
Miskolc Mathematical Notes
Vol. 19 (2018), No. 2, pp. 983–996

HU e-ISSN 1787-2413
DOI: 10.18514/MMN.2018.2529

EFFICIENT APPROXIMATION FOR COUNTING OF FORMAL CONCEPTS GENERATED FROM FORMAL CONTEXT

L. KOVÁCS

Received 15 February, 2018

Abstract. The number of formal concepts generated from the input context is an important parameter in the cost functions of concept formation algorithms. The calculation of concept count for any arbitrary context is a hard, NP-complete problem and only rough approximation methods can be found in the literature to solve this problem. This paper introduces an efficient numerical approximation algorithm for contexts where attribute probabilities are independent from the objects instances. The preconditions required by the approximation method are usually met in the FCA applications, thus the proposed method provides an efficient tool for practical complexity analysis, too.

2010 *Mathematics Subject Classification:* 06B04; 41A04

Keywords: formal concept analysis, numerical approximation, optimization

1. INTRODUCTION

In the development of Formal Concept Analysis (FCA) applications, a key issue is the cost efficiency of concept set and concept lattice management. The cost function [12] [14] usually contains the following key parameters:

- N : number of objects
- M: number of attributes
- L: the average number of attributes related to an arbitrary object (context density)
- C: total number of concepts generated.

Although the value C is a function of the context, parameter C is considered as an independent base parameter. The reason of this simplification is that the relationship to calculate C is too complex, no simple analytical description is known.

This paper introduces a probabilistic model and a related efficient numerical approximation method to determine the count of generated concepts. The probabilistic model is based on the model presented in [9], where the goal of the work was to determine the significance of the generated formal concepts. Unlike the original paper, our model is aimed at an efficient approximation of the total number of concepts.

The proposed method can be used among others in complexity analysis of FCA algorithms in many application areas.

The FCA provides tools to manage and to investigate the concept set generated from an input formal context. A formal context is defined as a triplet $< \mathcal{G}, \mathcal{M}, \mathcal{I} >$, where $\mathcal{I}$ is a binary relation between $\mathcal{G}$ (set of objects) and $\mathcal{M}$ (set of attributes). The property $(g,m) \in \mathcal{I}$ is met if and only if the attribute m is true for the object g. Two derivation operators are introduced as mappings between the powersets of $\mathcal{G}$ and $\mathcal{M}$. For $A \subseteq \mathcal{G}, B \subseteq \mathcal{M}$:

$$f(A) = A^I = \{m \in \mathcal{M} | \forall g \in A : (g,m) \in I)\},$$

$$g(B) = B^I = \{g \in \mathcal{G} | \forall m \in B : (g,m) \in I)\}.$$

For a context $< \mathcal{G}, \mathcal{M}, \mathcal{I} >$, a formal concept is defined as a pair (A,B), where $A \subseteq \mathcal{G}$, $B \subseteq \mathcal{M}$, $A = B^I$, $B = A^I$ are met. The composition of these derivations are closure operators, $A \mapsto A^{II}, A \subseteq \mathcal{G}$ and respectively $B \mapsto B^{II}, B \subseteq \mathcal{M}$. Regarding the derivation operator, the components of a formal concept satisfy the $A = A^{II}$, $B = B^{II}$ conditions, too. The A component is called the extent of the concept, while B is the intent part.

On the set of formal concepts $\mathcal{C}$ generated from the context $< \mathcal{G}, \mathcal{M}, \mathcal{I} >$, a partial ordering relation is defined in the following way:

$$(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2.$$

It can be shown that $A_1 \subseteq A_2$ if and only if $B_2 \subseteq B_1$. The obtained partially ordered set $(\mathcal{C}, \leq)$ is in fact a complete lattice, called the concept lattice of the context $< \mathcal{G}, \mathcal{M}, \mathcal{I} >$.

