2017 International Conference on Data Science and Advanced Analytics

Enriching Course-Specific Regression Models with Content Features for Grade Prediction

Qian Hu
Department of Computer Science
George Mason University
Fairfax, VA, USA
Email: qhu3@gmu.edu

Agoritsa Polyzou
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN, USA
Email: polyz001@umn.edu

George Karypis
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN, USA
Email: karypis@umn.edu

Huzefa Rangwala
Department of Computer Science
George Mason University
Fairfax, VA, USA
Email: rangwala@cs.gmu.edu

***Abstract*—An enduring issue in higher education is student retention and timely graduation. Early-warning and degree planning systems have been identified as a key approach to tackle this problem. Accurately predicting a student's performance can help recommend degree pathways for students and identify students at-risk of dropping from their program of study. Various approaches have been developed for predicting students' next-term grades. Recently, course-specific approaches based on linear regression and matrix factorization have been proposed. To predict a student's grade, course-specific approaches utilize the student's grades from courses taken prior to that course. However, there are a lot of factors other than student's historical grades that influence his/her performance, such as the difficulty of the courses, the quality and pedagogy of the instructor, the academic level of the students when taking the courses and so on. In this paper, we propose a course-specific regression model enriched with features about students, courses and instructors. Our proposed models were evaluated on datasets from two large public universities for academic programs with varying flexibility. The experimental results showed that incorporating content features can boost the performance of the course-specific model. For some degree programs with high flexibility, our experiments showed that predicting the grades with informative content features demonstrated better prediction accuracy.**

I. Introduction

The past few years have seen the rise of technologies that capture and leverage massive quantities of education-related data to deliver and improve all levels of learning and education in our society. The Department of Education Report [1] specifically highlighted the current successes of learning analytics and critical need for further research focused on development of robust applications that lead to better student outcomes, improved instructor pedagogy, enhanced curriculum and higher graduation rates for all students irrespective of their backgrounds from kindergarten through college. Currently, higher education institutions face a critical challenge of retaining students and ensuring their successful graduation [2]. Towards this end, several universities seek to deploy accurate and effective *degree planners* that assist students in choosing academic pathways towards a successful and timely graduation; and *early-warning systems* that aid academic advisors in identifying students who are at the risk of failing or dropping out of a program for timely intervention.

In this paper we present approaches that analyze in a systematic and careful manner, the large and diverse type of education-related data collected at two large public Universities with the objective of assisting students to make informed decisions about their future course selections. Specifically, we develop methods that perform next-term grade prediction i.e., predict the grade for students in future courses that they have not taken yet.

Course-specific models have been applied to predict student's next-term grades by using grades of prior courses, which better addresses pertinent challenges associated with the reliable estimation of the low-rank models [3]. However, course-specific models that use the grades of prior courses can only capture the information of student's knowledge evolution. Course-specific models also suffer from inaccurate prediction if the degree program is flexible (i.e., has several electives). In addition, there are some other factors that can influence student's grades, such as his/her academic level when taking a certain course, instructor's teaching quality and courses' difficulty. To solve this problem we incorporate content features, which can capture diverse information about students, courses and instructors. Based on course-specific models, we present a model which not only uses the grades of prior courses but also different kinds of content features.

We evaluated our proposed method on a dataset from George Mason University (GMU) collected from Fall 2009 to Spring 2016 and on a dataset from University of Minnesota (UMN) collected from Fall 2003 to Spring 2014. The results showed that our proposed method outperformed competing methods to some degree. Another finding was that when the prior-course information was sparse, the included content features were more likely to help. However, as the availability

978-1-5090-5004-8/17 $31.00 © 2017 IEEE
DOI 10.1109/DSAA.2017.74

of content features in the two universities is different, namely in GMU we have more informative content features, for majors with flexible degree programs in GMU, the course-specific model with content features achieves the best performance; for majors with flexible degree programs in UMN, the proposed course-specific model with grades of prior courses and content features performs better. This suggests that the availability of the content features can influence the performance of the proposed model.

The paper is organized as follows. Section 2 investigates the related work in the area of student's performance prediction. Section 3 describes the notations we used in the paper. Section 4 discusses our proposed method and other comparison methods. Section 5 is about protocol. In Section 6, we present our experimental results and analysis. The last section gives some conclusions and future direction.

II. Related Work

In recent years, data mining and machine learning techniques have been applied to improve educational quality including areas related to learning and content analytics [4], [5], knowledge tracing [6], [7], learning material enhancement [8] and early warning systems [9], [10]. A key problem in this area is that of predicting student performance at course activities, examinations, final grades or in terms of a student's GPA [11], [12], [13].

Various approaches have been developed in the context of intelligent tutors that model and predict the success or failure of a student in a specific task. Models such as regression [14], [15], [16], HMMs and bagged decision trees [17], collaborative filtering [18], matrix completion [19], [20], [21], and tensor factorization [22], [23] have been applied to this problem.

Based on the scope of this paper, we only review approaches for next-term student grade prediction in detail.

Knowing student's performance in advance can help instructors identify at-risk students early and advise them in choosing appropriate courses that fit their current knowledge state better. As such, several methods have been developed to tackle the next-term prediction problem. Most of the methods are inspired from recommender systems literature [24], [25], such as matrix factorization [3] and collaborative filtering [18], [26], [27]. Approaches based on standard classification approaches such as random forests trees have also been applied [28], [25]. A majority of the algorithms proposed are "one-size-fits-all", namely, trying to model all the students with one model. To model students with different characteristics, personalized grade prediction approaches have been proposed [29], [30]. Using features mined from student interaction with learning management systems, Elbadrawy *et. al.* proposed a personalized mutli-regression model [31] for in-class grade prediction. This was also extended to predict in-class assignment grades within the setting of Massive Open Online Courses [32].

Recently course-specific models proposed by Polyzou *et. al.* [3] achieved better prediction accuracy than existing approaches, assuming that students acquire knowledge in an cumulative manner. Course-specific models are cumulative, in the sense that to predict a student's grade in a target course, the students' grades from courses taken prior to the target course are utilized. Course-sepcific regression models cannot correctly capture students' knowledge state when the same knowledge can be acquired by taking different subsets of courses. To solve this problem, Morsy *et. al.* [33] developed Cumulative Knowledge-based Regression Model (CKRM), which represents the knowledge state of students in knowledge component vectors. In educational environment, the student-course enrollment patterns exhibit grouping structures which leads to not missing at random grade data (NMAR). To handle the NMAR characteristics of the grade data, Elbadrawy *et. al.* [34] proposed domain-aware grade prediction algorithms. Ren *et. al.* [35] proposed Matrix Factorization with Temporal Course-wise Influence (MFTI) algorithm which can capture the course-wise influence between courses.

