Sentiment Identification Using Maximum Entropy Analysis of Movie Reviews

Nipun Mehra
Department of Computer Science
nmehra@stanford.edu

Shashikant Khandelwal
Department of Computer Science
kshashi@stanford.edu

Priyank Patel
Department of Computer Science
priyank9@stanford.edu

ABSTRACT:

Sentiment classification is one of the most challenging problems in Natural Language Processing. A sentiment classifier recognizes patterns of word usage between different classes and attempts to put unlabeled text into one of these categories in an unsupervised manner. Therefore, the attempt is to classify documents not by topic but by overall sentiment. We have used reviews of movies to train and test our classifier. Our system uses the Maximum Entropy method of unsupervised machine learning. We present our observations, assumptions, and results in this paper. We conclude by looking at the challenges faced and the road ahead.

1. INTRODUCTION:

The number of electronic documents today is gargantuan. With the amount of data increasing everyday foraging for information is an exceedingly difficult task. One possibility to enable a better understanding of this data is text classification. Documents can, in general, be distinguished based on their content. Exploiting this feature of documents can enable a classification mechanism. However, it would be impossible for a human classifier to go over all documents, compare them according to their content and place them in a category (in this paper we use *category* and *class* interchangeably) for the number of documents that are present today. Even if an attempt were made, the classification would not be free from biases introduced sub-consciously or otherwise. An automated machine learning system that could interpret class from context could be expected to outperform human classification if it could be personalized to the tastes and preferences of the person using the classification. Such sentiment classification finds use in intelligent applications (e.g. Mindfuleye's Laxent system) and recommender systems (e.g. Terveen *et al*)

Perhaps the best exemplifying feature of the differences in tastes of people is the preference of movies. For example, romantic comedies might appear to be interesting to some while they might seem like drivel to others. The reviews written by experienced authors does provide some idea of the quality of a movie, but is written taking into account the general tastes and preferences based on trends observed in tastes over a period of time. Expert reviews are an excellent source of classification for movies that lie on extreme ends of "good" or "bad". However, most movies do not lie on these extreme ends. Users tend to have their own beliefs over qualities of most movies. It is our attempt to recognize this variance and to customize a classification system that can gauge the level of interest a user might place in a particular movie.

The concept of personalized classification performs the essence of our work. We recognize that different people have different preferences, likes and dislikes. Therefore, it would be inappropriate to use a uniform classification system for all persons using the classification.

The modeling of a stochastic system can be done based on a sample of output for a particular set of input. This data sample does not provide complete information of the system. In order to be able to model

such processes exactly, one needs an infinite set of inputs and outputs within the domain permitted by the system. However, a reasonable approximation can be made of the system using a finite set of values if the values are sufficiently random and the occurrence of one in no way influences that of the other. Text classification is one such system that needs to be modeled based on the data provided by a training sample – a finite input set.

A number of text classification algorithms have been examined. We present the motivation behind our use of the Maximum Entropy classification system.

i) *Naïve Baye's Classification Algorithm:* In naive Bayes, the probability of each word appearing in a document of a certain class is estimated directly from the training data (Manning and Schutze 1999). The occurrence of words is believed to be independent of each other. The conditional probability of each word is calculated given the class to which a document belongs.

$$P(w|c) = \frac{s + \sum_{d \in C} I(w,d)}{s + N_{d \in C}}$$

In this equation, c is a specific class, d is a document within the class c, $N\tilde{d}C$ is the number of documents within the class c, and s is a pseudocount to compensate for unseen events. $I(w,d)$ is an indicator function that is 1 if the word w is in the document d, 0 otherwise. Once these probabilities are estimated the probability that a document belongs to a particular class is merely a product of the probabilities of each of the words occurring in the document.

However, this classification system makes one fundamental assumption - words in a document, category pair occur independent of other words. Therefore, the word "Operating" is equally likely to be present in a document that contains the word "System" as another document that does not contain the word "system".

ii) *Nearest Neighbor Classification:* In nearest-neighbor classification, a distance metric is employed to calculate the distance between the word vector of an unclassified abstract in the test set and each of the abstracts in the training set (Manning and Schutze 1999). An unknown document is then classified as the most prevalent category among a pre-specified number of closest training documents in the document vector space.

