Similarity Based Regularization for Online Matrix-Factorization Problem: An Application to Course Recommender Systems

Dhruv Shah
Tata Consultancy Services
Pune, India
s.dhruv@tcs.com

Pratik Shah
IIIT
Vadodara, India
pratik@iiitvadodara.ac.in

Asim Banerjee
DA-IICT
Gandhinagar, India
asim.banerjee@daiict.ac.in

***Abstract*—The design of a recommender system is largely influenced by its domain of application. A recommender system for niche application requires more accuracy as it targets a specific audience or a specific genre of products to recommend. Certain examples of niche domains include course recommendation for university courses, text recommendation for translators etc. In this paper, we address the problem of designing a recommender system for one such niche domain, a course recommender system. It generates recommendation of university courses for students based on the courses previously preferred by the student. Since such recommendations play a role similar to decision support systems for students, it is evident that it has to be relevant in predicting preferences. Also, every new choice made by the user unfolds additional information about a user which was previously unknown to the system. Literature suggests that course recommender systems have been developed mostly without the use of machine learning techniques. Treating student preference information as a general recommendation problem, it can be represented in a matrix format and generating a low-rank matrix representation have provided encouraging results. Inclusion of additional information in order to make more accurate predictions, leads to higher computational complexity and instability in learning parameters. To overcome such hurdles in designing a Course Recommender System, we propose a similarity based regularization for low-rank matrix factorization algorithm which learns the prediction matrix very fast and is stable.**

***Index Terms*—Course Recommender System, Recommender System, Low-rank matrix factorization, Online Matrix Completion**

I. Introduction

Courses that are offered in a University, are designed to cater to, a student's interest and industry's demand. Most of the time, student's choices are influenced by peer reviews. Such an influence on decisions made by students cannot be counted as a standard approach to choose subjects, but such decisions have stronger impacts on their future achievements and motivations. A standard decision support system, takes into consideration all the available information and processes it to generate a sensible and meaningful decision. In case of a course recommender system, there are many parameters that affect a student's decision. Figure 1 shows the different parameters that influence the choice of a student.

Recommender systems are popular and effective tools for information filtering and seem very suitable for such tasks

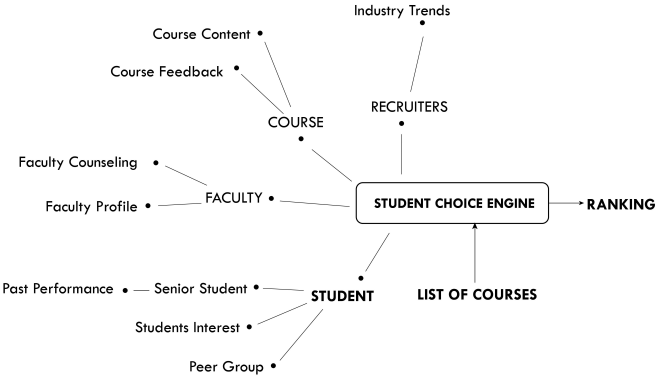

Fig. 1. Student's Choice Engine

of decision support [10]. In general, a Course Recommender System typically refers to a utility tool, which generates course recommendations, based on the performance of the student in other courses and also the performances of the other students in different courses.

Recommender systems have become an important part of the information and e-commerce ecosystem [8]. They represent a powerful method for enabling users to filter through a large amount of information and products. Course recommender system is a special case of a general recommender system. The users of this system are students and items to be recommended are courses. In this case, our objective is to predict student's performance in different courses that are yet to be offered and recommend courses based on suitable criteria. This requires the system to understand and learn the factors that affect the performance of a student in each course. Recommender systems for an educational framework has to be accurate and relevant in predicting student performance based on his past performance and other factors.

In this paper, we present a machine learning approach for predicting student's performance in different courses, by applying a popular approach used for prediction in recommender systems. Also, we propose a regularization technique

978-1-5090-1134-6/17/$31.00 ©2017 IEEE

for overcoming the problem of over-fitting while learning.

A. Past Work

Course recommender systems are being used at institutes around the world. Web-based applications for the same have been reported. A widely reported approach in designing course recommender systems involve *pattern discovery using association rule based approach* [13]. A machine learning approach has been reported in [7] which involves use of singular value decomposition (SVD) technique for predicting student performance. This approach requires representing the student performance in various courses as an incomplete matrix.

