A Graduate School Recommendation System Using the Multi-Class Support Vector Machine and KNN Approaches

Alisha Baskota
Computer Science Department
Brigham Young University
Provo, Utah 84602, USA
baskotaalisha@gmail.com

Yiu-Kai Ng
Computer Science Department
Brigham Young University
Provo, Utah 84602, USA
ng@compsci.byu.edu

***Abstract*—With the advancement in technology and increased demand on skilled workers these days, education becomes a stepping stone in securing jobs with long-term perspective. As competition for admission into higher education increases, it becomes even more important for applicants to find graduate schools that fit their requirements and expectation. Selecting appropriate schools to apply, however, is a time-consuming process, especially when looking for schools at graduate level due to the various factors in decision making imposed by the schools and applicants. In this paper, we propose a recommendation system that suggests appealing graduate programs to students based on the Support Vector Machine and K-Nearest Neighbor approaches. As graduate programs make decisions based on applicants' qualification, our recommender considers user's personal data and data of various graduate programs obtained from online education portals to make suggestions. We conduct an empirical study using data of current graduate schools and former graduate school applicants, and the performance evaluation validates the merit of our suggestions.**

***Keywords*-graduate school; recommender; machine learning**

I. Introduction

Webometrics[1], in its January 2017 edition, lists 26,368 universities from all over the world and among them 3,281 are based in the USA. According to this data, there is a myriad of options to choose for pursuing graduate studies even though not all universities are equally appropriate for every applicant. For example, an applicant may be looking for top universities that offer a well-established PhD program in Computer Science; however, these universities may not be appealing to the applicant due to unaffordable tuition and fees. While another applicant may not be sure which graduate degree program to apply, the applicant is fully aware that (s)he is not willing to pay beyond $20,000 each year for tuition. Meanwhile, there may be another applicant who is particularly looking at universities that offer a public health Master's degree with required GRE score of 320, CGPA of 3.6, and an annual tuition of around $12,000. Hence, although the search criteria for these applicants might be different, they are expected to spend significant amount of time and efforts on the Internet looking for appropriate universities. For this reason, a graduate school recommendation system becomes useful in terms of speeding up the search process and finding appropriate graduate schools to apply. It would be efficient and cost effective if an applicant could obtain the information about graduate programs of interest through a single platform without having to go through the hassle of searching for schools through multiple educational websites that archive information about universities or visiting the websites of the universities themselves.

In this paper, we propose a recommendation system that eliminates the tedious application search process imposed on graduate school applicants by designing a single platform which can shortlist the universities/colleges appealing to the applicants. Existing graduate school recommendation systems fail to achieve this goal, since most of them either rely solely on applicant data or the data of students who have already enrolled in graduate schools. Some even have uncommon features like caste and merit number which are not applicable for majority of the graduate schools throughout the world. The proposed recommendation system, on the other hand, suggests different alternatives to applicants using the data about graduate schools. Also, we use different features of graduate schools that can be extracted online to yield a more robust and scalable recommendation approach. Our source of data comes from educational portals, such as Edulix.com and Yocket.com, which archive data of universities and these websites are built to assist both graduate and undergraduate students to search for information about graduate schools. The proposed system analyzes data from these websites, selects appropriate features from the data, runs the Support Vector Machine (SVM) classification algorithm [9] and K-Nearest Neighbors (KNN) [12] machine learning algorithm on them, and suggests appropriate universities to the applicants accordingly.

SVMs have been applied with success to information retrieval problems, particular text classification. SVM is a promising method for the classification of both linear and nonlinear data, and graduate school data include nonlinear

[1] http://webometrics.info/en

type of data. Even though the training time of SVMs can be slow, they are highly accurate, owing to their ability to model complex linear and nonlinear decision boundaries. In addition, they are much less prone to overfitting than other classification methods [13]. SVM is chosen as our classifier in finding the most promising, called the *core*, university/college, which is the most appealing graduate program to an applicant suggested by our recommendation system based on the fitness of the qualification and requirements between the applicant and the university/college.

KNN, on the other hand, is adopted for finding graduate schools that are similar to a *core* university/college, since it is computationally effective and has proven to provide better results in comparison to other machine learning approaches [14]. When given a data item, which is a *core* university in our case, the KNN algorithm searches the pattern space, which is an n-dimensional vector space, for the k-nearest neighbors, which are other universities/colleges in our case, that are closest to the given data item [4]. The accuracy of recommendations achieved by SVM and KNN is relatively high as presented in Section IV.

