Scalable Estimates of Concept Stability

Aleksey Buzmakov, Sergei O. Kuznetsov, Amedeo Napoli

Scalable Estimates of Concept Stability

Aleksey Buzmakov[1,2], Sergei O. Kuznetsov[2], and Amedeo Napoli[1]

[1] LORIA (CNRS – Inria NGE – U. de Lorraine), Vandœuvre-lès-Nancy, France
[2] National Research University Higher School of Economics, Moscow, Russia
aleksey.buzmakov@inria.fr, amedeo.napoli@loria.fr, skuznetsov@hse.ru

Abstract. Data mining aims at finding interesting patterns from datasets, where "interesting" means reflecting intrinsic dependencies in the domain of interest rather than just in the dataset. Concept stability is a popular relevancy measure in FCA. Experimental results of this paper show that high stability of a concept for a context derived from the general population suggests that concepts with the same intent in other samples drawn from the population have also high stability. A new estimate of stability is introduced and studied. It is experimentally shown that the introduced estimate gives a better approximation than the Monte Carlo approach introduced earlier.

Keywords: formal concept analysis, stability, pattern selection, experiment

1 Introduction

Given a dataset, data mining methods may reveal a huge number of patterns, so filtering patterns w.r.t. some relevancy measures can be necessary. The question of how much a pattern is interesting arises in many areas of data mining, including those that employ tools of Formal Concept Analysis (FCA). FCA is a mathematical formalism having many applications in data analysis [1]. It aims at computing concepts and their lattices from a formal context, a triple (G, M, I) where G is a set of objects (experiments or elements of a dataset), M is a set of attributes used to build the description of every object, and $I \subseteq G \times M$ is a relation between objects and attributes. The number of concepts for a given context can be exponential w.r.t. the size of the context, and thus, a special procedure for selecting the most relevant concepts is needed. Two options can be distinguished. The first one is to introduce background knowledge into the procedure for computing concepts [2–6]. Background knowledge allows one to sort concepts which are likely to be useful for the current goal. In this case, although the number of concepts can be significantly reduced, the size of the lattice can still be huge. The second option is to rank concepts in the lattice using a relevance measure.

The authors of [7] provide several measures for ranking concepts that stem from human behavior. Stability is another measure for ranking concepts, introduced in [8] and later revised in [9–11]. Several other methods are considered

	m_1	m_2	m_3	m_4	m_5	m_6
g_1	x					x
g_2		x				x
g_3			x			x
g_4				x		x
g_5					x	

Table 1: A toy formal context.

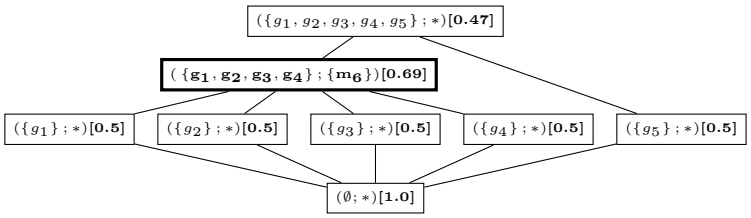


Fig. 1: Concept Lattice for Table 1 with corresponding stability indexes.

in [12], where it is shown that stability is more reliable for artificially noised data. Although there is a number of methods for ranking concepts, there is neither a reliable comparison nor a deep research on relevancy of the selection methods mentioned above. In this work we focus on the stability measure and its estimates. The intuition behind stability is the probability of preserving the concept intent when some objects of the context are removed. In this paper we study the behavior of stability computed in several datasets coming from the same general population. It is done by spliting given datasets into two disjoint subsets called reference and test datasets. The stability behaviour is shown to be similar in reference and test datasets independently of the general population.

Since computing stability is #P-complete [8, 9] one needs to use estimates or approximations in order to compute stability over large lattices. Correspondingly, in the second part of our paper we introduce estimates of stability. It is shown empirically that their performance is better then the performance of the known Monte Carlo approximation [13].

The rest of the paper is organized as follows. Section 2 introduces the formal definition of stability, its estimate and Monte Carlo approximation and discusses their relation. In Section 3 experiments on relevancy of stability are explained and discussed. Then Section 4 validates the introduced estimate.

2 Stability of a Formal Concept

2.1 The Definition of Stability

Stability is a relevancy measure of a formal concept introduced in [8] and later revised in [9–11].

Function `FindStabilityLimits`
Data: A context $\mathbb{K} = (G, M, I)$, A concept C.
Result: $< Left, Right >$, a pair of left and right limits for the stability.
$Left \leftarrow 1$;
$Right \leftarrow 1$;
$children \leftarrow$ `FindChildren(`$\mathbb{K}, C$`)` ; `/*` $O(|N| \cdot |M|^2$ `*/`
$minDiffSize \leftarrow \infty$;
foreach $ch \in children$ **do** `/*` $O(|M|)$ `iterations at most */`
  $diffSize \leftarrow |$`Ext(`C`)` $\setminus$ `Ext(`ch`)`$|$;
  $minDiffSize \leftarrow \min(minDiffSize, diffSize)$;
  $Left \leftarrow Left - 2^{-diffSize}$;
$Right \leftarrow 1 - 2^{-minDiffSize}$;
return $< Left, Right >$;

Algorithm 1: An algorithm computing stability bounds according to (2)

Function `FindStabilityLimitsPlusMC`
Input: Context $\mathbb{K} = (G, M, I)$; concept C; precision ε and error rate δ for Monte-Carlo.
Output: $< Left, Right >$, a pair of left and right limits for the stability.
$< Left, Right > \leftarrow$ `FindStabilityLimits(`$\mathbb{K}, C$`)`;
if $Right - Left > 2 \cdot \varepsilon$ **then**
  $stabilityMC \leftarrow$ `FindStabilityByMonteCarlo(`$\mathbb{K}, C, \varepsilon, \delta$`)`;
  $Left \leftarrow \max(Left, stabilityMC - \varepsilon)$;
  $Right \leftarrow \min(Right, stabilityMC + \varepsilon)$;
return $< Left, Right >$;

Algorithm 2: An algorithm based on combination of (2) and Monte-Carlo approach.

