Wine Recommendation for CellarTracker Course Assignment of CSE255

Ruogu Liu
Department of Computer Science and Engineering
University of California, San Diego
Email: rul052@eng.ucsd.edu

***Abstract*—This article mainly describes the process of trying to build a model for wine recommendation system. Based on people's review on various wines, it is possible to learn their taste and predict what other kinds of wines they would prefer. Besides, wine is such kind of product that closely related to time. Production time, wine brand as well as reviewers' reviewing time are all influencing people's choices. In this article, I will introduce the wine review data-set I am going to use, the exploration I have done on it, what prediction task I am going to solve, related work done by others, what criteria my model will be based on, how to train the model and final result and conclusions.**

Keywords*—*wine, recommendation, CellarTracker.

I. Introduction

Recommending system is going to be a must-have part in today's online shopping website. Good recommendation will stimulate the shopping desire of its users, or keep users on the website longer. Wine recommendation is among those difficult tasks which people's preferences are hard to predict: some user prefer much sourer flavor and will praise a vinegar more than oridinary reds. *CellarTracker* is a website that stores information about wines and wine collections. Created in 2003 by Eric LeVine, *CellarTracker* has grown to be one of the world's most comprehensive wine databases. It provides a great number of user reviews on various wines. Based on that, this project will try to find a model which recommend wines to old and new users.

II. Dataset Description and Exploration

The dataset I'm going to use is provided by SNAP: Stanford Network Analysis Project[1]. It consists of 2,025,995 wine reviews from *CellarTracker*, involving 44,268 users and 485,179 kinds of wines. This dataset spans a period over 10 years, up to October, 2012. Every piece of review data includes 9 cells of information:

1 name of the wine
2 wine unique id
3 variant of the wine
4 production year of the wine
5 reviewers name
6 reviewers id
7 the points given to this wine
8 UNIX time for this review
9 the review text

A. Data Preprocessing

First of all, the representation of this dataset needs to be changed to a more organized way which is easy for program to load and parse. Basically, each of those reviews can be divided into two parts: information of wine and information of user's review. So the data should also follow this hierarchy. The parsing process took about 90 seconds and generate a 898 Megabytes Json file. The review hierarchy will be:

1. wine
   (1) name of the wine
   (2) wine unique id
   (3) variant of the wine
   (4) production year of the wine
2. review
   (1) reviewers name
   (2) reviewers id
   (3) the points given to this wine
   (4) UNIX time for this review
   (5) the review text

After those very first preprocessing, some basic explorations can be done on the dataset, which are helpful to reveal some pattern underneath those reviews and ratings, make decision of what to predict and which features are good to do the prediction.

B. Production year and average rating

Figure 1 depicts the relationship between the production year of wines and their corresponding average rating points. The x-axis is the production year of wines and the y-axis is the average rating for certain wine. The size of each circle shows the relative number of reviews on those wines. Several facts can be immediately read from this figure: old wines are higher rated than newly produced wines; users are reviewing more new wines than old wines.

C. Length of review text and average rating

It seems that wine year is a good indicator of users' rating points. Let's keep digging. The idea that users' review text can store great amount of information on their attitude towards certain wines leads to my next experiment. The length of review text in characters is a very basic feature showing reviewers' attention. Figure 2 is the visualized result of it.