The rest of the paper is organized as follows. In Section II, we present existing approaches to graduate program recommendation. In Section III, we introduce our graduate school recommendation system. In Section IV, we evaluate the performance of our recommendation approach. In Section V, we give a concluding remark and discuss future work for our recommendation system.

II. Related Work

There are only a few college/university recommendation systems that have been proposed and developed in the past. Dikhale et al. [5] introduce a college recommendation system in which they compare Naïve Bayes and Weka's J48 implementation of the C4.5 algorithm to generate recommendations. They use 160 entries about students which include students' demographical information like gender, university, caste, merit number, previous college, and stream as their six features to create classification results for five colleges based in India. In contrast to Dikhale et al.'s approach, we generate recommendations for graduate schools in the USA and the number of schools to be considered are much higher than 160. Also, our features are widely different from theirs because our online dataset consists of information about graduate schools in addition to students. Besides using a larger dataset, we take a different approach, i.e., SVM and KNN, in making recommendations to their work.

Hasan et al. [7] examine the idea of using K-Nearest Neighbors to generate relevant graduate school recommendation. They create a weighting score for their training set and another one for test set to calculate the similarity between the two scores, and return top K similar results. However, they do not articulate how the weighted scores are computed and how the similarity of those scores is calculated. For testing their system, they recommend top-K similar users to a user without specifying how the value of K is determined and which value of K they finally use. In contrast, we conduct an experiment on varying values of K to determine the most ideal value, i.e., the one that gives the highest accuracy, and use the value further into the process of generating recommendations. We have also considered different distance functions, i.e., Euclidean and Manhattan distances, to calculate the nearest neighbors. The idea of using the data of similar users who have already enrolled in some graduate programs for testing the system is a feasible design technique as we also have the data available.

Bokde et al. [1] introduce dimensionality reduction techniques in university recommendation. They develop a university recommendation system using the multi-collaborative filtering approach, address issues like scalability and sparsity, and combine dimensionality reduction techniques with collaborative filtering (CF) algorithms. They use Principle Component Analysis (PCA) mean and Higher Order Singular Value Decomposition (HOSVD) for mapping a higher dimensional input space into lower dimensional latent space. They also adopt PCA for identifying reducing factors in matrix factorization (MF) and HOSVD for tensor factorization. Their reduction techniques, however, are mainly in conjunction with collaborative filtering techniques and involve user preference. The user preference is a constraint, since they are not widely available, and is thus a restriction in making recommendation. Instead, we adopt feature selection algorithms rather than the reduction techniques.

III. Our Recommendation System

Our graduate school recommendation system suggests universities/colleges to applicants that are appealing to them. A set of features is chosen based on the online information we gather about graduate schools that were used by applicants in the past for university selection. The recommender first trains a multi-class Support Vector Machine (SVM) [6] classifier to find the most appealing graduate school for an applicant using an online dataset and then apply the KNN algorithm to suggest other relevant universities to the applicant according to different credentials of the applicant and graduate schools. Hence, our recommendation system adapts a hybrid approach, i.e., the multi-class SVM and KNN algorithms. This hybrid approach is unique among existing graduate school recommendation systems.

The factual data of schools includes the acceptance rate of graduate schools, CGPA, GRE scores, location, number of students admitted to the schools, private/public universities, ranking of the schools, safety (i.e., how likely the universities would accept a student based on his/her credential), and tuition. Our recommender gathers acceptance rate, location, private/public information, ranking, and tuition from Yocket.com, and CGPA, GRE requirements, total number of admissions, and safety from Edulix.com. Initially, we collect

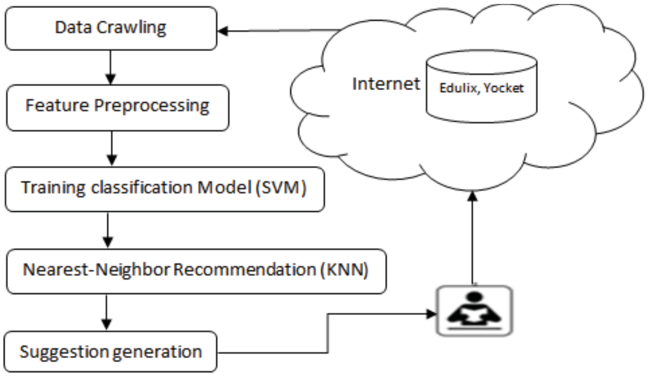


Figure 1. The architecture of our recommendation system

every feature that can be extracted from these websites. Hereafter, these features are cleaned and filtered which are then used by the trained SVM and KNN algorithms to suggest relevant universities for students. The SVM and KNN algorithms generate the most appealing graduate schools for each user, which are presented as graduate school recommendations to the user. (See Figure 1 for the architecture of our recommendation system.)