Definition 1. *Given a concept c, concept stability $Stab(c)$ is defined as*

$$Stab(c) := \frac{|\{s \in \wp(Ext(c)) \mid s' = Int(c)\}|}{2^{|Ext(c)|}} \tag{1}$$

i.e. the relative number of subsets of the concept extent (denoted by $Ext(c)$), whose description (i.e. the result of $(\cdot)'$) is equal to the concept intent (denoted by $Int(c)$) where $\wp(P)$ is the power set of P.

Example 1. Figure 1 shows a lattice for the context in Table 1, for simplicity some intents are not given. The extent of the highlighted concept c is $Ext(c) = \{g_1, g_2, g_3, g_4\}$, thus, its power set contains 2^4 elements. The descriptions of 5 subsets of $Ext(c)$ ($\{g_1\}, \ldots, \{g_4\}$ and $\emptyset$) are different from $Int(c) = \{m_6\}$, while all other subsets of $Ext(c)$ have a description equal to $\{m_6\}$. So, $Stab(c) = \frac{2^4-5}{2^4} = 0.69$.

Stability measures the dependence of a concept intent on objects of the concept extent. More precisely this intuition behind stability can be described by the following proposition originally introduced in [14, 11].

Proposition 1. *Let $\mathbb{K} = (G, M, I)$ be a formal context and c a formal concept of $\mathbb{K}$. For a set $H \subseteq G$, let $I_H = I \cap H \times M$ and $\mathbb{K}_H = (H, M, I_H)$. Then,*

$$Stab(c) = \frac{|\{\mathbb{K}_H \mid H \subseteq G \text{ and } Int(c) \text{ is closed in } \mathbb{K}_H\}}{2^{|G|}}$$

The proposition says that stability of a concept c is the relative number of subcontexts where there exists the concept c with intent $Int(c)$. A stable concept can be found in many such subcontexts, and therefore is likely to be found in an unrelated context built from the population under study. This "likely" was never studied and one of the goals of this paper is to check if stability is useful to find significant patterns within the whole population.

It was shown that, given a context and a concept, the computation of concept stability is #P-complete [8, 9]. One of the fastest algorithm for processing concept stability using a concept lattice L is proposed in [11], with a worst-case complexity of $O(L^2)$, where L is the size of the concept lattice. This theoretical complexity bound is significantly higher than that of algorithms computing all formal concepts and in practice computing stability may take much more time than lattice building algorithms [15]. Moreover, this algorithm needs the lattice structure to be computed, requiring additional computations and memory usage. Thus, finding a good estimate of concept stability is an important question. Here we present an efficient way for such an estimate.

2.2 Estimation of Stability

Given a concept c and its descendant d, we have $(\forall s \subseteq Ext(d))(s'' \subseteq Ext(d) \wedge s' \supseteq Int(d) \supset Int(c))$ i.e. $s' \neq Int(c)$. Thus, we can exclude all subsets of the extent of a descendant while computing the numerator of stability in (1). On the other hand only subsets of the extents of descendants should be excluded from the numerator in (1). Thus, if we exclude the subsets of the extents of all immediate descendants, we exclude everything that is needed but probably some subsets can be excluded several times. Hence we obtain a lower bound for stability:

$$1 - \sum_{d \in DD(c)} \frac{1}{2^{\Delta(c,d)}} \leq Stab(c) \leq 1 - \max_{d \in DD(c)} \frac{1}{2^{\Delta(c,d)}}, \tag{2}$$

where $DD(c)$ is a set of all direct descendants of c in the lattice and $\Delta(c, d)$ is the size of the set-difference between extent of c and extent of d, i.e. $\Delta(c, d) = |Ext(c) \setminus Ext(d)|$. The pseudo-code for computing this estimate is shown in Algorithm 1. The time complexity of this approach for a concept is equal to the complexity of finding immediate descendants of the concept, i.e. $O(n \cdot m^2)$.

Example 2. If we want to compute stable concepts (with stability more than 0.97), then according to the upper bound in (2) we should compute for each

concept c in the lattice $\Delta_{\min}(c) = \min\limits_{d \in DD(c)} \Delta(c,d)$ and select concepts obeying $\Delta_{\min}(c) \geq -\log(1-0.97) = 5.06$.

The upper bound of the equation can be found in [11], while the lower bound has not been studied yet. We know that given a context (G, M, I), the number of children for any concept is limited by cardinality of M. Every summand in the lower bound of stability in (2) is smaller than $2^{-\Delta_{\min}(c)}$. This gives the following estimate.

$$1 - |M| \cdot 2^{-\Delta_{\min}(c)} \leq 1 - \sum_{d \in DD(c)} 2^{-\Delta(c,d)} \leq Stab(c) \tag{3}$$

This suggests that stability can have an exponential behavior w.r.t. the size of the context and, thus, most of the concepts have stability close to 1 when the size of the context increases. This behavior of stability is also noticed by authors of [16] for their dataset. So, to use stability for large datasets it is worth computing logarithmic stability for every concept c:

$$LStab(c) = -\log_2(1 - Stab(c)) \tag{4}$$

Taking into account the bounds in (2) and in (3), we have the following:

$$\Delta_{\min}(c) - \log_2(|M|) \leq -\log_2(\sum_{d \in DD(c)} 2^{-\Delta(c,d)}) \leq LStab(c) \leq \Delta_{\min}(c) \tag{5}$$

This approach is referred as the *bounding method*. It can efficiently bound stability for any concept of the lattice. However, the tightness of this bound cannot be ensured.

