Master Thesis in Statistics and Data Mining

Automated Bug Report Routing

Caroline Svahn

Division of Statistics
Department of Computer and Information Science
Linköping University

Supervisor
Mattias Villani

Examiner
Anders Nordgaard

Abstract

As the software industry grows larger by the minute, the need for automated solutions within bug report management is on the rise. Although some research has been conducted in the area of bug handling, new, faster or more precise approaches are yet to be developed. A bug report typically contains a free text observations field where the issue can be described by a human. Research regarding processing of this type of field is extensive, however, bug reports are often accompanied with system log files which have been given less attention so far. In the 4G LTE telecommunications network, the available system log files are many and several are likely to aid the routing of bug reports. In this thesis, one system log file was chosen to be evaluated; the alarm log. The alarm logs are time series count data containing alarms raised by the system. The alarm log data have been pre-processed with data mining techniques. The Apriori algorithm has been used to mine for specific alarms and alarming objects which indicates that the bug report should be solved by a particular developer group. We extend the Apriori algorithm to a temporal setting by using a customised time dependent confidence measure. To further mine for interesting sequences of events in the logs, the sequence mining approach SPADE has been used. The extracted class-associated sequences from both pre-processing approaches are transformed into binary features possible to use as predictors in any prediction model.

The results have been evaluated by predicting the correct developer group with two different methods; logistic regression and DO-probit. Logistic regression was regularised with the elastic net penalty to avoid computational issues as well as handling the sparse covariate set. DO-probit was used with a horseshoe prior; it is well suited for the sparse covariate regression problem as it is customised to obtain signals in sparse, noisy data. The results indicate that a data mining approach for processing alarm logs is promising.

The results show that the rules obtained with the Apriori mining process are suitable for mining the alarm logs as most binary representations of the rules used as covariates in logistic regression are kept in the equations for the expected classes with strongly positive coefficients. Although, the overall improvement in accuracy from using the alarms logs in addition to the learned topics from free text fields is modest, the alarm logs are concluded to be a good complement to the free text information as some Apriori covariates appears to be better suited to predict some classes than some topics.

Acknowledgments

My greatest gratitude to my supervisor, Mattias Villani, for the thorough supervision and for making it easy to ask stupid questions. Thanks for the freedom I have had and for showing me the meaning of the word "humble".

Thanks to my opponent, Araya, for giving a thorough opposition.

I would like to thank my Ericsson supervisors Henrik Rydén and Björn Magnusson for good guidance within the telecom area.

I also extend my gratitude to Daniel Nilsson and Sixten Johansson for great help with understanding of the data, and to everyone at Ericsson Research for making me feel welcome. Thanks to my fellow statistician Maxime for the chats when none of us had the energy to keep on going, and to Martina for being the best lab partner anyone can wish for.

A special thanks to Leif Jonsson for taking the time to get me started with the Java code base and for answering all my questions.

Finally, I want to thank Mårten for keeping me from spiraling during the critical moments.

Contents

Abstract iii

Acknowledgments iv

Contents v

List of Figures vi

List of Tables vii

1 Introduction 1
1.1 Motivation . . . 1
1.2 Aim . . . 2

2 Data 3

3 Method 5
3.1 Feature construction . . . 5
3.2 Predictive modeling . . . 11
3.3 Evaluation . . . 14

4 Results 17
4.1 Feature construction . . . 17
4.2 Choice of α and λ . . . 21
4.3 Predictive performance . . . 21
4.4 Coefficient evaluation . . . 24

5 Discussion 35
5.1 Results . . . 35
5.2 Method . . . 37

6 Conclusion 39

Bibliography 40

A Confusion matrices 42

B MCMC chains of log-posteriors 45

C Results for topics as covariate set 47

List of Figures

2.1 Distribution of the response variable (MHO). 4

3.1 Timeline describing importance of log data 9
3.2 Weights for time elapse. 9
3.3 Examples of weights by time span within a pattern r. 10

4.1 Comparison of accuracy for the Apriori item construction options. 18
4.2 The DO-probit log-posterior for the first four folds using Apriori covariates, version 1. 18
4.3 The DO-probit log-posterior for the first four folds using Apriori covariates, version 2. 19
4.4 Misclassification by α for all models. 21
4.5 Beta coefficients of logistic regression model with Apriori rules as covariates. . . . 25
4.6 Beta coefficients of DO-probit model with Apriori rules as covariates. 26
4.7 Beta coefficients of logistic regression model with Apriori rules and topics as covariates. 27
4.8 Beta coefficients of DO-probit model with Apriori rules and topics as covariates. . 28
4.9 Beta coefficients of logistic regression model with SPADE sequences as covariates. 29
4.10 Beta coefficients of DO-probit model with SPADE sequences as covariates. 30
4.11 Beta coefficients of logistic regression model with SPADE sequences and topics as covariates. 31
4.12 Beta coefficients of DO-probit model with SPADE sequences and topics as covariates. 32
4.13 Difference in topics coefficients in DO-probit model, classes 1-6. 33
4.14 Difference in topics coefficients in DO-probit model, classes 7-12. 33
4.15 Difference in topics coefficients in DO-probit model, classes 13-14. 34

B.1 The DO-probit log-posterior for the first four folds using Apriori rules and topics. 45
B.2 The DO-probit log-posterior for the first four folds using SPADE sequences. 46
B.3 The DO-probit log-posterior for the first four folds using SPADE sequences and topics. 46

C.1 Misclassification rate for different values of α for observations field topics. 47
C.2 Beta coefficients of logistic regression model with observations field topics as covariates. 48
C.3 Beta coefficients of DO-probit model with observations field topics as covariates. . 49
C.4 The DO-probit log-posterior for the first four folds using the topics as covariates. . 49

List of Tables

2.1 Structure of one alarm log. 3
2.2 Description of alarm data. 4

3.1 Confusion matrix example for the binary classification problem. 16

4.1 Class distribution of confident rules, Apriori 19
4.2 Class distribution of final rules, Apriori 20
4.3 Class distribution of confident sequences, SPADE 20
4.4 Class distribution of final sequences, SPADE 20
4.5 Sensitivity by class and covariate set. 22
4.6 Specificity by class and covariate set. 23
4.7 Overview of the models. 24

A.1 Confusion matrix for Apriori rules, logistic regression 42
A.2 Confusion matrix for Apriori rules, DO-probit 42
A.3 Confusion matrix for Apriori rules and topics, logistic regression 43
A.4 Confusion matrix for Apriori rules and topics, DO-probit 43
A.5 Confusion matrix for SPADE sequences, logistic regression 43
A.6 Confusion matrix for SPADE sequences, DO-probit 44
A.7 Confusion matrix for SPADE sequences and topics, logistic regression 44
A.8 Confusion matrix for SPADE sequences and topics, DO-probit 44

C.1 Confusion matrix for topics, logistic regression 47
C.2 Confusion matrix for topics, DO-probit 48

1 Introduction

1.1 Motivation

The procedure of localising faults and bugs in software or programs is, despite rather thorough research in the area, very tedious. It is time consuming and expensive to debug manually and the importance of finding effective automatic techniques for fault detection is grand for large corporations with extensive code bases [20].

Parnin and Orso [18] discuss the benefits of automatic debugging tools. The conclusions are based on a study where a set of developers were given two tasks to perform, with and without an automated debugging tool. An interesting finding is that with manual debugging, developers tend to treat the symptoms or work around the problem rather than finding the core code error. An example is given as a bug introduced in the popular puzzle game Tetris; the configuration of the square shaped piece was given an invalid number of possible orientations. Debugging manually, several developers modified the rotation calculation instead of localising the actual bug, merely patching the code to get around the bug. None of the test objects using the debugging tool tried to correct the error in this manner. The article reasons about the trust issues of an automated tool; programmers lack faith in tools not supplying measures of uncertainty or when the process is too difficult to understand.

Open source bug report collections are often used for evaluating methods. Anvik, Hiew and Murphy [2] use machine learning and the repositories for Eclipse and Firefox to predict on developer level, i.e. where the target variable is a group of developers. They search for indicators of which types of reports that are usually assigned and solved by which developer with a two-phase solution. A supervised algorithm, Support Vector Machines, predicts the developer and a human determines whether a report is meaningful or not. With this method, they achieved 50% accuracy for the Eclipse data (49 classes) and 64% for the Firefox data (118 classes).

Attempting to calibrate for the uncertainty of the classifier, Kim et al. [14] propose a two-phase model with a multiple option ranked output on file level. If the prediction is estimated to be certain enough, the automated tool will be used, otherwise the bug report will be sent for manual inspection. The classifier was evaluated on the Firefox (118 classes) and Core (39 classes) repositories. The first step of the algorithm filtered out about 50% of the reports due to lack of confidence and the second phase (Naïve Bayes) managed to correctly classify in average 70% of the more certain reports.

Jonsson et al. [13] introduce a Bayesian approach to the problem at the developer level, i.e. where the target variable is a software component. The project addresses two issues mentioned by Parnin and Orso [18]; the reliability of the tool from the developer's point of view and the issue of understandability of the results. The authors present a supervised topic model, DOLDA which is a combination of the Diagonal Orthant Probit Model and the Latent Dirichlet Allocation model with a Bayesian Horseshoe regularisation prior. As Kim et al. [14], Jonsson et al. [13] also experiment with a two-phase approach to filter out the reports combined with too much uncertainty in their prediction, which are instead sent for manual inspection. The DOLDA model is evaluated on the Firefox and Eclipse open source bug repositories, as well as a corpus of bug reports with 26 classes from the telecommunication company Ericsson. The model outperforms alternative models in the literature and in addition offers quantification of prediction uncertainty as well as interpretable predictions using topics relateable to humans. The authors find a Bayesian approach to be suitable to handle the quantification of the uncertainty and they argue that they have found a preferable way to display the results to improve understandability. The model also offers both unstructured text and structured variables as input.

1.2 Aim

There is still a lot to wish for in the models proposed for localising bugs in systems; a lot of the issues concern unreliability of the automated tools. While the textual bug reports are indeed important for automated systems, there is typically a wealth of additional data sources to enhance the results. System log files are yet unexplored for this purpose despite considered useful when manually localising bugs in systems and are often easy to interpret by humans. To extract important features from unstructured alarms is a challenge for automated bug localisation systems. This thesis aims to construct and select essential features from alarm log files to find an improved classifier for automatic fault detection on the developer level. More distinctively, the objective is the following:

- Are alarm log files useful for automated bug report routing?
- Is association analysis an effective approach for constructing features from system log files?

The questions will be evaluated on bug reports from Ericsson's repository. The thesis will use Ericsson 4G LTE bug reports only, as the problem becomes too wide when including developer groups for older generations of telecommunications systems as well. 4G is chosen since earlier generations of technology will be phased out sooner.

2 Data

The dataset is a collection of alarm log files connected to bug reports from the telecommunications company Ericsson's 4G LTE network. Symptoms of bugs are detected internally by simulation in test environments or by customers, if the bug happened to pass all internal tests. In order to resolve these issues, bug reports are submitted and used to track what caused the issue. A bug report typically contains an observation field where the problem can be described in free text and similar details. Each report also contains information about which instance (*Modification Handling Office* or *MHO*) finally handled the error/bug, which will be used as the response variable. To aid the process of localising the bug, each report is issued with various (separate) log files. The log files are retrieved from the nodes in the radio access network and provide details about alarms, updates, crashes, restarts etcetera.

This thesis will focus on the alarm log files specifically. When a problem emerges in the radio base station which is not automatically solvable by the system, an alarm is raised. The time points of all alarms and when they ceased are stored in an event log for traceability. Alarms cease when the cause of the alarm is fixed. For each bug report, the alarm data is a several years long irregular time series giving the time points of all alarms raised by the system according to the structure given in Table 2.1.

Entry	Date	Time	Managed Object	Specific alarm
1	2013-01-01	00:00:01	Security	Password File Fault
2	2013-01-01	00:00:10	RiLink	Link Failure
⋮	⋮	⋮	⋮	⋮
1319	2016-12-12	15:42:09	EUtranCellTDD	Service unavailable

Table 2.1: Structure of one alarm log.

The logs are connected to individual base stations and they are rarely cleared, therefore, the log files are likely to cover and describe older problems in addition to the problem described by the bug report. There are 5891 bug reports at hand, and a summary of the variables in the alarm data expected to be of importance can be found in Table 2.2.

Response variable	Explanatory variables	Data type
MHO	Date	YYYY-MM-DD
	Time	HH:MM:SS
	Managed Object	Nominal, 121 levels
	Specific alarm	Nominal, 259 levels

Table 2.2: Description of alarm data.

The Managed Object (*MO*) is the origin of the alarm (software component). There are 121 distinct MOs. The MOs cannot be used directly for prediction since the alarm is unlikely to be the root cause, merely a symptom. The *specific alarm* is the label of the raised alarm. There are 259 different alarm labels in the data. *MHO* is the instance (e.g. developer group) which finally managed the bug report in question and it has 14 levels, distributed according to Figure 2.1.

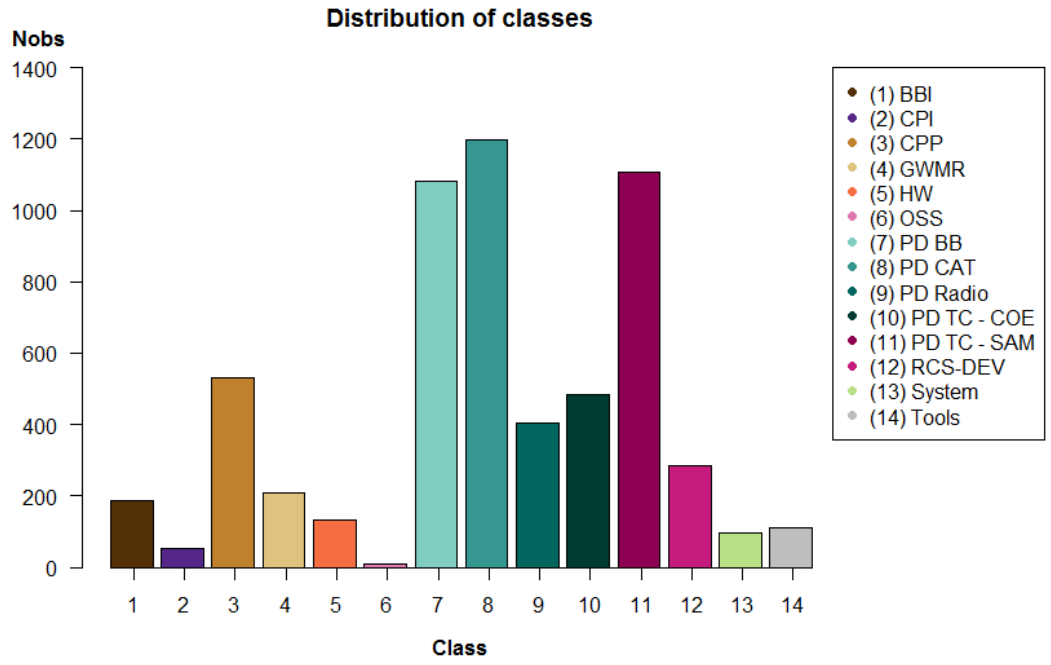


Figure 2.1: Distribution of the response variable (MHO).

The distribution is somewhat uneven; PD BB, PD CAT and PD TC - SAM are represented about twice the remainder of the classes while CPI and OSS constitutes a rather limited part of the dataset.

The free text field has been processed in previous studies [13] with Latent Dirichlet Allocation. The results are available and will constitute a baseline for the evaluation of the results in this thesis to determine whether the alarm logs can, in addition to the free text field, add to the precision.

3 Method

This chapter aims to describe the methods to be used in the thesis and how they will be applied to this problem.

3.1 Feature construction

Free text field processing

Since the topic means obtained by Jonsson et al. [13] will be used, the methodology behind Latent Dirichlet Allocation will be explained in this section.

Latent Dirichlet Allocation

The method for localising faults in software systems used by Jonsson et al. [13], Diagonal Othant Latent Dirichlet Allocation (DOLDA), was proposed by Magnusson et al. [15] and is a supervised topic model for high-dimensional classification. The DOLDA model combines the topic model, Latent Dirichlet Allocation (LDA), proposed by [4] for unsupervised learning on unstructured text in bug reports with the supervised Diagonal Orthant probit (DO probit) model described by [12].

LDA is an unsupervised probabilistic model for text classification. The method relies on the Dirichlet distribution. The probability density function of the Dirichlet distribution given $K \geqslant 2$ categories is of the form

$$f(x_1, \ldots, x_k; \alpha_1, \ldots, \alpha_k) = \frac{1}{B(\alpha)} \prod_{i=1}^{K} x_i^{\alpha_i - 1} \tag{3.1}$$

where

$$B(\alpha) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)} \tag{3.2}$$

and $\alpha_1, ..., \alpha_K$ are *concentration parameters* which control the probabilities of the distribution; for large values of α the resulting distribution concentrates around a single point as the variance decreases. For $\alpha < 1$, the Dirichlet distribution resembles the shape of a tub. For $\alpha = 1$,

the distribution is uniform over the unit simplex, $x_1 + x_2 + \cdots + x_k = 1$. The Dirichlet distribution can be seen as a multivariate generalisation of the Beta distribution [3].

Furthermore, LDA assigns individual words to a discrete set of topics, therefore, LDA also relies on the Multinomial distribution [4]. Its probability mass function is

$$f(x_1, \ldots, x_k; n, p_1, \ldots, p_k) = \frac{n!}{x_1! \cdots x_k!} p_1^{x_1} \cdots p_k^{x_k}, \tag{3.3}$$

where n is the number of trials, K is the number of possible outcomes and p_k is the probability of outcome k. The Multinomial distribution is a generalisation of the Binomial distribution [3]. The Multinomial distribution can be expressed as

$$f(x_1, \ldots, x_k; n, p_1, \ldots, p_k) = \frac{\Gamma(\sum_i x_i + 1)}{\prod_i \Gamma(x_i + 1)} \prod_{i=1}^{k} p_i^{x_i} \tag{3.4}$$

The resemblance to the Dirichlet distribution is evident, and the Dirichlet distribution is in fact a conjugate prior to the Multinomial distribution - Multinomial data with a Dirichlet distributed prior will give a Dirichlet distributed posterior.

