RECOMMENDING MASTERS’ COURSES: ENRICHING SINGULAR VALUE DECOMPOSITION WITH STUDENT PROFILING
Fábio Carballo, Cláudia Antunes
Instituto Superior Técnico – Universidade de Lisboa, Portugal

Abstract
After bachelor, many students strive to select the masters’ courses that are most likely to meet their interests. Although this decision may have a big impact on students’ motivation and future achievements, usually no support is offered to contest this problem. The use of recommendation systems to suggest items to users has well-known success in several domains, as in e-commerce and movies recommendation. Some of the most successful techniques use Singular Value Decomposition to reduce databases’ dimensionality and capture hidden latent factors, so as to produce high quality recommendations. In this paper, we propose the use of this technique, alongside with a contextual mapping to the educational paradigm, to capture relationships between courses grades, and recommend masters’ courses that are suitable to each student’s skills given their bachelor achievements. Our results show that using SVD to predict the masters’ courses grades has potential to serve as basis for the recommendation production. We also propose the use of as-soon-as-possible classifiers (ASAP classifiers) to enrich student profiling in our recommendation process so as to recommend masters courses when we don’t have all the information over students’ bachelor achievements.

Keywords: Singular Value Decomposition, Courses Recommendation, Student Profiling, Recommendation Systems

1 INTRODUCTION
The decision that students have to make on which master’s courses to enroll has way more impact than it looks: this choice can have a direct effect on their academic and personal goals and may define their future professional area. A bad choice of courses may demotivate a student, which can cause the student to drop out or to not take advantage of the fullness of its capabilities. Therefore, understanding students’ particularities is needed, so as to recommend courses that are not only interesting to them, but also adequate to their capabilities.

Educational Data Mining is concerned on developing methods that use several types of data originated from educational contexts aiming to improve the learning process. Although several interesting results have been achieved on student modeling and performance prediction, when it comes to recommend courses we are not able to find a lot of diversity on the existing solutions [1]. However, the good results that recommendation systems have on other areas are acclaimed [2] and, despite some few approaches, their (natural) application to this problem is not very explored. This is due to the fact that, the educational context has some given particularities that must be carefully addressed. Furthermore, the challenges and constraints on recommending courses are several, as students present the most different backgrounds.

Current solutions have a tendency to recommend courses based on its contents or potential interest to the students, not considering how those courses can affect the students’ overall academic performance [3][4][5][6]. Other solutions demand too much participation by the students [7][8][9]. Therefore, we propose the creation of a system that, with the minimal user-participation, automatically recommends masters’ courses that are not only interesting but that also add value to students’ academic achievements, given their bachelor path. To do it, we explore Singular Value Decomposition (SVD) [10][11] so as to capture hidden factors in the historical students grades and then identify the best courses to recommend. We compare this approach with some baselines in terms of the quality of the grades prediction and on how the recommendations affect students’ performance. We also take a look to the recommendation when we don’t have all the information on the students’ bachelor achievements, for example, when the student has some bachelor courses to finish, despite wanting to enroll on some masters’ courses. To approach this problem we will use ASAP classifiers [12], so as to use the students’ past grades on bachelor to predict the missing ones. Once the missing grades are predicted we can follow the regular recommendation process that we propose below.
The paper is organized as follows: in section 2, we introduce some important concepts and techniques in the area of recommendation systems, one of them being Singular Value Decomposition. We also overview existing literature about courses recommendation and the systems that address it. We then present our proposal in section 3, explaining how we perform recommendations in the educational context using SVD, taking into account how will the courses recommendation suit students’ skills. In this section we present the achieved results against some baseline predictors and recommenders. In Section 4 we present how we propose to ally the ASAP classifiers to our recommendation process and which were the achieved results. The paper ends with a discussion about the quality of our recommendations and the recommendation process.

2 BACKGROUND

Recommendation Systems have become an important multidisciplinary research field in the mid-1990’s, and many people have exhaustively dedicated large amounts of time and effort to the problem of recommending items from some fixed databases. A recommendation system on a common formulation can be seen as a set of tools and techniques used with the goal of providing suggestions of items to individuals who lack sufficient competence to evaluate the potentially overwhelming number of alternative items available. They have grown to become fundamental applications in electronic commerce (Amazon students [2]), information access, entertainment (Netflix [12]) and various types of services, providing suggestions that effectively prune large information spaces so that users are directed to those items that best meet their needs and preferences.