The size of $\mathcal{C}$ is a key factor in the cost analysis of FCA algorithms. Due to the complex relationship between the size of $\mathcal{C}$ and the context parameters, there is no simple and efficient approximation to determine the total number of concepts. The first related result was presented by Ganter and Wille in [7] showing that the size of $\mathcal{C}$ may increase exponentially in the parameters N and M. Beside the parameters N and M, also the density of the context plays an important role in the complexity analysis. A context with large values of N and M, but having a sparse $\mathcal{I}$, yields a small concept set. One of the first analytical results in counting the concepts can be found in [16] presented by Schutt. The paper provides the following upper approximation for the count value:

$$C \leq 3/2 \cdot 2^{\sqrt{|I|+1}} - 1.$$

Later, Kuznetsov has been proved in [10] that calculation algorithm for the total number of concepts belongs to the NP-complete problem class.

A more sharper theoretical upper bound was shown by Prisner in [15] and by Albano and Chornomaz in [1]. They have investigated a special type of contexts, the contranomial scale free contexts. For a given set $\mathcal{S}$, the context $< \mathcal{S}, \mathcal{S}, \neq >$ is a contranomial scale context. If the set $\mathcal{S}$ contains k elements then the context belongs

to the class $N^c(k)$. For any $N^c(k)$-free context, the upper bound for the concept number is given with

$$C \leq (|G||M|)^{k-1} + 1.$$

According to the literature, there are only few proposals to provide a precise approximation method. One important result is presented in [4], where a sampling approach was used to estimate the concept count. The sampling method traverses the concept lattice by random walk and it works with a series of increasing sub-contexts. The candidate concepts are checked whether they are contained in other subcontexts already tested or not. The main drawback of the proposed algorithm is the high calculation cost as the number of candidate concepts for sampling is very high.

Due to these efficiency problems, our method uses a different approach. We take a simplified probability model where the attribute occurence probabilities are independent from the objects. This independence model was used also in [9] and [6], where the goal was to calculate the relevance of the discovered concepts.

The applied data matrix model is based on a fixed matrix marginal approach presented also in [8]. In the approach, the sums of the columns (the probabilities of the attributes) are fixed. The article presents also a novel algorithm calculating the concept probability index, but the cost of the proposed algorithm is too high for large practical data contexts. The concept probability index can be used also in the fuzzy FCA models [5] to provide an uncertainty level for knowledge engineering.

An interesting generalization of the probability model can be found in [3] to determine the basic level of concepts generated by FCA. Basic level concepts are those concepts which are used to refer to objects of our everyday life. The basic level can be seen as a compromise between the accuracy of classification at a maximally general level and the predictive power of a maximally specific level [13]. Using the example given in [3], when we refer to a particular dog then we usually say 'It is a dog.' rather than 'This is a German Shepherd.' or 'This is a mammal.'. The elements of the basic level concept sets are characterized by the fact that they have significantly larger cohesion than its upper neighbors and they have only a slightly smaller cohesion than its lower neighbors. A similar research was presented also in [11] to calculate concept interestingness using the concept probability, concept stability and concept robustness as the main components of the interestingness measure.

2. CONCEPT PROBABILITY MODEL

A basic assumption in our model is that the input context is generated randomly. The probability that object i is linked to attribute j is denoted by

$$p_{ij}.$$

We assume that

$$\forall j, i_1, i_2 : p_{i_1,j} = p_{i_2,j}.$$

The elements of the context matrix are either 1 (the attribute is true) or 0. A concept corresponds to a special sub-matrix of the context matrix. An example is shown in Figure 1, where K denotes the context matrix, A is the concept subcontext, while B, C, D, E are the complementary sub-contexts of A. Although the concept subcontext is a single rectangle in the example, the subcontexts are usually fragmented.

A sub-context A corresponds to a formal concept if and only if:

- all elements in A are set to 1
- for each column in sub-contexts B, D, one of the elements is equal to 0
- for each row in sub-contexts C, E, one of the elements is equal to 0.