In [13] the authors suggest a method for approximating concept stability based on a Monte Carlo approach. Given a concept c, the idea is to randomly count the number of "good" subsets $s \subseteq Ext(c)$ of the extent of c such that $s' = Int(c)$. Then knowing the number of iterations N and the number of "good" subsets N_{good}, stability can be calculated as the relation between them: $Stab(c) = \frac{N_{good}}{N}$. In their paper authors provide the following approximation of the number of iterations:

$$N > \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta} \tag{6}$$

where ε is the precision of the approximation and δ is the error rate, i.e. if one have computed stability approximation s, then the exact value of stability is within the interval $[s - \varepsilon; s + \varepsilon]$ with the probability $1 - \delta$. This method will be later referred as the *Monte Carlo method*.

Example 3. In order to approximate stability with precision $\varepsilon = 0.01$ and error rate $\delta = 0.01$, it is necessary to make at least $N = 2.65 \cdot 10^4$ iterations.

Example 3 shows that the number of iterations for one concept can be huge and, thus, the Monte Carlo method should be less efficient than the bounding method. Nevertheless the Monte Carlo method can ensure a certain level of tightness. Consequently the bounding method and the Monte Carlo method

Dataset	Shortcut	Size	Max. Size	Max. Lat. Size	Lat. Time	Stab. Time
Mushrooms[1]	Mush	8124	8124	$2.3 \cdot 10^5$	324	57
Plants[2]	Plants	34781	1000	$2 \cdot 10^6$	45	10^4
Chess[3]	Chess	3198	100	$2 \cdot 10^6$	30	$7.4 \cdot 10^3$
Solar Flare (II)[4]	Flare	1066	1066	2988	0	0
Nursery[5]	Nurs	12960	12960	$1.2 \cdot 10^5$	245	5

Table 2: Datasets used in the experiments. Column 'Shortcut' refers to the short name of the dataset used in the rest of the paper; 'Size' is the number of objects in the dataset; 'Max. Size' is the maximal number of objects in a random subset of the dataset the lattice structure can be computed for; 'Max. Lat. Size' is the size of the corresponding lattice; 'Lat. Time' is the time in seconds for computing this lattice; 'Stab. Time' is the time in seconds to compute stability for every concept in the maximal lattice.

can be used in a complementary way as follows. First, the stability bounds are computed. Second, if the tightness of the bounding method is worse than the tightness of the Monte Carlo method, the latter should be applied. The pseudo-code of this approach is shown in Algorithm 2. In this paper it is referred as the *combined method*.

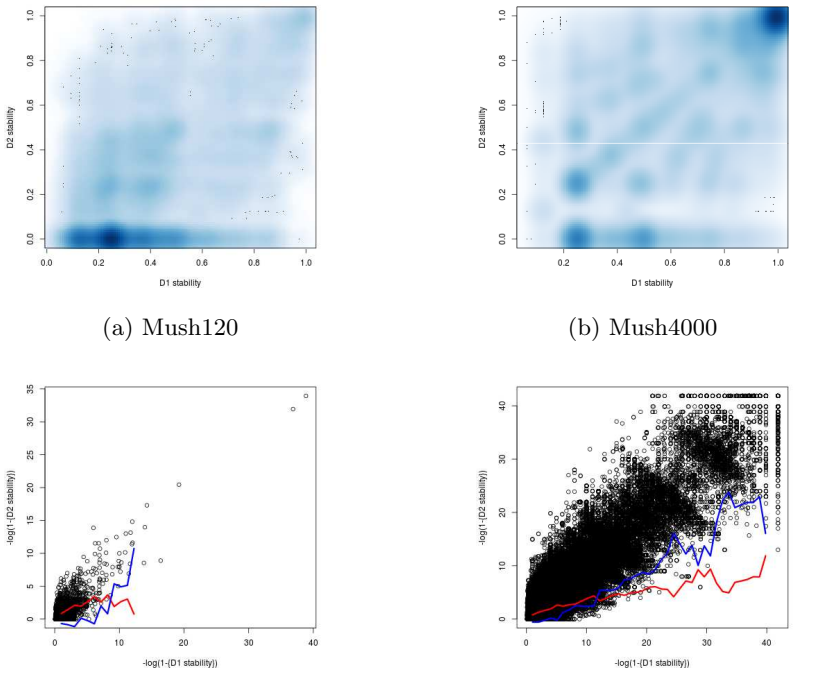
Recall that there are three other estimates of stability [8, 9, 11] whose study is out of the scope of the present paper. Two of these estimates are applicable incrementally, i.e. when stability is known for a concept from some context and several objects are added to this context authors estimate the stability of the corresponding concept in the new lattice. For the third estimate no efficient computation is known for the moment.

In the next section we present two types of experiments. In Subsection 3.1 an experiment on the predictability of stability is presented. The discussion continues in Subsection 3.3 with the behaviour of stability thresholds and in Subsection 3.4 with stability ordering ability.

3 Experiment on Predictability of Stability

The experiments are run on an "Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz" computer with 8Gb of memory under Ubuntu 12. The algorithms are not parallelized. Public datasets available from the UCI repository [17] are used for the experimentation. These datasets are shown in Table 2. With their different size and complexity, these datasets provide a rich experimental basis. Complexity here stands for the size of the concept lattice given the initial number of objects in the corresponding context. For example, `Chess` is the most complex dataset

[1] `http://archive.ics.uci.edu/ml/datasets/Mushroom`
[2] `http://archive.ics.uci.edu/ml/machine-learning-databases/plants/`
[3] `http://archive.ics.uci.edu/ml/datasets/Chess+(King-Rook+vs.+King-Pawn)`
[4] `http://archive.ics.uci.edu/ml/datasets/Solar+Flare`
[5] `http://archive.ics.uci.edu/ml/datasets/Nursery`



(a) Mush120

(b) Mush4000

(c) Mush120 logarihtmic scale

(d) Mush4000 logarihtmic scale

Fig. 2: Stability in the test dataset w.r.t the reference one.

as for only 100 objects in the context there are already $2 \cdot 10^6$ of concepts in the concept lattice.

