# Reducing Cold Start Problems in Educational Recommender Systems

Stanislav Kuznetsov, Pavel Kordík, Tomáš Řehořek, Josef Dvořák and Petr Kroha
Faculty of Information Technology
Czech Technical University in Prague
Corresponding Email: kuznesta@fit.cvut.cz

***Abstract*—Educational data can help us to personalise university information systems. In this paper, we show how educational data can be used to improve the performance of interaction-based recommender systems. Educational data is transformed to student profiles helping to prevent cold start problems when recommending projects to students with few user interactions. Our results show that our hybrid interaction based recommender boosted by educational profiles significantly outperforms best-seller recommendation, which is a mainstream recommendation method for cold start users.**

***Keywords*—recommender systems, educational data mining, ontology, cold start problems**

## I. Introduction

Education data, especially student evaluations are often only stored and not utilised because it is hard to generalise information contained and use it in a way that is beneficial. It happens because there are not processes available that transform the data to non-confidential form and still preserve their value. Educational data when efficiently transformed, can be used to personalise university information system, detect and predict problems, recommend projects or students with required skills.

Our approach has solved this problem by a knowledge transformation from an educational space into an ontology-based space that we used to generate specialisation profiles of our students, lecturer, etc. In other words, we are converting knowledge from an educational space (courses, lectures, marks, etc.) to a professional skills space. The new representation of knowledge can help us to move knowledge to the next, more abstract level and remove the informational noise.

This research focuses on recommender systems and ontologies. It describes the basic characteristics of recommender systems (from now on referred to as RS) and a methodology for their comparison. Ontologies and benefits resulting from their use in the field of RS are also introduced. We showed, how extracted ontologies can improve the performance of recommender systems that we used for student project matchmaking and how to solved the cold start problems by profile-based recommendation and outperformed best seller recommendation.

### A. Related work

Most recommender systems rely on historical data of user interactions (collaborative filtering approaches) or attributes of users or items. These attributes can be organized in ontologies. There are approaches using ontology either as the main tool for recommending or as a part of the recommending process. Attributes or ontologies can help mostly in case of cold start problem when user/item do not have historical interactions.

Procópio [1] described several studies involving ontology based recommender systems with about the division of recommender systems on the above mentioned groups, Collaborative (outlined in section IV), Content-Based and Hybrid System. Furthermore, he presents a table comparing the individual studies with each other. Since there are many studies in this area, we choose only those that are semantically close to our work.

In [2], authors describe how to create a framework that provides personalised Internet advertising in the form of a service. First, advertising categories are created according to commercial categories appearing on the Yahoo portal. Subsequently, keywords are extracted from the texts of adverts. Incorporating keywords into a category is determined using the TF-IDF [1] analysis. Subsequently, the ontology of advertising words is built. NGD [2] is used to derive a feature vector of each advert. Logistic regression is then used to describe the preferences of each user; the responses on an already displayed advert will be used to do so. Finally, a recommended value is calculated by whether a user has seen the advert or not.

Biperdia [3] is an ontology that is used to support a search engine. It contains more than 1.6M pairs, attributes, and entities. The ontology extends Freebase [3] database by an extract of syntactic analysis of user queries typed during a user search. For each attribute were saved a set of synonyms and word patterns where found the synonyms. This approach makes it possible to describe the attributes of a much broader context. We build our ontology as proof of concept. The next step of our research will be extending our attributes with additional information.

Zhou [4] focuses on the definition of human skills in mechanical design. It proposes a metadata set of skills such as the level of significance, experience, description, know-how, and multimedia. The ontology is generated by using metadata and their connections (component and association). The ontology meets all standards of the Semantic Web and can be applied in both recruiting and e-learning education. We also use some metadata (subject description) for generating an ontology, however at the present, we are doing it manually. In the future, we try to use a text mining to automate the process.

Yang [5] describes the development of customised recommender systems based on domain ontology. The aim is to create user ontologies of interest using a mapping of user preferences on the domain ontology. After creating the

[1] Term Frequency and Inverted Domain Frequency
[2] Normalised Google Distance
[3] https://www.freebase.com/

978-1-5090-0620-5/16/$31.00 

ontology, the authors use kNNs [4] algorithms to generate recommendations. We use a particular version of kNN algorithms described in Section IV. This algorithm seems to be a very promising in case of users preferences.

Hexin [6] describes the Framework for recruiting new people into jobs where they seek to increase the likelihood of finding a suitable candidate for the position using the ontology of skills and reciprocally, it helps job seekers to find suitable employment. Ontology is composed of nodes having weights in the interval 0 to 1 between them. If the relationship between the nodes is defined as is-a, the weight is set to 1; if the relationship is defined as part-of, the weights are adjusted manually. For each industry is created a special sub-ontology. We try to recommend the students for positions in the project correctly as well as try to offer the projects to the right students.

