Hotness classification on RateMyProfessor.com

Zidong Wang

School of Information

University of Michigan

wzidong@umich.edu

Yuzhou Gong

School of Information  
University of Michigan

yzhgong@umich.edu

Seungbum Park

School of Information  
University of Michigan

purblue@umich.edu

**ABSTRACT**

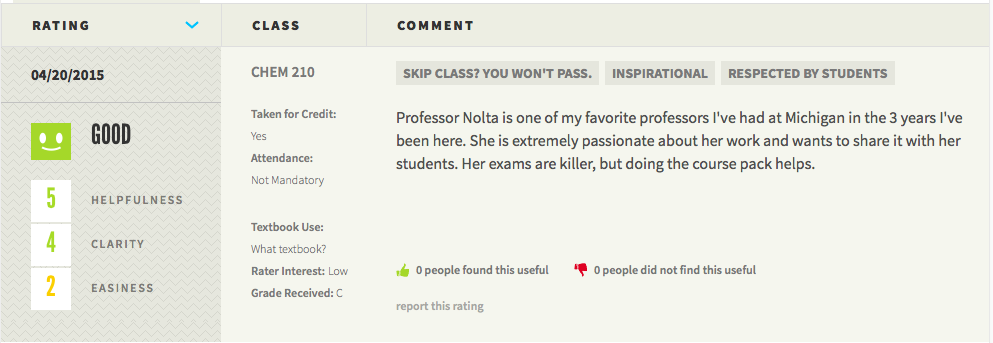
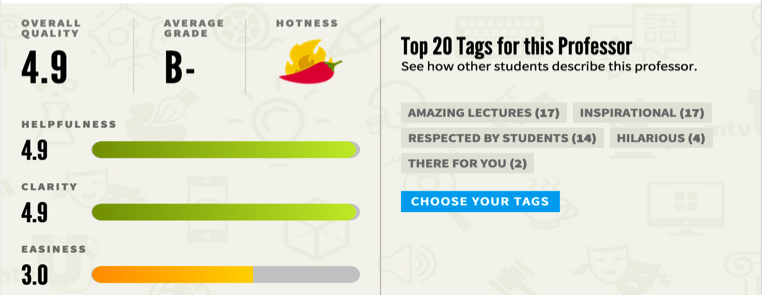


Figure . Example of ratings in RateMyProfessor.com

As well as typical faculty evaluation officially conducted by institution, new source of faculty rating data has recently become available on the Web, which has known as online faculty rating sites. RateMyProfessor.com is one of the famous online faculty rating sites. In addition to explicit measures of teaching effectiveness such as overall quality, helpfulness, clarity and easiness, the website also provide hotness that delivers much more subjective feature of faculty evaluation, which is how attractive a professor is.

**Keywords**

Keywords are your own designated keywords.

# INTRODUCTION

In the context of the evaluation of professors and their classes, it has been important to both faculty and students since for faculty, it is one of major consideration for promotion, and for students, they can decide which classes to take. Typically, the evaluation has been based forms that are filled out anonymously by students in a classroom using formal, well-defined and controlled processes [1]. However, as well as the typical resources, new source of faculty rating data has recently become available on the Web, which has known as online faculty rating sites. In these sites, students can informally share information about professors and their lectures based on their personal experiences. Since students can be much freer and open, different from the traditional evaluation process, many students have used it to evaluate faculty and decide lectures fitting well to their desire, while the information is hard to be officially used in institution due to their inherent bias. Due to its new possibility, there have been various researches to verify their effectiveness [1].

RateMyProfessor.com is one of the famous online faculty rating sites. The site has several categories in which a professor is ranked: overall quality, helpfulness, clarity, easiness and hotness. Figure 1 shows an example of ratings and comments in the site. While most of measures directly indicates teaching effectiveness of a given professor, the hotness factor identified by red-chili denotes interesting characteristic, how attractive a professor is. In fact, the hotness is meant to be not only physical beauty, but also other various subjective characters of professors, such as passion about their subject, which are possible to have great impact on students’ decision when class registration. While attractiveness is something that tends to be different depending of the viewer, there is often a general society-created standard. Therefore, we want to analyze which features are important for the hotness, and based on the analysis, we try to predict the hotness.