The majority of recommendation systems are based on Collaborative Filtering (CF). In pure CF, one identifies users whose tastes are similar to those of the target user and recommends items they have liked, never doing any analysis over the items at all. For this reason it is said that it uses the common principle of word of mouth. By using other users recommendations, it is possible to deal with any kind of content and receive items with dissimilar content to those seen in the past. Since other users’ feedback influences what is recommended, there is the potential to maintain effective performance given fewer ratings from any target user. However, this approach is not ridden of problems: a user profile for a new user will only be valid after making some few ratings, in order to enable the identification of the most similar users to the new user. Other weakness is that if a new item appears in the database there is no way it can be recommended to a user until more information about it is obtained through another user either rating it or specifying which other items it is similar to. If the number of users is small, relative to the volume of information in the system, then there is a danger of the coverage of ratings becoming very sparse. Another problem is that it will create poor recommendations for a user whose tastes are not according to the majority of the population.

Collaborative filtering techniques can be split in two categories: user-based and item-based. User-based systems employ techniques to find a set of neighbors that are the most similar users to the target user. As soon as the neighbors are defined, the system combines the preferences of neighbors to produce a prediction for the active user. These methods are popular due to their simplicity, interpretability and ability to produce accurate recommendations. But there are several disadvantages also: they depend on human ratings and their performance is poor when there is too much data sparsity, since it prevents the scalability of the solution. Item-based techniques first analyze the user-item matrix to identify relationships between different items, and then use the relations to indirectly compute recommendations for users. Despite the fact that these techniques handle the sparsity better than the user-based ones, they also require an expensive model construction [10][14][15].

After the Netflix challenge [13], there was a huge trend to use the so-called latent factor models, aiming to reveal the hidden latent features that somehow explain the observed ratings. One of the most applied techniques with these models is a matrix factorization technique - Singular Value Decomposition - due to its accuracy and scalability. This technique factors an \( m \times n \) matrix \( R \), into three matrices as in (1).

\[
R = U \times S \times V' \quad (1)
\]

where \( U \) and \( V \) are two orthogonal matrices of size \( m \times r \) and \( n \times r \) respectively, while \( r \) represents the rank of the matrix \( R \) (see Fig. 1 - a). Matrix \( S \) is a diagonal matrix, and its entries are stored in decreasing order of their magnitude. Each entry of matrix \( S \) represents a hidden feature and the stored value in it stands for the weight the feature has to the variance of the values on \( R \). The sum of all entries represents the total variance on matrix \( R \). SVD has many applications of particular interest, but it is especially useful as a way to find the best rank-k approximation, \( R_k \), to the matrix \( R \), such that
the Frobenius norm of $R - R_k$ is minimized. The Frobenius norm ($\| R - R_k \|_F$) is defined as simply the sum of squares of elements in $R - R_k$. To reduce the rank $r$ to $k$, where $k < r$, one should only use the first $k$ diagonal values of the matrix $S$ (the singular values), and then reduce both $U$ and $V$ accordingly. The result is the closest $k$-rank approximation $R_k = U_kS_kV_k^T$ (see Fig. 1 – b). This allows for decomposing the usual user-product matrix into a $k$-dimensional space where just the $k$ most relevant features are taken into account: the noise in the data is reduced, and this enables the production of better quality recommendations.

The usual idea, when using this technique on recommendation systems, is to use $R$ as a users-items matrix, where $m$ is the number of users and $n$ the number of items. The value of each cell holds the rating that a user has given to a certain item. The idea is that after the decomposition we can calculate both the users-features space and items-features space. In the users-features space, $U_i\sqrt{S_i}$ (let’s call it $P$), each row is a vector with the preference values of a user over the discovered features. On the other hand, in the items-features space, $\sqrt{S_k}V_k^T$, (let’s name this one $W$) each row is a vector that represents how the item is weighted in each feature.

Nevertheless, SVDs are not known for dealing well with sparse matrices, where there are a lot of missing values. Hence, as in usual recommendation problems there exists a lot of sparsity on the ratings matrices, some care must be taken in how to overcome this problem. Initially, some proposals tried to fill in the unknown values with normalized averages, but this is highly prone to overfitting.