Our approach is to build an ontology of student skills on top of their study results and use it to assist interaction based recommender in the case of cold start users (with no historical interactions available). One of the biggest problems is that our data are too sparse to generate meaningful associations. Therefore, we have to obtain the ontology from external data.

## II. Background

### A. Ontology

Ontologies are traditionally regarded as a branch of philosophy known as metaphysics, where the ontologies are used to describe the existence systematically [7]. We could say that ontology is a formal specification of a dictionary with concepts and their mutual relationships associated with a single domain area. The ontologies represent a knowledge model with the semantic description and structure in computer science and information systems. Also contains a description logic. Ontology plays a very important role in areas such as knowledge engineering, object-oriented analysis, knowledge representation, informational retrieval and extraction, and database design [8].

Each domain area contains information specific only for itself. On the other hand, there is information that is shared with other areas. Therefore, the decision whether to use our own ontologies or the ones already existing in the RS poses a dilemma. If we decide to use our ontology, we will design it to fit our purposes, although we will not be able to utilise the full potential of the existing ontologies relevant to the area. If we decide to use an existing ontology, it brings us benefits in the form of smaller human effort in the initial draft, the quality of our ontology will be increased as we will be using the already tested elements, and finally, it will facilitate information sharing between other systems. However, to find a suitable, already existing ontology is hard because there is a large number of them, and not all are entirely relevant. Therefore, we decided to build our ontology at an early stage in this research and to find a suitable ontology for mapping subsequently.

One of the appropriate areas the use of ontologies is RS. Where we can improve the accuracy of the recommendation by using the knowledge that contained in the ontology. In the next section, we describe the main information about RS and their problems, introduce the methodology of their comparison and the performance metrics.

[4] k-nearest neighbour search algorithm

### B. Ontology-based recommender systems

Ontologies are the basis for ontology-based recommender systems. The main idea of the entire concept is to create a model that could describe the mapping between a domain area and users belonging to this area. In practice, this means to create user profiles which would include both implicit and explicit information [9]. It is hard to obtain the explicit information because the preferences of users can often change, and not all users are willing to report these changes. One possibility is to ask the users to fill out a questionnaire in the system; unfortunately, the users are not always want to do so. Therefore, it is required to let the system make use of the implicit information, i.e. the information that can be derived from the user interaction, and modify user preferences (profile) based on them.

### C. Cold start problems

We can say that the larger set of data about users and their interactions we have, the better the RS work [10]. The problem comes with having little of this data or none at all. Then, we have to decide by which attributes the recommendations will be carried out. If we cannot find a suitable solution, a paradox can occur lying in the fact that if we cannot carry out quality recommendations, users will stop using a given system, which means that we cannot improve the quality of recommendations with the decline in user activity. This nontrivial problem is called a cold-start problem [11]. We will look at two types of the problem in this section.

The new system cold-start problem is that there are no initial ratings by users, and hence no user profiles. In this situation, most recommender systems have no basis on which to recommend, hence perform very poorly.

The new-user cold-start problem is that the system has been running for a while, and a set of user profiles and ratings exist, but no information is available about the new user. Most recommender systems perform poorly in this situation, too [10].

Collaborative-based recommender systems cannot cope with this problem because they are unable to find a group of users with the same behaviour as there is no preceding user interaction. One of the solutions can be used Cross Domain Triadic Factorisation (CDTF) or CDTF implicit feedback (CDTF-IF). This model can better capture the triadic relation between users, items, and domains. Description of the models can found in [12]. Content-based and hybrid-based recommender systems are doing a little better because they need a smaller number of user interactions to find similarities.

None of these recommender systems can manage the absolute cold-start problem on its own as the content-based recommender system needs at least an initial set of interactions. The most common solution is to recommend the top N "best sellers." This solution is not optimal because it lost any kind of individuality. Another solution would be to use the RS in the ontology. Ontologies can provide us with another kind of information, based on which, the system may begin recommending without having to wait for interaction from users.

### D. Pareto optimality

Let $X$ be a set and $f_1, \ldots, f_n$ be real functions $f_i: X \to R$, values of which we want to maximise (also called the *objective* functions). We say that $x \in X$ *Pareto-dominates* $\hat{x} \in X$ *iff* the following two conditions are satisfied:

1) $\forall i \in \{1,\ldots,n\}: f_i(x) \geq f_i(\hat{x})$,
2) $\exists i \in \{1,\ldots,n\}: f_i(x) > f_i(\hat{x})$.