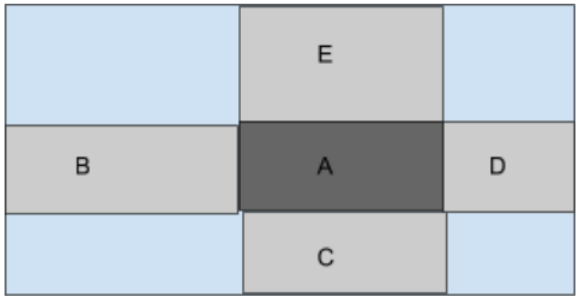

FIGURE 1. Example context and sub-contexts

We introduce a random variable ξ_A for every candidate sub-contexts A, where the value is set to 1 if the A belongs to a concept in the current experiment. Otherwise, the value is equal to 0. The mean value of ξ_A shows the probability that A belongs to a concept. Next, we take a new random variable ξ which is equal to the sum of the candidate level random variables. The total number of concepts in the context is estimated with the expected value of ξ.

The calculation of the mean value for ξ_A is based on the following considerations. First, we know that all matrix elements in A must be equal to 1 an the corresponding probability is equal to

$$\prod_{(i,j)\in A} p_{i,j}.$$

The probability that every column in the region $B \cup D$ contains at least one element with value 0:

$$\prod_{o\in B\cup D} (1 - \prod_{(i,j)\in o} p_{i,j}),$$

where o denotes a column. The corresponding probability that every row in the region $C \cup E$ contains at least one element with value 0 can be given with

$$\prod_{s\in C\cup E} (1 - \prod_{(i,j)\in s} p_{i,j}).$$

where s is a row element. Based on these considerations, the mean value of ξ, i.e. the sum over the set of all possible candidate sub-contexts is equal to:

$$C = \sum_{A \subset K} \prod_{(i,j) \in A} p_{i,j} \prod_{o \in B \cup D} (1 - \prod_{(i,j) \in o} p_{i,j}) \prod_{s \in C \cup E} (1 - \prod_{(i,j) \in s} p_{i,j}). \tag{2.1}$$

This expression shows our base formula for the approximation of concept count.

2.1. *Uniform distribution*

In this case, we assume that the probability for every object-attribute pair is the same:

$$\forall i, j : p_{i,j} = p.$$

The corresponding formula for the approximation concept count can be transformed into the following simple formula:

$$C = \sum_{n=1}^{N} \sum_{m=1}^{M} \binom{N}{n} \binom{M}{m} p^{n \cdot m} (1 - p^n)^{M-m} (1 - p^m)^{N-n}. \tag{2.2}$$

This formula can be implemented with the following R code:

```
cptcnt = function(N,M,P) {
  val = 0;
  for (Nx in 1:N) {
    for (My in 1:M) {
      val = val + getval(N,M,Nx,My,P)
    }
  }
  return (val);
}

getval = function (N, M, Nx, My,P) {
  c = choose(N , Nx)*choose(M , My)*(P^(Nx*My))*
    (1 - P^Nx)^(M - My)*(1 - P^My)^(N - Nx);
  return (c);
}
```

To demonstrate the accuracy of the presented model, Figure 2 shows the result of a comparison test, where the theoretical calculation is compared with the experimental measurement. The figure shows the dependency of the calculated and measured concept counts (C) in dependency form attribute probability (P). For the eperimental measurement, we used our Java implementation of the InClose [2] algorithm. The test is based on random generation of the context using the following parameter settings: N=15, M=10; P=0.1..1.0 and the length of the runs to calculate the mean

value is 5. The dashed line corresponds to the measured values. From the viewpoint of the practice, the result shows a good estimation accuracy.

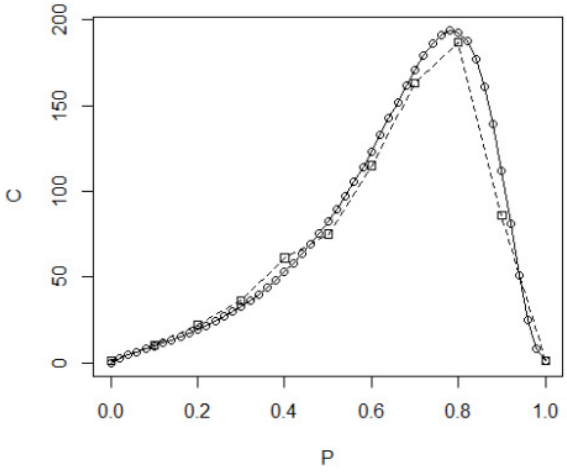

FIGURE 2. Accuracy test of the approximation formula

2.2. *Not uniform distribution*

Next, we turn to the general case, when different attributes may have different probability values. On the other hand, according to our base condition, this probability is independent from the single objects:

