Domain-Aware Grade Prediction and Top-*n* Course Recommendation

Asmaa Elbadrawy
Computer Science
University of Minnesota, Twin Cities
asmaa@cs.umn.edu

George Karypis
Computer Science
University of Minnesota, Twin Cities
karypis@cs.umn.edu

ABSTRACT

Automated course recommendation can help deliver personalized and effective college advising and degree planning. Nearest neighbor and matrix factorization based collaborative filtering approaches have been applied to student-course grade data to help students select suitable courses. However, the student-course enrollment patterns exhibit grouping structures that are tied to the student and course academic features, which lead to grade data that are not missing at random (NMAR). Existing approaches for dealing with NMAR data, such as Response-aware and context-aware matrix factorization, do not model NMAR data in terms of the user and item features and are not designed with the characteristics of grade data in mind. In this work we investigate how the student and course academic features influence the enrollment patterns and we use these features to define student and course groups at various levels of granularity. We show how these groups can be used to design grade prediction and top-n course ranking models for neighborhood-based user collaborative filtering, matrix factorization and popularity-based ranking approaches. These methods give lower grade prediction error and more accurate top-n course rankings than the other methods that do not take domain knowledge into account.

Keywords

Grade Prediction, Top-n Course Ranking, Multi-Granularity Grouping

1. INTRODUCTION

While the flexibility of degree requirements provides college students with ample choices, it can complicate course selection. From among the courses that the students are eligible to take in the next term, they need to select the ones that they like, they are expected to perform well in, and also satisfy their degree requirements. Efficient college advising is essential for helping students select the right courses and thus, maintain high student retention rates and timely graduation. Automated course recommendation can help improve college advising by recommending courses that are suitable for the students degrees. Moreover, predicting student grades in the next term can help students and educators make informed decisions about course enrollments in order to produce better learning outcomes.

Collaborative filtering approaches have been previously used for grade prediction and course recommendation [23, 6, 8]. The majority of these methods rely on user-based collaborative filtering (User-CF) [11] which makes recommendations by relating to the courses that were taken by similar students. Recently, techniques based on matrix factorization (MF) have been used for movie and product recommendations [14] and also applied for course recommendation and grade prediction [27, 26].

The grade data has special characteristics as the student-course enrollments are influenced by the academic features (e.g., student majors, academic levels and course subjects). Consequently, the student-course grade matrix exhibits grouping structures as students with certain majors tend to enroll in courses of certain subjects, resulting in not missing at random (NMAR) data. Response-aware MF uses missing data theory to model the NMAR user response patterns [16]. However, the response patterns are not tied to the user and item features. Features-based MF methods can incorporate the user and/or item features into the prediction model. However, they do not explicitly model how the features determine the grouping structures in the data.

In this paper we analyze grade data and show how the student and course academic features determine the enrollment patterns. We use these features to define student and course groups and show how they can be incorporated in matrix factorization, user-based collaborative filtering, and popularity-based ranking.

We investigate various ways to define the groups at multiple levels of granularity using different amounts of academic features. We show that in some cases the small sample sizes associated with finer granularity groups make the prediction models prone to poor generalization, especially with matrix factorization based methods. To overcome this issue, we build multiple models using the various granularity groups. We then generate multiple grade predictions and combine them based on the sample sizes associated with the various groups.

We tested our methods on a dataset obtained from the University of Minnesota. The dataset spans 13 academic years and includes over 1,700,000 grades. Our results show that the methods that utilize finer groups give significantly

RecSys '16, September 15-19, 2016, Boston , MA, USA

DOI: http://dx.doi.org/10.1145/2959100.2959133

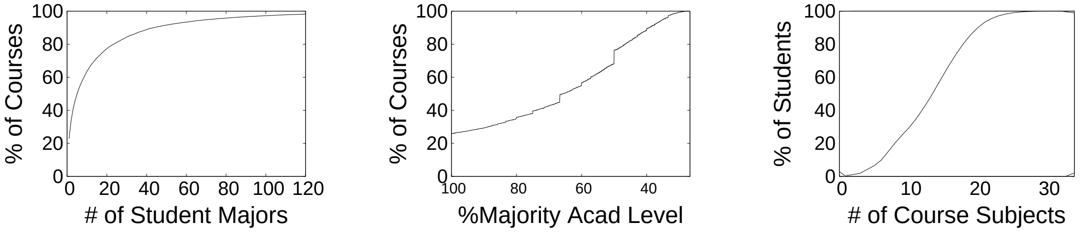


Figure 1: Cumulative percentage plots for student-course enrollments in a grade dataset. Left: percentage of courses vs. number of student majors. An (x,y) point indicates that y% of the courses were taken by students from at most x majors. For example, 40% of the courses were taken by students from at most three majors. Middle: percentage of courses vs. percentage of students belonging to the academic level that most of the enrolled students belong to. An (x,y) point indicates that for y% of the courses, at least x% of the enrolled students belong to the same academic level. For example, for 27% of the courses, at least 96% of the enrolled students belong to the same academic level. Right: percentage of students vs. number of subjects for the courses they have enrolled in. An (x,y) point indicates that y% of the students took courses that belong to at most x different subjects. For example, 70% of the students took courses that cover at most 16 different subjects.

more accurate top-n course rankings than the methods that utilize coarser groups and than other methods from the literature; even when domain knowledge is used to pre-filter the recommended courses. Moreover, defining the groups using the academic features gives better top-n rankings than clustering the students and courses using the enrollment data. For grade prediction, utilizing the finer groups gives more accurate predictions than the coarser groups only when they are associated with reasonable sample sizes. For the matrix factorization methods, where model training suffers more from the sample-size issue, combining the various model predictions while accounting for the sample sizes associated with the model parameters gives higher prediction accuracy than the various individual MF models and than the other methods from the literature.