In LDA, each *document* in a *corpus* can be considered as represented by a mixture of underlying *topics*, where a specific distribution over the vocabulary represents and distinguishes topics. Thus, the goal with the learning is to obtain the topic specific word distributions, topic assignments/indicators for all words in all documents and finally, document specific topic proportions. Given a fixed vocabulary of size V, D documents in a corpus, N words in document d and in total K topics, the process of generating a corpus from a topic model is

1. Simulate N $\sim Poisson(\boldsymbol{\lambda})$
2. For each topic $k = 1, \ldots, K$:
   a) Simulate a distribution over words $\phi_k \sim Dir_V(\gamma)$
3. For each document $d = 1, \ldots, D$:
4. Simulate topic proportions $\theta_d|\alpha \sim Dir_K(\alpha)$
   a) For $i = 1, \ldots, N$
      i. Simulate a topic assignment $z_{i,d}|\theta_d \sim Multinomial(\theta_d)$
      ii. Draw a word $w_{i,d}|z_{i,d}, \boldsymbol{\phi}_{z_{i,d}} \sim Multinomial(\boldsymbol{\phi}_{z_{i,d}})$

where $Dir(\cdot)$ is the Dirichlet distribution and $\boldsymbol{\phi}$ is the vector of word probabilities for each topic. The unknowns of the model and therefore the parameters we need to infer are the topic proportions for the documents ($\theta_{1:D}$), the topic assignments for the words ($z_{1:D}$) and the word distributions for all topics ($\phi_{1:K}$).
Since $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are unknowns, Dirichlet priors for all ϕ_i and all θ_d are introduced to obtain the joint posterior distribution for the hidden and observed variables:

$$p(\theta_{1:D}, z_{1:D}, \phi_{1:K}|w_{1:D}) \propto \prod_{i=1}^{K} p(\phi_i) \prod_{d=1}^{D} p(\theta_d) \left(\prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|\phi_{1:K}, z_{d,n}) \right) \tag{3.5}$$

$z_{1:D}$, $\theta_{1:D}$ and $\phi_{1:K}$ can then be obtained by Gibbs sampling [8]. *Partially Collapsed Gibbs sampling*, proposed in [16], will be used in this thesis to sample from the posterior distribution of $\theta_{1:D}$, $\phi_{1:K}$ and $z_{1:D}$. As explained in [16], partially collapsing makes it possible to do parallel sampling and to efficiently exploit document sparsity. The procedure uses the conjugacy property between the Dirichlet prior of $\theta_{1:D}$ and the multinomial distribution of $z_{1:D}$ to *collapse* (integrate out) $\theta_{1:D}$ [8]. The Gibbs sampling updates of the remaining parameters, $z_{1:D}$

and $\phi_{1:K}$, can be parallelized since the topic indicators $z_{1:D}$ in the different documents are independent given ϕ and $\phi_{1:K}$ are independent given the topic indicators. The following distribution is used for updating the topic indicators for a given document, and the topic indicator for different documents can be sampled independently in parallel:

$$p(z_i = j|\mathbf{z}_{-i}^{(d)}, \Phi, w_i) \propto \phi_{j,w} \cdot \left(n_{-i,j}^{(d_i)} + \alpha\right) \tag{3.6}$$

where α is the prior hyperparameter for θ and $n^{(d)}$ are the topic counts for each document and topic. Furthermore, the appropriate distribution for updating each topic in parallel is

$$\phi_k|z_{1:D} \sim Dir\left(n_k^{(\mathbf{w})} + \beta\right) \tag{3.7}$$

where β is a prior hyperparameter for ϕ and $n^{(w)}$ are topic indicator counts [16].

Since the topic means of a document d is in fact a vector of probabilities for each topic in document d, LDA is a multi-assigning labeler; each document is summarised as a probability distribution over the K topics. This property makes the topic model useful for construction of features for supervised classification models.

Log data processing

As previously mentioned, the alarm logs are extensive and are likely to contain many alarms not related to the issue which lead to the filing of the report. Thus, some pre-processing of the data is needed, aiming to find a way to filter out common alarms present in most logs as well as outdated alarms.

Apriori algorithm

Some alarms are very common and can be found in most logs. These alarms are rarely grounds for a trouble report and often refer to external issues or mishandling. These alarms are concluded to be of insignificant value. Due to this noise, a way to mine for important relations in the logs is needed. In this thesis, association analysis will be used. More distinctively, the Apriori algorithm.

The Apriori algorithm is presented by Agrawal and Srikant [1] and is typically used to obtain item association rules in high dimensional data. Each observation is called a *transaction*, and an *item* in this context is for instance the state of a nominal variable contained in one or more transactions. The algorithm generates supersets of items, or *candidates*, and search the database for supporting evidence of the relationship. The *size k* of an itemset is the number of items in the set. C_k is all candidates for itemsets of size k, the *support* for an itemset is the count of occurrences of the itemset in the database. L_k is the set of all candidates in C_k with support larger than a pre-specified threshold. For example, L_1 contains all single items which occurs at least $minsup$ times in the database. The procedure for finding frequent itemsets or *patterns* in a database $\mathcal{D}$ is defined in Algorithm 1.

Algorithm 1 Apriori Algorithm

```
1: procedure GENERATE L_k
2:     L_{k-1} = {large k-1-itemsets};
3:     for k; L_{k-1} ≠ ∅ do
4:         C_k = candidates(L_{k-1});
5:         for all transactions t ∈ D do
6:             C_t = subset(C_k, t);
7:             for all candidates c ∈ C_t do
8:                 c.count++;
9:         L_k = {c ∈ C_k | c.count ⩾ minsup}
```

As long as frequent patterns of size $k-1$ were found in the previous iteration or until the maximum length of a pattern is reached, candidates are generated for large frequent itemsets of size k. For every database entry t, we identify all candidates for k-large itemsets (the subset denoted by C_k) present in entry t and define this candidate subset as C_t. For each candidate in C_t, we add one to its support count (this as we have found evidence of the pattern in entry t). Finally, L_k is defined as all candidates in C_k occurring at least $minsup$ times. $candidates(L_{k_1})$ is defined as a function of two steps. In the first step, a self join is performed on L_{k-1} to get all possible combinations of k items and thereby all potential candidates for k large patterns (C_k). In the second step, C_k is pruned to only contain itemsets of k for which *all* subsets of $k-1$ items are frequent, i.e, are present in L_{k-1}. The general procedure of generating candidates is thereby defined by Algorithm 2, where rows $2-4$ is the first step explained and $5-8$ is the second step.

Algorithm 2 Candidate generation

```
1: procedure GENERATE C_k
2:     select p.item_1, p.item_2, ..., p.item_{k-1}, q.item_{k-1}
3:     from L_{k-1} as p, L_{k-1} as q
4:     where p.item_1 = q.item_1, ..., p.item_{k-2} = q.item_{k-2}, p.item_{k-1} < q.item_{k-1};
5:     for all itemsets c ∈ C_k do
6:         for all (k-1)-subsets s of c do
7:             if s ∉ L_{k-1} then
8:                 delete c from C_k;
```

The subset function given in Algorithm 1 traverses a hash-tree to find what entries t contain which itemsets c. The items in the transactions as well as the items in the candidate sets are stored lexically to achieve efficiency. Since an entry t needs to contain all items in the itemset c and since the itemset c is sorted lexically, the algorithm can match the items index by index; if the items of index i do not match, we know that entry t does not contain all items in c and we can proceed to the next c in C_t.

By default, Apriori mines for rules which can include any item in the database. However, using the Apriori algorithm to find frequent patterns connected to a specific MHO in the alarm logs, the only interesting rules will be rules containing the class variable MHO. Thus, only rules connected to specific MHOs will be regarded in this thesis.

In association analysis, three measures of importance are commonly used. The previously mentioned *support*, the *confidence* and the *lift* of a rule. The confidence is defined as

$$Conf(X \Rightarrow Y) = \frac{sup(X \cup Y)}{sup(X)} \tag{3.8}$$

where X is the right hand side of the rule, Y is the left hand side of the rule and sup is the support. As we restrict the rules to only imply MHOs, the left hand side can be any itemset and the right hand side will always be an MHO. The formula can be interpreted as the number of times the items in X has been seen in a bug report routed to MHO Y divided by the number of times the items in X has been seen overall, regardless of MHO routing. Furthermore, the lift measure is defined as

$$Lift(X \Rightarrow Y) = \frac{sup(X \cup Y)}{sup(X) \times sup(Y)} \tag{3.9}$$

The lift measure is an independence notion; if $Lift(X \Rightarrow Y) = 1$, X and Y are found together in a transaction the same number of times X and Y are found independently. This evidence would therefore imply that X and Y are in fact independent.

A modified confidence measure for time dependent rule mining

The traditional measurements of importance of rules in Apriori do not consider time and merely consider all events in a log to be of equal relevance. As time elapsed between log events are likely to be of importance, a modified confidence measure which considers the time aspect will hereby be proposed.

As each log file often cover several years of alarms, it is desirable to cut the logs to some specified window. Figure 3.1 gives a description of the relevance by time.

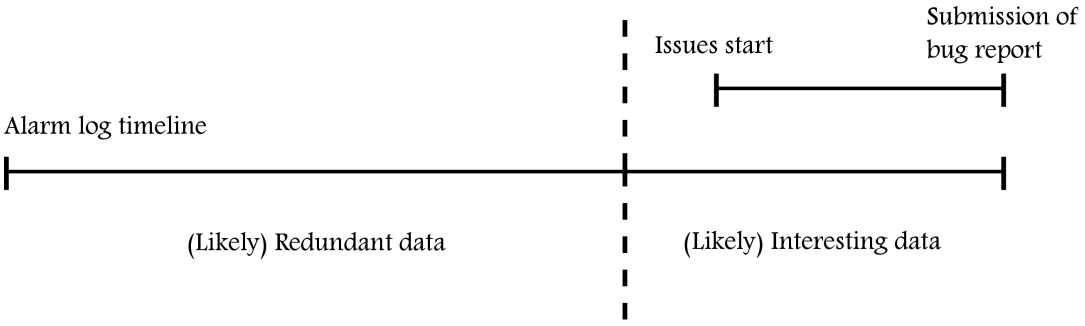


Figure 3.1: Timeline describing importance of log data

A window of 3-6 months is considered to be of use. Therefore, primarily, alarms risen 6 months or more since the trouble report was issued will be disregarded.

Furthermore, Apriori assumes all alarms in the evaluated *patterns* (the left hand side of the rules) to be of equal importance. Alarms risen a long time before the issue of the bug report (even within the chosen time span) are not as likely to help the prediction. Therefore, a time dependent weight term is added to the confidence measure given in the previous subsection. The importance dependence by time is considered, by area experts, to be approximated by

$$w_t = 1 - F(t), \quad t \in [0, 183] \tag{3.10}$$

where $F(t)$ is the cumulative distribution function (CDF) of the $Beta(a, b)$ distribution and t is the number of days between an alarm and the issue of a bug report. This gives continuous weights, $w_t \in [1, 0]$. The plot in Figure 3.2 describes an example given $Beta(8, 3)$.

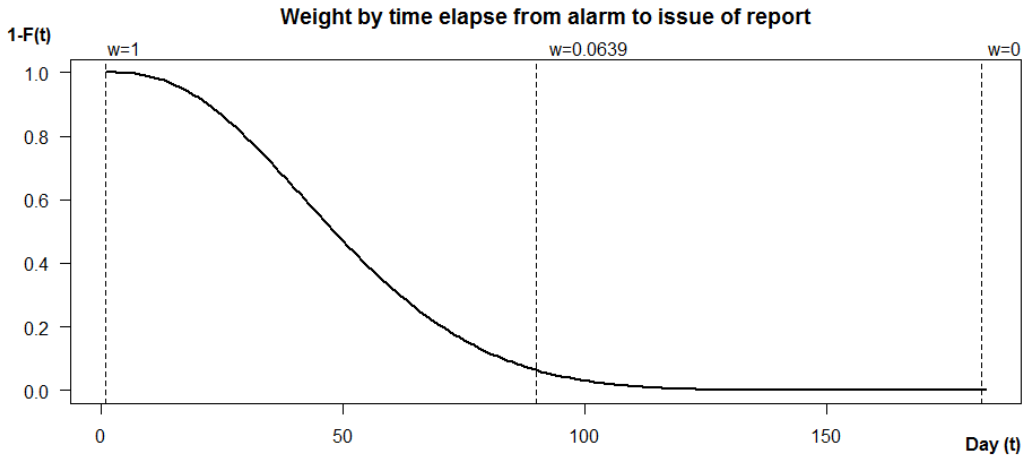


Figure 3.2: Weights for time elapse.

A specific alarm can occur multiple times in the same log. The Apriori algorithm is, however, a binary algorithm. The frequencies of specific alarms will therefore not be regarded,

merely the boolean of the presence in a specific log. Furthermore, the latest alarms are regarded as the most important, therefore, only the last occurring alarm of a specific alarm will be considered. Given that $\boldsymbol{\lambda}_{ij}$ is a vector of **all weights** for a specific alarm j in log i,

$$w_{ij} = max(\boldsymbol{\lambda}_{ij}), \quad \boldsymbol{\lambda}_{ij} = \text{ weights of alarm } j \text{ in log } i \tag{3.11}$$

w_{ij} is then the weight of a specific alarm j in a specific log i. The connection between alarms are also likely to depend on time elapsed between them. Therefore, a time difference weight is introduced to reduce the evidence provided by alarms raised within a wide interval in time:

$$\gamma_{ir} = 1 - (max(\mathbf{w}_{ir}) - min(\mathbf{w}_{ir})), \quad \mathbf{w}_{ir} = \text{ weights of alarms in pattern } r \text{ in log } i \tag{3.12}$$

For alarms in pattern r in log i within a narrow timespan, $\gamma_{ir} \to 1$. If the timespan of the alarms is wide, $\gamma_{ir} \to 0$. An example of different γ_{ir} for some pattern time spans can be seen in Figure 3.3.

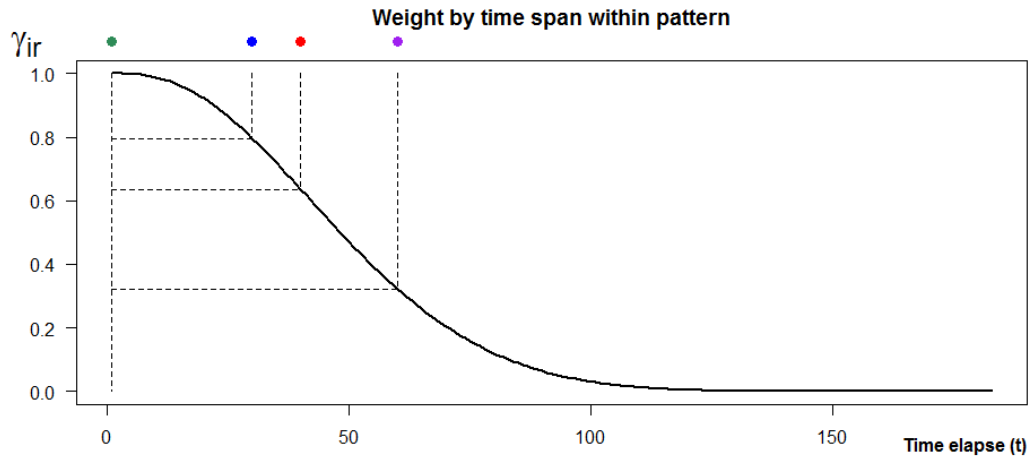


Figure 3.3: Examples of weights by time span within a pattern r.

The γ_{ir} weight is then multiplied to the weight of the latest alarm in the pattern:

$$\beta_{irc} = \gamma_{ir} \cdot max(\mathbf{w}_{ir}) \tag{3.13}$$

β_{irc} is then the final evidence of pattern r in log i with MHO c. If $\gamma_{ir} \to 0$, $\beta_{irc} \to 0$ regardless of the value of $max(\mathbf{w}_{ir})$. If $max(\mathbf{w}_{ir}) \to 0$, $\beta_{irc} \to 0$ regardless of the value of γ_{ir}. Therefore, $\beta_{irc} = 1$ if and only if all alarms in pattern r in log i were raised at the same day as the bug report was issued. $\sum_i \beta_{irc}$ is then used as a weighted count of pattern r which has been mapped to MHO c:

$$C_{rc} = \frac{\sum_i \beta_{irc}}{\sum_c \sum_i \beta_{irc}} \tag{3.14}$$

where $\sum_c \sum_i \beta_{irc}$ is the summed weights for pattern r in all logs for all c. C_{rc} aims to replace the classic confidence measure in association analysis, where n_r is the number of logs containing the pattern:

$$\text{If } \sum_c \sum_i \beta_{irc} \to n_r, \quad C_{rc} \to \text{ Standard confidence} \tag{3.15}$$

The MOs will be handled in the same manner as the alarms. For any raised alarm, both the specific alarm and the component alarming (the MO) will be considered as features in the

association analysis. The same weight will be assigned to the alarm and to the MO, and they will be handled independently in the analysis. As a first step, the data will not be considered sequential. Indicator variables for each alarm and each MO will be created before making use of association analysis to mine for important relations in data. Important connections between features and MHOs can then be used as features in a supervised classifier.

SPADE

While traditional association analysis only discover itemsets without concern to the order of events, temporal data mining enables pattern discovery. Therefore, to attempt to account for sequences of events in the alarm logs and not only time elapse between events, SPADE (Sequential PAttern Discovery using Equivalence classes), proposed by Zaki [21], will be used.

If we denote a set of m distinct *items* as $\mathcal{I}$ and a temporal *event* as an unordered occurrence of a subset of these events, a *sequence* is a chain of events. A sequence with events α_i is defined as $\mathcal{S} = (\alpha_1 \rightarrow \alpha_2 \cdots \rightarrow \alpha_q)$. Furthermore, the database $\mathcal{D}$ is a collection of transactions with unique input-sequence IDs *sid* and events *eid*. Denote X ($X \in \mathcal{S}$) in the sequence lattice as an *atom* and $\mathcal{L}(X)$ as an id-list giving the input-sequence ID and the event ID pairs (sid, eid) for all occurrences of the atom. In Algorithm 3, the generation of frequent sequences $\mathcal{F}$ is explained.

