# Using weather data for wine recommendations
# Final Report
(CS221 Final Project)

Paul Shved
pshved@stanford.edu

December 14, 2018

## 1 Summary

In this project, we've built an unsupervised learning system that makes wine recommendations. The system first analyses tasting notes from the Wine Spectator dataset of expert reviews, and describes each wine as a feature vector of small length ($< 30$). Based on the extracted features, the system answers queries of type "for a given wine $W$, sort all entries in the dataset by how closely the entry's taste resembles $W$.

In addition to building this system, we compare if augmenting the dataset with the weather-related features (on top of tasting notes features) improves the recommendation.

Based on observing the metrics and expert evaluation, we conclude that the recommendation quality *reduced* after adding weather features.

## 2 Task Definition

### 2.1 Motivation and Scope

Wine drinkers appreciate the variety of tastes, food pairings, and the history of the drink and of the winemaker as well as the soothing effect of alcohol. Typically, climate and tradition influence the end product as well as the winemaker's choice of grapes to mix and of the aging technique. Winemakers in the old, developed regions such as Bordeaux, Burgundy, or Rioja, consistently produce wines of a certain, expected taste profile. Due to cross-border shipping constraints, finding the local, more available wines that match the taste profile of a known but less available foreign style lets wine drinkers discover and enjoy hidden gems while saving money.

In this project we build a wine recommendation system; its intended use is to pick one "target" wine from an "established" region, sort the dataset by the "proximity" of every wine to the "target", and apply filters (such as

"2010 vintage or above, origin "California"). The features are based on the text analysis of the review notes and weather data (see section 3). See section 4 for the definition of "proximity" and **??** for extraction of features into text.

### 2.2 Evaluation

We run the same query twice: using "weather" as the measure of proximity and assumeing that the weather everywhere is the same. Based on the results (section **??**), we conclude that weather data is not helping.

## 3 Data Preparation

We used the following datasets:

- **Wine Reviews**: a dataset of Wine Spectator expert reviews ($270,000$ reviews, $20,000$ wineries) from 1990 through 2016 that we scraped for this project.

  We tried to use the Kaggle dataset [1], but it does not have the vintage (year) data which makes it impossible to understand what weather we need to associate with the wine.

- **Weather**: GHCN-M monthly average world temperature data from NCEI ($471,000$ station-years) [2] Many stations did not have complete weather data for all years 1990-2016, so we only selected those with all years present.

- **Google Maps API**: 580 API calls for $2.90

The weather dataset contains Geographical locations of the stations that make the observations. However, the wine review dataset did not contain the geographical locatinos. For each winery, we converted the country of origin and region into a string, and then used Google Maps API to convert it into latitude and longitude. The (lat,long) pair was used to select the closest weather station for each wine, and that stations features.

## 4 Problem definition

In effect, we're going to build a "distance" metric between wines. Let's define a wine as

$$W = \{R, T\} \tag{4.1}$$

where $R$ is "text review features" and $T$ is "annual temperature features".

We define two "distance metrics": one that takes weather into account and one that does not:

$$D_1(W_1, W_2) = \begin{cases} ||R_1 - R_2||_1, & \text{if } T_1 = T_2 \\ +\infty, & \text{if } T_1 \neq T_2 \end{cases} \tag{4.2}$$

$$D_2(W_1, W_2) = ||R_1 - R_2||_1 \tag{4.3}$$

Where $T$ is a clustering category defined in
When we want to find a wine similar to $w_0$, we search the whole dataset for

$$w = \text{argmin}_{w \in W} D_i(w, w_0) \tag{4.4}$$

In this project we will compare the sets of $w$ for both $D_1$ and $D_2$.

# 5 Weather feature extraction

In order to extract feature vectors, we run a clustering algorithm on average temperature vectors. For every wine, there exists one vector with 12 average temperatures for the year. After normalization, and converting to [-1; +1] based on average weather for this location, we run a clustering algorithm evaluated in the next section.

# 6 Comparison of Clustering Algorithms

Since we're performing unsupervised learning without ground truth available, we have limited choice of quality assessment. We chose two ways to measure the quality:

- Silhouette score [3]
- Size of the 10-th cluster divided by total dataset size. The motivation for this metric is that we aim to get at least 10 "big" clusters so that every weather feature "cuts off" at most 90% of the dataset from considerations for proximity.

Before we start a deeper look into the evaluation of the other hyperparameters, we should evaluate a sensible dataset size to perform the experiments on. The entire dataset has $270,000$ entities, and no clustering algorithms available to us were able to complete on this dataset on our home computer. See figure 1 for the results.

Based on figure 1, we will perform other evaluation with the size of the dataset of 10000.