The Nearest Neighbor Classification system too makes an independence assumption by representing each word in the vector space as an axis. A dependence assumption would mean that the axis are not perpendicular to each other.

iii) *Maximum Entropy Classification:* In Maximum Entropy classification, the probability that a document belongs to a particular class given a context must maximize the entropy of the classification system. By maximizing entropy, it is ensured that no biases are introduced into the system.

The model makes no assumptions of the independence of words. However, it is computationally more expensive. This classification algorithm will be discussed later in the methods used.

2. EARLIER WORK:

There has been considerable work in text classification using different techniques and optimizations. Related work in the field of classification has predominantly been in genre identification [Karlgren and Cutting, 1994], ontology based classification [Raychaudhari and Altman, 2002], or for subjective genres [Kessler et al, 1997].

The procedure we use for determining the opinions or sentiments from the text is similar to the one used by Raychaudhari *et al* used for classification of Biomedical Abstracts. However, since the categories in such ontology based classification are well-defined and distinct, the categories used in ranking the movies are ordered and not equidistant. This caused a reduced accuracy in the reduced classifier. Raychaudhari et al use a statistical feature selection method called chi-square. Chi-square is a test for statistical significance that provides words that are most skewed across all categories in the training set [Manning and Schutze, 1999].

Most previous work on sentiment classification has been language based rather than statistical. For example, the orientation of words and phrases provided an idea of the direction of sentiment [Hatzivassiloglou and McKeown, 1997]. They take certain seed words that are manually provided to the system

3. THE ANALYSIS DATA:

The reviews we used, to serve as the input for text analysis, were obtained from the newsgroup rec.arts.movies.reviews. The reviews are available on the web-site www.imdb.com. . The reviews are classified by author. The names of the authors were obtained from a list of names, again made available by the imdb.com. The reviews were read into a file based on the author names and later transformed into files based on movie titles by picking up the reviews of a movie from various author files.

4. RESULTS:

It was found that the accuracy of the system varies significantly form user to user. For users with strong likes and dislikes, the system tended to perform well. However, for some others with tastes that varied across genres, castes and other features of movies, tended to have a low accuracy.

The accuracy of the system also tended to vary with the number of features. For a low number of features, the accuracy tended to be low. For example, with just 150 features, the system tended to behave like a random system with an average accuracy of around 25%. Similarly, for a large number of features also the accuracy is low. The optimum accuracy was found to occur at 350 features at 1500 iterations for the data.

The number of iterations too affected the accuracy of the system. For a small number of iterations the parameters alphas tended not to satisfy the expected versus observed constraint. However, a large number of iterations caused the parameters to over-fit the sample data causing degradation of the accuracy. 1500 iterations of the GIS algorithm achieved the optimum results.

We used four classification techniques:

1. The first technique was a simple probability analysis of the classification. Since Maximum Entropy provides the probabilities with which a document belongs to a particular class, we simply picked the most probable class as the system's guess.

Table 1: Results for 3 methods using a single classifier

		Accuracy			Pearson			Kenadll Tau	
Users	Method1	Method2	Method3	Method1	Method2	Method3	Method1	Method2	Method3
nmehra	35	26	65	0.1	0.77	0.31	0.28	0.668	0.336
kshashi	39	39	56	0.3471	0.6606	0.1482	0.268	0.5033	0.085
garima	37	37	68	-0.1	0.7	-0.034	0.1602	0.645	0.2208
ruchir	40	40	65	0.295	0.7475	0.1886	0.2211	0.6737	0.0947
gussy	37	45	64	-0.1702	0.7341	-0.03	0.16	0.645	0.2208
ashu	64	59	84	0.2361	0.8562	0.3107	0.4144	0.8837	0.3763

2. In the second method, we take the better of the system's top two guesses. It is interesting to note that by doing this, the Karl Pearson's coefficient between the observed and expected became high though the accuracy reduced.

