Sentiment Classification of Online Product Reviews Using Product Features

Neda Ale Ebrahim1, Mohammad Fathian2, Mohammad Reza Gholamian3

1. Corresponding Author
Department of Electronic Commerce, Iran University of Science and Technology, n_ale_brahim@yahoo.com
2, 3 Department of Electronic Commerce, Iran University of Science and Technology, Fathian@just.ac.ir, Gholamian@just.ac.ir

Abstract
There is a great number of online product reviews on the Internet which needs to be organized. In this paper, we consider the problem of sentiment classification of online reviews to determine the overall semantic orientation of customer reviews. Our proposed method for review classification is a supervised machine learning method based on extracting product features and the polarity of opinions expressed about the features.

Keywords: Customer Reviews, Sentiment Classification, Semantic Orientation, Product Features

1. Introduction

The web sites contain customer reviews are plentiful source of information which helps customer to decide whether or not to buy a product or which product to buy. On the other hand the information extracted from these websites enables manufacturers to know their customers preferences [1]. Automatic sentiment classification to determine the polarity of customer reviews is an effort to organize this large amount of information [2].

Turney (2002) in [3], employed an unsupervised learning technique based on PMI1 and LSA2 as two measures for computing semantic association between the extracted phrases and the words “excellent” and “poor”. The extracted phrases in his work are based on some predefined patterns of POS3 tags that contain adjectives as indicators of subjective sentences. At last to classify a review as positive or negative, he computes the average Semantic Orientation of all extracted phrases in the review.

Pang et al (2002) in [2], applied three machine learning algorithms: Naïve Bayes, Maximum Entropy, and Support Vector Machines. They used different representation vectors of review documents based on frequency or presence/absence of the extracted unigrams and bigrams as the input variables for classification. They performed sentiment classification similar to topic-based classification with two positive and negative classes as topics.

In this paper, we use supervised machine learning techniques to classify reviews as positive or negative. But instead of using n-grams as the input variables for classification, we extract frequent features of product by Association Rule Mining as in [4] and use these features as the input variables of classifier, and the semantic orientation of each feature as the value of variables in review document vectors.

2. Data Set

The data set used in this paper is 160 reviews about a digital camera gathered from Amazon4 website. Table 1 shows the description of the data set.

---

1 Pointwise Mutual Information
2 Latent Semantic Analysis
3 Part Of Speech
4 http://www.amazon.com/
3. The proposed system

Figure 1 gives the architectural overview of the proposed system.

![An Overview of The Proposed System](image)

3.1. Tokenization & POS Tagging

The first step of the proposed system is to split reviews into tokens or words, which are space character is used as splitters.

The second step is to assign Parts-Of-Speech (POS) tag to every word using NLP tools. POS Tag is the linguistic category of words. It is used to locate different types of information inside text documents. For example, generally noun phrases correspond to product features, adjectives represent opinions, and adverbs are used as modifiers to represent the degree of expressiveness of opinions [5].

3.2. Product Features Generation

The task of identifying negative or positive polarity of a review contains two major steps: product feature generation and training classifier [6].

Hu & Liu (2004) in [4] employed association rule mining to extract frequent nouns or noun phrases as the features of product from customer reviews. Association rule mining contains two steps: Frequent itemset generation and rule generation. To extract frequent features from reviews, just the first step is needed.

Let \( N = \{n_1, n_2, \ldots, n_m\} \) be the set of nouns extracted from review set and \( S = \{s_1, s_2, \ldots, s_k\} \) be the set of sentences. Considering each sentence as a transaction and the nouns exist in a sentence as the items of each transaction, the frequent itemsets (nouns or noun phrases) can be generated by the association rule mining algorithm.

In this paper, Apriori algorithm is used to generate frequent itemsets. An itemset is defined as frequent if it appears in more than 2% of the review sentences (experimentally determined minimum 2%)

---

5 Natural Language Processing
support). In addition, frequent itemsets with one word or two words are extracted as product features in this work.