3.1 The Experiment Flow

Recall that the stability of a concept c can be considered as the probability for the intent of c to be preserved in the lattice when some objects are removed. However, when computing stability, one wants to know if the intent of a stable concept is a general characteristic rather than an artefact specific for a dataset. For that it is necessary to evaluate stability w.r.t. a test dataset different from the reference one. Reference and test datasets are two names of disjoint datasets on which the stability behaviour is evaluated. In order to do that the following scheme of experiment is developed:

1. Given a dataset $\mathbb{K}$ of size K objects, experiments are performed on dataset subsets whose size in terms of number of objects is N. This size is required to be at least half the size of K. For example, for a dataset of size $K = 10$ the size of it subset can be $N = 4$.

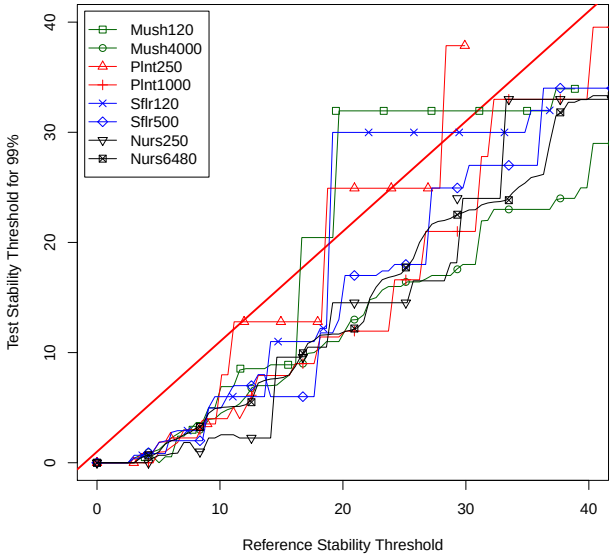


Fig. 3: Stability threshold in the test dataset ensuring that 99% of concepts corresponding to stable ones in the reference dataset are stable.

2. Two disjoint dataset subsets $\mathbb{K}_1$ and $\mathbb{K}_2$ of size N (in terms of objects) of dataset $\mathbb{K}$ are generated by sampling, e.g. $\mathbb{K}_1 = \{g_2, g_5, g_6, g_9\}$ and $K_2 = \{g_3, g_7, g_8, g_{10}\}$. Later, $\mathbb{K}_1$ is used as a reference dataset for computing stability, while $\mathbb{K}_2$ is a test dataset for evaluating stability computed in $\mathbb{K}_1$.
3. The corresponding sets of concepts $\mathcal{L}_1$ and $\mathcal{L}_2$ with their stability are built for both datasets $\mathbb{K}_1$ and $\mathbb{K}_2$.
4. The concepts with the same intents in $\mathcal{L}_1$ and $\mathcal{L}_2$ are declared as corresponding concepts.
5. Based on this list of corresponding concepts, a list of pairs $S = \{\langle X, Y\rangle, \dots\}$ is built, where X is the stability of the concept in $\mathcal{L}_1$ and Y is the stability of the corresponding concept in $\mathcal{L}_2$. If an intent exists only in one dataset, its stability is set to zero in the other dataset (following the definition of stability). Finally, the list $LS = \{\langle X_{\log}, Y_{\log}\rangle, \dots\}$ includes the stability pairs in S in logarithmic scale as stated in formula (4).
6. Then sets of pairs S and LS are further used to study the behaviour of stability on disjoint (independent) datasets coming from the same general population.

The idea of evaluating stability computed on a reference dataset w.r.t. a test dataset comes from the supervised classification methods. Moreover, this idea

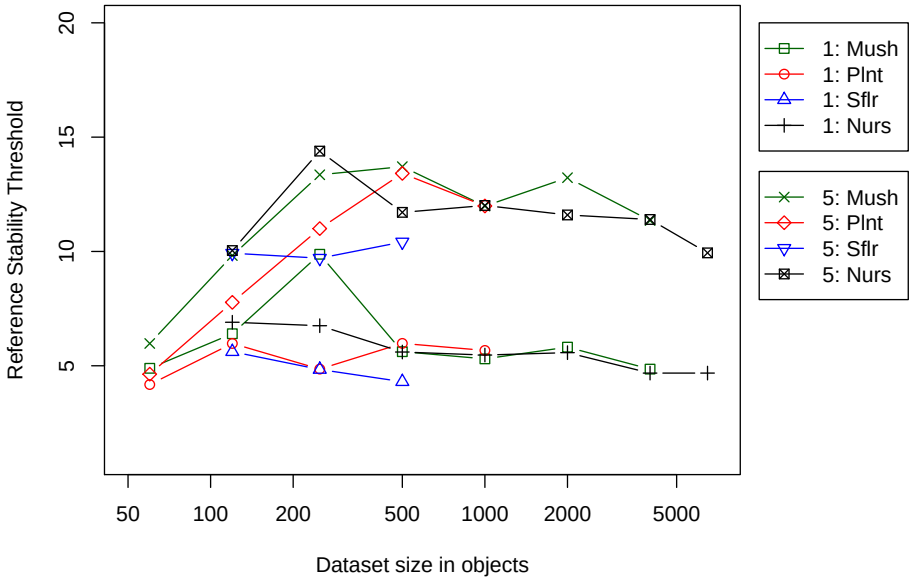


Fig. 4: Stability threshold in the reference dataset ensuring that 99% of concepts in the test dataset corresponding to stable concepts in the reference dataset are stable with stability thresholds 1 or 5.

is often used to evaluate statistical measures for pattern selection and can be found as a part of pattern selection algorithms with a good performance [18].