Matrix-completion techniques have been extensively employed for predicting user preferences in a recommender system. Use of such techniques for predicting student performance in a course recommender system have not been reported. Singular value decomposition algorithm presented in [7], involves decomposing the originally incomplete matrix into factors and approximate the unknown entries of matrix using eigen vectors corresponding to highest eigen vectors and approximating all other eigen values to 0. Mathematically, for an incomplete matrix X of dimensions $n \times m$, its SVD decomposition can be shown as :

$$X = U\Sigma V^T \tag{1}$$

Here U and V are of dimensions $n \times k$ and $m \times k$ respectively. Σ is a diagonal matrix containing eigen values of incomplete matrix X in decreasing order along the diagonal. The problem with this method is that it is generic and does not involve use of any information apart from student's performance. There are many factors which affect student's performance, such as course work, topics covered, student's interest etc. Thus, it is very important to learn and use those factors to generate accurate and relevant predictions.

Our approach involves factorizing the original incomplete student performance matrix and learning the factors that affect student's performance based on available information and generating a prediction matrix from the factors learnt. In the next section, we give an introduction to matrix factorization technique and its online version. Section 3 describes the proposed algorithm and the regularization technique followed by Section 4 on performance of proposed approach over two original datasets obtained from two universities, namely DA-IICT and IIIT-V. We end with conclusions in Section 5.

B. Notations

We define the course recommender problem, by using A as set of students and B as set of courses. Total number of students and courses are represented using n and m respectively. Original performance of a student i in course j is represented by r_{ij} and predicted performance is represented by $\hat{X}_{ij}$.

II. Problem Formulation

We address the problem of performance prediction for a course recommender system by representing the student performance as an incomplete matrix. Thus, the problem now gets converted into a matrix completion problem. Most of the algorithms for matrix completion involve batch-processing, where matrix gets updated every-time an entry of matrix becomes available [4]. Such models, are computationally and memory-wise very costly. A novel solution to such problems is presented in [2]–[4] which focus on developing an online-processing model for learning predictions. Such models are computationally efficient, since their update schedule involves less number of computations as compared to the batch processing model.

A. Low-Rank Matrix Factorization

Low-rank Matrix factorization technique involves decomposing and approximating the original matrix into factors and learning the decomposed factors based on the available entries of the incomplete matrix. For an incomplete matrix r_{ij} of size $n \times m$, suppose X represents its low-rank approximation. Then, given a rank k, matrix X can be represented as a product of a $n \times k$ matrix U and a $m \times k$ matrix V. Mathematically, this can be expressed as:

$$X = UV^T \tag{2}$$

Since, UV^T represent the compressed low-rank approximation of original incomplete matrix r_{ij}, our main objective is to minimize the error between both these matrices, by using the available entries of r_{ij}. The objective of such a low-rank approximation is [4]:

$$\underset{U \in \mathbb{R}^{n \times k}\ V \in \mathbb{R}^{m \times k}}{\operatorname{argmin}} \|r_{ij} - U.V^T\|_{\otimes}^2 \tag{3}$$

Here $\|X\|_{\otimes}$ represents Frobenius norm of the matrix X, also known as matrix norm i.e $\sum_i^j X_{ij}^2$. The authors in [4] suggest that in cases where the matrix r_{ij} is partially observed, SVD algorithm is not a suitable option. However, an optimal low-rank approximation can still be achieved using the available entries of matrix r_{ij}. If u_i and v_j represent i^{th} row and j^{th} row of U and V matrices respectively, then given a set of partially observed entries in r_{ij}, an optimization problem can be expressed as:

$$\underset{U \in \mathbb{R}^{n \times k} V \in \mathbb{R}^{m \times k}}{\operatorname{argmin}} \sum_{(i,j) \in \Omega} (u_i v_j^T - r_{ij})^2 \tag{4}$$

Here Ω represents the set of ordered pairs of (i, j) for which observations are available. The above optimization takes into consideration only the student's performance in terms of grades. The advantages of using additional information in predicting user preferences have been described and studied extensively in [5], [10], [15], [17]. Thus, the above problem can be expanded by addition of feature information regarding students and courses. Also, a regularization term is added to optimization problems in many cases, to avoid over-fitting learning parameters U and V.