Since some of the frequent features generated by Apriori algorithm are redundant and irrelevant features, two types of pruning method is proposed by Hu & Liu (2004) to remove those redundant features: Compactness pruning and Redundancy pruning.

**Compactness pruning:** A two words frequent feature whose words appear together or the distance between them is three or less than three in a sentence, is compact in that sentence. If these frequent itemsets is compact in at least 2 sentences it will not be pruned otherwise it is not a compact phrase and will be pruned.

**Redundancy pruning:** consider a single word feature which is subset of a two words feature. The single word feature will be pruned if the p-support of the feature is lower than the experimentally determined minimum p-support (in our case, it is 2), p-support (pure support) of a feature as defined in [4]:

“The count of sentences that the feature appears in, while no superset of that feature exists in these sentences”

We test this feature extraction algorithm on our dataset. The result is shown in Table 2.

<table>
<thead>
<tr>
<th></th>
<th>On word features</th>
<th>Two words features</th>
<th>All features</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of manual extracted features</td>
<td>83</td>
<td>19</td>
<td>102</td>
</tr>
<tr>
<td>No. of correct extracted features (tp)</td>
<td>46</td>
<td>11</td>
<td>57</td>
</tr>
<tr>
<td>No. of incorrect extracted features (fp)</td>
<td>29</td>
<td>14</td>
<td>43</td>
</tr>
<tr>
<td>No. of not extracted correct features (fn)</td>
<td>37</td>
<td>8</td>
<td>45</td>
</tr>
<tr>
<td>Precision tp/(tp+fp)</td>
<td>0.613</td>
<td>0.44</td>
<td>0.57</td>
</tr>
<tr>
<td>Recall tp/(tp+fn)</td>
<td>0.554</td>
<td>0.579</td>
<td>0.559</td>
</tr>
</tbody>
</table>

3.3. Sentiment words extraction & creating vectors of review documents

The features extracted in previous step are considered as the input variables of our sentiment classifier. In this step we need to determine the semantic orientation of opinion words expressed about the features (as the value of each variable) in every review. This results in making document vector of each review.

As we mentioned before, presence of adjectives in a sentence usually means that the sentence is subjective and contains opinions [5]. Consider two following sentences:

“*The photos are blurry.*”

“*The auto focus is unreliable.*”

The sentences are subjective due to presence of adjectives in them. The adjectives: “blurry” and “unreliable” are opinions about features: “photo” and “auto focus”.

So, in this step, we must extract adjectives from the sentences containing the frequent features. After extracting these adjectives, we need to determine the semantic orientation of each of them.

---

* True positive  
* False positive  
* False negative
To classify each extracted adjective into a positive or negative class, we need to use WordNet. WordNet\(^9\) is a large lexical database of English, which groups words into sets of synonyms called synsets and records the various semantic relations between these synonym sets [7].

Let G be a graph of all words with the same part-of-speech in WordNet that each two synonym words are connected in this graph. Considering each synset in WordNet as a part of this graph, the distance (shortest path) between each two nodes (words) of this graph can be computed [8].

To measure similarity between extracted adjectives and Osgood’s evaluative factors (see [9]) (e.g. good and bad), a function EVA is defined as follow:

\[
EVA(w) = \frac{d(w_{\text{bad}}) - d(w_{\text{good}})}{d(\text{good}, \text{bad})}
\]

Where \(d(w_1, w_2)\) is the shortest path between two words \(w_1\) and \(w_2\). If the semantic orientation of the word is negative, the result of this function will be a value in the interval \([-1, 0]\), and this value will be in the interval \([0, 1]\), if the semantic orientation of the word is positive.

We represent each review document as a vector of features and consider semantic orientation of them (the computed value of EVA function) as the value of the features in each vector.

Table 3 shows part of three review document vectors by 12 sample features (value 1 shows positive, -1 shows negative and 0 shows neutral semantic orientation in the vectors).

Consider that not all reviews talk about all features, so these features is also assigned by zero value in the review vectors. The name of the product is replaced with “_ProductName” in each review, to improve generalization.

