STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVIII**, Number 1, 2013

FUZZY COMPUTING FOR COMPLEXITY LEVEL OF EVALUATION TESTS

TIBERIU BAN

Abstract. Students tend to make mistakes in evaluation tests that follow a pattern which can be mined; this allows the level of complexity for a given set of tasks in a test paper to be computed in advance. The goal of this paper is to present a mathematical way to compute the level of complexity for a given set of tasks based on a fuzzy model as an improvement from the crisp model. This indicator can be effectively used to predict in advance the degree in which the association rules already mined will trigger chains of mistakes in a given set of evaluation tasks.

1. Background

This paper focuses on mining existing data association between chains of mistaken items from a evaluation test. The main rationale is that if a specific subset of items are mistaken, then there is a computable chance with a specific threshold of confidence level that this chain will trigger one or several other test items to be mistaken as well. After the point where enough data association rules have been already mined, this information can be used in analysing a new set of test items in order to determine the extent to which chain of items can trigger other items to be solved incorrectly. This will be presented with the aid of an indicator called Complexity Level of a set of tasks within a test paper.

The field of Data Mining and its main concepts are taken into consideration as presented in [10] and [11] and continues the path of Data Association presented in [3] and [6], as well as the Discovery of Significant Rules techniques presented in [13]. The APriori algorithm used are presented in [4] and also [12]. Applications of Data Analysis Technique in the field of detecting and mining mistakes made by students are also covered in [9], as well as adaptation of

Received by the editors: January 15, 2013.
2010 *Mathematics Subject Classification.* 68P15.
1998 *CR Categories and Descriptors.* H.2.8. [**Database Applications**]: DataMining – *Data Association Methods*.
Key words and phrases. Data Mining, Data Association, APriori, Complexity of Test Paper, Mathematical Model of Mistakes.

fuzzy approach to this domain of study, as well as applications of Data Analysis Technique in assessing other performance criteria from students projects in [8].

The Complexity Level will present the degree to which existing chains of item tasks can cascade trigger more items to be also mistaken in turn. The higher the value of a complexity level, the lower the score for a statistically average student will be. The Complexity Level is only relevant if its value is relative, not absolute, ranging between 0 and 1.

Allthough it builds on several widely used techniques of Data Association and APriori Algorithm, this application is original in its nature of applying the steps from a supervised learning approach to the known methodology. Also, the field of study is opened to fuzzy approach that comes as a natural extension of the crisp model, each element is taken into account not only with its support and confidence level but also the degree of membership to fuzzy classes. This approach builds on the fiability of the model, making it more robust. The algorithm and the data processing techniques are adapted to the field of study with respect to the robustness of the approach.

Definition 1: An *evaluation test* is defined as a set of tasks.

Each task can either be solved correctly or solved incorrectly. A correctly solved task will be graded with the entire score for the particular task. An incorrectly solved task will be graded with zero score.

Definition 2: A *mistake* is defined as a task that is solved incorrectly.

The first mathematical model that was presented in [1] had a crisp approach to it, thus making it a simplified model. In the first mathematical model each task was considered as having the same amount of points as total score.

The crisp approach consisted in the fact that it was taken into consideration the possibility to divide the tasks from a given evaluation paper in distinct subsets, with each subset consisting of items that had a support level above a predefined threshold.

Also, in order to keep the mathematical model crisp, more assumptions and restrictions have been in place [2] such as each student is present at all tests and each student had the same number of items presented.

The structure of the paper has two main parts. The first part sets the theoretical basis of the Fuzzy Mathematical Model in Section 2 and a review of the current notions in Data Association used in Section 3 and the adaptation of APriori algorithm to the business domain in Section 4. The second part presents an experiment made on real data instances that is presented in Section 5 and the results in Section 6.