$$\forall j, i_1, i_2 : p_{i_1,j} = p_{i_2,j}.$$

In this case, the general formula can be transformed into the following form:

$$C = \sum_{n=1}^{N} \binom{N}{n} \sum_{Y \subset T} \prod_{j \in Y} p_j^n \prod_{j \in B \cup D} (1 - p_j^n)(1 - \prod_{j \in Y} p_j)^{N-n}. \tag{2.3}$$

In the expression, the symbol Y denotes an arbitrary subset of the attributes. In the following figures, the results of some comparison tests can be observed. In the tests, the results of the theoretical calculations are compared with the experimental measurements. Figure 3 is related to the parameter set (N=100; M=8,10,12,14,16; P=0.1-0.3). As the comparisons show the theoretical model provides a very good approximation.

3. SAMPLING-BASED COST REDUCTION

Although, the approximation algorithm presented in the previous section, provides a good accuracy, it has a significant weakness: it has a very high execution cost. The

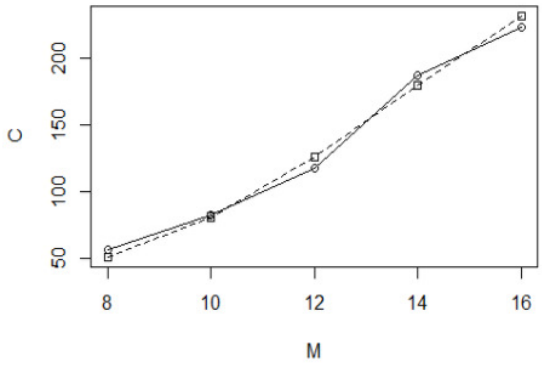

FIGURE 3. Accuracy test of the approximation formula

algorithm requires a large amount of time for smaller problems too. For example, taking the context with parameterset (N=100,M=16, P=0.1..0.3), the runtime is over 967 seconds. Figure 4 shows the corresponding dependency between the execution time and the size of the attribute set (M). In order to apply the approximation algorithms to real-life size problems, the base algorithm must be updated to an optimized version.

The full enumeration of the candidates implemented in the baseline version of the approximation algorithm is not suitable to handle larger contexts. The aim of the optimization is to eliminate some candidates in the evaluation process. The initial formula containing full enumeration is equal to

$$C = \sum_{n=1}^{N} \binom{N}{n} \sum_{Y \subset T} \prod_{j \in Y} p_j^n \prod_{j \in B \cup D} (1 - p_j^n)(1 - \prod_{j \in Y} p_j)^{N-n}. \tag{3.1}$$

This formula processes all candidates ordered by the sub-context size. Our analysis shows that the candidates with different size values usually have very different probability weights. Considering all the candidates with size parameter (n,m), the corresponding sub-total is given with

$$C_{n,m} = \binom{N}{n} \sum_{Y \subset T, |Y|=m} \prod_{j \in Y} p_j^n \prod_{j \in B \cup D} (1 - p_j^n)(1 - \prod_{j \in Y} p_j)^{N-n}. \tag{3.2}$$

Investigating the $cnt_{n,m}$ values for different (n,m) parameters, we can see that there are dominating (n,m) pairs where the values are significantly higher than the values on the complement area. Fig F5 shows the distribution for the context (N=30, M=14, P_1=0.2,P_2=0.6). The x and y axes correspond to n and m, while the z axis denotes the count value. In this example, the dominance area involves only small (n,m) values.