3.2 The General Behaviour of Stability

Sets of pairs S and LS can be drawn by matching every point $\langle X, Y \rangle$ to a point in a 2D-plot. The best case is $y = x$, i.e. stability for a concept in $\mathcal{L}_1$ is equal to stability of the corresponding concept in $\mathcal{L}_2$, meaning that stability is not dependant on the dataset. However, this is hardly the case in real-world experiments. All relevancy measures depend on the dataset, while any measure should be able to predict its value independently of the dataset. Figures 2a and 2b show the corresponding diagrams for the datasets `Mush120` and `Mush4000`.[6,7] These figures also highlight the fact that many concepts have stability close to 1, and that the larger is the dataset, the larger is the number of concepts with stability close to 1. It is in accordance with the work [16] where most of the concepts have the stability close to 1. However, when the logarithmic set LS is used, a blurred line $y = x$ can be perceived in Figures 2c and 2d. Moreover, selecting the concepts which are stable w.r.t. a high threshold, say θ_r, in the

[6] From here, the name of a dataset followed by a number such as '$NameN$' refers to an experiment based on the dataset $Name$ where $\mathbb{K}_1$ and $\mathbb{K}_2$ are of the size N.

[7] See `http://www.loria.fr/~abuzmako/stability-meaning/` for other diagrams.

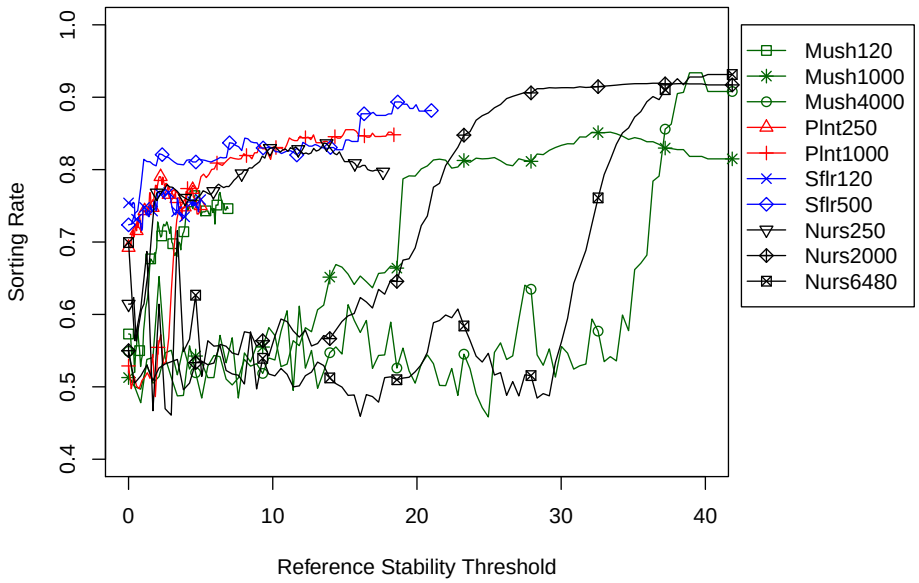


Fig. 5: Local sorting rate for different datasets. The rate is computed for the test dataset concepts corresponding to the first 1000 stable concepts in the reference dataset with stability above a given threshold.

reference dataset $\mathbb{K}_1$, the corresponding concepts in $\mathbb{K}_2$ are stable w.r.t. a lower threshold, say θ_t. Thus, we can conclude that stability is more tractable in the logarithmic scale, and then we only consider this logarithmic scale in the rest of the paper.

3.3 Setting a Stability Threshold

The dependency between two thresholds θ_r and θ_t of stability are shown in Figure 3. The x-axis corresponds to the stability threshold in the reference dataset $\mathbb{K}_1$, while the y-axis corresponds to the stability threshold in the test dataset $\mathbb{K}_2$. The lines correspond to the 99% level, i.e. given the stability in K_1, what should be the stability threshold in the test dataset $\mathbb{K}_2$ such that 99% of stable concepts in K_1 are also stable in K_2. In this figure one can see that lines begin to grow from 5 meaning that given stability threshold less than 5 in $\mathbb{K}_1$ no stability threshold in the test dataset $\mathbb{K}_2$ can ensure 99% of stable concepts. We can also see two types of lines. The lines with stairs correspond to the datasets with small number of stable concepts, while the others behave nearly the same. This behavior suggests that in order to ensure that a concept remains stable in another dataset with threshold $\theta_{\log}$, its stability in the reference dataset should be within $[\theta_{\log} + 5, \theta_{\log} + 10]$.

Let us consider the behavior of the stability thresholds w.r.t the size of the dataset. The dependency between the size of the dataset and the difference

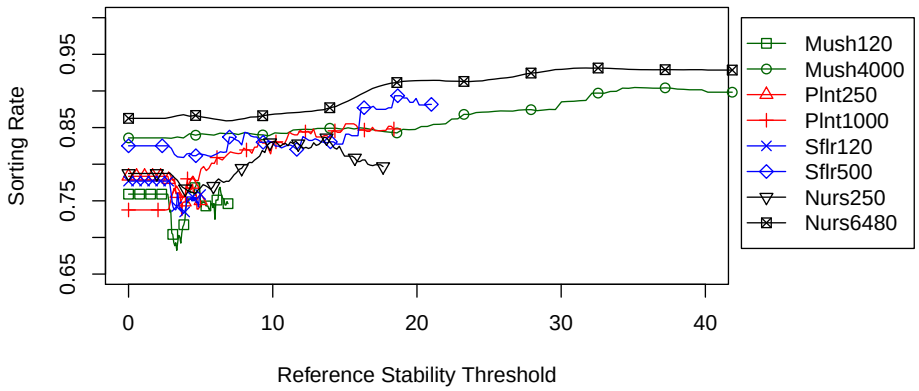


Fig. 6: Global sorting rate for different datasets.