2. The Fuzzy Model for Computing the Level of Complexity of an Evaluation Test

The first mathematical model presented in [1] is not sustainable in real life situation due to the fact data regarding the itemset each question belongs to is not crisp. A problem arises in the fact that the same question can be included in multiple itemsets. A good way of handling this is refering to a new fuzzy approach to computing the complexity of a test paper.

Definition 3: The *complexity level of a test paper* is defined as the relative indicator best estimate of the percent an average student is likely to lose out of the total amount of points for the given test paper.

Before presenting a formula to compute this indicator, here are a few theoretical observations:

Let's consider a test paper that consists of four questions labeled A, B, C and D. For this test let's consider a sample of 10 papers with the list of mistaken questions from Table 1.

Paper No.	Mistaken Items
1.	A, B
2.	A, B
3.	A, B
4.	D
5.	D
6.	A
7.	A
8.	D
9.	n/a
10.	A

Table 1. Ten data instances - Mistakes gathered from test papers

After running the APriori algorithm described in the following sections the candidate itemsets discovered are given in Table 2.

These candidate itemsets can be visualized in a lattice structure. In this lattice the void itemset acts as the general infimum and the candidate itemset with all the items as the general supremum for the given lattice. The lattice would be constructed as follows, with respect to [5]:
Let $\mathbf{C}$ be the set of candidate itemsets. $\forall C_i, C_j \in \mathbf{C}$,
$inf(C_i, C_j) = C_i \cap C_j \, and \, sup(C_i, C_j) = C_i \cup C_j$.

In order to further clarify this notion, for the previous example we will show the corresponding lattice. With respect to style and clarity and also in order to focus only on the important nodes in the lattice, we will only

Itemset	Support Level
{A}	0.6
{B}	0.3
{C}	0.2
{D}	0.3
{A, B}	0.3
{C, D}	0.2

TABLE 2. Candidate itemsets for Table 1

implement the nodes that correspond to candidatesets with nonzero support. Figure 1 illustrates the lattice generated with the candidatesets. Figure 2 adds more information to Figure 1, by also displaying the corresponding support level for each candidateset.

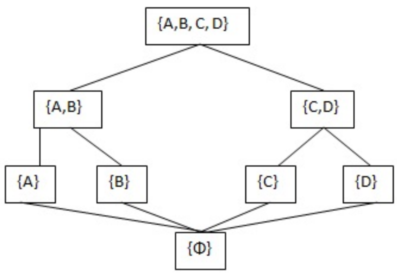

FIGURE 1. Lattice of candidatesets

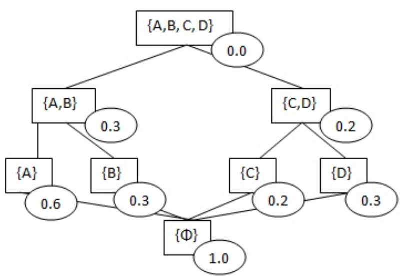

FIGURE 2. Lattice of candidatesets with support values

However, the itemset {B} can be pruned down because of the fact that itemset {B} never appeared isolated and is always as subset of {A, B}. For the same reason candidate itemset {C} can also be pruned down. The remaining itemsets are given in Table 3.

Itemset	**Support Level**
{A}	0.6
{D}	0.3
{A, B}	0.3
{C, D}	0.2

Table 3. Candidate itemsets after prunning down

Question A belongs also to the itemset {A} as well as to the itemset {A, B}. Same situation occurs for question D which belongs also to itemset {D} as well as to itemset {C, D}.

In a fuzzy approach we need to build a membership function that will estimate the degree of membership for a given question to an itemset. This process is with respect to the number of occurences of the question to each of the itemsets.

With respect to [14], let's consider the following definitions:

Definition 4: A *fuzzy set* is a "class" with a continuum of grades of membership.

Definition 5: Let X be a space of points (objects), with a generic element of X denoted by x. Thus, X = {x}. A fuzzy set(class) A in X is characterized by a *membership (characteristic) function* $f_A(x)$ which associates with each point in X a real number in the interval [0, 1], with the value of $f_A(x)$ representing the "'grade of membership" of x in A.

