\
&= \log\Gamma(\sum_{i=1}^{K^g}\alpha_i^g) - \sum_{i=1}^{K^g}\log\Gamma(\alpha_i^g) + \sum_{i=1}^{K^g}(\alpha_i^g-1)(\Psi(\gamma_i^g) - \Psi(\sum_{j=1}^{K^g}\gamma_j^g)) \qquad (14)
\end{aligned}$$

$$\begin{aligned}
E_q[\log p(\theta^l|\alpha^l)] &= \sum_{i=1}^{T+S-1} E_q[\log p(\theta_i^l|\alpha^l)] \\
&= \sum_{i=1}^{T+S-1}[\,\log\Gamma(\sum_{j=1}^{K^l}\alpha_j^l) - \sum_{k=1}^{K^l}\log\Gamma(\alpha_k^l) + \sum_{k=1}^{K^l}(\alpha_k^l-1)(\Psi(\gamma_{ik}^l) - \Psi(\sum_{j=1}^{K^l}\gamma_{ij}^l))\,] \qquad (15)
\end{aligned}$$

$$\begin{aligned}
E_q[\log p(\psi|\lambda)] &= \sum_s E_q[\log p(\psi_s|\lambda)] \\
&= \sum_s \int q(\psi_s) \log p(\psi_s|\lambda)\ d\psi_s \\
&= \sum_{i=1}^{S} [\ \int q(\psi_i)\ \ \{ \sum_{k=1}^{T+S-1} (\lambda_k - 1) \log \psi_{ik} + \log \Gamma(\sum_{j=1}^{T+S-1} \lambda_j) - \sum_{k=1}^{T+S-1} \log \Gamma(\lambda_k) \}\ d\psi_i\ ] \\
&= \sum_{i=1}^{S} [\ \log \Gamma(\sum_{j=1}^{T+S-1} \lambda_j) - \sum_{k=1}^{T+S-1} \log \Gamma(\lambda_k)\ \ + \int q(\psi_i) \sum_{k=1}^{T+S-1} (\lambda_k - 1) \log \psi_{ik}\ d\psi_i\ ] \\
&= \sum_{i=1}^{S} [\ \log \Gamma(\sum_{j=1}^{T+S-1} \lambda_j) - \sum_{k=1}^{T+S-1} \log \Gamma(\lambda_k)\ \ + \sum_{k=1}^{T+S-1} (\lambda_k - 1) \int q(\psi_{ik}) \log \psi_{ik}\ d\psi_{ik}\ ] \\
&= \sum_{i=1}^{S} [\ \log \Gamma(\sum_{j=1}^{T+S-1} \lambda_j) - \sum_{k=1}^{T+S-1} \log \Gamma(\lambda_k)\ \ + \sum_{k=1}^{T+S-1} (\lambda_k - 1)(\Psi(\epsilon_{ik}) - \Psi(\sum_{j=1}^{T+S-1} \epsilon_{ij}))\ ]
\end{aligned} \tag{16}$$

$$\begin{aligned}
E_q[\log p(v|\psi)] &= \sum_{n=1}^{N} E_q[\log p(v_n|\psi_n)] \\
&= \sum_{n=1}^{N} \sum_{i=1}^{T+S-1} \int q(v_n = i|\sigma_n) q(\psi_{ni}|\epsilon_{ni}) \log p(v_n = i|\psi_n) \\
&= \sum_{n=1}^{N} \sum_{i=1}^{T+S-1} \sigma_{ni} E_q[\log \psi_{ni}] \\
&= \sum_{n=1}^{N} \sum_{i=1}^{T+S-1} \sigma_{ni} (\Psi(\epsilon_{ni}) - \Psi(\sum_{j=1}^{T+S-1} \epsilon_{nj}))
\end{aligned} \tag{17}$$

$$\begin{aligned}
E_q[\log p(\pi|\alpha^m)] &= \sum_{i=1}^{T+S-1} E_q[\log p(\pi_i|\alpha^m)] \\
&= \sum_{i=1}^{T+S-1} \int q(\pi_i|\gamma_i^m) \log p(\pi_i|\alpha^m) d\pi_i \\
&= \sum_{i=1}^{T+S-1} \int q(\pi_i|\gamma_i^m) \sum_{k=0}^{1} (\alpha_k^m - 1) \log \pi_{ik} + \log \Gamma(\sum_{k=0}^{1} \alpha_k^m) - \sum_{k=0}^{1} \log \Gamma(\alpha_k^m) \\
&= \sum_{i=1}^{T+S-1} \log \Gamma(\sum_{k=0}^{1} \alpha_k^m) - \sum_{k=0}^{1} \log \Gamma(\alpha_k^m) + \sum_{k=0}^{1} (\alpha_k^m - 1) \int q(\pi_{ik}|\gamma_i^m) \log \pi_{ik} \\
&= \sum_{i=1}^{T+S-1} [\ \log \Gamma(\sum_{k=0}^{1} \alpha_k^m) - \sum_{k=0}^{1} \log \Gamma(\alpha_k^m) + \sum_{k=0}^{1} (\alpha_k^m - 1)(\Psi(\gamma_{ik}^m) - \Psi(\sum_{j=0}^{1} \gamma_{ij}^m))\ ]
\end{aligned} \tag{18}$$

$$\begin{aligned} p(\pi_i|\gamma_i^m) &= \frac{\Gamma(\sum_{j=0}^{1}\gamma_{ij}^m)}{\prod_{k=0}^{1}\Gamma(\gamma_{ik}^m)}\prod_{k=0}^{1}\pi_{ik}^{\gamma_{ik}^m-1} \\ &= \exp\log\frac{\Gamma(\sum_{j=0}^{1}\gamma_{ij}^m)}{\prod_{k=0}^{1}\Gamma(\gamma_{ik}^m)}\prod_{k=0}^{1}\pi_{ik}^{\gamma_{ik}^m-1} \\ &= \exp[\;\log\Gamma(\sum_{j=0}^{1}\gamma_{ij}^m) - \sum_{k=0}^{1}\log\Gamma(\gamma_{ik}^m) + \sum_{k=0}^{1}(\gamma_{ik}^m-1)\log\pi_{ik}\;] \end{aligned} \tag{19}$$

Natural parameter $\eta = (\gamma_{ik}^m - 1)$, sufficient statistics $T(\pi) = \log\pi_{ik}$ and the log-normalizer $A(\gamma_i^m) = \sum_{k=0}^{1}\log\Gamma(\gamma_{ik}^m) - \log\Gamma(\sum_{j=0}^{1}\gamma_{ij}^m)$. Hence, $E[\log\pi_{ik}] = \Psi(\gamma_{ik}^m) - \Psi(\sum_{j=0}^{1}\gamma_{ij}^m)$.