However, one of the drawbacks of course-specific models is that they show poor performance if the degree program is flexible [3]. In addition, the grades of the prior courses cannot completely capture all the factors that affect students' performance. In this paper, based on course-specific models, we proposed a hybrid model to predict students' next-term performance by taking some informative factors into consideration.

III. Problem Formulation and Notation

Formally, we assume that we have records of n students and m courses, comprising a $n \times m$ sparse grade matrix $\mathbf{G}$, where $g_{s,c} \in [0-4]$ is the grade a student s earned in course c. The objective of next-term grade prediction problem is to estimate the grade $\hat{g}_{s,c}$, a student s will achieve in course c in the next term. Besides the grade matrix $\mathbf{G}$, we have information that can be associated with the student (e.g., academic level, previous GPA, major) and course offering (e.g., discipline, course level, prior courses frequently taken, instructor, etc) that can be combined to extract a feature vector per dyad. We denote this feature vector as $\mathbf{x}$ of p dimensions. As a convention, bold uppercase letters are used to represent matrices (e.g., $\mathbf{X}$) and bold lowercase letters represents vectors (e.g., $\mathbf{x}$).

IV. Methods

A. Course-Specific Regression with Prior Courses

Polyzou et.al. [3] motivate the use of course-specific regression models that leverage the sequential structure of undergraduate degree programs. These regression models assume that the performance of a student in a future course is strongly correlated with past performance on a subset of courses related to the degree program taken earlier. Specifically, this regression model estimates the grades for a future class as a sparse linear combination of grades obtained on prior courses. For a course c the grades that students obtained on courses taken prior to c are extracted from the grade matrix $\mathbf{G}$, and denoted by $\mathbf{G}_c^{pr}$. Each row of this matrix corresponds to

students that have taken the course c. Assume that n_c students have taken the course c so far and m_c represents the union set of courses taken by students prior to c, then the dimensions of $\mathbf{G}_c^{pr}$ is $n_c \times m_c$. $\mathbf{g}_{:,c}$ is the vector representing the grades that students obtained for course c. We learn the parameters of this Course-Specific Regression (CSR) model by solving the least square regression problem enforcing $\ell 1$ and $\ell 2$ norms. The optimization problem is given below:

$$min \underbrace{||\mathbb{1}w_{c0} + \mathbf{G}_c^{pr}\mathbf{w}_c^{pr} - \mathbf{g}_{:,c}||_2^2}_{\text{loss}} + \underbrace{\lambda_1||\mathbf{w}_c^{pr}||_2^2}_{\ell 2} + \underbrace{\lambda_2||\mathbf{w}_c^{pr}||_1}_{\ell 1} \tag{1}$$

where $\mathbb{1}$ is a vector of ones of dimension n_c, $\mathbf{w}_c^{pr} \in R^{m_c}$ denotes the weight vectors associated with each course c and $w_{c,0}$ is the bias term. The $\ell 1$ norm promotes sparsity and $\ell 2$ norm prevents overfitting.

Having learned the weight vectors and bias terms, the grade estimate for a student s enrolling in course c is given by:

$$\hat{g}_{s,c} = w_{c0} + \mathbf{x}_{s,c}^T\mathbf{w}_c^{pr} \tag{2}$$

where $\mathbf{x}_{s,c} \in \mathbb{R}_c^m$ is a feature vector representing the grades on prior courses that the student has taken so far. We denote this Course-Specific Regression model with Prior Courses as *CSR$_{PC}$*.

In this approach, prior to estimating the model using equation 1, we row-centered each row of matrix $\mathbf{G}_c^{pr}$ and $\mathbf{g}_{:,c}$, which is done by subtracting the GPA of corresponding students from the non-zero entries in each row of $\mathbf{G}_c^{pr}$ and $\mathbf{g}_{:,c}$ [3]. We found that row-centering gives better performance by mitigating the negative influence of missing grades from prior courses.

B. Course-Specific Regression with Content Features

The CSR$_{\text{PC}}$ model described above is able to provide accurate estimates of student performance in a course provided that the students taking that course has commonly taken sufficient number of prior courses. We seek to extract key features associated with students and courses and incorporate them within the prediction formulation. Based on course-specific idea, instead of training one global model for all the courses as done in existing work [25], we propose to train independent course-specific regression models with content features. We refer to this model by CSR$_{\text{CF}}$. In terms of formulation, the proposed CSR$_{\text{CF}}$ is similar to CSR$_{\text{PC}}$ except that the feature vector is a composite of student, course and instructor-related features as described below.

We denote the weight vector learned by this formulation as $\mathbf{w}_c^f$ and the feature vectors $\mathbf{x}_{s,c} \in \mathbb{R}^p$ where p is the total number of features. The predicted grade estimate is then given by:

$$\hat{g}_{s,c} = w_{c0} + \mathbf{x}_{s,c}^T\mathbf{w}_c^f \tag{3}$$

The CSR$_{\text{CF}}$ model is estimated in a similar manner as CSR$_{\text{PC}}$ and given by:

$$min \underbrace{||\mathbb{1}w_{c0} + \mathbf{X}_c^f\mathbf{w}_c^f - \mathbf{g}_{:,c}||_2^2}_{\text{loss}} + \underbrace{\lambda_1||\mathbf{w}_c^f||_2^2}_{\ell 2} + \underbrace{\lambda_2||\mathbf{w}_c^f||_1}_{\ell 1} \tag{4}$$

where $\mathbf{X}_c^f$ is a matrix of stacked feature vectors from the different students who have taken the course c in the past. Each row of this matrix is a feature vector for a student enrolled in the course c.