A solution to this sparsity problem was found during the Netflix challenge, whose goal was to predict the ratings users would give to all movies. To predict each rating, Simon Funk [16] proposed to use a gradient descent algorithm in order to compute the best rank-$k$ matrix approximation using only the known ratings of $R$ (the user-movie matrix). This process follows the same idea as the one used on training neural networks. With the error in a prediction of user $i$ to movie $j$ being $(R_{ij} - R_{ij}^k)$, Funk’s approach takes the derivative of the square of the error with respect to $P_{ik}$ and then with respect to $W_{jk}$. Since $R$ is constant, and $R_k = P_kW^T$ (note that $P$ and $W$ contain the $S$ matrix, that usually results from the matrix decomposition), the updates for the user and item spaces, $P$ and $W$ then become (2) and (3), respectively:

\[
P_{ij}(t+1) = P_{ij}(t) + \text{learning\_rate} \times (R - R_k)_{ij} \times W_{ij}(t) \quad (2)
\]
\[
W_{ij}(t+1) = W_{ij}(t) + \text{learning\_rate} \times (R - R_k)_{ij} \times P_{ij}(t) \quad (3)
\]

In summary, the final solution of this learning problem is the combination of feature weights on both $P$ and $W$ such that the error in the approximation $R_k$ is minimized. This solution is determined iteratively, as the gradient of the error function is computed at each iteration step. You should also note that all features vectors have to be initialized with some values. Funk’s basic approach is to fill in the values with the global rating average with some random noise introduced.

### 2.1 Courses Recommendation

One of the most challenging problems faced by university students is to correctly choose which academic path to take, based on the available information. Thus, students need counseling on making adequate choices to complete their academic degrees with success. In the last years, course recommendation systems have been suggested as the tool that should be able to provide the guidance students need.
AACORN [8] is a case-based reasoning system that recommends courses to graduate students. The case-based reasoning component first retrieves the most similar student histories for the given inquiry. So, it requires a partial history of the courses followed by a student before it can provide useful recommendations. In order to determine the similarities between course histories, the system uses the edit distance metric. AACORN adapts a solution by building a list of courses found in the retrieved histories but not found in the target student data. Finally, it ranks the courses in the following way: each time a course appears in one of the retrieved students’ history it counts as one vote and these votes are weighted according to the distance of the retrieved student history to the target student (the less distant, the more the weight). Hence, the courses are ranked according to their total vote weight.

In 2011, Unelsrød [5] developed a recommendation system for course selection in higher education. The system has a component that uses a specialized user-based CF: it weights each user, based on their chosen degree major and whether or not the two users compared are friends. His expectations over the importance of the users friends were dashed, but there were improvements on the accuracy of the recommendations when focusing on users with common degree major. The system also has a content-based component that uses the courses features that a student has shown interest, in order to find the courses that suit best his preferences. Surpatean et al. [17] proposed a user-based collaborative system for recommending masters programs. The system uses the academic profiles of alumni students labeled by the masters program they have chosen. Then, given the academic profile of a student, the recommendation system returns the set of masters programs of the alumni students whose academic profiles are among k-closest in the training data. They used both ECTS (credits of each of the courses) and binary representation (if the student has enrolled or not on each course) on the students’ academic profile and the former representation was the one that has shown better results.

Vialardi et al. [18] presented a decision tree based recommendation system with the goal of creating awareness of the difficulty and amount of workload entailed by a chosen set of courses. They based their research on the generation of two domain variables: the student’s potential - that is calculated for each course and represented as the average of the grades a student has obtained in the prerequisites of that course, and courses’ difficulty - represented by the average of the grades obtained by students in the course. Lastly, during enrollment students choose a set of courses and the system forecasts if they will pass or fail each of the courses using the C4.5 algorithm.

In the set of approaches using only unsupervised techniques, we have Bendakir et al. [19] that created RARE, a course recommender system based on association rules. It starts by extracting association rules from previous course selection data that relates academic courses followed by former students. These rules are later used to infer recommendations. To improve the experience, RARE allows students to rate the recommendations, which may result in rules improvement by adding or removing courses from rules. To the best of our knowledge, there is no proposal to use SVD in courses recommendation, and so we present one in the next section.

