Is Concept Stability a Measure for Pattern Selection?

Aleksey Buzmakov, Sergei O. Kuznetsov, Amedeo Napoli

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 00 (2014) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

Information Technology and Quantitative Management , ITQM 2014

Is concept stability a measure for pattern selection?

Aleksey Buzmakov[a,b,*], Sergei O. Kuznetsov[a], Amedeo Napoli[b]

[a]*National Research University Higher School of Economics, 20, Myasnitskay street, 101000, Moscow, Russia*
[b]*LORIA (CNRS – Inria NGE – U. de Lorraine), 615, Jardin Botanique street, 54600, Vandœuvre-lès-Nancy, France*

Abstract

There is a lot of usefulness measures of patterns in data mining. This paper is focused on the measures used in Formal Concept Analysis (FCA). In particular, concept stability is a popular relevancy measure in FCA. Experimental results of this paper show that high stability of a pattern in a given dataset derived from the general population suggests that the stability of that pattern is high in another dataset derived from the same population. At the second part of the paper, a new estimate of stability is introduced and studied. It es performance is evaluated experimentally. And it is shown that it is more efficient.

Keywords:
formal concept analysis; stability; pattern selection; pattern discovery; experiment

1. Introduction

In data mining, many usefulness measures of patterns are introduced. For example, more than 30 statistical methods are enumerated and discussed in [1]. Such a high number of different approaches to pattern selection emphasize the importance of the problem. In this paper we would like to focus on a measure which is introduced within Formal Concept Analysis (FCA). FCA is a mathematical formalism having many applications in data analysis [2]. Starting from the set of objects and the corresponding sets of attributes FCA tends to generalize the descriptions for any set of objects. Although this approach is less efficient than the statistical methods it is still feasible and ensures that no potentially interesting pattern is missed.

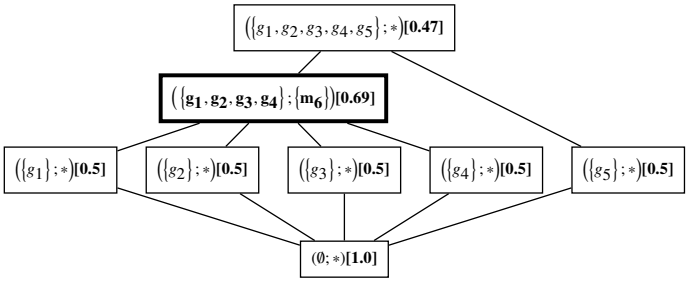
Within FCA there are several approaches for pattern selection. Two disjoint kinds of approaches can be distinguished. The first one is to introduce background knowledge into the procedure computing concepts [3, 4, 5, 6, 7]. These approaches allows one to find patterns which are likely to be useful for the current task. Although the number of resulting patterns can be significantly reduced, they are still numerous. The second kind of approaches can be applied in a composition with the first ones, ranking the resulting patterns w.r.t. a relevance measure.

The authors of [8] provide several measures for ranking concepts that stem from the algorithms possibly underlying human behavior. Stability is another measure for ranking concepts, introduced in [9] and later revised in [10, 11, 12]. Several other methods are considered in [13], where it is shown that stability is more reliable in noisy data. For the moment, stability seems to be the most widely used usefulness measure around the FCA community. Thus, in this paper we are going to focus on stability. Although this measure is often used, there is

*Corresponding author. Tel.: +7-926-712-19420 .
E-mail address: aleksey.buzmakov@inria.fr.

	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	

(a)



(b)

Fig. 1: (a) A toy formal context. (b) Concept Lattice with corresponding stability indexes.

neither a reliable comparison nor a deep research on its usefulness. Consequently, the first goal of this paper is to evaluate the usefulness of stability, i.e. computing stability for a pattern, is it coherent with the stability computing for the same pattern but w.r.t. a different dataset coming from the same population (the similarly distributed dataset). It should be noticed that the comparison of different approaches is difficult mainly because it requires a wide set of experts that could manually evaluate the approaches. Thus, many of the introduced measures are, first, proved to be statistically sound (which we are going to show experimentally) and, second, evaluated w.r.t. a formal mathematical condition that could quite far from the real meaning of the pattern.

The second goal of this paper is to deal with the computational complexity of stability. It is shown that computation of stability is #P-complete [9, 10]. In order to compute it for large concept lattices, one needs to use estimates and approximations. Correspondingly, in the second part of our paper we introduce an estimate of stability and empirically evaluate its performance w.r.t. known approximations.