Content features for GMU

1) Student Features. Student-related features include their demographic data, such as their age, race, gender, high school GPA and so on. For each term, we have the GPA of the student from the previous term and the accumulative GPA as of last term. As students might take courses from other departments which has less influence than those from their own departments, we can extract GPA of courses only from their own departments. When taking a course, different students might come from different academic level, therefore, it might be beneficial to incorporate their academic level into the model.
2) Course Features. The features relating to a course include its discipline, the credit hours and course level (e.g. 100, 200, 300, 400-level). As the difficulty of a course can influence the performance of the students, we include the course difficulty information into the model. We use the GPA of the course from last term to represent the difficulty of the course.
3) Instructor Features. As the factors from instructors can also influence the performance of the students, we extract content features about the instructors which include rank, tenure status and the GPA of the courses he has taught.

Content features for UMN

1) Student Features. Same as in GMU apart from the features related to demographic data. Considering a specific term for which a student has taken a course, we extracted their GPA of the previous term, the accumulative GPA as of last term, the GPA over only courses from their own departments, as well as, the students' academic level.
2) Course Features. Same as the ones extracted for GMU.
3) Instructor Features. No instructor features are available.

We one-hot-encoded categorical features in $\mathbf{X}_c^f$ and standardized the continuous features.

C. Hybrid Model

We also combine the feature vectors $\mathbf{X}_c^f$ and $\mathbf{G}_c^{pr}$ obtained from the student-course content and prior grades and learn weight vectors per course, respectively. We refer to this hybrid model as CSR$_{\text{HY}}$ and learn a course-specific regression model as discussed above.

D. Baseline Methods

In the experiments, we compare the proposed methods with the following baseline approaches.

1) BiasOnly (BO): BiasOnly method only takes into consideration student's bias, course's bias and global bias which are estimated using Equation 5.

$$\hat{g}_{s,c} = b_0 + b_s + b_c \tag{5}$$

where b_0, b_s and b_c are the global bias, student bias and course bias respectively.

2) Matrix Factorization (MF): The use of MF for grade prediction is based on the assumption that the students and courses' knowledge space can be jointly represented in low-dimensional latent feature space [3]. Each component in the latent feature space corresponds to knowledge components. The grade of student s in a future course c is estimated as:

$$\hat{g}_{s,c} = b_0 + b_s + b_c + \boldsymbol{p}_s^T \boldsymbol{q}_c \tag{6}$$

where b_0, b_s and b_c are the global bias, student bias and course bias respectively and $\boldsymbol{p}_s$, $\boldsymbol{q}_c$ are the latent vectors representing student s and course c.

3) Course-specific Matrix Factorization (CS_{MF}): CS_{MF} is similar to MF except that the grade matrix $\mathbf{G}_c$ for CS_{MF} only includes the grades of students taking the course and their grades of courses taken prior to the course we are going to predict [3].

V. Experimental Protocol

A. Dataset description and preprocessing

We evaluated our proposed methods on two datasets obtained from George Mason University (GMU) and University of Minnesota (UMN), for the following four departments: (i) Computer Science (CS), (ii) Electrical and Computer Engineering (ECE), (iii) Biology (BIOL) and Psychology (PSYC). We will indicate the departments from GMU with the suffix "_A" and from UMN with the suffix "_B". The two universities from two separate states in the United States have different characteristics. For GMU, there are around 33,000 students, the acceptance rate is 69%, the six-year graduation rate is 66.8%, there are about 140 programs that students can select. For UMN, the total enrollment is about 51,000, acceptance rate is 45%, the six-year graduation rate is 75%, there are around 260 programs. Both universities exhibit diversity. In GMU, 44.7% of students are White, 18.5% Asian, 12% Hispanic/Latino, 10% African American. In UMN, 69.1% of the students are White, 11.3% Asian, 5.2% African American, 3.4% Latino.

The data was collected from Fall 2009 to Spring 2016 at GMU and from Fall 2003 to Spring 2014 at UMN. According to the University Catalogs [36] [37], we kept the courses that were required by the degree program and electives within the same major. The statistics of the four majors are shown in Table I.

For UMN that has very flexible degree programs, we also consider courses outside of the department that were taken by at least 50% of the students. We consider those as unstated prerequisites. Moreover, we removed any course that was taken by less than 10% of the students, in order to reduce the size of the universal of courses, i.e., the possible courses that a student might take. We consider that these courses are not offered on a regular basis and their availability is limited.

For both datasets, we removed any courses whose grades were pass/fail. If a course was taken more than once by a student, only the last grade was kept. We removed the students who took less than half of the prior courses (less than one third of the prior courses for UMN). For course c whose prior-course grade matrix is $\mathbf{G}_c^{pr}$, if the number of rows of $\mathbf{G}_c^{pr}$ is smaller than the number of columns, we remove course c from training and testing dataset. In addition to that, if the number of testing instances of a course is smaller than 5, we also remove it.

To form the test and training dataset, we use the data extracted from last term (i.e., Spring 2016 at GMU and Spring 2014 at UMN) as test dataset and all the data before then as training. The training dataset was split into 80/20, of which 80% was training data, 20% was validation data.

As the flexibility of a degree program can influence the course-specific models' performance, the flexibility associated with each department is computed according to [33]. The major's flexibility is the average course flexibility over all courses belonging to that major, weighted by the number of pairs of students in that offering. We computed the flexibility of a course c as one minus the average Jaccard coefficient of the courses that were taken by the students that took c prior to taking this course. The flexibility of a course will be low if the students have taken very similar prior courses and high otherwise.

To compute the flexibility of a major, assume there are N courses in that major; the prior-course grade matrices for these courses are denoted as $\mathbf{G}_i^{pr}, i = 1 \dots N$, each of which has $S_i, i = 1 \dots N$ students. From matrix $\mathbf{G}_i^{pr}$, we can extract an indicator matrix $\mathbf{I}_i^{pr}$, in which 1 means the corresponding course is taken, 0 means not. $\mathbf{R}_{i,a}$ means the ath row of matrix $\mathbf{I}_i^{pr}$.

$$F_i = 1 - \frac{1}{\binom{S_i}{2}} \sum_{a=1}^{S_i} \sum_{b=a+1}^{S_i} Jaccard(\mathbf{r}_{i,a}, \mathbf{r}_{i,b}) \tag{7}$$

$$F = \sum_{i=1}^{N} \frac{S_i}{S} F_i \tag{8}$$

where $Jaccard$ is the Jaccard coefficient, S is the total number of students in that major, F_i is the flexibility of course i and F is the flexibility of the major.

B. Evaluation Metrics

To assess the performance of the models, we used three kinds of metrics, namely mean absolute error (MAE), root mean squared error (RMSE) and tick error. MAE and RMSE are computed by pooling together all the grades across all the courses.

