Identifying high-quality cosmetics and similar alternatives from text

Cassandra L. Jacobs
CS 410 Final Project
cljacob2@illinois.edu

Abstract

This project looks at how well text reviews provided by an expert can be used to predict the quality of a cosmetics product as well as whether the product is similar to others. In this paper I compare a number of different classifiers to each other on a small dataset of 110 cosmetics products (22 categories with 5 exemplars each). Ultimately, identifying product quality is an easy task achieving 80% accuracy, but identifying what products look similar based on textual similarity is less trivial. Instead, products that look similar are not necessarily described using the same words. Word features tended to be more robust to noise than more complex features. A larger dataset might reduce the shortcomings of some of the algorithms used here because of richer semantic representations that can capture the relationship between words, which is at least one contributing factor to the poor performance of the similarity classifiers.

1 Introduction

Most consumers today are concerned with getting a number of different things out of their purchases. One of these things may be quality, and another may be a price-to-quality ratio. A number of different products on the market are reviewed extensively for their quality, which consumers can use to guide their decision making process when making a new purchase. On Amazon, for example, a large number of 5-star reviews may be enough to convince a person to go for a slightly more expensive product. Many products are generic household items, foods, books, or movies. Formal reviews of more personal products like hair dye, makeup, or other beauty products are distributed across many different websites, so a number of people end up following just one or two bloggers or "gurus" for up-to-date reviews.

There are two general problems for text retrieval and knowledge acquisition in the beauty industry. First of all, there is a large number of very similar products that have similar application, coloration, or texture that image recognition software alone would not necessarily be able to discern. Consumers of cosmetics rely heavily on text reviews in conjunction with visual information to determine whether they will purchase a product. Expert reviewers are therefore a highly valuable source of information, especially when literally tens of thousands of products exist with very similar functions.

Being able to leverage the text of reviews for cosmetics products would be incredibly useful. First of all, text descriptions for products that are very similar to another product that is similar in quality but significantly less expensive may be good to recommend. In addition to this, the text of the reviews might be an indicator of the quality of the product, as is common in sentiment analysis. In this project I hope to demonstrate that similar products are described using similar language, and that high quality products tend to have different properties and descriptors than lower quality products by doing an analysis of text reviews from the cosmetics review website Temptalia.com.

2 Dataset

While there are a number of different sources of cosmetics reviews, such as retailer websites like Sephora, Ulta, or Amazon, these resources lack relational structure between the reviewed product and other products. Similarity scores are useful because highly rated products are often expensive. Many recommender systems on cosmetic retailer websites will recommend different colors of the same line but not similar colors from different brands. Alternatively, they will recommend products of a completely different type that tended to be purchased together, such as a blush and a lipstick, which are not two similar products.

The reviews for products are typically written by novices and those products with especially small numbers of reviews may not be reliably scored as a good or bad product. This is especially true because many people who write reviews either strongly recommend or strongly advise against the product that they are reviewing. As an alternative to using novice reviews from cosmetic retailers' websites to understand whether we can identify similar products and product quality based on the text alone, reviews for these experiments come from Temptalia.com. This is a review website where a single person assigns grades to products and identifies products with similar properties, while also assigning a similarity score to these products. Temptalia.com rates different products for their quality on a numeric scale (F to A+) in addition to identifying similar products to the reviewed product based on properties like finish, shade, or variations in color. Importantly, an expert assigns all scores, so we can evaluate the model's performance relative to some ground truth.

I aggregated reviews of 22 products by hand where each product had a minimum of four similar products identified by the expert rater. Reviews were selected such that each of the other four reviews represented a different brand, rather than multiple similar products from the same brand. These brands also could not be the same as the brand of the primary product. I also extracted the reviews of all 88 other similar products, leading to a total of 110 product reviews. Products ranged from very low scores to very high scores. There were 54 of the scores an A- or higher, and 56 had a score of B+ or lower. Grades were converted to numeric scores. Below in Figure 1 is a histogram of the distributions of scores assigned to each product. If the text review contained information about similar products, this information was removed because two similar products are likely to be similar to the same products, making one versus one similarity classification a trivial task. If information was available about the whole product line, that information was included in the review. Reviews ranged in length from 29 words to 511 words. The average review was 208 words long with standard deviation of 91 words.