between stability thresholds in the reference ($\mathbb{K}_1$) and in the test ($\mathbb{K}_2$) datasets is shown in Figure 4. The x-axis corresponds to the size of the dataset, the y-axis corresponds to the stability threshold in K_1 such that 99% of concepts selected by this threshold are stable in the test dataset $\mathbb{K}_2$ with a certain threshold (1 or 5). For example, the line '5: Mush' corresponds to the stability threshold θ ensuring that all concepts having stability more than θ in $\mathbb{K}_1$ correspond to concepts having stability at least 5 in the test dataset $\mathbb{K}_2$. We can see that for large datasets the stability threshold is independent of the dataset, while for small datasets the diversity is higher. Here for large datasets the stability threshold should be set to 5–6 in a reference dataset in order to ensure that 99% of stable concepts have corresponding concepts in another dataset. This threshold should be set to 12 in order to ensure that 99% of stable concepts correspond to concepts having stability at least 5 in another dataset.

3.4 Stability and Ranking

Stability can be used for ranking concepts by decreasing its value. Thus, it is useful to study the linear order corresponding to the ranking relation. A way to study an order of an array ar is to compute its sorting rate r, i.e. the relative number of pairs in the array sorted in the ascending order: $r = 2 \cdot \frac{\{(i,j) | i<j \text{ and } ar_i \leq ar_j\}}{|ar| \cdot (|ar|-1)}$. A sorting rate equal to 1 means that the array is in the ascending order, while 0 means that it is in the descending order; the value 0.5 means that there is no order at all. Figure 5 shows local sorting rate (LSR), i.e. given a threshold the first 1000 stable concepts in $\mathbb{K}_1$ are taken and the sorting rate for the array of stabilities of the corresponding concepts in $\mathbb{K}_2$ is computed. This plot shows that for large datasets, the LSR is high (around 0.8–0.9) only for high stability thresholds in $\mathbb{K}_1$. For the smaller datasets the local sorting rate is around 0.7–0.8 for all thresholds. It means that stability preserves LSR only for the most stable concepts where the difference in stability between concepts is high enough, i.e. an error in order is less likely.

Dataset	Size	Lattice	Stab.	FCbO	Freq.	Est. Method	Comb. Method	MC calls
Mush8124	$2.3 \cdot 10^5$	324	57	0.7	0	$2 \cdot 10^3$	$6 \cdot 10^3$	$6 \cdot 10^4$
Plnt1000	$2 \cdot 10^6$	45	10^4	78	0	181	446	$3 \cdot 10^3$
Chss100	$2 \cdot 10^6$	46	10^4	3.5	0	90	192	$2.3 \cdot 10^3$
SFlr1066	2988	0	0	0	0	0.7	11	284
Nurs12960	$1.2 \cdot 10^5$	245	5	0.2	0	425	$1.2 \cdot 10^3$	$4 \cdot 10^4$
Chss3196	$4.4 \cdot 10^6$	–	–	42	1000	$2 \cdot 10^4$	$3.5 \cdot 10^4$ (2%)	?
Plnt34781	$5.8 \cdot 10^6$	–	–	795	1750	$4.1 \cdot 10^5$	$4.6 \cdot 10^5$ (4.7%)	?

Table 3: Execution time for different steps on different datasets. `Size` is the number of concepts in the lattice; `Lattice` is the time for lattice computation with its structure; `Stab.` is the time for computing exact stability; `FCbO` is the time for computing the set of concepts by FCbO; `Freq.` is the frequency threshold applied for big datasets; `Est. Method` is the execution time for computing the estimate of stability by the estimate method; `Comb. Method` is the execution time for computing the estmate of stability be the combined method; the percentage here means that the program has been stopped after a certain amount of work; `MC calls` is the number of calls to the Monte-Carlo routine. All times are given in seconds.

Finally, Figure 6 shows the global sorting rate (GSR) for different datasets, i.e. the sorting rate of stabilities in $\mathbb{K}_2$ for all concepts corresponding to the concepts selected by a threshold in K_1. We can see that the GSR for all datasets is slowly increasing and for small thresholds it is higher than the LSR. It shows that stability gives a global ordering of concepts, while the local ordering is not reliable for small thresholds.

4 Computing an Estimate of Stability

In this section we study the efficiency of computing various estimates of stability. Table 3 shows computation times for different methods and datasets. The lattice structure is built by our implementation of AddIntent [19] and the set of concepts is computed by FCbO [20][8]. The datasets selected for experiments are the datasets of maximal tractable size (see Table 2) plus `Chess` and `Plants` with all the objects. For the last two datasets the numbers of concepts is huge. Such datasets can be analyzed by finding only frequent concepts, i.e. concepts with significantly large extents. Although an incomplete set of concepts without lattice structure cannot be processed by the algorithm from [11], stability can be estimated using formula (5), by Monte Carlo approach or their combination. For the cases where the estimation of stability takes too much time, the percentage of the processed concepts before termination is shown in the brackets. For the sake of efficiency, an estimation or an approximation of stability for a concept

[8] The implementation is taken from `http://icfca2012.markuskirchberg.net`.