Essentially, the problem is a binary classification problem labeled by the hotness, which is either hot or not hot. In order to analyze the hotness, we first divide entire data into three types: numeric, categorical and text data. Numeric data provides quantitative information of teaching effectiveness, such as overall quality of a professor, helpfulness, clarity, and easiness. Categorical data refers to tags that students choose, and there are 20 pre-defined tag set. Lastly, text data means all text reviews of a professor. The reason why we try to divide the data into the three types is that different characteristics depending on the types lead to different approaches. Therefore, in order to evaluate our approaches, we first apply various classification algorithms into a single large matrix that contains all three data types. After that, we try to train three classifiers for the three data types separately, and then integrate them with dynamic classifier selection algorithm.

In Section 2, we first formulate our problem statement, and in Section 3, we provide specific information of our data. In Section 4, we elaborate our classification approaches. Section 5 evaluates our classifiers and analyzes the evaluation results, and finally, Section 6 concludes the paper.

# PROBLEM STATEMENT

# DATASET

We directly gather data from RateMyProfessors.com using our own crawler since the official API of RateMyProfessors.com is not available any more. We target professors in public universities in Michigan, Illinois, and California. For each professor, we crawl

not only overall scores, such as hotness, quality, average grade, helpfulness, clarity, easiness, and top 20 tags, but also all individual comments, which contains various numeric measures like the overall scores, selected tags and text reviews. As a result, we get 33,565 professor data in total, which are labeled by hotness (either hot or not hot), and all comments for each professor. Average number of comments per professor is 12.1 with 20.3 standard deviation, and maximum number of the comments is 737. In addition, since 70% of our test data are labeled as “hot”, a baseline of classification would be 70%, which equals to the accuracy if we forcibly predict all the data as “hot”. We used 80% of entire data as training set, and the remaining as test set with maintaining the proportion of the labels. Finally, we divide data into three types in order to use different approach depending on the type: numeric, categorical (or tag) and text data.

## Numerical data

For numeric data, we use the 8 aggregated scores including overall quality, helpfulness, clarity, easiness, average interest, average textbook-used-or-not, average taken-for-credit-or-not and number of comments. Note that we do not use average Grade since 70% of data are missing, while other variables do not contain missing values. In addition, since the distribution of the number of comments approximately follows the power law, we normalize the variable with log scale as follows:

Specific description and distribution of the numeric variables are summarized in Table 2.

## Categorical Data

Tag features are categorical data. A student could choose tags from a pool of 20 and attach them to a professor. A professor’s tags would be the union of all the tags he gets from students. Therefore, the dimension of tag matrix is 20, each column corresponding to a tag. And the numeric value of that column is intuitively how many that tags the professor gets.

## Text Data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Variable Name** | **Min** | **Mean** | **Median** | **Max** | **Std.dev** | Distribution |
| Overall Quality | 1.0 | 3.732 | 3.9 | 5.0 | 1.01 |  |
| Helpfulness | 1.0 | 3.763 | 4.0 | 5.0 | 1.06 |  |
| Clarity | 1.0 | 3.696 | 4.0 | 5.0 | 1.037 |  |
| Easiness | 1.0 | 3.152 | 3.1 | 5.0 | 0.91 |  |
| Interest-level | 0.0 | 3.314 | 3.364 | 5.0 | 0.93 |  |
| TextBookUse-OR-Not | 0.0 | 2.305 | 2.250 | 5.0 | 1.42 |  |
| TakenForCredit-or-Not | 0.0 | 0.9178 | 1.0 | 1.0 | 0.22 |  |
| Number of Comments | 1.0 | 12.16 | 6.0 | 737.0 | 20.64 |  |

For text reviews, we aggregate all text reviews into a single text for each professor, and then apply several natural language processing steps. After stop word and special character removal, and lemmatization with NLTK package, we vectorize the text data with both unigram and bigram for feature selection by using Scikit-Learn package [2]. Since the dimensionality is about 1.3 million, we apply Chi-Square feature selection in order to reduce the dimension.

