



# Initialization of Matrix Factorization Methods for University Course Recommendations Using SimRank Similarities

Alisa Krstova<sup>✉</sup>, Bozhidar Stevanoski<sup>✉</sup>, Marija Mihova,  
and Vangel V. Ajanovski

Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University,  
Skopje, Macedonia

{krstova.alisa,bozhidar.stevanoski}@students.finki.ukim.mk  
{marija.mihova,vangel.ajanovski}@finki.ukim.mk

**Abstract.** The accurate estimation of students' grades in prospective courses is important as it can support the procedure of making an informed choice concerning the selection of next semester courses. As a consequence, the process of creating personal academic pathways is facilitated. This paper provides a comparison of several models for future course grade prediction based on three matrix factorization methods. We attempt to improve the existing techniques by combining matrix factorization with prior knowledge about the similarity between students and courses calculated using the SimRank algorithm. The evaluation of the proposed models is conducted on an internal dataset of anonymized student record data.

**Keywords:** Course recommendation engine  
Study plan development · Collaborative filtering · Matrix factorization

## 1 Introduction

With the rapid development of information technology, data-driven decision making in higher education has become a global trend aiming to support universities to meet both external standards, as well as internal self-evaluation and improvement requirements. The latter point includes, among other things, improving student retention and success rates, increasing motivation and overall satisfaction during the course of their studies. An indispensable aspect of this process is the careful collection, organization and analysis of educational data, which can be generated by different sources including student data, teacher data and data gathered from the process of teaching and assessment. Higher education is getting close to a time when personalization of study plans and career paths will become a common practice. Rather than using the “one-size-fits-all” approach when constructing a study program and requiring all students to enroll

same or similar subjects, universities begin to turn to building systems that would provide relevant, accurate course recommendations and corresponding grade predictions that are specifically tailored to each individual student. The vast amount of student and teacher-related data provides a basis for the development of intelligent systems that model the prediction of the final grades and systems that allow students to customize their degree plans to better match their career goals, personal interests and predispositions.

As more and more students choose to pursue a degree in higher education, the universities start to face the problem of having a large number of students with a wide range of abilities, skills, interests and potential, attempting to make an informed choice when choosing their elective courses. The level of freedom of choice has increased significantly over the last decade, and at the moment it amounts to 50% of the credits needed to successfully finish the undergraduate studies [9].

The lack of official guidance within the process of semester enrollment and course selection has led to a situation in which students rely predominantly on the word-of-mouth of their colleagues. Such recommendations are usually biased and do not take into account the student's personal abilities and inclinations. Motivated by these observations, our goal is to develop a prototype of a system that will assist university students in making an optimal choice when it comes to elective courses. We seek to analyze and use existing methods which have already been proven to result in accurate recommendations and predictions and explore possible modifications to the well-established algorithms. We aim to provide accurate grade predictions and recommendations over a wider set of electives best suited for the student in question.

The paper is organized as follows. Section 2 contains literature review of related work done in the past by other authors. Section 3 describes the dataset that was used in the experiments, as well as a short clarification on the data acquisition process. Section 4 defines the problem addressed in this paper. In Sect. 5 we make a short overview of the well-known algorithms: Probabilistic Matrix Factorization (PMF) [5], Bayesian Probabilistic Matrix Factorization (BPMF) [10] Alternating Least Squares (ALS) [13] and SimRank [4], and propose a novel approach for the initialization of the first three using the last one. We shortly explain the metrics used to evaluate the performance of the algorithms, and our approach. Section 6 compares the performance of the base versions of PMF, BPMF and ALS with each other, and with the enhanced versions using the SimRank initialization. Finally, Sect. 7 concludes the paper.

