Predicting Course Grades

Martin Hunt, Sharon Lin, Chinmay Kulkarni
{chinmay, mghunt, sharonl}@stanford.edu

Abstract

Recommendation systems have been used in a variety of domains, ranging from online-joke recommendations to automatic educational course recommendations (CourseRank). However, users may want different recommendations for different reasons. In order to help each user select the recommendation that aligns best with his or her goals and values, a recommendation system can present users with estimations of various attributes of a particular recommendation. Using the data from the CourseRank recommendation system, we attempt to construct one potential value-predictor: the estimated grade a student will receive for a given course. Both SVMs and collaborative filtering techniques performed well, but neither could provide significantly better grade predictions than a baseline estimate of the average grade for each student.

1 Introduction

Recommendation systems attack the general problem of recommending items to a user that are likely to be of interest. This typically requires estimating *ratings* for a set of items with unknown ratings based on another set of items with known ratings. The items with the highest expected ratings are recommended to the user. Recommendation systems have been applied to various types of items including physical goods, movies, research papers, webpages, educational courses, etc. [6]. Similarly, the ratings could either be explicitly provided by the user (e.g. using a 1-to-5 star rating, or a Likert scale) or obtained implicitly by the system (e.g. the number of times a webpage was visited).

Recommendation systems are primarily either model-based or memory-based. A model-based system abstracts a probabilistic model that predicts ratings for a given user and item. A memory-based system uses unsupervised learning to cluster either user or item data based on known ratings. An unknown rating is then approximated with a weighted average of known ratings. The weights are determined by computing the distance between unknown and known items with a reasonable distance metric. Memory-based systems have been particularly popular, since they make fewer assumptions regarding what factors contribute to a rating and are more likely to produce diverse recommendations than a model-based approach.

Users may have specific needs for recommendations that are not fully addressed by an overall rating. In these cases, the user has to evaluate the given set of recommendations in order to find the most useful ones. In this paper, we explore one particular instance where users can be assisted in this task. One particular aspect that we focus on is value prediction for more specific item-attributes. This is useful for answering questions such as, "how *suspenseful* is this recommended movie?", or "how *important* is this research paper?".

In our system, the exemplar question we try to answer is "what *grade* will I make?" for a University course recommendation. Course grades are particularly interesting because they vary by student and when the course is taken. We use both SVM and collaborative-filtering-based approaches to predict student grades based on student transcript and schedule data.

2 Predicting Grades

Our data is drawn from the CourseRank website, which is a course-recommendation website run at Stanford. The CourseRank database contains information on 10,000 anonymized student transcripts and around 7,000 courses.

2.1 SVM

We trained an SVM to detect A versus non-A grades, where doing well in a course corresponds to receiving an A. We also trained a multi-class SVM to predict grades across the whole letter grade spectrum from A+ to F. One SVM was trained per course.

All SVMs were trained using a Gaussian kernel. The ξ and γ parameters for the kernel were chosen via a rough grid-based search on

$$\xi = [12864328421] \quad \text{and} \quad \gamma = \left[\frac{1}{\text{numFeatures}} \frac{2}{\text{numFeatures}} \frac{4}{\text{numFeatures}} \frac{8}{\text{numFeatures}}\right]$$

to maximize the 10-fold cross validation accuracy. The multi-class SVM was implemented using multiple SVMs that compared grades higher than a certain threshold against the grades lower than the threshold. That is, there is one SVM that predicts each of A+ vs other grades, A or above vs. below A, A- or above vs. below A-, and so on for each unique grade appearing in the course. The final predicted grade is then decided by a majority vote among all the individual SVMs.

Given the CourseRank transcript data, we chose six features that we hypothesized may affect course grades– the students' previous course grades, recent GPA by department (last three quarters), major, concurrent courses, planned weekly workload, and the number of courses previously taken.

These features provide some information on the students' experience with related courses and subjects at Stanford, whether the course is a requirement or an elective for their major, and how much time they have to spend on the course. The previous course grades, recent GPA by department, student major, and concurrent courses were represented by vectors where each index corresponded to one course, department, or major.

2.2 Collaborative Filtering

We consider another model for estimating grades based on the same methods used in the original recommendation system. Here, we make the hypothesis that similar students will make similar grades. Thus, for a student s with course history C_s and course $c \notin C_s$, we find students $s_i \in S$ with $|C_{s_i} \cup C_s| > 0$ and $c \in C_{s_i}$ (students who have taken some of the same courses as s as well as course c). Then u's grade in c can be estimated as a weighted average of the grades g_{s_i}.

To determine the relative weights of users, we used cosine similarity and Pearson's correlation as the primary two measures. Other measures of similarity such as Kendall, Spearman, and adjusted cosine similarity have been explored in [3] and [9], but they tend to perform similarly or worse. We expect our choice of similarity will have little impact on the predicted grades.