3. The third method of classification recognized the ordinal nature of the data. Since the rankings are ordered, we can get a better measure of accuracy by choosing the best document from a grouping of the four probabilities as follows:

   <Group: {Categories},{Categories}>

   Group1: {1},{2,3,4}

   Group2: {1,2},{3,4}

   Group3: {1,2,3},{4}

   We calculate the probability of a movie belonging to a particular set of categories by summing up the probabilities of its constituent elements. We then see which category is common to all three groupings and pick that as the most likely category. As seen in the table, the accuracy for this method is remarkably high achieving as much as 84.09% accuracy.

4. The fourth classification model we used was with three separate classifiers. We use the scheme similar to Method 3 above but instead of summing probabilities based on one classifier, we treat each group as a 2-category system and repeat the whole process from feature generation to alpha determination for three different sets. The reasoning behind this modeling is the fact that each group can be treated as a "good" vs "bad" system at each step. Once the likelihood of the document belonging to either of the two categories in each group is known, we obtain the most likely classification by obtaining the most consistent classification across the groups. If there is any inconsistency, we choose the classification provided to us by Group 2 since that is the most uncertain of all groups.

This process achieved the highest accuracy of 85%. Also, the average accuracy was always greater than 50% for every user.

5. METHODS:

5.1 Overview:

i) *User Interaction:* We provide human users to rank the movies according to their tastes and preferences. This is accomplished by providing radio buttons corresponding to ranking numbers 1,2,3 and 4 beside a movie title on a web interface located at

http://www.stanford.edu/~kshashi/movierater

These ratings are then stored into a directory meant for that user.

ii) *Creation of Training and Test Tests:* We split the user provided rankings of movies into two sets – training set and test set.

The test set is created by assigning every fifth movie rated by the user to the Test data file. All remaining movies constitute the training set.

iii) *Document Pre-processing* - All documents are read along with the categories to which they belong. We realized that the "token access" operation is the most frequent operation in our system. To improve the efficiency of the system, we implemented our data structures as "hash arrays". Three hash tables are made. The first is the Lexicon containing the words along with the respective numerical identifier for the word.. The second table contains the count of the occurrence of a word across all documents over all categories and also the number of documents in a particular category that contain the word. The third hash table is simply a list of unique words contained in a document. The hashing function was written to uniquely hash words into the tables.

After reading all the documents into the memory as hash arrays, we eliminated words that occurred in two or less documents and those that occurred in more than three quarters of all documents read, thus eliminating stop words that do not describe any category.

5.2 The Maximum Entropy Classifier:

Maximum Entropy is a machine learning method based on empirical data. Nigam et al and Berger et al showed that in many cases it outperforms Naïve Baye's classification. Raychaudhari et al also found that Maximum Entropy worked better than Naïve Baye's and Nearest Neighbor classification for their classification. Unlike the Naïve Baye's machine learning, Maximum Entropy makes no independence assumptions about the occurrence of words.

The Maximum Entropy modeling technique provides a probability distribution that is as close to the uniform as possible given that the distribution satisfies certain constraints. We provide only a terse overview of Maximum entropy. A full description of the method can be found in Manning and Schutze, 1999 and Ratnaparkhi 1997.

The classification system is well described by Adwait Ratnaparkhi as:

"….Maximum Entropy models offer a way to combine diverse pieces of contextual evidence in order to estimate the probability of a certain linguistic class occurring with a certain linguistic context….in which task is to estimate the probability of class 'a' occurring with context 'b'….." [A Simple Introduction to Maximum Entropy Models for NLP, Adwait Ratnaparkhi, 1997].