## 2 Related Work

There have been extensive academic research efforts and numerous industrial implementations of recommender systems in the past. Since predicting course grades and recommending subjects differs significantly from recommending music or movies, we will focus on reviewing the work most relevant to our context. Several authors have taken a similar approach by identifying groups of

similar students when predicting a grade for a course [7, 14]. These implementations also rely on neighborhood-based collaborative filtering methods. The past grades of the student's colleagues are used to make an estimate of the grade that might be obtained in a hypothetical enrollment of the course by using some similarity-weighted aggregation function.

Several course grade prediction models are proposed in [3, 6, 8], that use methods based on sparse linear models and low-rank matrix factorization. When using such an approach, the students' success in the past courses plays a deciding role. The factorization matrix is built so that rows represent students, columns represent the available courses and each matrix element stores the grade that the  $i^{th}$  student obtained in the  $j^{th}$  course. A missing value signals that the respective student has not yet enrolled/passed the respective course. As will be explained in the following sections, we will use this approach when describing the specifics of our own data.

As the algorithm described in [13] resolves scalability and handles sparseness of the data which is a major issue in recommender systems, and as the probabilistic techniques in [5, 10] considerably outperform standard matrix factorization as well as other approaches, essentially motivated by their performances in the context of recommending university courses in [9], we chose to work with these methods and tried to improve them.

The rationale of using grade prediction for future course enrollments is not in the sheer possibility to accurately predict the students' future, but in the opportunities that open in the processes of guiding and advising students. Course recommender systems have been successfully integrated in educational dashboards and learning analytics systems, such as the one described in [1]. The described Virtual Student Adviser enables the students to explore different study programs and guide them on their own personalized academic path – annotating risky mandatory courses and paths on one hand, and recommended elective courses among the pool of freely selectable options on the other hand.

From the conducted literature review, it can be concluded that the problems of grade prediction and course recommendations have been tackled by various approaches. These processes are beneficial not only for increasing the satisfaction of the student after choosing a certain course, but also for increasing her chances of success during her studies. Grade prediction can also be used as a tool to perform a hypothetical (what-if) analysis on the impact of one selection from a list of courses against another, and assessing the courses which may result in risk of failure. This would provide the student with an indicator of where to direct her efforts and time.

However, most of the reviewed systems suffer from the well-known problem which often arises in constructing recommender systems – the cold start problem. There are always students with very few grades, i.e. those who are currently in the first year of undergraduate studies. This results in low amount of information on the preferences and performance of the students based upon which we would like to make predictions. Therefore, special attention must be paid in order to remedy this issue.

### 3 Dataset

The experiments were conducted on real, fully anonymized records of student course enrollments and grades at the authors' institution. The dataset includes data course enrollments in the period from 2011 to 2016. The process of anonymization was performed by administrative staff, prior to the process of acquisition of data for research purposes. As researchers, we did not have any access to the full official data records, but only to already anonymized replicas. For the purpose of this research, we acquired records in the form of triplets (student, course, grade) – whereas the student was represented by an anonymous identifier, the course was represented by the real name of the course, and the grade was the grade achieved by the student (5 meaning a failed course, 6 being the lowest passing grade and 10 being the highest grade, null meaning a course that was enrolled but was not yet graded by the respective teachers).

We ran the experiment over a 25-day interval in 5 cycles, having 6 batches of data records. The first batch contained data from the previous academic years, while the other subsequent batches add records that were newly input during the respective cycles. The size of the dataset for each cycle was divided into a training and test dataset, as shown in Table 1.

**Table 1.** Number of training and test records for each cycle of the dataset

	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5
Training records	58915	59492	60039	61544	62899
Test records	595	541	968	1261	1127

The time frame of data acquisition was chosen to be the final exams period, immediately prior to the deadline for the process of course enrollment, the reason being two-fold:

1. The finals are typically the last responsibility the students should pass in order to get their grade.
2. The end of the exam period typically concurs with the opening date of the period when the students choose the courses they would like to enroll in for the next semester. Hence, this is the period when course recommendations are most needed and sought for.