A. *Support Vector Machine (SVM)*

Categorizing graduate programs is in and of itself a challenging problem, since there are many factors that affect the admission decision made by a graduate program. As we attempt to avoid for both false positives and false negatives in the classification process, we use a multi-class support vector machine (SVM), since there is some control to help minimize the errors encountered using SVM. A multi-class SVM is not only chosen because of this control, but also because of its well-integrated support for kernels.

Traditional SVMs deal with data element classification in a binary way. To categorize with more than two classes, i.e., universities in our case, a multi-class SVM can be used. A multi-class SVM can be defined using a one-against-one approach, which is adopted by our graduate school recommendation system for classifying multiple graduate schools. The *one-against-one* approach is implemented by chaining together multiple SVMs, one per class for each class to be included in the categorization. Each SVM determines if the data point is or is not within the corresponding class.

Defining the data attributes is also very important for the success of a SVM. The data attributes determine how each element is mapped to the vector space. A data attribute can be anything from a measurement, such as the length of an element, to a categorical representation, such as color. For useful mapping, the attributes should be represented in a numeric way,[2] which means categorical attributes must be converted to a numeric system, and be meaningful to the categorization process. As for being meaningful, suppose one would like to design a SVM to categorize animals into either the cats or dogs class. An attribute such as animal weight would be meaningful while an attribute such as whether the animal has fur would not. The later is not useful, since most cats and dogs have fur, and thus fur would not help to distinguish between a cat and a dog. However, since dogs generally weigh more than cats, the animal's weight would be useful for classification. To deal with graduate program classification, we have chosen the LibSVM implementation [3]. It is a multi-class SVM that uses the one-against-one approach for classification.

B. *K-Nearest Neighbors (KNN)*

In processing a student's request for graduate program recommendations, after our recommender invokes the trained multi-class SVM to determine the core university to be recommended, it is followed by using the KNN algorithm to suggest similar appealing graduate programs offered by other universities that are closely related to the core's program to enhance the choices of appropriate programs for the student to consider. We have chosen KNN as the algorithm for our recommender, since it is a machine learning approach that has been widely used in statistical estimation and pattern recognition. In addition, it outperforms other machine learning approaches in predicting universities to be recommended based on our empirical study.

Our recommender relies on the regression analysis adopted by KNN, which is based on the idea of finding out the values for the unknown of the *core*'s neighbors. For regression analysis, KNN computes the value of each of the neighbors of the core that should be assigned, i.e., graduate programs to be recommended. The neighbors are calculated by finding the *distance* between the core and another instance, and the number of neighbors are decided empirically. The best number of neighbors are those which give the maximum accuracy for the recommendation task. Various distance functions, such as Euclidean, Manhattan, and Mahalanobis distance, can be used. We have tried Euclidean and Manhattan distance as they are commonly used and are considered appropriate if there are real-valued data in the dataset. *Euclidean distance* is defined as

$$E(V,W) = \sqrt{(a_1 - b_1)^2 + \ldots + (a_n - b_n)^2} \qquad (1)$$

where $V =< a_1, \ldots, a_n >, W =< b_1, \ldots, b_n >$, and V and W are the vector representations of the attribute values of two different instances.

The *Manhattan distance* is defined as

$$M(V,W) = |a_1 - b_1| + \ldots + |a_n - b_n| \qquad (2)$$

where V and W are as defined in Equation 1.

[2] Each attribute must be numerical, since an input element is represented by a vector made from the data attributes.

Table I
THE SET OF FEATURES ADOPTED BY OUR RECOMMENDER THAT IS CHOSEN BY THE *Forward_Feature_Selection* ALGORITHM USING THE INITIAL SET OF FEATURES SHOWN IN TABLE II

Acceptance Rate	CGPA
GRE AWA	GRE Quantitative
GRE Verbal	Number of Admissions
Ranking	Tuition

Table II
A SET OF FEATURES CONSIDERED BY OUR RECOMMENDATION SYSTEM

Acceptance Rate	CGPA
GRE AWA	GRE Quantitative
GRE Total Score	GRE Verbal
Location	Number of Admissions
Private/Public	Ranking
Safety (Probability)	Tuition