2. DEFINITIONS AND NOTATIONS

Let G denote the grade matrix; each row in G represents a student, denoted by s, and each column represents a course, denoted by c. The entry $g_{s,c}$ in G represents the grade obtained by student s in course c. A predicted grade is denoted by $\hat{g}_{s,c}$.

3. CHARACTERISTICS OF GRADE DATA

In a university setting, each student enrolls in a certain college/school and declares a certain major. Each student selects from among a variety of courses to take in order to fulfill the requirements of his major (also referred to as the degree requirements). Each course has a subject that it falls under and a level that describes its difficulty. As the student takes more courses, his academic level advances and he can take higher level courses.

Students tend to enroll in courses that are related to their majors and are appropriate to their academic levels. This is illustrated in Figure 1 which shows various characteristics that were extracted from a grade dataset that was obtained from the University of Minnesota. These plots show that: (i) each course is taken by students that belong to a limited number of majors; (ii) each course is mostly taken by students belonging to one academic level; and (iii) each student takes courses that cover a limited number of subjects. For example, from Figure 1-left we can see that 22% of the courses are taken by students that all come from one major, and 80% of the courses are taken by students that come from at most 24 majors. From Figure 1-middle we can see that for 50% of the courses, at least 66% of the enrolled students belonged to the same academic level. Finally, from Figure 1-right we can see 90% of the students took courses that covered less than 20 subjects.

These characteristics imply that the missing entries in the grade matrix are not missing at random. This resulted because students with certain student features tend to enroll in courses with certain course features. We refer to this as the *grouping structures* in the grade data.

4. RELATED WORK

Response aware techniques [16, 17, 12] model NMAR data by utilizing a data model that is based on missing data theory. The method proposed in [16] modified probabilistic matrix factorization by introducing two variations to model NMAR data. The first variation assumes that the probability of observing a rating depends only on the value of the rating. The second variation assumes that the probability also depends on the user and the item latent factors. None of these methods incorporate the user and item features that influence the response patterns.

Feature-based MF methods incorporate user and/or item features within the rating prediction or the top-n ranking models. The method proposed in [2] linearly transformed the user and item features to the latent space in order to predict a user's preference over a given item. Other methods incorporated the features within a top-n recommendation model in order to estimate user preferences or bias the recommendations based on the item features [18, 10]. None of these methods were designed to address how the user and item features determine the grouping structures in the data.

Context-aware methods make recommendations in accordance with the different contexts [4, 13, 3, 25, 28, 1]. Some of these methods utilized the context information to pre-filter items. Other techniques incorporated contextual information within the model.

Methods for course recommendation applied various data mining techniques to tackle the problem. The work done in [5] applied association rule mining to recommend relevant courses. The method in [15] estimated course recommendation scores by accumulating weights for subject importance within the study field, satisfied prerequisites and the extent by which a course broadens the student's knowledge state. Methods for course recommendation with constraints focused on satisfying the degree program requirements [20, 22, 21, 19]. They take course prerequisites into consideration in order to generate valid course recommendations. They focus on finding a short path to fulfill the degree requirements and thus, reduce time to graduation.

5. DOMAIN-AWARE METHODS FOR COURSE RECOMMENDATION

We develop methods that model the grouping structures of the grade data by using the academic features to define student and course groups. These groups are defined at various levels of granularity by utilizing various amounts of features. Then they are incorporated within the recommendation methods for the purpose of performing (1) grade prediction and (2) top-n course ranking.

For grade prediction, the grade of a student s in a course c should be predicted by relating to how students of the same group as s performed in c, and how s performed in courses of the same group as c. Since the groups are defined at different levels of granularity, various models can be built that account for various academic features. In general, the finer groups are more homogeneous and thus, utilizing them can give more accurate predictions than utilizing the coarser groups. However, based on how the groups are incorporated into the prediction models, some models can be affected when the finer groups have small sample sizes and they can become prone to poor generalization. Such cases are addressed by building multiple models using different granularity groups and combining the predictions of all the models based on the group sample sizes.

For top-n course ranking, it is required to generate a list of n relevant courses for each student to consider enrolling in them. Unlike other recommendation scenarios, course recommendation has special considerations. Students need to enroll in courses that they are interested in, and that fulfill their degree requirements. Accordingly, students sometimes need to take some courses in order to fulfill some degree requirement, regardless of their expected grade in these courses. Moreover, in the typical user-item-rating scenario, when a user likes an item, he gives that item a high rating. This is not always the case with the student-course-grade scenario where a student might like a course, but this does not necessarily mean that he will get a high grade when he takes that course. Based on that, course ranking should rely on the enrollment patterns more than relying on the expected grades. In this sense, the grade matrix is considered as binary where all grades are set to 1's and the rest of the entries are considered 0's. Similar to grade prediction, multiple models can be built by utilizing various student and course groups. Also, utilizing finer groups can give more accurate recommendations that the coarser groups. However, unlike with grade prediction, the fact that some finer groups are associated with small sample sizes is an indicator for less relevant courses and as such, should not hurt model generalization.