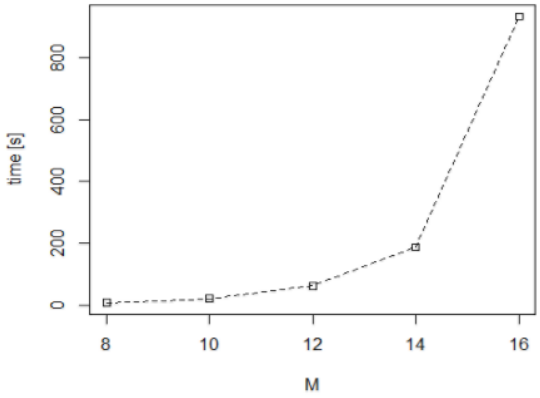

FIGURE 4. Execution cost function

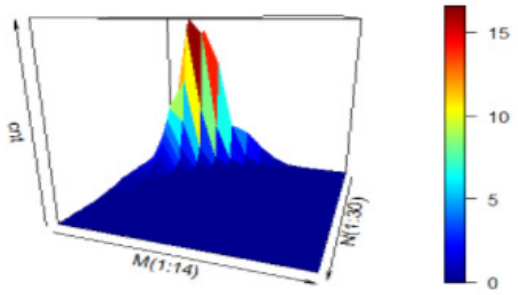

FIGURE 5. The $C_{n,m}$ distribution

The position of the dominance area depends on the attribute probabilities of the input context. The Figures 6-9 show the maximum positions for different P values taking uniform attribute probability distribution. The parameter of the input context is (N=40,M=14,P=(0.2,0.6,0.8,0.96)).

An important observation is that the dominance zone for the not very dense contexts is always near the origo position. In the case of not uniform attribute probability distribution, the dominance zone is an area near the corresponding uniform dominance positions. This case is shown in Figure 10, where the input context is generated with the parameterset(N=10,M=14,P=0.6-0.8).

Based on the presented properties of the dominance zones, the implemented optimization applies the following steps:

- For the calculation of a $C_{i,j}$ component, a sampling technique is applied instead of full enumeration of the corresponding candidates.

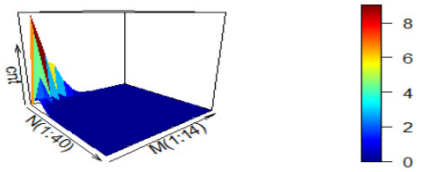

FIGURE 6. The $C_{n,m}$ distribution ($P = 0.2$)

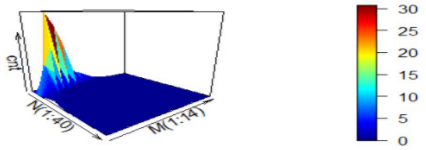

FIGURE 7. The $C_{n,m}$ distribution ($P = 0.6$)

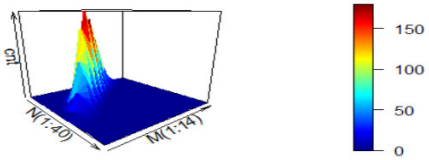

FIGURE 8. The $C_{n,m}$ distribution ($P = 0.8$)

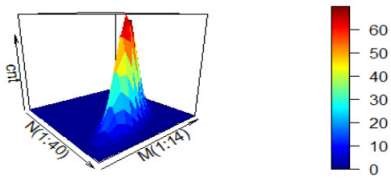

FIGURE 9. The $C_{n,m}$ distribution ($P = 0.96$)

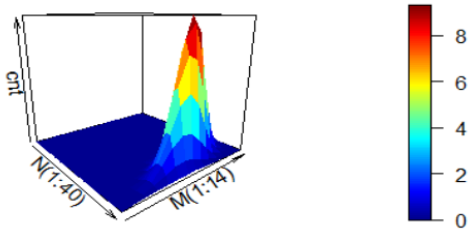

FIGURE 10. The $C_{n,m}$ distribution

- The enumeration of the candidates is restricted only to the dominance zones, the candidates outside the dominance zone are eliminated.