The principle of the Maximum Entropy modeling states that:

"….The Maximum Entropy probability distribution, P*, is the unique distribution that maximizes:

$$H = \Sigma\, P(V) \log(P(V)) \;\forall\; V$$

While satisfying the supplied constraints…." [MIT/LCS/TR – 391]

The Maximum Entropy classification requires a set of features, which define a category. For example, in case of documents features could be the words that belong to the documents in that category. A feature *f* is a binary function that maps to '1' if a document belonging to a category contains the feature (word). Thus:

$f_{example} = 1$ *iff "profit"* ε *d and c="earnings"*

The probability that a document belongs to a particular category is given by:

$$P(c_j \mid d) = \frac{1}{Z(d)} \exp(\Sigma_i \lambda_i f_i(d, c_j))$$

Where P(cj|d) is the probability that a class occurs for a given document. Z(d) is the normalizing constant that is obtained by summing over all P(cj | d) over all values of j. The probability distribution P*(V) is calculated by an iterative method called Generalized Iterative Scaling [Darrcoch and Ratcliff, 1972], which begins with a representation of the uniform distribution and converges towards the maximum entropy distribution. The values of λi are obtained so that the system satisfies the constraint that the observed Expectation of a feature in the universe should match the expectation of the feature in the given sample set.

Feature Selection:

The words, which serve as features for a text are chosen using the chi-square method of [Manning and Schutze, 1999]. Chi-square selects words that have the most skewed distribution across categories. This includes words whose occurrence in a category is either much larger than the expected distribution or is much lesser than the expected distribution. For example, we encountered words like "Gibson" and "Arnold" in a particular user's profile who tended to show a preference for action movies. However, certain words that are never present in a category are also selected as features since their absence causes the distribution to be skewed. However, this leads to the expectation of this feature to be zero during the Maximum Entropy calculation because Maximum Entropy does not recognize the parity of the features. This causes zero probabilities to propagate iteratively causing alphas to go to infinity. To get around this difficulty we set the minimum observed expectation to be 0.01 times the minimum observed expectation of the features during the training set expectation calculation.

Generalized Iterative Scaling:

Using the Generalized iterative scaling algorithm [Darroch and Ratcliff, 1972], we find parameters or weights of the features selected. The details of this method are beyond the scope of this paper. [Darroch and Ratcliff, 1972] proved that this iteration converges to the most random distribution satisfying certain constraints.

FUTURE WORK:

Recognizing semantic and linguistic features instead of just the statistical significance of words can enhance performance of the system. Also, the iterative scaling can be improved by using Improved Iterative Scaling [Berger, 1997]. Since the corpora is small, the classification accuracy can be somewhat improved by grouping users with similar tastes.

REFERENCES:

[1] Manning, Christopher D. and Schutze, Hinrich. Foundations of Statistical Natural Language Processing.

[2] Terveen L, Hill W., *Beyond Recommender Systems:* Helping People Help Each Other (2001),

[3] Domings P. and Pazzani M. J, On the optimality of simple Bayesian classifier under zero-one loss *Machine Learning (*1997)

[4] Finn A., Kushmerick, N. and Smith B. Genre classification and domain transfer for information filtering. In Proc of *European Colliquium on Information Retrieval Research (*2002)

[5] Agresti A., *Categorical Cluster Analysis*

[6] Raychaudhuri S, Chang JT, Sutphin PD, and Altman RB Associating genes with Gene Ontology codes using a maximum entropy analysis of biomedical literature. *Genome Research* 12:203-214, 2002

[7] Berger A., A Brief Maximum Entropy Tutorial

[8] Turney P. Thumbs up or Thumbs down. Semantic Orientation applied to unsupervised classification of reviews. In *Proc of the ACL (*2002)

[9] Nigam K., Lafferty J., and McCallum A. using maximum entropy for Text Classification. In *Proc of the IJCAI-99 Workshop on Machine Learning for Information Filtering(*1999)

[10] Goldman S.A.. Efficient Methods for calculating maximum entropy distributions *MIT/LCS/TR-391*

[11] Ratnaparkhi A., A Simple Introduction to Maximum Entropy Models for Natural Language Processing.

[12] Berger A., The Improved Iterative scaling algorithm: A Gentle Introduction, 1997