We also say that $x^* \in X$ is *Pareto-optimal* state *iff* it is not *Pareto-dominated* by any $x \in X$ different from $x^*$. We call *Pareto-optimal front* the set $X^*$ of all *Pareto-optimal* states. In our case of Recall-Coverage evaluation, $X$ is the set of available models (by example, a set of $k$-NN models of different $k$), and there are two objective functions present: $f_{rec}$ and $f_{cov}$. These stay for the *recall* and the *catalogue coverage* on a fixed data set being examined. One model, $m$, may be viewed as *Pareto-dominating* another model, $\hat{m}$, if at least one of the two following conditions is met:

- $f_{rec}(m) > f_{rec}(\hat{m}) \wedge f_{cov}(m) \geq f_{cov}(\hat{m})$, i.e. $m$ has higher recall than $\hat{m}$, but still possesses at least as high catalogue coverage as $\hat{m}$,
- $f_{cov}(m) > f_{cov}(\hat{m}) \wedge f_{rec}(m) \geq f_{rec}(\hat{m})$, i.e. $m$ has higher catalogue coverage than $\hat{m}$, but still possesses at least as high recall as $\hat{m}$.

If we accept the set $X$ of objective functions as the relevant measures of model quality, then $m$ may be considered strictly better then $\hat{m}$, and there is a good sense in using $m$ instead of $\hat{m}$.

## III. Our Approach

### A. Data preprocessing

Before we can build ontologies, we must collect data and carry out their preprocessing. The basic idea of the entire system is not only the resulting recommender algorithm but primarily the creation of the entire process from the acquisition of data through their processing to their subsequent use. This process should be maximally automated and should use only a minimal human intervention. It should be further divided into independent components so that we would always be able to replace any component. The process consists of extracting accreditation materials from a data warehouse and their automatic classification into a set of keywords, acronyms, and text. Text mining is performed next, thanks to which we will obtain proposed skills. Additionally, we created a website to collect information from teachers, where each teacher indicates what skills belong to a subject based on the proposed skills. Finally, this information will be written into the matrix from which the ontology is generated. Besides a list of skills that cover a given subject, we assigned a value to each skill (from 0 to 1), this is the level that the given subject covers a specific skill. This matrix represents the core of the entire system.

Example III-A shows the mapping of subjects with code MI-PDD [5] to skills.

- MI-PDD:
  - Datamining weight = 1.0
  - Mathematics weight = 0.2
  - Database weight = 0.4
  - ...

### B. OntoSkill tree

Our ontology represents a simplified skill tree inspired by the ACM tree. We simplified the tree by removing skills that are not taught by the faculty. We did not want the tree to be excessively deep (a maximum of 2 levels) so that clients (people without experience in the education system) can easily understand it. The higher levels of the tree contain skills that represent abstract levels of knowledge (centroid) and which are made up of more than one specific skill. Around each of these skills, we have a cluster of specific skills. The representation of a skill tree is shown in Figure 1. The advantage of this ontology is the fact that we can work with skills at the different levels of abstraction thanks to a precise hierarchy. With this feature, we can easily display only the first level of skills in the profiles for users, thus making the profiles well arranged. The biggest drawback is that the tree structure is not flexible and does not allow us to combine the skills directly, thus creating a problem with the fact that some skills can belong to multiple sub-trees (i.e. security, it belongs to many areas). Another disadvantage is the fact that the decision about the superiority of one skill over the other cannot be determined automatically, it must be decided by a specialist in this area, therefore, slowing down the development greatly.

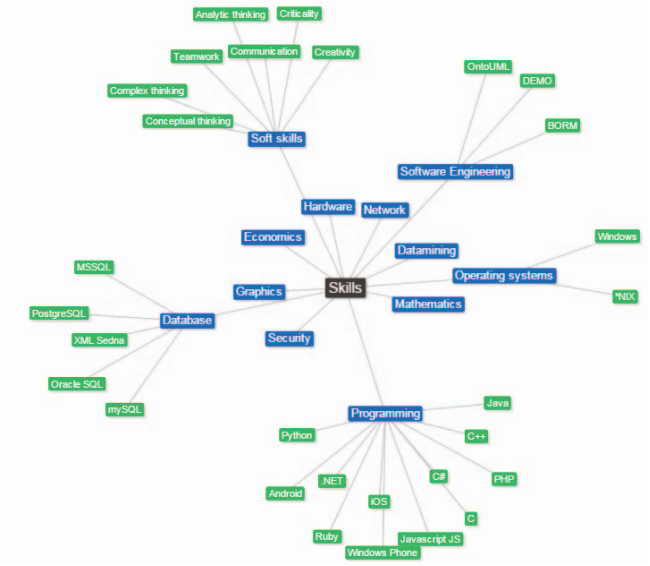


Fig. 1. Skills are represented by the tree, so higher level nodes generalise skill of leaf nodes.

### C. Computation of Students profile

First, we created a process that would be able to generate a student's profile based on the students study results. Then we compare these profiles against each other and recommend suitable students. The final process contains 3 phases.

*a) Score Calculation Phase:* In this phase, we work with the grades of the student using algorithms; we transfer them into absolute values which we call scores. All calculations made are done individually for each student, i.e. no comparison is made.