$$\begin{aligned} E_q[\log p(\delta|\pi,v)] &= \sum_{n=1}^{N}E_q[\log p(\delta_n|\pi_{v_n})] \\ &= \sum_{n=1}^{N}\sum_{k=0}^{1}\sum_{v_n}\int_{\pi_{v_n}} q(\delta_n=k|\xi_n)q(\pi_{v_n}|\gamma_{v_n}^m)q(v_n|\sigma_n)\log\pi_{v_n k}\;\; d\pi_{v_n} \\ &= \sum_{n=1}^{N}\sum_{k=0}^{1}\sum_{i=1}^{T+S-1}\xi_{nk}q(v_n=i|\sigma_n)\int q(\pi_i|\gamma_i^m)\log\pi_{ik}\;\; d\pi_i \\ &= \sum_{n=1}^{N}\sum_{k=0}^{1}\sum_{i=1}^{T+S-1}\xi_{nk}\sigma_{ni}\int q(\pi_i|\gamma_i^m)\log\pi_{ik}\;\; d\pi_i \\ &= \sum_{n=1}^{N}\sum_{k=0}^{1}\sum_{i=1}^{T+S-1}\xi_{nk}\sigma_{ni}E_q[\log\pi_{ik}] \\ &= \sum_{n=1}^{N}\sum_{k=0}^{1}\sum_{i=1}^{T+S-1}\xi_{nk}\sigma_{ni}(\Psi(\gamma_{ik}^m) - \Psi(\sum_{j=0}^{1}\gamma_{ij}^m)) \end{aligned} \tag{20}$$

$$
\begin{aligned}
E_q[\log p(z|t,v,\delta,\theta)] &= \sum_{n=1}^{N} E_q[\log p(z_n|\theta, v_n, \delta_n, t_n)] \\
&= \sum_{n=1}^{N} \sum_{z_n} \sum_{\delta_n} \int_{\theta} q(z_n, v_n, \delta_n, t_n, \theta) \log p(z_n|\theta, v_n, \delta_n, t_n) \;\; d\theta \\
&= \sum_{n=1}^{N} \sum_{z_n} \sum_{v_n} \sum_{\delta_n} \int_{\theta} q(z_n) q(v_n) q(\delta_n) q(\theta) q(t_n) \log p(z_n|\theta, v_n, \delta_n, t_n) \;\; d\theta \\
&= \sum_{n=1}^{N} \sum_{h=0}^{1} \sum_{k} \sum_{v_n} \int_{\theta} q(z_n = k) q(v_n = j) q(\delta_n = h) q(\theta) q(t_n) \log p(z_n = k|\theta, v_n = j, \delta_n + t_n = h) \;\; d\theta \\
&= \sum_{n=1}^{N} [\sum_{k=1}^{K^g} \sum_{j=1}^{T+S-1} \int_{\theta^g} q(z_n^g = k) q(v_n = j) q(\delta_n = 0) q(\theta^g) q(t_n = 0) \log p(z_n = k|\theta^g, \delta_n = 0, t_n = 0) \;\; d\theta^g \\
&\quad + \sum_{k=1}^{K^l} \sum_{j=1}^{T+S-1} \int_{\theta^l} q(z_n^l = k) q(v_n = j) q(\delta_n = 1) q(t_n = 0) q(\theta^g) \log p(z_n = k|\theta^g, \delta_n = 1, t_n = 0, v_n = j) \\
&\quad + \sum_{k=1}^{K^l} \sum_{j=1}^{T+S-1} \int_{\theta^l} q(z_n^l = k) q(v_n = j) q(\delta_n = 1) q(t_n = 1) q(\theta^l) \log p(z_n = k|\theta^l, \delta_n = 1, t_n = 1, v_n = j) \\
&\quad + \sum_{k=1}^{K^l} \sum_{j=1}^{T+S-1} \int_{\theta^l} q(z_n^l = k) q(v_n = j) q(\delta_n = 0) q(t_n = 1) q(\theta^l) \log p(z_n = k|\theta^l, \delta_n = 0, t_n = 1, v_n = j) \\
&= \sum_{n=1}^{N} [\sum_{k=1}^{K^g} \int_{\theta^g} \phi_{nk}^g \xi_{n0} \rho_{n0} q(\theta^g) \log \theta_k^g \;\; d\theta^g \; + \sum_{k=1}^{K^g} \int_{\theta^g} \phi_{nk}^g \xi_{n1} \rho_{n0} q(\theta^g) \log \theta_k^g \;\; d\theta^g \;] \\
&+ \sum_{k=1}^{K^l} \sum_{j=1}^{T+S-1} \int_{\theta^l} \phi_{nk}^l \sigma_{nj} \xi_{n1} \rho_{n1} q(\theta^l) \log \theta_{jk}^l \;\; d\theta^l \;] \; + \sum_{k=1}^{K^l} \sum_{j=1}^{T+S-1} \int_{\theta^l} \phi_{nk}^l \sigma_{nj} \xi_{n0} \rho_{n1} q(\theta^l) \log \theta_{jk}^l \;\; d\theta^l \;] \\
&= \sum_{n=1}^{N} [\sum_{k=1}^{K^g} \phi_{nk}^g \xi_{n0} \rho_{n0} (\Psi(\gamma_k^g) - \Psi(\sum_{j=1}^{K^g} \gamma_j^g)) \; + \sum_{k=1}^{K^l} \sum_{j=1}^{T+S-1} \phi_{nk}^l \sigma_{nj} \xi_{n1} \rho_{n0} (\Psi(\gamma_{jk}^l) - \Psi(\sum_{h=1}^{K^l} \gamma_{jh}^l)) \;] \\
&+ \sum_{k=1}^{K^l} \sum_{j=1}^{T+S-1} \phi_{nk}^l \sigma_{nj} \xi_{n1} \rho_{n1} (\Psi(\gamma_{jk}^l) - \Psi(\sum_{h=1}^{K^l} \gamma_{jh}^l)) \;] \; + \sum_{k=1}^{K^l} \sum_{j=1}^{T+S-1} \phi_{nk}^l \sigma_{nj} \xi_{n0} \rho_{n1} (\Psi(\gamma_{jk}^l) - \Psi(\sum_{h=1}^{K^l} \gamma_{jh}^l)) \;]
\end{aligned}
$$

$$
\begin{aligned}
E_q[\log p(w|z,\delta,\beta,t)] &= \sum_{n=1}^{N} E_q[\log p(w_n|z_n,\beta,\delta_n,t_n)] \\
&= \sum_{n=1}^{N}\sum_{z_n}\sum_{\delta_n} q(z_n,\delta_n,t_n)\log p(w_n|z_n,\beta,\delta_n,t_n) \\
&= \sum_{n=1}^{N}\sum_{\delta_n}\sum_{z_n} q(z_n)q(\delta_n)q(t_n)\log p(w_n|z_n,\beta,\delta_n,t_n) \\
&= \sum_{n=1}^{N}\sum_{h=0}^{1}\sum_{z_n} q(z_n=k)q(\delta_n)q(t_n)\log p(w_n|z_n=k,\beta,\delta_n=h) \\
&= \sum_{n=1}^{N}[\sum_{z_n^g} q(z_n^g=k)q(\delta_n=0)q(t_n=0)\log p(w_n|z_n^g=k,\beta,\delta_n+t_n=0) \\
&\quad + \sum_{z_n^l} q(z_n^l=k)q(\delta_n=1)q(t_n=0)\log p(w_n|z_n^l=k,\beta,\delta_n+t_n=1)\ ] \\
&\quad + \sum_{z_n^l} q(z_n^l=k)q(\delta_n=1)q(t_n=1)\log p(w_n|z_n^l=k,\beta,\delta_n+t_n=1)\ ] \\
&\quad + \sum_{z_n^l} q(z_n^l=k)q(\delta_n=0)q(t_n=1)\log p(w_n|z_n^l=k,\beta,\delta_n+t_n=1)\ ] \\
&= \sum_{n=1}^{N}[\sum_{k=1}^{K^g}\sum_{j=1}^{V}\phi_{nk}^g\xi_{n0}\rho_{n0}\log\beta_{kj}^g I_{w_n=j}\ +\ \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi_{nk}^l\xi_{n1}\rho_{n0}\log\beta_{kj}^l I_{w_n=j}\ ] \\
&+ \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi_{nk}^l\xi_{n1}\rho_{n1}\log\beta_{kj}^l I_{w_n=j}\ +\ \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi_{nk}^l\xi_{n0}\rho_{n1}\log\beta_{kj}^l I_{w_n=j}\ ]
\end{aligned}
\tag{22}
$$

