Sentiment Analysis

A LOOK INTO MACHINE LEARNING APPROACHES FOR SENTIMENT ANALYSIS

Submitted To: Prof. R. Kala

Group Members

RIT2012012, RIT2012028

RIT2012037, RIT2012047

RIT2012063, RIT2012068

Table of Contents

I. Introduction

Sentiment Analysis today is a rapidly developing field with a range of techniques targeting the recognition of sentiment reflected in documents [13].

Formally, "**Sentiment Analysis** (also known as **opinion mining**) refers to the use of natural language processing, text analysis and computation linguistics to identify and extract subjective information in source materials. "[1]

In our project, Sentiment Analysis refers to task of identifying whether the opinion expressed in a text is positive or negative. For example – "Pacific Ream is such a good movie, highly recommended 10/10." expresses positive sentiment towards the movie named Pacific Ream.

Sentiment-related information can be encoded lexically within the actual words of the sentence, syntactically, and morphologically through changes in attitudinal shades of word meaning using suffixes. [2]

Nowadays, due to emergence of web2.0 and sites like IMDb™ and Rotten Tomatoes™, the task of sentiment analysis becomes more interesting. In this work, we used dataset of manually annotated text of movie reviews from IMDb™ extracted by Bo Pang and Lillian Lee. [3]

This report shows Machine Learning Approaches as a viable technique for Sentiment Analysis. We used following Approaches:

1) Naïve Dictionary lookup.
2) Naïve Bayes.
3) Support Vector Machines.

II. Terminology

There has been to date no uniform terminology established for this relatively young field. In this section, we attempt to explain some of the terms that are currently in vogue, and what these terms tend to mean in research papers.

To see that the distinctions in common usage can be subtle, consider how interrelated the following set of definitions given in **Merriam-Webster's Online Dictionary** are:

Synonyms: opinion, view, belief, conviction, persuasion, sentiment

- **Opinion** implies a conclusion thought out, yet open to dispute *(each expert seemed to have a different **opinion**)*.

- **View** suggests a subjective opinion (*very assertive in stating his **views***).
- **Belief** implies often deliberate acceptance and intellectual assent (*a firm **belief** in her party's platform*).
- **Conviction** applies to a firmly and seriously held belief *(the **conviction** that animal life is as sacred as human*).
- **Persuasion** suggests a belief grounded on assurance (as by evidence) of its truth (*was of the **persuasion** that everything changes*).
- **Sentiment** suggests a settled opinion reflective of one's feelings (*her feminist **sentiments** are well-known*).

According to Dave et al. [5], the ideal **opinion-mining** tool would "process a set of search results for a given item, generating a list of product attributes (quality, features, etc.) and aggregating opinions about each of them (poor, mixed, good). "

According to Pang et al. [4], the term **Sentiment Analysis** parallels to **Opining mining** in certain aspects. A sizeable number of papers mentioning "sentiment analysis" focus on the specific application of classifying reviews as to their polarity (either positive or negative). However, nowadays it refers to broader meaning of doing computational treatment of opinion, sentiment, and subjectivity in text.

Thus when broad interpretations are applied, "Sentiment Analysis" and "Opining Mining" denote the same field of Study (Which itself can be considered a sub-area of subjectivity analysis).

Bag of Words is simple representation of a document/text used in Natural Language Processing and Information Retrieval. In this model, a text is considered as a bag of its words, disregarding its grammar, word order etc. and only keeping its words and their frequency.

Decision Planes are affine Hyper Planes which can be described by a single linear equation in Cartesian Coordinates.

Margin is distance of the object/Point from the decision hyper plane.

Smoothing [9] refers to the idea of avoiding noise in data.

Tokenization [10] is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens.

Stemming [11] is the process for reducing inflected (or sometimes derived) words to their stem, base or root form—generally a written word form.

Stop Word [12] are the words which don't carry much lexical information". For ex – "the", "or" etc.

III. Background

Major of the Background and description has been taken from Erik et al. [14]

Existing approaches to sentiment analysis can be grouped into three main categories: **keyword spotting**, **lexical affinity**, and **statistical methods**.

Keyword spotting is the most naive approach, also known as **Naïve dictionary lookup Approach**, and probably also the most popular because of its accessibility and economy. Text is classified into categories based on the presence of fairly unambiguous affect words like happy, sad, afraid, and bored. The weaknesses of this approach lie in poor recognition of affect words when negation is involved, and many others. As an example of the weakness, while the approach can correctly classify the sentence “today was a happy day” as being positive, it’s likely to fail on a sentence like “today wasn’t a happy day at all.”

Lexical affinity, is slightly more sophisticated than keyword spotting as, rather than simply detecting obvious words, it assigns arbitrary words a probabilistic affinity for a particular emotion. For example, “accident” might be assigned a 75 percent probability of indicating a negative effect, as in “car accident” or “hurt by accident.” These probabilities are usually trained from linguistic corpora. Though often outperforming pure keyword spotting, there are two main problems with the approach. First, lexical affinity, operating solely on the word-level, can easily be tricked by sentences like “I avoided an accident” (negation) and “I met my girlfriend by accident” (other word senses). Second, lexical affinity probabilities are often biased toward text of a particular genre, dictated by the source of the linguistic corpora. This makes it difficult to develop a reusable, domain-independent model.