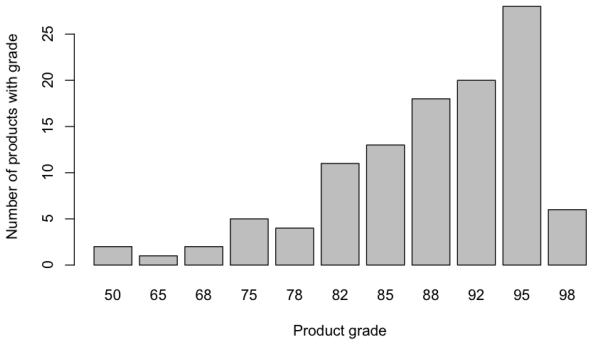

Figure 1: Distribution of product scores in dataset from F (50) to A+ (98)

3 Experiment 1: Product Quality Classification

3.1 Features

For the purposes of the experiments below, text features were extracted from the reviews. Because many reviews contain idiomatic expressions that are indicative of quality (e.g. "satin sheen", "color payoff"), the first set of classifiers used below contain word, bigram, and trigram features extracted using the CountVectorizer function of the Python package Scikit-learn removing all stop words. All words were transformed into count vector representations; features from the test set were transformed into vector representations from the training set. The second set of classifiers use TF-IDF weighted features (TfidfVectorizer) rather than raw counts. No features were excluded from the model.

Classifier	Precision	Recall	F1
Naive Bayes	0.58	0.58	0.58
SVM	0.24	0.49	0.32
Logistic regression	0.66	0.65	0.65
Perceptron	0.69	0.69	0.69

Table 1: Classification accuracy (good versus less good products) by algorithm using count vectors of unigram, bigram, and trigram features.

Classifier	Precision	Recall	F1
Naive Bayes	0.62	0.62	0.62
SVM	0.24	0.49	0.32
Logistic regression	0.66	0.64	0.65
Perceptron	0.64	0.64	0.64

Table 2: Classification accuracy (good versus less good products) by algorithm using tf-idf weighted vectors of unigram, bigram, and trigram features.

3.2 Classification

Here we test four binary quality classifiers: Naive Bayes, support vector machine (SVM), logistic regression, and perceptron across four variants of the classification task with different proportions of train and test data as well as different feature sets.

For the classification task there are two outcomes: grades of 1) an A- or higher or 2) a B+ or lower. For half of the experiments, a random subset of 55 of the reviews were used for training, with the other 55 serving as the test set. There were 28 positive examples and 27 negative examples. In the other half of the experiments, items were split into a 100-10 train-test split. 5 test items were high-quality products and 5 were not.

3.3 Results

In general, the Perceptron model tended to outperform every other classifier on this task. By contrast, the SVM model performed substantially worse on this task than any other task. These results are summarized in Table 3.3 A closer analysis of the errors on this task reveals that the SVM classifier never assigned a positive (A- or higher) label to any of the test items, and in fact seems to have learned the same decision boundary in both the count and tf-idf case. I thought that the SVM's performance could improve with more examples, so I tested the models' performance on a 100-10 train-test split using tf-idf weighted features, whose results are in Table 3.3. It is obvious that SVM performs poorly independent of the specific features. In fact, SVM performs even more poorly with more data, as seen in the next two experiments. As can be seen in Table 3.3, it appears that with more training examples, SVM performs even worse than with fewer training examples, which suggests that some features are introducing noise into the model. To test this possibility, this next set of models contains only unigram features with tf-idf weighting. After this change, SVM is performing on par with logistic regression but not as well as perceptron, whose performance is unchanged from the full feature set. The results of this analysis may be found in Table 3.3.

Classifier	Precision	Recall	F1
Naive Bayes	0.68	0.60	0.62
SVM	0.09	0.30	0.14
Logistic regression	0.68	0.60	0.62
Perceptron	0.80	0.80	0.80

Table 3: Classification accuracy of good versus less good products using tf-idf weighted features on a 100-10 train-test split. Performance here is generally better than the previous two experiments with more examples for every model except the SVM.