For two student (sparse) vectors S_a and S_b with each element corresponding to the student's grade in a course, we compute the similarity using:

$$\text{similarity} = \frac{S_a \cdot S_b}{\|S_a\| \|S_b\|} \qquad \text{Pearson's} = \frac{\text{cov}(S_a, S_b)}{\sigma_{S_a} \sigma_{S_b}} = \frac{\sum_{i=1}^{n} (S_a^{(i)} - \overline{S}_a)(S_b^{(i)} - \overline{S}_b)}{\sqrt{\sum_{i=1}^{n} (S_a^{(i)} - \overline{S}_a)^2} \sqrt{\sum_{i=1}^{n} (S_b^{(i)} - \overline{S}_b)^2}}.$$

Grade predictions for CF follow the multiclass SVM model with 1: A+, 2: A, 3: A-, ..., 13: F. Other implementation details follow the formulas and algorithms outlined by [2] and [4].

3 Results

3.1 SVM Results

We trained A vs. non-A and multi-class SVMs for each of 17 courses that had at least 100 student records. We compared the 10-fold cross-validation accuracy against the average course grade and

student's GPA baselines. For the student's GPA baseline, if a student has not taken a course before, his or her grade is predicted to be the average course grade. The results are shown in the Tables 1 and 2 below.

3.1.1 Prediction Accuracy Compared to Baselines

On average, both types of SVMs outperformed the mean course grade baseline and are comparable to the student's GPA baseline. Overall, the multi-class SVM correctly predicts a slightly greater percentage of students within a half letter grade when compared to the baselines.

Predictor	Average Accuracy
A vs. non-A SVM	64.2%
Student's GPA	66.4%
Mean course grade	60.6%

Table 1: Average accuracy for the A vs. non-A SVMs compared to the baselines (higher is better)

Predictor	Average MAE	Within a half letter grade
Multi-class SVM	1.38	65.2%
Student's GPA	1.37	61.1%
Mean course grade	1.45	57.0%

Table 2: Comparing the multi-class SVMs compared to the baselines. An error of 1 corresponds to a half letter grade error, while an error of 2 corresponds to a full letter grade error.

3.1.2 Most Important Features

We evaluated the contribution of each set of features by training the A vs. non-A SVM exclusively on each set and comparing the 10-fold cross-validation accuracies against the baseline statistical accuracy. Table 3 shows the results.

Overall, the top four features in order were previous course history, recent grades by department, student's major, and concurrent courses. Weekly workload did not contribute to the accuracy of any of the selected courses, and the number of taken courses only influenced accuracy in CS161 out of the 17 courses tested.

However, across different courses, the features that most influenced prediction accuracy varied. For example, in CS161, all of the features except for weekly workload contributed to the prediction rate, and the student's previous course grades were the most significant predictor. On the other hand, in ARTSTUDI60, the student's major was the most significant predictor. In CHEM31A, none of the features contributed to the accuracy, suggesting that different factors may better characterize course grades in this case.

3.2 Results of Collaborative Filtering

For CF, we filtered data to remove all courses with fewer than 3 grades and all students with fewer than 4 courses, resulting in 5,930 students and 3,603 courses. We tested against the entire dataset and specific classes. For the entire dataset, 10% of the students were randomly separated with 3 courses per student removed. Each student was compared with the remaining 90% (5337) to compute the top-n most similar students. The results are shown in Table 4.

The primary reason the estimates from CF are not much better than the estimate based on students' GPAs is that the dataset is too sparse to find similar users. Upon analysis of similar users, we find that the similarity coefficients are very small ($O(10^{-}3)$). We further validate this by noting that analysis shows the error of the student GPA estimates and the CF estimates are highly correlated. Because the similarity coefficients determine the deviation of the estimate from the average, the CF estimates generated with small similarity coefficients are very similar to the student GPA estimates.

Recognizing that this user-based CF approach requires more data to be effective, we experimented with tuning various CF parameters in a non-rigorous manner with minor reductions in the MAE

Course	Dept. GPA	Concurr. courses	Course grades	Major	Workload	Num. Prev. Courses
ARTSTUDI60	0	1.83	1.83	5.50	0	0
CHEM31A	0	0	0	0	0	0
CHEM33	3.11	0.35	1.29	0.09	0	0
CS105	0.17	0.50	2.67	0.50	0	0
CS106A	0.63	0.00	1.40	3.29	0	0
CS161	10.21	8.80	16.90	5.99	0	11.27
CS229	0	1.96	0	0	0	0
EE108B	0	1.86	0	1.24	0	0
HUMBIO2A	0.31	0	6.16	0	0	0
HUMBIO2B	5.24	0	7.26	0	0	0
IHUM2	7.99	1.91	6.25	5.90	0	0
IHUM57	0	1.01	0	0	0	0
IHUM63	0	0.66	0	0	0	0
MATH42	4.40	0	0.51	0	0	0
PHYSICS43	4.93	1.15	5.92	1.15	0	0
PSYCH1	1.67	1.12	1.12	1.49	0	0
POLISCI1	0.34	1.02	3.07	2.39	0	0
Average	2.29	1.30	3.20	1.62	0.00	0.66

Table 3: Percent accuracy increases for A vs. non-A SVM when including only one of the feature types. The largest contributor for each course is highlighted.

