A Naive Bayes Approach to Amazon Sentiment Classification

By Rene Bidart and Kevin Pei

Abstract

The Naive Bayes approach has traditionally been a fast and effective method to classify and categorize corpuses within the text-mining community. In this research paper, we apply the same approach in classifying Amazon product reviews. We show that through a simple approach, such as Naive Bayes, it is possible to classify consumer sentiment with high accuracy on the training set and moderate accuracy on test set. Lastly, we find evidence that single product classification is much more accurate than within-category classification due to more tightly clustered and relevant words. We also find that the Naive Bayes classifier tends to perform better in out-of-sample testing when trained from large sample sizes.

Table of Contents

---

1. Introduction

Naïve Bayes was one of the first classification methods used for text mining and is often used as an initial approach because it is relatively easy to implement and is not computationally expensive [3]. The main benefits of this model are its simplicity and thus, its ability to be trained quickly when there is a large data set. It is also much more interpretable that other machine learning and classification methods such as neural networks and SVMs. The objective of this research is to examine how Naive Bayes can assist in classifying consumer sentiment in Amazon reviews. We want to ideally examine the relevant words that can separate between a negative review and positive review based on user ratings. Furthermore, we want to examine the degree of predictability for within-category and single-product reviews. We hypothesize that the classifier should have a much higher accuracy in classifying sentiment for single products since the universe of words will be more tightly clustered around a certain set.

The rest of this paper is organized as follows: first, we will first introduce the Naive Bayes model along with some of its properties. In section 3, data sample and methodology will be outlined. Results and statistics are presented in section 4 and lastly, a brief conclusion in section 5.

2. Model

The Naive Bayes classifier is a simple probabilistic predictor that is both efficient and scalable in its approach. Given a $K \times 1$ vector of discrete classes, it calculates the probability of a given observation of features X belonging to the class C_k through applying Bayes theorem:

$$P(C_k \mid x_1, x_2 \ldots, x_n) = P(C_k \mid X) = \frac{P(C_k)P(X \mid C_k)}{P(X)}$$

Where $P(C_k)$ is commonly referred to as the prior probability often calculated by the percentage of times C_k appears in the entire training set. Furthermore, $P(X \mid C_k)$ is commonly referred to as the likelihood function where it is directly proportional to the probability of all features given their class or

$$P(X \mid C_k) \propto \prod_{i=1}^{N} P(x_i \mid C_k)$$

The equation above implies is that in order to use Naive Bayes, it is necessary that the conditional probabilities of the feature variables are independent. Therefore, $P(X)$ becomes a normalizing constant and can be written as

$$P(X) = \sum_{i=1}^{K} \prod_{j=1}^{N} P(x_j \mid C_i) P(C_i)$$

In practicality, the normalizing constant does not play a role in determining the most likely class and only the numerator of the posterior is often considered.

Feature Types

Naïve Bayes can be applied to both continuous and discrete data. The discrete case is quite simple, and is usually modelled using a multinomial distribution. See [6].

$$P(X \mid C_k) = \frac{(\sum_{i=1}^{N} x_i)!}{\prod_{i=1}^{N} x_i!} \prod_{i=1}^{N} P(x_i \mid C_k)^{x_i}$$

This arises because in the discrete case each feature vector X will be a list of counts of occurrences. With the independence assumption, the probability of observing these features is found by multiplying the probabilities of each feature, resulting in a multinomial distribution. An example of this would be doing text classification with the bag of words model. Each independent variable would correspond to a word, and its values would be the number of occurrences in the document. It is also possible use Naïve Bayes with a Bernoulli

distribution, which would correspond to the bag of words model if counts were ignored and a binary variable was used to indicate if a word occurred or not.