Statistical methods, such as **support vector machines (SVM)**, have been popular for classification of texts. By feeding a machine learning algorithm a large training corpus of affectively annotated texts, it’s possible for the system to not only learn the keywords (as in the keyword spotting approach), but also to take into account the probabilities of other arbitrary keywords (like lexical affinity), punctuation, and word co-occurrence frequencies.

IV. Previous Research Work

There have been many papers written on sentiment analysis for the domain of blogs and product reviews. (Pang and Lee, 2008) [4] Gives a survey on sentiment analysis. Researchers have also analyzed the brand impact of microblogging. Overall text classification using machine learning is well studied field (Manning and Schütze, 1999 et al) [6].

(Pang and Lee, 2002) [7] Researched the effect of various machine learning techniques (Naïve Bayes (NB), Maximum Entropy (ME), and Support Vector Machines (SVM)) in the specific domain of movie reviews. They were able to achieve an accuracy of 82.9% using SVM and a unigram model.

Researchers have also worked on detecting sentiment in a text. (Turney 2002) [8] Presents a simple algorithm, called sematic orientation, for detecting sentiment. (Pang and Lee 2004) [3] Presents a hierarchical scheme in which text is first classified as containing sentiment and then classified as positive or negative.

Work has been done in using emoticons as labels for positive and negative sentiment. This is very relevant to Twitter because many users have emoticons in their tweets.

V. Machine Learning Approaches

5.1 Naïve Bayes

Naïve Bayes is simple (Naïve) Classification based on Bayes rule. It relies on very simple representation of document, Bag of Words. Naïve Bayes, mathematically, can be treated as follows:

For a document *(d)* and class *(c).* By Bayes Theorem

$$P(c \mid d) = \frac{P(c)\, P(d \mid c)}{P(d)} \qquad \text{eq 5.1}$$

We derive the Naive Bayes (NB) classifier by

$$c^* = \arg max_c \;\; P(c \mid d\,) \qquad \text{eq 5.2}$$

Where (c*) is the class assigned to document *(d)* by Naïve Bayes Classifier.

To estimate the term $P\,(d\,/\,c)$, we make some assumptions:

- Document is represent by a **set of Features** and considered as a bag of words.
- Probability of document (d) given a class (c) is **joint probability** of each feature (x_i) given class (c).
- **Conditional Independence**: Assumes the feature probability $P\,(x_i\,/\,c_j)$ are independent given the class (c).

$$P(d \mid c) = \; P\big(x_1, x_{2,} \ldots \ldots . x_n \mid c\big)\, P(c) \qquad \text{eq 5.3}$$

Where $x_{1,}\ x_2 \ldots.\ x_n$ are features of the document.

So Naïve Bayes decomposes to:

$$P_{NB}(c \mid d) = \frac{P(c)(\prod_{i=1}^{m} P(x_i \mid c)^{n_i(d)}}{P(d)} \qquad \text{eq 5.4}$$

Training method consists of relative-frequency estimation of *P(c)* and *P* (x_i / c).

In practice, it needs smoothing [9] to avoid zero probabilities. Otherwise, the likelihood will be 0 if there is an unseen word when it's making prediction.

This Method performs pretty well in movie review analysis.

5.2 Support Vector Machine

Support vector machine (SVM) is a supervised learning approach to analyze data and recognize patterns. SVM is used in classification and regression tasks.

SVM is based on finding planes that defines decision boundaries. Decision Planes are affine Hyper Planes which can be described by a single linear equation in Cartesian Coordinates. Decision plane separates objects by its class memberships. Decision boundary can be a line or a hyper plane. There are two types of classifier used in SVM classification namely, linear and Non-Linear classifiers.

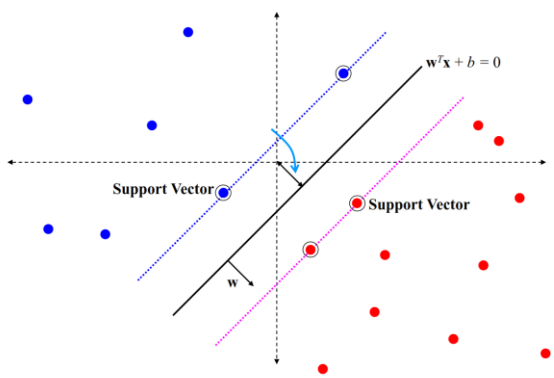

Fig 5.2.1: Linearly Separable Data

(Source: C19 Machine Learning lectures Hilary 2014)

SVM separates objects according to the class such that separation boundary offers a maximum margin (margin is distance of the object from the decision hyper plane) with objects of all classes closer to him. