Whisky Recommender

Alex Omid-Zohoor, Ashkan Eghtesadi

Abstract

A whisky recommender system for users at whiskybase.com is implemented. Two separate algorithms are designed using content-based filtering and collaborative filtering. These two algorithms are then optimally combined in a linear blend to maximize performance. The final algorithm achieves a 33% improvement over predicting the average rating assigned to a whisky.

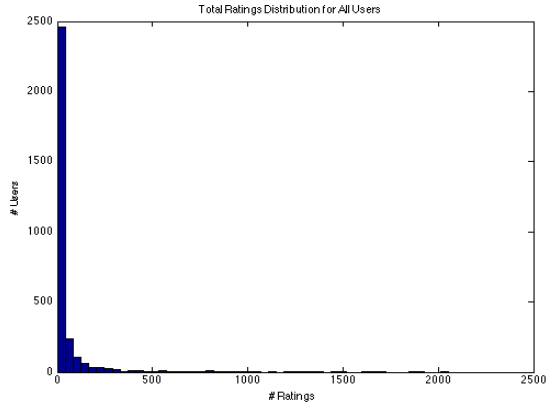
1. Introduction

We aim to build a whisky recommender system to predict the rating that a user would assign to a particular whisky. Broadly, there are three main approaches to such recommendation systems: content-based filtering, collaborative-filtering, and hybrid (combining content-based and collaborative-based filtering) [1]. We have separately implemented and optimized content-based filtering and collaborative filtering algorithms, and then combined them into a hybrid model.

2. Data

We collected data from whiskybase.com—"The Biggest Whisky Database of the World"—which is a website where users can rate scotch whiskies on a scale of 0 to 100. There are a total of 3,091 active users and 26,103 unique whiskies that have been rated. Since the data is not publicly available in a clean form, we wrote python scripts to scrape and organize relevant data from the website.

The vast majority of users in our dataset rated relatively few whiskies. As shown in the histogram below, there is a roughly exponential relationship between the number of ratings (per user) and number of users in our dataset.



We computed the average number of ratings per whisky to be around 7, with a maximum of 423 ratings for the most popular whisky.

We also collected the average rating for each whisky (across all users), as provided on whiskybase.com. We assume that most users consider this metric when deciding whether they would enjoy a particular whisky. Thus, we use this metric to define the target performance of our recommendation system. We aim to build a recommender system that can better predict a user's rating of a whisky than that whisky's average rating.

3. Content-Based Filtering

3.1 Linear Regression

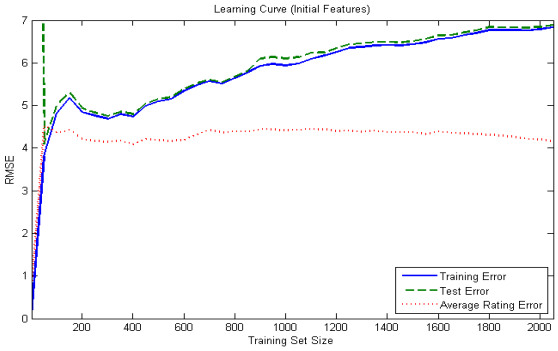
In content-based filtering, we train a model for each user using only that user's previous whisky ratings. To gauge the achievable performance of content-based filtering, we focus on the user with the most ratings (2,052), whom we refer to as the golden user. For each of the golden user's rated whiskies, we extract features and train a model (θ) based on linear regression, in which:

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x.$$

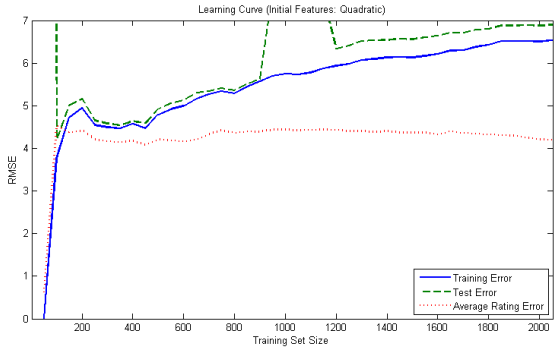
In all cases, we include an additional feature, $x_0 = 1$, known as the intercept term.