*b) Comparison and Normalisation Phase:* This phase, compare the students and then calculate the number of stars using a normalisation function. The faculty has two main groups of students, bachelor's degree students, and masters degree students; we call it programs. The algorithms compare two students that are in the same programs of study and the same year.

*c) Practical applications:* In the last phase of our process, we assign students to an assignment. Any assignment may contain a random number of positions, e.g. team researcher, database analyst, C++ programmer, etc. The client

[5] Data Preprocessing for master degree

may select the necessary amount and level of skills of every position. We are then able to calculate the most suitable students based on these values.

### D. Application domain - SSP Portal

The Student Cooperation Portal (SSP) is an information system that has been designed and developed by the Faculty of Information Technology, CTU, Prague. This portal provides a platform for industrial partners (referred to as clients) who input their projects into the system and for students who look through these projects and apply if interested.

Because a large number of students are involved in the portal, it is critical to recommend the "correct" students for positions in the project correctly. This approach also applies vice versa; it is necessary to offer the projects to the right students. The calculated skills also enable us to generate a student's CV and their profile specialisation automatically.

## IV. Recommendation models

In our experiments, we are using several algorithms. We briefly describe them in this section.

### A. k-Nearest Neighbours Algorithm

We are using user-based $k$-Nearest Neighbours algorithm with cosine similarity and voting. In [13], such an algorithm is referred to as the *Popularity-Stratified, Non-normalised Cosine Neighbourhood*, which we will, for simplicity, keep referring to as the *User-kNN* throughout the rest of this article. To rank items in a user's neighbourhood, the following formula is used:

$$\mathrm{rank}(u,i) = \frac{\sum_{\hat{u}\in N^k(u)} \mathrm{sim}(u,\hat{u}) \cdot (r_{\hat{u},i} - \overline{r_{\hat{u}}})}{\left(\sum_{u\in\mathcal{U}} (r_{\hat{u},i} - \overline{r_{\hat{u}}})\right)^{\beta}}, \qquad (1)$$

where $N^k(u)$ in the set of the $k$ nearest neighbours, $\mathcal{U}$ is the set of all the users in the database, and $\beta$ is the long-tail biasing parameter as proposed in [14].

### B. Association Rules

Other algorithms are Association Rules (AR) which allow us to construct pattern-driven recommendations. *AR* models are largely different from neighbourhood-based approaches, allowing us to explore the behaviour of multiple algorithms in the Recall-Coverage state space.

Given $(\mathcal{I}, \mathcal{U}, s_{min})$, we are able to find a set of association rules holding the minimal support. An association rule is an implication $X \Rightarrow Y$ such that $X \cup Y \in \mathcal{U}$, $X \neq \emptyset$, $Y \neq \emptyset$, $X \cap Y = \emptyset$. We say that association rule $X \Rightarrow Y$ holds the *minimal support iff*

$$\frac{|\{T \in \mathcal{U} \mid X \cup Y \in T\}|}{|\mathcal{U}|} \geq s_{min},$$

i.e. we require at least $s_{min} \cdot 100\ \%$ of users to contain $X \cup Y$ as relevant items in their rating history.

One of the most popular algorithms to solve the AR-mining task is the APRIORI algorithm also proposed in [15]. APRIORI first searches for frequent itemsets in a bottom-up manner. From the frequent itemsets, association rules are generated exhaustively. Since the original publication of APRIORI, many other algorithms have been proposed to mine association rules more efficiently, such as the FP-growth [16], and many others. In our experiments, we use a modified, lazy version of the APRIORI algorithm, which finds the set of rules at the time of recommendation, pruning the search space of frequent itemsets on a test user.

In our article, we use $\beta$ parameter to provide a smooth transition from the confidence to the lift based on this formula:

$$\mathit{lift}(X \Rightarrow Y) = \frac{\mathit{conf}(X \Rightarrow Y)}{\mathit{supp}(Y)^{\beta}}, \qquad (2)$$

we obtain the *confidence* for $\beta = 0.0$, and the *lift* for $\beta = 1.0$.

A more thorough description about *confidence* and *lift* can be found in [17] [15].

### C. Ontology bestseller

The last algorithm is an ontology bestseller. This algorithm is based on our ontology and has two parts. The first part is computed by the Euclidean distance between the vector of assignment skill and the vector of student skills (computed profile). If $\vec{v_1}$ is the vector of assignment skill and $\vec{v_2}$ is the vector of student skills then the Euclidean distance is given by the Pythagorean formula:

$$d(\vec{v_1}, \vec{v_2}) = \sqrt{\sum_{i=1}^{n} (\vec{v_1}_i - \vec{v_2}_i)^2} \qquad (3)$$

This distance is computed for all students and assignments in the system. The second parameter is the number of unique visitation of students from a web page with assignment description; we call it $visit(x)$. For final rank, we normalise both parameters and use this formula:

$$rank(u,a) = (1 - d_i)^{\gamma} \cdot visit(i)^{1-\gamma} \qquad (4)$$

Where $u$ is a user, $a$ is an assignment and $\gamma$ is the parameter designed for weight distribution between these two components. Note that we subtract the $d$ value from one, because we want to maximise the rank, but the Euclidean distance works in reversal. The more similar vectors are then the distance is less.