For continuous data, probabilities cannot be simply computed, as in the discrete case, and a distribution of the data must be assumed. Often a normal distribution is assumed to model the independent variables. The independence assumption makes it possible to calculate the likelihood easily by multiplying the densities, and finding the most probable class. There can be issues with using the normal distribution, as it can be very inaccurate if the distribution is misspecified for the data. The R package (e1071) [5], which will be used in classification assumes normal distribution for continuous data.

Relationship with Logistic Regression

The Naive Bayes model is quite similar to the logistic regression model. The main difference between the two is that logistic regression is a discriminative classifier, while Naïve Bayes is a generative classifier. This means that Naive Bayes models the distribution of features, while logistic regression attempts to find an optimal decision boundary. Discriminative model are generally considered to perform better asymptotically, but it has been shown in [4] that generative models can work better with smaller sample sizes, because they reach their asymptotic efficiency faster.

It is worth noting that Naive Bayes, under certain distributional assumptions, is nearly identical to a simple logistic regression in that it fits a linear decision boundary in the feature spaces. More specifically, if the features are assumed to have a distribution within the power family, e.g normal, exponential, gamma, etc. the likelihood factors $P(X \mid C_k)$ can be reparameterized to be a linear combination of the sufficient statistic of the distribution and a set of weights [7]. To see this, assume two classes $C_k \in \{0,1\}$ with equal prior probability ($C_1 = C_2 = \frac{1}{2}$) and a feature vector X such that $x_i \sim N(\mu_i, \sigma_i^2) \quad \forall\ i$. The Naive Bayes classifier for $C = 1$ can be re-expressed in the form of a sigmoidal function $g(z) = (1 + e^{-z})^{-1}$ (For full derivation, see Appendix 1)

$$P(C_1 \mid X) = \frac{P(X \mid C_1)P(C_1)}{P(X \mid C_1)P(C_1)+P(X \mid C_0)P(C_0)}$$

$$= g(\sum_{i=1}^{N} ln\frac{P(x_i \mid C_0)}{P(x_i \mid C_1)} + ln\frac{P(C_0)}{P(C_1)})$$

Under the Gaussian distribution, the inner terms can be reparameterized as $W^T\tilde{N}(x_i) + c$ where

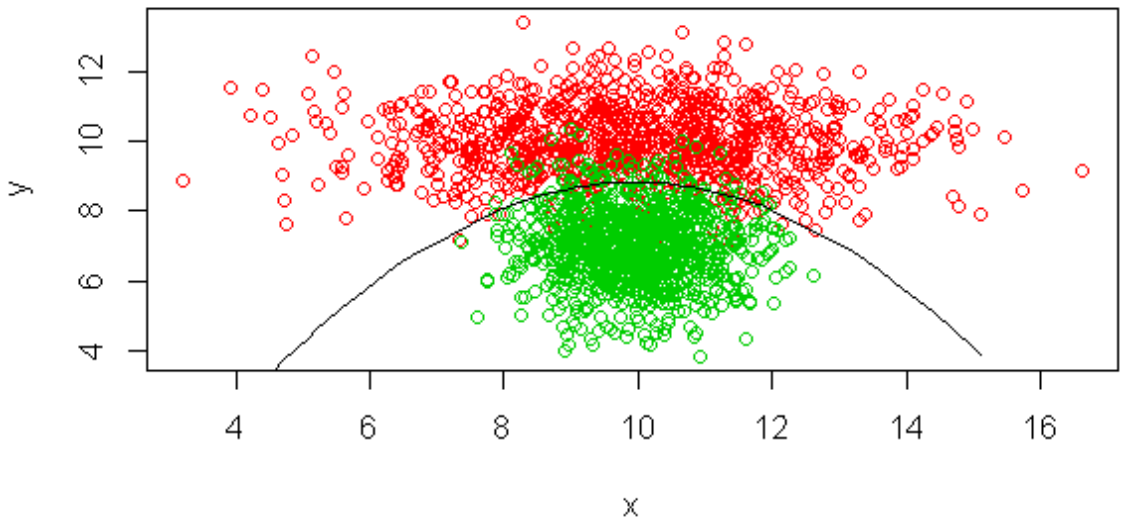
$$W = [\frac{\mu_{i,c=1}}{\sigma^2_{i,c=1}} - \frac{\mu_{i,c=0}}{\sigma^2_{i,c=0}}, \frac{1}{2\sigma^2_{i,c=0}} - \frac{1}{2\sigma^2_{i,c=1}}] \ , \ \tilde{N}(x_i) = [x_i, x_i^2] \ , \ c = ln\frac{\sigma_{i,c=0}}{\sigma_{i,c=1}}$$