The rest of the paper is organised as follows. Section 2 introduces definition of stability and discusses known stability estimates. In Section 3 experiments on relevancy of stability are discussed. In Sections 4 the experiments validating the introduced estimates are explained.

2. Stability of a formal concept

2.1. Formal concept analysis (FCA)

FCA [2] is a formalism for data analysis. FCA starts with a formal context and builds a set of formal concepts organized within a concept lattice. A formal context is a triple (G, M, I), where G is a set of objects, M is a set of attributes and I is a relation between G and M, $I \subseteq G \times M$. In Figure 1a, a formal context is shown. A Galois connection between G and M is defined as follows:

$$A' = \{m \in M \mid \forall g \in A, (g, m) \in I\}, \qquad A \subseteq G$$
$$B' = \{g \in G \mid \forall m \in B, (g, m) \in I\}, \qquad B \subseteq M$$

The Galois connection maps a set of objects to the maximal set of attributes shared by all objects and reciprocally. For example, $\{g_1, g_2\}' = \{m_6\}$, while $\{m_6\}' = \{g_1, g_2, g_3, g_4\}$.

Definition 1. *A formal concept is a pair (A, B), where A is a subset of objects, B is a subset of attributes, such that $A' = B$ and $A = B'$, where A is called the extent of the concept, and B is called the intent of the concept.*

For example a pair $(\{g_1, g_2, g_3, g_4\}; \{m_6\})$ is a formal concept. Formal concepts can be partially ordered w.r.t. the extent inclusion (dually, intent inclusion). For example, $(\{g_3\}; \{m_3, m_6\}) \leq (\{g_1, g_2, g_3, g_4\}; \{m_6\})$. This partial order of concepts is shown in Figure 1b.

2.2. The definition of stability

Stability is an interestingness measure of a formal concept introduced in [9] and later revised in [10, 12].

Definition 2. *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 1b shows a lattice for the context in Figure 1a, 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, 12].

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.

The second goal of the paper is related to the high complexity of the stability. In fact, given a context and a concept, the computation of concept stability is #P-complete [9, 10]. One of the fastest algorithm for processing concept stability using a concept lattice L is proposed in [12], 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.

2.3. 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. $O(n \cdot m^2)$. Here n and m are the cardinalities of G and M correspondingly.

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

```
Function FindStabilityLimits
    Data: A context 𝕂 = (G, M, I), A concept C.
    Result: < Left, Right >, a pair of left and right limits for the stability.
    Left ← 1;
    Right ← 1;
    children ← FindChildren(𝕂, C) ;                          /* O(|N| · |M|^2 */
    minDiffSize ← ∞;
    foreach ch ∈ children do                                 /* O(|M|) iterations at most */
        diffSize ← |Ext(C) \ Ext(ch)|;
        minDiffSize ← min(minDiffSize, diffSize);
        Left ← Left − 2^−diffSize;
    Right ← 1 − 2^−minDiffSize;
    return < Left, Right >;
```

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

The upper bound of the equation can be found in [12], 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 [17] 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.*

Table 1: 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 correpsopnidng 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.

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

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 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. In this paper it is referred as the *combined method*.

Recall that there are three other estimates of stability [9, 10, 12] 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 sections we present experiments on general behaviour of stability and efficiency of the introduced estimates.

3. Experiment on relevancy of stability

Experiments about the meaning and the estimation of stability are carried out on public datasets available from the UCI repository [18]. These datasets are shown in Table 1. With their different size and complexity, these datasets provide a reach 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 as for only 100 objects in the context there are already $2 \cdot 10^6$ of concepts in the concept lattice.

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 $N = 4$.
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 $\mathbb{K}_2$.

[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`

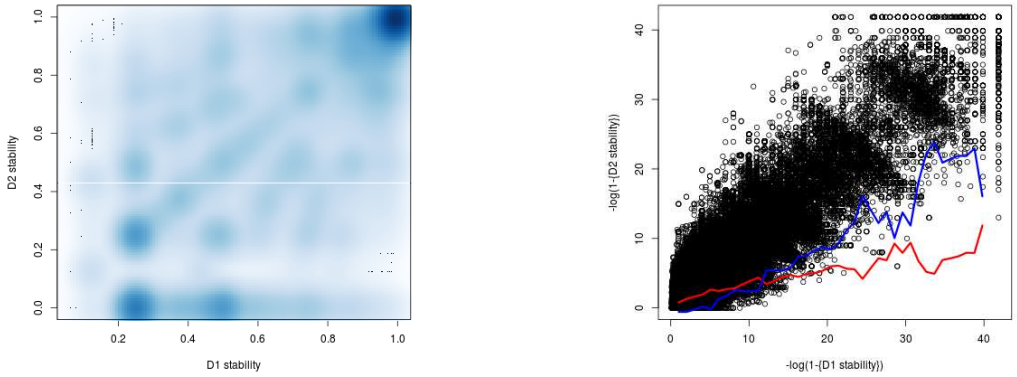


Fig. 2: Stability in the test dataset w.r.t the reference one in Mush4000 in (a) plane scale (b) logarithmic scale.