As an illustration for the outlook of the dataset, Table 2 contains all instances from the first cycle for the student 1773012. It is essential to note that the ID 1773012, as all the other IDs in our dataset, is not a real identifier. This ID does not correspond to an existing student identifier and the example we give for illustration purposes is not based on any student's real data. The complete dataset is an extension of Table 2, so that it contains similar records for all the students.

**Table 2.** Illustrative dataset records from the first cycle

Student	Course	Grade
1773012	Business and management	10
1773012	Introduction to Internet	9
1773012	Introduction to Informatics	7
1773012	Discrete mathematics 1	7
1773012	Marketing	8
1773012	Object-oriented programming	10
1773012	Introduction to Web design	9
1773012	Professional skills	6
1773012	Structured programming	7

## 4 Problem Formulation

From the student-course-grade relation, a grade matrix  $G$  can be constructed. Suppose we represent the courses and the students with integer IDs of the intervals  $[1, M]$  and  $[1, N]$ , where  $M$  and  $N$  are the number of courses and students respectively. The grade matrix  $G$  will be such that the element in the  $i^{th}$  row and  $j^{th}$  column represents the grade the student  $i$  obtained in the course  $j$ .

Since the number of available courses is much larger than the number of courses the students are required to pass in order to graduate, and some students are in their first years of studying,  $G$  will be sparse. In this paper, we aim to predict the missing elements of the matrix, or in other words, we predict the performance of each student in the courses she has not already been enrolled in, and based on such obtained performances, we recommend her university courses.

Having the entries from the train set represented with  $G$ , for evaluation purposes, we compare the predicted grades with the real ones from the test set. The predictions are made using commonly utilized methods, however, here we improve them by introducing a novel approach in their initialization.

## 5 Methodology

### 5.1 Matrix Factorization Methods

Matrix factorization methods have gained popularity in recent years [12] due to their good predictive accuracy. Techniques based on matrix factorization are effective because they allow the discovery of the latent features underlying the interactions between users and items. This idea can be mapped to the concept of making course grade predictions by observing students as users and their courses as items. The algorithm will be applied in order to predict the missing grades of students interested in a particular course, and afterwards to make personalized course recommendations based on the computed grade predictions.

As the name suggests, matrix factorization and its variation intends to find two low-dimensional matrices  $S_{D \times N}$  and  $C_{D \times M}$  that factor  $G$ , where  $G$  is the aforementioned student-course-grade matrix and  $D$  is a positive integer representing the number of latent features to be considered. In other words, the product of  $S^T$  and  $C$  should approximate  $G$ .

A convenient interpretation of the matrices  $S$  and  $C$  is to consider that each student and each course is mapped to a  $D$  dimensional latent feature space, and  $S$  and  $C$  have the corresponding feature vectors of the students and courses respectively. Having these vectors, a grade prediction for student  $i$  in the course  $j$  is just a dot product of their corresponding vectors, i.e.

$$g_{i,j} = s_i^T c_j \quad (1)$$

where  $g_{i,j}$  is the predicted grade and  $s_i$ ,  $c_j$  are the  $i^{th}$  and  $j^{th}$  columns in  $S$  and  $C$  respectively.

To learn the factor vectors  $(s_i, c_j)$ , the algorithm strives to minimize the regularized squared error on the set of known grades:

$$\min_{(s,c) \in K} (g_{i,j} - s_i^T c_j)^2 + \lambda(\|s_i\|^2 + \|c_j\|^2) \quad (2)$$

where  $K$  is the set of student-course  $(s, c)$  pairs for which the grade is known from the training set and  $\lambda$  is a regularization term.

**Alternating Least Squares (ALS).** As described in the previous section, our goal is to minimize the loss function. Derivatives are an obvious method for minimizing functions in general, so several derivative-based methods have been developed. One of the most popular approaches is the Alternating Least Squares (ALS) algorithm [13].