The final model is almost identical to a logistic regression, which is a linear classifier, and the training data would be linearly separated under the feature space defined by the sufficient statistic $\tilde{N}(x_i)$.

$$P(C_1 \mid X) = g(\sum_{i=1}^{N} W^T\tilde{N}(x_i) + c)$$

We should note here that the actual decision boundary is not linearly separated but rather is able to fit a quadratic boundary to the data. This is due to the sufficient statistics containing both x_i and x_i^2. To visually see this, we simulated two independent gaussian feature clouds on two dimensions (features), trained the naive bayes classifier and plotted the decision boundary where $45\% \leq P(C_1 \mid X) \leq 55\%$. See figure 1 below for the plot.

Figure 1 - Two simulated Gaussian feature clouds fitted by a Naive Bayes classifier [5]. The coloring scheme refers to the unique class that they belong to.



Application to Text Mining

In text mining, the independence assumption is obviously not correct, and we know there is a large amount of dependence between the words contained in a document. This being said, there are still reason to believe that the model can perform quite well, even with the violation of the assumptions [1]. Furthermore, it is shown in [2], through evaluating a cost function, there can be significant violation of the independence assumption without losing its effectiveness. Compromising the lack of complexity and power in comparison to other models such as SVM, etc. we believe Naive Bayes can still be a feasible model due to its tractability and ability to generalize on large data sets at an efficient rate. Ignoring the structure of the document and only focusing on individual words has been quite effective and often as good as interpreting the structure. [1, p6].

3. Data Sample & Methodology

To apply the Naive Bayes concept on text mining, we decided to analyze a subset of publicly available amazon user reviews provided by Leskovec and McAuley (2013) [8]. We

arbitrarily selected six categories to examine product reviews: Watches, Office Products, Automotives, Art, Video Games and Sports & Outdoor. These classes were chosen on the basis that they would separately have unique results and less overlap than for example, electronics and software.

Methodology

The process of text mining and classification for our data set first begins through parsing the sample into a workable matrix. In general, reviews are transformed into a master corpus (Ω). The master corpus is then extensively cleaned to prepare for the bag-of-words model as such:

1. All letters are transformed to lower case letters. This is to ensure the same words are counted once in the bag-of-words model.
2. All numbers are removed since we want to ideally just isolate words as they provide more information than numbers.
3. All punctuations are removed for the same reason as (1).
4. All excess white spaces are stripped to ensure bag of words model does not capture empty values
5. Each document within the master corpus is stemmed or narrowed down to their root word. See Lovins (1968) [9].

The next step is to develop a training set and a test set. We have decided to split the data set into a 70% training and 30% test set to first train the classifier and test its ability to generalize on new reviews. To control for training time and noise, we introduce a filter (D) to reduce the unique word count through maintaining a subset of words that appears more than five times in the master corpus. It is important to explain that the word filter (dictionary) is created from the training set and then the same dictionary is subsequently applied to the test set. What this implies is that there can be important and unique words that were present in the test set but not in the training set and as a result, these words are omitted in the test phase. This is done on purpose to prevent us to overstate the

performance of the naive bayes classifier. The dependent variable in our study is defined to be a vector of discrete ratings (1-5).