($O(0.1)$). We tried various values of n from all students to 5 at logarithmic intervals and found that smaller n tended to reduce the MAE. We also looked at the similarity coefficients: as defined in [2], the similarities are normalized before computing the weighted average. By increasing the normalization coefficients, we further reduced the MAE. We hypothesize that increasing the weights favors outliers and skews the estimate away from the average. We also tried removing the influence of negatively correlated students as both Herlocker and Yao indicate that this improves error. Though all these adjustments reduced the MAE, none significantly improved the estimates.

For comparison with support vector machines, we ran tests on three random courses to compute the MAE for just those courses. Here, for the chosen course c, we consider all students S with $c \in C_{S_i} \forall i$. We compare each student S_i with students $S_j, j \neq i$ by computing the similarity $\text{sim}(C_{S_J}, C_{S_I} \setminus c)$ and then using the weighted average as before to estimate S_i's grade in c. We used courses with over 50 students to run these tests. The MAE for these tests did not differ significantly from the average student estimates.

	cosine sim	pearson's corr	student's gpa	mean course grade
MAE	1.3896	1.3704	1.4231	3.2140
L_2 error	3.9439	3.7505	4.0451	14.4421

Table 4: MAE and L_2 error for estimates using collaborative filtering averaged over 25 test runs. Each test estimated approximately 1,500 grades using 10,453 known course grades to compute the similarities using a base corpus of 5337 students.

4 Conclusion and Future Work

Past course history can only go so far in predicting future course grades. A significant percentage of students in many of the courses had not taken any courses previously, which may partially explain the low accuracy of the baselines and the SVMs. In addition, there may be more promising features that characterize the current quarter if course schedule data were available, such as potential schedule and deadline conflicts between concurrent courses and student activities. Although the concurrent courses features was intended to help capture some of the effects of schedule conflict, it is flawed in that most courses do not have a fixed schedule and instead change term by term. Thus courses that have conflicted in the past may not conflict in the future.

It is also possible that the unexpected occurrences throughout the quarter may have greater influence on grades. For example, a student may just not have studied enough for a final exam, which heavily influences his or her grade. Individual personality factors may also play a part, where doing poorly

on a similar course in the past may drive some students to work harder on a similar course in the future.

Improving the collaborative filtering results would most likely require additional data from the CourseRank system. We would hope to find more similarities between students on which to base our estimations. This may not be possible: though more grades will be added to the system each year, the relevancy of the older grades will diminish.

An alternative approach to address data sparsity might be to broaden our initial hypothesis. We could broaden our hypothesis that similar students will perform similarly in courses. Instead, given a student s and course c, we could try to find students that have not only taken the same courses as s an also taken c, but who have taken similar courses to both c and courses similar to those taken by s. Jun Wang, et. all have demonstrated a method similar to this that combines user-based and item-based collaborative filtering to overcome sparseness [8].

Finally, it is possible that we see poor performance because the underlying assumption that similar students in courses from some set C_a will receive similar grades in some other set C_b may be false. Factors such as concurrent course workload, interest in course, different professors, and others may instead be responsible for deviations in expected course grades. Some of these external factors may be irrelevant as our analysis of SVMs indicates, but others may require collection of additional data before we can significantly improve course grade estimates.

References

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, pages 734–749, 2005.

[2] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. pages 43–52. Morgan Kaufmann, 1998.

[3] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Research and Development in Information Retrieval*. American Association of Computing Machinery, American Association of Computing Machinery, 8/1999 1999.

[4] G. Lebanon. C/matlab toolkit for collaborative filtering. `http://nyc.lti.cs.cmu.edu/IRLab/11-743s03/lebanon/IR-lab.htm`, Aug. 2003. Language Technologies Institute, Carnegie Mellon University.

[5] J. O'Donovan and B. Smyth. Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces*, pages 167–174. ACM, 2005.

[6] J. Schafer, J. Konstan, and J. Riedi. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166. ACM, 1999.

[7] A. Tversky and I. Gati. Similarity, separability, and the triangle inequality. *Psychological Review*, 89(2):123–154, 1982.

[8] J. Wang, A. P. D. Vries, and M. J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 501–508. ACM, 2006.

[9] Y. Yu, Z. Shanfeng, and C. Xinmeng. Collaborative filtering algorithms based on kendall correlation in recommender systems. *Wuhan University Journal of Natural Sciences*, 11:1086–1090, 2006. 10.1007/BF02829215.