$$
\begin{aligned}
E_q[\log p(t)] &= \sum_{n=1}^{N} E_q[\log p(t_n)] \\
&= \sum_{n=1}^{N}\sum_{k=0}^{1} q(t_n=k)\log\zeta_k \\
E_q[\log p(t)] &= \sum_{n=1}^{N}\sum_{k=0}^{1}\rho_{nk}\log\zeta_k
\end{aligned}
\tag{23}
$$

$$
\begin{aligned}
E_q[\log q(\theta^g)] &= E_q[\sum_{i=1}^{K^g}(\gamma_i^g-1)\log\theta_i^g+\log\Gamma(\sum_{i=1}^{K^g}\gamma_i^g)-\sum_{i=1}^{K^g}\log\Gamma(\gamma_i^g)] \\
&= \sum_{i=1}^{K^g}(\gamma_i^g-1)E_q\log\theta_i^g+\log\Gamma(\sum_{i=1}^{K^g}\gamma_i^g)-\sum_{i=1}^{K^g}\log\Gamma(\gamma_i^g) \\
&= \sum_{i=1}^{K^g}(\gamma_i^g-1)(\Psi(\gamma_i^g)-\Psi(\sum_{j=1}^{K^g}\gamma_j^g))+\log\Gamma(\sum_{i=1}^{K^g}\gamma_i^g)-\sum_{i=1}^{K^g}\log\Gamma(\gamma_i^g)
\end{aligned}
\tag{24}
$$

$$\begin{aligned} E_q[\log q(\theta^l)] &= \sum_{i=1}^{T+S-1} E_q[\log q(\theta_i^l)] \\ &= \sum_{i=1}^{T+S-1} [\, \sum_{k=1}^{K^l} (\gamma_{ik}^l - 1)(\Psi(\gamma_{ik}^l) - \Psi(\sum_{j=1}^{K^l} \gamma_{ij}^l)) + \log\Gamma(\sum_{j=1}^{K^l} \gamma_{ij}^l) - \sum_{k=1}^{K^l} \log\Gamma(\gamma_{ik}^l)\,] \end{aligned} \tag{25}$$

$$\begin{aligned} E_q[\log q(\pi)] &= \sum_{i=1}^{T+S-1} E_q[\log q(\pi_i)] \\ &= \sum_{i=1}^{T+S-1} [\, \sum_{k=0}^{1} (\gamma_{ik}^m - 1)(\Psi(\gamma_{ik}^m) - \Psi(\sum_{j=0}^{1} \gamma_{ij}^m)) + \log\Gamma(\sum_{k=0}^{1} \gamma_{ik}^m) - \sum_{j=0}^{1} \log\Gamma(\gamma_{ij}^m)\,] \end{aligned} \tag{26}$$

$$\begin{aligned} E_q[\log q(\psi)] &= \sum_{i=1}^{S} E_q[\log q(\psi_i)] \\ &= \sum_{i=1}^{S} [\sum_{k=1}^{T+S-1} (\epsilon_{ik} - 1)(\Psi(\epsilon_{ik}) - \Psi(\sum_{j=1}^{T+S-1} \epsilon_{ij})) + \log\Gamma(\sum_{j=1}^{T+S-1} \epsilon_{ij}) - \sum_{k=1}^{T+S-1} \log\Gamma(\epsilon_{ik})] \end{aligned} \tag{27}$$

$$\begin{aligned} E_q[\log q(z|\delta = g)] &= \sum_{n=1}^{N} E_q[\log q(z_n|\delta_n = g)] \\ &= \sum_{n=1}^{N} q(z_n) \log q(z_n|\delta = g) \\ &= \sum_{n=1}^{N} \sum_{i=1}^{K^g} q(z_n = i) \log q(z_n = i|\delta_n = g) \\ &= \sum_{n=1}^{N} \sum_{i=1}^{K^g} \phi_{ni}^g \log \phi_{ni}^g \end{aligned} \tag{28}$$

$$E_q[\log q(z|\delta = l)] = \sum_{n=1}^{N} \sum_{i=1}^{K^l} \phi_{ni}^l \log \phi_{ni}^l \tag{29}$$

$$\begin{aligned} E_q[\log q(\delta|\xi)] &= \sum_{n=1}^{N} E_q[\log q(\delta_n|\xi_n)] \\ &= \sum_{n=1}^{N} \xi_n \log \xi_n + (1 - \xi_n) \log(1 - \xi_n) \end{aligned} \tag{30}$$

$$\begin{aligned} E_q[\log q(v|\sigma)] &= \sum_{n=1}^{N} E_q[\log q(v_n|\sigma_n)] \\ &= \sum_{n=1}^{N} \sum_{i=1}^{T+S-1} q(v_n = i|\sigma_n) \log q(v_n = i|\sigma_n) \\ &= \sum_{n=1}^{N} \sum_{i=1}^{T+S-1} \sigma_{ni} \log \sigma_{ni} \end{aligned} \tag{31}$$

$$
\begin{aligned}
E_q[\log q(t)] &= \sum_{n=1}^{N} E_q[\log q(t_n)] \\
E_q[\log q(t)] &= \sum_{n=1}^{N} \sum_{k=0}^{1} \rho_{nk} \log \rho_{nk}
\end{aligned}
\tag{32}
$$

B Computing EM algorithm

Computing ϕ^l

$$
\begin{aligned}
\mathcal{L}_{\phi^l} &= \sum_{n=1}^{N}\sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1} \phi^l_{nk}\sigma_{nj}\xi_{n1}(\Psi(\gamma^l_{jk}) - \Psi(\sum_{h=1}^{K^l}\gamma^l_{jh})) + \sum_{n=1}^{N}\sum_{k=1}^{K^l}\sum_{g=1}^{V} \phi^l_{nk}\xi_{n1}\log\beta^l_{kg} I_{w_n=g} - \sum_{n=1}^{N}\sum_{k=1}^{K^l}\phi^l_{nk}\log\phi^l_{nk} \\
\max_{\phi^l_{nk}} \ \mathcal{L} & \qquad s.t. \sum_{k=1}^{K^l}\phi^l_{nk} = 1 \\
\mathcal{Q} &= \sum_{n=1}^{N}\sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1} \phi^l_{nk}\sigma_{nj}\xi_{n1}(\Psi(\gamma^l_{jk}) - \Psi(\sum_{h=1}^{K^l}\gamma^l_{jh})) + \sum_{n=1}^{N}\sum_{k=1}^{K^l}\sum_{g=1}^{V} \phi^l_{nk}\xi_{n1}\log\beta^l_{kg} I_{w_n=g} - \sum_{n=1}^{N}\sum_{k=1}^{K^l}\phi^l_{nk}\log\phi^l_{nk} - \rho(\sum_{k=1}^{K^l} \\
\frac{\partial\mathcal{Q}}{\partial\phi^l_{nk}} &= \sum_{j=1}^{T+S-1} \sigma_{nj}\xi_{n1}(\Psi(\gamma^l_{jk}) - \Psi(\sum_{h=1}^{K^l}\gamma^l_{jh})) + \xi_{n1}\log\beta^l_{kg} I_{w_n=g} - 1 - \log\phi^l_{nk} - \rho = 0 \\
\phi^l_{nk} &\propto (\beta^{l\ \xi_{n1}}_{kg} I_{w_n=g}) \exp(\sum_{j=1}^{T+S-1} \sigma_{nj}\xi_{n1}(\Psi(\gamma^l_{jk}) - \Psi(\sum_{h=1}^{K^l}\gamma^l_{jh}))
\end{aligned}
$$