The two distance functions yield similar results. For this reason, we simply choose *Euclidean distance*. Using the Euclidean distance, we can find how far apart an unknown set of values for all the attributes is from its neighbors. Given below is the pseudocode of KNN that we use for university/college recommendation. Note that in the KNN algorithm a data instance consists of the features, i.e., data items, as shown in Table I, of a university/college (represented as *label*), whereas an *core instance* is a core university in our recommendation system.

```
Function KNN $(X, x)$
Input. $X$: Data instances,
       $x$: A core instance whose neighbors to be found
Output. Labels of the list of K-Nearest Neighbors of $x$
1. Calculate the distance $d(x, p_i)$ between $x$ and each
     instance $p_i$ $(i = 1, \ldots, n)$ in $X$ without using
     the corresponding label
2. Sort the distances $d(x, p_i), i = 1, \ldots, n,$ in increasing
     order to generate the sorted list, $Sorted_x$
3. Extract the labels of the first $K$ instances in $Sorted_x$
     from $X$
4. Return all the extracted labels in Step 3
```

C. Data

Our recommendation system considers the user's own data as well as the requisite data of different universities/colleges before making suggestions. *Two* datasets were created for our recommender.

The first dataset on various universities/colleges, called *Grad_Sch*, is generated by combining data from Edulix.com and Yocket.com, and is used by the KNN algorithm. The data attributes for each instance in Grad_Sch are shown in Table I, in addition to the *names* of universities/colleges. As mentioned earlier, some of the data items in Grad_Sch used are not available in a single platform. Hence, we extract the data from both these sites and consolidate them. Altogether, there are slightly more than 3,000 instances in Grad_Sch with around 500 distinct (university/college) names.

The second dataset, called *Std_Info*, consists of former graduate school applicants extracted from Edulix.com. The features, i.e., data attributes, in each instance of Std_Info, i.e., a graduate school applicant, include the *GRE verbal score, GRE quantitative score, GRE AWA (Analytical Writing Measure), CGPA* of the applicant, and status (pubic/private), *tuition, ranking, safety, location, name*, and *number of admission* of the graduate school, and *scores* required by the school, which consists of GRE verbal score, GRE quantitative score, and GRE AWA. We partitioned Std_Info into two subsections, with 14,000 instances used for training the multi-class SVM model, and another 2,000 instances in another subset for test purpose. Note that all the students in Std_Info have already enrolled in graduate schools, along with other universities/colleges that have admitted them into their respective graduate programs. These data serve as the *gold standard* for our empirical study.

1) Data Processing: The data extracted from Yocket.com have uneven naming convention for graduate schools. Some are abbreviated (SUNY), some have location information attached to the name (Clemson University South Carolina) and some are only full-length names of the graduate schools (Iowa State University). Edulix.com, on the other hand, includes full-length university names. We replace the abbreviations with full names, remove locations from the Yocket.com and convert them into same format as Edulix's naming convention. When removing locations, we only get rid of the redundant location information from the names. For example, Brigham Young University-Provo and Brigham Young University-Idaho must contain location names as they are two different universities, but Clemson University South Carolina is redundant, since there is only one Clemson University which is located in South Carolina.

2) Data Scaling: The datasets includes data items lying in different ranges, since the data are collected over a long period of time and from different sources. For example, older GRE scores were ranged within 1600, whereas new ones are within 340. We convert all GRE scores on a scale of 340 by referring to ETS,[3] the official site of GRE. Also, some applicants have CGPA on a scale of 10 while others on a scale of 4.0. We convert all the grades on a scale of 10 by using Msinus.[4]

D. Feature Selection

Using data extracted from Edulix.com and Yocket.com for the KNN algorithm, we determine the criteria specified by applicants and graduate schools that are *useful* in selecting appropriate graduate schools for the applicants. Given all the features, our recommendation system applies the *forward*

[3] https://www.ets.org/s/gre/pdf/concordance_information.pdf
[4] http://www.msinus.com/content/convert-cgpa-into-gpa-389/

selection algorithm under the wrapper method of feature selection [8] to find out the subset of features that yields the maximum accuracy. *Feature selection* is performed, since we cannot simply base on our personal opinions to determine how important each feature is. Applying the *forward feature selection* algorithm (given below) on the set of features given in Table II, the subset of features that yields the highest accuracy of university/college prediction is shown in Table I.

```
Function Forward_Feature_Selection ($Features$)
Input. A list of features to be considered, $Features$
Output. A subset of $Features$ with the highest accuracy
1. Set $SF := \{\}$, Quit := $false$
2. Repeat
      2.1. $\hat{SF} := SF$
      2.2. Select feature $F$ from $Features$ that achieves the
              highest accuracy measured using the 10-fold
              cross validation
      2.3. $SF := SF \cup \{F\}$
      2.4. If the overall accuracy using features in $SF$ does
              not improve the accuracy using features in $\hat{SF}$
           Then
              Quit := $true$
           Else
              $Features$ := $Features - \{F\}$
   Until Quit
3. Return $\hat{SF}$, set of features that yields the highest accuracy
```