ALS minimization starts with holding one set of latent vectors constant (for example, the student vectors), then taking the derivative of the loss function with respect to the other set of vectors (the course vectors), setting the derivative equal to zero and solving the resulting equation for the non-constant vectors. Afterwards, these new vectors are held constant and the derivative of the loss function with respect to the previously constant vectors is taken. The steps are repeated until convergence is achieved. This can be formulated mathematically as:

$$\frac{\partial L}{\partial x_s} = -2 \sum_i (g_{s,c} - x_s^T y_c) y_c^T + 2\lambda x_s^T \quad (3)$$

$$0 = -(g_s - x_s^T Y^T) Y + \lambda x_s^T \quad (4)$$

$$x_s^T (Y^T Y + \lambda_x I) = g_s Y \quad (5)$$

$$x_s^T = g_s Y (Y^T Y + \lambda_x I)^{-1} \quad (6)$$

Here, the course items  $y_c$  are held constant and the derivative is taken with respect to the student vectors  $x_s$ . The symbol  $Y$  refers to a matrix consisting of

all student row vectors. The row vector  $g_s$  contains all course grades for student  $s$  taken from the grades matrix and  $g_{s,c}$  is the grade obtained by student  $s$  in course  $c$ . Finally,  $I$  represents the identity matrix.

**Probabilistic Matrix Factorization (PMF).** Minimization of the loss function described in (2) appears as a goal if we take the probabilistic way of approximating the grade matrix. One simple algorithm is the Probabilistic Matrix Factorization (PMF) [5] that outperforms many others, as we will see in the results' section. PMF assumes normal distribution for the user and item latent features as priors, as well as for the result, such that the grade predicted in (1) is taken as the mean for the distribution.

An important note to take from this algorithm is that it scales linearly with the number of records, and therefore, it outperforms the other similar methods in execution time.

**Bayesian Probabilistic Matrix Factorization (BPMF).** This method takes a Bayesian approach to the problem which includes integrating out the model parameters, i.e. it gives a fully Bayesian treatment to the already described Probabilistic Matrix Factorization (PMF) model. What makes Bayesian Probabilistic Matrix Factorization (BPMF) different is that it uses Markov chain Monte Carlo (MCMC) methods for approximate inference [10]. As in PMF, the prior distributions over the students and courses features are assumed to be Gaussian. Inference is, however, achieved through the Gibbs sampling algorithm that iterates through the latent variables and samples each from its distribution conditional on the current values of the rest of the variables. This algorithm is normally used when conditional distributions are easy to sample from – due to the use of conjugate priors for the parameters and hyper-parameters in the BPMF, the Gibbs algorithm is very well applicable. It has been demonstrated that the BPMF model may outperform in specific cases the classical PMF approach on user-rating datasets, such as the large Netflix dataset [10].

## 5.2 SimRank

The previous sections demonstrate three different variations of the matrix factorization algorithm which use different ways to initialize the matrix. The majority of the approaches used in literature use random initialization or initialization based on taking the averages of the respective rows or columns. We seek to explore the effects of adding a semantic component to matrix initialization, i.e. to augment the algorithm by providing it with some initial similarities between courses and students. However, in order to do this, we must first define course similarity. Intuitively,

- Two courses are similar if similar students are enrolled in both of them, and
- Two students are similar if they are enrolled in similar courses.

These definitions are cyclic and the intuition behind them is comparable with the one behind Google's PageRank algorithm, which ranks websites based

on their importance. Hence, we turn to a somewhat related algorithm for computing similarities between items and users (in our case courses and students) – SimRank [4].

In order to use the algorithm, we form a directed bipartite graph whose nodes represent the students and the courses from the dataset described in Sect. 3. There is an edge from student  $s$  to course  $c$ , if and only if  $s$  has enrolled  $c$ .