We next describe how to define the multi-granularity groups.

5.1 Defining the Multi-Granularity Student and Course Groups

The student groups define the various student subpopulations that can take a course. At the coarsest level, the set of all students is defined without using any student academic features. Then finer groups are defined by using one student feature at a time to segment the student group(s) further based on the value of that feature to give smaller and more homogeneous groups. The course groups are defined similarly using the course academic features.

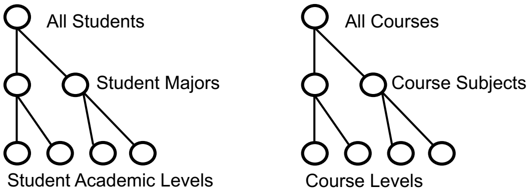


Figure 2: An illustrative example for defining the student(left) and course(right) multi-granularity groups.

One way to define the student groups is shown in Figure 2-left. The node at the top represents the group of all students. The second level segments the students based on their majors and it has one node (group) for each major. The third level segments the students further based on their academic levels and it has one node for each academic level within each major. Similarly, the course groups can be defined as shown in Figure 2-Right. The node at the top represents the group of all courses. The second level segments the courses based on their subjects and it has one node (group) for each subject. The third level segments the courses further based on their levels and it has one node for each course level within each subject. In the rest of the paper, we will use the groups defined in Figure 2 as an illustrative example.

We next describe how the groups are incorporated into popularity based ranking, neighborhood based user collaborative filtering and matrix factorization.

5.2 Popularity based Top-n Course Ranking

A popularity ranking scheme ranks the courses based on how frequently they were taken by the students. In our case, we rank the courses for a student s based on how frequently they were taken by students of the same group as s. The ranking score of course c for a given student s is computed as $|\varphi_{s\to c}|$, where $\varphi_{s\to c}$ is the set of students in the same group as s that have taken c and $|\mathcal{X}|$ represents the cardinality of set $\mathcal{X}$.

Utilizing a different student group from the multi granularity groups gives a different model. The various models are referred to as Grp-Pop-h_{φ_s}, where h_{φ_s} is the student group level in the multi-level groups. For example, the model that utilizes the groups at the second level of the multi-level student groups in Figure 2-left is referred to as Grp-Pop-2.

5.3 User based Collaborative Filtering

User-CF predicts a grade of a student s in a course c by relating to how students that have taken same courses as s performed in c as

$$\hat{g}_{s,c} = \bar{g}_s + \frac{\sum_{s'\in N_s} \mathrm{sim}(s,s')(g_{s',c} - \bar{g}_{s'})}{\sum_{s'\in N_s} |\mathrm{sim}(s,s')|}, \qquad (1)$$

where $\bar{g}_s$ is the average grade of s, N_s is the set of neighbor students to s, and $\mathrm{sim}(s,s')$ is some similarity between students s and s'.

5.3.1 Grade Prediction

For grade prediction, the neighborhood set N_s is selected based on the student groups as follows. Any student in the same group as s and has taken at least n_c courses that were taken by s is selected as part of N_s. Moreover, the size of N_s is limited so that it only contains the n_n students that are most similar to s. The threshold parameters n_c and n_n are fine-tuned using a validation set. In the case where not enough neighbors are found, the grade is then estimated as

$$\hat{g}_{s,c} = \frac{1}{2}(\bar{g}_s + \bar{g}_c),$$

where $\bar{g}_c$ is the average grade for course c.

By utilizing different student groups, different models are built. The various models are referred to as User-CF-h_{φ_s}. For example, the model that utilizes the groups at the second level of the multi-level student groups in Figure 2-left is referred to as User-CF-2.

5.3.2 Top-n Course Ranking

Since in this case the enrollment patterns are the main indicators and not the grade values, G is converted into a binary matrix with all grades set to 1's and other entries considered as 0's. The recommendation scores are then estimated as in Equation 1. In practice this gives better recommendations than using the actual grade values. In the case where not enough neighbors are found, this indicates an irrelevant course and the course rank is set to 0.

5.4 Matrix Factorization

MF predicts the grade of student s in course c as

$$\hat{g}_{s,c} = b_s + b_c + \boldsymbol{u}_s^t \boldsymbol{v}_c, \quad (2)$$

where b_s and b_c are the bias terms of s and c, and $\boldsymbol{u}_s$ and $\boldsymbol{v}_c$ are the latent factor vectors of s and c, respectively.

5.4.1 Grade Prediction

While the literature is rich with feature-based and context-aware MF techniques that can be modified and used as a framework to implement our ideas, we choose to modify the context-aware technique in [4] as we find it most relevant. This technique accounts for context via additional bias terms that are defined for each (item, context) pair.

In our case, we use the student groups to describe the contexts in which a course is taken, and use the course groups to describe the contexts in which a student takes a course. Considering the example in Figure 2, the third level student groups describe course-side contexts in terms of the student majors and academic levels. Similarly, the third level course groups describe student-side contexts in terms of the course subjects and levels. Accordingly, we define multiple bias terms per student and per course to account for the various student- and course-side contexts. The recommendation score of a given student s and course c is estimated as

$$\hat{g}_{s,c} = b_s^{\varphi_c} + b_c^{\varphi_s} + \boldsymbol{u}_s^t \boldsymbol{v}_c, \quad (3)$$

where $b_s^{\varphi_c}$ is some student bias that accounts for the context described by the course group of c, $b_c^{\varphi_s}$ is some course bias that accounts for the context described by the student group of s, and u_s and u_c are the latent factor vectors for s and c, respectively.