The *forward feature selection* algorithm imposes an iterative process in which it starts without any feature to feed our recommendation system, i.e., offering an empty subset of best features to begin with. It then selects a random feature one at a time from the input set of features and checks if the selected feature gives the best prediction accuracy using 10-fold cross validation among all the remaining features in the input set. Once it finds the feature that gives the highest prediction accuracy (compared to other features), it adds the feature to the subset of best features and runs the process in a similar manner until any addition of more features stops showing any improvement in the prediction accuracy. Also, we do not want to overfit our model by introducing redundant features. After the best features are extracted, only the chosen features in an instance of a core university, which serves as an input to the KNN algorithm, are considered by the algorithm to generate other suggestions to its user, which is the central approach of our recommender.

IV. Experimental Results

We have conducted experiments to determine the accuracy and relevance of suggestions generated by our proposed recommendation system and if the suggestions are satisfactory to an applicant. In order to verify the accuracy and relevance, we adopt two evaluations.

For the first evaluation, we obtained data of students in the USA (through Edulix.com) who are already studying at a particular university/college and have been accepted by other graduate schools, which serve as the *gold standard* of our performance evaluation. We compared the suggestions made by our recommendation system with the gold standard and determined its accuracy. For the second evaluation, we asked offline volunteers, i.e., appraisers, to judge the relevancy of the suggestions made by our recommender. The volunteers judged the suggestions as "Relevant", "Likely Relevant", and "Non-Relevant". Through these two mechanisms, we analyze the performance of our recommender.

A. Online Evaluation

Based on the gold standard data, we analyze the accuracy of our multi-class SVM classification approach in finding the core university/college and the KNN algorithm in determining other appealing graduate school suggestions to a user, i.e., an applicant, using the Std_Info and Grad_Sch datasets, respectively.

1) An Ideal Classifier: In order to ensure that our multi-class SVM is the most ideal choice for our recommendation task in terms of determining the core university/college for an applicant, we compare the multi-class SVM with three other machine learning classification approaches in finding the core university/college in making recommendations. We introduce each of these classification algorithms below.

- C4.5. C4.5 is a basic decision tree learning algorithm. It is a statistical classifier that learns decision trees by constructing them top-down. The classifier [10] builds a tree based on the concept of information entropy and classifies data by following the path of the decision tree using the values of different attributes in the data. C4.5 adopts a greedy, non-backtracking approach in which decision trees are constructed in a recursive divide-and-conquer manner.
- Multilayer perceptron. They are another category of classifiers. These classifiers compose of a number of layers of neurons or units. Each neuron receives an input, applies a function to it, and then passes the output to another layer. The output layer then generates the output class for each instance. Backpropagation (BP) [10], which is based on the neural network learning approach, is one of the most popular multilayer perceptron algorithms. Advantages of multilayer perception include their high tolerance of noisy data and ability to classify patterns on which they have not been trained. They can be used when one have little knowledge of the relationships between attributes and classes.
- Naïve Bayes Classifier (NBC) [9]. NBC is a probabilistic classifier that assumes conditional independence among features based on the Bayes' theorem. The classifier, which predicts class membership probabilities,

Table III
Metrics for different Machine Learning classifiers, where ACC denoted Accuracy

Algorithm	Precision	Recall	F-measure	ACC
C4.5	0.058	0.061	0.059	0.340
MLP (BP)	0.001	0.013	0.001	0.050
Naïve Bayes	0.001	0.011	0.002	0.090
MC-SVM	**0.612**	**0.626**	**0.615**	**0.616**

provides a probabilistic approach to inference. Since it simplifies the computations involved, the approach is considered "naïve". Based on the highest predicted probabilities, a given data item is classified.

We have chosen Naïve Bayes as the representative of probabilistic-model classifiers, Backpropagation (BP) among the multilayer perceptron neural network models, and C4.5 for the tree-based models. Since we know the output labels of our dataset in advance, our model is a supervised-learning model. For this reason, we do not consider unsupervised methods like K-means clustering [2]. We train these classifiers to determine which trained model generates the highest accuracy in the determination of core universities/colleges, using the Std_Info training dataset with 14,000 instances, the same dataset used for training our multi-class SVM.