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 $LS = \{\left\langle X_{\log}, Y_{\log}\right\rangle, \dots\}$ includes the stability pairs in S in logarithmic scale as stated in formula (4). Then sets S and LS are studied.

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 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 [19].

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 $y = x$. It means that stability computed for dataset part $\mathbb{K}_1$ is exactly the same as stability computed for the dataset part $\mathbb{K}_2$. However, this is hardly the case in real-world experiments. For example, Figure 2a shows the corresponding diagram for the dataset `Mush4000`.[6,7]This figure also highlights the fact that many concepts have stability close to 1. It is in accordance with the work [16] where the authors find the same behaviour on their dataset. However, when the logarithmic set LS is used, a blurred line $y = x$ can be perceived in Figure 2b. Moreover, selecting the concepts which are stable w.r.t. a high threshold in the reference dataset $\mathbb{K}_1$, the corresponding concepts in $\mathbb{K}_2$ are stable w.r.t. a lower threshold. Thus, we can conclude that stability is more tractable in the logarithmic scale, and, thus, we only consider this logarithmic scale in the rest of the paper.

3.1. Setting a stability threshold

In the previous subsection it is mentioned that concepts stable in the reference dataset are stable in the test dataset with a smaller thresholds. But what is it "smaller"? Imagine that in the reference dataset $\mathbb{K}_1$ we have the threshold θ_1, i.e. if $Stab(c) \geq \theta_1$ then c is stable, while in the $\mathbb{K}_2$ we have θ_2. Then, we want to know the threshold θ_1 such that at least 99% of stable concepts in $\mathbb{K}_1$ corresponds to stable concepts in $\mathbb{K}_2$. Figure 3 shows the reference thresholds θ_1 (x-axis) w.r.t. the size of the datasets (y-axis) for $\theta_2 = 1$ and $\theta_2 = 5$. For example, the line '5: Mush' corresponds to the line of θ_1, where θ_2 is fixed to 5 w.r.t. to the size of the dataset built from dataset Mushrooms. The value $\theta_2 = 1$ means that any stable concept is just found in the test dataset, while $\theta_2 = 5$ requires that they are quite stable in the test dataset. We can see that for large datasets the stability threshold is independent of the dataset, while for small datasets the diversity is higher. We can see that the value of θ_1 should be set to 5–6 in order to ensure that 99% of stable concepts have corresponding concepts in another dataset.

[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.

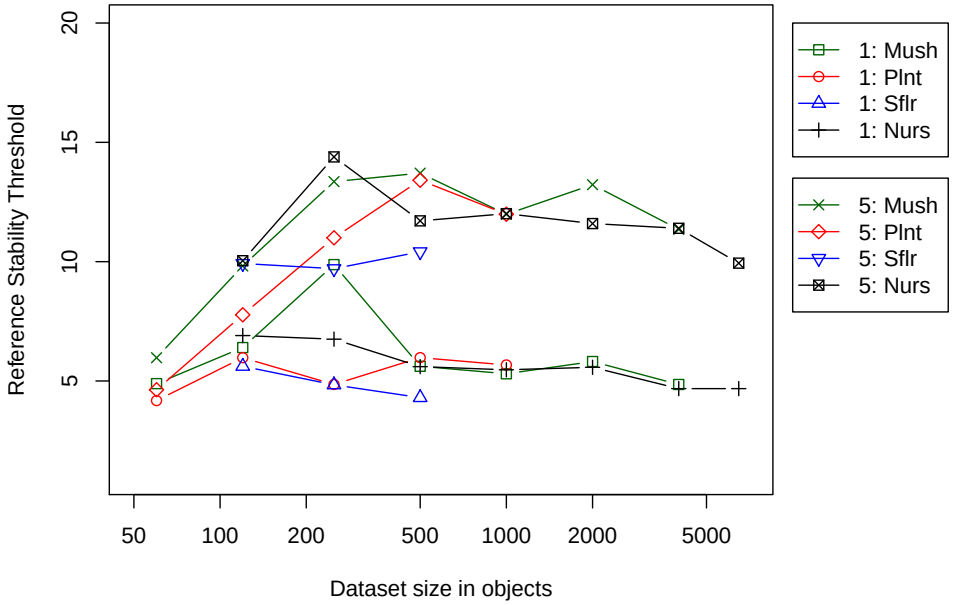


Fig. 3: Stability threshold in the reference dataset ensuring that 99% of concepts in the test datasets corresponding to stable concepts are stable with stability thresholds 1 or 5.