A difference exists between the similarities of the nodes in our graph – the similarity between students  $s_1$  and  $s_2$ , and the similarity between courses  $c_1$  and  $c_2$  can be calculated with the following recursive relations:

$$s(s_1, s_2) = \frac{C_1}{|O(s_1)||O(s_2)|} \sum_{i=1}^{|O(s_1)|} \sum_{j=1}^{|O(s_2)|} s(O_i(s_1), O_j(s_2)) \quad (7)$$

$$s(c_1, c_2) = \frac{C_2}{|I(c_1)||I(c_2)|} \sum_{i=1}^{|I(c_1)|} \sum_{j=1}^{|I(c_2)|} s(I_i(c_1), I_j(c_2)) \quad (8)$$

$I(v)$  and  $O(v)$  are the set of in-neighbors and out-neighbors of node  $v$ , and  $I_i(v)$  and  $O_i(v)$  is an individual in-neighbor and out-neighbor respectively.  $C_1$  and  $C_2$  are decay factors and have values in the range  $(0, 1)$ , As originally proposed in [4],  $C_1$  and  $C_2$  are taken to be 0.8.

### 5.3 Matrix Factorization with SimRank Weights Initialization

As we discussed previously, the initial features for both users and items are chosen randomly, and then, they are tweaked with different methods, depending on the algorithm used. Under this assumption, at the beginning, we do not differentiate neither the students, nor the courses.

Having only the student-course-grade records, and no other additional information, limits the possibilities for initialization. However, they are enough to compute the SimRank results, that can be later used in the initialization of the matrices.

We propose a method of initialization of latent features using the previously computed SimRank similarities. For each of the above matrix factorization algorithms we take the resultant student and course latent features,  $S'_0$  and  $C'_0$ , after one run of the algorithm with the usual random initialization. Let us denote the student-student similarity matrix with  $S_{sim}$  and the course-course one with  $C_{sim}$ . In them, the  $(i, j)$  entry shows the similarity of  $i^{th}$  and  $j^{th}$  student or course. The new student and course latent features  $S_1$  and  $C_1$  are computed as weighted arithmetic mean of  $S_0$  and  $C_0$  with weights  $S_{sim}$  and  $C_{sim}$ , or

$$S_1 = \frac{S'_0 \cdot S_{sim}}{S_{avg}} \quad (9)$$

$$C_1 = \frac{C'_0 \cdot C_{sim}}{C_{avg}} \quad (10)$$



where  $S_{avg}$  is a  $D \times N$  matrix that as  $i^{th}$  column has the mean vector of the similarities for the  $i^{th}$  student.  $C_{avg}$  is the analogous  $D \times M$  matrix for courses. The fractions in (9) and (10) denote an element-wise matrix division.

One can easily notice that new initial latent features  $S_2$  and  $C_2$  can be constructed in a similar way using (9) and (10), i.e.  $S_2 = \frac{S'_t \cdot S_{sim}}{S_{avg}}$  and  $C_2 = \frac{C'_t \cdot C_{sim}}{C_{avg}}$ , using the matrices  $S'_t$  and  $C'_t$  that are resultant features after algorithm converged. In fact, the above discussion can be generalized for any non-negative integer  $t$  in the following manner

$$S_{t+1} = \frac{S'_t \cdot S_{sim}}{S_{avg}} \quad (11)$$

$$C_{t+1} = \frac{C'_t \cdot C_{sim}}{C_{avg}} \quad (12)$$

We terminate with such initializations once the latest is outperformed by a former one.

#### 5.4 Evaluation Metrics

As a method for testing the performance of our models, we use the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE) measures. These metrics can be defined mathematically as:

$$RMSE = \sqrt{\frac{1}{F} \sum_{g_{i,j} \in T} (g_{i,j} - \hat{g}_{i,j})^2} \quad (13)$$

$$MAE = \frac{\sum_{g_{i,j} \in T} |g_{i,j} - \hat{g}_{i,j}|}{F} \quad (14)$$

$$MAE_i = \frac{1}{H} \sum_{j=1}^H |g_{i,j} - \hat{g}_{i,j}| \quad (15)$$

Here,  $F$  is the number of total available grades,  $H$  is the number of grades the current student has obtained until the point of computation,  $T$  is the set of the test records,  $g_{i,j}$  refers to the actual grade, whereas  $\hat{g}_{i,j}$  to the predicted grade.