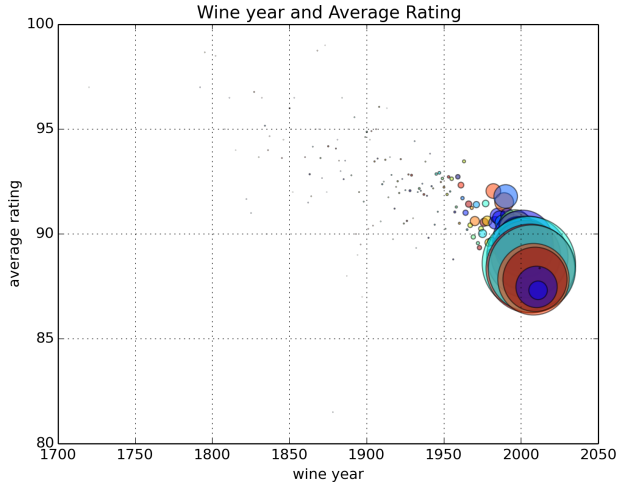

Fig. 1. Relationship between wine year and average rating

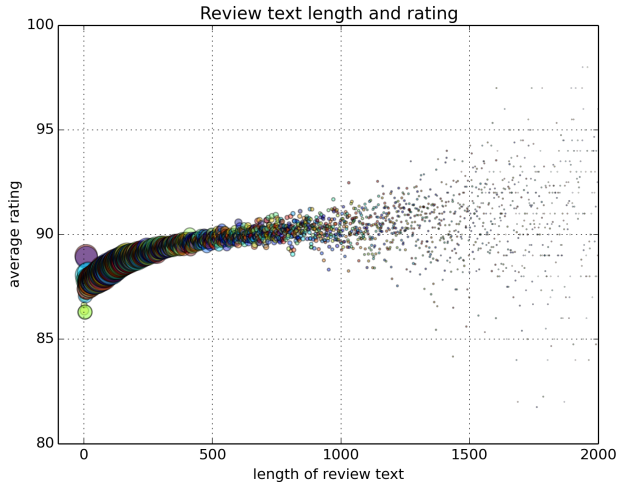

Fig. 2. Relationship between length of review text and average rating

The figure is just like a horizontal grown tree rooted from "no review" and expanded to 2000 or more characters. Before the length reaches approximately 1000, the average rating tend to be increase with the length. After that, as length of review text grows, the rating points diverge: some tend to be higher while others tend to be lower. The size of the circle corresponds to the amount of reviews with such length, and it's trend is very clear: most people would like to give short or no reviews while there are some diligent reviewers or enthusiastic wine lovers who spend their time writing long reviews to both their favored reds and disliked wines.

But the length of review text may not be available at the time when the system do the recommendation, since users will rate the wine first and write the review after that. So the time we can see the review of the $< user, wine >$ pair we are going to predict is too late.

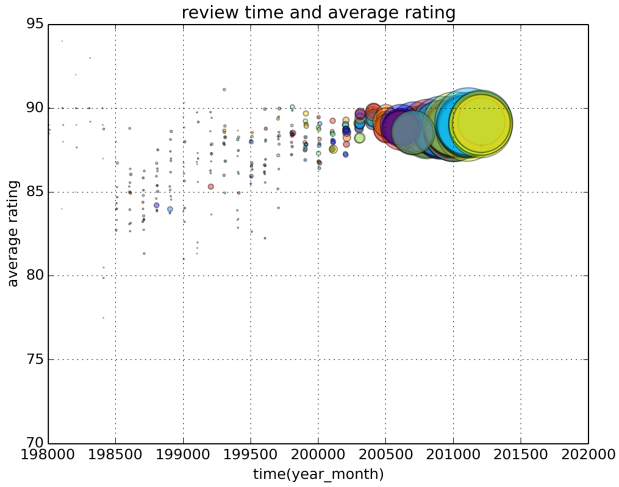

Fig. 3. Relationship between reviewing time and average rating

D. Reviewing time and average rating

Next thing that I would like to know is that whether the reviewing time matters. My guesses are as time goes by, people's choice and preference would change somehow: some people begin to become more mean while others are becoming more generous. So reviewing time may be a good predictor on people's reviewing points. Figure 3 is the result including reviews dating back to 1980s.

However, it is weird that reviewing time would be more ancient than the website! As far as I could guess is that during some point of the website running, it changed the way of representing time to UNIX time. So I tried a zoomed version, leaving out all data before the year 2003. The result is shown in Figure 4.