MAE and RMSE are averaged errors between the predicted grades and the actual grades. To gain a better insight into the quality of the predictions, we also report the tick error as done in [3], [33]. The grading system used in GMU has 11 letter grades (A+, A, A-, B+, B, B-, C+, C, C-, D, F) which correspond to (4, 4, 3.67, 3.33, 3, 2.67, 2.33, 2, 1.67, 1, 0). UMN uses the same grading, with the addition of D+, corresponding to 1.33, and excluding A+. We refer to the difference between two successive letter grades as a tick. The performance of a model is assessed based on how many ticks away the predicted grade is from the actual grade. We first

TABLE I: Data Statistics and Characteristics for GMU and UMN.

Major	#Students	#Courses	Universal of courses	#Grades	Grades Mn	Grades StD	Flexibility
CS_A	988	18	53	21,880	3.05	0.82	0.283
ECE_A	396	16	69	16,170	3.09	0.77	0.272
BIOL_A	1629	19	42	20,602	3.02	0.84	0.339
PSYC_A	1114	20	60	14,851	3.26	0.74	0.429
CS_B	708	24	39	78,882	3.15	0.71	0.493
ECE_B	551	16	44	86,478	3.12	0.72	0.430
BIOL_B	997	11	31	57,966	3.12	0.74	0.603
PSYC_B	1380	18	37	77,896	3.07	0.82	0.809

#Students is the number of major students.
#Courses is the number of courses for which we predict the grades.
Universal of courses is the total number of prior courses, i.e., the required and elective courses in the corresponding major according to university catalog.
#Grades is the total number of grades in prior-course grade matrices and the grades we predict.
Grades Mn and Grades StD are the mean and standard deviation of grades, respectively.
Flexibility is the flexibility of a major.

TABLE II: MAE of different methods (↓ is better).

Method	MAE							
	CS_A	ECE_A	BIOL_A	PSYC_A	CS_B	ECE_B	BIOL_B	PSYC_B
BO	0.7359	0.7285	0.5853	0.5882	0.4697	0.4356	0.4516	0.4648
MF	0.8150	0.8447	0.6169	0.5648	0.4859	0.4309	0.4452	0.4940
CS_{MF}	0.7609	0.7015	0.5579	0.5240	0.4776	0.4433	0.4410	0.4695
CSR_{PC}	0.6805	0.6739	0.5372	0.4933	**0.4520**	0.4346	0.4394	0.4932
CSR_{CF}	0.7183	0.6775	**0.4769**	**0.4743**	0.4670	0.4395	0.4488	0.4588
CSR_{HY}	**0.6693**	**0.6630**	0.5057	0.4859	0.4622	**0.4219**	**0.4328**	**0.4526**

TABLE III: RMSE of different methods (↓ is better).

Method	RMSE							
	CS_A	ECE_A	BIOL_A	PSYC_A	CS_B	ECE_B	BIOL_B	PSYC_B
BO	0.9622	0.9748	0.7794	0.7829	**0.6534**	0.5359	0.5855	0.6180
MF	1.0879	1.1104	0.8173	0.8035	0.6773	0.5408	0.5922	0.6574
CS_{MF}	1.0126	0.9623	0.8045	0.7372	0.6685	0.5472	0.5763	0.6318
CSR_{PC}	0.9288	0.9699	0.7943	0.7348	0.6613	0.5447	0.5679	0.6351
CSR_{CF}	0.9539	0.9680	**0.7205**	**0.6732**	0.6543	0.5457	0.5825	0.6064
CSR_{HY}	**0.9199**	**0.9542**	0.7679	0.7283	0.6607	**0.5298**	**0.5659**	**0.5946**

converted the predicted grades into their closest letter grades and then computed the percentages of each of the x ticks [3], [33].

VI. Results and Discussion

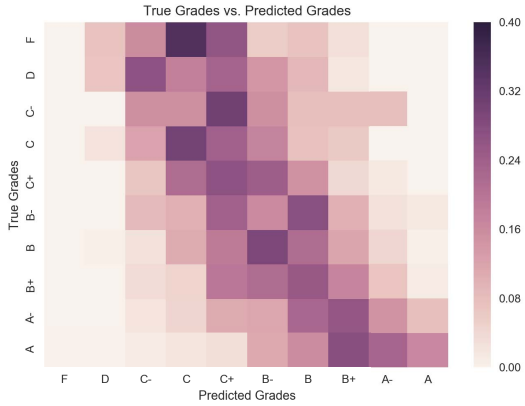
Tables II and III show the comparative performance of different methods on four different departments by using metrics MAE and RMSE. Generally, in most cases course-specific models outperform non-course-specific models, which means focusing on a course-specific subset of data can result in better performance. In GMU, for departments with less flexibility such as Computer Science and Electrical Engineering, we observe that the hybrid model has the best performance. Thus incorporating content features into course-specific model further improves its performance; the model with only grades of prior courses performs better than model with only content features. For departments with high flexibility such as Biology and Psychology, the model with only content features shows the best performance, which suggests that if a department has a flexible degree program, content features might be more informative than the grades of prior courses.

The corresponding departments in UMN are more flexible than GMU. The performance of CSR_{PC} and CSR_{CF} is very comparable, or even better (for the Psychology Department). Their combination, CSR_{HY}, is the best performing method in terms of MAE and RMSE, even if the content features included in UMN are less informative. An exception is the Computer Science Department, which seams to have very hard-to-predict courses, as it has the highest RMSE. For CS_B, CSR_{PC} is performing the best in terms of MAE, but BiasOnly achieves better RMSE, closely followed by CSR_{CF}, with just 0.0009 difference.

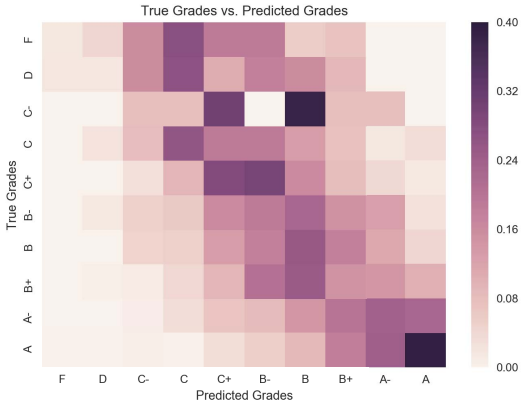
In the two universities, we can see that for the majority of the departments, the hybrid model performs the best. GMU models take more advantage of the rich content features to

TABLE IV: Prediction performance of different methods based on Ticks (↑ is better).