Computing ϕ^g

$$
\mathcal{L}_{\phi^g} = \sum_{n=1}^{N}\sum_{k=1}^{K^g} \phi^g_{nk}\xi_{n0}(\Psi(\gamma^g_k) - \Psi(\sum_{j=1}^{K^g}\gamma^g_j)) + \sum_{n=1}^{N}\sum_{k=1}^{K^g}\sum_{j=1}^{V} \phi^g_{nk}\xi_{n0}\log\beta^g_{kj} I_{w_n=j} - \sum_{n=1}^{N}\sum_{k=1}^{K^g}\phi^g_{nk}\log\phi^g_{nk} \tag{35}
$$

$$
\begin{aligned}
\max_{\phi^g_{nk}} \ \mathcal{L} & \qquad s.t. \sum_{k=1}^{K^g}\phi^g_{nk} = 1 \\
\mathcal{Q} &= \sum_{n=1}^{N}\sum_{k=1}^{K^g} \phi^g_{nk}\xi_{n0}(\Psi(\gamma^g_k) - \Psi(\sum_{j=1}^{K^g}\gamma^g_j)) + \sum_{n=1}^{N}\sum_{k=1}^{K^g}\sum_{j=1}^{V} \phi^g_{nk}\xi_{n0}\log\beta^g_{kj} I_{w_n=j} - \sum_{n=1}^{N}\sum_{k=1}^{K^g}\phi^g_{nk}\log\phi^g_{nk} - \rho(\sum_{k=1}^{K^g}\phi^g_{nk} - 1) \\
\frac{\partial\mathcal{Q}}{\partial\phi^g_{nk}} &= \xi_{n0}(\Psi(\gamma^g_k) - \Psi(\sum_{j=1}^{K^g}\gamma^g_j)) + \xi_{n0}\log\beta^g_{kj} I_{w_n=j} - 1 - \log\phi^g_{nk} - \rho = 0
\end{aligned}
$$

$$
\phi^g_{nk} \propto \beta^{g\ \xi_{n0}}_{kj} \exp(\xi_{n0}(\Psi(\gamma^g_k) - \Psi(\sum_{j=1}^{K^g}\gamma^g_j))) \tag{36}
$$

(37)

Computing ξ

$$
\begin{aligned}
\mathcal{L}_{\xi_0} &= \sum_{n=1}^{N}[\sum_{k=1}^{K^g}\phi^g_{nk}\xi_{n0}\rho_{n0}(\Psi(\gamma^g_k)-\Psi(\sum_{j=1}^{K^g}\gamma^g_j)) + \sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1}\phi^l_{nk}\sigma_{nj}\xi_{n0}\rho_{n1}(\Psi(\gamma^l_{jk})-\Psi(\sum_{h=1}^{K^l}\gamma^l_{jh})) \\
&+ \sum_{k=1}^{K^g}\sum_{j=1}^{V}\phi^g_{nk}\xi_{n0}\rho_{n0}\log\beta^g_{kj}I_{w_n=j} + \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi^l_{nk}\xi_{n0}\rho_{n1}\log\beta^l_{kj}I_{w_n=j} - \xi_{n0}\log\xi_{n0}\ ] \\
\frac{\partial\mathcal{L}}{\partial\xi_{n0}} &= \sum_{k=1}^{K^g}\phi^g_{nk}\rho_{n0}(\Psi(\gamma^g_k)-\Psi(\sum_{j=1}^{K^g}\gamma^g_j)) + \sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1}\phi^l_{nk}\sigma_{nj}\rho_{n1}(\Psi(\gamma^l_{jk})-\Psi(\sum_{h=1}^{K^l}\gamma^l_{jh})) \\
&+ \sum_{k=1}^{K^g}\sum_{j=1}^{V}\phi^g_{nk}\rho_{n0}\log\beta^g_{kj}I_{w_n=j} + \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi^l_{nk}\rho_{n1}\log\beta^l_{kj}I_{w_n=j} - 1 - \log\xi_{n0} \\
\xi_{n0} &\propto \exp(\sum_{k=1}^{K^g}\phi^g_{nk}\rho_{n0}(\Psi(\gamma^g_k)-\Psi(\sum_{j=1}^{K^g}\gamma^g_j)) + \sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1}\phi^l_{nk}\sigma_{nj}\rho_{n1}(\Psi(\gamma^l_{jk})-\Psi(\sum_{h=1}^{K^l}\gamma^l_{jh})) \\
&+ \sum_{k=1}^{K^g}\sum_{j=1}^{V}\phi^g_{nk}\rho_{n0}\log\beta^g_{kj}I_{w_n=j} + \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi^l_{nk}\rho_{n1}\log\beta^l_{kj}I_{w_n=j}) \qquad (38)
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{L}_{\xi_1} &= \sum_{n=1}^{N}[\sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1}\phi^l_{nk}\sigma_{nj}\xi_{n1}\rho_{n0}(\Psi(\gamma^l_{jk})-\Psi(\sum_{h=1}^{K^l}\gamma^l_{jh})) \\
&+ \sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1}\phi^l_{nk}\sigma_{nj}\xi_{n1}\rho_{n1}(\Psi(\gamma^l_{jk})-\Psi(\sum_{h=1}^{K^l}\gamma^l_{jh}))\ + \sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1}\phi^l_{nk}\sigma_{nj}\xi_{n0}\rho_{n1}(\Psi(\gamma^l_{jk})-\Psi(\sum_{h=1}^{K^l}\gamma^l_{jh})) \\
&+ \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi^l_{nk}\xi_{n1}\rho_{n0}\log\beta^l_{kj}I_{w_n=j} \\
&+ \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi^l_{nk}\xi_{n1}\rho_{n1}\log\beta^l_{kj}I_{w_n=j}\ + \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi^l_{nk}\xi_{n0}\rho_{n1}\log\beta^l_{kj}I_{w_n=j} - \xi_{n1}\log\xi_{n1}\ ]
\end{aligned}
\qquad (39)
$$