3.2. Stability and ranking

Another way of using usefulness measures is pattern raking. Thus, it is an interesting question if the oder of patterns could be preserved by using stability. 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 4 shows the sorting rate (SR) for different datasets, i.e. the sorting rate of concept stabilities in $\mathbb{K}_2$, ordered w.r.t. stabilities of the correpsponding stable concepts in K_1. We can see that SR for all datasets is slowly increasing preserving nearly the same value along the stability threshold in $\mathbb{K}_1$. And, thus, concept stability can be used to rank concepts.

4. Computing an estimate of stability

In this section we study the efficiency of computing various estimates of stability. Table 2 shows computation times for different methods and datasets. The lattice structure is built by our implementation of AddIntent [20] and the set of concepts is computed by FCbO [21][8]. The datasets selected for experiments are the datasets of maximal tractable size (see Table 1) 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 [12], 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 is stopped whenever it is clear that the concept is unstable i.e. stability is less than 3.

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. The estimates are more efficient in terms of computational time for large lattices, i.e. lattices with a high number of

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

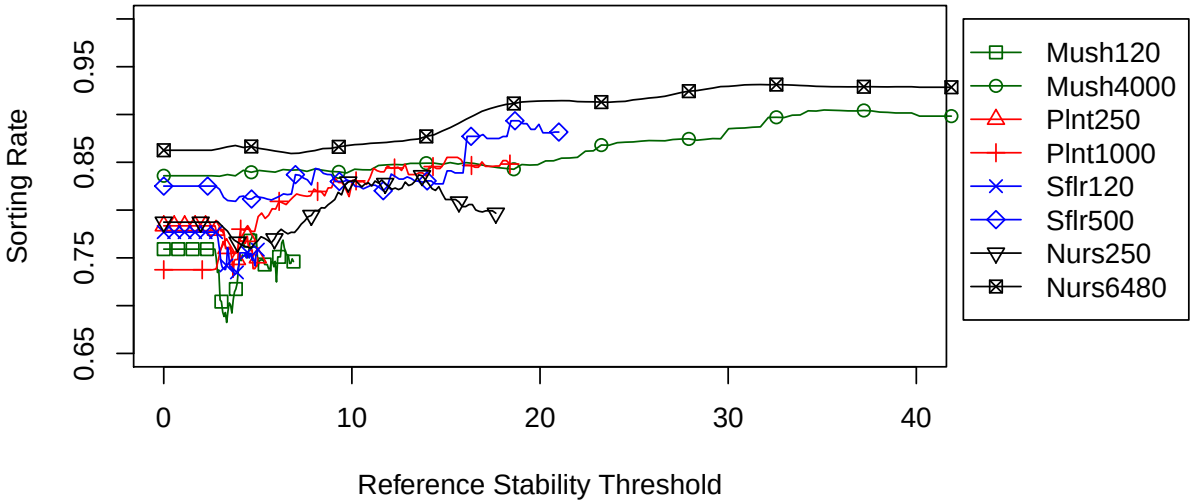


Fig. 4: Global sorting rate for different datasets.

Table 2: 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.

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%)	?

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.

Taking into account (5), we can try to find stable concepts w.r.t. to one of the bounds. If we use upper bound than we never lose stable concepts, while we can mark some unstable concepts as stable. Oppositly, if we find stable concepts by the lower bound, we lose some stable concepts, while everything found is really stable. Figure 5 shows frequencies of false stable and false unstable discoveries. Here we can see that with the upper bound we can found up to 40% of additional concepts which are unstable. However the number of false stable discoveries can vary quite a lot along the stability threshold. While with lower bound most of unstable concepts are really unstable, i.e. we can lose normally only a few of stable concepts.

But having a stability bounds how well can we order the patterns w.r.t. stability? Figure 6 shows the losing rate of the estimates, i.e. the relative number of concept pairs which cannot be compared by the estimate. Normally, we lose less then 20% of concept relations independently from the threshold. In the interval $[0 - 10]$ for the threshold we can find that the losing rate can be high. However, in this interval the Monte-Carlo approach can be applied, and, thus, can significantly reduce the losing rate.