3 SVD-BASED COURSES RECOMMENDATION

As we stated above, we aim for exploring SVD to recommend masters’ courses to students given only their bachelor’s courses grades. Hence, we must start of an historical record over triplets in the form of <Student, Course, Grade> into a structure that SVD can be used.

As seen in section 2, SVD makes use of a matrix $R$, which in general represents knowledge over the rating that a user gave to an item. For instance, in usual representations, users are placed in rows and items in columns, and each cell $R_{ij}$ in the matrix corresponds to the rating that the $i^{th}$ user attributes to the $j^{th}$ item.

As a first step to map our problem to the appliance of SVD we must transform our historical students’ grades record into a matrix $R$ that holds our knowledge over students’ capabilities in each course taken. Our proposal is a novel mapping where we can look to the grade a student achieved in a course as the usual rating in recommendation systems. More precisely, matrix $R$ will have students represented on rows and courses on columns, and each entry $R_{ij}$ will be filled with the grades obtained by the $i^{th}$ student on the $j^{th}$ course. In the cases that there are missing values, that is, in the courses where the student didn’t enroll, we will represent the grade with the zero value. This is a natural mapping, as we also want to recommend the items (courses) with predicted better ratings (grades), with the constraint of recommending only a subset of the courses, the masters’ courses.
For illustration purposes consider Fig. 2, which shows a student-courses matrix $R$, with the information of a limited universe of students and courses: six students, three bachelor courses and two masters courses. The grades’ scale for each course goes from 1 to 20, where 10 is the minimum positive grade, i.e. the minimum grade required for a student to be approved on any course. In the presented matrix we can see that John had 16 on the Artificial Intelligence (AI) course during bachelor and then had 17 on the Decision Supporting Systems (DSS) course on his masters. On the other hand you can see that Matt hasn’t enrolled in any of the masters’ courses, as its “grades” on both courses are zero.

With our students-courses matrix constructed, we can now start to look to how we will use it to produce the masters’ courses recommendations. Our idea is to apply a user-based approach to select the most similar historical students (neighbors) to a target student (considering their bachelor achievements) and then use those neighbors’ bachelor and masters’ achievements to construct a new a dimensional space using SVD. Afterwards, we use this new dimensional space to predict the grades of all masters’ courses so as to sustain the production of our recommendations. However, as we have seen in Section 2, SVD do not behave well when the knowledge over the matrix is incomplete and our domain is no exception when it comes to the big sparsity of the matrix, especially on the grades of masters’ courses (students only take a small subset of these courses). Though, we stated that Funk’s gradient descent algorithm to calculate SVD only uses the known ratings. We will use his approach into our problem in order to produce both the users and courses dimensional spaces, containing all students and courses features’ vectors, respectively.

For better understanding, we can imagine that Matrix $R$ contains the found neighbors for a certain target-student $X$ (we calculate similarities using Pearson Correlation). At this stage of the process, we would have to apply Funk’s gradient descent SVD to matrix $R$. The resultant matrices, which can be seen in Fig. 3, represent the users’ and courses’ dimensional spaces (with the number of hidden features set to two, so as to ease the task of visualizing the data). Matrix $P$ represents the users’ features dimensional space, where row $i$ stands for the student $i$ features vector, which relates student $i$ with each of the features (two features in the example). Analogously, each row of matrix $W$ shows how each course is related to each one of the two features. The product $PW'$ constitutes a 2-rank approximation of the original matrix $R$. 

<table>
<thead>
<tr>
<th>Students</th>
<th>Bachelor’s Courses</th>
<th>Masters’ Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>13</td>
<td>16</td>
</tr>
<tr>
<td>Kim</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>Matt</td>
<td>14</td>
<td>18</td>
</tr>
<tr>
<td>Tom</td>
<td>15</td>
<td>13</td>
</tr>
<tr>
<td>Fred</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>Ben</td>
<td>12</td>
<td>14</td>
</tr>
</tbody>
</table>