Next, let's evaluate the Clustering algorithm by choosing some cluster sizes. We use Hierarchical Clustering and MiniBatchKMeans, and evaluate the metrics presented above. The results on figure 2 demonstrate that while K-Means delivers smaller Silhouette score values, the clusters are more even when Hierarchical clustering is used.

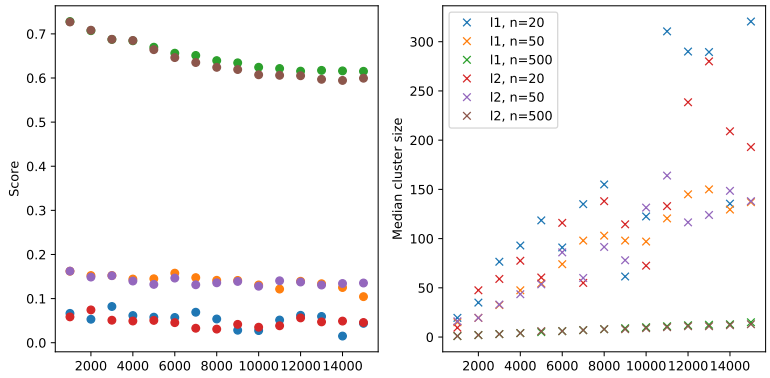


Figure 1: Evaluation of a representative dataset size for different clustering based on the quality of the clustering as measured by Silhouette score.

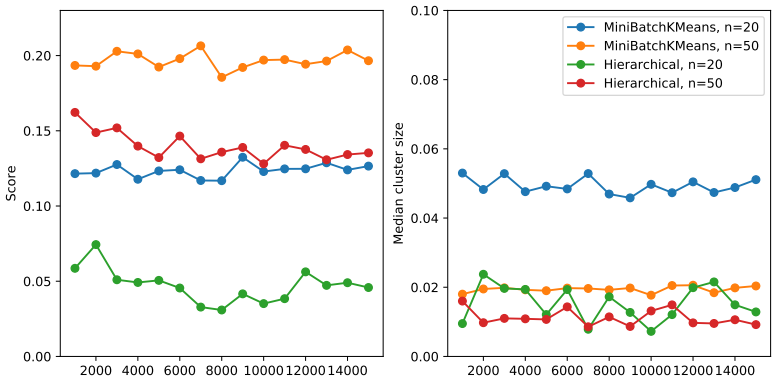


Figure 2: Evaluation of MiniBatchKMeans clustering algorithm vs Hierarchical clustering algorithm for weather feature vectors $T$.

We will use Hierarchical Clustering for further evaluations in this report.

Talk about how Hierarchical clustering was the only thing that worked and measure some timings.

Now let's choose the distance metric for the features, and the number of clusters. We will use the same objective metrics. The results are on 3. We see that the highest size of the 10-th cluster is attained at 50 clusters when the distance metric is $L1$. This motivates the choice of $L1$ as metric in (4.2) and (4.3).

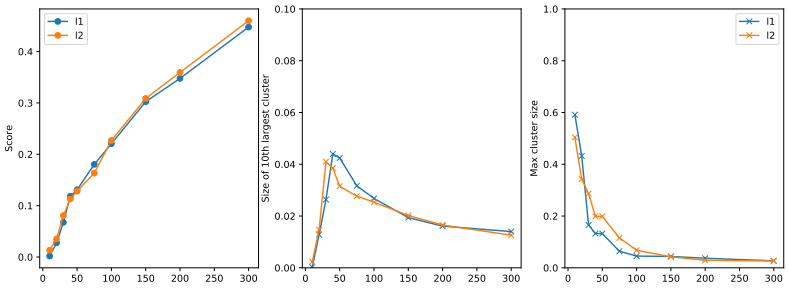


Figure 3: Evaluation of L1 and L2 metrics and of number of clusters for the Hierarchical Clustering algorithm for weather features $T$.

### 6.1 Chosen clustering algorithm

Thus we perform the following clustering on weather features:

- Perform Hierarchical clustering with $L1$ metric and 50 clusters on weather vectors of 10000 randomly selected weather vectors.
- Find centroids of these clusters.
- Define the weather feature $T$ as the cluster number of the closest centroid out of the previously determined ones.

## 7 Winery distance heuristic

We noticed that for wineries with fewer than 5 reviews the original formula boosted them dramatically. We increased the coefficient to 5.

## 8 Text feature extraction

In order to compute $||R_1 - R_2||$ for wine review features, we convert wine reviews to a feature vector. word2vec embeddings are popular with similar projects like [4], however in this project we deliberately performed our own feature extraction.

In this section we compare two feature extraction algorithms: TF-IDF and Latent Dirichlet Allocation (LDA).