The membership function in the first iteration is represented in Table 4.

Question	**Itemset**	**Membership value**
A	{A}	0.5
A	{A, B}	0.5
B	{A, B}	1
C	{C, D}	1
D	{D}	0.33
D	{C, D}	0.66

Table 4. Values for Membership Function on Candidate Itemsets

In order to clarify, let x be an item (i.e. mistake in a test). Let X be a candidateset that is a fuzzy set. The grade of membership of x in X is computed as the normalized "contribution" of supp(x) towards computing supp(X). This approach is derived from supervised learning techniques in such manner that it uses known computable values from already known data instances in order to gain knowledge to be used in future data instances.

Definition 6: The *membership function* is defined as $\mu : Q \times I \rightarrow [0, 1]$, where Q is the set of tasks from the evaluation test and I is the total set of candidate itemsets. The value $\mu(q, i)$ represents the membership value of the question q to the itemset i

The following definitions extend the work from [2] towards the fuzzy sets.

Definition 7: The *cost function* is defined as $C : I \rightarrow \mathbf{R}$, where I is the total set of candidate itemsets. $C(i)$ represents the total amount of score that can be graded in the event that all the items from the itemset i would be solved correctly.

Definition 8: The *support function* is defined as $sup : I \rightarrow [0, 1]$, where I is the set of all candidate itemsets.

Definition 9: The *indicator TotalScore* is defined as the sum of costs of all tasks in the set of tasks Q. TotalScore will be computed as follows:

$TotalScore = \sum_{q \in Q} C(\{q\})$

Definition 10: The *Complexity Level* for a given evaluation test will be computed with the following formula:

$Complexity = \frac{\sum_{i \in I, q \in i} \mu(q,i) \times sup(i) \times C(i)}{TotalScore}$

3. Association Rules Notions

The aim of this section is to review the main concepts of Association Rules as well as the working algorithm used to mine valid association rules from a large set of data instances. We reccomend [12] for further reading.

As presented in [12], Association Rules are similar in nature to Classification Rules. This technique aims to extract valid knowledge from existing dataset in form of rules like the following:

$X_1 \wedge X_2 \wedge ... \wedge X_n \rightarrow Y[C, S]$

where $X_1, X_2, ..., X_n$ are attributes from the dataset. If in a single item these attributes have a distinct combination of values, there is a computable probability to predict a distinct value for attribute Y. Naturally, on the right hand side of the rule there can be more than a single attribute, thus making the discovery process more difficult because of the nature of predicting more variables in a single rule.

A brute force algorithm might be able to attempt to have every combination of a subset of attributes for both the left hand side and the right hand side of the association rule and to consider this a hypothesis that needs to be validated, but such attempt would require an enormous computing power gone to waste, considering only very few of these artificial hypothesis would check out as valid after confronting them to existing items in the dataset.

A better technique would be to prune down rules and their branches that are constructed from these rules by adding more items on both sides of the rule. The criteria on which such pruning can be safely done is the coverage of the rule (the number of insances the rule can correctly predict) and the accuracy (the proportion of the number of items or instances from the dataset the rule can be applied to).

As first mentioned [4] and later refined in [10], the coverage of a rule is expressed by its support level, while the accuracy is computed based on its confidence level.

In order to define more clearly these two indicators, let's consider an association rule in the form of $\{itemi_1, ..., itemi_k\} \rightarrow \{itemj_1, ..., itemj_l\}$ and let's define the support set of the association rule, as presented in [10] the set defined by the reunion of the items both in the left hand side and the right hand side of the rule

$\{itemi_1, ..., itemi_k, itemj_1, ..., itemj_l\}$

Let's formally define the support level as follows [10]:

Definition 10. The *support level of a subset* $\{itemi_1, itemi_2, ..., itemi_k\}$ is the percentage where all the items from the given subset were present in the same transaction, out of the total number of transactions.