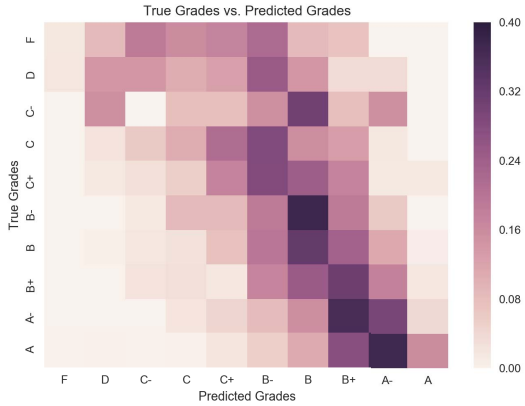
#Ticks	Method	CS_A	ECE_A	BIOL_A	PSYC_A	CS_B	ECE_B	BIOL_B	PSYC_B
Percentage of Grades predicted with no error	BO	15.02	18.58	19.41	19.75	25.48	27.58	24.90	34.40
	MF	13.04	9.84	19.95	23.89	26.68	28.48	24.90	31.91
	CS_{MF}	15.22	18.58	24.53	23.25	24.76	**29.09**	**30.12**	**34.75**
	CSR_{PC}	19.57	20.77	28.84	34.08	**29.33**	26.06	25.70	23.76
	CSR_{CF}	13.44	16.39	28.03	27.39	25.96	28.48	25.30	31.91
	CSR_{HY}	**19.76**	**22.40**	**30.73**	**35.35**	25.00	28.18	29.32	30.50
Percentage of grades predicted with an error of at most one tick	BO	44.27	44.26	55.26	53.82	65.38	66.36	61.85	65.60
	MF	42.29	39.34	51.75	54.46	63.70	66.67	**65.06**	62.77
	CS_{MF}	43.08	40.44	58.76	61.78	63.94	64.85	**65.06**	**68.44**
	CSR_{PC}	48.22	**55.19**	62.80	61.15	**69.23**	64.85	62.65	57.45
	CSR_{CF}	44.66	51.37	**70.89**	**64.97**	64.42	66.67	63.05	**68.44**
	CSR_{HY}	**49.80**	**55.19**	67.38	61.78	68.03	**66.97**	64.66	66.31
Percentage of grades predicted with an error of at most two ticks	BO	69.17	66.67	77.63	75.80	86.54	**89.09**	87.15	**88.65**
	MF	64.82	63.38	76.82	77.07	82.69	88.79	86.75	83.69
	CS_{MF}	67.59	72.68	82.21	78.66	85.34	**89.09**	86.75	85.11
	CSR_{PC}	74.31	73.22	81.40	79.62	**87.26**	85.76	**88.35**	83.69
	CSR_{CF}	73.52	**75.96**	**87.87**	**83.44**	86.06	88.79	85.54	87.94
	CSR_{HY}	**75.10**	74.32	82.75	78.66	85.82	88.18	86.35	86.88



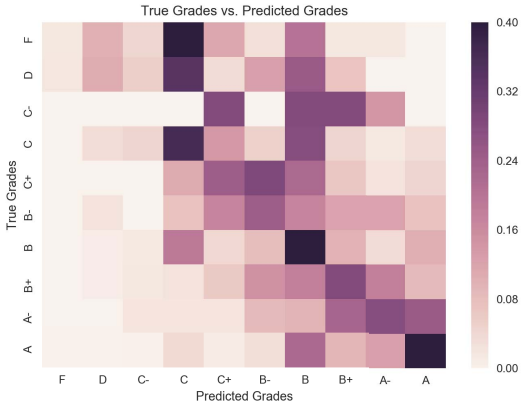
(a) True vs. Predicted Grades for BO



(b) True vs. Predicted Grades for CSR_{PC}

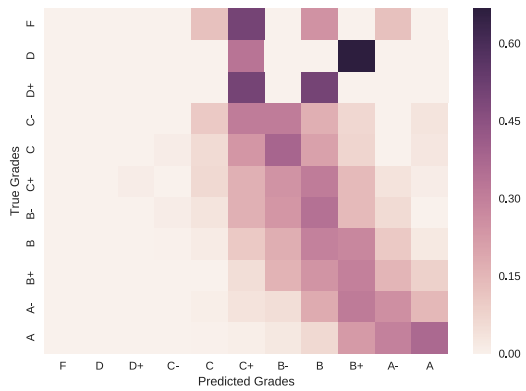


(c) True vs. Predicted Grades for CSR_{CF}

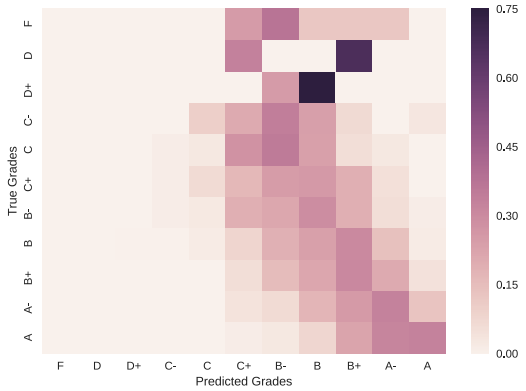


(d) True vs. Predicted Grades for CSR_{HY}

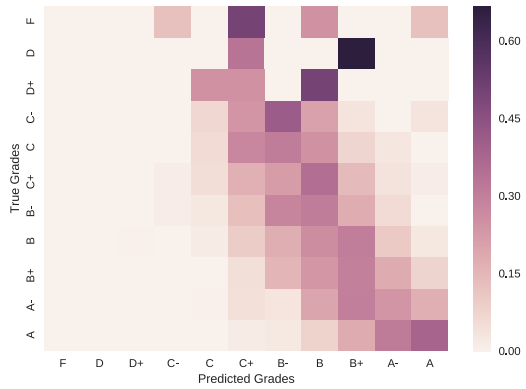
Fig. 1: True vs. Predicted Grades for BiasOnly and Course-specific Models for GMU.



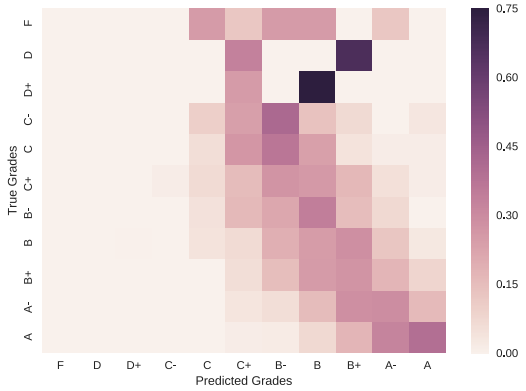
(a) True vs. Predicted Grades for BO



(b) True vs. Predicted Grades for CSR_{PC}



(c) True vs. Predicted Grades for CSR_{CF}



(d) True vs. Predicted Grades for CSR_{HY}

Fig. 2: True vs. Predicted Grades for BiasOnly and Course-specific Models for UMN.

improve the predicted grades, especially for the most flexible departments.

To gain deeper insights into the types of errors made by different methods, Table IV reports the percentage of grades predicted with no error, with an error of at most one tick and with an error of at most two ticks. Comparing the performance achieved by the methods we notice that the course-specific models have relatively better performance than non-specific approaches. In GMU, in terms of the exact prediction (i.e., no error), the hybrid model has the best performance. For departments with rigid degree program, such as Computer Science and Electrical Engineering, the hybrid model has better performance than other methods. If minor errors are allowed (i.e., one or two ticks), for flexible departments, model with only content features gives better performance. In UMN, the picture is not that clear, as there is variation in the performance depending on the degree of accuracy and the department. The highest percentage of grades predicted with no error is achieved by course-specific methods(CS_{MF} and CSR_{PC}). The fact that other methods are the best performing in terms of ticks, while CSR_{HY} has the lowest RMSE for most of the cases, indicates that CSR_{HY} does not predict many grades with significant error, in contrast with the other methods.

From the two universities' results, we can see that incorporating content features into the course-specific model can improve the prediction performance. For flexible degree programs, as the prior-course grade matrix is sparse, the model with content features has better predicting accuracy. This is not evident in the results from UMN, as there are not enough content features.

The distribution of true (ground truth) and predicted grades for BiasOnly, CSR_{PC}, CSR_{CF} and CSR_{HY} are also plotted for GMU and UMN in Figures 1 and 2, respectively. Each row of the figure represents the ratio of the predicted grades. For example, in Figure 1b the bottom row represents that a high proportion of A's are predicted as such. We see that BiasOnly tends to smooth the predicted grades i.e., it predicts most of the grades around the average GPA (around B-). However, for high grades most of the predicted grades are around the true grades in course-specific models and for lower grades all the

models tend to over predict.

Table V and VI show the detailed statistics of the courses from the two universities of the departments with the least and most flexible degree program, and the errors (RMSE) made by three course-specific regression models. For GMU these departments are the CS and PSYC, while for UMN are the EE and PSYC. From the two tables, we can see that if the grades in test set have high standard deviation or higher than that of training set, the prediction error is high. The reason might be that the course-specific models used in this work and previous works are linear. In the future, we will explore non-linear course-specific models.

Overall, incorporating content features into the course-specific models can improve the prediction performance. In GMU, for departments with less flexible degree programs, the hybrid model achieves better performance than traditional course-specific models. However, for departments with more flexible degree programs, the grades of prior courses are less informative than content features, therefore, it is more appropriate to include only content features. In UMN, CSR_{HY} achieves the best performance. The existance of some content features can boost the performance of the regression methods when used alone(CSR_{CF}) or in addition to the grades(CSR_{HY}).

VII. Conclusions

In this paper, we proposed a hybrid model to further improve the performance of the course-specific models. We evaluated the proposed model on datasets from two Universities with different characteristics. The experiments show similar results in the two universities, which suggests the proposed model is generalizable. In conclusion, it is beneficial to incorporate content features into course-specific model, which motivates us to explore other kinds of side information. In the future, we will utilize side information mined from learning management systems.

Acknowledgement

This work is partialy supported by the National Science Foundation grant #1447489.

References

[1] M. Bienkowski, M. Feng, and B. Means, "Enhancing teaching and learning through educational data mining and learning analytics: An issue brief," *US Department of Education, Office of Educational Technology*, pp. 1–57, 2012.

[2] R. Stillwell and J. Sable, "Public school graduates and dropouts from the common core of data: School year 2009-10. first look (provisional data). nces 2013-309." *National Center for Education Statistics*, 2013.

[3] A. Polyzou and G. Karypis, "Grade prediction with models specific to students and courses," *International Journal of Data Science and Analytics*, pp. 1–13, 2016.

[4] A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk, "Sparse factor analysis for learning and content analytics." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1959–2008, 2014.

[5] A. S. Lan, C. Studer, and R. G. Baraniuk, "Time-varying learning and content analytics via sparse factor analysis," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 452–461.

[6] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon, "Individualized bayesian knowledge tracing models," in *International Conference on Artificial Intelligence in Education*. Springer, 2013, pp. 171–180.

[7] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User modeling and user-adapted interaction*, vol. 4, no. 4, pp. 253–278, 1994.

[8] R. Agrawal, M. Christoforaki, S. Gollapudi, A. Kannan, K. Kenthapadi, and A. Swaminathan, "Mining videos from the web for electronic textbooks," in *International Conference on Formal Concept Analysis*. Springer, 2014, pp. 219–234.

[9] L. P. Macfadyen and S. Dawson, "Mining lms data to develop an early warning system for educators: A proof of concept," *Computers & education*, vol. 54, no. 2, pp. 588–599, 2010.

[10] H. P. Beck and W. D. Davidson, "Establishing an early warning system: Predicting low grades in college students from survey of academic orientations scores," *Research in Higher education*, vol. 42, no. 6, pp. 709–723, 2001.

[11] E. Osmanbegović and M. Suljić, "Data mining approach for predicting student performance," *Economic Review*, vol. 10, no. 1, 2012.

[12] A. Ogunde and D. Ajibade, "A data mining system for predicting university students? graduation grades using id3 decision tree algorithm," *Journal of Computer Science and Information Technology*, vol. 2, no. 1, pp. 21–46, 2014.

[13] M. A. Al-Barrak and M. Al-Razgan, "Predicting students final gpa using decision trees: A case study," *International Journal of Information and Education Technology*, vol. 6, no. 7, p. 528, 2016.