Figure 2 - Example of a student-courses matrix $R$

<table>
<thead>
<tr>
<th>Students</th>
<th>Feature 1</th>
<th>Feature 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>2,446</td>
<td>2,505</td>
</tr>
<tr>
<td>Kim</td>
<td>2,325</td>
<td>2,437</td>
</tr>
<tr>
<td>Matt</td>
<td>2,757</td>
<td>2,676</td>
</tr>
<tr>
<td>Tom</td>
<td>2,569</td>
<td>2,563</td>
</tr>
<tr>
<td>Fred</td>
<td>3,163</td>
<td>2,907</td>
</tr>
<tr>
<td>Ben</td>
<td>2,435</td>
<td>2,496</td>
</tr>
</tbody>
</table>

a) The users’ features space ($P$)

<table>
<thead>
<tr>
<th>Courses</th>
<th>Feature 1</th>
<th>Feature 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>NA</td>
<td>2,463</td>
<td>2,521</td>
</tr>
<tr>
<td>AI</td>
<td>2,848</td>
<td>2,712</td>
</tr>
<tr>
<td>DS</td>
<td>2,616</td>
<td>2,580</td>
</tr>
<tr>
<td>DPoW</td>
<td>2,538</td>
<td>2,506</td>
</tr>
<tr>
<td>DSS</td>
<td>2,757</td>
<td>2,694</td>
</tr>
</tbody>
</table>

b) The courses’ features space ($W$)

Figure 3 - Resultant matrices of Funks’ SVD gradient descent calculation
The relation that each student and each course hold with each feature, represented through their feature vectors in $P$ and $F$ respectively, can be well understood through Fig. 4. This figure shows a 2-dimensional feature space, and how are the students, bachelor's and masters' courses distributed along this dimensional space. With little observation we should take some interesting conclusions about the power of these students’ and courses’ features vectors. For instance, you should note that Ben and John are very close in this dimensional space. In fact, by inspection of Fig. 2, you can see that Ben and John have similar grades in average - 13.4 and 13.6 respectively. You can also note that Fred is a singular student, achieving grades above average and for that reason he is very distant of any other student. Regarding courses you should note an apparent similarity between the Artificial Intelligence (AI) bachelor’s course and the Decision Support Systems (DSS) masters’ course. Once again, if we give a closer look to the information in Fig. 2, it seems that the results on both courses appear to be correlated, as in most of the cases the deviation between the grades is only 1 grade point. As you can see, with a small number of features we were able to distribute both students and courses according to their contribution on each of the discovered hidden features.

With these dimensional spaces understood, we can now use the resulting matrices $P$ and $W$ to calculate any grades. For any student $i$, his grade on course $j$ (on the SVD dimensional space) corresponds to the dot product between $P_i$ and $W_j$. This is, to the dot product between student $i$ feature vector and course $j$ feature vector. The way we predict each student masters’ grade is by aggregating the grades of the SVD dimensional space on the following way:

$$r = \hat{r}_k + \frac{\sum_{u' \in U} PearsonCorrelation(u, u')(r_{u'j} - \hat{r}_{u'})}{\sum_{u' \in U} |PearsonCorrelation(u, u')|}$$

where $U$ denotes the set of neighbors to user $u$, and where $\hat{r}_k$ stands for the bachelor average of user $u$. Note that all the used grades in the aggregation are gathered from the SVD dimensional space (calculated using the neighbors) using the dot product method as we explained above.

After predicting the grade, it may be necessary to apply some bound restrictions in order to not have any predicted grade above or below the used grade scale, i.e. to guarantee that the prediction is between 1 and 20, in the scope of the example we have been following.

Finally, assuming that we predict the target-student X grades on every master’s course, we could take a simple approach to recommend courses to him: verify what is the set of $N$ masters’ courses with best predicted grades and then recommend them.

### 3.1 Evaluation Metrics

Current research on recommendation systems uses several types of measures for evaluating the success of recommendations. However, the correct way to evaluate each system depends heavily on the system’s goals and domain.

As we mentioned above, our recommendation process can be decomposed into two different problems: the grades prediction and the decision over which courses to recommend given the computed predictions. We believe that the overall quality of the recommendations, whichever method is used to produce them, depends on a great level of the quality of our predictions. Hence, this will be the main focus of our experiments.

To achieve it, we have to use one of the statistical accuracy metrics. These metrics aim to compare the prediction value against the actual real-values for the customer-product pairs, i.e. student-course pairs in our specific problem. Some of the most used metrics are the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) between the predicted and the real values. As the research
experience shows that both metrics typically track each other, we opted to use only one, in this case the MAE, because it is the most commonly used and easiest to interpret.