## V. Offline Experiments

In this section, we present the offline experiments on two data sets. All data sets are derived from SSP [6] database which contains approximately 1k of students, 300 assignments and more than 7k of interactions (assignment page visits). In our portal, we do not have heavy traffic for online experiments, so we decided to use offline experiments and analyse the behaviour of students over a longer period (one and half of year). All graphs which will be introduced have the x-value shifted to see the course of curves better.

### A. Metrics

There are several ways to simulate user behaviour in a system [11]. The most common method is to use historical data when some user interactions are hidden and then used to make a RS predict their value. Ideally, a time-stamp interaction is used, based on which, we can simulate the system behaviour in time. The most commonly used procedure for large data sets is one, in which we randomly select test users, randomly choose a time for these users, hide all the interactions of these users from this time, and try to recommend items for those users. The problem with this approach is that it needs to perform the entire process before each recommendation which is not very efficient regarding performance.

Since our RS always recommends Top-N items (N = 5) for each user, a leave-one-out cross-validation methodology [18]

[6] The Student Cooperation Portal

seems like as a good compromise for testing the RS. Within this method, a cross-validation is carried out across all users in the system.

For model comparisons, we use recall and coverage [7] (also called catalogue-coverage) metrics. We chose these metrics because the users prefer more diversified recommendation with greater coverage. That is the reason, why we optimise the both metrics. The exact description of the methodology and discussion of the selection of these metrics can be found in [18].

### B. Leave-one-out cross-validation methodology

We will construct an interaction matrix for all users of the RS and all items, where each row will be a user and each column an item. A single cell of the matrix will contain the number of interactions between an item and a user. If no interaction has taken place, the cell is filled out with zero. During each step of the cross-validation, we will assign 90% of the users to a training set and the remaining 10% to a test set. For each user, we will always choose the Top 5 items (inputs). The following procedure is divided into steps:

1) We will teach a model on training data for each cross-validation step.
2) We will hide user interaction from the system and let the Top 5 recommendations be generated for each user from the test set and each item (matrix cell).
3) If the current item is between the recommended items while finding out that the value in a cell is greater than zero [8], the recall value of a single test is one. Otherwise, it is zero.
4) After we go through all matrix cells, we will calculate the total cross-validation step recall by dividing the number of successful tests by the number of all tests (all matrix cells). The total coverage is thus the number of unique items to all recommended items.
5) Then, we move on to the next step of cross-validation, i.e. to point 1.
6) After completing the cross-validation, we can count the average recall and the average coverage. This procedure appears to be optimal since it allows us to calculate the average recall across all tested users as well as to solve the precision-recall dilemma described in [19].

### C. Data sets

There are two data sets. The first data set contains students data, which got into the portal and made exactly one interaction, we call them "Cold start data set". These students are new, so we do not have any information regarding their interactions. It means that we cannot use either the *User-kNN* nor the *Association rules* algorithms because of the cold start problem that described in Section II-C. In practice, this means, that if an algorithm fails to make a recommendation, instead of the empty set, the system returns the *Top N* best sellers. In our case, the *Top N* most visited assignments where *N* is the usual number of recommendations.

The second data set is "Regular data set". This data set contains all data and includes previous data set so that we can demonstrate the behaviour of all models described in Section IV. The Fig. 2 shown the frequency of regular data set.

[7] This metric shows whether a system can recommend variously, i.e. if it can offer other items than best sellers.

[8] If we have a system with a large number of interactions we can set a certain threshold, e.g. greater than 5

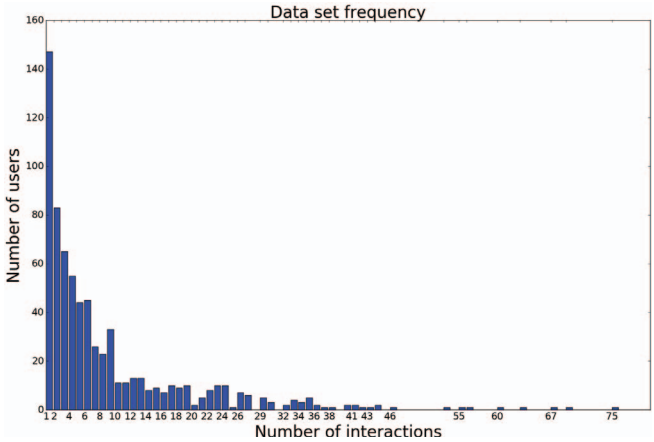


Fig. 2. The frequency of regular data set. Each value at x-axis represents the class with the number of interactions. At the y-axis, there are users belong to each class.

