Extensible Python Library for Managing Probabilistic Knowledge Structures

Dragan Vidaković*, Milan Segedinac*, Zora Konjović**, Goran Savić*
* Department of Computing and Control Engineering, University of Novi Sad, Novi Sad, Serbia
** Singidunum University, Novi Sad, Serbia
{vdragan, milansegedinac, savicg}@uns.ac.rs*, zkonjovic@singidunum.ac.rs**

***Abstract*—In this paper we have proposed an extensible Python library aimed at utilization and managing probabilistic knowledge structures. A key part of the library is implemented focusing on the probabilistic knowledge structures application to educational domain. The library is structured as three Python modules aimed respectively at: (1) data representation and conversion, (2) basic modeling of local independency, and (3) gradation of a probabilistic knowledge structure. The library was verified against the requirements specification and the obtained models are validated by the chi-square test. The data set used for verification and validation is a subset of PISA testing data containing responses of 340 German students on 5 questions from mathematical literacy.**

***Keywords*—probabilistic knowledge structures; knowledge spaces; parameter estimation;**

I. Introduction

Knowledge assessment starts with defining and asking appropriate question. Starting from the student's response, it is easier to define and ask other questions. After a couple of questions, knowledge state of an individual becomes identifiable. Such question definition and response analysis are labor-intensive tasks, due to the large number of students, whose knowledge has to be assessed and improved based on their current knowledge. Successful creation of an automatic procedure for question definition and response analysis allows better knowledge management and advancement via personalized learning based on current knowledge state of an individual.

Ideal conditions for knowledge assessment of an individual are when the individual is not under pressure. In reality, those conditions are not always possible, and therefore careless errors occur. Also, there are situations, especially with multiple choice questions, when the individual luckily guesses the correct answer to the question, without any understanding of the material. Therefore, approach to these problems has to be probabilistic.

Usage of Python programming language can ease knowledge management with probabilistic approach due to a large number of available libraries for mathematics and engineering, data modelling and analysis, visualization and parallel computing. It is widely used in academia and scientific projects because it is easy to master and performs well. The scientific Python ecosystem (SciPy[1]) provides open source software for scientific computing and it has a large community that uses and develops the ecosystem. Still, as far as the the authors of this paper are informed, currently there aren't any Python libraries for knowledge management based on probabilistic knowledge space theory.

The motivations for this research are multiple. The first one is the importance of a knowledge management and probabilistic knowledge structures in general and in education in particular. The second one is a clear tendency of introducing new models of probabilistic knowledge structures. The third one is a popularity and wide use of the Python programming language across diverse application domains. The last one is the lack of Python library supporting management of probabilistic knowledge structures. That is why the subject of this paper is a development of an extensible Python library supporting management of probabilistic knowledge structures.

The rest of the paper is organized as follows. In Section II, several related works are presented and discussed. In section III, we described the methodology we used to develop the library. In section IV the experimental results are presented and analyzed. Finally, we make a brief concluding mark and give future work in Section V.

II. Related work

Doignon and Falmagne [1] were first to introduce the Knowledge Space Theory (KST). Their work was motivated by shortcomings of the psychometric approach to the competence assessment. Applications of psychometric models resulted in placing an individual in one of a few dozen categories. Such a classification is too coarse to be useful. Doignon and Falmagne proposed a fundamentally different theory with the basic idea that an assessment in an educational course should reveal the individual's knowledge state that represents the exact set of concepts mastered by the individual. Concept represents a type of problem that the individual has mastered.

In KST [2], a knowledge structure is defined as a pair *(Q, **K**)* in which Q is a (finite) nonempty set, and $\boldsymbol{K}$ is a family of subsets Q, containing at least Q and the empty set $\emptyset$. The set Q is called the domain of the knowledge structure, and its elements are referred to as items or problems. The elements of $\boldsymbol{K}$ are *knowledge states*. A knowledge state represents the subset of items in the considered domain that an individual has mastered.

Probabilistic framework in KST [2] handles two types of unexpected responses:

[1] https://www.scipy.org/index.html

- careless errors – an individual does not solve a problem although they have mastered it;
- lucky guess – an individual solves a problem without actually mastering it.