Another aspect that must be measured is the system’s capacity to provide distinct recommendations, as we don’t want the system to recommend only a small subset of the possible masters’ courses. We can measure this as the percentage of student-course pairs for which a recommendation can be made.

Additionally, it would be interesting to measure the average grade that students achieve with our recommendations. This can be computed with a simple average between the sum of grades achieved in followed recommendations and the total number of followed recommendations.

3.2 Experimental Results

We tested our approach with data from a bachelor and a masters program at Instituto Superior Técnico, Universidade de Lisboa, in Portugal. This dataset describes 9149 courses’ results achieved by nearly 500 students on both bachelor (of three years) and masters (of two years). The grades scale goes from 1 to 20, where 10 is the minimum grade that a student must achieve to be approved on any course. To perform our experiments we used the Java framework Recommender101 [20]. This framework enables us to carry out offline experiments for recommendation systems. It provides several metrics and a set of recommendation techniques to apply. One of the provided algorithms is Funk’s SVD descent gradient, which is the one used on our approach. However, we did some extensions to the framework in order to retrieve the results presented in this paper.

To have a comparison to our grades’ predictions results we used two inferior baselines. The first baseline sets the predicted grade of each student as the average of the grades by the student during his bachelor. Hence, the predicted grade for a student i on masters’ course j corresponds to student i bachelor’s grades’ average. The second baseline takes a similar approach but looks into the average grade achieved in each masters’ course, i.e. the predicted grade of student i on masters’ course j will be the average grade achieved by historical students on course j.

To do our prediction experiment, we started by constructing the students-courses matrix $R$ with the training data (the courses results of 70% of all students). Then, for each student of the test student, and considering bachelor information we identified the nearest neighbors, and calculated the SVD using those neighbors information, so as to be able to predict the grade in every master’s course.

As Funk’s SVD gradient descent has some free parameters, such as the learning rates and the number of features, we were interested in seeing how both these parameters could influence the quality of the predictions. First, we fixed the learning rate in 0.005 (small values produce small fluctuations), set the number of features to 20 and varied the number of neighbors, from 1 until 300. For each neighbor we recorded the average MAE on 3 runs of the algorithm. This need of performing a set of runs is due to the fact that the feature vectors are initialized with some random noise when the gradient descent algorithm begins. The achieved results in terms of MAE may be seen on Fig. 5.

It is clear that our approach with SVD has less error than any of the baselines comparators. In average, our recommendation predictions reach an MAE of 1.497, while the best baseline, the simple user based technique (reaching an MAE of 1.6), uses a big number of neighbors. Hence, we may affirm that our predictions sustain an above average basis from where to recommend masters’ courses to students. It is also interesting to see that with a small number of neighbors the SVD is able to obtain great results. This reflects the big power of this technique in identifying the most important hidden features that explain the values of every grade.
We now focus on how our recommendations may affect students’ academic achievements, in particular, their masters’ average grade. As we are doing an offline evaluation of our system – using historical data – we can’t predict the grade for all masters’ courses and then verify if the prediction was correct. For each student, in the best case we will have 11 masters’ courses from where to compare the predictions. Yet, we did an experiment where we fixed the free parameters of the Funk’s SVD (20 features and 0.005 on the learning rate value), predicted the grades as in the experiment above and then recommend the 11 courses with best predicted grades. To have a comparison to our approach we also used two baselines to recommend courses. One baseline looks into the student-courses matrix and recommends the 11 most popular courses, this is, the most frequented courses in the historical training data. The second baseline recommends the 11 courses where the students achieve better grades on the training data. Then, for each one of the approaches (SVD-based and baselines) we recommended and that the students in the testing set enrolled on.

From observation of Table 1, we can see that the average grade that students achieve with the recommendations of our SVD approach is better than any of the baselines. These results support our idea on the importance of a small grade prediction error in order to produce good quality recommendations. Though, to be completely accurate the number of followed recommendations for each of the approaches should be the same, which is not.