2) Classification: We used a test set of 2,000 instances in the Std_Info dataset as the gold standard from where the name, i.e., label, of each graduate school is the university/college that the corresponding student is currently enrolled and is supposed to be predicted by a classifier. We ran our trained SVM and the three other classifiers, i.e., C4.5, BP, and NBC, independently to suggest the label for each instance in the test set.[5] If the label of an instance in the test set matches the predicted output, then the classifier has learned well and generates relevant results. Table III shows the experimental results of the four classifiers on the test set, which clearly demonstrates that the multi-class SVM outperforms the others and the results are statistically significant ($p < 0.001$) [11].

B. Neighbor Generation

In developing our recommender, we are not trying to simply develop a classifying framework that generates just one label, i.e., one recommendation. Instead of simply finding one graduate school that satisfies the applicant's need, we are looking for multiple similar recommendations because we would like to provide multiple graduate school suggestions for each applicant to choose from.

In Section IV-A2, we have demonstrated that our trained multi-class SVM has correctly predicted more than half of graduate schools. Even though our trained classifier is a good measure on the performance of our recommender, only classification is not enough for the scope of our recommendation problem, since our goal is to generate more than one university/college suggestion for each applicant to consider. For example, if the actual label of a graduate school in the gold standard dataset is University of Florida (UF), we would like to recommend graduate schools which are similar to UF in terms of qualification and expectation, such as University of North Carolina, University of Washington, Ohio State University, and University of Southern California. We generate multiple suggestions so that a user can have a comprehensive list of potential graduate schools to examine, without having to look for it elsewhere. Hence, besides using the trained multi-class SVM to predict a graduate school for an applicant, we recommend multiple similar universities/colleges of the graduate school using KNN and measure the *Mean Sum Squared Error* (MSE) between the graduate school and its neighbors. MSE is defined as

Table IV
The MSE values for varying value of K nearest neighbors which determines the similarity between an instance and its neighbors

Iteration/Neighbors	3	4	5	6
1	0.155	0.169	0.300	0.236
2	0.119	**0.011**	0.330	0.616
3	0.123	0.200	0.170	0.162
4	0.010	0.100	0.200	0.170
5	0.094	0.110	0.290	0.290
Average	0.101	0.130	0.260	0.296

$$MSE = \frac{(X' - X_i)^2}{K} \tag{3}$$

where X' is the vector representation of the attribute values of an instance I, X_i is the vector representation of the attribute values of I's neighbor i ($i = 1, \ldots, K$), and K is the number of I's neighbors.

The *smaller* the MSE value, the *higher* is the similarity and the greater is the relevance of the neighbors. We run five iterations of our recommendation model on the gold standard data with varying values of K, and Table IV reports the results. The value of MSE shows that running the model for K = 4 yields the best result.

1) SVM with KNN: Our trained multi-class SVM and the KNN algorithm together generate a number of graduate schools which are the actual *ranked* recommendations. The number of neighbors of KNN used in our recommendation system is empirically decided. Besides using the MSE values, we run the KNN algorithm with varying value of K from 1 to 5 based on the Std_Info training dataset. We stop at 5, since the accuracy starts dropping as $K > 5$. (Note that the higher the value of K, the larger is the number of suggestions generated.) Our experiment demonstrates that K = 4 has the highest accuracy and is the optimal value of K, as shown in Figure 2, which is consistent with the K value chosen for the minimal MSE value as shown in Table IV.

[5] The test and training sets extracted from Edulix.com separately are disjoint subsets.

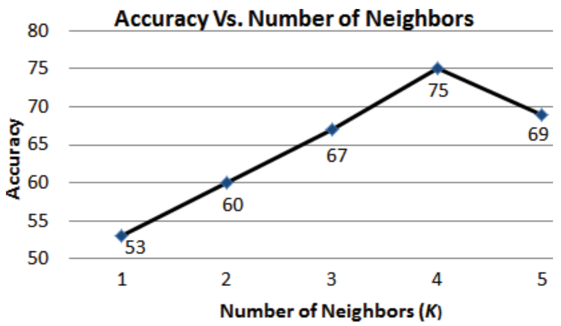


Figure 2. Accuracy versus number of neighbors to determine the optimal value of $K = 4$

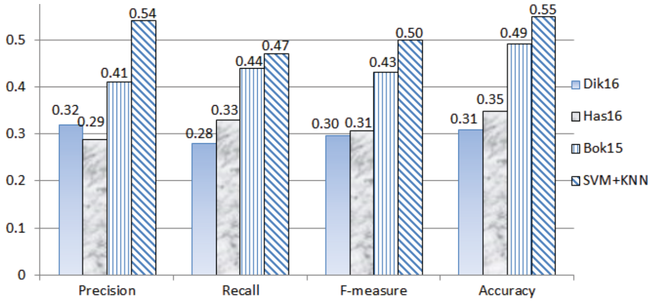


Figure 3. Performance Measures of Dik16 [5], Has16 [7], Bok15 [1], and SVM+KNN (our recommender) based on the *Std_Info* dataset

2) Recommendation Systems to be Compared: In this section, we compare the experimental results achieved by graduate-school recommenders introduced in Section II and our recommender.