3.2 Features and Results

Since our data was not provided, but needed to be carefully scraped from whiskybase.com, we started with the following initial features which could be most easily obtained: age, alcohol content, and price. To boost performance (in all subsequently described linear regression model training), we perform basic feature selection by iterating through all subsets of features, and choosing the one that minimizes leave-one-out cross-validation (LOOCV) test error. We use LOOCV, rather than hold out cross validation, since most users have far fewer ratings than the golden user. Using these initial features, we generated the following learning curve:



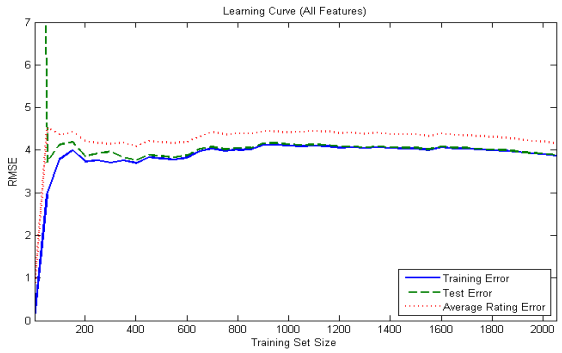
This learning curve shows a large gap between the test error and target performance (the error obtained by simply predicting each whisky's average rating), which suggests high bias [2]. To address this, we added quadratic features and obtained the following learning curve:



As expected, this reduced training error, but actually increased test error. So using higher order features lead to over-fitting and increased variance. Based on these analyses, we concluded that a larger set of features was needed. Thus, we developed a more sophisticated script to scrape the following features from each whisky's associated page on whiskybase.com:

1. Number of Notes (User Comments)
2. Regional District of Origin
3. Alcohol Content (% ABV)
4. Average Rating
5. Page Views
6. Availability
7. Size (mL)
8. Price ($)
9. Rank
10. Age

Using this more complete set of features, we obtained the following learning curve:



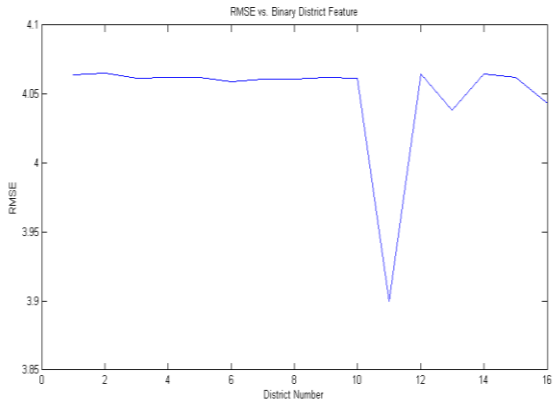
Adding richer features significantly reduced bias. On this user, our content-based model achieved a **7% improvement** in RMSE over simply predicting each whisky's average rating.

3.3 Feature Preprocessing

In order to expand our feature set, it was necessary to perform basic feature pre-processing. Certain raw features were not available for all whiskies. For example, a small number of whiskies had unlisted Age, Rank, or Price.

Many young or blended whiskies are intentionally released without an age statement. Thus, for Age, we replaced missing values with a default value of 8 years, which represents a reasonable lower bound on typical whisky age. For all other unlisted features, we replaced missing values with the feature's average value.