3.4 Discussion

While the results show that none of the models exactly learns the concept of a high scoring review, they do show some promise. Tf-idf weighted perceptron in particular seems to identify informative features with relatively few examples, although it outperforms every single model regardless of the feature set. Perceptron is robust to non-linear (bigram and trigram) features, which are useful in quality classification. With more data, it is likely that most of the models would improve in the classification task, but we can tentatively say that high-quality products are distinguishable from low-quality products on the basis of the text reviews on Temptalia.com alone.

4 Experiment 2: Product Similarity Classification

In this task we are concerned with being able to identify products that are similar to each other as a

Classifier	Precision	Recall	F1
Naive Bayes	0.65	0.60	0.60
SVM	0.70	0.70	0.69
Logistic regression	0.72	0.70	0.70
Perceptron	0.80	0.80	0.80

Table 4: Classification accuracy of good versus less good products using tf-idf weight unigram-only features on a 100-10 train-test split. Performance for most models improves when fewer features are used, but Perceptron performance without bigram and trigram features is equivalent to unigram performance.

Classifier	Precision	Recall	F1
Naive Bayes	0.22	0.36	0.26
Logistic regression	0.26	0.41	0.30
Perceptron (1 vs. 1)	0.27	0.41	0.31
Perceptron (1 vs. all)	0.27	0.41	0.30
K nearest neighbors	0.12	0.18	0.13

Table 5: Product classification using unigram tf-idf weighted features.

way of enhancing product recommendations. If two products are described using similar words, then it should be easy to recommend another product in the place of one that is currently being considered. For example, if both products are described in similar ways, then two similar products should be recommended. For example, in reviews of three similar products, each of them is described as "muted" with "warm undertones."

There are some limitations to using a small dataset such as this one because there are many different ways of saying the same thing. It would be ideal to capture the similarity in color between the products, but one product may be described as "rosy terracotta", another as "peachy-brown" and another as "brown with rosy undertones." While it is obvious that these are all related descriptions, a larger, potentially distributed semantic space would be ideal for classification. This is one limitation of a dataset of this size.

4.1 Classification

As in the quality classification task, we are interested in identifying products that are similar to others. The models used are a multinomial Naive Bayes classifier, logistic regression, two variants of perceptron, and the k nearest neighbors algorithm. Each of these models is attempting to predict which product category (e.g. a set of pink blushes) that the test item is most similar to.

To do this, we reserve one each of the 22 products to be classified out of the five total. So, training for each category takes 4 exemplars of that category and tests on the final fifth. In the case of the one versus all classifiers, there are 22 classifiers for each product category to predict the main product of a category from the four other similar products with 4 positive examples and 84 negative examples. One classifier is a one versus one perceptron, indicating 441 classifiers containing four examples for each of the two classes.

As with the prior experiment, unigram features were generated using the TfidfVectorizer function while removing stop words. The models for this task seemed especially sensitive to the number of features, so we performed a chi-square test on the training set to identify the top 150 features used during training. We used the top 150 features in order to avoid a well-known bias-variance tradeoff, in which models with greater numbers of features can greatly overfit the data with little to no classification benefit. The test set was transformed to fit into this feature space.

4.2 Results

Most of the classifiers do poorly at the similarity classification task. While the models can occasionally identify the products that are similar to others (precision and recall equal to 1), many products are never said to be similar to anything (precision and recall both equal to 0), which makes the model evaluation difficult to interpret. This is likely due to a sparsity of the features as well as the lack of sufficient data to learn what makes similar products similar.

Remarkably, k nearest neighbors performs dramatically worse than all of the other models, suggesting that a product's similarity to other products is not best approximated by linguistic distance. In light of this, it is important to understand whether similar products actually occupy similar linguistic spaces. To better determine whether similar products are indeed similar to each other, we analyze k-

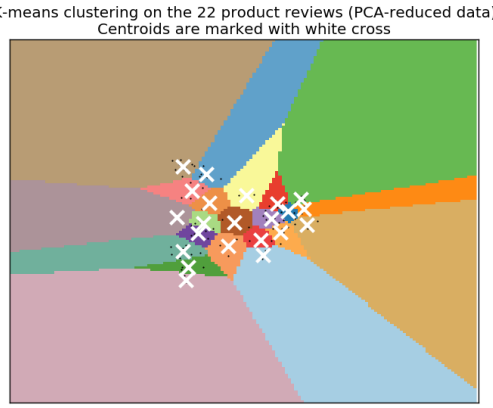

Figure 2: Visualization of location of 22 k-means cluster centroids along two principle components. Cluster centroids tend to be very close to each other and it is not obvious that these centroids represent the intended 22 product categories. Instead, they may represent clusters of features, like texture, sheen, or price.

means clusters instead of classification accuracy in the next section.