Algorithm 3 SPADE

1: **procedure** GENERATE $\mathcal{F}_k$
2: **for all** atoms $A_i \in \mathcal{S}$ **do**
3: $T_i = \varnothing$
4: **for all** atoms $A_j \in \mathcal{S}$, with $j \geqslant i$ **do**
5: $R = A_i \bigvee A_j$
6: $\mathcal{L}(R) = \mathcal{L}(A_i) \bigcap \mathcal{L}(A_j)$
7: **if** $\sigma(R) \geqslant min\_sup$ **then**
8: $T_i = T_i \bigcup \{R\}; \mathcal{F}_{|R|} = \mathcal{F}_{|R|} \bigcup \{R\}$

Any sequence $\mathcal{S}$ can be found as a temporal join of an atom and the lattice, giving all occurrences of the lattice after the atom. The id-list can then be updated, and X grows with each step of the algorithm. The sequences found to be frequent are taken to the next level for evaluating more extensive sequences. The process is repeated until there are no frequent sequences left to evaluate. The main difference from the previous approach, Apriori, is thus the respect taken to the event ID. The approach also supports arbitrary time gaps between event, and does not only mine for frequent consecutive sequences. As SPADE allows non-temporal relationships within atoms as well as temporal relationships, *equality* joins can also be performed to evaluate several items in the same event. The equality join is therefore performed to find all occurrences of, for instance, two items with the same sequence ID and event ID.

The Apriori and SPADE algorithms will be run with the R-packages `arules` and `arulesSequences` [9] by Hahsler et al., respectively.

3.2 Predictive modeling

This section will introduce the methods used for evaluating the predictive performance of the data mining algorithms.

Multinomial logistic regression

Logistic regression is a probabilistic classifier with a linear boundary. To obtain the response in terms of probability, a softmax transformation is performed. For a categorical response

variable $\mathcal{G}$ with $K > 2$ levels, the softmax transformed probability of the class variable $\mathcal{G} = l$ given a feature vector x for the multinomial case is modeled as

$$Pr(\mathcal{G} = l|x) = \frac{\exp(a_l)}{\sum_{k=1}^{K} \exp(a_k)}, \tag{3.16}$$

where $l = 1, \ldots, K$ and a_l are activations according to

$$a_l = \beta_{0,l} + \beta_l^T x. \tag{3.17}$$

The class-conditional parameters $\{\beta_{0,l}, \beta_l\}$ can be obtained by maximum likelihood estimation. The multinomial log-likelihood function is the following:

$$l(\{\beta_{0,l}, \beta_l\}_1^K) = \frac{1}{N} \sum_{i=1}^{N} \log\, p_{g_i}(x_i) \tag{3.18}$$

where $g_i \in \{1, 2, \ldots, K\}$ is the ith response and $p_{g_i}(x_i) = Pr(\mathcal{G} = g_i|x_i)$ the probability of the response g_i given the value of observation i and the parameters $\{\beta_{0,l}, \beta_l\}_l^K$. There is, however, no closed-form solution for obtaining the maximum likelihood estimate for multinomial logistic regression due to nonlinearity in the softmax function [6].

Elastic net regularisation

When the dimension of the feature vector is high, especially for $p \gg N$ or for highly correlated predictors, regularisation is useful to avoid overfitting. Furthermore, the parameters for class l needs to be constrained, since any values for $\{\beta_{0,l}, \beta_l\}_1^K$ will give identical probabilities as $\{\beta_{0,l} - c_0, \beta_l - c\}_1^K$. When regularising, a penalty factor is added to the maximum likelihood function when maximising. LASSO, proposed by Tibshirani [19], and Ridge, proposed by Hoerl and Kennard [11], are commonly used regularisation methods. The LASSO penalty is an addition to the loss function on the form

$$l_1 = -\lambda \sum_{j=1}^{p} |\beta_j| \tag{3.19}$$

and performs variable selection. The LASSO penalty does, however, have some shortcomings. For $p > N$, LASSO selects at most N variables. Furthermore, if a group of variables are highly correlated, LASSO selects one variable with little regard to the choice of which of the variables to choose. Finally, for the $N > p$ scenario, Ridge usually performs better in terms of prediction. The Ridge penalty is:

$$l_2 = -\lambda \sum_{j=1}^{p} \beta_j^2. \tag{3.20}$$

Ridge keeps all parameters, merely shrinks the coefficients to reduce redundancy. Thus, Ridge fails to produce a compressed model in terms of using as few variables as possible [22]. Elastic net, proposed by Zou and Hastie [22], offers a combination of LASSO and Ridge as the elastic net can perform both adjustable variable selection and coefficient shrinking with a parameter α which can be used as a bridge between LASSO and Ridge. Friedman et al. [6] adapt the elastic net to multinomial regression. The aim is then to solve the following:

$$max \left[\frac{1}{N} \sum_{i=1}^{N} \log p_{g_i}(x_i) - \lambda \sum_{l=1}^{K} P_\alpha(\beta_l) \right] \tag{3.21}$$

where $0 \leqslant \alpha \leqslant 1$ and P_α is the elastic net penalty factor combining l_1 and l_2:

$$P_\alpha(\beta) = (1-\alpha)\frac{1}{2}||\beta||_{l_2}^2 + \alpha||\beta||_{l_1} \tag{3.22}$$

$$= \sum_{j=1}^{p}\left[\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j|\right] \tag{3.23}$$

Note how the value of α determines the weights of the LASSO and Ridge penalties. For $\alpha \to 0$, the penalty approaches the Ridge penalty and for $\alpha \to 1$, it approaches the LASSO procedure. As the solution of this penalised maximum likelihood procedure cannot be obtained directly, Taylor expansion is used to approximate a quadratic function with a unique minimum. If we denote $\mathbf{Y}$ with the $N \times K$ indicator matrix with elements $y_{il} = I(g_i = l)$, the (unpenalised) log-likelihood is

$$l(\{\beta_{0,l}, \beta_l\}_1^K) = \frac{1}{N}\sum_{i=1}^{N}\left[\sum_{l=1}^{K} y_{il}(\beta_{0,l} + x_i^T\beta_l) - \log\left(\sum_{l=1}^{K} e^{\beta_{0,l}+x_i^T\beta_l}\right)\right] \tag{3.24}$$

As the multinomial case is non-trivial, the simplest way to optimise the log-likelihood is to let the coefficients vary for one class at a time with a procedure called *partial Newton steps*. This procedure and the Taylor expansion form a partial quadratic approximation of the log-likelihood for l as

$$l_{Ql}(\{\beta_{0,l}, \beta_l\}) = -\frac{1}{2N}\sum_{i=1}^{N} w_{il}(z_{il} - \beta_{0l} - x_i^T\beta_l)^2 + C(\{\tilde{\beta}_{0,l}, \tilde{\beta}_l\}) \tag{3.25}$$

where $C(\{\tilde{\beta}_{0,l}, \tilde{\beta}_l\})$ is the remainder term of the Taylor expansion. The approximation is initialised with a value of the penalty factor λ. For a class l, the partial quadratic approximation for l is updated using the current estimated parameters $\tilde{\beta}_{0,l}$ and $\tilde{\beta}_l$. The co-ordinate descent algorithm is then applied to solve

$$max\left[l_{Ql}(\{\beta_{0,l}, \beta_l\}) - \lambda P_\alpha(\beta_l)\right] \tag{3.26}$$

with respect to $\beta_{0,l}$ and β_l. The procedure can then be performed with different values of the shrinking factor λ to obtain the most beneficial value of λ.

For the logistic regression with elastic net penalty, the R-package `glmnet` [6] by Friedman et al. will be used.

Multinomial Diagonal Orthant Probit

In comparison to Bayesian multinomial logistic regression, Bayesian probit regression models offer more trivial distributions for the latent variables when performing Gibbs sampling. This approach can, however, be inefficient with the standard MCMC approach for multinomial probit due to highly dependent latent variables and parameters. To address the inefficiency, Johndrow et al. [12] propose the Multinomial Diagonal Orthant Probit model (DO-probit) which relies on conditional independence of the latent variables to enhance performance. Furthermore, the approach does not require a reference category for the parameter estimation.

Let $\gamma_{[1:K]}$ be independent binary variable representations of y, an unordered categorical response variable with K levels. For $y = l$, $\gamma_l = 1$ and $\gamma_k = 0$ for $k \neq l$. Location-scale distributed latent variables z_l can represent each binary variable, such that

$$z_l \sim f(\mu_l, \sigma) \tag{3.27}$$

and $z_l > 0 \leftrightarrow \gamma_l = 1$, where μ_l is the location parameter and σ is the common scale for all z. The z's are restricted to the set

$$\Sigma = \bigcup_{j=1}^{K} \{z \in \mathbb{R}^K : z_l > 0,\, z_k < 0,\, k \neq l\} \tag{3.28}$$

to ensure all $\gamma_k = 0$ for $k \neq l$. Specifically, when the location-scale function f is the univariate normal pdf, the result is a DO-probit model. The joint probability density of the latent variables is a multivariate normal distribution with K independent levels (unit covariance matrix) and location parameters restricted to one positive sign and the rest negative. Thus, the marginal distributions of any two latent variables are restricted to be diagonally apposed, yielding appropriateness to the name Diagonal Orthant. The probability of class l in a DO-probit model is

$$Pr(y_i = l|x_i, \boldsymbol{\beta}_{[1:K]}) = \frac{(1 - F(x_i\beta_l)) \cdot \prod_{k \neq l} F(-x_i\beta_k)}{\sum_{s=1}^{K} (1 - F(x_i\beta_s)) \cdot \prod_{k \neq l} F(-x_i\beta_k)} \tag{3.29}$$

where $F(\cdot)$ is the standard normal CDF [12].

The horseshoe estimator

As the feature vectors used in the thesis are likely to be sparse, it is suitable to use a regularisation method adapted for finding signals in noisy feature vectors. For the DO-probit model, the horseshoe prior, proposed by Carvalho et al. [5] will be used for shrinkage of the model coefficients. Given a sparse feature vector β and a p-dimensional vector $y|\beta \sim \mathcal{N}(\beta, 1)$, the horseshoe prior relies on the Gaussian and the positive reals half-Cauchy distribution according to the following:

$$\beta_i|\lambda_i \sim \mathcal{N}(0, \tau^2\lambda_i^2) \tag{3.30}$$

where λ_i is the global shrinkage term following

$$\lambda_i \sim C^+(0, 1) \tag{3.31}$$

and τ is the local shrinkage term distributed as

$$\tau \sim C^+(0, 1). \tag{3.32}$$

The posterior mean under the model can be shown to be

$$E(\beta_i|y) = \int_0^1 (1 - \kappa_i) y_i p(\kappa_i|y) d\kappa_i = \Big[1 - E(\kappa_i|y)\Big] y_i \tag{3.33}$$

where

$$\kappa_i = \frac{1}{1 + \tau^2\lambda_i^2}. \tag{3.34}$$

$E(\kappa_i|y)$ can be seen as the degree of shrinkage. The name "horseshoe" origins from the horseshoe-alike shape of the $Beta(1/2, 1/2)$ distributed prior for κ_i induced by the half-Cauchy prior on λ_i. If $\kappa_i \to 0$, the feature is an important signal and the shrinkage is small. If $\kappa_i \to 1$, the shrinkage is large as the feature is considered non-interesting noise [5].

The DO-probit with the horseshoe prior will be fitted using the Java implementation by Jonsson et al. [15].

3.3 Evaluation

The following section will describe the intended evaluation methods.

Multinomial model with Dirichlet prior

To evaluate the impact of the class distribution, a bayesian multinomial model will be used to predict test observations using relative frequencies.

Let y_k be counts of observations in class k, then

$$y = (y_1, \ldots, y_K) \sim Multinomial(n; \theta_1, \ldots, \theta_K) \tag{3.35}$$

where θ_k is a Dirichlet distributed prior on the probability of belonging to class k:

$$\theta = (\theta_1, \ldots, \theta_K) \sim Dirichlet(\alpha_1, \ldots, \alpha_K). \tag{3.36}$$

As mentioned previously, the Dirichlet distribution is conjugate to the Multinomial distribution, therefore, the posterior of θ is

$$\theta|y \sim Dirichlet(\alpha_1 + y_1, \ldots, \alpha_K + y_K). \tag{3.37}$$

The posterior mean

$$E(\theta_l|y) = \frac{\alpha_l + y_l}{\sum_{k=1}^{K} \alpha_k + y_k}, \tag{3.38}$$

which is also the predictive mean for a new test observation [7].

Cross validation

To maximise the information obtained from one dataset, K-fold cross validation can be used. The procedure divides the dataset into K roughly equally sized pieces, where $K-1$ of the data folds are used at a time to train the model and the last fold is used for validation of the obtained model [3]. The procedure is then repeated for all K folds and the K performance measures are averaged to achieve a cross validated model. For $\kappa \in \{1, \ldots, K\}$, the cross-validated prediction error is

$$CV(\hat{f}) = \frac{1}{N} \sum_{i=1}^{N} M(y_i, \hat{f}^{-\kappa(i)}(x_i)) \tag{3.39}$$

where $\hat{f}^{-k}(x)$ is the fitted model without the k:th fold and $M\big(y, \hat{f}(x)\big)$ is the misclassification rate. When there is some tuning parameter/parameters to be fitted with the model, as for instance one or more penalty factors, the prediction error becomes

$$CV(\hat{f}, \lambda) = \frac{1}{N} \sum_{i=1}^{N} M(y_i, \hat{f}^{-\kappa(i)}(x_i, \lambda)). \tag{3.40}$$

$CV(\hat{f}, \lambda)$ is then the estimate of the error curve which is minimised $\hat{\lambda}$, the optimal value of λ [10]. Cross validation will be used for obtaining the most beneficial values of the shrinkage parameter λ in the multinomial logistic regression models. As an alternative evaluation measure, the logit (logistic regression) models (with optimised α and λ), the *deviance* will be used, which is defined as

$$D\big(y, \hat{f}(x)\big) = -2\Big[\log\big(p(y|\hat{y}_0)\big) - \log\big(p(y|\hat{y}_s)\big)\Big] \tag{3.41}$$

where $\hat{y}_0$ are the fitted values of the model and $\hat{y}_s$ are the fitted values of the *saturated model*; a model in which one coefficient for every observation is fitted.

Accuracy, sensitivity and specificity

Some classic measures of performance for classification methods are accuracy, sensitivity and specificity. Given a confusion matrix for $K = 2$ (see Table 3.1),

	Prediction outcome	
Actual value	True positive	False negative
	False positive	True negative

Table 3.1: Confusion matrix example for the binary classification problem.

the *accuracy* is the percentage of correctly classified observations and can be obtained as

$$Accuracy = \frac{\text{True positive} + \text{True negative}}{\text{True positive} + \text{True negative} + \text{False positive} + \text{False negative}}. \tag{3.42}$$

For the multinomial case, the accuracy will be found as the sum of the diagonal elements divided by the sum of all elements. The *sensitivity* is the percentage of correctly classified positive cases and is therefore found as

$$Sensitivity = \frac{\text{True positive}}{\text{True positive} + \text{False negative}} \tag{3.43}$$

The multinomial case will yield one sensitivity estimate per class. For class l, the true positive rate is the number correctly classified as class l. The false negative is the number of cases where an object of class l is falsely classified as any other class. Finally, the *specificity* is the percentage of correctly classified negative cases:

$$Specificity = \frac{\text{True negative}}{\text{True negative} + \text{False positive}} \tag{3.44}$$

The specificity will also be found by class, where the true negative rate is the number of observations not belonging to class l not classified as l and the false positive rate is the number of non-l observations classified as l by the algorithm [10].

Validation comments

The rule mining will be performed on 70% of the data. To properly evaluate the results of the rule mining, the logistic regression and the DO-probit model should be applied to the same 70% raining set and cross-validated. The validated models should then be used to obtain out-of-sample predictions of the remaining 30%. However, the implementation of DO-probit is limited to evaluating the model with cross-validation. The evaluation of the two models will therefore be conducted in different manners; logistic regression will be evaluated using merely one training and test set (70/30) to minimise the bias of the rule mining results and the DO-probit model [12] will be evaluated by 10-fold cross-validation. The elastic net parameter, α, will not be cross-validated due to insufficient data. The parameter will be chosen by obtaining the optimal deviance from a grid of α. The parameters of the Beta distribution used when accounting for time elapse will be chosen by area experts rather than optimised by deviance due to computational limitations.

4 Results

The results will be presented step-wise. The first section will give the results of the rule mining algorithms, the second section will motivate the choice of α in the logistic regression models and give the optimal λ's, the third section will give the predictive performance of the rules obtained in the mining step and finally, the fourth section will examine how introducing the alarm data as covariates in the models impacts the topic coefficients.

4.1 Feature construction

As mentioned in the method section, the Apriori algorithm and the SPADE algorithm were run on 70% of the data, leaving 30% for pure evaluation. As the managed object (MO) as well as the specific alarm was believed to be of interest, the Apriori algorithm was primarily run with the MOs and the alarms as separate items. However, since the SPADE algorithm does not allow multiple events at one time point, the MOs and alarms for a specific time point needed to be combined into one item (recall the structure of data in Table 2.1). To properly evaluate the need for separating MOs from alarms, both item options were tested with the Apriori algorithm and evaluated with logistic regression and DO-probit. As the accuracy and deviance do not differ substantially between the two approaches, the combined MO and alarm option was used for better comparability between Apriori and SPADE. The results of the comparison will be presented in the following section.

Apriori rules

In the first subsection, the choice of itemset structure for further analysis will be motivated and in the second subsection the results of the chosen approach will be presented.

Choice of itemset structure

The choice of itemset construction was evaluated with the results of a logistic regression model and a DO-probit model using the resulting rules as covariates. The result of the logistic regression predictions are compared in terms of misclassification rate in Figure 4.1. The version 1 items are items of the form $\{MO, Alarm\}$ (MO and Alarm compressed into one

item) and version 2 items are items of the form $\{MO\}, \{Alarm\}$ (MO and Alarm considered as two independent items).