## Matrix Representation of Data

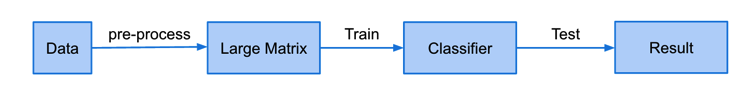
As we mentioned in the section 2, we represent the three data types using two different matrix representations. Let and be dimension of numeric, categorical and text data respectively. The first one is a large single matrix including all the three types of data with dimension. This representation allows us to apply a single classifier to the data. The second one is representing three data types into three separate matrices , , and in order to apply separate classification approaches to each data type.

# CLASSIFICATION APPROACHES

In this section, we provide two different approaches for classification; one is using a single classifier on a single large and another one is using three separate classifiers into each dataset and then integrates the prediction results with dynamic classifier selection.

## Single Classifier

Although we have 3 feature groups, numerical, tags, and text, a classical approach would be considering them equally and combining the 3 feature groups into a big matrix. The pipeline flow of this approach is shown in figure below.



We employ different underlying classifiers in Scikit-Learn package [2], which are Multinomial Naive Bayes, Gaussian Naive Bayes, Logistic Regression, and Random Forest to implement this method. Naive Bayes is a popular classifier based on strong independence assumptions between features. In other words, this classifier uses “word bags” concept that makes it usually performs well on text features. Logistic Regression method is known to perform well and run fast when applied to binary classification. Random Forest is an ensemble learner, which could effectively avoid over-fitting problem, though it takes longer to train especially when features dimension is high. We finally decide to use those 4 classifiers to implement this approach and make comparisons.

## Dynamic Classifier Selection

Dynamic classifier selection is to take advantages of the strengths of the individual classifiers in multiple classifier systems. In the ideal case, let us consider an *oracle* that always knows classifiers that give correct predictions for all test patterns. If this is available, we can always avoids weakness of each classifier and improve classification performance, so it is the ideal case of dynamic classifier selection. There has been many researches related to this area, and there are four major approaches [3,4,5]; overall local accuracy (OLA), local class accuracy (LCA), and a Priori selection method (A Priori), and A Posteriori selection method (A Posteriori). Even though they have different characteristics, they basically use a similar concept that finds local region of a given test pattern from training patterns and then estimates each classifier’s local accuracy in the local region. Based on the estimated local accuracy, we choose the most suitable classifier for the given test pattern. To find local region is done with the K-nearest neighbor. Therefore, the differences between those four methods come from ways to estimate local accuracies. In this section, we briefly describe those four methods.

Table . Distributions of numeric data

### Overall Local Accuracy (OLA)

This method estimates local accuracy from the probability of correct classified patterns in the local region. Let us consider the local region formed by k-nearest neighbor. In this region, suppose that out of the total samples in the neighborhood are correctly classified by classifier . Then OLA estimates the local accuracy as following [4]:

Therefore, this method does not take account the prediction results assigned by classifier , but only local accuracy in the neighborhood.

### Local Class Accuracy (LCA)

This method is similar with OLA, but estimate the local accuracy in terms of output classes assigned by classifier . If classifier assign the test pattern **X** to a class , the local accuracy is

where is the number of neighborhood patterns that have been correctly assigned by to the class , and is the total number of neighborhood patterns that have been assigned to the class by [4]. In other words, this method focus on conditional probability of neighborhood patterns given the class by which the test pattern is assigned. Therefore, it gives the proportion of neighborhood patterns assigned to by that have been correctly classified.