Features representing selection of students and courses are essential parameters in recommender systems. Such information can be useful in developing an a priori estimate of student's inclination towards certain courses. Consider a student i represented by set of row feature vector x_i, such that $x_i \in \mathbb{R}^C$ and similarly a course j represented by row

feature vector y_j, where $y_j \in \mathbb{R}^D$. Here, C and D are number of features representing the entity.

Abernethy et. al. in [4] suggest that incorporation of features in the prediction algorithm casts the problem of finding a low-rank approximation into determining a low-rank function $f(i,j)$ in a tensor product of two reproducing kernel Hilbert spaces. This approach suggests working with a class of functions that operate on both user's performance(grades) and user's features. Thus we would like to estimate a function $f : \mathcal{A} \times \mathcal{B} \rightarrow \mathbb{R}$, given a finite set of entries in form of $(i,j,r_{ij}) \subset \mathcal{A} \times \mathcal{B} \times \mathbb{R}$. Since feature information for students and courses is available, similarity can be represented as two positive semi-definite kernel functions, $k : \mathcal{A} \times \mathcal{A} \rightarrow \mathbb{R}$ and $g : \mathcal{B} \times \mathcal{B} \rightarrow \mathbb{R}$ referring to similarity functions between students and courses respectively. Then the set of functions mapping student and courses to the preferences can be defined using the similarity kernels. If $\mathcal{K}$ and $\mathcal{G}$ be the reproducing kernel Hilbert spaces (RKHS) of kernels k and g, then the function we intend to approximate can be defined as closure of set of finite sums of product of functions in $\mathcal{K}$ and $\mathcal{G}$.

$$\mathcal{K} \otimes \mathcal{G} = \text{cl} \left\{ f(i,j) = \sum_{l=1}^{p} u_l(i) v_l(j) : u_l \in \mathcal{K}, v_l \in \mathcal{G} \right\} \quad (5)$$

Here each term $u(i)v(j)$ is called an atomic term. Given, a function $f \in \mathcal{K} \otimes \mathcal{G}$, if f can be written with k atomic terms $\sum_{l=1}^{k} u_l(i)v_l(j)$, and if k is minimal, then f is of rank k [2]. The representation of f in terms of atomic terms is equivalent to matrix factorization representation presented in equation (2). Thus, it becomes clear that $u(i)$ and $v(j)$ represent factors of users and courses respectively, which we learn using the available entries of incomplete matrix. Authors in [3] suggest that if $\mathcal{A}$ and $\mathcal{B}$ are finite spaces and if K and G are $n \times n$ and $m \times m$ kernel matrices of k and g respectively, then the the optimization problem of predicting preference by using features can be expressed as:

$$\min E_{(i,j,r_{ij})} \left\{ \sum_{(i,j \in \Omega)} (f(x_i, y_j) - r_{ij})^2 + \lambda \parallel f \parallel_{\otimes}^2 \right\} \quad (6)$$

An optimal solution for the optimization problem presented in equation (6) is given by application of representer theorem [14], which states that the solution lies in the expansion of training examples. Thus, our prediction matrix X, can be represented as $X = (K.U)(G.V)^T$, where U and V are $n \times k$ and $mtimesk$ factor matrices respectively, and K and G are similarity kernel matrices of user and item space respectively.

For convenience of representation, we replace U with α and V with β. Now, using the given partial entries (i,j,r) of the matrix in form of student performance, the gradient of loss can be computed as:

$$\nabla l(i,j,r_{ij}) = \nabla (r_{ij} - (K\alpha)_i (G\beta)_j^T)^2 \quad (7)$$

The update equations can be derived from equation (7). Equation (6) contains Frobenius norm regularization. In the next subsection, we describe the proposed regularization and its merits over the existing Frobenius norm regularization.

B. Proposed Regularization

Existing method to prevent over-fitting in learning the parameters α and β in equation (7) use Frobenius norm to restrict the spectral divergence of the prediction matrix.Matrix norm regularization has provided valuable support in many matrix completion algorithms [2], [4]. Since such regularization does not include any information pertaining to user-item relationship we believe that a better regularizer may help solve over-fitting problem faced by the matrix norm regularized approach. Also, the storage requirement for such a regularization is higher since it requires to store all the entries of a matrix for computing the regularization term.