- **Dikhale et al. (Dik16)** consider Naïve Bayes and Weka's J48 implementation of the C4.5 algorithm to generate recommendations. They use six features to create classification results for five colleges.
- **Hasan et al. (Has16)** use K-Nearest Neighbors (KNN) as the classifier as well as nearest neighbor extractor to suggest relevant graduate schools. Since they have not specified the ideal K value for their approach, we set K to be 4, which yields the optimal results for their recommender too.
- **Bokde et al. (Bok15)** adopt the multi-collaborative filtering approach, which is based on the Principle Component Analysis (PCA) mean and Higher Order Singular Value Decomposition (HOSVD) in matrix factorization (MF) to make recommendations. We have implemented their approach for comparison purpose.
- **SVM+KNN** is our recommender based on a trained multi-class SVM for classification and the K-Nearest Neighbors (KNN) algorithm for extracting closest neighbors, i.e., similar universities/colleges, for recommendation.

Figure 3 shows the Precision, Recall, F-measure, and Accuracy values of the four recommendation systems listed above using the (i) *Std_Info* test dataset that includes for each student in the dataset the graduate school the student is currently enrolled, in addition to other universities/colleges that the student have been admitted as discussed in Section III-C, and (ii) Grad_Sch dataset that includes the graduate school information of various universities/colleges based on which the KNN algorithm determines similar universities/colleges according to their requirements and qualification. We consider the overlap between the universities/colleges suggested by our recommender (the other three recommenders, respectively) for each student in the Std_Info test dataset against the graduate schools that the student is currently enrolled and schools that have admitted the student. We assume that if an applicant of a graduate school recommendation system is interested in a particular graduate school, the applicant is likely also interested in other universities/colleges who have admitted a student of the graduate school. Figure 3 clearly shows that our recommendation system outperforms the other three, and the results are statistically significant ($p < 0.01$).

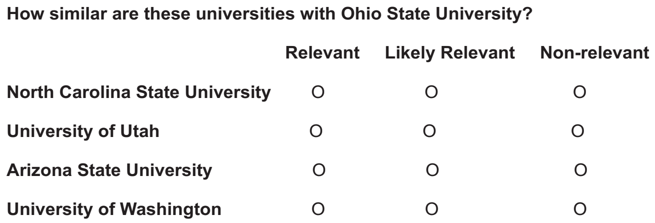
C. *User Study*

Besides the evaluations shown above, we conducted an online user study. We sent out an online survey to 100 computer science graduate students, who play the role of appraisers of our recommendation system,[6] at various universities in the USA who started school in or before fall 2017.

1) Relevant Versus Non-Relevant Suggestions: Our recommender first predicted the university/college an appraiser is currently enrolled based on the application data provided by the appraiser, and the prediction accuracy is **58%**, i.e., 58 out of the 100 universities/colleges currently attended by the corresponding appraisers were correctly predicted using the trained SVM. Hereafter, a list of other universities/colleges generated by using KNN is provided to the appraiser based on the predicted graduate school. (A sample of the survey for the neighbor universities/colleges suggested by our recommendation system is shown in Figure 4.) We asked each appraiser to mark the nearest neighbor suggestions as "Relevant" if they are similar to his/her graduate school,[7] "Likely Relevant" if they are somehow similar to his/her graduate school, and "Non-Relevant" if they are not at all related to the his/her graduate school. As shown in Figure 5, out of 1,200 ($= 100 \times 4 \times 3$) potential judgments, majority of the universities/colleges are marked as "Relevant" and "Likely Relevant", and only a small percentage of universities/colleges are marked as "Non-Relevant". The results are statistically significant ($p < 0.001$).

[6] The students are either former classmates of the first co-author or their friends, in addition to graduate students in the first author's department.

[7] The similarity refers to the strong likelihood that the appraiser would also consider attending the corresponding graduate school.



How similar are these universities with Ohio State University?