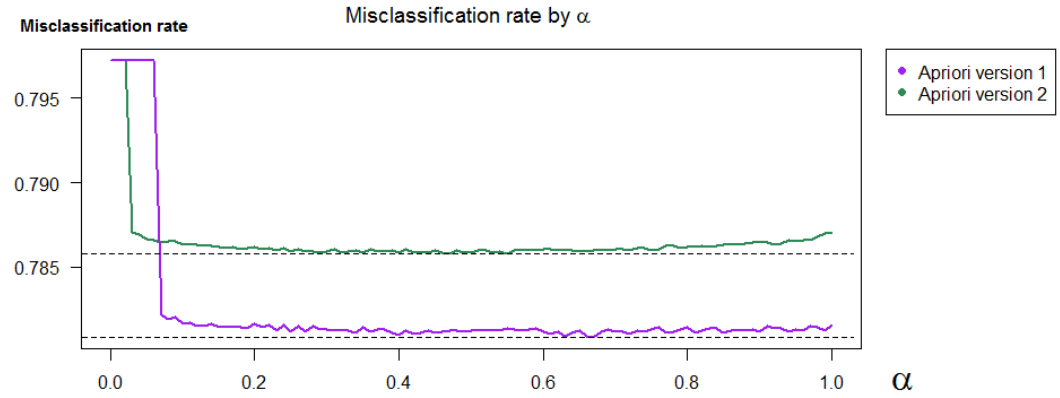


Figure 4.1: Comparison of accuracy for the Apriori item construction options.

The first option appears to be beneficial since the misclassification rate is lower for most values of α. Note, however, the scale of the y-axis; the difference is not extensive. The misclassification error for the optimal α's are approximately 0.781 and 0.786, thus, the difference is merely about 0.005. The DO-probit evaluation yields 21.2% and 21.4% accuracy respectively, a difference of 0.2 percentage points. The stability of the log-posterior for the first four folds in the 10-fold cross-validated DO-probit evaluation using the first itemset composition is presented in Figure 4.2 and Figure 4.3 shows the results for the second composition.

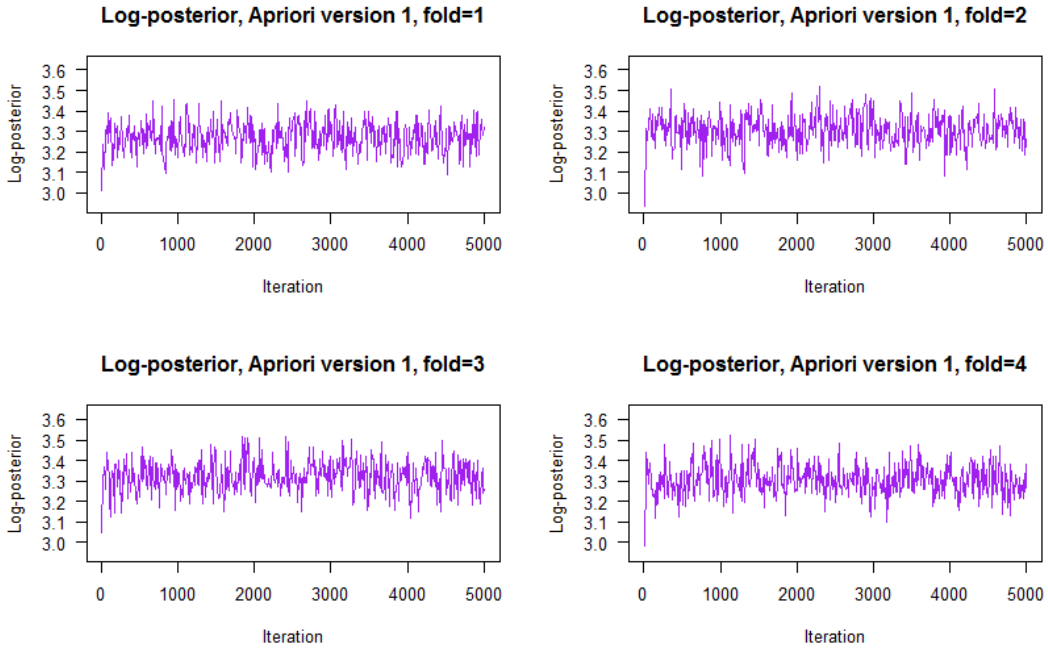


Figure 4.2: The DO-probit log-posterior for the first four folds using Apriori covariates, version 1.

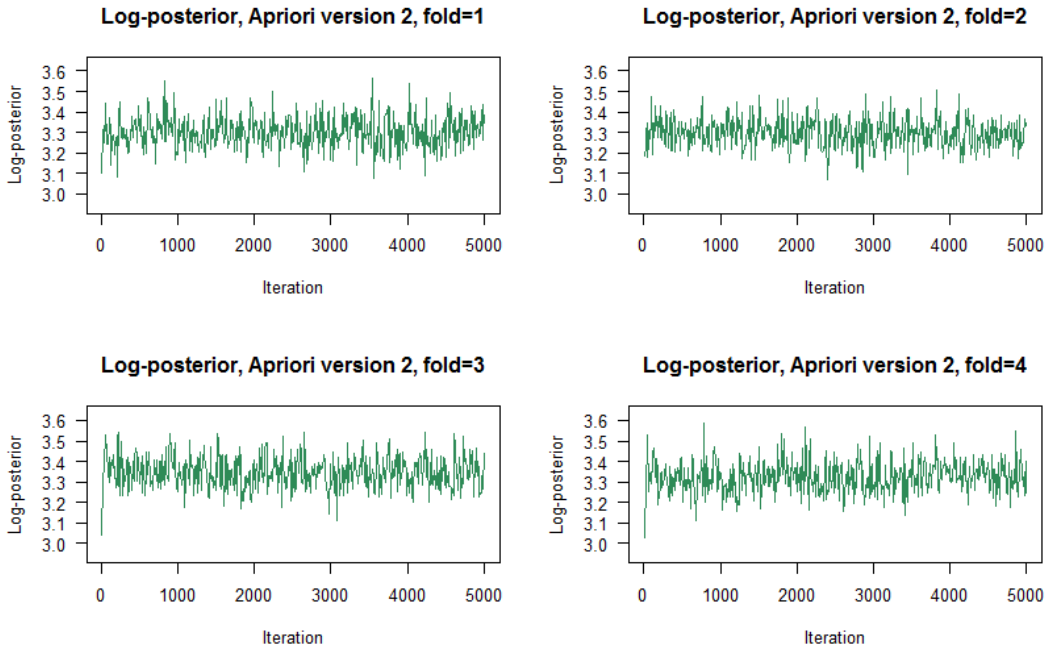


Figure 4.3: The DO-probit log-posterior for the first four folds using Apriori covariates, version 2.

The remaining folds show similar patterns, therefore, they are not displayed. The log-posteriors appear similar for the two approaches; both models stabilise quickly and appear to have obtained the posterior distribution. The first alternative is concluded to be the best option since this is the composition used for the SPADE sequences and therefore further analysis is performed for this set.

Resulting rules

The threshold for confidence was set high since a lower confidence level yielded an extensive number of rules and the maximum length of a pattern was set to four items for the same reason. The support level was, to obtain the most interesting rules, set as high as possible without restricting the search space too much. A higher minimum support resulted in a lower confidence level.

Setting the minimum confidence to be 100% and the minimum support count to 6 yielded 169 rules implying various MHOs. Restricting the time elapse with the customised confidence measure (Equation 3.14) with a minimum of 0.8 limited the result to 115 rules. The right hand side distribution of the rules can be found in Table 4.1.

Nr.	Class	Rule count
7	PD BB	71
8	PD CAT	37
12	RCS-DEV	7
	Total	**115**

Table 4.1: Class distribution of confident rules, Apriori

As can be seen, most rules describe one developer group in particular (PD BB), and only 3 out of 14 groups are represented. Therefore, the number of rules aiming to describe a specific class was limited to #rules/#unique classes in the rule set. The rules with the highest lift values by class were chosen. The right hand side distribution is given in Table 4.2.

Nr.	Class	Rule count
7	PD BB	38
8	PD CAT	37
12	RCS-DEV	7
	Total	**82**

Table 4.2: Class distribution of final rules, Apriori

The final set of rules used as covariates in the logistic and DO-probit models therefore consisted of 82 rules.

SPADE sequences

As SPADE does not rely on a customised threshold for time elapse, the minimum confidence was set to 0.8 directly, and the minimum support to 6 occurrences with a maximum sequence length of four events. Running the algorithm on these terms generated 235 sequences suggesting specific classes. The represented classes are presented in Table 4.3.

Nr.	Class	Rule count
8	PD CAT	189
4	GWMR	27
7	PD BB	12
11	PD TC - SAM	4
12	RCS-DEV	2
9	PD Radio	1
	Total	**235**

Table 4.3: Class distribution of confident sequences, SPADE

The number of sequences by represented class was limited in the same manner as the Apriori rules, yielding the final represented class distribution according to Table 4.4.

Nr.	Class	Rule count
8	PD CAT	39
4	GWMR	27
7	PD BB	12
11	PD TC - SAM	4
12	RCS-DEV	2
9	PD Radio	1
	Total	**85**

Table 4.4: Class distribution of final sequences, SPADE

Thus, the number of sequences passed as covariates to the predictive modeling was 85.

The Apriori rules and the SPADE sequences were transformed into binary variable representations and fitted in logistic regression models and DO-probit models with and without *topics* as covariates to obtain the impact on the model predictions and coefficients when adding the alarm data. The topic covariates is the result of pre-processing the free text information with LDA performed by Jonsson et al. [13]. The authors tested 40 and 100 topics, however, as 100 topics did not perform substantially better than 40 and it is desired to compress the models, 40 topics have been used in this thesis.

4.2 Choice of α and λ

This section will motivate the choice of α in the logistic regression models and give the optimal regularisation parameter λ for each model. The misclassification rate on a grid of α can be found in Figure 4.4.

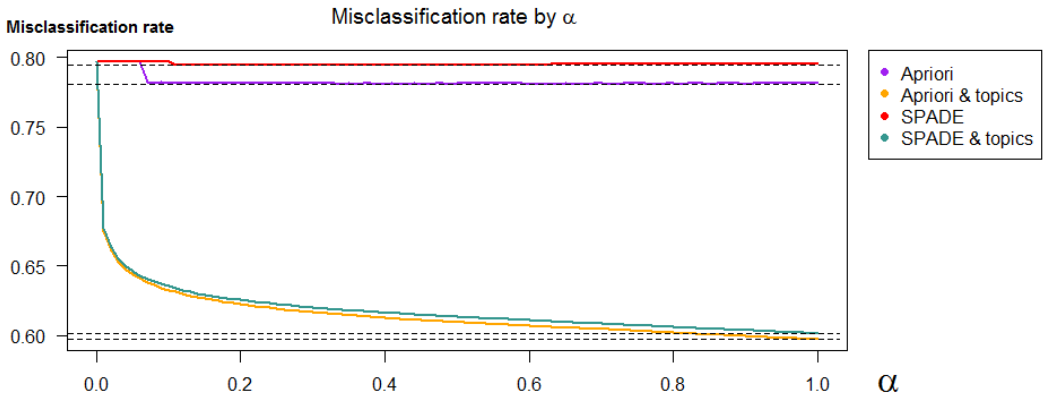


Figure 4.4: Misclassification by α for all models.

It can be seen that the logistic regression models with only Apriori rules or only SPADE sequences yield higher misclassification rate than the models using both rules or sequences and topics. Furthermore, for the models using both rules or sequences and topics, the misclassification rate is high for $\alpha \to 0$ and decreases when $\alpha \to 1$. The optimal α's were therefore concluded to be $\alpha = 1$ (a LASSO model) for the covariate sets containing both Apriori rules and topics and SPADE sequences and topics. For the Apriori and SPADE sets, the misclassification is also higher for $\alpha \to 0$, however, appears to stabilise rather than continuously decrease when $\alpha \to 1$. The optimal α for the Apriori set was found as $\alpha = 0.67$ and for the SPADE set the optimal α was found as $\alpha = 0.41$.

For the stated α's, the optimal penalty parameters λ were, using cross-validation, found as $\lambda = 2.50e^{-5}$ for the Apriori set, $\lambda = 1.87e^{-5}$ for the Apriori rules combined with the topics, $\lambda = 2.63e^{-4}$ for the SPADE sequences and, finally, $\lambda = 2.71e^{-5}$ for the SPADE sequences and topics combined.

4.3 Predictive performance

In this section, the specificity and sensitivity for each model will be presented. Note the difference in evaluation sets; the logistic regression models were tested on the 30% hold-out data not mined by Apriori nor SPADE, while the DO-probit models predicted the entire data set as all observations are predicted in the cross-validation. The test sets for the two models are, therefore, not directly comparable as the bias may differ. However, both measures are estimating the generalisation (out-of-sample predictive) performance of the two models.

Sensitivity

The sensitivity by class for all models can be found in Table 4.5.

Nr.	Class	Model	Apriori	Apriori & topics	SPADE	SPADE & topics
1	BBI	Logit	0.00	0.0566	0.00	0.0566
		DO-probit	0.00	0.0160	0.00	0.0160
2	CPI	Logit	0.00	0.00	0.00	0.00
		DO-probit	0.00	0.00	0.00	0.00
3	CPP	Logit	0.00	0.291	0.00	0.266
		DO-probit	0.00	0.259	0.00	0.239
4	GWMR	Logit	0.00	0.410	0.00	0.361
		DO-probit	0.00	0.378	0.00	0.359
5	HW	Logit	0.00	0.00	0.00	0.00
		DO-probit	0.00	0.00	0.00	0.00
6	OSS	Logit	0.00	0.00	0.00	0.00
		DO-probit	0.00	0.00	0.00	0.00
7	PD BB	Logit	0.0357	0.620	0.00974	0.643
		DO-probit	0.0555	0.622	0.00	0.631
8	PD CAT	Logit	0.0856	0.732	0.981	0.740
		DO-probit	0.181	0.776	0.987	0.784
9	PD Radio	Logit	0.00	0.294	0.00	0.279
		DO-probit	0.00	0.271	0.00	0.266
10	PD TC - COE	Logit	0.00	0.0552	0.00	0.0621
		DO-probit	0.00	0.0373	0.00	0.0393
11	PD TC - SAM	Logit	0.988	0.413	0.0274	0.413
		DO-probit	0.887	0.444	0.0171	0.418
12	RCS-DEV	Logit	0.0556	0.389	0.0111	0.378
		DO-probit	0.0495	0.360	0.0353	0.385
13	System	Logit	0.00	0.00	0.00	0.00
		DO-probit	0.00	0.00	0.00	0.00
14	Tools	Logit	0.00	0.00	0.00	0.00
		DO-probit	0.00	0.00	0.00	0.00

Table 4.5: Sensitivity by class and covariate set.

It can be seen that the sensitivity is typically low for many classes and covariate sets. BBI has a sensitivity below 10% for all models. Examining the confusion matrices (see Appendix A), the models with only Apriori rules and only SPADE sequences fail to correctly classify any BBI observations. It can further be found in the confusion matrices that the classes CPI, OSS, System and Tools are not represented in the fitted values for any covariate set or model. There are a few predictions to HW, however, they are never correctly classified. The highest sensitivities are obtained for PD TC - SAM (0.988 for logistic regression and 0.887 for DO-probit) with the Apriori rules. Again examining the confusion matrices, this is due to the fact that both models predict almost all observations to class PD TC - SAM (some observations are predicted to PD BB, PD CAT and RCS-DEV), therefore, few observations of this group are misclassified. In distinction to the models using only Apriori rules as covariates, the models using only SPADE sequences obtain high sensitivity for PD CAT. This as most observations are classified as PD CAT. The sensitivities for the combined sets, Apriori & topics and SPADE & topics, are similar. They are non-zero for the same classes and are in general about equally high. The combined sets yield higher sensitivity for most classes represented in the fitted values.

Specificity

The specificity by class for all models is presented in Table 4.6.

Nr.	Class	Model	Apriori	Apriori & topics	SPADE	SPADE & topics
1	BBI	Logit	1.00	0.998	1.00	0.998
		DO-probit	1.00	0.999	1.00	1.00
2	CPI	Logit	1.00	1.00	1.00	1.00
		DO-probit	1.00	1.00	1.00	1.00
3	CPP	Logit	1.00	0.958	0.999	0.953
		DO-probit	1.00	0.959	0.999	0.960
4	GWMR	Logit	1.00	0.972	1.00	0.975
		DO-probit	1.00	0.976	1.00	0.976
5	HW	Logit	1.00	0.999	1.00	0.999
		DO-probit	1.00	1.00	1.00	1.00
6	OSS	Logit	1.00	1.00	1.00	1.00
		DO-probit	1.00	1.00	1.00	1.00
7	PD BB	Logit	0.997	0.851	0.992	0.838
		DO-probit	0.998	0.861	0.999	0.852
8	PD CAT	Logit	0.989	0.756	0.0263	0.744
		DO-probit	0.905	0.793	0.504	0.784
9	PD Radio	Logit	1.00	0.970	1.00	0.972
		DO-probit	1.00	0.973	1.00	0.977
10	PD TC - COE	Logit	1.00	0.988	1.00	0.985
		DO-probit	1.00	0.995	1.00	0.995
11	PD TC - SAM	Logit	0.0466	0.859	0.992	0.876
		DO-probit	0.538	0.875	0.995	0.888
12	RCS-DEV	Logit	0.998	0.968	0.996	0.971
		DO-probit	0.999	0.980	0.996	0.981
13	System	Logit	1.00	1.00	1.00	1.00
		DO-probit	1.00	1.00	1.00	1.00
14	Tools	Logit	1.00	1.00	1.00	1.00
		DO-probit	1.00	1.00	1.00	1.00

Table 4.6: Specificity by class and covariate set.

The specificity is close to 1 for most classes. CPI, OSS, System and Tools have a specificity of 1.00 for all models and all covariate sets. This since they are not represented in the fitted values and therefore they are never classified as false positives. Logistic regression with SPADE covariates yields the lowest specificity, 0.0263, for PD CAT. DO-probit yields a substantially higher specificity for SPADE, yet rather low in comparison to most other models and covariate sets. The Apriori set shows similar behavior but for class PD TC - SAM, where logistic regression obtains a specificity of 0.0466 while the DO-probit obtains a value of 0.538 for the class. For the Apriori rules, the specificity is high for all classes except for PD TC - SAM. For Apriori & topics, the numbers are slightly lower and the lowest sensitivity is obtained for PD BB. The SPADE sequences give a high specificity for all classes except PD CAT, and the SPADE & topics set also yield the (marginally) lowest specificity for PD CAT.