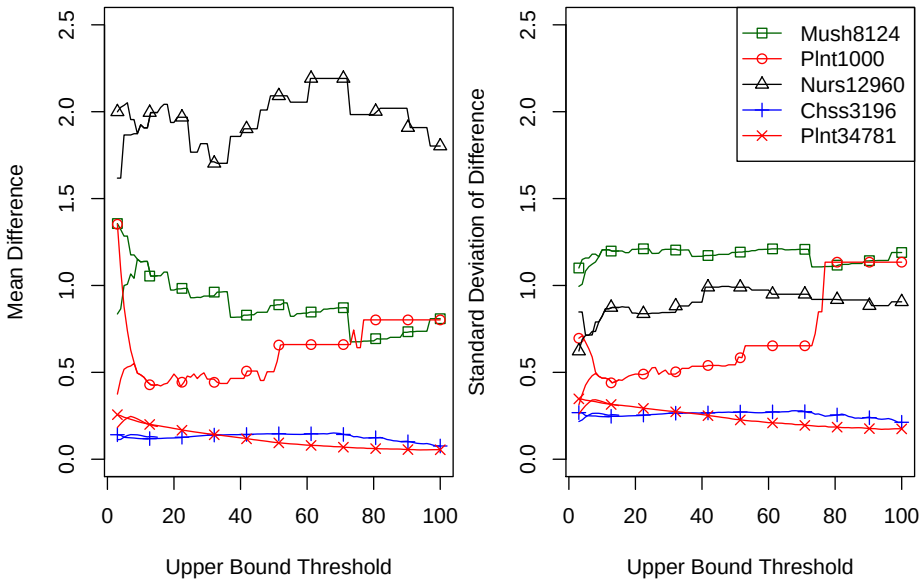


Fig. 7: The mean and the standard deviation of the stability estimate interval

is stopped whenever it is clear that the concept is unstable i.e. stability is less than 3 in the logarithmic scale.

We can see that even the combined method is significantly slower than the bounding method and, hence, there is no reason to only work with the Monte Carlo method as it is slower and does not provide a better precision. Moreover, although the number of calls to Monte Carlo routine is small in the combined method, the computational efficiency of the stability estimate can dramatically decrease, making the usage of combined method unfeasible. The estimates are more efficient in terms of computational time for large lattices, i.e. lattices with a high number of concepts for one object from the context. We can see that in some cases the estimates for small lattices take much more time than the estimates for large lattices. This can be explained by the fact that the corresponding contexts contain many objects and attributes and that the computational efficiency of the estimates is highly dependent on the size of the context.

The tightness of the estimates is shown in Figure 7. On the x-axis the values of the upper bound stability threshold are plotted while on the y-axis the mean difference in the estimate are plotted. The plots are split in area of $[0, 10]$; the bottom line corresponds to the improvement achieved by additional use of Monte Carlo in the combined method. According to formula (5) Monte Carlo can give any improvements only in the case where stability upper bound is less than 13 (taking into account that for these datasets there are less than 100 attributes, and Monte Carlo parameters are in accordance with Example 3). In practice, however, this bound is even smaller (less then 10). These plots show that generally mean and standard deviation of the estimate difference do not

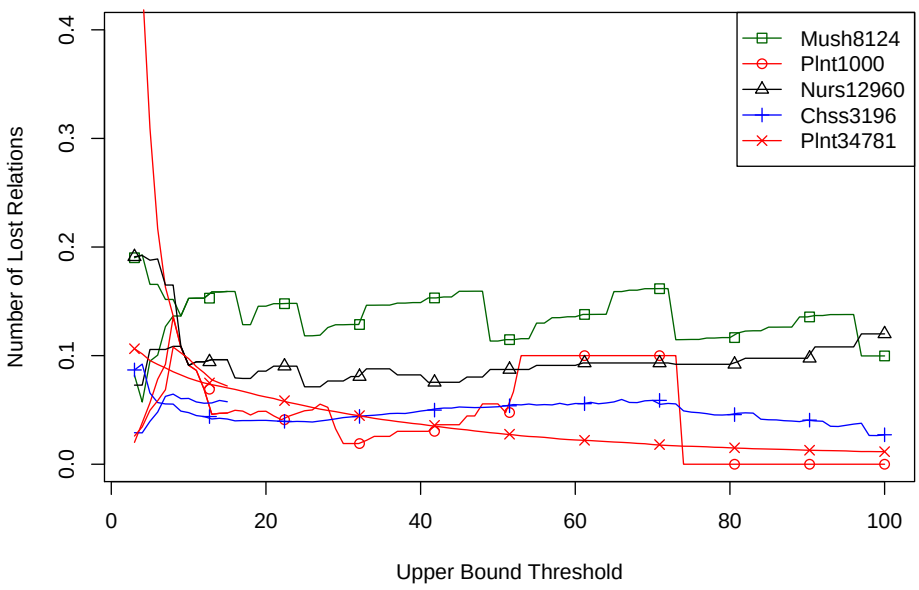


Fig. 8: Losing rate of relations for stability estimate

change w.r.t. the upper bound, however they can significantly depend on the dataset. In our experiments it appears that the well-structured dataset (`Mush`, `Nurs`) has higher mean value then the unstructured ones, while the big datasets with only frequent concepts have low mean-values and standard deviations.

If we want to rank concepts w.r.t. stability, how many pairs of concepts become incomparable when we use the estimates? Figure 8 shows the loss rate of the estimates, i.e. the relative number of concept pairs which cannot be compared by the estimate. Although the loss rate for the interval $[0, 10]$ can be high, it can be efficiently reduced by using the combined method.

5 Conclusion

In this paper we study concept stability and its estimates on different datasets. It is shown that stability computed in the logarithmic scale is more easy to interpret. Our experiments show that stability of a concept is correlated with the probability that the concept intent occurs in another dataset with high stability, i.e. it is an efficient measure for ranking patterns. However, independently of a dataset, as found experimentally, a concept should have a value of logarithmic stability greater than 5 in order to reflect any property of the population. Moreover, if the stability threshold in a reference dataset is θ, then the stability of the corresponding concept in another dataset is likely to be higher than $\theta - 10$ or even $\theta - 5$. We also remarked that stability is able to sort concepts in two independent datasets with nearly the same order by selecting concepts with stability greater than a certain threshold. However, the sorting rate of the first 1000

concepts from two independent datasets with stability above a certain threshold is high if the threshold is very high.