In order to train the classifier, the filtered and cleaned master corpus must be transformed into a features matrix Θ of the dimensions $N \times M$ where N is equal to the number of reviews in the training (or test) sample and M is equal to the number of unique words in the dictionary D for the sample. We simplify the process and let each value within the matrix to be equal to 1 if the word appears in the document or

$$\Theta_{i,j} = \{1 \; iff \; D_j \in \Omega_i, \; 0 \; otherwise\} \quad \forall \; i, j$$

Although the loss of information through assumption of bernoulli distribution is present, this is to ensure the test set is able to work with the classifier. If we were to introduce multinomial variables, errors such as missing factor likelihoods from the test set would produce errors in creating final result. e.g the word "great" appears only at most 5 times within the training set but appeared 6 times in the test set. The only solution to this issue would be to create in-sample fit of the entire dataset which opens opportunities for bias from overfitting.

Categorical & Single-Product Classification

With most machine learning algorithms applied on language processing data sets, training and testing can be time constraining without localized computing power. Thus, our data set is often a subset of the entire category. These subsets are defined in two manners: Categorical and Single-product.

In categorical classification, we want to examine the classification of naive bayes for all products that fall within a certain category. Doing this allows us to analyze relevant variables in the entire product space that both define the type of products as well as characterize sentiments with category-relevant words. Furthermore, the categorical performance will be contrasted with the single-product performance to test our hypothesis. Categorical subsets are constructed by uniformly sampling the entire category for 2500 product reviews if the total number of reviews in the entire data set exceeds that number.

In single-product classification, we subset our data set through extracting reviews for the most reviewed item. These single product reviews are additionally subsetted through the same uniform sampling procedure if the number of reviews exceed 2500. Through single-product examination, we believe accuracy rates will improve since the commonality of words for a single product would ideally be higher than a category of product reviews.

4. Results & Discussion

Table 2.0 below presents our results for both training and test results for categorical and single-product user reviews for the six selected Amazon categories. For single-product classification, Table 2.1 lists the most popular product for each category that we analyzed for sentiments.

Table 2.0 - Training and Test set classification accuracy based on categorical and single-production classification. N is equal to the number of sample rows in each dataset.

Category	**Categorical Classification**		**Single-Product Classification**	
	Training (N=1750)	Test (N=750)	Training	Test
Watches	85.57%	57.97%	92.41% (N=1726)	82.92% (N=774)
Office Products	83.90%	56.01%	82.91% (N=907)	71.50% (N=386)
Automotives	87.67%	64.17%	100% (N=1265)	100% (N=539)
Arts	82.78%	60.24%	88.64% (N=977)	68.5% (N=416)
Video Games	76.03%	51.47%	89.98% (N=1753)	84.34% (N=747)
Sports & Outdoor	84.93%	58.55%	98.97% (N=1753)	98.79% (N=747)

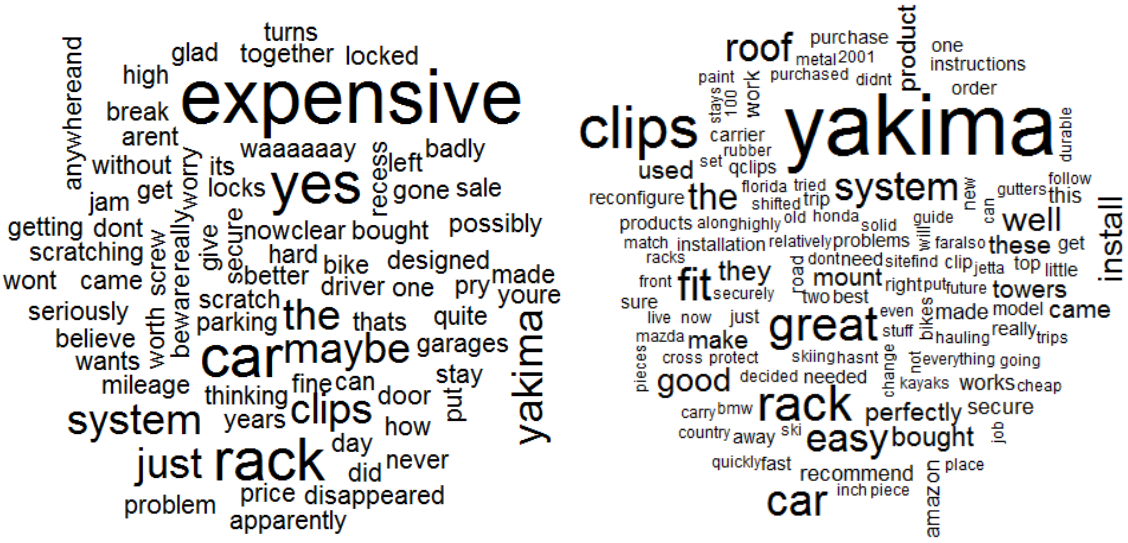
Table 2.1 - Most popular product in each category for single-product classification