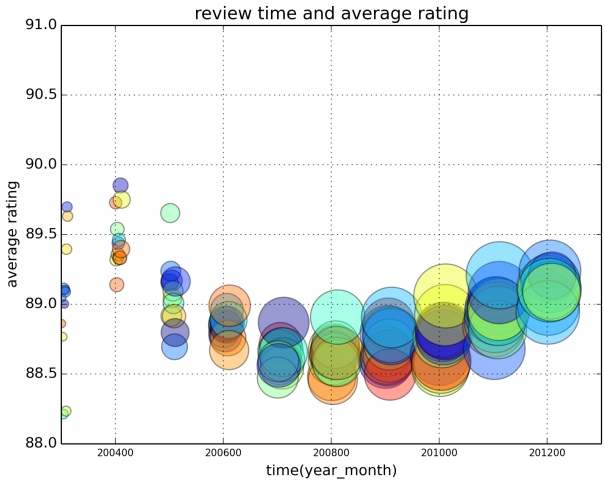

Fig. 4. Relationship between review time and average rating

It shows that since 2003, the number of reviews each year

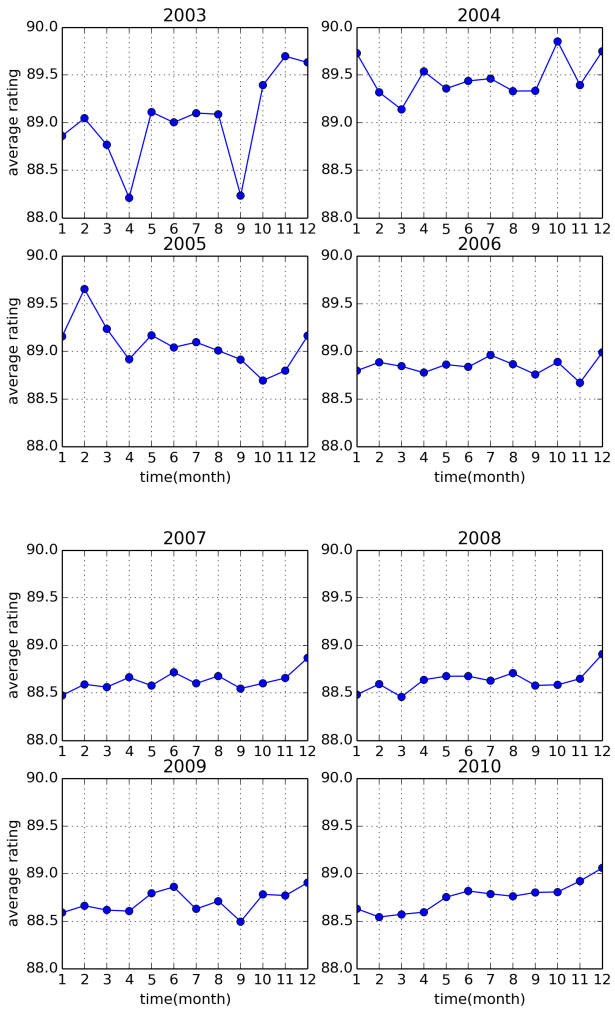

Fig. 5. month and average rating over eight years

increase a lot. In the beginning, people tends to give higher point to wines, but then, maybe they become more strict or better understand how to tell the difference among these wines, the average rating point drops. The lowest average rating takes place in 2008. After that, it increases back to around 89.5 again.

Trying to find if there is some information hiding in finer-grain time span, the relationship between month and average rating should be revealed. Figure 5 shows the trends of average rating along with months in eight years from 2003 to 2010.

In these eight years, average rating seems to have no obvious connections with month, Figure 6 shows this result more clearly. So month will not be considered to be an important in my predicting model.

E. Expertise

As users review more wines, their knowledge about wines are likely to accumulate and they will become more *expert*[2].

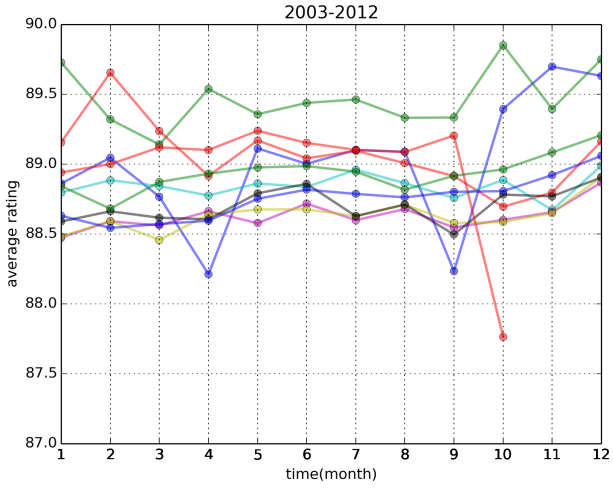

Fig. 6. month and average rating overall

Their experience level seems to be another good factor which strongly influence their preference. Here I assume that users' *experience* is closely related to the number of wines they have reviewed. After some quick calculation, the average rating times for each user is 45.8. So let 45 as a classifier for *expert* and *novice*. Then calculate the difference of their preference to find whether there are some patterns.

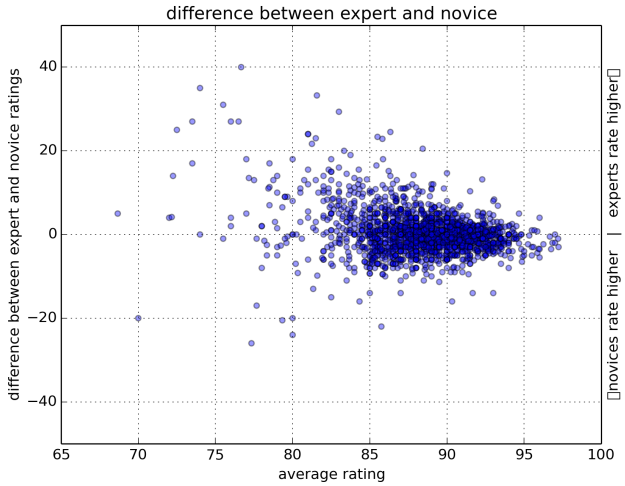

Fig. 7. expert and novice rating

Figure 7 is depicted based on the reviews in the year 2012 since it is the latest data and we will be less influenced by the problem such as *expert* made a lot of mis-rating during his *novice* time. Figure 7 shows the result of my idea. In this figure, users who has reviewed more than 98 wines are treated as experts while users who has less than 3 reviews are treated as novice. The x-axis represents the average rating for certain wines, and the y-axis represents the difference between the average rating from expert group and average rating from novice group. It turns out that expert and novice