The sampling process applies the method of simple sampling without replacement approach. Using this method, the corresponding confidence interval can be given with

$$\triangle \bar{x_n} = t \frac{S_n}{\sqrt{n}} \sqrt{(1 - \frac{n}{N})},$$

where n: size of the sample, N: size of the population, $\bar{x_n}$: mean value, S_n: standard deviation, t: t-score value. Using this formula and the desired t-score value, we can

determine an optimum sample size with the following formula:

$$n_o = \frac{N}{1 + \frac{N \Delta \bar{x}}{t^2 S_n^2}}.$$

Our proposed algorithm uses this formula to determine the length of the sampling. To determine the required apriori values, we use a pre-sampling phase with a moderate fixed sample size. The estimation of the deviation value is based on the result of this pre-sampling phase.

According to our experiences, the initial pre-sampling phase generates usually an approximation value significantly higher than the real deviation. To optimize this process, we have implemented a mechanism to stop the sampling process if the deviation of the last time window is below of the threshold. Thus there are two termination criteria in the sampling process :

- the length of the sampling is equal to the calculated;
- the deviation of the last time window is below a the threshold.

The second cost reduction method restricts the full enumeration on the parameter space (n,m) to the dominance region. In general case, this reduction is performed with the following algorithm:

- build up a coarse grid on the (n,m) parameter space
- calculate $C_{n,m}$ for each node of the grid
- determine $(n_0,m_0) = argmax_{(m,n)}\{cnt_{n,m}\}$
- select a dominance factor $1 > \alpha > 0$
- process all elements (n,m) which are connected to (n_0,m_0).

A connectivity relationship is used to determine the dominance zone as a maximal cluster. The relationship is defined on the usual way. Two elements (n_s,m_s), (n_l,m_l) are connected if

- $C_{n_s,m_s} \geq \alpha \cdot C_{n_0,m_0}, C_{n_l,m_l} \geq \alpha \cdot C_{n_0,m_0}$
- there exists a sequence of neighboring connected elements:$(n_1 = n_s, m_1 = m_s), (n_2,m_2), (n_2,m_3), ..., (n_i = n_e, m_i = m_e)$.

The method merges only those elements (n,m) into the dominance zone where $C_{n,m} \geq \alpha \cdot C_{n_0,m_0}$. The exploration starts at the element (n_0,m_0). At a given position it will test all the neighboring elements. The method implements a greedy algorithms and it terminates if no new element with high $C_{n,m}$ value can be discovered. The sampling process will be executed for all elements of the discovered dominance zone.

In the case of sparse contexts, the general algorithm can be reduced to a faster variant. In this case, the dominance zone is located near the origo position. In the practical applications, the contexts usually are sparse contexts, otherwise we would manage exponential large set of concepts. For the sparse context the dominance zone is explored in this way:

- initially, we take the element (0,0) and calculate $C_{0,0}$

- a nested loop on the elements is started, where both the object and attribute indexes start at value 0. Both loops terminate if the increase of the accumulated cnt value is below a threshold. If the increase of the accumulated value is very low, then the element last tested is outside the dominance value.

The proposed method is implemented as an algorithm returning the expected mean value and the corresponding deviation value. The deviation is estimated with the following approach. At a given element (n,m), we determine first the standard deviation of the sample values:

$$s = \frac{\sqrt{\sum_i^l \frac{(f_i - \bar{f})^2}{l}}}{\sqrt{l}}.$$

The standard deviation related to $cnt_{n,m}$ is equal to

$$s_{n,m} = \binom{M}{m}\binom{N}{n}s.$$

Considering the whole element space, the total deviation can be calculated with

$$S = \sqrt{\sum_{(m,n)} s_{m,n}^2}.$$

4. TEST RESULTS

Based on the performed tests, we can say that the proposed algorithm provides a unique and fast approximation tool to determine the expected number of concepts for contexts where the attribute probabilities are independent from each others and from the object instances. Some typical test results are shown in Table 1. The meaning of the columns is the following: C_e: measured average, C_a: value by base approximation; C_{ao}: value by optimized base approximation; t_e: time for concept enumeration, t_a: time of base approximation; $time_{ao}$: time of optimized base approximation; Prisner: the value of the Priosner approximationIn the table some values are left blank, because they could not be calculated due to high execution cost or to high memory demand. The test results show that the baseline upper approximation provide a very inaccurate values, they cannot be used for practical cost estimations.