An alternative regularization can be designed by considering the similarity between users and items. Similarities between users and between items can be considered as good additional information for predicting user preferences. In case of predicting student performance for similar courses, the grades would mostly remain consistent. For such scenario, student performance over different courses and similarity between different courses can act as good regularization. Such information can be used to penalize faulty predictions based on the degree of error. Suppose we intend to predict performance of student i in course j, then the regularization term can be mathematically expressed as:

$$\sum_{\forall j' \neq j \ (i,j') \in \Omega} \frac{v_j v_{j'}^T}{\|v_j\| \|v_{j'}\|} \times (r_{ij'} - X_{ij})^2 \quad (8)$$

Here r_{ij} refers to performance of student i in course j. The penalty term represented in equation (8) contains intrinsic information regarding the distribution of grades in different courses and the similarity between them. The similarity term in equation (8) is represented by $G(j,j')$ in the following equations. The proposed term penalizes the prediction based on its divergence from original distribution. The regularization parameter is set keeping in mind the assumption, that a student's performance may remain same in subjects of with higher similarity.

Since the proposed penalty term presented in equation (8) assumes similarity between courses and performance of students in them, it constitutes not only a point of interest to the student, but is also adequate to his/her skills and respects any kind of constraints (for example, courses prerequisites). Thus, for a given observation for student i and course j, if the original performance is denoted by r_{ij} and the predicted performance is denoted by $X_{i,j}$, the objective function can be mathematically expressed as:

$$\underset{i,j,r}{\text{minimize}} \quad (r_{ij} - X_{ij})^2 + \lambda \{ \sum_{b'=1}^{m} G(j,j').(r_{ij'} - X_{ij})^2 \} \quad (9)$$

Equation (9) presents the final objective function which contains the proposed regularization term. Here, the term X_{ij} is the predicted preference of student i for course j. Here it is important to note that, the prediction term X_{ij} contains two parts, namely, α term which comes from factorization

of student performance matrix and K kernel similarity term=. Thus, for cases where features don't convey much information, prediction solely depends on student performance matrix, while in cases where we don't have information regarding student performance, its features help in generating prediction. Now, to obtain the final update equations for parameter matrices α and β referred in (7) can be obtained from equation (9) by differentiating the cost function (l) with respect to α and β in the following manner:

$$\begin{aligned} \frac{dl}{d\alpha} &= 2(\beta.G)^T(r-\hat{r}) \\ \frac{dl}{d\beta} &= 2\alpha(\sum_{b'=1}^{m} G(b,b'))(r-\hat{r}) \end{aligned} \tag{10}$$

Above equations only change the update step of the algorithm mentioned in the previous chapter. Also, for each observation the whole matrix doesn't need to be updated as was the case with matrix norm regularization technique. In the next chapter, we discuss the performance of proposed approach and compare it with that of existing approach.

III. Experiment and Results

In this section, we present and compare results obtained using proposed regularization approach on two original datasets of student performance data from DA-IICT, Gandhinagar and IIIT-V. We refer to the dataset obtained form IIIT-V as Dataset-1 and the other as Dataset-2. Features representing courses in both the datasets were extracted in form of most frequent terms from their course descriptions by using a text-parser. Dataset-1 contains 300 students and 10 courses, while dataset-2 contains 84 students and 26 courses. Dataset-1 contains large number of courses and small set of students, hence insights from that dataset were used to study the performance of proposed method on large number of courses. On the other hand, Dataset-2 has higher number of students as compared to the number of courses, thus it provided insights regarding student profiling done by the proposed approach.

The configuration of the machine on which all the simulations were performed are as below:

- RAM : 4GB
- Operating System: Mac OSX
- Processor: Core i7 3^{rd} Generation

A. Effect of Regularization Coefficient

The objective function in the present problem involves a regularization term, for avoiding over-fitting of the training data. In this experiment, we study the effect of different regularization coefficients. The learning process gets affected by different regularization coefficients. Figure 2 presents the plot of Prediction error vs Number of Iterations, for different values of regularization coefficients (λ), for the proposed approach.

This exercise clearly states, that we'll obtain higher learning rate and lower error in case of regularization coefficient (λ) = 3 $\times 10^{-5}$. For all the following experiments, the value of regularization coefficient is chosen accordingly.