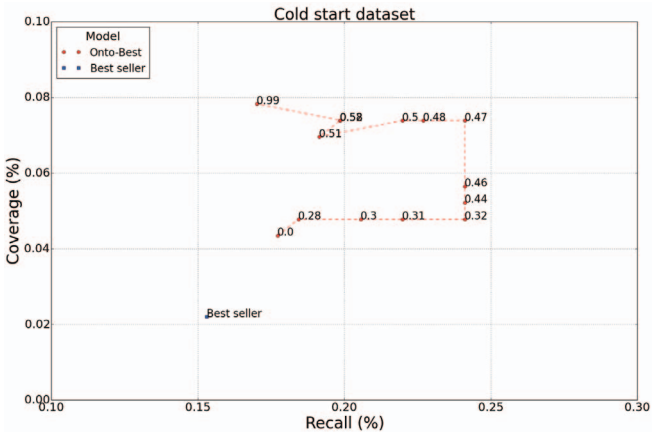


Fig. 3. Results for the *Onto-best* model and *Best seller* model on the cold start data set. The *Best seller* model has no parameter; it represents by a single dot. On the other hand, *Onto-best* model has $\gamma$ parameter; at the chart, it represents a curve which shows the behaviour of a model, depending on the $\gamma$ setting.

### D. Cold start data set results

As shown in Fig. 3, the *Best seller* model is *Pareto dominated* by *Onto-best* model. The highest recall and coverage is obtained for the *Onto-best* model with $\gamma$ = 0.47. That means that for the algorithm is important the both component, ontology weight, and most visited weight. The *Best seller* model represented by a dot in a figure has not got the parameters to influence model settings. The best results are summarised in Table I.

TABLE I. SUMMARISATION OF THE RESULTS FOR *Onto-best* AND *Best seller* MODELS AT COLD START DATA SET.

| model | recall | coverage |
|---|---|---|
| Onto-best | 24.11% | 7.39% |
| Best seller | 15.33% | 2.17% |

TABLE II. SUMMARY OF THE RESULTS FOR ALL MODELS AT REGULAR DATA SET.

| model | recall | coverage |
|---|---|---|
| User-kNN | 30.03% | 46.13% |
| Association Rules | 27.83% | 45.20% |
| Onto-best | 14.32% | 4.78% |
| Best seller | 13.7% | 2.2% |

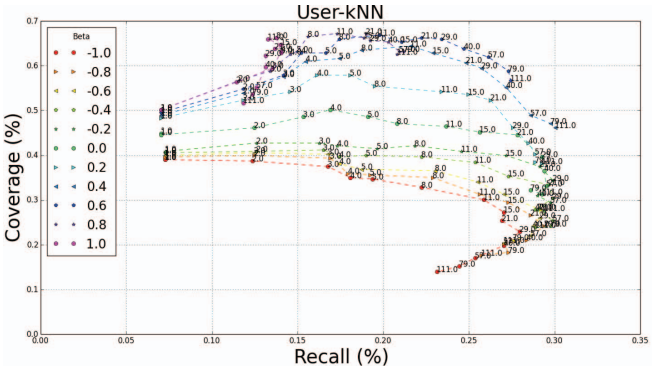


Fig. 4. Results for the *User-kNN* model on the regular data set. Each curve on the chart represents the setting of $\beta$ value. The values on curves are the settings of $k$ parameter.

### E. Regular data set results

As shown in Fig. 4, for the *User-kNN* model without the $\beta$ term, the highest recall is obtained for the 15-NN model. Higher numbers of neighbours deteriorate both recall and the coverage. In contrast, lowering the number of neighbours has an adverse impact on the recall due to over-fitting, but the catalogue coverage increases as the recommendations become more accurate for small, local groups of users. $\beta$ term add the plasticity for a model. For models with higher plasticity (low number if neighbours), $\beta$ significantly reduces the recall and increases the coverage. The recall and coverage are maximised for the 79-NN model with $\beta = 0.4$.

As shown in Fig. 5, for *AR* model without $\beta$ term, the highest recall is obtained for the $s_{min} = 0.0141$. The Lower value of $s_{min}$ has a negative impact on the recall; the higher value has a negative impact on the coverage. As with the previous algorithm, term $\beta$ provides the plasticity for the model. The recall and coverage are maximised for the $s_{min} = 0.0001$ with $\beta = 0.6$.

As shown in Fig. 6, for *Onto-best* model the higher $\gamma$ term, deteriorates recall in the same way as lower value. The golden mean seems to be a value from 0.33 to 0.47 where we have the higher recall and the higher coverage. It alludes to the fact that students are interested in both kinds of assignments, the most popular (best seller) one and the most appropriate one to their profile.