can reach an agreement for very good wines, but for ordinary wines, they have great different opinions. Besides, experts can are more mild to ordinary wines while novices seems to be relative mean to give high score to common reds.

III. Prediction Task

After these exploration, I had made my prediction task clear: provided with the user's id and wine id, predict whether user would like this wine or not. Here several points should be defined in advance: a successful recommendation will recommend a wine which this user would give more than his average rating point for other wines in the past. So it will be a miss if we have recommend a wine which user is likely to give low rating point. The model should be like this:

$$rec(user, wine) = rating_{predict}$$

By doing that, we can get a rating point $rec_{(user,wine)}$ for user on such wine. To evaluate the performance, the mean square error (MSE) needs to be calculated on our test dataset T:

$$MSE(T) = \frac{1}{|T|} \sum_{r_{(user,wine)} \in T} (rec_{(user,wine)} - r_{(user,wine)})^2$$

Test dataset is all the reviews made in year 2011 and 2012, training data is all the reviews made between year 2003 and 2011.

The baseline for this task is to calculate the average rating among all the history reviews and use it as predicted rating for this user. This naive approach just ignore the identity or any other features of user, just use the favor of the whole community to judge whether individual would love to praise certain wines. The MSE of baseline approach is 18.953.

IV. Related Literature

The dataset I am using comes from SNAP[1] and has been used for the paper[2]. That paper modified latent factor model by introducing user experience as a function of time. It based on the assumption that "by individually learning for each user the rate at which their experience progresses, we are about to account for both types of behavior". Basically, the model has been changed to:

$$\begin{aligned} rec(u,i) &= rec_{e_{u,i}}(u,i) \\ &= \alpha(e_{u,i}) + \beta_u(e_{u,i}) + \beta_i(e_{u,i}) \\ &+ < \gamma_u(e_{u,i}), \gamma_i(e_{u,i}) > \end{aligned}$$

And the result of this model is pretty good: it shows in the paper that in both users' most recent reviews and randomly sampled reviews, its MSE are all reduced greatly. Besides, the paper also shows some interesting behavior for users from various level of expertise. It reveals that beginners and intermediate users are hard to predict their behavior while experts tend to agree with each other and easier to predict. But "almost experts" are surprisingly least predictable. And its explanation is that users do not become experts smoothly, but evolute stage by stage. Here, its performance is treated as a goal for my model to achieve, even though it is extremely hard to beat their performance, it still let me know how much we could do to improve my model.

V. Relevant Features for Prediction Task

The features I am going to use to predict user rating on certain wine is *wine's production year* and *user's experience*. The first feature, *wine production year* is easy to see why it is relevant. According to the exploration part of section II, wine's production year is strongly influenced its average rating (Figure 1). More specifically speaking, the older its year, the higher people's rating on it. So there exist some negative correlations between production year and its reputation. Besides, the older its year, the less reviews about it. This may result from that the older the year, the higher its price would be. It verified that negative correlations in another way.

The second feature I am going to use is *experience*. It is a feature closely related to the review history before the time of prediction task. There are two pieces of data store the information of experience: the number of reviews made by the user in the history and the total period of time since this user joined community. As shown in Figure 7, people who make a lot of reviews and people who review little have different behaviors over ordinary wines, but they tend to agree that some wines are really "the best". And it is reasonable that user who stay in the community are more experienced on wine reviewing, and that can be approximately revealed by the time span between their first and last reviewing time. It is very hard for a small project like this to model experience better than the paper[2] I mention before, but I will try two models to reveal that these two pieces of data can make some difference.

VI. Model Description

There are several models I have tried in this project. The first model is a very intuitive linear model:

$$rec(u,i) = w_0 + w_1 \times WineYear(i) + w_2 \times NumOfReview(u)$$

Here u is user variable (user ID) and i means wine variable (wine ID). Here I did not consider the difference of each user's learning rate. So all users are treated with the same learning rate, which means after writing certain number of reviews, a user will reach to the expertise level with others who also finish this workload.