Table 2 shows some values related to the approximation of the standard deviation C_e: measured average, ad_a: measured deviation, C_{ao}: calculated average, sd_{ao}: calculated deviation)

The approximation algorithm, can be used also for extreme parameter values where the available concept enumeration methods would require extreme large execution time. The initial approximation algorithm with evaluation of all components can be used only for small sized contexts ($N < 3000, M < 30$). The best concept set enumeration algorithms can process also larger contexts, in our test environment, the threshold value is about ($C < 5000000$). On the other hand, the proposed optimized

TABLE 1. Test results on cost efficiency of the approximation method

N	M	P	C_e	C_{ao}	t_e	t_{ao}	Prisner
3000	50	0.1	23889	23460	1.42	0.002	10^{25}
1000	30	0.1	1910	1850	0.11	0.001	10^{13}
3000	100	0.1	183210	189700	7.26	0.002	10^{43}
10000	100	0.1	901331	906413	35.9	0.002	10^{60}
100000	1000	0.1	-	4.71e9	-	0.002	10^{120}
100000	5000	0.01	-	8.1e6	-	0.003	10^{90}
1000000	2000	0.05	-	1.77e12	-	0.004	10^{140}
100	12	0.1-0.3	120	120	0.02	0.1	10^{6}
100	16	0.1-0.3	231	223	0.02	0.1	10^{7}
200	13	0.1-0.2	220	162	0.03	0.1	10^{8}
1000	13	0.1-0.2	461	422	0.06	0.2	10^{10}
10000	13	0.1-0.2	1446	1336	0.18	0.25	10^{12}
10000	30	0.1-0.2	49954	49367	1.55	0.28	10^{20}

TABLE 2. Test results on standard deviation of the approximation method

N	M	P	C_e	sd_e	C_{ao}	sd_{ao}
1000	100	0.1-0.2	238000	12707	247583	12870
5000	100	0.1-0.2	3056012	168400	3029816	144998

method can be applied for larger contexts too ($N < 1000000000, M < 10000, C < 1e25$) with a maximal execution time of 5 seconds. This execution time data shows that the algorithm is very efficient and it can be used for larger complexity analysis too. Next figures presents some complexity functions for larger contexts, Figure 11: ($N = 10000..100000, M = 200, P = 0.1$), Figure 12:($N = 10000, M = 100..1000, P = 0.1$) and Figure 13: ($N = 10000, M = 200, P = 0.02..0.24$).

5. CONCLUSIONS

The calculation of the concept count for any arbitrary context is a hard, NP-complete problem and only rough approximation methods can be found in the literature to solve this problem. The paper proposes a novel algorithmic approach to approximate the total number of concepts where the attribute probabilities are independent from each others and from the single objects. The algorithm is very efficient especially for rare contexts which are mainly used in the practical FCA applications. The proposed algorithm provides a better practical approximation with a significantly better execution cost than the baseline approximation [16], [15]. The method can be used among others also for complexity analysis of large scale FCA problems.