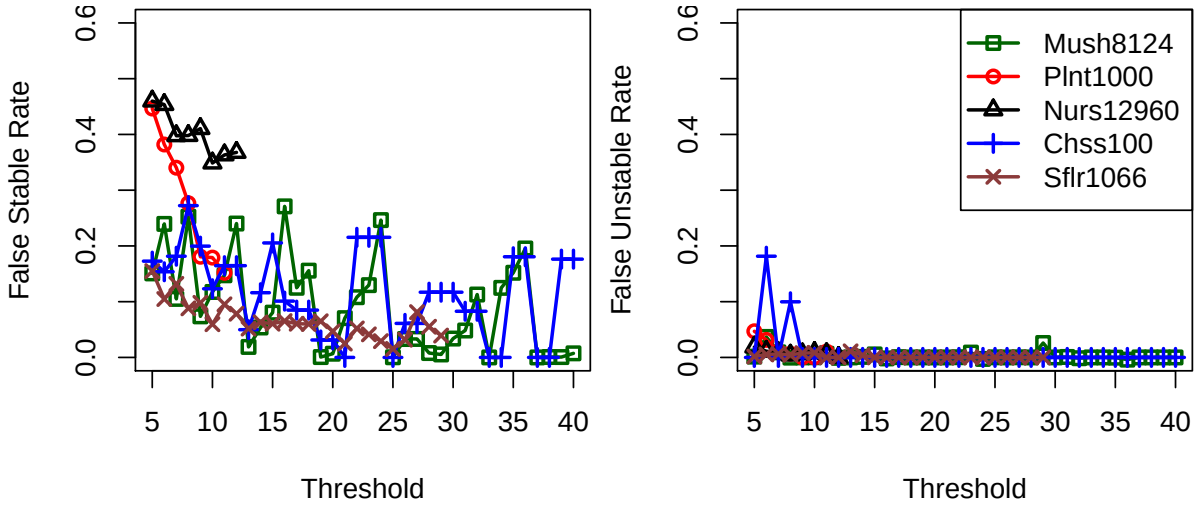


Fig. 5: Over- and under- estimation rate for selecting stable concepts w.r.t. upper and lower bound of stability.

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 convenient. Given a threshold of stability, patterns that stability are above the threshold in a given dataset are likely to have stability above a smaller threshold in another dataset coming from the same distribution. However, independently of a dataset, as found experimentally, a concept should have logarithmic stability more than 5 in order to reflect any property of the population. We also show that stability is able to sort concepts in two independent datasets with nearly the same order by selecting concepts with stability above a certain threshold.

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.

There are many future research directions. The found 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. The second important direction is to compare stability and other known measures. Finally, the proposed approximation approach can be efficiently realized, e.g. in parallel computation.

Acknowledgements

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] A. Masood, S. Soong, Measuring Interestingness – Perspectives on Anomaly Detection, Computer Engineering and Intelligent Systems 4 (1) (2013) 29–40.
[2] B. Ganter, R. Wille, Formal Concept Analysis: Mathematical Foundations, 1st Edition, Springer, 1999.
[3] B. Ganter, S. Kuznetsov, Pattern Structures and Their Projections, in: H. Delugach, G. Stumme (Eds.), Conceptual Structures: Broadening the Base, Vol. 2120 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2001, pp. 129–142.
[4] R. Bělohlávek, V. Vychodil, Formal Concept Analysis with Constraints by Closure Operators, in: H. Schärfe, P. Hitzler, P. Ohrstrom (Eds.), Conceptual Structures: Inspiration and Application, Vol. 4068 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, pp. 131–143.
[5] R. Belohlavek, V. Vychodil, Formal Concept Analysis With Background Knowledge: Attribute Priorities, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 39 (4) (2009) 399–409.