Multiple models can be defined using the different groups. For the example in Figure 2, considering the various student and course group combinations, we can build nine different models. The various models are referred to as MF-h_{φ_s}-h_{φ_c}, where h_{φ_s} is the level of the student group, which defines the granularity of the course bias, and similarly h_{φ_c} is the level of the course group, which defines the granularity of the student bias. For example, considering Figure 2, MF-1-3 is used to refer to the model that uses the coarsest-grain student groups (at the 1st level) to define the coarsest-grain course biases, and uses the finest-grain course groups (at the 3rd level) to define the finest-grain student biases.

Since the finer groups are more homogeneous than the coarser groups, the MF models that utilize them can give more accurate predictions. However, the student groups are recognized through defining multiple biases for each course, and similarly with the course groups and the student biases. For example, if we have 2,000 student groups and 1,000 course groups, then 2,000 biases are defined per course and 1,000 biases are defined per student. Only a handful of biases for each student/course are associated with some data points. and the remaining majority of the biases are associated with very few or no data points. Therefore, the models that utilize finer groups can become prone to poor generalization. To understand why this happens, consider the following example. Assume an Artificial Intelligence course c that is offered by the Computer Science department was taken by 47 Computer Science major students and 2 other Liberal Arts major students. If we define student groups using the major, and if we have 100 different majors, then c will have one bias associated with 47 data points, one bias associated 2 data points and 98 biases associated with 0 data points. Obviously, the biases with 0 and 2 data points cannot be as accurately estimated as the other bias.

To overcome this problem, we build multiple models utilizing various groups and use them to generate multiple predictions. Then the predictions are combined based on the sample sizes that are associated with the bias terms of the various models as described next.

5.4.1.1 Combining the Predictions of the Different MF Models.

Before discussing how the various model predictions are combined, it is worth noting that the user and item latent factors are not shared among the various models but each model has its own factors. The various predictions are combined while accounting for the associated sample sizes as follows. Each model MF-h_{φ_s}-h_{φ_c} has a combination weight given by

$$w_{\{h_{\varphi_s}, h_{\varphi_c}\}} = sup(b_s^{\varphi_c}) + sup(b_c^{\varphi_s}),$$

where $sup(b_s^{\varphi_c})$ is the sample size (i.e., number of training samples) associated with the bias term $b_s^{\varphi_c}$. The total weight is aggregated over the individual model weights as

$$w_{total} = \sum_{h_{\varphi_s}} \sum_{h_{\varphi_c}} w_{\{h_{\varphi_s}, h_{\varphi_c}\}}.$$

The final prediction is then given by

$$\hat{g}_{\{s,c\}} = \sum_{h_{\varphi_s}} \sum_{h_{\varphi_c}} \alpha_{\{h_{\varphi_s}, h_{\varphi_c}\}} \times \frac{w_{\{h_{\varphi_s}, h_{\varphi_c}\}}}{w_{total}} \times \hat{g}_{\{s,c\}}^{\{h_{\varphi_s}, h_{\varphi_c}\}}, \quad (4)$$

where $\hat{g}_{\{s,c\}}^{\{h_{\varphi_s}, h_{\varphi_c}\}}$ is the prediction given by model MF-h_{φ_s}-h_{φ_c}, and $\alpha_{\{h_{\varphi_s}, h_{\varphi_c}\}}$ is some global combination weight for

Table 1: The student and course features used to define various multi-level groups and the resulting number of groups.

	2nd level student feature	3rd level student feature	2nd level course feature	3rd level course feature
H-1	student major (565)	student academic level (565×4)	course subject (570)	course level (570×8)
H-2	student academic level (4)	student major (565×4)	course subject (570)	course level (570×8)
H-3	student college (10)	student academic level (10×4)	course subject (570)	course level (570×8)
H-4	student academic level (4)	student college (4×10)	course subject (570)	course level (570×8)
H-5	student major (565)	student academic level (565×4)	course level (8)	course subject (8×570)
H-6	Students/courses are clustered by splitting a nearest-neighbor similarity graph into k-clusters via min-cut graph partitioning. Clustering is repeated 2 times with (k_1=10, k_2=30 for students) and (k_1=5, k_2=25 for courses) to generate 3-level groups.			

that model. This method is referred to as INTRP-MF, the interpolative multi-granularity MF method.

5.4.1.2 *Model Parameter Estimation.*

Parameter estimation is done via a step-wise optimization process in which the parameters of each of the individual models are first estimated, and then the $\alpha_{\{h_{\varphi_s}, h_{\varphi_c}\}}$ global combination weights are estimated.

The parameters of each of the various models are estimated via a regularized optimization process of the form

$$\underset{\Theta}{\text{minimize}} \ \ \mathcal{L}(\Theta) + \mathcal{R}(\Theta), \qquad (5)$$

where Θ represents the model parameters, $\mathcal{L}(\Theta)$ is the loss function and $\mathcal{R}(\Theta)$ is a regularization function to avoid overfitting. We use a squared error loss function of the form

$$\mathcal{L}(\Theta) = \sum_{g_{s,c} \in G} (g_{s,c} - \hat{g}_{s,c}(\Theta))^2,$$

where $\hat{g}_{s,c}(\Theta)$ is given by Equation 3. This loss function is suitable as the letter grades can be transformed to numeric values. The regularization function $\mathcal{R}(\Theta)$ is given by

$$\mathcal{R}(\Theta) = \lambda_u(||U||_F^2 + ||B_S^{\varphi_C}||_F^2) + \lambda_v(||V||_F^2 + ||B_C^{\varphi_S}||_F^2), \quad (6)$$

where λ_u and λ_v are the regularization parameters and $||U||_F$ is the ℓ-2 norm of the matrix U.