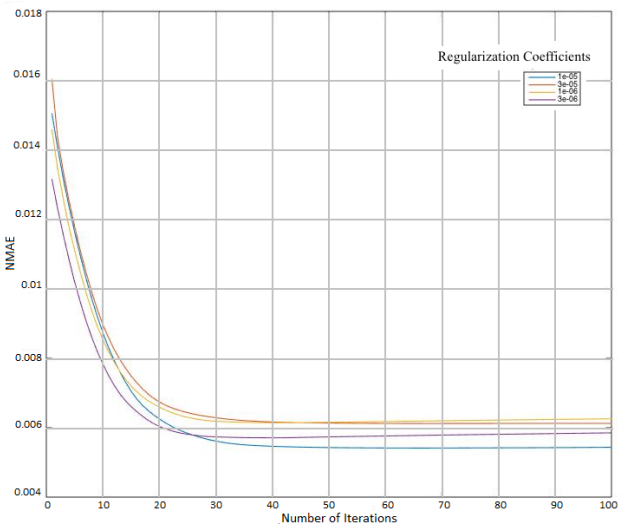

Fig. 2. NMAE vs. Number of iterations for various values of regularization coefficient

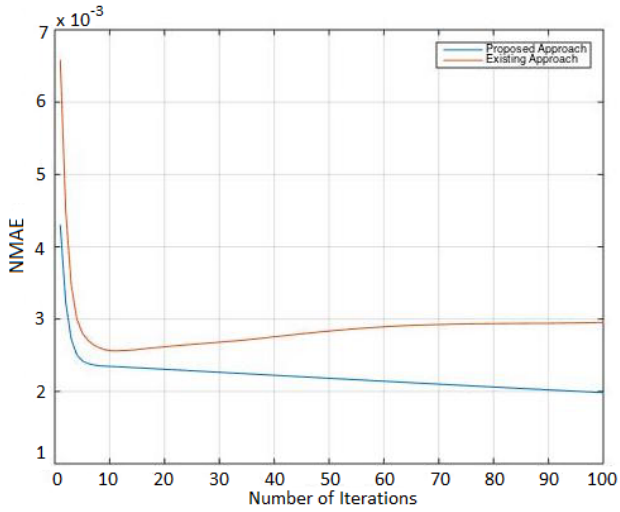

Fig. 3. Comparative performance analysis for dataset-1

B. Prediction

The performance of proposed algorithm was studied by plotting Error Performance versus Number of Iterations. This plot suggests the learning rate of the algorithm for a given dataset. At each iteration, prediction matrix was generated, using available observations and performance for test cases was computed. The regularization coefficient λ (equation (9)) was chosen to be 1×10^{-5}. Error performance plotted is average over 20 iterations. The test set and training set for experiment was chosen randomly. In most of the experiments described, the test case size is 10% of the total data.

From the above results it is clear that after a certain number of iterations, error converges to a minima, suggesting stability in the learning process. Such a performance is expected due to the regularization term used. The algorithm for learning preference matrix using existing approach generated instability

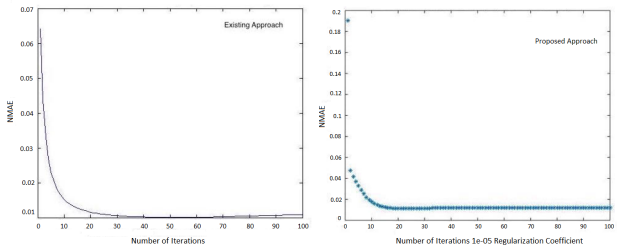

Fig. 4. Comparative performance analysis for dataset-2

in the learning procedure, which is overcome by the proposed approach. Also, in both the cases (figure 3 and 4), the observations for which maximum prediction error was achieved was analysed It was found that, when there are less number of observations available for a student/course, there is a major error in prediction for such user-course pair.

C. Optimal Rank Calculation

The objective function for the problem presented in equation (9) does not contain any provisions for selection of rank of prediction matrix X. In such cases where data representation via rank constraint plays a very vital role in prediction accuracy, we designed an experiment to determine optimal rank for each dataset. For this purpose, we plotted error in prediction for different rank values and selected the rank with least error as the optimal rank. Since, the data matrix being represented posses any rank between 1 and n $\times$ m (n, m are dimensions of the data matrix), we plotted the maximum error obtained for all the possible values of rank for a given data matrix.