In the second part of this paper we showed that the introduced estimate is an efficient way for ranking concepts w.r.t. stability. It can be applied for an incomplete set of concepts and, hence, has more potential applications than the exact methods. The introduced approach can be meaningfully combined with a Monte Carlo method, providing better precision for weakly stable concepts by means of additional computational time. The precision and the sorting rate of the studied approximations are reasonably high and can be efficiently used for the stability computation.

There are many future research directions. One of them is to study other approaches for ranking formal concepts with a similar technique. An interesting question is to adapt the above approach to the comparison of different ranking methods. Next, the properties of stability suggest that interesting concepts can be found by resampling, i.e. analyzing many small parts of a large dataset, thus providing a key to an efficient processing of datasets with Formal Concept Analysis. Finally, the estimate we have proposed in this paper can be combined with an efficient realization, e.g., by means of parallel computation.

Acknowledgments: this research was supported by the Basic Research Program at the National Research University Higher School of Economics (Moscow, Russia) and by the BioIntelligence project (France).

References

1. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. 1st edn. Springer (1999)
2. Ganter, B., Kuznetsov, S.: Pattern Structures and Their Projections. In Delugach, H., Stumme, G., eds.: Conceptual Structures: Broadening the Base. Volume 2120 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2001) 129–142
3. Bělohlávek, R., Vychodil, V.: Formal Concept Analysis with Constraints by Closure Operators. In Schärfe, H., Hitzler, P., Ohrstrom, P., eds.: Conceptual Structures: Inspiration and Application. Volume 4068 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2006) 131–143
4. Belohlavek, R., Vychodil, V.: Formal Concept Analysis With Background Knowledge: Attribute Priorities. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **39**(4) (July 2009) 399–409
5. Dias, S.M., Vieira, N.J.: Applying the JBOS reduction method for relevant knowledge extraction. Expert Systems with Applications **40**(5) (April 2013) 1880–1887
6. Buzmakov, A., Egho, E., Jay, N., Kuznetsov, S.O., Napoli, A., Raïssi, C.: On Projections of Sequential Pattern Structures (with an application on care trajectories). In: Proc. 10th International Conference on Concept Lattices and Their Applications. (2013) 199–208
7. Belohlavek, R., Trnecka, M.: Basic Level in Formal Concept Analysis: Interesting Concepts and Psychological Ramifications. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. IJCAI'13, AAAI Press (August 2013) 1233–1239

8. Kuznetsov, S.O.: Stability as an Estimate of the Degree of Substantiation of Hypotheses on the Basis of Operational Similarity. Automatic Documentation and Mathematical Linguistics (Nauch. Tekh. Inf. Ser. 2) **24**(6) (1990) 62–75
9. Kuznetsov, S.O.: On stability of a formal concept. Annals of Mathematics and Artificial Intelligence **49**(1-4) (2007) 101–115
10. Kuznetsov, S., Obiedkov, S., Roth, C.: Reducing the Representation Complexity of Lattice-Based Taxonomies. In Priss, U., Polovina, S., Hill, R., eds.: Conceptual Structures: Knowledge Architectures for Smart Applications. Volume 4604 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2007) 241–254
11. Roth, C., Obiedkov, S., Kourie, D.G.: On succinct representation of knowledge community taxonomies with formal concept analysis A Formal Concept Analysis Approach in Applied Epistemology. International Journal of Foundations of Computer Science **19**(02) (April 2008) 383–404
12. Klimushkin, M., Obiedkov, S.A., Roth, C.: Approaches to the Selection of Relevant Concepts in the Case of Noisy Data. In: Proc. of the 8th International Conference on Formal Concept Analysis. ICFCA'10, Springer (2010) 255–266
13. Babin, M., Kuznetsov, S.: Approximating Concept Stability. In Domenach, F., Ignatov, D., Poelmans, J., eds.: Formal Concept Analysis. Volume 7278 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 7–15
14. Roth, C., Obiedkov, S., Kourie, D.: Towards concise representation for taxonomies of epistemic communities. In: Proceedings of the 4th international conference on Concept lattices and their applications. CLA'06, Berlin, Heidelberg, Springer-Verlag (2006) 240–255
15. Buzmakov, A., Egho, E., Jay, N., Kuznetsov, S.O., Napoli, A., Raïssi, C.: The representation of sequential patterns and their projections within Formal Concept Analysis. In: Workshop Notes for LML (PKDD). (2013) 65–79
16. Jay, N., Kohler, F., Napoli, A.: Analysis of Social Communities with Iceberg and Stability-Based Concept Lattices. In Medina, R., Obiedkov, S., eds.: Formal Concept Analysis. Volume 4933 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2008) 258–272
17. Frank, A., Asuncion, A.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. University of California, Irvine, School of Information and Computer Sciences (2010)
18. Webb, G.I.: Discovering Significant Patterns. Machine Learning **68**(1) (2007) 1–33
19. Merwe, D.V.D., Obiedkov, S., Kourie, D.: AddIntent: A new incremental algorithm for constructing concept lattices. In Goos, G., Hartmanis, J., Leeuwen, J., Eklund, P., eds.: Concept Lattices. Volume 2961. Springer (2004) 372–385
20. Krajca, P., Outrata, J., Vychodil, V.: Advances in Algorithms Based on CbO. In: Proc. of the 8th International Conference on Concept Lattices and Their Applications (CLA'10). (2010) 325–337