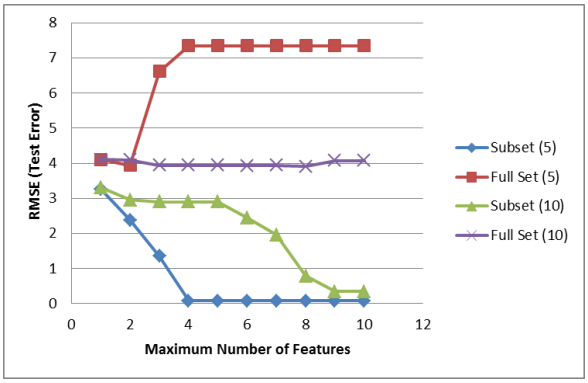
Unlike the other nine features, the raw Regional District of Origin feature was not a numerical quantity, but rather a text string with one of sixteen unique values. To quantify this feature, we assigned it a binary value (1 if district "x", 0 otherwise). Using LOOCV on the golden user, and sweeping the district associated with this binary feature, we generated the following plot:



Based on this plot, we associated the most discriminative district (11) with our binary Regional District of Origin feature. This district corresponds to the Islay region of Scotland, which makes intuitive sense, as the Islay region is distinctly known for producing intensely smoky and peaty whiskies [3].

3.4 Overfitting Considerations

Since many users have fewer ratings than available features (10), we also analyzed the effect of overfitting. Even though our feature selection chooses the subset of features that yields the lowest test error (as opposed to training error), if we allow the number of features to exceed the number of training examples, we still run the risk of overfitting our model. To illustrate this, we randomly chose subsets of 5 and 10 ratings from our user's 2,052 total ratings. We then performed feature selection on the subsets of 5 and 10 ratings, imposing a maximum allowed number of features. Finally, we performed LOOCV on the full set of ratings using the features selected from each subset.



The plot above shows that although increasing the maximum number of allowed features monotonically reduces the test error for the subset, it eventually tends to increase the test error on the full set. The minimum test error for the full set roughly corresponded to setting the maximum number of features to one half of the training set size. Thus, for users with fewer ratings (m) than available features (n), we set: $Max\ Number\ of\ Features = \lfloor m/2 \rfloor$

4. Collaborative Filtering

4.1 Background

In the realm of collaborative filtering, we use the whisky ratings of the community to predict the whisky ratings of individual users. Thus, to begin, we construct no feature vectors and instead rely solely on user ratings. To do so, first we collected all the ratings in the whiskybase.com database and stored them in an m x n matrix M. We have m = 3,091 active users and n = 26,103 rated whiskies with 171,797 total ratings. There are over 80 million entries and only about 200,000 are nonzero, so we have a very sparse matrix (0.2% density of non-zeroes). Therefore, it is attractive to use low-rank approximation as a prediction algorithm [4]. Formally, for matrices $X \in \mathbb{R}^{m\times r}$ and $Y \in \mathbb{R}^{n\times r}$, we aim to minimize the objective function:

$$J(X,Y) = \frac{1}{2}\|M - XY^T\|_F^2 + \frac{\lambda}{2}\|X\|_F^2 + \frac{\lambda}{2}\|Y\|_F^2$$

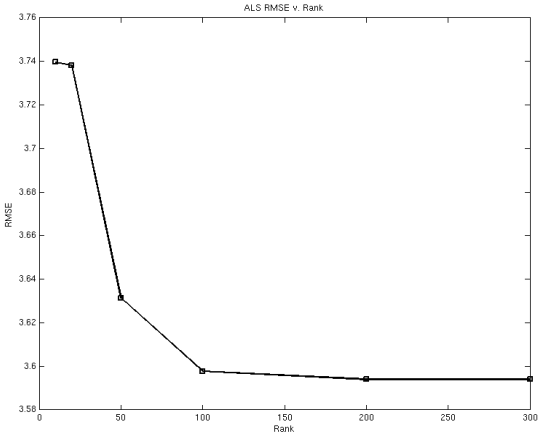
Where we introduce λ as a regularization parameter, and define the Frobenius norm $\|\cdot\|_F^2$ as $\sum_{i,j} A_{ij}^2$ for any matrix A. We can solve this by iterating through the rows of X and Y (denoted as x_i and y_j) as such:

$$(\lambda + \sum y_j y_j^T)x_i = \sum M_{ij} y_j$$
$$(\lambda + \sum x_i x_i^T)y_j = \sum M_{ij} x_i$$