Summary

The probabilities of each class were obtained in each fold of the DO-probit models to examine how the class distribution may impact the predictions. Since no prior information was at hand, an uninformative prior was used: $\alpha_1 = \cdots = \alpha_K = 1$, and the training data for each fold was used as y. The test sets were predicted to be one of three classes; PD BB (7), PD CAT

(8) or PD TC - SAM (11). Since the observations contained in each fold of the DO-probit varied for each covariate set, the class distributions for each training set were not equal. Therefore, the accuracy of the prediction varies for the different covariate sets. The accuracy is provided in Table 4.7 with an overview of all models. The summarised results when using only the topics as covariates are also presented to show difference in performance when adding any rule set. The misclassification rate on a grid of α, the confusion matrices, the log-posterior and the coefficients for the models using only the topics can be found in Appendix C. The deviance refers to the deviance of the logistic model for each feature set. Note that the multinomial Dirichlet model predicts all observations in the test set as the most frequent class in the training set (combined with the prior).

Covariates	α	**Dev.**	**Acc. logistic**	**Acc. DO-probit**	**Acc. multinom. Dir**
Topics	1.0	12784	42.3%	42.9%	15.6%
Apriori	0.67	17602	21.0%	21.6%	15.2%
SPADE	0.41	18006	20.8%	20.6%	17.8%
Apriori & topics	1.0	12491	42.4%	43.2%	13.9%
SPADE & topics	1.0	12627	42.4%	42.9%	15.3%

Table 4.7: Overview of the models.

The accuracy of the multinomial Dirichlet for the 70/30 training and test data used in the logistic regression models is 20.5%. All covariate sets achieve a higher accuracy than predicting using merely the distribution of the classes (the multinomial Dirichlet).

In regards to the models using any covariates, it is evident that the logistic regression models are optimised for $\alpha = 1$ (LASSO penalty) when the topics are included. The deviance is minimised for the covariate set with topics and Apriori rules, which also gives the highest accuracy. Note, however, that the deviance is merely comparable for models where one model contains a subset of the covariates in the second model. Therefore, the most interesting relations are the deviances for topics, Apriori & topics and SPADE & topics. The deviance is lower for the Apriori & topics set than for the SPADE & topics set, even though more SPADE sequences were introduced. In general, the DO-probit achieves a higher accuracy than the logistic regression models. Recall, however, that the approaches are not directly comparable since the validation has been conducted in different manners. Furthermore, the differences are not extensive neither between models nor feature sets.

The MCMC chains for the log-posteriors can be found in Appendix B. For the Apriori rules and the SPADE sequences, the chains appear rather stable, however, are more varying in magnitude for the Apriori & topics and SPADE & topics sets.

4.4 Coefficient evaluation

Since the aim of this thesis is to determine whether the alarm log data can contribute to the predictions, it is interesting to examine the coefficients of each model to find whether the introduced rules or sequences were included in the models as expected (i.e. used in the equations of the expected classes). This section will therefore present the coefficients for each set of covariates and each model as well as a closer look at the model coefficients of the most promising pre-processing method. Both the logistic regression models and the DO-probit models were fitted with an intercept, which is included in the plots as the first coefficient.

Apriori rules

The class specific coefficients of the logistic regression model using Apriori rules as covariates is found in Figure 4.5.

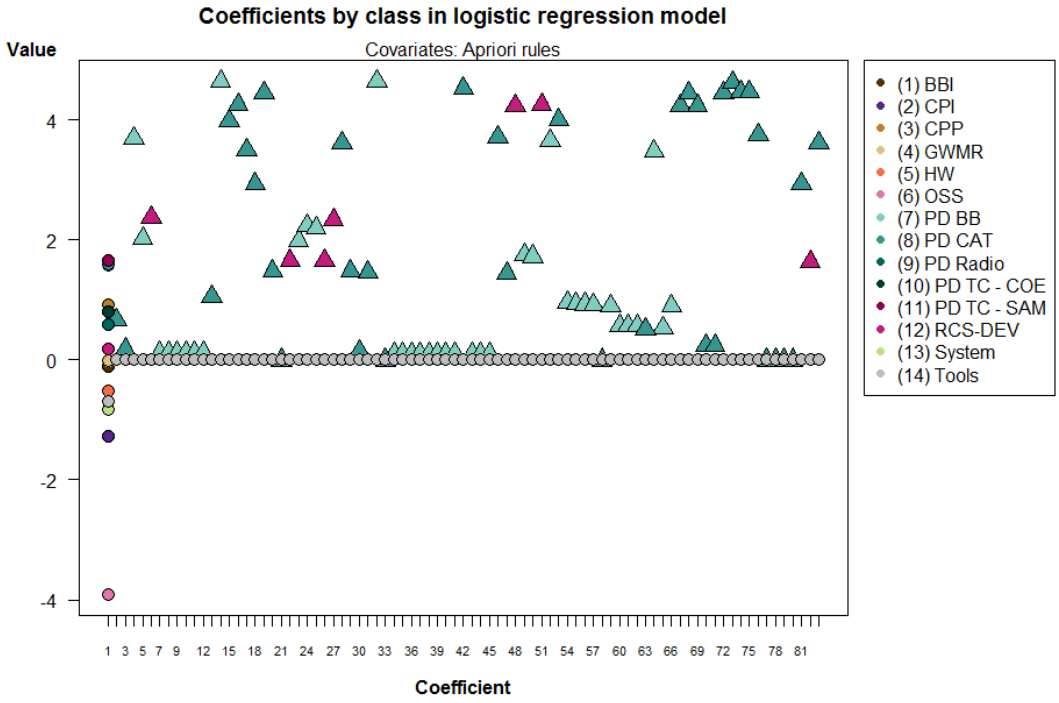


Figure 4.5: Beta coefficients of logistic regression model with Apriori rules as covariates.

The triangles are the coefficients which are expected to be positive and the circles are the coefficients expected to be small or negative (the intercepts are all represented by circles). A (class specific) coefficient is expected to be positive if the corresponding variable was introduced to predict to the class the coefficient is connected to. The largest coefficients are indeed in the shape of triangles. None of the non-zero coefficients are positive if not expected (no triangles fall below zero). It is evident, furthermore, that all coefficients expected to be zero are in fact set to zero or close to zero. PD BB and PD CAT along with RCS-DEV have the largest absolute values. OSS has an unusually low intercept, however, all other coefficients are zero or close to zero.

The coefficients plot for the Apriori rules and the DO-probit is found in Figure 4.6.

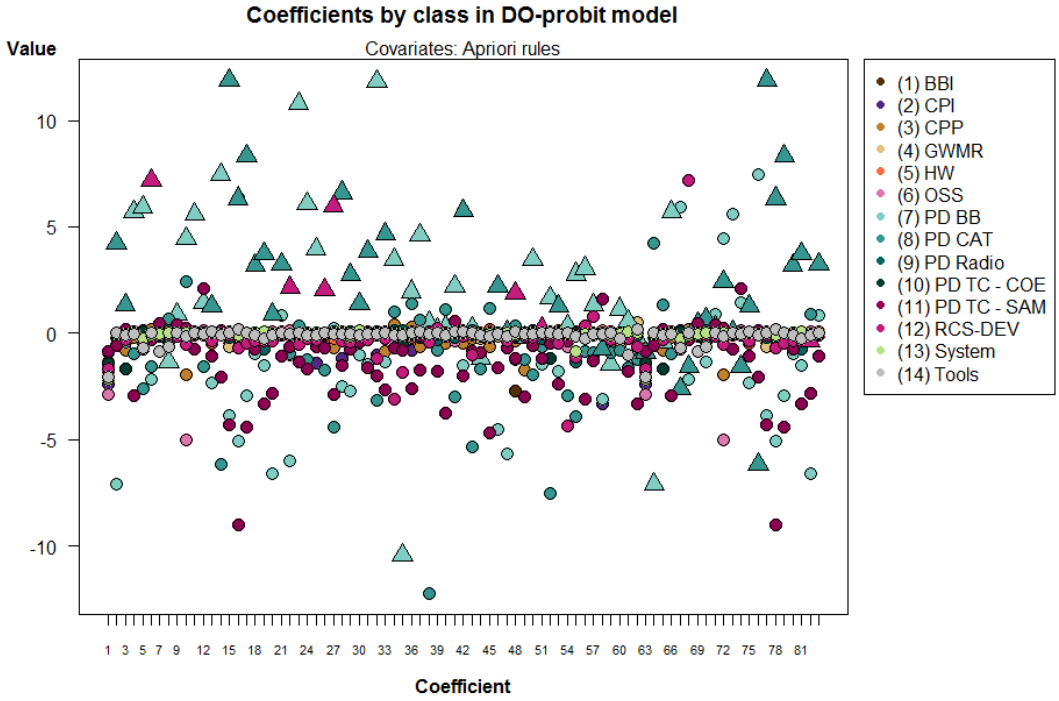


Figure 4.6: Beta coefficients of DO-probit model with Apriori rules as covariates.

The absolute values of the coefficients are greater for the DO-probit model and no coefficients are equal to zero since the horseshoe shrinkage does not exclude variables completely. Thus, elastic net with the set α appears to shrink the coefficients in a greater extent. As well as with the logistic regression, most strongly positive coefficients are triangle shaped. There are, however, some unexpected coefficients greater than zero. It appears PD BB, PD CAT, RCS-DEV and PD TC - SAM are the classes for which the coefficients are the most deviating from zero. The 38:th coefficient has a large negative impact on PD CAT.

Figure 4.7 displays the coefficients of the logistic regression model when combining the Apriori rules and the topics; the latter 40 coefficients regard the topics.

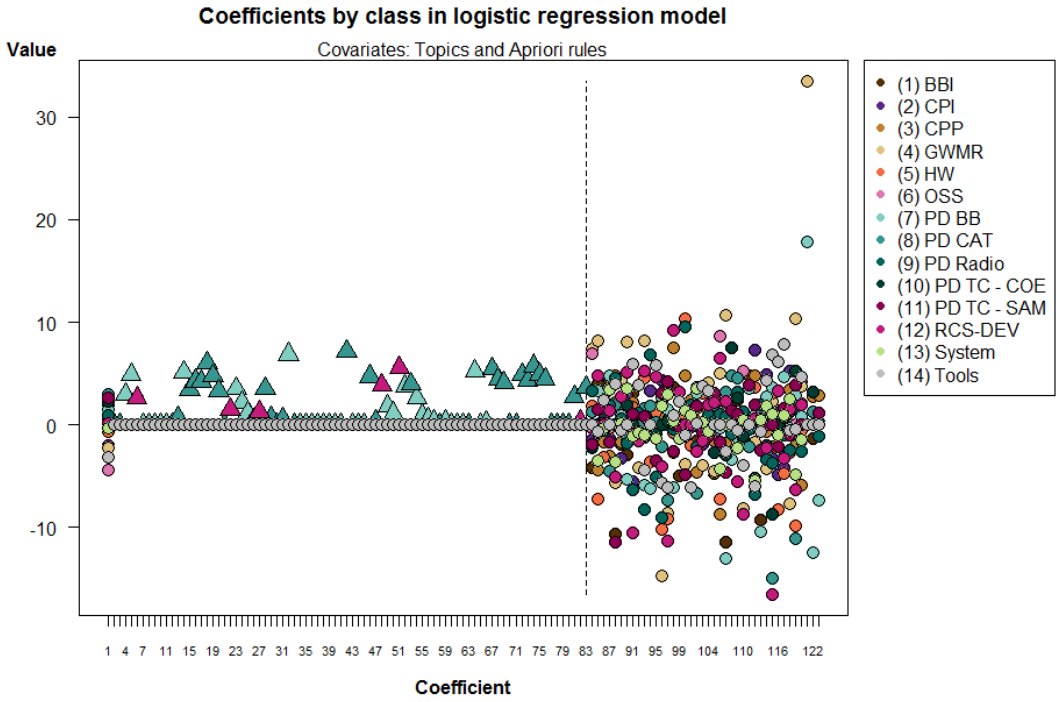


Figure 4.7: Beta coefficients of logistic regression model with Apriori rules and topics as covariates.

The dashed line represents the breaking point between rules and topics. The plot resembles the coefficient plot for the logistic model with only Apriori rules as covariates, as most rule coefficients are set to be zero or close to zero, except for rule coefficients for PD BB, PD CAT and RCS-DEV. These classes are expected as all or most points are triangles. There appear to be no rule coefficients below 0. Most classes seem to have larger absolute values of the topic coefficients than the Apriori rule representations.

Figure 4.8 gives the corresponding coefficients for the DO-probit.

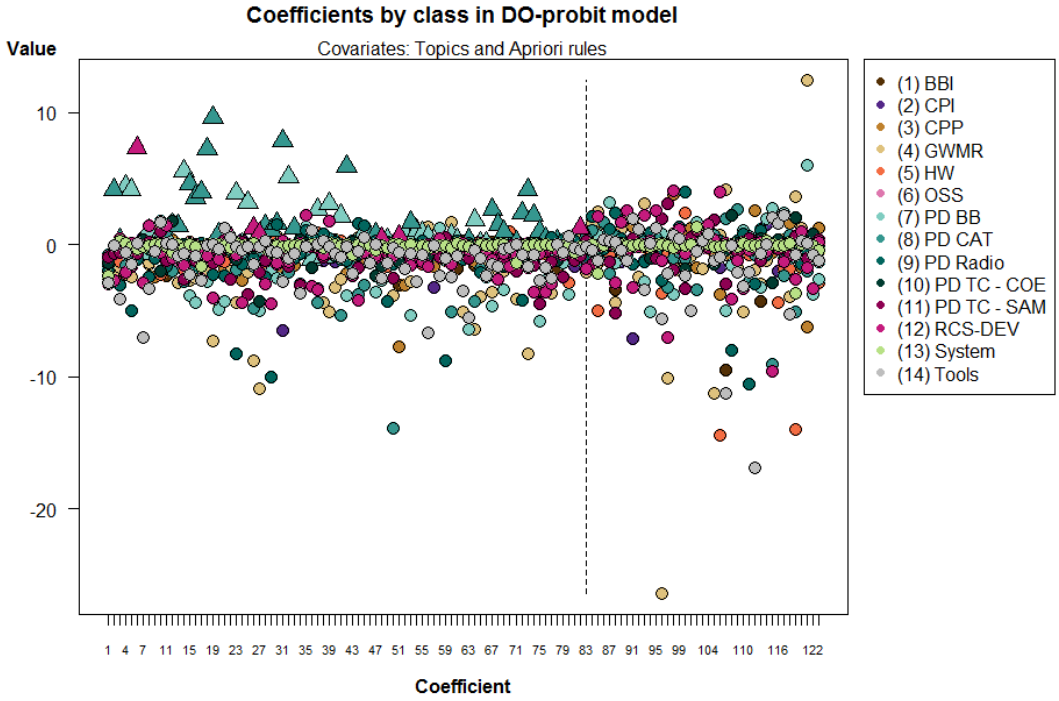


Figure 4.8: Beta coefficients of DO-probit model with Apriori rules and topics as covariates.

The DO-probit absolute values of the Apriori rule coefficients are again larger than for the logistic regression. DO-probit appears to have better grasped information from the rule set, as more coefficients are further from zero than for the logistic regression. Many strongly positive rule coefficients are expected, as they are represented by triangles. PD BB, PD CAT and in some sense RCS-DEV are still well represented among the larger rule coefficients. GWMR, however, has more distinctively negative coefficients than positive and PD TC - SAM has rather small absolute values of the coefficients. Besides for the previously seen distinct coefficient values for GWMR, the 113:th coefficient (a topic coefficient) for Tools is strongly negative.

SPADE sequences

The coefficients of the logistic regression model using only SPADE sequences as covariates are presented in Figure 4.9.

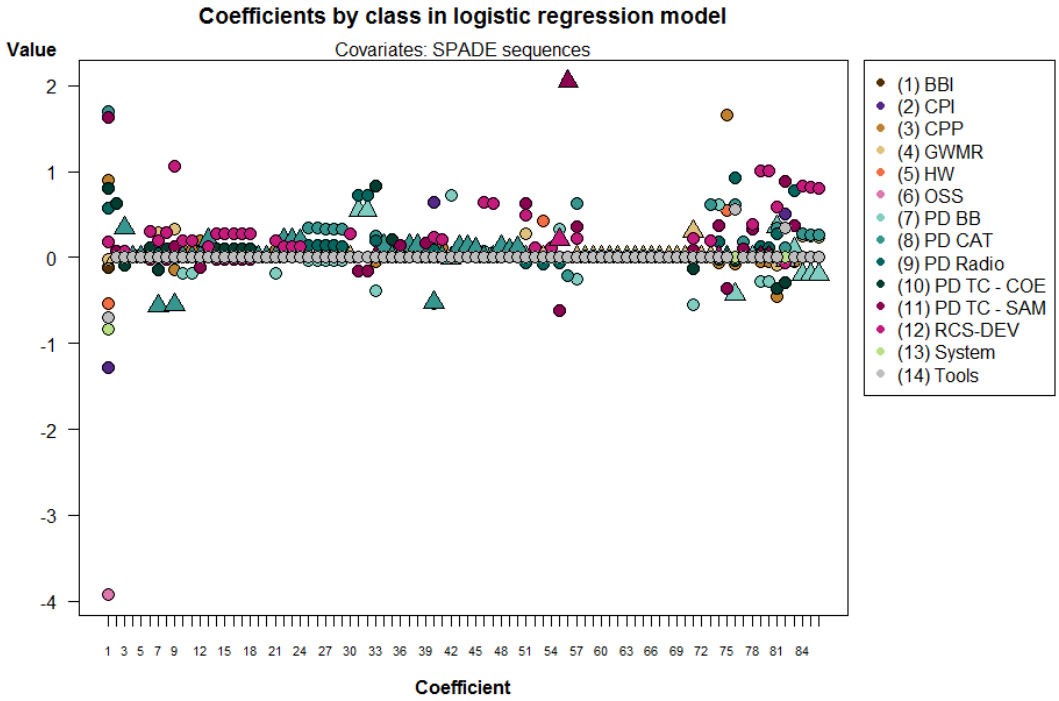


Figure 4.9: Beta coefficients of logistic regression model with SPADE sequences as covariates.

The SPADE sequences do not in general appear to have impact on the expected classes, as most triangles are centered around 0. As for the Apriori rules, OSS has somewhat an extreme intercept in the logistic model, however, the other coefficients are close to or equal to zero. Tools does not seem to be easily predicted with the SPADE sequences since all Tools coefficients are lined up around zero. The magnitude of the coefficients are overall smaller than for the Apriori rules, which in general spanned between -4 and 4.