4.3 Clustering with k-means

We construct a count vector of the vocabulary terms for each product without pruning features. We randomly initialized 22 clusters in this feature space and allowed k-means to organize the 110 products into these 22 clusters over 1000 iterations. To analyze accuracy of the k-means clustering, we looked both at cluster homogeneity and cluster completeness. Homogeneity is defined as the proportion of items in a cluster that come from a single class (the majority class). Completeness is a different sort of measure that assesses how much items from the same class tend to be in the same cluster. Both measures exist on a scale from 0 to 1.

The k-means completeness and homogeneity scores demonstrate one of the reasons why classification accuracy is so poor. Homogeneity scores were approximately 0.365, suggesting that many clusters contained items from a number of different categories, perhaps based on the type of product. Completeness scores, by contrast, were at 0.54, suggesting that any two similar products were almost as likely to be clustered together as not.

To better visualize the degree to which the clusters that k-means learned are similar or highly overlapping, we collapsed the features into two dimensions using singular value decomposition followed by principle components analysis and transformed those features for each review to fit into these two new dimensions. We then plotted the 22 cluster centroids on this space. As can be seen in Figure 2, the clusters are generally very close together, with potentially two major clusters being present on the left and right sides. Analysis of these clusters shows that lipsticks, blushes, and eyeshadows tend to cluster together, in addition to certain types of features like finish, opacity, or how flattering the product was. Distinguishing features from three of the 22 clusters may be found below in Table 4.3.

Cluster	Distinguishing features
19	color, lipstick, shades, finish, drying
20	matte, metallic, glitter, micro
21	application, dry, damp, pink

Table 6: Clusters identified by the k-means algorithm with 22 centroids intended to cluster 110 products with 5 exemplars per category. Generally, clusters identify similar categories of features rather than categories of items. For example, Cluster 21 describes ways to enhance the color of eyeshadows when applied, while Cluster 19 describes different features of liquid lipsticks.

4.4 Discussion

The results of the similarity analysis show that not all linguistic features can be used to identify similar-looking products. The features that are distinguishing as determined by a cluster analysis tend to have to do with the type of product, rather than specifics about any particular product. As a result, products that are identified as similar on Temptalia.com are not necessarily categorized together.

5 General Discussion

The results of the two experiments determined that given enough data and not too many features, it is possible to classify as many as 80% of test items accurately. When trained on a smaller dataset (50% at train and 50% at test) it was still possible to get nearly 70% accuracy in identifying high-quality products over low-quality products. To some extent classification accuracy depended on the classifier,

the type of features (raw counts or tf-idf weighted vectors), and the number of features (unigrams versus unigrams, bigrams, and trigrams).

Identifying whether a product was similar based on text alone, however, was not an easy task. The best classifier trained to identify the fifth item from a set of five similar products still only achieved an F1 score of 0.31. This suggests that the distribution of text features alone in each product are not sufficient to predict product similarity. Instead, natural clusters emerged with types of products, but not products that resemble each other.

In further work, it will be important to create a larger dataset and obtain more distributed representations of word meanings. In order to identify similar words, it may be usedful to mine different sources of reviews such as those of retailers where consumers are likely to reference attributes of the products they purchase. It may be possible to generate dense representations of these words and pairs of words, which would capture the distributional similarity of phrases like ”mustard” to ”yellow”. Additionally, it may be worthwhile to build or use a glossary, in which synonyms and definitions are provided for terms specific to cosmetics, leading to better features for classification.

An additional avenue of research would be to combine text reviews with the image data available for each product. Because there are photographs of every product, decomposing the images into color feature distributions might function an additional predictor of whether a product falls into a particular cluster or not, which would augment text data for classification purposes.

Altogether, this work has potential to be useful for retailers. It is easy to identify similar products, but more data and better features are needed to cluster products into similar-looking categories. As it stands, reviews of obviously similar products are not similar enough by themselves to reliably identify which ones are similar and further development will likely improve performance on this task.