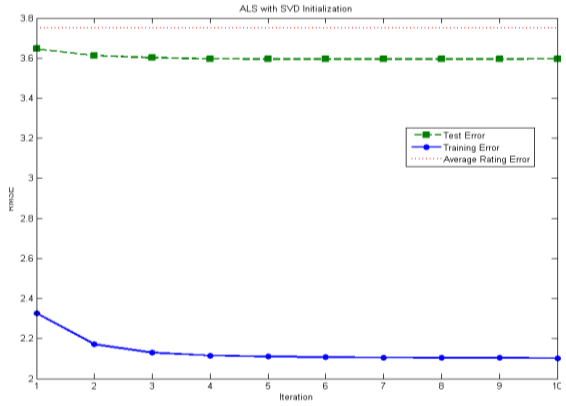
And solving for x_i and y_i using least squares. For this setting, we define a new RMSE. Let |S| denote the size of the training set, then: $RMSE_{cf} = \sqrt{\frac{1}{|S|}\sum_{i,j}\left(\widehat{M}_{ij} - M_{ij}\right)^2}$ where M_{ij} is the rating user i gave to whisky j and $\widehat{M}_{ij}$ is our prediction. To obtain training and test sets, we used 20% hold out cross-validation.

4.2 Results

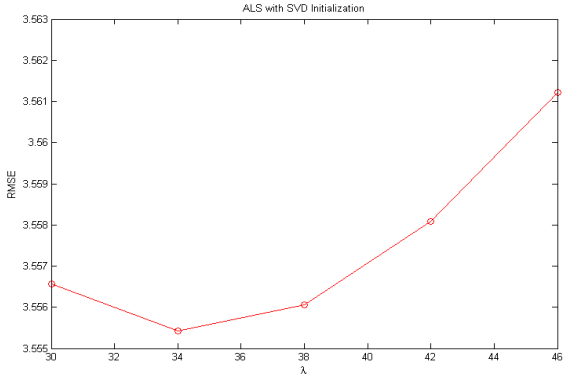
Since we do not know the rank a priori, we swept rank and found that a rank of 100 could sufficiently approximate M, as shown in the following plot:



The following plot shows typical convergence performance of the algorithm. X and Y are initialized using the SVD of M, and we observe convergence within around 5 iterations.



The red-dashed line in the previous plot shows the target performance (the error obtained by simply predicting each whisky's average rating), which this algorithm is able to beat. Finding an optimal value (34) of the regularization parameter λ increases our performance further, as shown in the following plot:



The final test error is 3.556, which represents a **5% improvement** in RMSE over simply predicting each whisky's average rating.

5. Hybrid

Combining content-based and collaborative models has been shown to improve the performance of recommender systems [5]. We implemented a naïve hybrid model by using a simple linear blend of our content-based and collaborative models:

$$h_{hybrid}(x) = c * h_{content}(x) + (1 - c) * h_{collab}(x),$$

$$0 \leq c \leq 1$$

Since the performance of content-based vs. collaborative models varies from user to user, we swept the weighting parameter, c, to optimize overall performance for each user.

6. Results and Conclusions

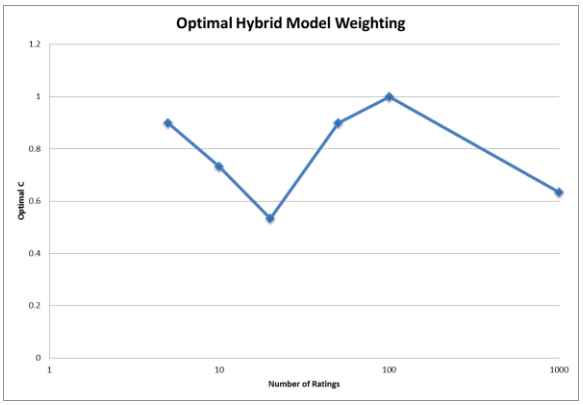
In order to test the performance of our various methods across a full spectrum of user demographics, we randomly selected 18 users (3 each from 6 subgroups). The subgroups, based on total ratings per user, were 5, 10, 20, 50, 100, and 1000. The graph below summarizes the achieved performance[1].

It seems that in general, the content-based model achieves better performance than the collaborative model. This is especially apparent for users with fewer ratings, which seems to make sense, as beginner whisky drinkers may exhibit stronger biases that deviate from the norm, which are more easily captured by a content-based model. For example, one common trend would be for beginner whisky drinkers to overrate whiskies, due to a lack of experience.

To support the claim that the content-based model performs better on users with fewer ratings due to stronger biases or limited experience as opposed to overfitting, we compared the average test error for users with 5 or 10 ratings with the average test error for the random subsets of 5 and 10 ratings taken previously from the golden user (2,052 ratings). Indeed the test error for users with 5 or 10 ratings (RMSE = 2.33) was less than the test error for the subsets of 5 and 10 ratings taken from the golden user (RMSE = 2.64).

It should be noted that although the content-based model generally performs better, the collaborative model does still significantly improve performance when combined in the hybrid model. The graph below shows that the optimal weighting of the hybrid model varies, but generally favors the content-based model, as expected.

[1] Please see Appendix for details on system performance measurement methodology.



On average, the final hybrid model achieved a **33% improvement** over simply predicting each whisky's average rating.

7. Future Work

7.1 Further Testing

In the future, we plan to first evaluate the developed content-based, collaborative, and hybrid models on a larger subset of users to reduce noise. Although 18 users is a reasonable starting point, increasing the size of this subset would obviously increase our confidence in the measured system performance.

7.2 Content-Based Filtering

In order to improve content-based filtering performance, we plan to add higher order features. Of course, since the 10 linear features would generate a very large number of higher order features, we would also need to implement a more sophisticated version of feature selection, such as forward search, to reduce computational overhead.

7.3 Collaborative Filtering

Though we initially implemented collaborative filtering using a low-rank approximation algorithm, we plan to also explore other methods, such as *k*-nearest neighbors (k-NN).

7.4 Hybrid

Once we have implemented k-NN collaborative filtering, we could also implement a hybrid approach known as content-boosted collaborative filtering. In this approach, all zero entries in the original sparse ratings matrix are replaced with their content-based predictions. Then k-NN collaborative filtering is performed on this dense matrix. It has been shown that this approach can outperform a simple linear blend [6], like the one we originally implemented.

Appendix

When individually evaluating the content-based filtering and collaborative filtering models, we used LOOCV and 20% hold out cross-validation, respectively. However, in order to evaluate a hybrid model, we needed to use a consistent methodology for both algorithms before linearly blending them. For the sets of users with 5 and 10 reviews, we performed LOOCV on both the content based method and the collaborative method. For the sets of users with 20, 50, 100, and 1000 reviews, we performed 20% hold out cross-validation on both methods (using identical holdout sets). We chose not to perform LOOCV on the sets of users with more reviews, due to the increased time required for computation and the diminishing value of LOOCV for larger datasets.

References

[1] Jafarkarimi, H., Sim, A.T.H., and Saadatdoost, R., *A Naïve Recommendation Model for Large Databases*, International Journal of Information and Education Technology, June 2012

[2] Ng, A., *Advice for Applying Machine Learning*, CS229: Machine Learning. Stanford University. 2013.

[3] Jackson, M., *Michael Jackson's Complete Guide To Single Malt Scotch*, (Running Press Book Publishers, 2004), pp. 48.

[4] Montanari, A., *Recommendation Systems*, EE378B: Inference, Estimation, and Information Processing. Stanford University. 2013.

[5] Bell, R.M., Koren, Y., and Volinsky, C. *The BellKor solution to the Netflix Prize*. Technical report, AT&T Labs, 2007. http://www.research.att.com/˜volinsky/netflix

[6] Melville, P., Mooney, R. J., and Nagarajan, R. *Content-Boosted Collaborative Filtering for Improved Recommendations*. Proc. of the 18th National Conference on Artificial Intelligence (AAAI2002), pp. 187–192, 2002