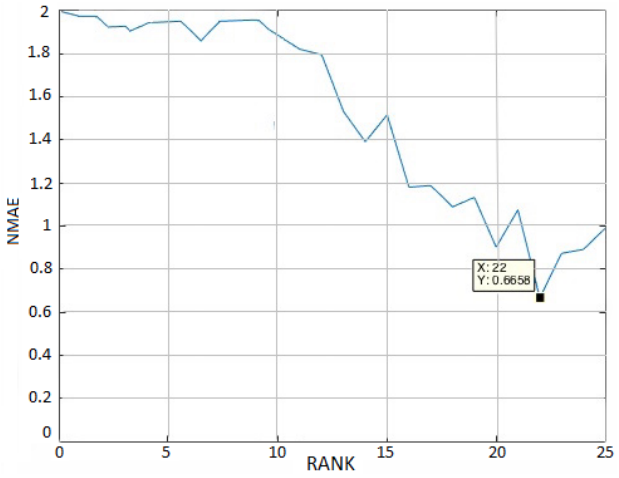

Fig. 5. Optimal Rank vs Error in Prediction

D. Training Sizes vs Prediction Error

To understand the behaviour of the proposed algorithm for different sets of training sizes, the prediction error for different set of training sizes was plotted.

It is observed that with the decrease in the number of observations in training dataset, the prediction error increases

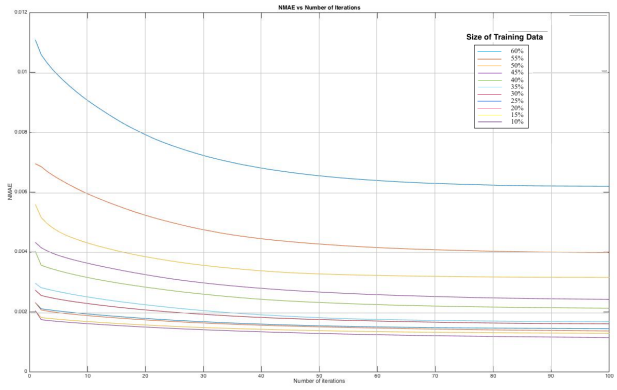

Fig. 6. NMAE vs Number of Iterations for Different sizes of Training Dataset

as the number of sufficient observations required to make prediction decreases.

E. Validation

A validation experiment to compare predicted student performance with the real-time obtained grades(CPI) by students for one semester is performed. Each semester contained 5 courses and the experiment involved predicting performance of semester- 2 based on semester-1s performance. Also the student performance of previous batches was also included in generating predictions. The predicted scores were generated beforehand and compared with actual scores obtained by students at the end of the semester. This study was performed on Dataset-1. The prediction is found to be very close to the original grades obtained in most of the cases.

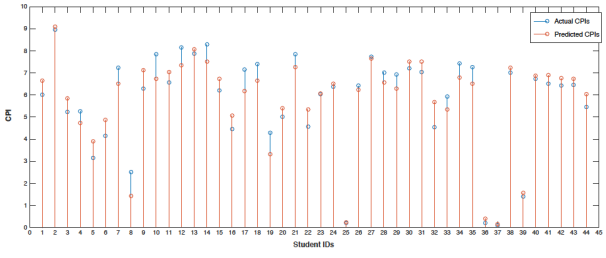

Fig. 7. Comparison of Predicted and Original CPIs of Students from Dataset-1

For some observations there has been observed a notable difference in the predicted and actual grades. This issue

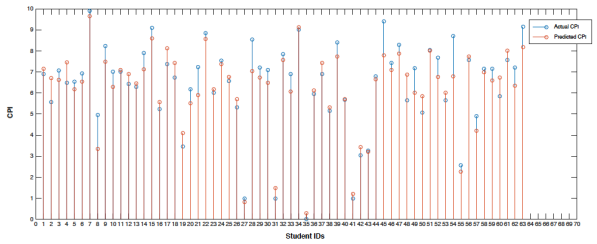

Fig. 8. Comparison of Predicted and Original CPIs of Students from Dataset-2

can be attributed to quantization of actual grades provided to students. This study also revealed an aspect where the system fails to predict accurately in cases where a students performance improves exceptionally during a semester.