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \xi_{n1}} &= \sum_{n=1}^{N}[\sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1}\phi^l_{nk}\sigma_{nj}\rho_{n0}(\Psi(\gamma^l_{jk})-\Psi(\sum_{h=1}^{K^l}\gamma^l_{jh}))\\
&+ \sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1}\phi^l_{nk}\sigma_{nj}\rho_{n1}(\Psi(\gamma^l_{jk})-\Psi(\sum_{h=1}^{K^l}\gamma^l_{jh})) + \sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1}\phi^l_{nk}\sigma_{nj}\xi_{n0}\rho_{n1}(\Psi(\gamma^l_{jk})-\Psi(\sum_{h=1}^{K^l}\gamma^l_{jh}))\\
&+ \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi^l_{nk}\xi_{n1}\rho_{n0}\log\beta^l_{kj}I_{w_n=j}\\
&+ \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi^l_{nk}\xi_{n1}\rho_{n1}\log\beta^l_{kj}I_{w_n=j} + \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi^l_{nk}\xi_{n0}\rho_{n1}\log\beta^l_{kj}I_{w_n=j} - 1 - \log\xi_{n1}\ ]\\
\xi_{n1} &\propto \exp(\sum_{n=1}^{N}[\sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1}\phi^l_{nk}\sigma_{nj}\rho_{n0}(\Psi(\gamma^l_{jk})-\Psi(\sum_{h=1}^{K^l}\gamma^l_{jh}))\\
&+ \sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1}\phi^l_{nk}\sigma_{nj}\rho_{n1}(\Psi(\gamma^l_{jk})-\Psi(\sum_{h=1}^{K^l}\gamma^l_{jh})) + \sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1}\phi^l_{nk}\sigma_{nj}\xi_{n0}\rho_{n1}(\Psi(\gamma^l_{jk})-\Psi(\sum_{h=1}^{K^l}\gamma^l_{jh}))\\
&+ \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi^l_{nk}\xi_{n1}\rho_{n0}\log\beta^l_{kj}I_{w_n=j}\\
&+ \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi^l_{nk}\xi_{n1}\rho_{n1}\log\beta^l_{kj}I_{w_n=j} + \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi^l_{nk}\xi_{n0}\rho_{n1}\log\beta^l_{kj}I_{w_n=j}\ ]) \qquad (40)
\end{aligned}
$$

Computing σ

$$
\begin{aligned}
\mathcal{L}_\sigma &= \sum_{n=1}^{N}\sum_{h=0}^{1}\sum_{i=1}^{T+S-1}\xi_{nh}\sigma_{ni}(\Psi(\gamma^m_{ih})-\Psi(\sum_{j=0}^{1}\gamma^m_{ij})) + \sum_{n=1}^{N}\sum_{k=1}^{K^l}\sum_{i=1}^{T+S-1}\phi^l_{nk}\sigma_{ni}\xi_{n1}(\Psi(\gamma^l_{ik})-\Psi(\sum_{h=1}^{K^l}\gamma^l_{ih}))\\
&+ \sum_{n=1}^{N}\sum_{i=1}^{T+S-1}\sigma_{ni}(\Psi(\epsilon_{ni})-\Psi(\sum_{j=1}^{T+S-1}\epsilon_{nj})) - \sum_{n=1}^{N}\sum_{i=1}^{T+S-1}\sigma_{ni}\log\sigma_{ni} \qquad (41\\
\max_{\sigma_{ni}}\ \mathcal{L} & \qquad \sum_{i=1}^{T+S-1}\sigma_{ni} = 1\\
\mathcal{Q} &= \mathcal{L}_\sigma + \rho(\sum_{i=1}^{T+S-1}\sigma_{ni} - 1)\\
\frac{\partial \mathcal{Q}}{\partial \sigma_{ni}} &= \sum_{h=0}^{1}\xi_{nh}(\Psi(\gamma^m_{ih})-\Psi(\sum_{j=0}^{1}\gamma^m_{ij})) + \sum_{k=1}^{K^l}\phi^l_{nk}\xi_{n1}(\Psi(\gamma^l_{ik})-\Psi(\sum_{h=1}^{K^l}\gamma^l_{ih})) + \Psi(\epsilon_{ni}) - \Psi(\sum_{j=1}^{T+S-1}\epsilon_{nj}) - 1 - \log\sigma_{ni} - \rho = 0\\
\sigma_{ni} &\propto \exp(\sum_{h=0}^{1}\xi_{nh}(\Psi(\gamma^m_{ih})-\Psi(\sum_{j=0}^{1}\gamma^m_{ij})) + \sum_{k=1}^{K^l}\phi^l_{nk}\xi_{n1}(\Psi(\gamma^l_{ik})-\Psi(\sum_{h=1}^{K^l}\gamma^l_{ih})) + \Psi(\epsilon_{ni}) - \Psi(\sum_{j=1}^{T+S-1}\epsilon_{nj})\ ) \qquad (42
\end{aligned}
$$

Computing ϵ

$$\begin{aligned}
\mathcal{L}_{\epsilon} &= \sum_{i=1}^{S}[\, \log\Gamma(\sum_{j=1}^{T+S-1}\lambda_j) - \sum_{k=1}^{T+S-1}\log\Gamma(\lambda_k) \ + \sum_{k=1}^{T+S-1}(\lambda_k-1)(\Psi(\epsilon_{ik}) - \Psi(\sum_{j=1}^{T+S-1}\epsilon_{ij}))\,] \\
&+ \sum_{n=1}^{N}\sum_{k=1}^{T+S-1}\sigma_{nk}(\Psi(\epsilon_{nk}) - \Psi(\sum_{j=1}^{T+S-1}\epsilon_{nj})) \\
&- \sum_{i=1}^{S}[\sum_{k=1}^{T+S-1}(\epsilon_{ik}-1)(\Psi(\epsilon_{ik}) - \Psi(\sum_{j=1}^{T+S-1}\epsilon_{ij})) + \log\Gamma(\sum_{j=1}^{T+S-1}\epsilon_{ij}) - \sum_{k=1}^{T+S-1}\log\Gamma(\epsilon_{ik})]
\end{aligned} \tag{43}$$

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial\epsilon_{ik}} &= (\lambda_k-1)\Psi'(\epsilon_{ik}) + \sum_{n=1}^{N}\sigma_{s_n k}I_{s_n=i}\Psi'(\epsilon_{ik}) - (\epsilon_{ik}-1)\Psi'(\epsilon_{ik}) - \Psi(\epsilon_{ik}) + \Psi(\epsilon_{ik}) = 0 \\
\epsilon_{ik} &= \lambda_k + \sum_{n=1}^{N}\sigma_{s_n k}I_{s_n=i}
\end{aligned} \tag{44}$$