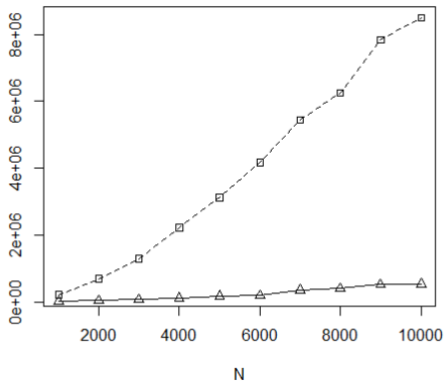

FIGURE 11. Mean and deviation of concept count

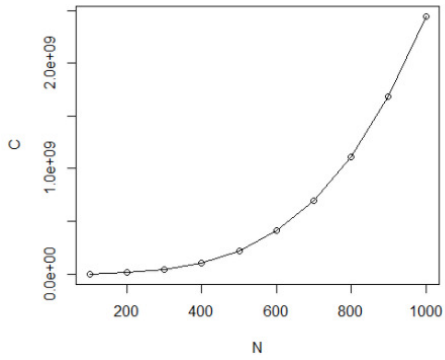

FIGURE 12. Mean and deviation of concept count

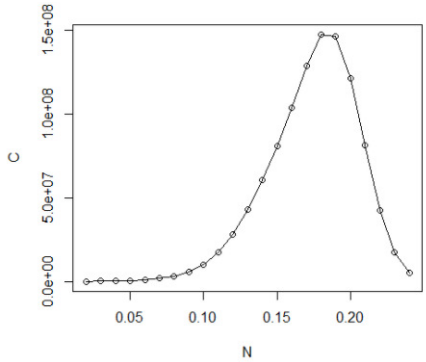

FIGURE 13. Mean and deviation of concept count

ACKNOWLEDGEMENT

This article was carried out as part of the EFOP-3.6.1-16-00011 Younger and Renewing University Innovative Knowledge City institutional development of the University of Miskolc aiming at intelligent specialisation" project implemented in the framework of the Szechenyi 2020 program. The project is supported by the European Union, co-financed by the European Social Fund.

REFERENCES

[1] A. Albano and B. Chornomaz, "Why concept lattices are large," *Proceedings of CLA*, pp. 73–91, 2015.
[2] S. Andrews, "In-close, a fast algorithm for computing formal concept," *Proceedings of ICCS*, pp. 1–14, 2009.
[3] R. Belohlavek and M. Trnecka, "Basic level of concepts in formal concept analysis," *Proc. of International Conference on Formal Concept Analysis*, pp. 28–44, 2012.
[4] M. Boley, T. Gartner, and H. Grosskreutz, "Direct local pattern sampling by efficient two-step random procedures," *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 582–590, 2011.
[5] D. Dubois and H. Prade, "Formal concept analysis from the standpoint of possibility theory," *Proceedings of International Conference on Formal Concept Analysi*, pp. 21–38, 2015.
[6] R. Emilion, "Concepts of a discrete random variable," *Selected Contributions in Data Analysis and Classification*, pp. 247—-258, 2007.
[7] B. Ganter and R. Wille, *Formale Begriffsanalyse – Mathematische Grundlagen*. Springer, 1996.
[8] A. Gionis, H. Mannila, T. Mielikainen, and P. Tsaparas, "Assessing data mining results via swap randomization," *ACM Trans. Knowl. Discov. Data 1(3), 14*, 2007.
[9] M. Klimushkin, S. Obiedkov, and C. Roth, "Approaches to selection of relevant concepts in the case of noisy data," *Proceedings of ICFCA*, pp. 255–566, 2010.
[10] S. Kuznetsov, "On computing the size of a lattice and related decision problems," *Order, 18.4*, pp. 313–321, 2001.
[11] S. Kuznetsov and T. Makhalova, "Concept interestingness measures: a comparative study," *CLA, Vol. 1466*, pp. 59–72, 2015.
[12] S. Kuznetsov and S. Obiedkov, "Comparing performance of algorithms for generating concept lattices," *Journal of Experimentation and Theoretical Artificial Intelligence*, 2002.
[13] G. Murphy, *The Big Book of Concepts*. MIT Press, Cambridge, 2002.
[14] L. Piskova and T. Horvath, "Comparing performance of formal concept analysis and closed frequent itemset mining algorithms on real data," *Proceedings of CLA 2013*, pp. 299–308, 2013.
[15] E. Prisner, "Bicliques in graphs i: Bounds on their number," *Combinatorica*, pp. 109–117, 2000.
[16] D. Schutt, *Abschatzungen fur die Anzahl der Begriffe von Kontexten, PhD Thesis*. TU Darmstadt, 1988.

Author's address

L. Kovács
University of Miskolc, Department of Information Technology, Miskolc-Egyetemváros, Hungary
E-mail address: kovacs@iit.uni-miskolc.hu