Also, let's formally define the confidence level of an association rule as follows:

Definition 11. The *confidence level of an association rule*

$\{itemi_1, ..., itemi_k\} \rightarrow \{itemj_1, ..., itemj_l\}$ represents the percentage of transactions where both set of items were present, out of the total number of transactions where the first set of items

$\{itemi_1, ..., itemi_k\}$ were present.

In order not to generate association rules that are too weak or that apply extremely infrequent, two more indicators are needed: minimum support level (called minsup) and minimum confidence level (called minconf).

According to [10], these two indicators should first be set a bit restrictive in order to avoid generating too many association rules and then slowly relaxing the value of the minimum confidence level, until an acceptable number of association rules are determined.

Basically, we only need to focus on association rules with high coverage. We will refer to the items from the support set of a specific association rules, regardless of the position of each item, either left hand side or right hand side. We would only seek combination of attributes that have a minimum coverage, more precise the support level to be above the preset threshold of minsup.

In order to discover Association Rules between mistakes belonging to a specific test paper the APriori algorithm can be used, since all its perquisites are met [6].

4. The Algorithm Used for Generating Rules Efficiently

There are several algorihms available that carry on the task of generating association rules with a specified minimum support and meeting the minimum confidence level. Each of these algorithms follow two general steps. Apriori algorithm has been chosen for this particular experiment. The basics for the APriori algorithm has been presented first in [3].

4.1. **General Algorithm for Generating Association Rules.** The algorithm described in detail in [12] starts by generating all one item sets with the given minimum coverage. Then it uses these sets as base in order to generate all two items sets, three item sets until either all items available in the attribute list are included in the itemset. Since this case would imply a hard partition of all available attributes in the left hand side and the right hand side of an association rule, this case is less likely to occur in every dataset.

As stated both in [3] and also in [4] another condition to end the first step of the algorithm after generating a k-item set meeting the minimum support level, no other (k+1)-item set can be obtained by adding an extra item to any existing k-item sets. This condition is more likely to be the real marker to stop this step and consider the existing item sets of up to k items to be candidates for support set of valid association rules.

In order to generate a (k+1)-item set, an extra item is added to an existing k-item set generated at the previous iteration. Even more, in order to prune down unnecessary computing effort in generating all (k+1)-item sets, each such set

$\{itemi_1, itemi_2, ...itemi_{k+1}\}$ needs to have all of its k item subsets already in the list of valid generated k-item sets at the previous iteration. If any k item subsets does not meet the minimum support level, then the (k+1) item set can not meet the minimum support level either.

A strategy to avoid unnecesary generations of (k+1) item sets is presented in [12]. All k-item sets are to be sorted using the same criteria, either alphabetically or ascending if the items are coded using numbers.

If there are two k item sets $S1 = \{itemi_1, itemi_2, ..., itemi_{(k-1)}, itemi_k\}$
$S2 = \{itemi_1, itemi_2, ..., itemi_{(k-1)}, itemi_{k'}\}$
that have k-1 common items and exactly one different item,

$Card(S_1 \cap S_2) = k - 1$ and
$Card(S_1 - S_2) = 1$ and
$Card(S_2 - S_1) = 1$,

then a new set S' can be obtained by joining the two sets S1 and S2,

$S' = S1 \cup S2$ and

$Card(S') = k + 1$

Furthermore, in order to avoid generating the same (k+1) item set out of several distinct pairs of k item sets, we could only take the k item sets that have the first (k-1) items in their intersection set.

For instance if we have the following 3-item sets that already meet the minimum support level and their items have been ordered alphabetically

$\{A, B, C\}, \{A, B, D\}, \{A, C, D\}, \{A, C, E\} and \{B, C, D\}$

the union of the two that have the same first two items identical

$\{A, B, C\} \cup \{A, B, D\} = \{A, B, C, D\}$

is to be considered, since any other union like

$\{A, B, D\} \cup \{A, C, D\} = \{A, B, C, D\}$

would end up returning the same 4-item candidate that we already collected.