The knowledge state K (representing what is mastered) cannot be identified with the the observable response pattern R (representing what is actually solved), but it has to be inferred from it. A probabilistic model is obtained by specifying a probability distribution on the states of $\boldsymbol{K}$ and the conditional probabilities $P(R|K)$ for all $R \in \boldsymbol{R}$ and $K \in \boldsymbol{K}$, where $\boldsymbol{R} = 2^Q$ denotes the set of all possible response patterns in domain Q. Probabilistic model predicts the marginal distribution on the response patterns as

$$P(R) = \sum_{K \in K} P(R|\, K) * P(K).$$

It may be estimated directly from data consisting of a frequency distribution on the set $\boldsymbol{R}$. N_R denotes the absolute frequency of the pattern $R \in \boldsymbol{R}$ in a sample of size

$$N = \sum_{R \in R} N_R.$$

Currently most prominent software implementation of the abovementioned theory is the DAKS [3] package in programming language R. It supports fitting and testing of probabilistic knowledge spaces. Even though the R programs can be executed in Python as sub-processes, for a more efficient work a native Python implementation that supports probabilistic knowledge structures management, and enables simple introduction of new probabilistic knowledge structure would be valuable.

III. Methodology

In order to comply with extensibility requirement, the library is structured as three Python modules:

1. Data representation and conversion;
2. Basic modeling of local independency;
3. Gradation of probabilistic knowledge structure.

First module handles raw data and converts it to data structures suitable for the next modules. Second module estimates parameters for probabilistic knowledge structures via basic modeling of local independency, while the third module calculates gradation of provided probabilistic knowledge structure.

A. Data representation and conversion

In order to simplify data usage, knowledge structure is represented as binary matrix where every column indicates if corresponding item is in certain knowledge state. Response patterns are represented using associative map data structure, where the key is the state from the knowledge structure, and the value is the response frequency of the state within the set of response patterns.

Implementation of this module was based on Pandas [4] library from SciPy ecosystem. Pandas provides high performance and easy-to-use data structures and tools for data manipulation. In the implementation proposed in this paper, the data are represented as Pandas' dataframes. Knowledge structure itself is also represented as a dataframe where the header contains all items from the domain, and the values are fields of a binary matrix of the knowledge structure. Probability distribution is represented as a dataframe in which the header contains knowledge states, and the values are corresponding probabilities. Probabilistic knowledge structure error probabilities are two dataframes (one for every error type) where the header contains items from the domain of the knowledge structure, and the values are corresponding error probabilities. In order to preserve order of knowledge states within knowledge structure, response patterns are represented as an ordered dictionary, where the keys are knowledge states, and the values are response frequencies.

The data consists of the results of the assessment of students' knowledge. It is necessary to extract the knowledge structure and response patterns from raw data for further usage in probabilistic knowledge structures. For that purpose, data representation and conversion module contains methods for:

- response patterns and response frequencies extraction from raw data;
- response patterns to knowledge structure conversion.

Following two modules rely on the functionalities of the described module.

B. Basic modelling of local independency

Basic local independence model (BLIM) is most widely utilized probabilistic model [2]. In the BLIM, for each $q \in Q$ it is assumed that there are real constants $0 \leq \beta_q < 1$ and $0 \leq \eta_q < 1$ such that for all $R \in \boldsymbol{R}$ and $K \in \boldsymbol{K}$:

$$P(R \mid \mathrm{K}) = (\prod_{q \in \mathrm{K} \setminus R} \beta_q)(\prod_{q \in \mathrm{K} \cap R} (1-\beta_q))(\prod_{q \in R \setminus \mathrm{K}} \eta_q)(\prod_{\overline{q \in Q \setminus (R \cup \mathrm{K})}} (1-\eta_q))$$

This condition encompasses two assumptions, both reflected on the given knowledge state $K \in \boldsymbol{K}$. The first assumption implies that the state $K \in \boldsymbol{K}$ subsumes all systematic effects on the solution behavior. Second, the solution of a problem $q \in Q$ depends on parameters β_q and η_q, which are interpreted as the probability of a careless error and a lucky guess, respectively. In particular, the β_q and η_q are independent of the knowledge state of the individual.

The module that implements BLIM provides two methods for parameter estimation:

- Minimum Discrepancy (MD);
- Maximum Likelihood (ML).