After the parameters of each model are estimated, the $\alpha_{\{h_{\varphi_s}, h_{\varphi_c}\}}$ weights are estimated by minimizing a mean squared error loss as well.

5.4.2 *Top-n Course Ranking*

We use a learning to rank approach to generate personalized course recommendations per student. The rank of course c for student s is estimated as in Equation 3. The model parameters of each model are estimated using a personalized pair-wise ranking loss function [7] of the form

$$\mathcal{L}(\Theta) = -\sum_{s \in G} \sum_{c \in \mathcal{C}_s} \sum_{c' \in \bar{\mathcal{C}}_{\varphi_s}} \phi(\hat{g}_{s,c}(\Theta) - \hat{g}_{s,c'}(\Theta)), \qquad (7)$$

where $\mathcal{C}_s$ is the set of courses taken by student s, $\bar{\mathcal{C}}_{\varphi_s}$ is the set of courses never taken by any student in the same group as s, and $\phi(z) = e^{-z}$. Although $\hat{g}_{s,c}(\Theta)$ is estimated using Equation 3, it represents a ranking score in this case and not a predicted grade because the model parameters are estimated using the ranking-based loss function. We use the same regularization function as in Equation 6 to avoid overfitting.

The ranking loss function is of order $O(n_u \times n_i)$, where n_u and n_i are the number of students and courses, respectively. The learning time can be reduced by sampling, for each student, from among his $\bar{\mathcal{C}}_{\varphi_s}$ instead of considering the whole set. If the number of samples is of order $O(\mathcal{C}_s)$, the run time is reduced to $O(NNZ_G)$, the number of non-zero entries in the grade matrix G.

6. EXPERIMENTAL DESIGN

In this section we describe the dataset that is used for evaluation, the evaluation metrics, the methods that we compare against, how the various methods are trained and how the student and course groups are defined.

6.1 Dataset

The dataset used for evaluation is obtained from the University of Minnesota. It spans 13 academic years and it has over 1.7 million letter grades that involve around 60,000 students, 10,000 courses, 10 colleges, 570 course subjects, 565 majors, 4 academic levels and 8 course levels. The grades are converted into numbers according to the 4.0 GPA standard[1]. All Pass/Fail grades are removed from the dataset.

The last term in the dataset is used for testing and the rest are used for training and model selection. The last term in the training set is used for model selection and the rest is used for training. Grades of the students that have graduated before the test term are included in the training set. Grades for the new courses and the new students that first appear in the test term are excluded.

6.2 Defining the Student and Course Groups

We experimented with six different ways to define the multi-level groups, namely, H-1 up to H-6. Each one contains three levels of student and course groups and thus, the corresponding MF models are referred to as MF-1-1 up to MF-3-3. The various User-CF and Grp-Pop methods that are defined based on the student groups are referred to as User-CF-1 up to User-CF-3, and Grp-Pop-1 up to Grp-Pop-3, respectively. The features used to define the groups are listed in Table 1.

6.3 Evaluation Metrics

Methods are evaluated for (1)the accuracy of top-n course ranking and (2)the accuracy of grade prediction.

Course top-n ranking is evaluated with **Recall@n** which is computed for each student s as

$$Recall@n_s = \frac{n_{s,n}}{n_{t_s}},$$

where $n_{s,n}$ is the number of courses that appeared in the test set of s and in his list of n recommended courses, and n_{t_s} is the number of courses in the test set of s. Recall@n is computed by averaging over Recall@n_s for all s and for n in the range [1,10]. The relative methods performances did not change with n and so, we only report results for $n = 5$.

Grade prediction accuracy is evaluated by computing the

[1] See "http://www.collegeboard.com/html/academicTracker-howtoconvert.html" for letter grade-grade point conversion.

Root Mean Squared Error on the testing grades $\mathcal{G}_{test}$ as

$$RMSE = \sqrt{\frac{\sum_{g_{s,c} \in \mathcal{G}_{test}} (g_{s,c} - \hat{g}_{s,c})^2}{|\mathcal{G}_{test}|}}.$$

6.4 Comparison with Other Methods

We compare the performance of our method against the following approaches:

User Collaborative Filtering: User-CF with Pearson correlation for user similarity [9].

Matrix Factorization: Typical MF as described in Equation 2.

Response-aware Matrix Factorization: The context aware response model described in [16]. We implemented RAPMFc as we think it is the most relevant because it captures the probability that rating an item depend on the rating value, the user and the item.

Regression Latent Factor Models: The feature-based MF-based method described in [2], referred to as RLFM. We used libFM [24] to generate the results for grade prediction only as it is not a top-n ranking technique.