## 6 Results and Discussion

In this section we present the results of the performance evaluation conducted on the dataset described in Sect. 3. First, we begin by analyzing the results of the three matrix factorization techniques in reference to the previously explained performance metrics RMSE and MAE.

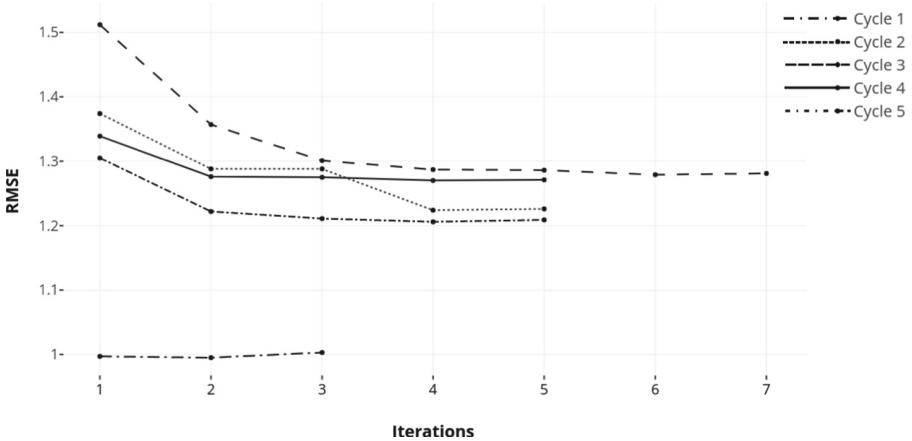
Table 3 summarizes the obtained RMSE and MAE scores on the dataset from the first cycle for ALS, BPMF and PMF with 7 different values for the parameter

$D$ , i.e. number of latent features (see Sect. 5.1). The comparative performances of the algorithms are similar to those on the data from the other cycles. The approach taken to obtain these results involves initializing the respective algorithms using their proposed random initialization - this provides us with initial insights on the performance of the different techniques and represents a reference point for our own SimRank-based initialization.

**Table 3.** Performance comparison of the three matrix factorization techniques

		D = 20	D = 30	D = 40	D = 50	D = 60	D = 70	D = 80
<b>ALS</b>	<i>RMSE</i>	1.637	1.509	1.476	1.438	1.447	1.430	1.430
	<i>MAE</i>	1.270	1.147	1.102	1.075	1.072	1.069	1.064
<b>BMPF</b>	<i>RMSE</i>	1.647	1.417	1.389	1.417	1.487	1.416	1.318
	<i>MAE</i>	1.236	1.081	1.040	1.088	1.126	1.086	1.007
<b>PMF</b>	<i>RMSE</i>	1.196	1.168	1.151	1.115	1.085	1.054	1.068
	<i>MAE</i>	0.837	0.808	0.796	0.768	0.736	0.721	0.737

For each algorithm, we take the value for  $D$  for which the smallest RMSE is observed, and take the student and course latent features for that  $D$ . Using them, we iteratively apply (11) and (12) as explained in Sect. 5.3. On the data from the first cycle, this initialization surpasses the simple random initialization of the algorithms. In particular, ALS notes RMSE of 1.014 in the first, and 1.011 in the second iteration, which is a significant improvement over the best value of 1.43 for the best case of  $D = 80$ . The RMSE for BMPF lowers to 1.17 after only 1 iteration.



**Fig. 1.** Improvements of the RMSE of PMF over all dataset cycles for the corresponding best value for  $D$ . Termination is done when an iteration yields worse results than the previous one.