Computing γ^g

$$\begin{aligned}
\mathcal{L}_{\gamma^g} &= \log\Gamma(\sum_{i=1}^{K^g}\alpha_i^g) - \sum_{i=1}^{K^g}\log\Gamma(\alpha_i^g) + \sum_{i=1}^{K^g}(\alpha_i^g-1)(\Psi(\gamma_i^g) - \Psi(\sum_{j=1}^{K^g}\gamma_j^g)) + \sum_{n=1}^{N}\sum_{i=1}^{K^g}\xi_{n0}\phi_{ni}^g(\Psi(\gamma_i^g) - \Psi(\sum_{j=1}^{K^g}\gamma_j^g)) \\
&- \ [\, \sum_{i=1}^{K^g}(\gamma_i^g-1)(\Psi(\gamma_i^g) - \Psi(\sum_{j=1}^{K^g}\gamma_j^g)) + \log\Gamma(\sum_{i=1}^{K^g}\gamma_i^g) - \sum_{i=1}^{K^g}\log\Gamma(\gamma_i^g)\,] \\
\frac{\partial\mathcal{L}}{\partial\gamma_i^g} &= (\alpha_i^g-1)\Psi'(\gamma_i^g) + \sum_{n=1}^{N}\xi_{n0}\phi_{ni}^g\Psi'(\gamma_i^g) - (\gamma_i^g-1)\Psi'(\gamma_i^g) - \Psi(\gamma_i^g) + \Psi(\gamma_i^g) \\
&= (\alpha_i^g-1) + \sum_{n=1}^{N}\xi_{n0}\phi_{ni}^g - (\gamma_i^g-1) = 0
\end{aligned} \tag{45}$$

$$\gamma_i^g = \alpha_i^g + \sum_{n=1}^{N}\phi_{ni}^g\xi_{n0} \tag{46}$$

Computing γ_{ik}^l

$$\begin{aligned}
\mathcal{L}_{\gamma^l} &= \sum_{i=1}^{T+S-1}[\, \log\Gamma(\sum_{j=1}^{K^l}\alpha_j^l) - \sum_{k=1}^{K^l}\log\Gamma(\alpha_k^l) + \sum_{k=1}^{K^l}(\alpha_k^l-1)(\Psi(\gamma_{ik}^l) - \Psi(\sum_{j=1}^{K^l}\gamma_{ij}^l))\,] \\
&\quad + \sum_{n=1}^{N}\sum_{k=1}^{K^l}\sum_{i=1}^{T+S-1}\phi_{nk}^l\sigma_{ni}\xi_{n1}(\Psi(\gamma_{ik}^l) - \Psi(\sum_{h=1}^{K^l}\gamma_{ih}^l)) \\
&- \sum_{i=1}^{T+S-1}[\, \sum_{k=1}^{K^l}(\gamma_{ik}^l-1)(\Psi(\gamma_{ik}^l) - \Psi(\sum_{j=1}^{K^l}\gamma_{ij}^l)) + \log\Gamma(\sum_{j=1}^{K^l}\gamma_{ij}^l) - \sum_{k=1}^{K^l}\log\Gamma(\gamma_{ik}^l)\,] \\
\frac{\partial\mathcal{L}}{\partial\gamma_{ik}^l} &= (\alpha_k^l-1)\Psi'(\gamma_{ik}^l) + \sum_{n=1}^{N}\phi_{nk}^l\sigma_{ni}\xi_{n1}\Psi'(\gamma_{v_n k}^l) - (\gamma_{ik}^l-1)\Psi'(\gamma_{ik}^l) - \Psi(\gamma_{ik}^l) + \Psi(\gamma_{ik}^l) \\
&= (\alpha_k^l-1) + \sum_{n=1}^{N}\sigma_{ni}\xi_{n1}\phi_{nk}^l - (\gamma_{ik}^l-1) = 0
\end{aligned} \tag{47}$$

$$\gamma^l_{ik} = \alpha^l_k + \sum_{n=1}^{N} \phi^l_{nk}\sigma_{ni}\xi_{n1} \tag{48}$$

Computing γ^m

$$\begin{aligned}
\mathcal{L}_{\gamma^m} &= \sum_{i=1}^{T+S-1} [\, \log\Gamma(\sum_{k=0}^{1}\alpha^m_k) - \sum_{k=0}^{1}\log\Gamma(\alpha^m_k) + \sum_{k=0}^{1}(\alpha^m_k - 1)(\Psi(\gamma^m_{ik}) - \Psi(\sum_{j=1}^{K^l}\gamma^m_{ij}))\,] \\
&+ \sum_{n=1}^{N}\sum_{k=0}^{1}\sum_{i=1}^{T+S-1} \xi_{nk}\sigma_{ni}(\Psi(\gamma^m_{ik}) - \Psi(\sum_{j=0}^{1}\gamma^m_{ij})) \\
&- \sum_{i=1}^{T+S-1} [\, \sum_{k=0}^{1}(\gamma^m_{ik} - 1)(\Psi(\gamma^m_{ik}) - \Psi(\sum_{j=0}^{1}\gamma^m_{ij})) + \log\Gamma(\sum_{k=0}^{1}\gamma^m_{ik}) - \sum_{j=0}^{1}\log\Gamma(\gamma^m_{ij})\,] \qquad (49) \\
\frac{\partial\mathcal{L}}{\partial\gamma^m_{ik}} &= (\alpha^m_k - 1)\Psi'(\gamma^m_{ik}) + \sum_{n=1}^{N}\xi_{nk}\Psi'(\gamma^m_{ik})\sigma_{ni} - (\gamma^m_{ik} - 1)\Psi'(\gamma^m_{ik}) - \Psi(\gamma^m_{ik}) + \Psi(\gamma^m_{ik}) = 0 \\
\gamma^m_{ik} &= \alpha^m_k + \sum_{n=1}^{N}\xi_{nk}\sigma_{ni} \qquad (50)
\end{aligned}$$

Computing ρ

$$\begin{aligned}
\mathcal{L}_{\rho_{n0}} &= \sum_{n=1}^{N}\sum_{k=0}^{1}\rho_{nk}\log\zeta_k + \sum_{n=1}^{N}[\sum_{k=1}^{K^g}\phi^g_{nk}\xi_{n0}\rho_{n0}(\Psi(\gamma^g_k) - \Psi(\sum_{j=1}^{K^g}\gamma^g_j)) \; + \; \sum_{k=1}^{K^g}\sum_{j=1}^{V}\phi^g_{nk}\xi_{n0}\rho_{n0}\log\beta^g_{kj}I_{w_n=j} \\
&- \sum_{n=1}^{N}\sum_{k=0}^{1}\rho_{nk}\log\rho_{nk} \\
\frac{\partial\mathcal{L}}{\partial\rho_{n0}} &= \log\zeta_0 + \sum_{k=1}^{K^g}\phi^g_{nk}\xi_{n0}(\Psi(\gamma^g_k) - \Psi(\sum_{j=1}^{K^g}\gamma^g_j)) \; + \; \sum_{k=1}^{K^g}\sum_{j=1}^{V}\phi^g_{nk}\xi_{n0}\log\beta^g_{kj}I_{w_n=j} \\
&\quad -1 - \log\rho_{n0} \\
\rho_{n0} &\propto \exp(\log\zeta_0 + \sum_{k=1}^{K^g}\phi^g_{nk}\xi_{n0}(\Psi(\gamma^g_k) - \Psi(\sum_{j=1}^{K^g}\gamma^g_j)) \; + \; \sum_{k=1}^{K^g}\sum_{j=1}^{V}\phi^g_{nk}\xi_{n0}\log\beta^g_{kj}I_{w_n=j}) \qquad (51)
\end{aligned}$$