Ensemble-based Grade Combination: We compare the predictive performance of the the interpolative multi granularity method INTRP-MF against various ensembles. The Minimum, Maximum, Average and Median ensembles are considered where the minimum, maximum, average and median grades are selected as the final prediction, respectively. These ensembles are referred to as MIN-En, MAX-En, AVG-En and MED-En, respectively. We also include results for the interpolative method with excluding the α parameters in order to show how the sample-size-based weights perform. This method is referred to as WT-MF and it does not need a secondary learning step as the α weights are omitted.

6.5 Model Training and Selection

For training the MF models, we tried a number of latent factors in the range [1,10] and λ_u and λ_v in the range [1e-4,5]. The values that gave the best results were latent factors in the range [1,3], λ_u and λ_v in the range [0.1,3.5].

For User-CF, we have tried values for the parameters, n_n and n_c in the range [1,50]. The best results were obtained with values in the range [2,36].

For RAPMFc, we have tried parameter values for λ_u, λ_v and λ_μ in the range [10^{-3},10^1], β in the range [0,1] and number of factors in the range [1,10]. The values that gave the best results were in the range [0.01, 0.1] for λ_u, λ_v, 1 for λ_μ, 0.01 for β and [7,10] for l.

For grade prediction, and top-n ranking, model selection is based on the lowest RMSE and the highest Recall@n on the validation set, respectively.

7. EXPERIMENTAL RESULTS

We assess the effectiveness of the developed methods in order to answer the following questions:

Q1. Does incorporating the groups in the various methods lead to better top-n course rankings?

Q2. Does incorporating the groups in the various methods lead to better grade predictions?

Q3. How is the grade prediction performance of the MF models defined using various groups affected by the sample sizes that are associated with the biases?

Table 2: Recall@5 for the various groups. The highest Recall@5 for each set of methods within each group (column) is underlined.

Model	H-1	H-2	H-3	H-4	H-5	H-6
Grp-Pop-1	0.015	0.015	0.015	0.015	0.015	0.015
Grp-Pop-2	0.172	0.039	0.046	0.039	0.172	0.015
Grp-Pop-3	0.236	0.236	0.094	0.094	0.236	0.017
User-CF-1	0.046	0.046	0.046	0.046	0.046	0.046
User-CF-2	0.050	0.034	0.037	0.034	0.050	0.035
User-CF-3	0.054	0.054	0.037	0.037	0.054	0.036
MF-1-1	0.013	0.013	0.013	0.013	0.013	0.013
MF-2-1	0.174	0.066	0.047	0.066	0.174	0.021
MF-3-1	0.239	0.239	0.100	0.104	0.239	0.018
MF-1-2	0.041	0.041	0.040	0.041	0.018	0.021
MF-2-2	0.174	0.133	0.055	0.133	0.169	0.021
MF-3-2	0.238	0.238	0.109	0.115	0.230	0.018
MF-1-3	0.023	0.023	0.021	0.022	0.018	0.020
MF-2-3	0.172	0.081	0.047	0.081	0.171	0.019
MF-3-3	0.236	0.236	0.100	0.105	0.236	0.017
RAPMFc	0.023					

7.1 Top-n Course Recommendation Results

Prior to ranking the courses for each student, we apply a domain-aware pre-filtering in which courses that have never been taken by at least one student of the same major and academic level as the target student are filtered out. This approach performed the best among other similar pre-filtering rules that utilize various academic features.

Table 2 shows the Recall@5 for all the methods across all groups. Notice that the typical popularity ranking, User-CF and MF schemes are equivalent to Grp-Pop-1, User-CF-1 and MF-1-1, respectively.

For the popularity methods, Grp-Pop-3 and Grp-Pop-2 outperform Grp-Pop-1. Across the six groups H-1 to H-6, Grp-Pop-3 gives the highest recall.

For the User-CF methods, User-CF-2 and User-CF-3 only outperform User-CF-1 when the groups are defined in terms of the student majors (H-1, H-2 and H-5). For these groups, User-CF-3 gives the highest recall. For the other groups, User-CF-1 gives the highest recall.

For the MF methods, the ones that utilize groups outperform MF-1-1 by an order of magnitude. In general, defining the course biases using finer student groups gives better recall as it is the case, for example, with MF-1-2, MF-2-2 and MF-3-2 in H-1. On the other hand, defining student biases using finer course groups does not always give better recall as it is the case with MF-2-2 and MF-2-3 in H-2, H-3 and H-4. We believe this is related to the sample sizes associated with the student biases. Since each student takes a limited number of courses, the models utilizing the finest course groups have less than 2 training points associated with their student biases on average. RAPMFc performs better than MF-1-1 but worse than all models that utilize groups across H-1 to H-6.

Across all methods, the highest recalls are given by MF-3-2 and MF-3-1 which slightly surpass MF-3-3 and Grp-Pop-3. All User-CF methods outperform RAPMFc, MF-1-1, MF-1-2 and MF-1-3 for all groups. MF models with finer student groups, like MF-3-1, MF-3-2 and MF-3-3, always outperform all User-CF methods. The student groups that are defined in using majors (H-1, H-2 and H-5) give higher recall than the groups defined using colleges (H-3 and H-4). The clustering-based groups give the lowest recall, indicating that the clustering could not capture the groups that are defined by the student and course academic features.

Table 3: RMSE for all groups. The lowest RMSE for each set of methods within each group (column) is underlined.