**Table 4.** Confusion matrix of PMF for the first cycle, with  $D = 140$ 

		Predicted grades					
		5	6	7	8	9	10
Real grades	5	31	2	0	0	0	0
	6	9	124	113	10	2	0
	7	2	47	94	11	1	0
	8	2	14	35	13	9	0
	9	0	2	26	14	3	4
	10	0	0	5	5	13	4

PMF does not only outperform the others in the basic case, but also notices improvements for every cycle, and it continues yielding the best results after the SimRank initialization. Fig. 1 shows how RMSE drops in every iteration for PMF over all dataset cycles. Although in the first cycle, an improvement for all three algorithms is noted, BPMF and ALS algorithms have insignificant or even no benefit from this initialization in the other cycles. Therefore, we keep the focus on PMF’s results.

When considering an accurately predicted grade to be one that falls within deviation of  $\pm 1$  of the real one, using the SimRank initialization for PMF, we increase the accuracy of predicting grades from 70.1% to 77.5% for cycle 3. The best result we obtain is accuracy of **88.4%** for predicting grades belonging to cycle 1, and the confusion matrix of this case is shown in Table 4. It presents the number of records correctly predicted as belonging to the actual grade and the number of records that were predicted as not belonging to the actual grade. The x-axis presents the actual (real) grades, ranging from 5 to 10 and 6 being the lowest passing grade, whereas the y-axis contains the grades predicted by the SimRank-initialized version of PMF.

It is interesting to compare the proposed approach with other similar research agendas so as to perceive the benefits that may come with our decisions. The research presented in [2] employs techniques from collaborative filtering such as singular value decomposition (SVD) to predict grades and uses the Pearson correlation coefficient as a measure for the similarity between students. Several SVD initialization methods are evaluated and the effect of the size of the student-neighborhood taken into consideration is explored. This approach results in a MAE value of minimum 1.5, with some of the parameter configurations yielding an even higher value of up to 2.2. As can be seen from Table 3 and the discussion following it, the SimRank-initialized matrix factorization methods demonstrate

superior performance. Another sophisticated approach is described in [9], where the authors evaluate several matrix factorization techniques for grade prediction based on a historical dataset of over 10 years of student record data. The presented results show that around 80% of the predicted records fall in a deviation of  $\pm 1$  of the actual grade. The modification of the PMF algorithm presented in this research achieves an increase in accuracy of more than 8%.

It is important to note, however that the aforementioned comparisons have been done in different settings and tested on institution-specific datasets which might have less or more strict course programs and assessment procedures.

## 7 Conclusion and Future Work

In this paper, we presented a novel hybrid approach of initializing matrix factorization methods with student-course similarities obtained using the SimRank algorithm. We employ this approach in extending three well-known matrix factorization techniques (Alternating Least Squares, Probabilistic Matrix Factorization and Bayesian Probabilistic Matrix Factorization). The evaluation performed on a dataset of past student records showed that the Probabilistic Matrix Factorization (PMF) gives better results in all performance measures compared to ALS and BPMF. Furthermore, an overview of the algorithms' performance depending on the hyper-parameters was illustrated and discussed. We also demonstrate very promising results in accuracy achieved by the SimRank initialization method for PMF.

Our current research agenda includes re-evaluating the proposed method on an extended internal dataset that best reflects the current study programs offered at the authors' institution. This will be basis for the integration of the recommendation engine prototype in a user-friendly web application, whose purpose will be to allow the students to browse through course information, track their current study progress and display recommendations for prospective elective courses. Since the similarity matrices containing the SimRank scores are pre-computed and the matrix factorization methods are fast, the estimation of the most suitable courses for a particular student can be made almost real-time, i.e. on average for a single student the estimation lasts about 0.3 ms on a machine configuration with Intel<sup>®</sup> Core<sup>™</sup> i7-4720HQ CPU @ 2.6 GHz, 8 GB RAM on a 64-bit Ubuntu 16.04 LTS.