Category	Product Title
Watches	Invicta Men's 8926 Pro Diver Collection Automatic Watch
Office Products	Texas Instruments TI-83 Plus Graphing Calculator
Automotives	Yakima Q Clip Set
Arts	Brother CS6000i Feature-Rich Sewing Machine With 60 Built-In Stitches, 7 styles of 1-Step Auto-Size Buttonholes, Quilting Table, and Hard Cover
Video Games	Spore
Sports & Outdoor	Russell Athletic Men's Basic Cotton Long Sleeve Tee

The evidence obtained for each category is consistent with our hypothesis between categorical and single-product classification in that single-product should outperform categorical due to a more tightly clustered sample of words. We find high training classification rates for all samples but a failure for naive bayes model to generalize onto new samples as shown by the consequent test set underperformance. This spread is mostly large within categorical classification than single-product. We believe this is due to the fact that the unique bag of words are much larger in categories than single-product subsets and new data sets provide important information that are filtered out through the training dictionary (D).

Within categorical classification, the best training and test set prediction rates were in the automotive industry with a 64.17% accuracy in the test set. Furthermore, it performed exceptionally in the single-product classification with 100% on both the training and test set. To further discover the relationship, we created word clouds for reviews for the Yakima Clip Set. See figure 2.2 below.

Figure 2.2 - Word cloud of Automotive reviews for Yakima Q Clip Set. The cloud on the left denotes negative reviews (Rating of 1 - N=82) and the cloud on the right denotes positive reviews (Ratings of 5 - N=1230). The positive word cloud was controlled for word frequency of >= 40 in the master corpus.



The word clouds help dictate the relevant words in identifying why the product was rated negatively or positively across all consumers. In the negative space, the Yakima clips were perceived to be pricey, tend to jam, can scratch and overall hard to use for the car rack. In the positive space, the Yakima clips were perceived by consumers to be easy to use for the car rack, secure and a good fit for its price.

We have also noticed that both Arts and Office Product category tend to underperform on the test set with 68.5% and 71.5% classification accuracy respectively. We attribute this underperformance due to the small data-sets provided in the training set, thus the classifier was unable to generalize on new data.

5. Conclusion

The Naive Bayes classifier is a simple method of predicting class membership given a set of features. We applied its prominent use in text mining on Amazon user reviews for six different categories. This is motivated by the research question of whether data scientists are able to classify perceived consumer utility on a certain product proxied by user rating from one to five stars. We find convincing results that this is the case as the naive bayes

classifier was able to accurately classify for all levels of user ratings on the training set but fail to maintain its accuracy when given newer data. Furthermore, we find evidence in support of our hypothesis that single product classification tend to outperform in accuracy than categorical classification. Many extensions of this study can be applied such as relaxing the independence assumption or characterizing valuable traits for all products. The main implications of this research project suggests that firms can leverage the power of text mining to further enhance aspects of their products through identifying relevant features, or to help predict the success of a product through classification of reviews.