	Relevant	Likely Relevant	Non-relevant
North Carolina State University	O	O	O
University of Utah	O	O	O
Arizona State University	O	O	O
University of Washington	O	O	O

Figure 4. A sample of the survey circulated for user study

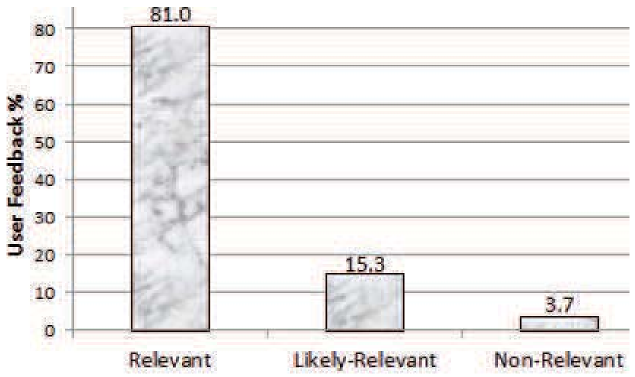


Figure 5. Percentage of the user study evaluation results split into Relevant, Likely Relevant, and Non-Relevant

2) nDCG (Normalized Discounted Cumulative Gain): Using the responses made by the 100 appraisers, we compute the $nDCG$, which is a widely-used measure for ranking quality, to determine the overall *ranking performance* of our recommender. We consider $nDCG_5$, as defined in Equation 4, for assessing the five ranked recommendations made by our recommender for each appraiser. $nDCG$ *penalizes* relevant suggestion that are ranked *lower* in the list of recommendations. The penalization is based on a relevance reduction, which is logarithmically proportional to the position of each relevant university/college in a ranked list (as shown in Equation 4). The *higher* the $nDCG_5$ score is, the *better* the ranking offered by our recommender is.

$$nDCG_5 = \frac{1}{5}\sum_{i=1}^{5}\frac{DCG_{5,i}}{IDCG_{5,i}}, \quad DCG_{5,i} = \sum_{j=1}^{5}\frac{(2^{rel_j}-1)}{log_2(1+j)} \tag{4}$$

Based on the responses provided by the appraisers, the overall average of $nDCG_5$ values on the recommended graduate schools is **0.59**, which indicates that majority of the useful recommendations are ranked in the top half.

V. Conclusions

Graduate school recommendation system, which suggests schools appealing to applicants without requiring the latter to search for them elsewhere, is a very effective and convenient tool for applicants. Unfortunately, only a handful of existing recommenders designed for finding appealing graduate schools have been developed in the past. In this paper, we have proposed a graduate school recommendation system which applies a multi-class SVM to classify a graduate school that is likely appealing to an applicant and the KNN algorithm to generate graduate schools with similar requirements and qualification. Our recommendation model is a unique one-step solution to applicants who are looking for graduate schools. We claim that our recommender is effective and advantageous as the suggestions generated by our recommendation system are relevant, and the user study has also verified that they are accurate. The graduate school data used by our recommender are easily accessible, which make our recommendation system applicable.

References

[1] D. Bokde, S. Girase, and D. Mukhopadhyay. An approach to a University Recommendation by Multi-Criteria Collaborative Filtering and Dimensionality Reduction Techniques. In *IEEE iNIS*, pages 232–235, 2015.

[2] S. Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, 2003.

[3] C. Chang and C. Lin. LIBSVM - A Library for Support Vector Machines. www.csie.ntu.edu.tw/∼cjlin/libsvm/, 2016.

[4] B. Dasarathy. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Comp. Soc. Press, 1991.

[5] N. Dikhale, H. Shrivastava, A. Dudhane, and U. Gholap. College Recommendation System. In *NCPCI*, pages 112–114, 2016.

[6] V. Franc and V. Hlavac. Multi-Class Support Vector Machine. In *ICPR*, pages 236–239. IEEE, 2002.

[7] M. Hasan, S. Ahmed, D. Abdullah, and M. Rahman. Graduate School Recommender System: Assisting Admission Seekers to Apply for Graduate Studies in Appropriate Graduate Schools. In *ICIEV*, pages 502–507, 2016.

[8] K. Kira and L. Rendell. The Feature Selection Problem: Traditional Methods and a New Algorithm. In *AAAI*, pages 129–134, 1992.

[9] C. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[10] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[11] M. Smucker, J. Allan, and B. Carterette. Agreement Among Statistical Significance Tests for Information Retrieval Evaluation at Varying Sample Sizes. In *ACM SIGIR*, pages 630–631, 2009.

[12] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison Wesley, 2006.

[13] H. Yu, J. Yang, and J. Han. Classifying Large Data Sets Using SVM with Hierarchical Clusters. In *KDD*, pages 306–315, 2003.

[14] M. Zhang and Z. Zhou. ML-KNN: A Lazy Learning Approach to Multi-label Learning. *Pattern Recognition*, 40(7):2038–2048, July 2007.