[14] J. E. Beck and B. P. Woolf, "High-level student modeling with machine learning," in *International Conference on Intelligent Tutoring Systems*. Springer, 2000, pp. 584–593.

[15] M. Feng, N. T. Heffernan, and K. R. Koedinger, "Looking for sources of error in predicting student?s knowledge," in *Educational Data Mining: Papers from the 2005 AAAI Workshop*, 2005, pp. 54–61.

[16] H. Cen, K. Koedinger, and B. Junker, "Learning factors analysis–a general method for cognitive model evaluation and improvement," in *Intelligent tutoring systems*. Springer, 2006, pp. 164–175.

[17] Z. A. Pardos and N. T. Heffernan, "Using hmms and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset," *Journal of Machine Learning Research W & CP*, 2010.

[18] A. Toscher and M. Jahrer, "Collaborative filtering applied to educational data mining," *KDD cup*, 2010.

[19] N. Thai-Nghe, L. Drumond, A. Krohn-Grimberghe, and L. Schmidt-Thieme, "Recommender system for predicting student performance," *Procedia Computer Science*, vol. 1, no. 2, pp. 2811–2819, 2010.

[20] N. Thai-Nghe, L. Drumond, T. Horváth, and L. Schmidt-Thieme, "Using factorization machines for student modeling." in *UMAP Workshops*, 2012.

[21] C.-S. Hwang and Y.-C. Su, "Unified clustering locality preserving matrix factorization for student performance prediction." *IAENG International Journal of Computer Science*, vol. 42, no. 3, 2015.

[22] N. Thai-Nghe, T. Horváth, and L. Schmidt-Thieme, "Factorization models for forecasting student performance," in *Educational Data Mining 2011*, 2010.

[23] F. M. Almutairi, N. D. Sidiropoulos, and G. Karypis, "Context-aware recommendation-based learning analytics using tensor and coupled matrix factorization," *IEEE Journal of Selected Topics in Signal Processing*, 2017.

[24] M. Sweeney, J. Lester, and H. Rangwala, "Next-term student grade prediction," in *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015, pp. 970–975.

[25] M. Sweeney, H. Rangwala, J. Lester, and A. Johri, "Next-term student performance prediction: A recommender systems approach," *arXiv preprint arXiv:1604.01840*, 2016.

[26] H. Bydžovská, "Are collaborative filtering methods suitable for student performance prediction?" in *Portuguese Conference on Artificial Intelligence*. Springer, 2015, pp. 425–430.

[27] A. Cakmak, "Predicting student success in courses via collaborative filtering," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 5, no. 1, pp. 10–17, 2017.

[28] M. A. Al-Barrak and M. Al-Razgan, "Predicting students final gpa using decision trees: a case study," *International Journal of Information and Education Technology*, vol. 6, no. 7, p. 528, 2016.

[29] Y. Meier, J. Xu, O. Atan, and M. van der Schaar, "Personalized grade prediction: A data mining approach," in *Data Mining (ICDM), 2015 IEEE International Conference on*. IEEE, 2015, pp. 907–912.

TABLE V: Per course statistics and errors for GMU.

Course	#training	#testing	density	Mn Tr	StD Tr	Mn Te	StD Te	CSR_{PC}	CSR_{CF}	CSR_{HY}
CS-2xx	322	76	0.766	2.640	1.249	2.548	1.455	1.179	1.226	1.176
CS-2xx	303	66	0.623	2.915	1.062	2.899	0.941	0.686	0.755	0.735
CS-3xx	138	19	0.748	3.049	0.803	3.158	0.597	0.463	0.417	0.434
CS-3xx	285	62	0.638	2.634	1.155	2.694	1.236	1.037	1.156	1.037
CS-3xx	181	41	0.711	3.063	0.779	3.041	0.617	0.527	0.465	0.539
CS-3xx	42	13	0.802	3.104	1.140	3.360	0.591	0.748	0.668	0.752
CS-3xx	189	35	0.754	2.783	1.032	2.657	1.053	0.876	0.949	0.876
CS-3xx	19	8	0.885	2.719	1.072	2.959	1.368	1.152	1.035	1.253
CS-3xx	156	29	0.768	3.088	0.762	2.897	1.175	1.072	1.045	1.066
CS-4xx	92	8	0.867	2.859	1.103	2.917	1.090	1.006	1.119	1.006
CS-4xx	29	15	0.868	2.426	1.181	2.311	1.341	1.243	0.972	0.830
CS-4xx	35	7	0.378	2.667	0.983	2.713	0.629	0.711	0.609	0.736
CS-4xx	105	36	0.909	3.137	0.810	3.297	0.965	0.951	0.913	0.994
CS-4xx	43	10	0.912	2.923	1.001	2.567	1.383	1.072	1.063	1.042
CS-4xx	46	19	0.896	2.725	1.111	1.983	1.111	1.090	1.081	1.143
CS-4xx	32	8	0.897	3.083	0.866	3.041	1.207	0.964	1.106	0.964
CS-4xx	115	32	0.868	3.018	0.914	3.229	0.659	0.655	0.643	0.655
CS-4xx	26	22	0.868	3.525	0.668	3.333	0.841	0.669	0.870	0.610
PSYC-2xx	195	24	0.608	3.165	0.802	3.639	0.429	0.709	0.604	0.694
PSYC-2xx	204	23	0.635	3.144	0.726	3.435	0.788	0.678	0.746	0.678
PSYC-3xx	247	23	0.670	3.263	0.813	3.580	0.654	0.796	0.656	0.799
PSYC-3xx	223	24	0.724	3.262	0.870	3.390	0.875	0.759	0.578	0.756
PSYC-3xx	44	5	0.825	3.212	0.943	3.600	0.490	0.507	0.829	0.653
PSYC-3xx	112	8	0.613	3.310	0.858	3.292	0.715	0.878	0.726	0.873
PSYC-3xx	86	7	0.558	3.535	0.758	3.620	0.516	0.696	0.467	0.678
PSYC-3xx	258	21	0.586	3.263	0.936	3.778	0.428	0.760	0.801	0.728
PSYC-3xx	69	14	0.718	3.251	0.667	3.357	0.672	0.481	0.475	0.467
PSYC-3xx	227	26	0.687	3.333	0.728	3.270	0.883	0.776	0.729	0.776
PSYC-3xx	94	9	0.723	3.394	0.617	3.630	0.618	0.600	0.521	0.602
PSYC-3xx	52	6	0.714	3.378	0.911	3.280	1.027	0.978	1.033	0.956
PSYC-3xx	216	22	0.731	3.048	0.951	2.803	1.013	0.940	0.642	0.940
PSYC-3xx	66	12	0.710	3.525	0.802	3.168	0.977	1.020	0.865	1.021
PSYC-3xx	121	18	0.715	3.488	0.705	3.371	0.745	0.692	0.627	0.700
PSYC-4xx	182	21	0.672	3.564	0.716	3.540	0.442	0.549	0.346	0.550
PSYC-4xx	48	5	0.789	3.771	0.409	4.000	0.000	0.424	0.253	0.424
PSYC-4xx	105	30	0.661	3.445	0.884	3.778	0.489	0.588	0.627	0.590
PSYC-4xx	34	12	0.798	3.657	0.521	3.112	0.736	0.809	0.893	0.802