Table I presents a comparative performance analysis of various methods for different datasets.

	Matrix-Norm Regularization [4]	Proposed Regularization
	NMAE Comparison	
Dataset-1	0.005	**0.0023**
Dataset-2	**0.0001**	0.0032
	Computational Time Comparison	
Dataset-1	95.69s	**39.30s**
Dataset-2	78.49s	**49.21s**

TABLE I
PERFORMANCE COMPARISON TABLE

The computation time for the methods have a major difference as shown in the above table. Also the absolute error in prediction by the proposed method is low as compared to existing method.

IV. CONCLUSION AND FUTURE WORK

In this work, we have proposed a machine learning approach for Course Recommender Systems by extending the online matrix factorization method. The results of experiments carried out suggest that the proposed regularization in the learning algorithm improves the prediction accuracy and computation times. The comparison based on prediction error, learning rate and computational times of the proposed method with the existing matrix-norm based regularization method indicates significant improvement for proposed regularization. The current experiments were carried out considering only three features of the students, i.e CPI Grade, IIT-JEE Marks and the Higher Secondary education board of the students, and for the courses, 30 features representing the courses were used. We believe that, there are many crucial factors which govern the decision making process of a student, as depicted in figure 1. As an extension of this work, we would like to include other features also in the recommender system.

REFERENCES

[1] Jacob Abernethy, Francis Bach, Theodoros Evgeniou, and Jean-Philippe Vert. A new approach to collaborative filtering: Operator estimation with spectral regularization. *Journal of Machine Learning Research*, 10, 2009.

[2] Jacob Abernethy, Francis R. Bach, Theodoros Evgeniou, and Jean-Philippe Vert. Low-rank matrix factorization with attributes. *CoRR*, abs/cs/0611124, 2006.

[3] Jacob Abernethy, Francis R. Bach, Theodoros Evgeniou, and Jean Philippe Vert. A new approach to collaborative filtering: Operator estimation with spectral regularization. *Journal of Machine Learning Research*, 10:803–826, 2009.

[4] Jacob Abernethy, Kevin Canini, John Langford, and Alex Simma. Online collaborative filtering. *University of California at Berkeley, Tech. Rep*, 2007.

[5] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749, 2005.

[6] Emmanuel J. Candes and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[7] Fbio Oliveira Garrido Carballo. *Masters Courses Recommendation: Exploring Collaborative Filtering and Singular Value Decomposition with Student Profiling*. 2014.

[8] Michael D. Ekstrand, John T. Riedl, and Joseph A. Konstan. Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction*, 4:81 173, 2010.

[9] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[10] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems: An Introduction*. Cambridge University Press, New York, NY, USA, 1st edition, 2010.

[11] Yehuda Koren, Robert M. Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[12] Monique Laurent. *Encyclopedia of Optimization*, chapter Matrix completion problemsMatrix Completion Problems, pages 1311–1319. Springer US, Boston, MA, 2001.

[13] Youngseok Lee and Jungwon Cho. An intelligent course recommendation system. *Smart CR*, 1(1):69–84, 2011.

[14] Jonathan H. Manton. A primer on reproducing kernel hilbert spaces. *Foundations and Trends in Signal Processing*.

[15] Trung V. Nguyen, Alexandros Karatzoglou, and Linas Baltrunas. Gaussian process factorization machines for context-aware recommendations. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, pages 63–72, 2014.

[16] Cristobal Romero and Sebastian Ventura. Educational data mining: A survey from 1995 to 2005. *Expert systems with applications*, 33(1):135–146, 2007.

[17] Yue Shi, Martha Larson, and Alan Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Comput. Surv.*, 47(1):3:1–3:45, 2014.

[18] Roberto Torres, Sean M. McNee, Mara Abel, Joseph A. Konstan, and John Riedl. Enhancing digital libraries with techlens+. In *ACM/IEEE Joint Conference on Digital Libraries, JCDL 2004, Tucson, AZ, USA, June 7-11, 2004, Proceedings*, pages 228–236, 2004.

[19] Kai Yu, Anton Schwaighofer, Volker Tresp, Xiaowei Xu, and Hans-Peter Kriegel. Probabilistic memory-based collaborative filtering. *IEEE Trans. Knowl. Data Eng.*, 16(1):56–69, 2004.