The specific coefficients for the DO-probit are found in Figure 4.10.

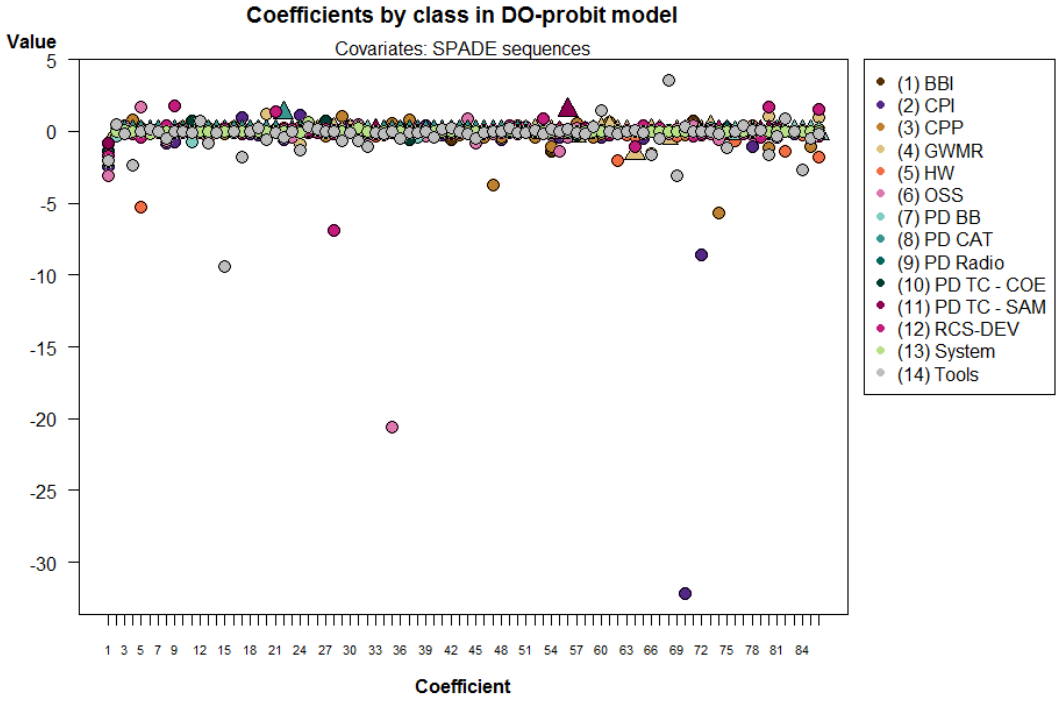


Figure 4.10: Beta coefficients of DO-probit model with SPADE sequences as covariates.

DO-probit shows triangles slightly further from zero than the logistic regression for the same covariate set. The absolute values of the coefficients are mostly within a five length distance from zero, however, some coefficients are substantially lower than most; CPI is provided strong negative coefficients for coefficients 70 and 72 which were introduced to predict GWMR. OSS again has one low coefficient, however, for this model it regards the 35:th rule, originally introduced for predicting to PD CAT (not displayed here).

The coefficients of the logistic regression for the covariate set constituting of both SPADE sequences and topics are given in Figure 4.11.

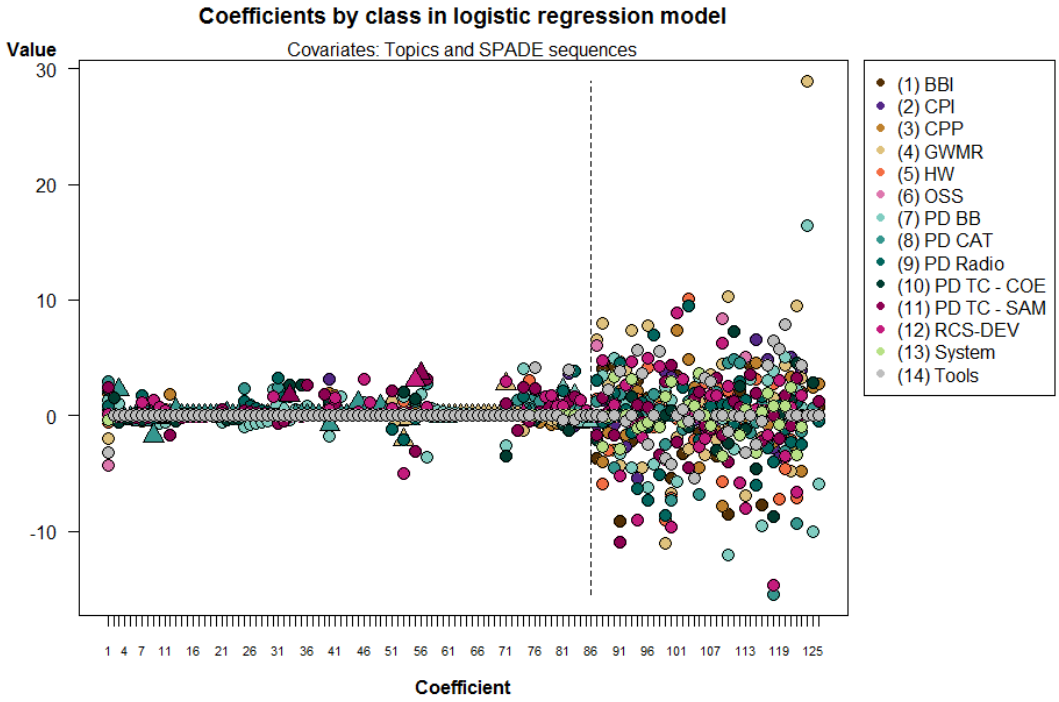


Figure 4.11: Beta coefficients of logistic regression model with SPADE sequences and topics as covariates.

The SPADE sequences do not seem to be complementing the topics as expected since most triangles are close or equal to zero. PD BB, PD CAT, PD TC - SAM and RCS-DEV appear to be the classes most using the SPADE sequences when predicting, and one of the RCS-DEV has the largest positive coefficient of the expected rules. In difference to the Apriori rules, some rule coefficients are negative. Some triangles also fall below 0. The topic coefficients have higher absolute values than the sequences, and GWMR and PD BB have rather large coefficient values for the same topic covariate.

Finally, the coefficients of the DO-probit for the same covariate set are given in Figure 4.12.

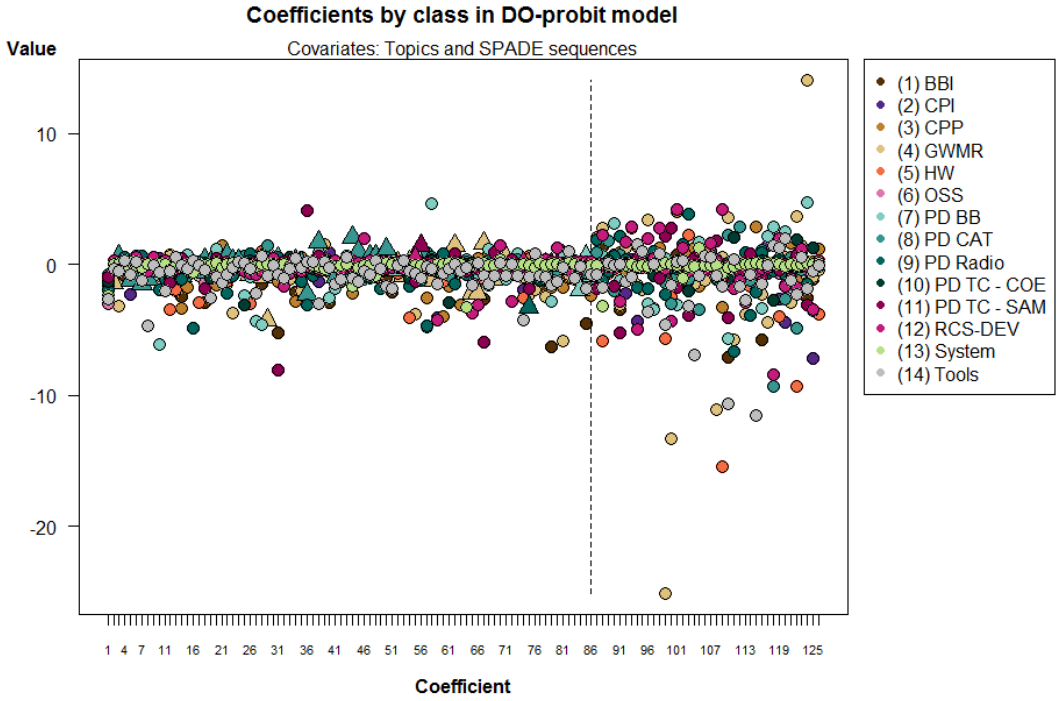


Figure 4.12: Beta coefficients of DO-probit model with SPADE sequences and topics as covariates.

The sequence variables have a smaller magnitude than the topics, however, not in the same extent as the coefficients of the logistic regression. Some triangles are slightly positive, however, most are equal to or close to zero. The absolute values of the coefficients for topics and sequences are not as different for DO-probit using this covariate set as for the topics and Apriori rule set. Tools appears to make more use of the sequences with this prediction model than with logistic regression since the coefficients are further from zero. The coefficients for the sequences are in general more evenly distributed among classes for DO-probit and this feature set. GWMR makes use of the same topics as in previous models.

Topic coefficient impact when introducing Apriori rules

As several Apriori rule coefficients proved to be of large magnitude despite adding the topic variables, it is interesting to investigate how the topic coefficients are affected when adding the rule set as predictors. In difference to the logistic model, the DO-probit model showed interesting negative dependencies as well as positive for the covariate set with Apriori rules and topics and, furthermore, when covariates are correlated in logistic regression, LASSO arbitrary chooses what variable to include and what to exclude. Therefore, the topic coefficients of the DO-probit models with and without the Apriori rules were investigated. In Figure 4.13, Figure 4.14 and Figure 4.15 the absolute values of the topic coefficients when the rule set is included in the model is plotted against the absolute values of the topic coefficients when no rules are added to the set. Note that the scales of the axes are not equal for all plots.

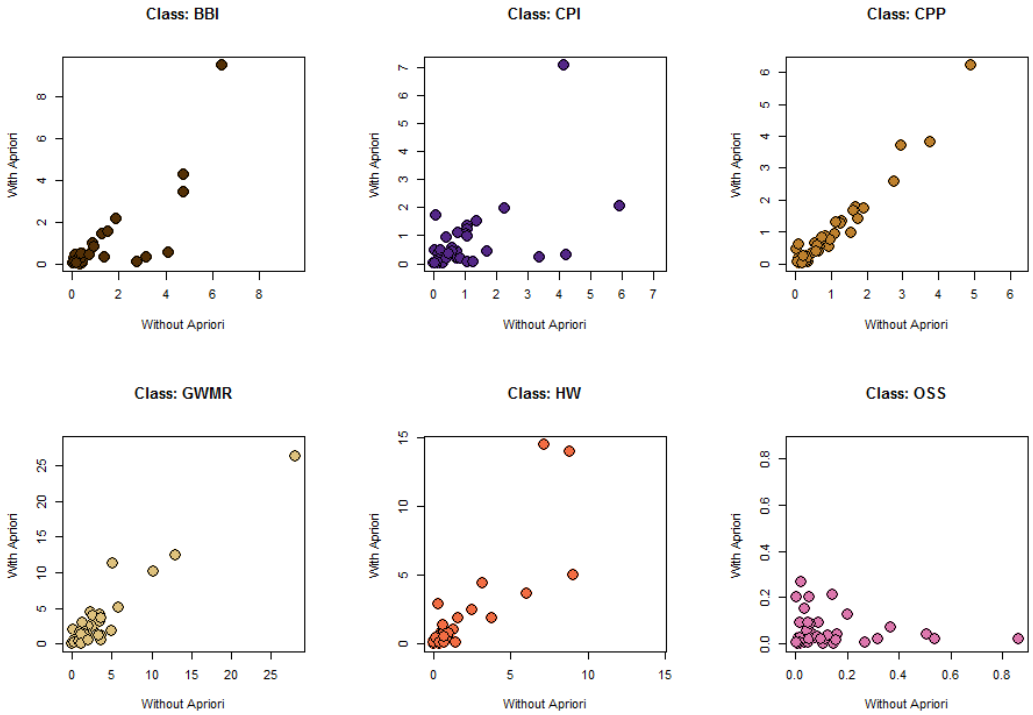


Figure 4.13: Difference in topics coefficients in DO-probit model, classes 1-6.

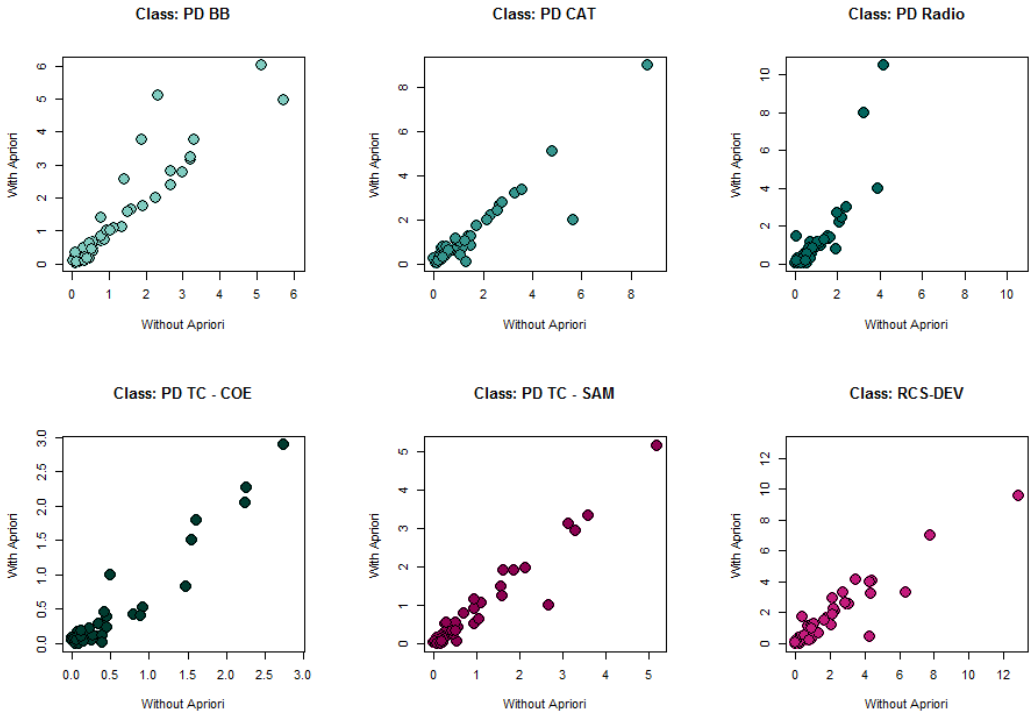


Figure 4.14: Difference in topics coefficients in DO-probit model, classes 7-12.

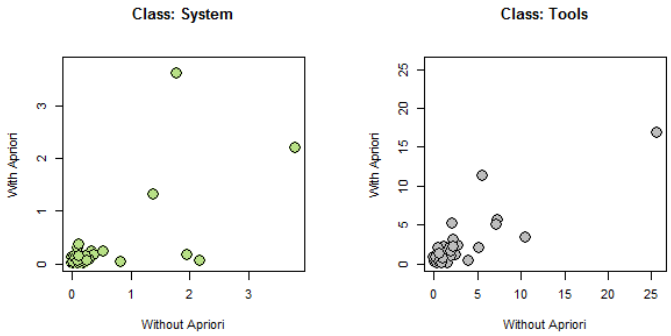


Figure 4.15: Difference in topics coefficients in DO-probit model, classes 13-14.

The horizontal axes are the topic coefficients when no rules are included in the model and the vertical axes are the topic coefficients when Apriori rules are included in addition to the topics. If adding the Apriori rules influences the coefficients of the topics, one would expect the absolute values of the topic coefficients to decrease when the rules are introduced. BBI, CPI, HW, PD TC - COE, System and Tools show some coefficients which decrease and some increase while PD CAT, PD TC - SAM and RCS-DEV show evidence of this type of behavior. For PD - CAT, PD TC - SAM and RCS-DEV, some topic coefficients decreased.

5 Discussion

This chapter discusses the results and the chosen methods.

5.1 Results

Choice of itemset structure for Apriori

The two options of item construction ($\{Alarm, MO\}$ and $\{Alarm\}$, $\{MO\}$) were evaluated using the accuracy (over a grid of α for the logistic model). The differences in accuracy proved to be small; merely a difference of 0.4 percentage points was found for the optimal α's. The corresponding numbers for the DO-probit was 0.2 percentage points. To evaluate the posterior obtained by DO-probit for the two versions, the MCMC mixing was evaluated. The posterior chains appeared stable for both versions. Since the two different item set versions give similar performance, it seems the better approach is to use the first itemset in order to improve comparability of the rule mining algorithms.

Apriori

Many rules were found when mining the alarm logs. Therefore, the confidence threshold could be set as high as 100% for the initial run. As many as 169 100% confident rules were found by the traditional Apriori approach (without correcting for time elapse). 54 rules were discarded by the customised confidence of 80% due to too wide a time window within the patterns. The class distribution of the final rules were displeasing, as only three out of the 14 classes were represented by some rule in the set. This indeed caused problems when predicting using only the Apriori rules as no variables for 11 of the target classes were introduced. However, when investigating the coefficients of the logistic and the DO-probit model using the rules only, the coefficients were typically strongly positive for the expected classes and smaller or negative for other classes. This implies that many of the obtained rules can indeed be interesting when predicting the developer groups. However, the logistic regression models typically used the introduced variables for more classes than expected (the coefficient absolute values were generally higher). This may be explained by the parameter settings. α was chosen as 0.67, which is not especially close to LASSO, which would discard more non-interesting variables rather than just shrink them. As there is no local shrinkage parameter

available for elastic net, this may however have resulted in discarding all variables not interesting to every class.

Some coefficients were large in magnitude for more than one class, which may also be of interest as a set of alarms found in the alarm log may be of significant value when determining what developer groups are *unlikely* to solve the bug. This would yield negative coefficients, which could be seen for the DO-probit.