MD method [5] is based on index that describes how observed response pattern fits knowledge structure $\boldsymbol{K}$. For every knowledge state $K \in \boldsymbol{K}$**,** probability distribution and both error probabilities are calculated directly, based on the distance between the response pattern R and knowledge state K. The distance is calculated as $d(R, K) = |(R \setminus K) \cup (K \setminus R)|$. ML method [5] is based on Expectation-Maximization (EM) algorithm. This algorithm artificially widens data assuming that in addition to the response pattern R we have to take into consideration the corresponding knowledge state K. In E-step algorithm calculates expected values of log-likelihood of complete data set paying attention to the response pattern frequencies and current estimation of probability and error parameters calculated in previous iteration:

$$m_{RK} = N_R * P(K| R, \hat{\beta}^{(t)}, \hat{\eta}^{(t)}, \hat{\pi}^{(t)}).$$

M-step maximizes expectations from E-step by calculating new estimations. The repetition of these steps for appropriate number of times guarantees convergence.

The implementation of these methods relies on NumPy library [6]. BLIM is represented as separate class in this module, requiring a constructor argument for a method that will be used for parameter estimation. Constructor also requires knowledge structure and frequencies of response patterns, and additional arguments for a specific estimation method (e.g. number of iterations for ML).

C. *Gradation of probabilistic knowledge structure*

There are two classes of knowledge structures in which error probabilities for some items can be set to zero while preserving model fitness. First class are knowledge structures where careless errors for some items can be set to zero, and they are labeled as backward graded knowledge structures. In second class of knowledge structures, lucky guess for some items can be set to zero, and they are labeled as forward graded knowledge structures [7].

Gradation of probabilistic knowledge structure module provides methods for determining if provided knowledge structure is forward or backward graded in an item. As a result, each method returns a list of boolean indicators one for every item.

IV. Evaluation

Solution is verified and validated on domain dataset consisting of the results of the assessment of students' knowledge.

A. *Data set*

Data set for verification of the library proposed in this paper is a part of PISA (Programme for International Student Assessment)[2] testing from 2003. It consists of 340 responses given by German students to 5 questions on mathematical literacy. Data is anonymized by removing student's personal information, and the questions themselves are replaced with letters *a*, *b*, *c*, *d* and *e*. Accurate and inaccurate responses on specific question are labeled with 1 and 0 respectively [3].

B. *Verification*

Module for Data representation and conversion was used to extract the knowledge structure and the response pattern frequencies from raw data. 23 knowledge states were identified. Table 1 contains identified knowledge states with corresponding frequency.

In order to estimate BLIM parameters, knowledge states with frequency 1 were removed from obtained knowledge structure. Probabilities obtained after estimations with both MD and ML methods are shown in Table 2. With uniform distribution of knowledge states and initial values for careless and lucky guess errors set to 0.1, ML method took 333 iterations to complete.

State	Frequency	State	Frequency
00000 = *{∅}*	20	10100 = *{a,c}*	14
00010 = *{d}*	1	10101 = *{a,c,e}*	1
00100 = *{c}*	4	10110 = *{a,c,d}*	5
01000 = *{b}*	11	10111 = *{a,c,d,e}*	1
01001 = *{b,e}*	2	11000 = *{a,b}*	61
01010 = *{b,d}*	1	11001 = *{a,b,e}*	10
01100 = *{b,c}*	9	11010 = *{a,b,d}*	16
01101 = *{b,c,e}*	1	11100 = *{a,b,c}*	67
01110 = *{b,c,d}*	2	11101 = *{a,b,c,e}*	17
10000 = *{a}*	41	11110 = *{a,b,c,d}*	40
10001 = *{a,e}*	3	11111 = *{a,b,c,d,e}*	12
10010 = *{a,d}*	1		

Table 1. Knowledge states and response frequencies

MD estimations	ML estimations
p*{∅}* = 0.059880	p*{∅}* = 0.100657
p*{c}* = 0.011976	p*{c}* = 0.020131
p*{b}* = 0.032934	p*{b}* = 0.028263
p*{b,e}* = 0.005988	p*{b,e}* = 0.012317
p*{b,c}* = 0.026946	p*{b,c}* = 0.033849
p*{b,c,d}* = 0.005988	p*{b,c,d}* = 0.010218
p*{a}* = 0.122754	p*{a}* = 0.109472
p*{a,e}* = 0.008982	p*{a,e}* = 0.015198
p*{a,c}* = 0.041916	p*{a,c}* = 0.039831
p*{a,c,d}* = 0.014970	p*{a,c,d}* = 0.021014
p*{a,b}* = 0.182635	p*{a,b}* = 0.130818
p*{a,b,e}* = 0.029940	p*{a,b,e}* = 0.037108
p*{a,b,d}* = 0.047904	p*{a,b,d}* = 0.057186
p*{a,b,c}* = 0.200599	p*{a,b,c}* = 0.141075
p*{a,b,c,e}* = 0.050898	p*{a,b,c,e}* = 0.063222
p*{a,b,c,d}* = 0.119760	p*{a,b,c,d}* = 0.117925
p*{a,b,c,d,e}* = 0.035928	p*{a,b,c,d,e}* = 0.061715