Fig. 7 shows the comparisons of all models in their best configuration. The *Pareto optimal* model is *User-kNN* model that dominates all models. Table II summarises the best results.

## VI. DISCUSSION

This article presents processes that transfer students final grades into scores and then the transformation of these scores into levels of skills (stars).

We introduced a total of three algorithms which described in our section IV. The measurement was done using parameters

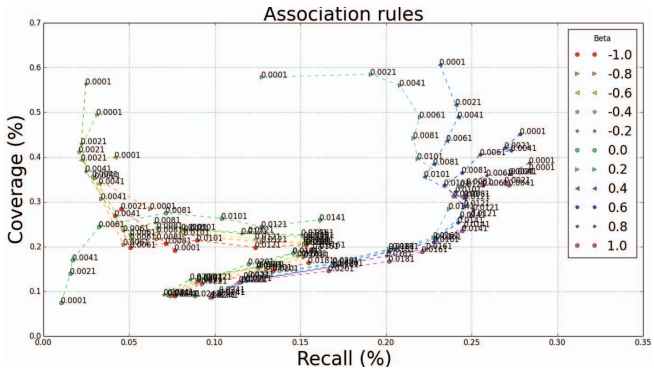


Fig. 5. Results for the *Association rules* model on the regular data set. Each curve on the chart represents the setting of $\beta$ value. The values on curves are the settings of $s_{min}$ parameter.

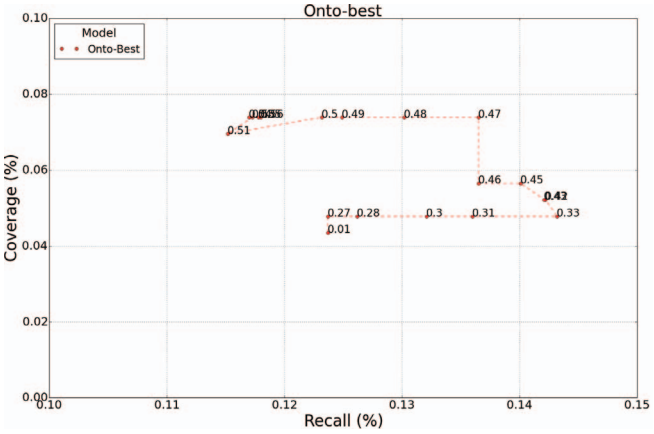


Fig. 6. Results for the *Onto-best* model on the regular data set. The curve on the graph represents the behaviour of a model depending on $\gamma$ parameter.

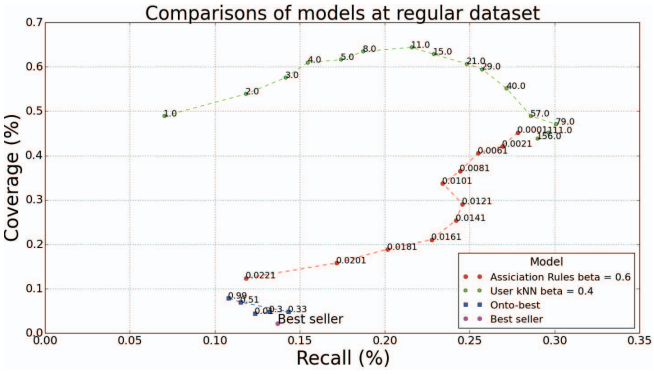


Fig. 7. Comparisons of all models at regular data set. The upper green curve is a behaviour of *User-kNN* model with fixed $\beta$ = 0.4 and various $k$. The curve in a middle represents *AR* model with fixed $\beta$ = 0.6 and various settings of $s_{min}$. The lower curve represents the results of *Onto-best* model depending on $\gamma$. On the bottom of a chart, there is a dot that represents *Best seller* model.

such as $k$ (number of nearest neighbours), $\beta$ (popularity biasing parameter), $s_{min}$ (minimal support for all algorithms), and $\gamma$ (ration between ontology weight and visited weight).

The biggest issue we have to solve is the sparsity of data or the small number of interaction. This problem is not related to the popularity of portals but with its use. Students use portal irregularly after they find an appropriate assignment they have no need to work in a portal. We supposed that we can improve it by adding the semantic layer to the description of students interactions. Ontology could represent this semantic layer. Experiments on the cold start data set show that the use of ontology and personal profile increase both recall and coverage.

The best algorithm in the regular data set is the *User-kNN* with parameter $k$ = 79 and $\beta$ = 0.4. This algorithm searches for similarities between users based on their interaction with the input. The reason for this algorithm being the best is probably that the system has more users, i.e. finding similarities between them is simpler than in the case of the input. The reason why the *AR* did not work the best is that the matrix between all students and all inputs is very sparse, making the pattern of behaviour hard to find.

## VII. Conclusion

This paper shows how student skills ontology can reduce cold start problem of educational recommender systems. Student profiles are created using their results in university courses.