Incorporating user feedback into the system, i.e. letting the user evaluate the quality of the recommended course with respect to their interests and predispositions will further improve future predictions. We would also like to explore the effect of adding additional sources of data to our models, for example records about the respective course instructors, as well as some background information of the students' past education.

**Acknowledgments.** This work is a result within the project SISng (Study Information Systems of the Next Generation) [11], which is currently ongoing at the Faculty of Computer Science and Engineering in Skopje. The authors would also like to thank Ljupcho Rechkoski for the provided materials.

## References

1. Ajanovski, V.V.: Guided exploration of the domain space of study programs: recommenders in improving student awareness on the choices made during enrollment. In: Proceedings of the Joint Workshop on Interfaces and Human Decision Making for Recommender Systems (INTRS17), pp. 43–47. CEUR-WS, Como (2017)
2. Carballo, F.O.G.: Masters Courses Recommendation: Exploring Collaborative Filtering and Singular Value Decomposition with Student Profiling, Instituto Superior Tecnico, Lisboa (2014). <https://fenix.tecnico.ulisboa.pt/downloadFile/563345090413333/Thesis.pdf>
3. Hu, Q., Polyzou, A., Karypis, G., Rangwala, H.: Enriching course-specific regression models with content features for grade prediction. In: 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 504–513 (2017)
4. Jeh, G., Widom, J.: SimRank: a measure of structural-context similarity. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 538–543. ACM Press, Edmonton (2002)
5. Mnih, A., Salakhutdinov, R.: Probabilistic matrix factorization. In: Proceedings of the 20th International Conference on Neural Information Processing Systems, NIPS 2007, Vancouver, British Columbia, Canada, pp. 1257–1264 (2008)
6. Morsy, S., Karypis, G.: Cumulative knowledge-based regression models for next-term grade prediction. In: Proceedings of the 2017 SIAM International Conference on Data Mining, pp. 552–560. SIAM, Houston (2017)
7. O’Mahony, M.P., Smyth, B.: A recommender system for on-line course enrollment: an initial study. In: Proceedings of the 2007 ACM Conference on Recommender Systems, pp. 133–136. ACM, New York (2007)
8. Polyzou, A., Karypis, G.: Grade prediction with models specific to students and courses. *Int. J. Data Sci. Anal.* **2**(3–4), 159–171 (2016)
9. Rechkoski, L., Ajanovski, V.V., Mihova, M.: Evaluation of grade prediction using model-based collaborative filtering methods. In: 2018 IEEE Global Engineering Education Conference (EDUCON), pp. 1096–1103. IEEE, Tenerife (2018). <https://doi.org/10.1109/EDUCON.2018.8363352>
10. Salakhutdinov, R., Mnih, A.: Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In: Proceedings of the 25th International Conference on Machine Learning, pp. 880–887. ACM, New York (2008)
11. Student Information System of the Next Generation (2009/2018). <https://develop.finki.ukim.mk/projects/sisng>
12. Symeonidis, P., Zioupos, A.: Matrix and Tensor Factorization Techniques for Recommender Systems. Springer, Cham (2016). <https://doi.org/10.1007/978-3-319-41357-0>
13. Zhou, Y., Wilkinson, D., Schreiber, R., Pan, R.: Large-scale parallel collaborative filtering for the netflix prize. In: Fleischer, R., Xu, J. (eds.) AAIM 2008. LNCS, vol. 5034, pp. 337–348. Springer, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-68880-8\\_32](https://doi.org/10.1007/978-3-540-68880-8_32)
14. Zhuhadar, L., Nasraoui, O., Wyatt, R., Romero, E.: Multi-model ontology-based hybrid recommender system in e-learning domain. In: 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, pp. 91–95. IEEE, Milan (2009)