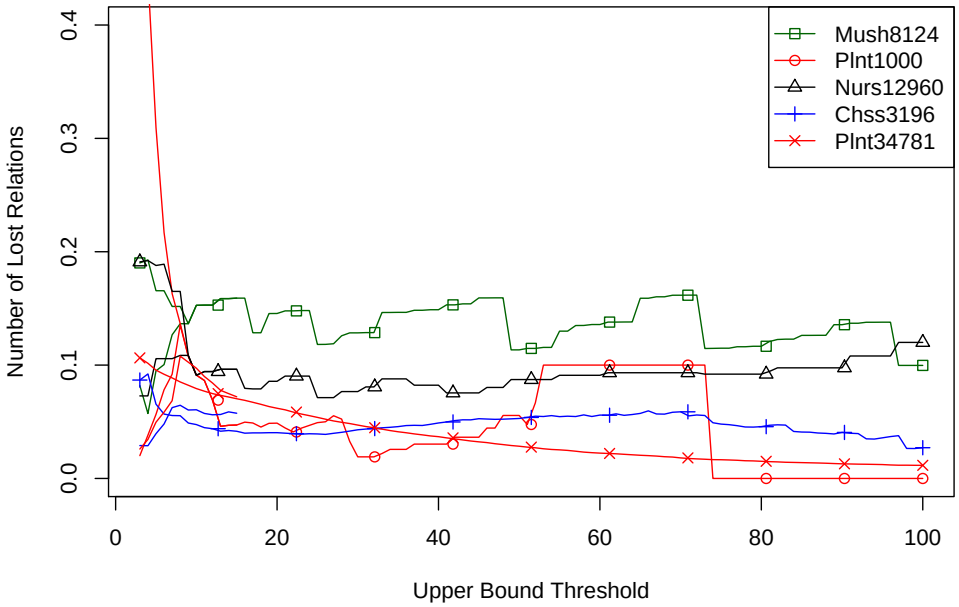


Fig. 6: Losing rate of relations for stability estimate

[6] S. M. Dias, N. J. Vieira, Applying the JBOS reduction method for relevant knowledge extraction, Expert Systems with Applications 40 (5) (2013) 1880–1887.
[7] A. Buzmakov, E. Egho, N. Jay, S. O. Kuznetsov, A. Napoli, C. Raïssi, On Projections of Sequential Pattern Structures (with an application on care trajectories), in: Proc. 10th International Conference on Concept Lattices and Their Applications, 2013, pp. 199–208.
[8] R. Belohlavek, M. Trnecka, 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, 2013, pp. 1233–1239.
[9] S. O. Kuznetsov, 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.
[10] S. O. Kuznetsov, On stability of a formal concept, Annals of Mathematics and Artificial Intelligence 49 (1-4) (2007) 101–115.
[11] S. Kuznetsov, S. Obiedkov, C. Roth, Reducing the Representation Complexity of Lattice-Based Taxonomies, in: U. Priss, S. Polovina, R. Hill (Eds.), Conceptual Structures: Knowledge Architectures for Smart Applications, Vol. 4604 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 241–254.
[12] C. Roth, S. Obiedkov, D. G. Kourie, 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) (2008) 383–404.
[13] M. Klimushkin, S. A. Obiedkov, C. Roth, 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, pp. 255–266.
[14] C. Roth, S. Obiedkov, D. Kourie, Towards concise representation for taxonomies of epistemic communities, in: Proceedings of the 4th international conference on Concept lattices and their applications, CLA'06, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 240–255.
[15] A. Buzmakov, E. Egho, N. Jay, S. O. Kuznetsov, A. Napoli, C. Raïssi, The representation of sequential patterns and their projections within Formal Concept Analysis, in: Workshop Notes for LML (PKDD), 2013, pp. 65–79.
[16] N. Jay, F. Kohler, A. Napoli, Analysis of Social Communities with Iceberg and Stability-Based Concept Lattices, in: R. Medina, S. Obiedkov (Eds.), Formal Concept Analysis, Vol. 4933 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2008, pp. 258–272.
[17] M. Babin, S. Kuznetsov, Approximating Concept Stability, in: F. Domenach, D. Ignatov, J. Poelmans (Eds.), Formal Concept Analysis, Vol. 7278 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 7–15.
[18] A. Frank, A. Asuncion, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml], University of California, Irvine, School of Information and Computer Sciences, 2010.
[19] G. I. Webb, Discovering Significant Patterns, Machine Learning 68 (1) (2007) 1–33.
[20] D. V. D. Merwe, S. Obiedkov, D. Kourie, AddIntent: A new incremental algorithm for constructing concept lattices, in: G. Goos, J. Hartmanis, J. Leeuwen, P. Eklund (Eds.), Concept Lattices, Vol. 2961, Springer, 2004, pp. 372–385.
[21] P. Krajca, J. Outrata, V. Vychodil, Advances in Algorithms Based on CbO., in: Proc. of the 8th International Conference on Concept Lattices and Their Applications (CLA'10)., 2010, pp. 325–337.