### 8.1 Inapplicability of TF-IDF to Wine Reviews

Initially we tried to use TF-IDF to extract features, using unigrams and bigrams. This produced 710409 features. Then we clustered the vectors into 40 clusters.

Clustering during our experiments did not meet our expectations. We assessed this by building a distribution of wines by style ("wine style" is a known

label) in the entire dataset and compared it to the distribution of some clusters. Here's a table that shows a percentage of Chardonnay (white wine) and Cabernet Sauvignon (red wine typically different from Chardonnay) in three clusters:

| Wine Style | Cluster 1 | Cluster 4 | Cluster 5 |
|---|---|---|---|
| Chardonnay | 0.11 | 0.10 | 0.09 |
| Cabernet Sauvignon | 0.09 | 0.08 | 0.08 |

Dimensionality reduction did not help either (using Truncated SVD to 100 dimensions from 710409). Using unigrams only (no bigrams) produced 28269 features, and this didn't improve the results either.

After examining the results more carefully, we concluded TF-IDF was the wrong algorithm to use on wine reviews specifically. In the wine reviews, terms that provide most information about a certain wine characteristic ("crisp", "apple", "fruit", from an example with index 1 had weights of $< 0.06$) *are also the most frequently used across the whole corpus.* TF-IDF grouped them with the same words as ("flavors", "shows", and "touch"). The terms that do not provide this information have the highest TF*IDF and represent the writer style more than the properties of the wine ("rind" had 0.219, "coalesce" 0.184, "eden" 0.182).

### 8.2 Latent Dirichlet Allocation

Let's use a different vectorizer, LDA (Latent Dirichlet Allocation). LDA produces $K$ numbers $\theta_k$ for each wine review that represent how well this review matches Topic $k$. In some sense, LDA already performs some sort of clustering by learning these hidden topics.

We used Term Frequency (without the Inverse Document Frequency) to produce uni- and bigram counts, then we used a custom, wine-specific blocklist, and then used LDA to vectorize features into 20 topics.

Top terms for these topics matched the author's understanding of wine taste. For example, topic #18 described characteristics typical for wines from Southern France, while topic #19 accurately captured traits of Cabernet Sauvignon.

```
Topic #18: plum, fruit, blackberry, currant, core, black, fig, tobacco, ...
Topic #19: acidity, lively, crisp, fresh, tart, lemon, clean, lime, bright, ..
```

When we ran the clustering algorithm defined in section 6.1, the results produced varying clusters, not subject to the same issues as described in section 8.1.

## 9 Result Analysis

First result is that the clustering algorithm on weather features produced results that matched the oracle evaluation. We chose one wine we were familiar with (index 111076) as a sample and evaluated the similarity. Top 46 similar wines were from the same region (from various years), which matches the expectation

of how a good clustering algorithm would perform. This result held with and withotu weather features.

However, after we added weather features and looked at different regions, the prediction dramatically reduced, as demonstrated in the following table:

| number | wine 1 | wine 2 |
|---|---|---|
| use weather | 92 | 43 |
| ignore weather | 7 | 14 |

In order to see if the reduced matching resulted in improved predictions (what if it actually filtered out wines that did not really match?) we performed a blind wine tasting:

- example wine (index 111071)
- best matching wine in California, using $D_1$ and thus considering weather features (51216)
- best matching wine in California using $D_2$, without considering weather features (48163)
- wine chosen by the expert at the reputable wine store.

The blind tasting revealed that the most matching wine was 48163, the close second 51216, and the worst performance demonstrated by the wine chosen by the store expert.

Another interesting observation was poor peformance on white wine similarities. However, we realized that there is just less variety in the taste profiles of white wines. Our algorithm was able to distinguish between two main taste profiles of white wines (Chardonnay vs Sauvignon Blanc) but not within them.

Thus, we conclude that wine similarity algorithms based on unsupervised text classification do provide useful recommendations for red wines, but adding weather features reduces the quality of the recommendation.

# References


[1] Z. Thoutt, "Wine enthusiast review dataset," *Kaggle*, 2017. Published at `https://www.kaggle.com/zynicide/wine-reviews`.

[2] J. H. Lawrimore, M. J. Menne, B. E. Gleason, C. N. Williams, D. B. Wuertz, R. S. Vose, and J. Rennie, "An overview of the global historical climatology network monthly mean temperature data set, version 3," 2016. Published at `ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/v3/`.

[3] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53 – 65, 1987.

[4] A. Effron, A. Ferris, and D. Tagliamonti, "Learning the language of wine," *CS229 final report*, 2017. Published at `http://cs229.stanford.edu/proj2017/final-reports/5244216.pdf`.