<table>
<thead>
<tr>
<th>_ProductName</th>
<th>Doc1</th>
<th>Doc2</th>
<th>Doc3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Picture</td>
<td>1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>Screen</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Zoom</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Price</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Weight</td>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>Night mode</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Video quality</td>
<td>0</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>Auto focus</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Battery life</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Point and shoot</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Manual instruction</td>
<td>0</td>
<td>0</td>
<td>-1</td>
</tr>
</tbody>
</table>

3.4. Training sentiment classifier

Given a set of reviews and two positive and negative classes, a sentiment classifier classifies each review into one of the two classes.

Sentiment classification represents predominant opinion on a review. This type of text classification is different from topic-based text classification. In topic-based classification, topic related words are important. However, in sentiment classification, sentiment words that present opinions are important. e.g., good, bad, amazing, nice, etc [5].

In this paper we use Support Vector Machine (SVM) to solve this two-class problem. SVM finds an optimal hyperplane represented by vector \(\overrightarrow{w}\), that separate a review in positive class from a review in negative class. This optimal hyperplane represents the largest separation, or margin, between the two classes. (See figure 2)

\(^9\) http://wordnet.princeton.edu/
A set of positive or negative labeled document vectors $\mathbf{x}$, said to be linearly separable if there exists a vector $\mathbf{w}$ and a scalar $b$ such that the following inequalities are valid:

$$
\text{Class} = \begin{cases} 
\text{Positive} & \text{if } \mathbf{w} \cdot \mathbf{x} + b \geq 1 \\
\text{Negative} & \text{if } \mathbf{w} \cdot \mathbf{x} + b \leq -1 
\end{cases} \quad (2)
$$

The margin of the decision boundary is given by the distance between the two hyperplanes $\mathbf{w} \cdot \mathbf{x} + b = 1$ and $\mathbf{w} \cdot \mathbf{x} + b = -1$, which is equal to $\frac{2}{\| \mathbf{w} \|^2}$. SVM tries to maximize this distance [10].

4. Evaluation results

Table 4 shows the sentiment classification performance of Support Vector Machine based on frequent features and semantic orientation of them as the values of the features.

We consider accuracy measure (percentage of correctly classified reviews), as the evaluation method.

Our performance evaluation is based on five-fold cross-validation, which divide data set into five subsets and every subset will be trained and tested five times, so the average performance can be obtained.

We compared our proposed method with a classification method in [2] that uses a different representation vector of each review document, which is based on presence/absence of the unigrams and bigrams extracted from the review set (see Table 5).

The results, shows that our proposed method provides a fairly good result compared to the other method.

<table>
<thead>
<tr>
<th>Method</th>
<th>No. of one word features</th>
<th>No. of two words features</th>
<th>Values</th>
<th>SVM (Average Accuracy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequent Features</td>
<td>46</td>
<td>11</td>
<td>Semantic Orientation</td>
<td>83.6%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>of Features</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2. A Representative of Support Vector Machine
5. Conclusion

In this paper, we proposed a system for predicting sentiment orientation of online product reviews. We discussed that the system contains two major steps: feature extraction and training classifier. Then we showed that, instead of using n-grams as the features for classification, we can generate frequent nouns or noun phrases by Apriori algorithm and use these frequent itemsets as the features of sentiment classifier. After that using WordNet, we computed the semantic orientation of the adjectives extracted from sentences containing the frequent features, and by assigning the value of computed semantic orientation to each feature in every review, we generated vectors of the reviews as the classifier inputs.

Finally, we trained SVM classifier by these vectors. The evaluation result produced by this classification technique was fairly good in comparison to using n-grams as the features of the classifier.

6. References


<table>
<thead>
<tr>
<th>Method</th>
<th>No. of unigrams</th>
<th>No. Of bigrams</th>
<th>Values</th>
<th>SVM (Average Accuracy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-grams</td>
<td>626</td>
<td>394</td>
<td>Presence/Absence of N-grams</td>
<td>75.8%</td>
</tr>
</tbody>
</table>