Table 1 – Average grade on followed recommendations

<table>
<thead>
<tr>
<th>Best Grades</th>
<th>Most Popular</th>
<th>SVD Based</th>
</tr>
</thead>
<tbody>
<tr>
<td>14,15</td>
<td>14,158</td>
<td>16,45</td>
</tr>
</tbody>
</table>

4 ESTIMATING MISSING VALUES WITH ASAP CLASSIFIERS

The other problem that we wanted to solve was in how to estimate missing values on the students’ bachelor information, so as to be able to still be able to produce the best recommendations. For example, a student may have made his bachelor on another university (with only a small subset of the bachelor courses of the new university). Another case is when a student wants to enroll in his masters, despite not having finished all of his bachelor courses.

Our suggestion to solve this problem was to use ASAP classifiers [12]. The idea behind these classifiers is that we can classify a problem instance using only the first $n$ of the $m$ attributes of the instance. This is done by estimating all the unobservable attributes so that in the end we have all needed information that a regular classifier needs. With this approach, we could fill the missing information on students’ profiles and use a regular process to retrieve the most similar students. This means, that using only the first $n$ of the $m$ bachelor courses that the student has enrolled on we would estimate the grades of the following $m - n$ bachelor courses.

4.1 Experimental Results

To test this technique, we wanted also to use the MAE as our evaluation metric. Our goal was to what was the difference between: i) using only the known information when the bachelor information is incomplete, ii) estimating the missing information using ASAP classifiers and iii) the regular approach presented above when all information is known.

To do it, we performed the experiments for two different cases. On one we used only the bachelor grades of the first year, either to directly apply the SVD-based approach presented in Section 3, or to first estimate the grades of the second and third year of the bachelor, or only then apply the SVD-based technique. On the second case we applied the same thought, but to using the information on the first two years (only the third year was unknown).

You can see in Fig. 6 and Fig. 7 that using only the information on the first year or the two first years of bachelor already gives a low MAE regarding the prediction of masters’ courses grades. Still, using ASAP classifiers to enrich the student profile gives a lower error. In both of the tested cases, the MAE using this estimation of the missing bachelor grades is very close to when the approach has all the values of the real grades of the students. However, we would like to achieve results as good as when
we have all the information on bachelor grades. That may be possible if we find a way to enrich ASAP classifiers so that the error on the estimations could be diminished. You should also note that the SVD-based approach presents a low MAE even the knowledge over the student’s bachelor grades is incomplete. This shows that even in the worst-case scenario we could provide useful recommendations.

5 CONCLUSIONS

The selection of which courses to attend during masters is one of the most decisive steps of a university student. However, students still struggle to choose the best courses for them, and most of the times do not have the required support and counseling needed to make such an important decision. Some approaches exist to the problem of recommending courses to students, but they are limited to model the potential interest of students in courses. Thus, no care is being taken regarding which courses are more suitable to students capabilities.

Recommendation systems are a powerful technology for extracting additional value in problems where the user is not sure of what are the best choices to make. Their application maps to our problem and so it is a natural choice from where to start to recommend courses to students. In this paper, we proposed the use of the gradient descent algorithm to calculate Singular Value Decomposition as a technique that enables the prediction of which grades students will have during their masters. Based on those predictions, we propose to recommend the best set of courses according students’ capabilities. Our study shows that Singular Value Decomposition may be used to recommend courses to students with a small error on which will be the students’ grades on masters. Funk’s SVD-based approach was consistently better than any of the baseline comparators in predicting students’ grades on masters. The main drawback of Funk’s SVD is that the algorithm may vary with the free parameters, and for that reason it is difficult to find the optimal set of choices to all parameters.

Future work is required to explore how SVD could be differently explored to predict recommendations. Item-based algorithms could be applied in the reduced features dimensional space to produce the grades predictions. This way, we could explore the relations between the grades of several courses to find the set of masters’ courses that are more adequate given the target student bachelor’s grades. Other ways to estimate the missing values, as an alternative to the ASAP classifiers, could be studied.

Some work could also be done in trying to associate SVD with personality diagnosis, which also tries to find the “hidden” personality of each user. Moreover, a deep analysis should be done on how to add more knowledge about students’ results, such as the number of courses failures the student had. More intelligent criteria could also be used to select the set of courses to recommend that not only to verify which are the courses with best grades.

6 ACKNOWLEDGEMENTS

This work is supported by Fundação para a Ciência e Tecnologia under research project educare (PTDC/EIA-EIA/110058/2009).

REFERENCES