Considering this same example, the following union

$\{A, C, D\} \cup \{A, C, E\} = \{A, C, D, E\}$

which does not meet the minimum support level and is not a valid 4-item candidate because a 3 item subset $\{C, D, E\}$ does not meet the minimum support level and was not included in the 3-item sets generated at the previous iteration.

The process of checking a (k+1) item set whether it meets the minimum support level or not is simplified even more by using hash tables. Each item is to be removed in turn and the remaining k-item set is checked whether it is part of the valid k item sets already known.

Finally, any (k+1) item sets needs to have its support level actually being computed, because the above method only tells for sure if it does not.

5. Presenting the experiment

A test consisting of 36 questions was presented to 75 students, each student having received the same set of test items, with a random factor for the order in which the items were presented. Each student received the same amount of time consisting of 45 minutes to complete the test.

Each test item consisted of one multiple choice question, with 4 distinct answer choices. The students were instructed that only one of the choices is correct for any given question. The 36 questions covered general topics of computer science, such as hardware devices, software concepts, operating system concepts, measurement units in Computer Science, general networking concepts, user safety and ergonomy concepts.

The main rationale for this experiment was the hypothesis already stated in [1] and in more details in [2] that Data Association Methods can be used in relevant data gathered from results of test papers, as follows. Association rules exists between test items such as if a student incorrectly solves a set of given test items, then there is a computable chance that the same student will incorrectly solve a different set of given test items.

The algorithm chosen for the experiment was APriori algorithm as described in [12], without the benefits given by the usage of hash tables for easier access to candidate itemsets. The confidence level used in the experiment was 75 percent and the minimum support level was set at 20 percent.

Each test paper was recorded as a data instance with an identifier alongside 36 relevant attributes. Each attribute had a correspondent question in the test paper. Considering the focus of interest is in tracking mistakes made, a question that had been mistakenly answered on the test has been coded with a value of 1, marking a mistake occured in the current data instance. In the same manner, a value of 0 marks no mistake occured in the current data instance for the respective question.

The goal is to use these data instances (records) over the first phase of APriori algorithm in order to check the main rationale, the grouping of mistakes (belonging in the same candidateset) to actually have a logical reasoning. This step is adapted from supervised learning techniques, where each candidateset with enough support would have to pass a human validation to see if the candidateset actually can be justified by the logic of the domain of the test.

6. Results of the Experiment

After adapting both the database structure to reduce the number of passes through the dataset and adjusting the minimum support and minimum confidence several candidate itemsets were generated, that had a very good support level.

Such candidate itemsets contained items that clearly belonged to the same general topic. Some of the generated candidate sets are presented in Table 5.

Other such candidate itemsets were also present but with lower support count. Out of each such candidate itemset, several association rules were formed, but the number of test papers analysed was insufficient to determine without doubt whether some test items were more important than others. As future work, an additional number of over a hundred test papers with the same 36 questions will be added to the existing 75 records.

The way the test items were grouped in the same candidate itemset clearly supports the main working hypothesis that mistaked test items do follow association rules, in this case based on the knowledge level of a distinct topic covered by the test.

Topic	**Questions - same candidate set**
Hardware Topic	Which of the following can improve a computer's performance? What is Hard Disk formatting used for? Which of the following devices is an input device? Which of the following devices is both an input and output device?
Software Topic	Which of the following programmes is a software application per se? Which of the following is a function of the operating system? What type of software controls resource allocation in a computer?
Network Topic	Which of the following is not a feature of online commerce? Which of the following is the main advantage of using a computer network? What is World Wide Web? Which of the following statements on the Internet is true?

TABLE 5. Sample of Generated Candidate Itemsets

7. Conclusions and Future Work

The experiment itself largely confirmed the working rationale first taken into consideration in [1]. Several false association had to be removed for not actually having a sustainability in the logic of the data itself. This confirms that a supervised learning approach is the correct way to analyse this particular problem.