There are times when a new user or a new wine come to the model when testing. In that case, we can not say anything about it, so if only user is new, this model will provide the average rating of this wine as predication, otherwise, this model will give the average rating of all wines as predication. So the model will become:

$ar(i) =$ average rating of wine i

$AR =$ average rating of all wines

$$rec(u,i) = \begin{cases} ar(i) & if u \notin History \\ AR & if u \text{ and } i \notin History \\ w_0 + w_1 WineYear(i) + & \\ w_2 NumOfReview(u) & otherwise \end{cases}$$

The testing result is $MSE = 16.6634$. It improves baseline by 11.4%. But it's still large. The problem might lie in the ignorance of the time span of users first and last review. If a user did spend some time in the community while not reviewing too much wines (maybe lazy), he should also gain some experience.

My second model aims to make some improvement. This time, it will take the time span between user's first review and last review. So the predicting model is:

$$\begin{aligned} ar(i) &= \text{average rating of wine i} \\ AR &= \text{average rating of all wines} \\ t &= \text{last day of review} - firstdayofreview \end{aligned}$$

$$rec(u,i) = \begin{cases} ar(i) & ifu \notin History \\ AR & ifu \text{ and } i \notin History \\ w_0 + w_1 \times WineYear(i) + & \\ w_2 \times t \times NumOfReview(u) & otherwise \end{cases}$$

This model improves the MSE to 16.6630. It's not a big improvement though but it shows some light that both time span and number of reivew is necessary to be considered.

My third model is using *k Nearest Neighbours*. Each point is a vector of three values: wine's production year, number of reviews, time span. By doing that, this model will treat all the training data as a model. Every test data is going to find some history data points which are really similar to it, then it can take the median of all those points to be its predicting data. Here I also split a validation set from training data to tune k. The MSE of it is around 2000.

I also tried using Latent Factor Model. Even though it ignores *temporal* information, it would also get excellent performance in practice[3]. The predicting function is:

$$rec(u,i) = \alpha + \beta_u + \beta_i$$

Here u refers to user and i refers to wine. So the problem will become try to solve:

$$arg\min_{\alpha,\beta} \sum_{u,i} (\alpha + \beta_u + \beta_i - R_{u,i})^2 + \lambda [\sum_u \beta_u^2 + \sum_i \beta_i^2]$$

And λ is set to 1.0. Alternate updating method is used to update α and β, until converge. The MSE of this model beats all the former models with 13.810, nearly 18 percent improvement.

VII. Result and Conclusion

The result of all three models as well as baseline is show below:

	baseline	linear model 1	linear model 2	k-nn	Latent factor
MSE	18.953	16.6634	16.6630	2420.2264	13.810

The result is not very encouraging. Especially k-NN model which I expected it to perfor well. In this section, I will analysis each model in detailed way.

First model is trying to build only one straight line for all the data points, which is nearly impossible to do the work well. Since this dataset seems hard to be linear split only based on those two values. It may not be linear at all. Just as Figure 7, though there is some relationship between expertise and rating points, but it is not linear.

Second model has the same problem, since its linear, too. But k-NN model's failure bring me to think that training data may include too much historical mis-rating of users and could not represent the current appetite of certain user. So I made a small modification to this model by shrinking training dataset to contain only reviews from year 2011 and use it directly to predict data points in year 2012. The MSE drops sharply down to 984.32. So I might assume if there is a really state-of-art dataset to feed this model, its performance would likely to be much better.

Latent model should be a good choice, and it turned out to be. It improves the performance very much while MSE is still large. I might guess the problem lies inside the training set and testing set. My testing set is all data of the year 2012, while training data is before that. So it has relative large bias on temporal variable. Maybe random sampling will be a better choice.

In conclusion, this dataset is very special since it lacks ordinary wine features as well as user features. All it has is the relation between users and wines, as well as time stamp. So all the three models I built can hardly meet requirements of real recommendation system. But these trial and failure provides good experience to other people and help them reveal the good way to dealing with a dataset like this and avoid pitfalls.

Acknowledgment

The author would like to thank Mark Qiao, who is also struggling through some messy datasets while helping me with some understanding of concepts for this project.

References

[1] Jure Leskovec and Andrej Krevl, *SNAP Datasets: Stanford Large Network Dataset Collection*. http://snap.stanford.edu/data, Jun, 2014

[2] J. McAuley and J. Leskovec, *From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews*. WWW, 2013.

[3] Y.Koren, R.Bell, and C. Volinsky, *Matrix factorization techniques for Recommender systems.* Computer, 2009