The second and third column stand for the number of training and testing instances, respectively
density means the density of the prior course matrix
Tr train, *Te* test, *Mn* mean, *StD* standard deviation

[30] M. Sheehan and Y. Park, "pgpa: a personalized grade prediction tool to aid student success," in *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 2012, pp. 309–310.

[31] A. Elbadrawy, R. S. Studham, and George Karypis, "Collaborative multi-regression models for predicting students' performance in course activities," *LAK, '15*, 2015.

[32] Z. Ren, H. Rangwala, and A. Johri, "Predicting performance on mooc assessments using multi-regression models," *arXiv preprint arXiv:1605.02269*, 2016.

[33] S. Morsy and G. Karypis, "Cumulative knowledge-based regression models for next-term grade prediction," 2017.

[34] A. Elbadrawy and G. Karypis, "Domain-aware grade prediction and top-n course recommendation," *Boston, MA, Sep*, 2016.

[35] Z. Ren, X. Ning, and H. Rangwala, "Grade prediction with temporal course-wise influence," 2017.

[36] GMU, "George mason university catalog," 2017. [Online]. Available: http://catalog.gmu.edu/

[37] UMN, "University of minnesota twin cities undergraduate catalog," 2017. [Online]. Available: http://www.catalogs.umn.edu/ug/index.html

TABLE VI: Per course statistics and errors for UMN.

Course	#training	#testing	density	Mn Tr	StD Tr	Mn Te	StD Te	CSR_{PC}	CSR_{CF}	CSR_{HY}
EExxx	514	22	0.441	2.82	0.73	2.95	0.70	0.603	0.547	0.544
EExxx	511	32	0.450	3.59	0.51	3.44	0.37	0.520	0.664	0.577
EExxx	540	5	0.352	2.88	0.67	2.73	0.25	0.454	0.393	0.419
EExxx	516	28	0.443	2.94	0.68	2.98	0.60	0.562	0.543	0.532
EExxx	520	21	0.405	3.05	0.67	2.84	0.79	0.574	0.589	0.573
EExxx	142	7	0.582	2.83	0.94	2.81	0.24	0.712	0.409	0.599
EExxx	88	31	0.837	3.27	0.63	3.10	0.69	0.559	0.585	0.568
EExxx	146	32	0.631	3.08	0.78	3.04	0.64	0.525	0.449	0.529
EExxx	51	13	0.587	3.86	0.28	3.95	0.18	0.378	0.494	0.383
EExxx	189	20	0.610	2.74	0.81	2.88	0.78	0.548	0.545	0.573
EExxx	225	11	0.576	3.11	0.75	3.12	0.94	0.825	0.835	0.833
EExxx	101	22	0.684	3.06	0.70	2.55	0.56	0.572	0.582	0.579
EExxx	331	29	0.558	3.24	0.63	3.47	0.54	0.445	0.416	0.419
EExxx	239	23	0.556	3.84	0.27	3.91	0.15	0.581	0.414	0.394
EExxx	407	26	0.655	3.65	0.43	3.88	0.45	0.486	0.585	0.485
EExxx	65	8	0.670	3.89	0.37	3.92	0.14	0.149	0.344	0.090
PSYCxxx	1031	18	0.207	3.30	0.59	3.26	0.57	0.452	0.429	0.433
PSYCxxx	464	7	0.259	3.21	0.83	3.14	0.59	1.027	0.998	1.023
PSYCxxx	444	10	0.263	2.90	0.80	3.17	0.43	0.733	0.693	0.784
PSYCxxx	606	17	0.261	3.21	0.77	3.24	0.72	0.490	0.509	0.510
PSYCxxx	557	18	0.254	2.97	0.87	3.48	0.92	0.795	0.794	0.802
PSYCxxx	488	13	0.220	3.03	0.89	3.56	0.48	0.430	0.495	0.438
PSYCxxx	34	12	0.482	2.80	0.90	3.42	0.71	0.873	0.870	0.867
PSYCxxx	399	12	0.259	3.13	0.79	3.39	0.45	0.512	0.389	0.375
PSYCxxx	288	13	0.261	2.97	0.79	2.95	0.76	0.468	0.468	0.485
PSYCxxx	554	7	0.271	3.35	0.67	3.48	0.43	0.471	0.629	0.538
PSYCxxx	743	13	0.162	3.17	0.78	2.87	0.78	0.626	0.812	0.650
PSYCxxx	346	9	0.268	3.30	0.78	2.74	0.91	0.676	0.699	0.705
PSYCxxx	301	10	0.229	3.46	0.59	3.67	0.42	0.907	0.679	0.684
PSYCxxx	366	5	0.276	3.22	0.82	3.07	0.44	0.593	0.660	0.618
PSYCxxx	1045	80	0.354	3.56	0.57	3.70	0.46	0.648	0.601	0.590
PSYCxxx	258	7	0.288	3.90	0.47	4.00	0.00	0.466	0.392	0.343
PSYCxxx	121	5	0.274	3.96	0.16	4.00	0.00	0.194	0.271	0.166
PSYCxxx	290	26	0.320	3.93	0.33	4.00	0.00	0.562	0.341	0.351

The second and third column stand for the number of training and testing instances, respectively.
density means the density of the prior course matrix.
Tr train, *Te* test, *Mn* mean, *StD* standard deviation.