$$\begin{aligned}
\mathcal{L}_{\rho_{n1}} &= \sum_{n=1}^{N}\sum_{k=0}^{1}\rho_{nk}\log\zeta_k + \sum_{n=1}^{N}[\,\sum_{k=1}^{K^g}\phi^g_{nk}\xi_{n0}\rho_{n1}(\Psi(\gamma^g_k) - \Psi(\sum_{j=1}^{K^g}\gamma^g_j)) \;+\; \sum_{k=1}^{K^g}\sum_{j=1}^{V}\phi^g_{nk}\xi_{n0}\rho_{n1}\log\beta^g_{kj}I_{w_n=j} \\
&\quad \sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1}\phi^l_{nk}\sigma_{nj}\xi_{n1}\rho_{n1}(\Psi(\gamma^l_{jk}) - \Psi(\sum_{h=1}^{K^l}\gamma^l_{jh})) \;+\; \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi^l_{nk}\xi_{n1}\rho_{n1}\log\beta^l_{kj}I_{w_n=j}\;] \\
\frac{\partial\mathcal{L}}{\partial\rho_{n1}} &= \log\zeta_1 + \sum_{k=1}^{K^g}\phi^g_{nk}\xi_{n0}(\Psi(\gamma^g_k) - \Psi(\sum_{j=1}^{K^g}\gamma^g_j)) \;+\; \sum_{k=1}^{K^g}\sum_{j=1}^{V}\phi^g_{nk}\xi_{n0}\log\beta^g_{kj}I_{w_n=j} \\
&\quad \sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1}\phi^l_{nk}\sigma_{nj}\xi_{n1}(\Psi(\gamma^l_{jk}) - \Psi(\sum_{h=1}^{K^l}\gamma^l_{jh})) \;+\; \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi^l_{nk}\xi_{n1}\log\beta^l_{kj}I_{w_n=j} - 1 - \log\rho_{n1} \\
\rho_{n1} &\propto \exp(\log\zeta_1 + \sum_{k=1}^{K^g}\phi^g_{nk}\xi_{n0}(\Psi(\gamma^g_k) - \Psi(\sum_{j=1}^{K^g}\gamma^g_j)) \;+\; \sum_{k=1}^{K^g}\sum_{j=1}^{V}\phi^g_{nk}\xi_{n0}\log\beta^g_{kj}I_{w_n=j} \\
&\quad \sum_{k=1}^{K^l}\sum_{j=1}^{T+S-1}\phi^l_{nk}\sigma_{nj}\xi_{n1}(\Psi(\gamma^l_{jk}) - \Psi(\sum_{h=1}^{K^l}\gamma^l_{jh})) \;+\; \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi^l_{nk}\xi_{n1}\log\beta^l_{kj}I_{w_n=j}) \qquad (52)
\end{aligned}$$

B.1 M step

Computing α^g

$$\mathcal{L}_{\alpha^g} = \sum_{d=1}^{M}\log\Gamma(\sum_{j=1}^{K^g}\alpha^g_j) - \sum_{i=1}^{K^g}\log\Gamma(\alpha^g_i) + \sum_{i=1}^{K^g}(\alpha^g_i - 1)(\Psi(\gamma^g_{di}) - \Psi(\sum_{j=1}^{K^g}\gamma^g_{dj})) \qquad (53)$$

$$\frac{\partial\mathcal{L}}{\partial\alpha^g_i} = M[\;\Psi(\sum_{j=1}^{K^g}\alpha^g_j) - \Psi(\alpha^g_i)\;] + \sum_{d=1}^{M}(\Psi(\gamma^g_{di}) - \Psi(\sum_{j=1}^{K^g}\gamma^g_{dj})) \qquad (54)$$

Computing α^l

$$\mathcal{L}_{\alpha^l} = \sum_{d=1}^{M}\sum_{i=1}^{T+S-1}[\;\log\Gamma(\sum_{j=1}^{K^l}\alpha^l_j) - \sum_{k=1}^{K^l}\log\Gamma(\alpha^l_k) + \sum_{k=1}^{K^l}(\alpha^l_k - 1)(\Psi(\gamma^l_{dik}) - \Psi(\sum_{j=1}^{K^l}\gamma^l_{dij}))\;] \qquad (55)$$

$$\frac{\partial\mathcal{L}}{\partial\alpha^l_k} = M(T+S-1)(\Psi(\sum_{j=1}^{K^l}\alpha^l_j) - \Psi(\alpha^l_k)) + \sum_{d=1}^{M}\sum_{i=1}^{T+S-1}(\Psi(\gamma^l_{dik}) - \Psi(\sum_{j=1}^{K^l}\gamma^l_{dij})) \qquad (56)$$

Computing α^m

$$\mathcal{L}_{\alpha^m} = \sum_{d=1}^{M}\sum_{i=1}^{T+S-1}[\;\log\Gamma(\sum_{j=0}^{1}\alpha^m_j) - \sum_{k=0}^{1}\log\Gamma(\alpha^m_k) + \sum_{k=0}^{1}(\alpha^m_k - 1)(\Psi(\gamma^m_{dik}) - \Psi(\sum_{j=0}^{1}\gamma^m_{dij}))\;] \qquad (57)$$

$$\frac{\partial\mathcal{L}}{\partial\alpha^m_k} = M(T+S-1)(\Psi(\sum_{j=0}^{1}\alpha^m_j) - \Psi(\alpha^m_k)) + \sum_{d=1}^{M}\sum_{i=1}^{T+S-1}(\Psi(\gamma^m_{dik}) - \Psi(\sum_{j=0}^{1}\gamma^m_{dij})) \qquad (58)$$

Computing λ

$$\mathcal{L}_{\lambda} = \sum_{d=1}^{M}\sum_{i=1}^{S}[\,\log\Gamma(\sum_{j=1}^{T+S-1}\lambda_j) - \sum_{k=1}^{T+S-1}\log\Gamma(\lambda_k) \;+ \sum_{k=1}^{T+S-1}(\lambda_k - 1)(\Psi(\epsilon_{dik}) - \Psi(\sum_{j=1}^{T+S-1}\epsilon_{dij}))\,] \tag{59}$$

$$\frac{\partial\mathcal{L}}{\partial\lambda_k} = MS(\Psi(\sum_{j=1}^{T+S-1}\lambda_j) - \Psi(\lambda_k)) + \sum_{d=1}^{M}\sum_{i=1}^{S}(\Psi(\epsilon_{dik}) - \Psi(\sum_{j=1}^{T+S-1}\epsilon_{dij})) \tag{60}$$

Computing β^g

$$\mathcal{L}_{\beta^g} = \sum_{d=1}^{M}\sum_{n=1}^{N}\sum_{k=1}^{K^g}\sum_{j=1}^{V}\phi^g_{dnk}\xi_{dn0}\rho_{dn0}\log\beta^g_{kj}I_{w_{dn}=j} \tag{61}$$

$$\max_{\beta^g_{ij}}\mathcal{L} \quad s.t.\sum_{j=1}^{V}\beta^g_{ij} = 1$$

$$\beta^g_{ij} \propto \sum_{d=1}^{M}\sum_{n=1}^{N}\phi^g_{dni}\xi_{dn0}\rho_{dn0}w^j_{dn} \tag{62}$$