### A Priori Selection Method (A Priori)

This method is essentially extension of OLA, but weight by distances between the test patterns and neighborhoods. Given a pattern belonging to the neighborhood, the provided by the classifier can be regarded as a measure of the classifier accuracy for the test pattern **X**. If there is *N* number of neighborhood, the local accuracy estimate of OLA can be reformulated as follows [4]:

With the weighed posterior probability by the Euclidian distances of the patterns form ***X*** [4]:

where . In other words, this method extends OLA with probabilistic representation of classifier results and weighting scheme with distance between the test and neighborhood patterns. Therefore, this method still does not use the prediction result of .

### A Posteriori Selection Method (A Posteriori)

This method is basically similar with LCA but uses probabilistic representation of prediction results and the Euclidian distance weights, similar with A Priori. Similar with the previous method, we can reformulate LCA with the class posterior probabilities, instead of the simple estimation [4]. From the Bayes theorem:

where is the probability to correctly classify the patterns belonging to the class , and it can be estimated as follows [4]:

In addition, the prior probabilities can be estimated as follows:

Therefore, after apply the Euclidian weights similar with A Priori, the final selection condition is

### An Algorithm for dynamic classifier selection

To decide the most appropriate classifier, for a given test pattern, we find a classifier that gives the maximum value of , which is computed by one of the four methods. To implement the algorithm, we follow the entire algorithm for classifier selection provided in [4] including tie breaking.

# EVALUATION

In this section, we evaluate our two classification approaches. We first discuss on performances of each approach, and then analyze the result in order to capture characteristics of the problem.

## Evaluation Setting

As we mentioned in the section 2, we use 80% of entire data as training set, and 20% of them as test set. For model selection, we use stratified 5-fold cross validation.

## Single Classifier

In terms of evaluation for the single classifier approach, we use cross-validation score to represent model accuracy. Since there are three different feature groups, we also evaluate the model based on leave-one-out method to look deep into interaction among feature groups. For this approach, we use accuracy as evaluation metric that is the fraction of correct answers.

Evaluation result is shown in Table 2, from which we could judge that Gaussian Naive Bayes yields best accuracy among 4 classifiers. This implies that text features are potentially following Gaussian distribution. In terms of leave-one-out method, we did not see huge decrease in accuracy when excluding numeric feature or tag feature from consideration. However, as the last column shows, accuracy decreases significantly when we leave text feature out of count. Therefore, text feature is shown to be the most informative group in our experiment.

Table 2. *Cross Validation Score of Single Classifier Method*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **All Feature Groups** | **No Numeric** | **No Tags** | **No Text** |
| **Multinomial NB** | 0.715 | 0.742 | 0.724 | 0.557 |
| **Gaussian NB** | 0.84 | 0.84 | 0.84 | 0.565 |
| **Logistic Regression** | 0.773 | 0.759 | 0.773 | 0.737 |
| **Random Forest** | 0.738 | 0.732 | 0.736 | 0.718 |

## Dynamic Classifier Selection

# CONCLUSION

Our thanks to ACM SIGCHI for allowing us to modify templates they had developed.

# REFERENCES

1. Otto, James, Douglas A. Sanford Jr, and Douglas N. Ross. "Does ratemyprofessor. com really rate my professor?." *Assessment & Evaluation in Higher Education* 33.4 (2008): 355-368.
2. Scikit-learn, http://scikit-learn.org
3. Ko, Albert HR, Robert Sabourin, and Alceu Souza Britto Jr. "From dynamic classifier selection to dynamic ensemble selection." Pattern Recognition 41.5 (2008): 1718-1731.
4. Giacinto, Giorgio, and Fabio Roli. "Methods for dynamic classifier selection." *Image Analysis and Processing, 1999. Proceedings. International Conference on*. IEEE, 1999.
5. Woods, Kevin, Kevin Bowyer, and W. Philip Kegelmeyer Jr. "Combination of multiple classifiers using local accuracy estimates." *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*. IEEE, 1996.
6. Sannella, M. J. 1994. *Constraint Satisfaction and Debugging for Interactive User Interfaces*. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.