Table 2. Probability estimations

Although it is slower that MD method, ML method gives better estimations for careless errors and lucky guesses. Table 3 contains error estimations with ML method, while the values estimated with MD method were so low they can be disregarded.

	a	***b***	***c***	***d***	***e***
$\boldsymbol{\beta}$	1.975e^{-34}	3.406^{e-36}	6.816^{e-19}	0.162	0.305
$\boldsymbol{\eta}$	0.3	0.149	0.000003	4.523e^{-69}	4.354e^{-45}

Table 3. ML error estimations

Using Gradation of probabilistic knowledge structure module, it was determined that the knowledge structure from data set is forward graded in items *a* and *b* and backward graded in items *c* and *d*.

[2] http://www.oecd.org/pisa/

C. Validation

Validation is used to determine if the estimated values are in accordance with data. Chi-square non-parametric test was used for validation, due to its robustness and independence from probability distribution. Unlike some parametric and other non-parametric tests, chi-square test results give significant information about results of every distinct group on the test. Therefore, chi-square test is the recommended validity test in knowledge-state-related research. For P-value the reference for accepting and rejecting hypothesis is: [2]

1. strong rejection: $P \leq 0.01$;
2. rejection: $P \leq 0.05$;
3. acceptance: $P > 0.05$.

After parameter estimation, chi-square test was conducted in order to estimate deviation of estimated values from data set. As we can see from chi-square test results shown in Table 4, both models can be accepted.

Method	Degrees of freedom	P-value
MD	5	0.99
ML	5	0.99

Table 4. Chi-square test result

V. Conclusion

In this paper we proposed an extensible Python library for managing probabilistic knowledge spaces. Library contains modules for data representation and conversion, parameter estimation via several methods and probabilistic knowledge structure gradation. It is published as an open source project[3]. The library was verified against the data set from education domain and validated by the chi-square test.

The results indicate that the proposed model is adequate for situations with a relatively small knowledge space. In addition, the model requires that each student gives an answer to every question from the domain. These limitations will be taken into account.

One particularly important constraint of the considered mathematical model, is that *the adoption of the new knowledge extends the set of individual's knowledge only quantitatively*. Therefore, one of the directions of the future research will be to develop mathematical model that deals wtih In order to lift this assumption, which is absolutely necessary for realistic situations, the forgetting as a part of the learning process.

The proposed library will be the starting point for the above-mentioned tracks of future research.

References

[1] J.P. Doignon and J.C. Falmagne, "Spaces for the assessment of knowledge", in *International Journal of Man-Machine Studies*, vol.23, pp 175-196, 1985.

[2] J.C. Falmagne and J.P. Doignon, "Learning Spaces", *Springer*, 2011.

[3] A. Ünlü and A. Sargin, "DAKS: An R Package for Data Analysis Methods in Knowledge Space Theory", in *Journal of Statistical Software*, vol. 37.2, pp 1-31, 2010.

[4] W. McKinney, "Data Structures for Statistical Computing in Python", in *Proceedings of the 9th Python in Science Conference*, pp 51-56, 2010.

[5] J. Heller and F. Wickelmaier, "Minimum Discrepancy Estimation in Probabilistic Knowledge Structures", in *Electronic Notes in Discrete Mathematics*, vol. 42, pp 49-56, 2013.

[6] T.E. Oliphant, "A guide to NumPy", *Trelgol Publishing*, 2006.

[7] A. Spoto, L. Stefanutti and G. Vidotto, "On the unidentifability of a certain class of skill multi map based on probabilistic knowledge structures", in *Journal of Mathematical Psychology*, vol. 56, pp 248-255, 2012.

[3] https://github.com/milansegedinac/kst