Computing β^l

$$\begin{aligned}\mathcal{L}_{\beta^l} &= \sum_{d=1}^{M}\sum_{n=1}^{N}\sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi^l_{dnk}\xi_{dn1}\rho_{dn0}\log\beta^l_{kj}I_{w_{dn}=j}\;] \\ &+ \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi^l_{dnk}\xi_{dn1}\rho_{dn1}\log\beta^l_{kj}I_{w_{dn}=j} \;+ \sum_{k=1}^{K^l}\sum_{j=1}^{V}\phi^l_{dnk}\xi_{dn0}\rho_{dn1}\log\beta^l_{kj}I_{w_{dn}=j}\;] \\ \beta^l_{ij} &\propto \sum_{d=1}^{M}\sum_{n=1}^{N}\phi^l_{dni}(\xi_{dn1}\rho_{dn0} + \xi_{dn1}\rho_{dn1} + \xi_{dn0}\rho_{dn1})w^j_{dn}\end{aligned} \tag{63}$$

C Sample Questions

```
There are 2 sections A and B. Please answer both sections.

This will be evaluated automatically using a script, kindly write your answer
after the ":" symbol only

SECTION A: Find the odd one out.
--------------------------------
Each word set in the questions below contains terms or words observed in code
statements (or comments) related to a particular purpose.

There is one word that does not belong to the set. Can you guess which one
it is ? Can you guess the purpose or assign a name to the set ?

** Hint:
---------------
The words in a set are typically sorted in order of relevance
```

```
to the purpose, except for the odd word that is introduced at a random
position index.

========================================================
Instruction: Pick the index of the odd word, and give a name to the set.
========================================================

Words set 1
-----------
1. stats
2. accumulator
3. walker
4. tree
5. acc
6. parse

ANSWER [give the index]: 6
LABEL [give a name to the set]: stats


Words set 2
-----------
1. cleanup
2. time
3. millis
4. now
5. completed
6. match

ANSWER [give the index]: 1
LABEL [give a name to the set]: time

======================================================================

SECTION B: Match the statement set provided with the most relevant word set
and the least relevant word set from the 4 choices given

Each statement set in the questions below contains code statements (or comments)
that reflect similar or related purpose, but sampled from multiple
files in a single application. Can you guess the purpose or assign a name
to the set ?

** Each word set listed in the choices contains terms or words observed in code
statements (or comments) related to a particular purpose (the words are
sorted in order of importance to the purpose)

Can you guess which word set from the choices is the most relevant match for
the statement set ? Can you identify the least relevant match ?

========================================================
Instruction: Pick the index of the most relevant and least relevant choices,
and give a name to the statement set.
```

```
========================================================

-----------------------------------------------------------
Statements set 1
-----------------------------------------------------------
 * Proxy to Cursor.getCursorImpl()
 * Proxy to EnvironmentConfig.setTxnReadCommitted()
 * Proxy to EnvironmentConfig.cloneConfig()
 * Proxy to EnvironmentMutableConfig.validateParams.
 * Proxy to DatabaseConfig.match(DatabaseConfig()


Choices
--------
1. stats accumulator walker tree acc
2. parse error enabled the get
3. pool methods versions large buffers
4. environment config txn env properties

Most relevant [give the index]: 4
Least relevant [give the index]: 1
LABEL [give a name to the set]: proxy config


-----------------------------------------------------------
Statements set 2
-----------------------------------------------------------
 exactSearch, lockType, bin.getLsn(index));
 if (lockResult.getLockGrant() != LockGrantType.DENIED) {
  return lockResult;
 * Try a non-blocking lock first, to avoid unlatching.  If the default
 lockResult = locker.nonBlockingLock


Choices
--------
1. match exact both equal get
2. lock owner waiter type grant
3. time millis now completed cleanup
4. parse error enabled the get

Most relevant [give the index]: 2
Least relevant [give the index]: 3
LABEL [give a name to the set]: search lock
```

D Stop-Words List

a a’s able about above according accordingly across actually after afterwards again against ain’t all allow allows almost alone along already also although always am among amongst an and another any anybody anyhow anyone anything anyway anyways anywhere apart appear appreciate appropriate are aren’t around as aside ask asking associated at available away awfully b be became because become becomes becoming been before beforehand behind being believe below beside besides best better between beyond both brief but by c c'mon c's came can can't cannot cant cause causes certain certainly changes clearly co com come comes concerning consequently consider considering contain containing contains corresponding could couldn't course currently d definitely described despite did didn't different do does doesn't doing don't done down downwards during e each edu eg eight either else elsewhere enough entirely especially et etc even ever every everybody everyone everything everywhere ex exactly example except f far few fifth first five followed following follows for former formerly forth four from further furthermore g getting given gives go goes going gone got gotten greetings h had hadn't happens hardly has hasn't have haven't having he he's hello help hence her here here's hereafter hereby herein hereupon hers herself hi him himself his hither hopefully how howbeit however i i'd i'll i'm i've ie if ignored immediate in inasmuch inc indeed indicate indicated indicates inner insofar instead into inward is isn't it it'd it'll it's its itself j just k keep keeps kept know knows known l last lately later latter latterly least less lest let let's like liked likely little look looking looks ltd m mainly many may maybe me mean meanwhile merely might more moreover most mostly much must my myself n name namely nd near nearly necessary need needs neither never nevertheless new next nine no nobody non none noone nor normally not nothing novel now nowhere o obviously of off often oh ok okay old on once one ones only onto or other others otherwise ought our ours ourselves out outside over overall own p particular particularly per perhaps placed please plus possible presumably probably provides q que quite qv r rather rd re really reasonably regarding regardless regards relatively respectively right s said same saw say saying says second secondly see seeing seem seemed seeming seems seen self selves sensible sent serious seriously seven several shall she should shouldn't since six so some somebody somehow someone something sometime sometimes somewhat somewhere soon sorry specified specify specifying still sub such sup sure t t's take taken tell tends th than thank thanks thanx that that's thats the their theirs them themselves then thence there there's thereafter thereby therefore therein theres thereupon these they they'd they'll they're they've think third this thorough thoroughly those though three through throughout thru thus to together too took toward towards tried tries truly try trying twice two u un under unfortunately unless unlikely until unto up upon us use used useful uses using usually uucp v value various very via viz vs w want wants was wasn't way we we'd we'll we're we've welcome well went were weren't what what's whatever when whence whenever where where's whereafter whereas whereby wherein whereupon wherever whether which while whither who who's whoever whole whom whose why will willing wish with within without won't wonder would would wouldn't x y yes yet you you'd you'll you're you've your yours yourself yourselves z zero abstract continue for new switch assert default goto package synchronized boolean do if private this break double implements protected throw byte else import public throws case enum instanceof return transient catch extends int short try char final interface static void class finally long strictfp volatile const float native super while org apache lucene java lang true false null xalan er string code ref exception error type br element license sax dom param path document attribute xsl object objects function functions func method ljava anewarray field invokevirtual xpath lorg getfield utils templates invokespecial init handler util putfield iterator buffer vector runtime javax base elem stylesheet invokeinterface context axis template symbol checkcast invokestatic source processor extension extensions script constructor invoke debug xsltc append stream impl getstatic expression factory print file hashtable security ljavax saxdtm axes support namespace xslt stack result loader prefix system compiler resolver manager size properties current content table text equals create tree root list array index expr serializer locator parser resource option invalid supported attrib illegal found length variable start iterators locale version args integer reader bundle listener putstatic suballocated message format println writer step call token match test walker throwable expanded property pool serialization configuration internal messages data filter patterns fast left wrapper software