When using both topics and rules as predictors in the models, the results were still thrilling. The logistic regression model shrunk most rule coefficients to be close or equal to zero for the previously believed classes yet kept the coefficients large for the classes expected to be large. For the logistic regression, no rule coefficients were found as negative which may indicate that the method missed some negative correlation. The topic coefficients were, however, spread around zero. The DO-probit model gave more evenly distributed coefficients for the rules and the topics set; some rule coefficients were about as distinct as the topic coefficients. PD CAT (8) especially had rather strongly positive coefficients for the expected rules.

Not surprisingly, the logistic regression failed to predict to more than some different classes. PD BB (7), PD CAT (8) and RCS-DEV (12) were all represented in the rules set, however, non-expectedly, PD TC - SAM (11) was one of the most recurring labels despite not being represented by any rules in the input. This may be explained by the class distribution in the entire dataset visualised in Figure 2.1. PD TC - SAM (11) is one of the majority classes and is well-represented in all test sets.

Compared to the 40 topics used as a baseline, adding the Apriori rules did not show notable improvement of the accuracy for the logistic regression, as the accuracy merely increased with 0.1 percentage points and DO-probit increased the accuracy with 0.3 percentage points. The deviance of the logistic model with the topics and Apriori rules as covariates resulted in the lowest deviance. The topic set gave a deviance of 12784 and the combined set gave 12491. The accuracy of DO-probit and the reduced deviance of the logistic regression may indicate that the Apriori rules were helpful for prediction. To further investigate this, the topic coefficients with the Apriori rules included in the DO-probit model were plotted against the topic coefficients without the rules included in the model. The coefficients of some classes showed no dependence between adding and not adding the rules (the coefficients remained the same) however, for several classes the absolute values of the topic coefficients decreased when adding the rules. This implies some rules are preferable for predicting some classes. The predictions of BBI (1), CPI (2), HW (5), PD CAT (8), PD Radio (9), PD TC - SAM (11) and RCS-DEV (12) appear to be aided by some rules, and in some sense OSS (6) although the absolutes were rather small for this class. The method should be further investigated with a more extensive dataset.

SPADE

Running SPADE with the same terms as Apriori resulted in a higher number of sequences; 235 versus 115. One can also see that SPADE found sequences connected to more distinct classes (6) than Apriori (3). Both SPADE and Apriori found patterns for PD BB (7), PD CAT (8) and RCS-DEV (12). SPADE also obtained sequences for GWMR (4), PD TC-SAM (11) and PD Radio (9). The logistic model using only SPADE sequences as covariates predicted to the classes CPP (3), PD BB (7), PD CAT (8), PD TC - SAM (11) and RCS-DEV (12). No predictions were made to GWMR (4) and PD Radio (9), which were both represented by sequences in the input set. This can again be explained by the class distributions in the entire data set; GWMR (4) is one of the minority classes whereas CPP (3) is the fourth largest class, increasing the probability of the prediction. Furthermore, PD Radio (9) was merely represented by one sequence in the input, not likely to compensate for the class having less than half of the observations of the three majority classes. The logistic coefficients for the SPADE sequences were typically smaller in magnitude than the Apriori coefficients, which is reasonable since

α was set closer to zero ($\alpha = 0.41$) and for $\alpha = 0$ the elastic net regularisation becomes Ridge regularisation. Unfortunately, the estimated coefficients for the SPADE sequences did, in general, not indicate the expected effect on the classes in neither the logistic regression nor the DO-probit; the coefficients were close to zero for most coefficients expected to be large. In the logistic regression, the coefficients were quite large for sequences in classes where we did not expect the sequence to be active. The DO-probit showed similar results, however, some (not previously expected) specific coefficients were found strongly negative. Naturally, both models showed similar coefficient behavior when adding the topics to the predictor set.

5.2 Method

Data mining as a pre-processing method for event log data

The CDF (the distribution of the weight measure) as well as the parameters used in our proposed confidence measure were chosen by experts in the alarm log area. Better performance can probably be obtained by learning these parameters from data, however, due to computational cost and insufficient data, this has not been attempted here. The Apriori algorithm with the customised measure is (in theory) of less interest than SPADE as SPADE takes the order of events into account, therefore, the customised measure is not likely to be worth further investigation if not attempting to adjust it for obtaining the order of events. Another benefit with SPADE compared to Apriori is the reduced number of required database scans. Apriori requires multiple scans, while SPADE merely requires a maximum of three scans. The largest drawback with SPADE is the need to handle all logs simultaneously in the main memory, which results in a slow mining process [17]. In this thesis, pre-built R-packages were used for Apriori, SPADE and logistic regression. The Apriori algorithm appears to function well, however, the SPADE-implementation proved to have unexpected limitations. SPADE (and the implemented package) was chosen in this thesis due to the possibility of accounting for both the order of events and time elapse between events. The package does not, however, appear to support time elapse given in seconds over several years. This resulted in the need to replace the true time variable with an indexed time-variable of the events rather than the true time elapses, losing the ability to set the algorithm to mine for sequences in a specific time span. Furthermore, as the maximum length of the patterns (for Apriori) and sequences (for SPADE) were set to 4, connections between longer patterns or sequences and a target class may have been overlooked.

Predictions

Multinomial logistic regression with the elastic net penalty is a suitable method for processing extensive covariate vectors with multiple level classes, as one is offered a trade-off between the Ridge penalty and the LASSO penalty [22]. Multinomial probit models are, however, better suited for bayesian classification. The targets of either logit or probit models can be modeled with latent variables to enable Gibbs sampling, however, while the latent distributions of the probit models are straightforward, the distributions of the multinomial logistic models are more complex. The Diagonal Orthant probit models furthermore sidesteps the posterior dependence between latent variables and parameters [12]. The horseshoe shrinkage is also well suitable for sparse covariate matrices as the horseshoe prior can shrink noise variables aggressively while allowing the coefficients of signal variables to remain large [5]. The usage of both global penalty terms and local penalty terms is indeed suitable for this thesis; the obtained rules and sequences regarded specific classes and are therefore mainly local predictors associated with a specific class. The local shrinkage proved to be beneficial for the rules, as DO-probit managed to find which Apriori rules were interesting for finding both positive and negative dependencies between patterns and classes while logistic regres-

sion failed to filter out local noise without removing or excessively regularising the covariates interesting for obtaining negative dependence.

Evaluation

The results lack some level of comparability. Foremost, due to implementation limitations, the DO-probit and the logistic regression were evaluated using different methods. Therefore, the fact that DO-probit often returned a slightly higher accuracy may be an artifact of the evaluation. The logistic regression was tested on an out-of-sample set which was not passed to the Apriori or SPADE algorithms, however, DO-probit was evaluated with cross-validation on the entire data set. Both evaluation methods are appropriate validation methods, however, the DO-probit models were partially trained on the data passed to the data mining algorithms which may bias the results. Furthermore, the elastic net penalty parameter α in the logistic regression was not cross-validated due to insufficient data and the rule mining procedures (Apriori and SPADE) were not cross-validated due to computational cost. Thus, the evaluation is not entirely satisfactory. As a customised confidence measure was used, it would also be interesting to evaluate the differences in performance of the Apriori algorithm with and without this filtering.

A multinomial model with an uninformative Dirichlet prior was fitted to evaluate the predictive performance using only the relative frequencies. The results showed that the test sets were predicted to be one of three classes; PD BB (7), PD CAT (8) or PD TC - SAM (11). These classes were also the most represented classes in the fitted values of the models using any covariate set. This underlines the issues with the distribution of the data. It may, therefore be of interest to attempt under- or oversampling to avoid issues with the imbalanced data. Another solution to the problem would be to further aggregate the classes, however, this was not performed in this thesis as the classes were likely to become too wide.

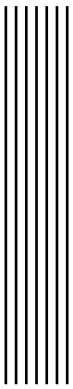
6 Conclusion

Are alarm log files useful for automated bug report routing?

The alarm data appears to be useful for predicting some classes, however, all developer groups are not likely to be directly connected to events in the alarm log which may have resulted in the low number of developer groups represented in the rule sets. The combination of the information available connected to a bug report (observations field, customer information, logs, etcetera) may well be better to use rather than each piece of information separately, especially as it cannot be expected that specific bugs solvable by one developer group show symptoms in the alarm logs. The same problem is a fact for the observations field as well, as all models attempted in this thesis failed to make any predictions, correctly or incorrectly, to the classes CPI, OSS, System and Tools, and some Apriori rules replaced the topics in the comparison. The failed predictions can, again, be partially explained by the insufficient class distribution of the data, as they are all small classes. It is, however, desired to replace topics with alarms as alarms are easier to interpret. The conclusion of this question is that the alarm data is indeed useful for making more confident predictions of some classes.

Is association analysis an effective approach for constructing features from system log files?

For the logistic regression, the rules achieved by Apriori did in general yield a large positive coefficient in the equations of the expected classes. Some rules were used as predictors for several classes, which can be of interest as a set of alarms found in the alarm log may be of significant value when determining what developer groups are unlikely to solve the bug. The SPADE sequences did not to the same extent contribute to the predictions of the expected classes. The limitations in the package implementation may, however, be a factor. SPADE may, therefore, have missed potentially interesting patterns and should be rerun with a different code base. The differences in accuracy for the two mining approaches appear insignificant. Furthermore, as most observations were predicted to one specific class, the accuracy is close to non-interesting. The SPADE sequences and topics set yielded a higher deviance than the Apriori rules and the topics set. Apriori with the customised confidence measure proved more rewarding than SPADE, as the coefficients of the Apriori rules behaved more as expected in comparison to the SPADE coefficients.



Bibliography

[1] R. Agrawal and R. Srikant. "Fast Algorithms for Mining Association Rules". In: *Proc. of the 20th Int. Conf. on Very Large Databases* (1994), pp. 487–499.

[2] J. Anvik, L. Hiew, and G. C. Murphy. "Who should fix this bug?" In: *Proceedings of the 28th international conference on Software engineering* (2006), pp. 361–370.

[3] C.M. Bishop. *Pattern recognition and machine learning*. 12th ed. Springer, 2006, pp. 32–33, 197–209, 685–691.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. "Latent Dirichlet Allocation". In: *Journal of Machine Learning Research* 3.4–5 (2003), pp. 993–1022.

[5] C. M. Carvalho, N. G. Polson, and J. G. Scott. "The horseshoe estimator for sparse signals". In: *Biometrika* 97.2 (2001), pp. 465–480.

[6] J. Friedman, T. Hastie, and R. Tibshirani. "Regularization Paths for Generalized Linear Models via Coordinate Descent". In: *Journal of Statistical Software* 33.1 (2010), pp. 1–22.

[7] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*. 3rd ed. Chapman & Hall/CRC Press, 2013.

[8] T. Griffiths and M. Steyvers. "Finding scientific topics". In: *Proceedings of the National Academy of Sciences* 101.1 (2004), pp. 5228–5235.

[9] M. Hahsler, B. Gruen, and K. Hornik. "arules – A Computational Environment for Mining Association Rules and Frequent Item Sets". In: *Journal of Statistical Software* 14.15 (Oct. 2005), pp. 1–25.

[10] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. 2nd ed. Springer, 2009, pp. 242, 314.

[11] A. Hoerl and R. Kennard. "Ridge regression". In: *Encyclopedia of Statistical Sciences* 8.1 (1988), pp. 129–136.

[12] J. E. Johndrow, K. Lum, and D. B. Dunson. "Diagonal Orthant Multinomial Probit Models". In: *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics* (2013), pp. 29–38.

[13] L. Jonsson, D. Broman, M. Magnusson, K. Sandahl, M. Villani, and S. Eldh. "Automatic Localization of Bugs to Faulty Components in Large Scale Software Systems using Bayesian Classification". In: *Proc. of IEEE Int. Conf. on Software Quality, Reliability & Security* (2016), pp. 423–430.

[14] D. Kim, Y. Tao, S. Kim, and A. Zeller. "Where should we fix this bug? A two-phase recommendation model". In: *IEEE Transactions on Software Engineering* 39.11 (2013), pp. 1597–1610.

[15] M. Magnusson, L. Jonsson, and M. Villani. "DOLDA - a regularized supervised topic model for high-dimensional multi-class regression". In: *ArXiv e-prints* (2016).

[16] M. Magnusson, L. Jonsson, M. Villani, and D. Broman. "Sparse Partially Collapsed MCMC for Parallel inference in Topic Models". In: *arXiv:1506.03784v2* (2016).

[17] T. Mitsa. *Temporal Data Mining*. Chapman & Hall/CRC Press, 2006, pp. 159–160.

[18] C. Parnin and A. Orso. "Are automated debugging techniques actually helping programmers?" In: *Proceedings of the 2011 International Symposium on Software Testing and Analysis ISSTA 11* (2011), pp. 199–209.

[19] R. Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society B* 58.1 (1996), pp. 267–288.

[20] W. E. Wong and V. Debroy. "A survey on software fault localization". In: *IEEE transactions on software engineering* 42.8 (2016), pp. 707–740.

[21] M. J. Zaki. "SPADE: An Efficient Algorithm for Mining Frequent Sequences". In: *Machine Learning* 42.1 (2001), pp. 31–60.

[22] H. Zou and T. Hastie. "Regularization and Variable Selection via the Elastic Net." In: *Journal of the Royal Statistical Society B* 67.2 (2005), pp. 301–320.

A Confusion matrices

Predicted class \ True class	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	**0**	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	**0**	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	**0**	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	**11**	0	1	2	2	0	0	0
8	1	0	2	0	2	0	0	**31**	4	2	2	1	1	0
9	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0
11	52	10	156	59	43	7	297	329	131	141	**325**	84	29	34
12	0	0	0	2	0	0	0	2	0	0	0	**5**	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**

Table A.1: Confusion matrix for Apriori rules, logistic regression

Predicted class \ True class	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	**0**	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	**0**	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	**0**	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	**60**	1	1	2	2	0	0	0
8	13	5	56	22	13	1	113	**217**	47	54	123	26	10	12
9	0	0	0	0	0	0	0	2	**0**	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0
11	175	47	476	186	121	9	909	976	358	427	**984**	243	86	97
12	0	0	0	1	0	0	0	2	0	0	0	**14**	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**

Table A.2: Confusion matrix for Apriori rules, DO-probit

		True class													
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Predicted class	**1**	**3**	0	0	0	0	0	1	0	0	0	3	0	0	0
	2	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0
	3	5	0	**46**	0	7	3	4	23	6	8	18	0	1	4
	4	5	0	1	**25**	0	0	4	11	1	6	7	13	0	0
	5	0	0	1	0	**0**	0	0	1	0	0	0	0	0	0
	6	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0
	7	20	3	17	1	2	0	**191**	9	15	23	95	3	13	17
	8	5	3	48	23	19	3	18	**265**	62	54	59	37	6	6
	9	0	1	13	0	13	0	1	18	**40**	3	0	0	0	0
	10	0	0	3	1	0	0	3	3	0	**8**	8	0	1	0
	11	14	2	27	4	4	0	77	17	8	32	**136**	2	9	7
	12	1	1	2	7	0	1	9	15	4	11	3	**35**	0	0
	13	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**

Table A.3: Confusion matrix for Apriori rules and topics, logistic regression

		True class													
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Predicted class	**1**	**3**	0	0	0	1	0	0	0	0	0	2	0	0	0
	2	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0
	3	18	2	**138**	0	10	3	17	67	10	26	60	4	4	6
	4	17	1	4	**79**	0	0	18	22	3	15	25	33	0	0
	5	0	0	0	0	**0**	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0
	7	80	8	63	12	18	0	**673**	35	45	88	319	3	36	68
	8	26	15	205	84	52	4	82	**930**	201	206	185	126	18	24
	9	0	2	37	0	41	1	5	47	**110**	7	5	2	4	0
	10	0	0	3	2	0	0	5	4	1	**18**	11	1	0	0
	11	40	22	81	13	12	0	264	61	28	103	**492**	12	34	11
	12	4	2	1	19	0	2	18	32	8	20	10	**102**	0	0
	13	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**

Table A.4: Confusion matrix for Apriori rules and topics, DO-probit

		True class													
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Predicted class	**1**	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0
	3	1	0	**0**	0	1	0	0	0	0	0	0	0	0	0
	4	0	0	0	**0**	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	**0**	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	**3**	4	2	0	4	1	0	0
	8	52	10	157	58	44	7	303	**355**	132	143	315	86	29	33
	9	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0
	11	0	0	1	0	0	0	1	3	2	1	**9**	2	1	1
	12	0	0	0	3	0	0	1	0	0	1	1	**1**	0	0
	13	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**

Table A.5: Confusion matrix for SPADE sequences, logistic regression

		True class													
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Predicted class	**1**	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0
	3	1	0	**0**	0	2	0	0	1	0	0	0	0	0	0
	4	0	0	0	**0**	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	**0**	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	**0**	1	2	1	2	1	0	0
	8	187	52	529	205	132	10	1072	**1183**	401	478	1083	268	95	107
	9	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0
	10	0	0	0	0	0	0	1	0	0	**0**	0	1	0	0
	11	0	0	3	0	0	0	7	7	2	1	**19**	3	1	2
	12	0	0	0	4	0	0	2	6	1	3	5	**10**	0	0
	13	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**

Table A.6: Confusion matrix for SPADE sequences, DO-probit

		True class													
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Predicted class	**1**	**3**	0	0	0	0	0	1	0	0	0	3	0	0	0
	2	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0
	3	5	0	**42**	0	6	2	6	20	7	10	15	0	1	4
	4	5	0	1	**22**	0	0	5	8	1	5	6	12	0	0
	5	0	0	1	0	**0**	0	0	1	0	0	0	0	0	0
	6	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0
	7	22	3	19	2	3	0	**198**	14	17	23	99	4	13	17
	8	5	3	56	24	20	4	20	**268**	64	56	60	36	5	7
	9	0	1	12	0	12	0	1	17	**38**	2	0	0	0	0
	10	0	1	3	1	0	0	4	5	0	**9**	7	2	1	0
	11	12	2	21	4	4	0	66	16	5	30	**136**	2	10	6
	12	1	0	3	8	0	1	7	13	4	10	2	**34**	0	0
	13	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0
	14	0	0	0	0	0	0	0	0	0	0	1	0	0	**0**

Table A.7: Confusion matrix for SPADE sequences and topics, logistic regression

		True class													
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Predicted class	**1**	**3**	0	0	0	1	0	0	0	0	0	1	0	0	0
	2	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0
	3	16	2	**127**	0	10	2	18	68	11	27	61	2	4	5
	4	17	1	3	**75**	1	0	21	20	3	15	24	33	0	0
	5	0	0	1	0	**0**	0	0	1	0	0	0	0	0	0
	6	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0
	7	86	11	69	15	19	0	**683**	42	49	92	338	4	42	70
	8	32	18	220	91	55	5	94	**939**	206	214	196	123	17	25
	9	0	1	33	0	36	1	5	35	**108**	7	4	2	3	0
	10	0	0	4	1	0	0	6	4	1	**19**	9	0	0	0
	11	31	17	72	9	12	0	238	62	19	92	**464**	10	30	9
	12	3	2	3	18	0	2	17	28	9	17	11	**109**	0	0
	13	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0
	14	0	0	0	0	0	0	0	0	0	0	1	0	0	**0**

Table A.8: Confusion matrix for SPADE sequences and topics, DO-probit

B MCMC chains of log-posteriors

The first four folds for each model are presented. The remaining six folds are not presented as they show similar behavior.