Model	H-1	H-2	H-3	H-4	H-5	H-6
User-CF-1	0.714	0.714	0.714	0.714	0.714	0.714
User-CF-2	$\underline{0.705}$	$\underline{0.704}$	$\underline{0.706}$	$\underline{0.704}$	$\underline{0.705}$	$\underline{0.706}$
User-CF-3	0.707	0.707	$\underline{0.706}$	0.706	0.707	0.709
MF-1-1	$\underline{0.666}$	$\underline{0.666}$	$\underline{0.666}$	$\underline{0.666}$	$\underline{0.666}$	$\underline{0.679}$
MF-2-1	0.692	0.672	0.674	0.672	0.692	0.689
MF-3-1	0.689	0.689	0.692	0.692	0.689	0.713
MF-1-2	$\underline{0.663}$	$\underline{0.663}$	$\underline{0.663}$	$\underline{0.663}$	0.669	0.689
MF-2-2	0.680	0.671	0.672	0.671	0.696	0.702
MF-3-2	0.682	0.682	0.680	0.680	0.696	0.716
MF-1-3	$\underline{0.664}$	$\underline{0.664}$	$\underline{0.664}$	$\underline{0.664}$	$\underline{0.664}$	0.687
MF-2-3	0.687	0.673	0.681	0.673	0.687	0.700
MF-3-3	0.694	0.694	0.689	0.689	0.694	0.713
MIN-En	0.715	0.704	0.713	0.709	0.730	0.722
MAX-En	0.711	0.680	0.688	0.686	0.705	0.716
AVG-En	0.660	0.660	0.660	0.662	0.665	0.681
MED-En	0.665	0.661	0.660	0.663	0.674	0.681
WT-MF	$\underline{0.658}$	$\underline{0.659}$	0.661	$\underline{0.662}$	$\underline{0.661}$	$\underline{0.678}$
INTRP-MF	$\underline{0.658}$	$\underline{0.659}$	$\underline{0.660}$	$\underline{0.662}$	$\underline{0.661}$	$\underline{0.678}$
RLFM	0.731	0.731	0.728	0.728	0.733	0.740
RAPMFc	1.175					

7.2 Grade Prediction Results

We first discuss the performance of the different methods, then we discuss the effect of the sample sizes on the performance of the different MF models.

7.2.1 *Performance of the different methods*

RMSE given by the different methods across all groups are listed in Table 3. For the User-CF methods, User-CF-2 and User-CF-3 that utilize finer groups give lower RMSE than User-CF-1. User-CF-2 gives the lowest RMSE when the student sgroups are defined using the academic level.

For the MF methods, MF-1-1 gives the lowest RMSE across the different groups. MF models that utilize finer groups tend to give higher RMSE. We believe this has to do with the effect of the sample sizes that are associated with the various groups, which is analyzed in more details in Section 7.2.2.

INTRP-MF gives lower RMSE than all the ensembles across the different groups. That is because it only gives higher weights to the finer models as their biases are associated with larger sample sizes, which indicates a better ability to generalize. INTRP-MF does only marginally better than WT-MF, which indicates that the improvement is largely due to the sample-size-based weights. The AVG-En gives the third lowest RMSE and in many cases it performs worse than MF-1-1. Among all methods and groups, INTRP-MF gives the lowest RMSE, followed by WT-MF.

RAPMFc gives higher RMSE than all MF and User-CF methods. This is consistent with the results presented in [16] since our test set represents inspected entries for courses that students have taken. RLFM gives lower RMSE than the User-CF methods but higher RMSE than the MF methods.

7.2.2 *Change in MF models' RMSE with the bias sample sizes*

To understand how the sample sizes associated with the bias terms of the various MF models affect their performance, and to show why INTRP-MF works, we analyze how the RMSEs of the various models change with the number of training samples associated with their biases. To do so, we extract multiple subsets from the test set. Each subset contains test cases whose corresponding student and course biases in the finest model, MF-3-3, are associated with a minimum number of training samples referred to as α and β. We try values for α and β in the range [0,100] to generate various test subsets. Then for each subset we compute the RMSE for all the models and plot the RMSEs against the subset coverage (number of test cases in the subset).

Figure 3 shows the RMSE against the coverage for the various models with the H-1 groups. Each coverage point represents a test subset with that amount of test cases. As the coverage decreases, the sample sizes associated with the biases of the various models increases. For each coverage point we plot the RMSE of the various models. Subfigures (a), (b) and (c) show how the models with various course groups perform given a fixed student group. At the highest coverage of 50,000 test cases, models with the coarsest course groups and thus, coarsest student biases (MF-1-1, MF-2-1 and MF-3-1) give the lowest RMSEs. For lower coverages between 10,000 and 500 (indicating that finer models have more training examples), models with finer course groups and thus, finer student biases (MF-1-2, MF-2-2 and MF-3-2) give the lowest RMSEs. We can conclude from this that INTRP-MF manages to yield lower RMSE as it gives higher weights to the finer models when their biases are associated with more samples, i.e., when they can give lower RMSE.

8. CONCLUSIONS & FUTURE WORK

In this paper we addressed the grade prediction and top-n course ranking problems. We showed how the student and course academic features determine the enrollment patterns and we defined multi-granularity student and course groups accordingly. We showed how these groups can be incorporated in user collaborative filtering, matrix factorization and popularity ranking methods.

By evaluating the various methods on a large dataset, we showed that incorporating the features-based groups into the various methods leads to better grade predictions and top-n course rankings. We also showed how the grade prediction accuracy of matrix factorization methods slightly degrades when their biases are associated with small sample sizes; an issue occurring with utilizing finer groups. We showed how this can be handled by building various models utilizing various-granularity groups and combining their predictions based on the sample sizes associated with their biases. Our results also showed that the student groups defined using the majors and academic level gave the best top-n rankings and the most accurate grade predictions.