The next logical step in terms of Future Work is the adaptation to be able to implement Fuzzy Association Rules and be able to compute the complexity level for a given set of data instances. This step needs to be adapted to supervised learning approach. A comparison between available data association rules is also a valid direction for study.

Also, having non crisp, continuous data values for attributes will open to new APriori adaptations to be taken into consideration, such as the Algorithm for Discovery of Arbitrary Length Ordinal Association (DOAR) presented in [7]. Moreover, having supervised learning data association rules already known, there is another direction of study in terms of matching new data instances to the already learned association rules and studying the outliers with a Fraud Detection approach in mind.

References

[1] Ban, T. - **"Concept Paper: Generating and Assessing Test Papers Complexity using Predictions in Evolutionary Algorithms"**, Knowledge Engineering: Principles and Techniques Conference (KEPT) 2009, Cluj-Napoca,

[2] Ban, T. - **"Using Predictions in Data Mining for Improving Students' Performance in Tackling Online Test Papers"**, Acta Universitatis Apulensis Special Issue, International Conference on Theory and Applications in Mathematics and Informatics (ICTAMI) 2009, Alba-Iulia

[3] Agrawal, R. & Imielinski, T. & Swami, A - **"Mining Association Rules between Sets of Items in Large Databases"**, Proceedings of the 1993 ACM SIGMOD international conference on Management of data, Washington D.C., May 26-28, pages 207 - 216

[4] Agrawal, R. & Srikant, R. - **"Fast Algorithms for Mining Association Rules"**, Proceedings of 20th VLDB Conference, Santiago, Chile, 1994, pages 487 - 499

[5] Birkhoff, G. - **"Lattice Theory"**, American Mathematic Society Colloq. Publ., Vol. 25, New York 1948, republished 1967

[6] Kotsiantis, S. & Kanellopoulos, D. - **"Association Rules Mining: A Recent Overview"**, GESTS International Transactions on Computer Science and Engineering, Vol.32 (1), 2006, pages 71 - 82

[7] Campan, A. & Serban, G. & Truta, T. M. & Marcus, A. - **"An Algorithm for the Discovery of Arbitrary Length Ordinal Association Rules"**, Proceedings of the 2006 International Conference on Data Mining, DMIN 2006, Las Vegas, Nevada, USA, June 26 - 29, 2006, CSERA Press 2006, pages 107 - 113

[8] Frentiu, M. & Pop, H. F. - **"A Study of Dependence of Software Attributes Using Data Analysis Techniques"**, Studia Universitatis Babes-Bolyai, Informatica, Volume XLVII, Number 2, 2002, pages 53 - 66

[9] Frentiu, M. & Pop, H. F. - **"Detecting Mistakes in Students Software Measurement Projects"**, Proceedings of Symposium "Colocviul Academic Clujean de Informatica", 2005, pages 105 - 111

[10] Pang-Ning, T. & Steinbach, M. & Kumar, V. - **"Introduction to Data Mining"**, Addison Wesley US Edition, 2005

[11] Sumathi, S. & Sivanandam, S.N. - **"Introduction to Data Mining and its Applications"'**, Studies in Computational Intelligence, Volume 29, 2006, pag 161 - 164

[12] Witten, I. & Frank, E. - **"Data Mining Practical Machine Learning Tools and Techniques"**, Morgan Kaufmann Publishers - Elsevier, 2005

[13] Webb, G. - **"Discovering Significant Rules"**, Proceedings of KDD'06 International Conference, Philadelphia, Pennsylvania, USA, August 20 - 23, 2006, pages 434 - 443

[14] Zadeh, L. A. - **"'Fuzzy Sets"**, Information and Control 8, 1965, pages 338 - 353

E-mail address: `tiberiu@cs.ubbcluj.ro`

Department of Computer Science, Babeş-Bolyai University, 1 M. Kogălniceanu St., 400084 Cluj-Napoca, Romania