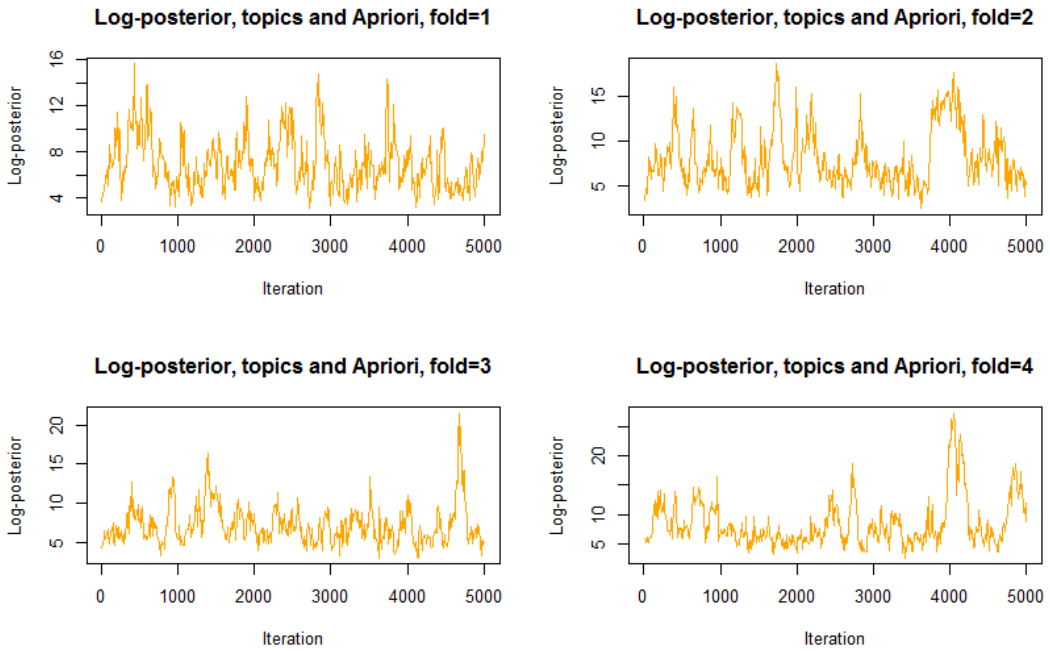


Figure B.1: The DO-probit log-posterior for the first four folds using Apriori rules and topics.

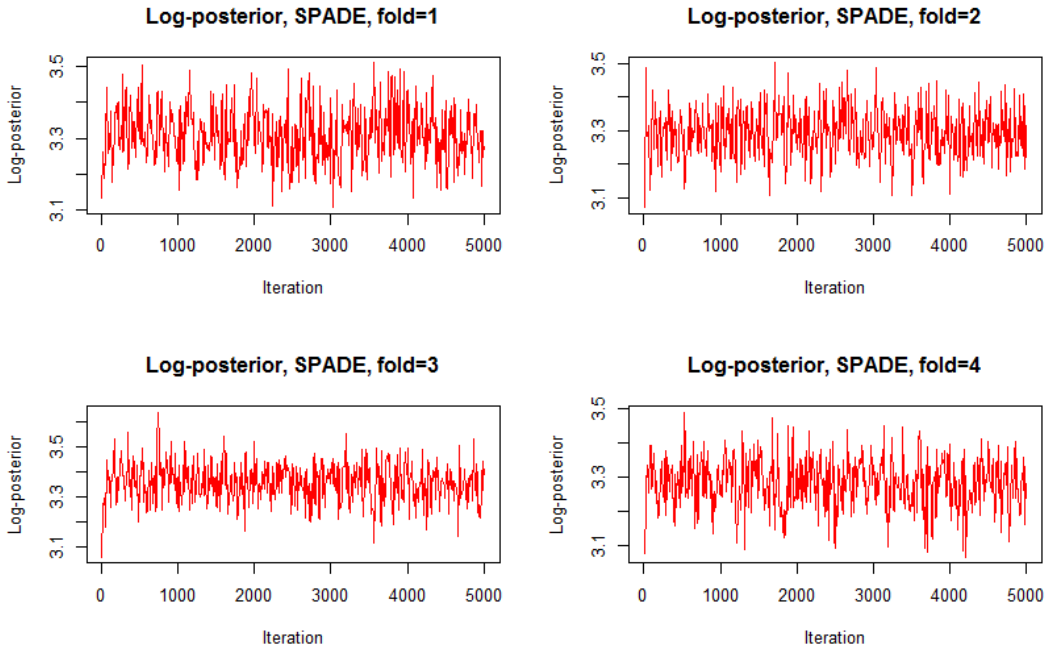


Figure B.2: The DO-probit log-posterior for the first four folds using SPADE sequences.

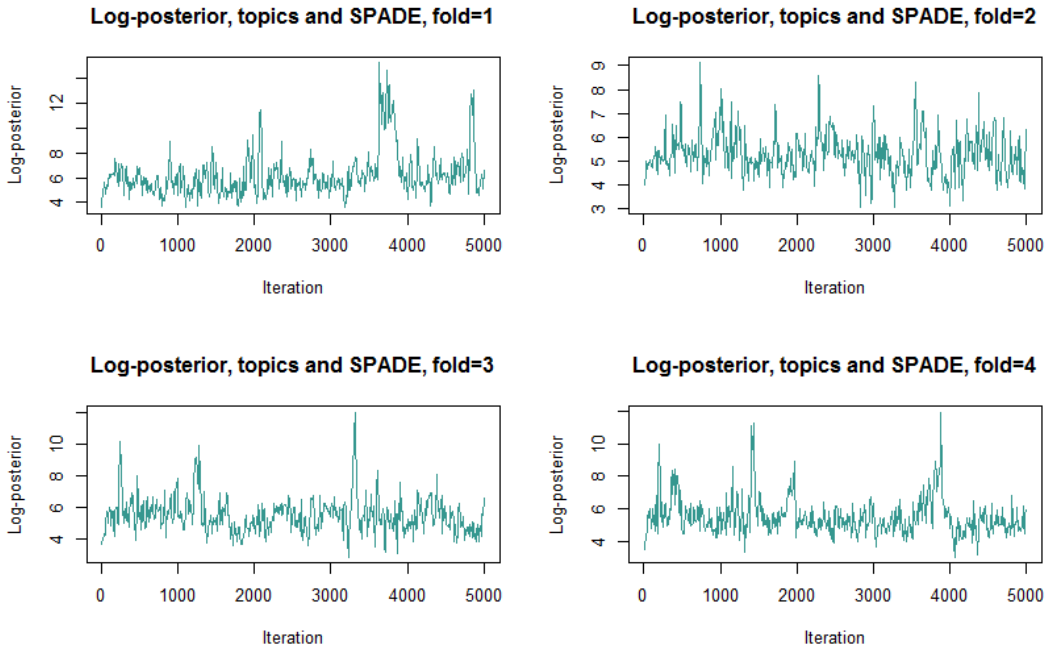


Figure B.3: The DO-probit log-posterior for the first four folds using SPADE sequences and topics.

C Results for topics as covariate set

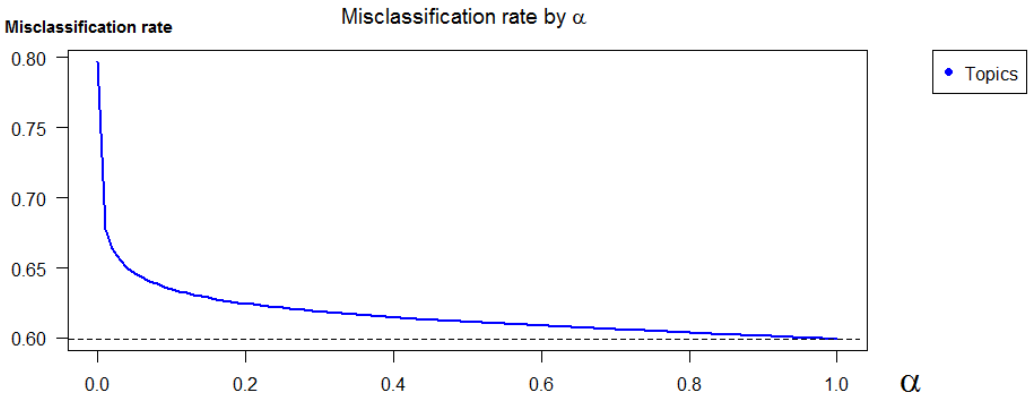


Figure C.1: Misclassification rate for different values of α for observations field topics.

		True class													
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Predicted class	**1**	**3**	0	0	0	1	0	1	0	0	0	3	0	0	0
	2	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0
	3	5	0	**41**	0	5	2	4	20	6	8	15	0	1	4
	4	5	0	1	**23**	0	0	4	8	0	6	7	12	0	0
	5	0	0	1	0	**0**	0	0	1	0	0	0	0	0	0
	6	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0
	7	21	3	19	1	3	0	**199**	10	18	24	103	3	14	17
	8	5	3	57	25	19	4	20	**270**	65	55	61	38	5	6
	9	0	1	12	0	13	0	1	18	**38**	2	0	0	0	0
	10	0	0	3	1	0	0	3	4	0	**8**	7	0	1	0
	11	13	2	21	4	5	0	67	17	5	31	**130**	2	9	7
	12	1	1	3	7	0	1	9	14	4	11	3	**35**	0	0
	13	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**

Table C.1: Confusion matrix for topics, logistic regression

The resulting confusion matrix when the topics were modeled with a DO-probit is found in Table C.2 .

		True class													
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Predicted class	**1**	**4**	0	0	0	1	0	0	0	0	0	3	0	0	0
	2	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0
	3	18	2	**121**	0	8	1	17	62	12	27	57	3	4	5
	4	17	1	3	**75**	1	0	17	18	3	14	19	36	0	0
	5	0	0	0	0	**0**	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0
	7	86	10	66	14	20	0	**687**	42	48	97	328	5	43	69
	8	27	15	222	91	54	5	87	**938**	203	212	205	126	18	25
	9	0	2	35	0	37	1	4	39	**110**	6	5	2	2	0
	10	1	0	4	1	0	0	6	5	1	**19**	8	1	0	0
	11	32	20	78	9	13	1	243	61	20	89	**473**	12	29	10
	12	3	2	3	19	0	2	21	33	9	19	11	**98**	0	0
	13	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**

Table C.2: Confusion matrix for topics, DO-probit

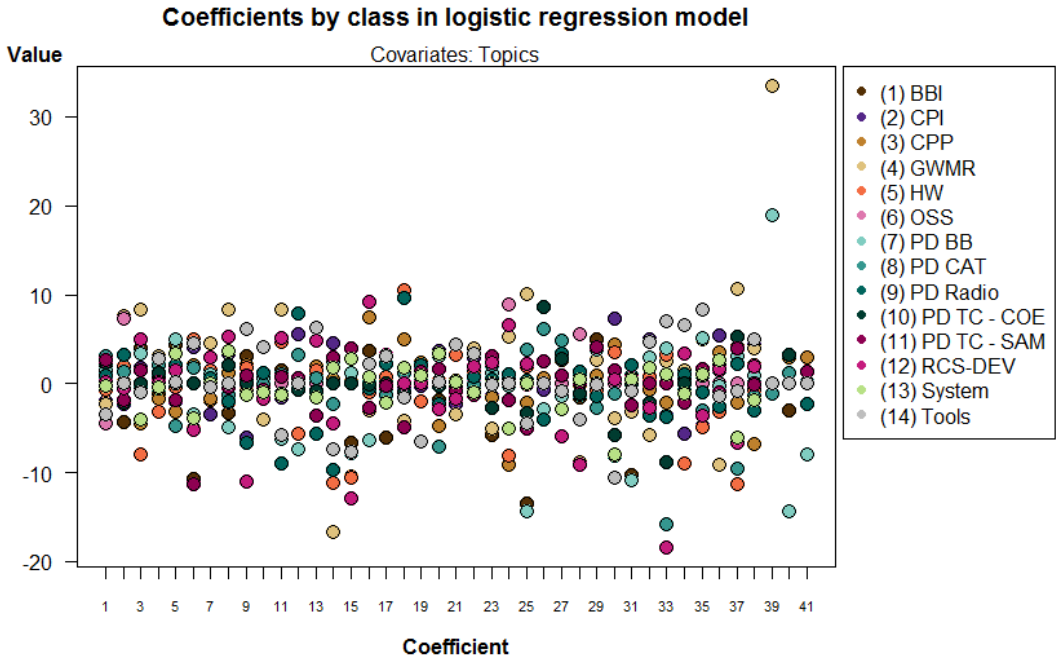


Figure C.2: Beta coefficients of logistic regression model with observations field topics as covariates.

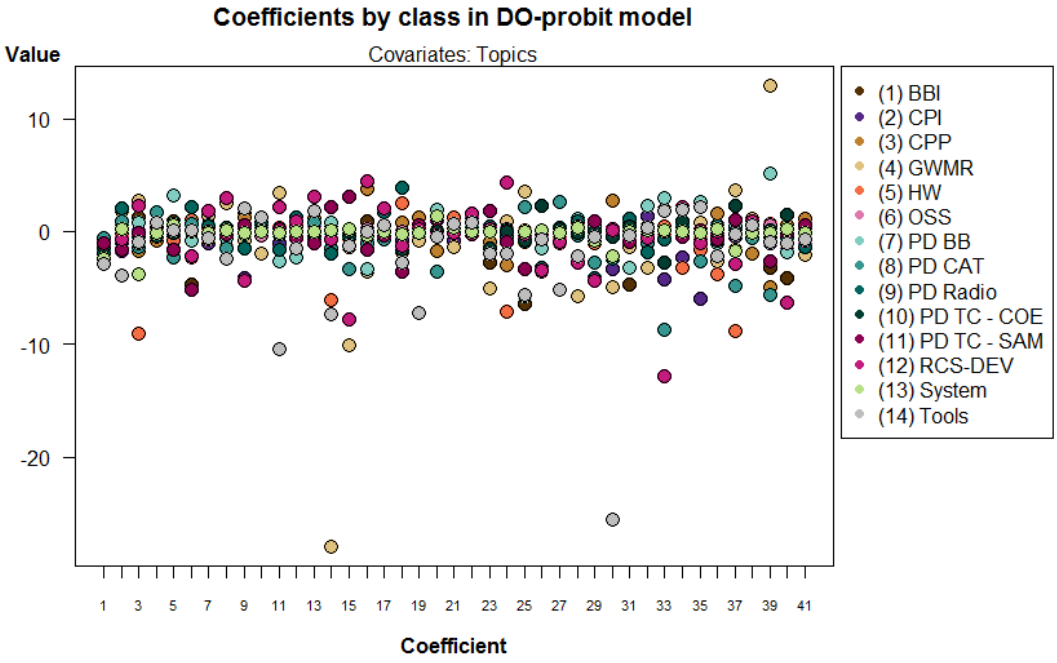


Figure C.3: Beta coefficients of DO-probit model with observations field topics as covariates.

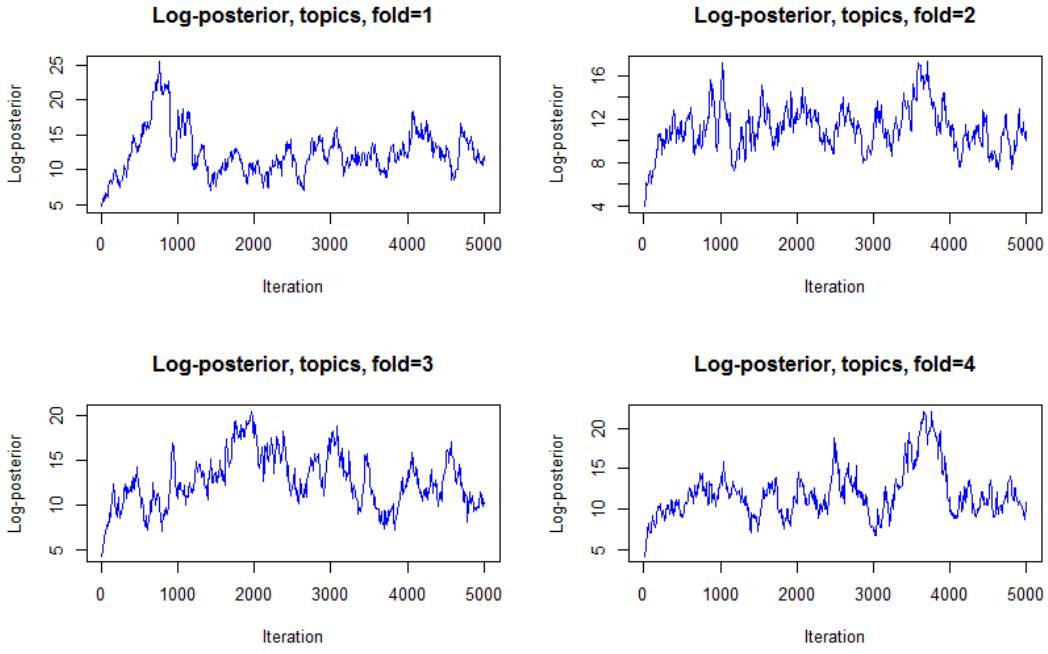


Figure C.4: The DO-probit log-posterior for the first four folds using the topics as covariates.

LIU-IDA/STAT-A--17/009—SE