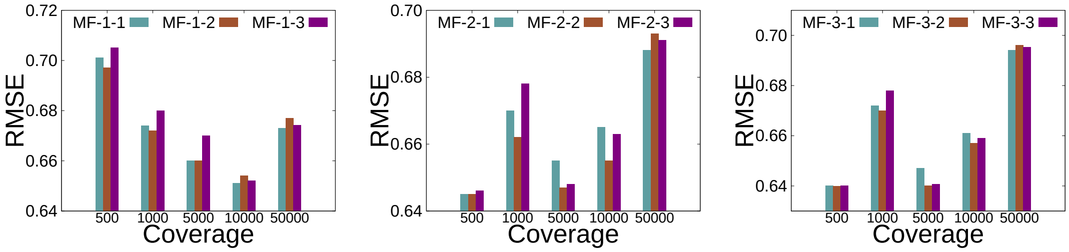
In the future we will consider special cases like ranking non-required courses while considering grades in evaluation.

9. ACKNOWLEDGEMENT

This work was supported in part by NSF (OCI-1048018, IIS-1247632, IIP-1414153, IIS-1447788), Army Research Office (W911NF-14-1-0316), Intel Software and Services Group, and the Digital Technology Center at the University of Minnesota. Access to research and computing facilities was provided by the Digital Technology Center and the Minnesota Supercomputing Institute.

10. REFERENCES

[1] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in



(a) (b) (c)

Figure 3: Coverage vs. RMSE for the MF models on common test cases using the H-1 groups.

recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1), 2005.

[2] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 19–28. ACM, 2009.

[3] D. Agarwal, B.-C. Chen, and B. Long. Localized factor models for multi-context recommendation. In *SIGKDD*, 2011.

[4] L. Baltrunas, B. Ludwig, and F. Ricci. Matrix factorization techniques for context aware recommendation. In *ACM RecSys*, 2011.

[5] N. Bendakir and E. Aimeur. Using association rules for course recommendation. *AAAI Workshop on Educational Data Mining*, 2006.

[6] H. Bydzovska. Are collaborative filtering methods suitable for student performance prediction? In *Progress in Artificial Intelligence*, volume 9273 of *Lecture Notes in Computer Science*.

[7] W. Chen, T. yan Liu, Y. Lan, Z. ming Ma, and H. Li. Ranking measures and loss functions in learning to rank. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 315–323. Curran Associates, Inc., 2009.

[8] T. Denley. Course recommendation system and method, 2013. US Patent App. 13/441,063.

[9] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan. Collaborative filtering recommender systems. *Found. Trends Hum.-Comput. Interact.*, 4(2), 2011.

[10] A. Elbadrawy and G. Karypis. User-specific feature-based similarity models for top-n recommendation of new items. *ACM Trans. Intell. Syst. Technol.*, 6(3), Apr. 2015.

[11] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR*, 1999.

[12] J. M. Hernández-Lobato, N. M. T. Houlsby, and Z. Ghahramani. Probabilistic matrix factorization with non-random missing data. In *ICML*, 2014.

[13] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *ACM RecSys*, 2010.

[14] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

[15] Y. Lee and J. Cho. An intelligent course recommendation system. *Smart CR*, 1(1):69–84, 2011.

[16] G. Ling, H. Yang, M. R. Lyu, and I. King. Response aware model-based collaborative filtering. *CoRR*, abs/1210.4869, 2012.

[17] B. M. Marlin and R. S. Zemel. Collaborative prediction and ranking with non-random missing data. In *ACM RecSys*, 2009.

[18] X. Ning and G. Karypis. Sparse linear methods with side information for top-n recommendations. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12. ACM, 2012.

[19] A. Parameswaran, P. Venetis, and H. Garcia-Molina. Recommendation systems with complex constraints: A course recommendation perspective. *ACM Trans. Inf. Syst.*, 29(4), 2011.

[20] A. G. Parameswaran and H. Garcia-Molina. Recommendations with prerequisites. In *ACM RecSys*, 2009.

[21] A. G. Parameswaran, H. Garcia-Molina, and J. D. Ullman. Evaluating, combining and generalizing recommendations with prerequisites. In *CIKM*, 2010.

[22] A. G. Parameswaran, G. Koutrika, B. Bercovitz, and H. Garcia-Molina. Recsplorer: Recommendation algorithms based on precedence mining. In *ACM SIGMOD*, 2010.

[23] S. Ray and A. Sharma. A collaborative filtering based approach for recommending elective courses. *CoRR*, abs/1309.6908, 2013.

[24] S. Rendle. Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22, May 2012.

[25] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *SIGIR*, 2011.

[26] N. Thai-nghe, L. Drumond, and L. Schmidt-thieme. Multi-relational factorization models for predicting student performance.

[27] N. ThaiNghe, T. Horváth, and L. Schmidt-Thieme. Factorization models for forecasting student performance. In *International Conference on Educational Data*, pages 11–20, 2011.

[28] K. Yu, B. Zhang, H. Zhu, H. Cao, and J. Tian. Towards personalized context-aware recommendation by mining context logs through topic models. In *PAKDD*, Lec Notes in Comp Sci, 2012.