6. Appendix

1. Derivation of Logistic function and Gaussian re-parameterization:

$$P(C_1 \mid X) = \frac{P(X \mid C_1)P(C_1)}{P(X \mid C_1)P(C_1)+P(X \mid C_0)P(C_0)}$$

$$= \frac{1}{1+\frac{P(X \mid C_1)P(C_1)}{P(X \mid C_0)P(C_0)}}$$

$$= (1 + e^{-ln[\frac{P(X \mid C_0)P(C_0)}{P(X \mid C_1)P(C_1)}]})^{-1}$$

$$= g(ln\frac{P(X \mid C_0)}{P(X \mid C_1)} + ln\frac{P(C_0)}{P(C_1)}) = g(\sum_{i=1}^{N} ln\frac{P(x_i \mid C_0)}{P(x_i \mid C_1)} + ln\frac{P(C_0)}{P(C_1)})$$

$$= g(\sum_{i=1}^{N} ln\frac{P(x_i \mid C_0)}{P(x_i \mid C_1)})$$

Under the same assumptions previously defined, the likelihood factor is the gaussian probability density function. Thus, the log ratio in the sigmoid function $ln\frac{P(x_i \mid C_0)}{P(x_i \mid C_1)}$ can be re-expressed as

$$ln(\frac{1}{\sigma_{i,c=1}\sqrt{2\pi}}e^{-\frac{(x-\mu_{i,c=1})^2}{2\sigma_{i,c=1}{}^2}}) - ln(\frac{1}{\sigma_{i,c=0}\sqrt{2\pi}}e^{-\frac{(x-\mu_{i,c=0})^2}{2\sigma_{i,c=0}{}^2}})$$

$$= ln(\frac{\sigma_{i,c=0}}{\sigma_{i,c=1}}) - \frac{x^2-2x\mu_{i,c=1}+\mu_{i,c=1}{}^2}{2\sigma_{i,c=1}{}^2} + \frac{x^2-2x\mu_{i,c=0}+\mu_{i,c=0}{}^2}{2\sigma_{i,c=0}{}^2}$$

$$= x(\frac{\mu_{i,c=1}}{\sigma^2_{i,c=1}} - \frac{\mu_{i,c=0}}{\sigma^2_{i,c=0}}) + x^2(\frac{1}{2\sigma^2_{i,c=0}} - \frac{1}{2\sigma^2_{i,c=1}}) + ln(\frac{\sigma_{c=0}}{\sigma_{c=1}})$$

$$= W_i^T\tilde{N}(x_i) + c \quad \forall\ i$$

Q.E.D

7. Works Cited

1. Lewis, David D. "Naive (Bayes) at forty: The independence assumption in information retrieval." *Machine learning: ECML-98* (1998): 4-15.
2. Domingos, Pedro, and Michael Pazzani. "On the optimality of the simple Bayesian classifier under zero-one loss." *Machine learning* 29.2-3 (1997): 103-130.
3. Rennie, Jason D et al. "Tackling the poor assumptions of naive bayes text classifiers." *ICML* 22 Aug. 2003: 616-623.
4. Jordan, A. "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes." *Advances in neural information processing systems* 14 (2002): 841.
5. Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., & Weingessel, A. (2008). Misc functions of the Department of Statistics (e1071), TU Wien. R package, 1-5.
6. McCallum, Andrew, and Kamal Nigam. "A comparison of event models for naive bayes text classification." *AAAI-98 workshop on learning for text categorization* 26 Jul. 1998: 41-48.
7. Mitchell, M Thomas. 1997. Machine Learning (1 ed.). McGraw-Hill, Inc., New York, NY, USA.
8. J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. RecSys, 2013.
9. Lovins, Julie B. *Development of a stemming algorithm.* MIT Information Processing Group, Electronic Systems Laboratory, 1968.