Our task is to recommend students for specific industrial projects and positions.

We employed collaborative filtering algorithms and rule-based recommendation from historical user interactions in our systems. We solved the cold start problem by profile-based recommendation and showed that this approach outperforms bestseller recommendation. Extensive experiments with parameters of the recommender system show that best recommendations regarding maximising recall and coverage are those produced by neighbourhood based collaborative filtering. User profiles and skill ontology improve recommendations for cold start users.

## Acknowledgment

This research was partly supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS15/117/OHK3/1T/18 New data processing methods for data mining.

Computational resources were provided by the MetaCentrum under the program LM2010005 and the CERIT-SC under the program Centre CERIT Scientific Cloud, part of the Operational Program Research and Development for Innovations, Reg. no. CZ.1.05/3.2.00/08.0144.

## References

[1] F. Procopio de Paiva, J. Ferreira Costa, and C. Rodrigues Muniz Silva, "A hierarchical architecture for ontology-based recommender systems," in *Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI CBIC), 2013 BRICS Congress on*, Sept 2013, pp. 362–367.

[2] W.-H. Hwang, Y.-S. Chen, and T.-M. Jiang, "Personalized internet advertisement recommendation service based on keyword similarity," in *Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual*, vol. 1, 2015, pp. 29–33.

[3] R. Gupta, A. Halevy, X. Wang, S. Whang, and F. Wu, "Biperpedia: An ontology for search applications," in *Proc. 40th Int'l Conf. on Very Large Data Bases (PVLDB)*, 2014.

[4] J. Zhou and K. Watanuki, "Skill ontology for mechanical design of learning contents," *Journal of Software*, vol. 7, no. 1, pp. 61–67, 2012.

[5] Q. Yang, J. Sun, Y. Li, and K. Cai, "Domain ontology-based personalized recommendation research," in *Future Computer and Communication (ICFCC), 2010 2nd International Conference on*, vol. 2, May 2010, pp. V2–247–V2–250.

[6] H. Lv and B. Zhu, "Skill ontology-based semantic model and its matching algorithm," in *Computer-Aided Industrial Design and Conceptual Design, 2006. CAIDCD '06. 7th International Conference on*, Nov 2006, pp. 1–4.

[7] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowl. Acquis.*, vol. 5, no. 2, pp. 199–220, Jun. 1993. [Online]. Available: http://dx.doi.org/10.1006/knac.1993.1008

[8] N. Guarino, *Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy*, 1st ed. Amsterdam, The Netherlands, The Netherlands: IOS Press, 1998.

[9] G. Jawaheer, M. Szomszor, and P. Kostkova, "Comparison of implicit and explicit feedback from an online music recommendation service," in *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, ser. HetRec '10. New York, NY, USA: ACM, 2010, pp. 47–51. [Online]. Available: http://doi.acm.org/10.1145/1869446.1869453

[10] S. E. Middleton, H. Alani, N. Shadbolt, and D. D. Roure, "Exploiting synergy between ontologies and recommender systems," in *Proceedings of the WWW2002 International Workshop on the Semantic Web, Hawaii, May 7, 2002*, 2002. [Online]. Available: http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-55/middleton.pdf

[11] G. Shani and A. Gunawardana, *Evaluating Recommendation Systems*. Springer, 2011, ch. 8. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=150492

[12] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and C. Zhu, "Personalized recommendation via cross-domain triadic factorization," in *Proceedings of the 22Nd International Conference on World Wide Web*, ser. WWW '13. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2013, pp. 595–606. [Online]. Available: http://dl.acm.org/citation.cfm?id=2488388.2488441

[13] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proceedings of the Fourth ACM Conference on Recommender Systems*, ser. RecSys '10. New York, NY, USA: ACM, 2010, pp. 39–46.

[14] H. Steck, "Item popularity and recommendation accuracy," in *Proceedings of the Fifth ACM Conference on Recommender Systems*, ser. RecSys '11. New York, NY, USA: ACM, 2011, pp. 125–132.

[15] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 20th International Conference on Very Large Data Bases*, ser. VLDB '94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, pp. 487–499.

[16] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," *SIGMOD Rec.*, vol. 29, no. 2, pp. 1–12, May 2000.

[17] P. J. Azevedo and A. M. Jorge, "Comparing rule measures for predictive association rules," in *Proceedings of the 18th European Conference on Machine Learning*, ser. ECML '07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 510–517.

[18] E. Campochiaro, R. Casatta, P. Cremonesi, and R. Turrin, "Do metrics make recommender algorithms?" in *Advanced Information Networking and Applications Workshops, 2009. WAINA '09. International Conference on*, May 2009, pp. 648–653.

[19] H. Steck, "Training and testing of recommender systems on data missing not at random," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '10. New York, NY, USA: ACM, 2010, pp. 713–722. [Online]. Available: http://doi.acm.org/10.1145/1835804.1835895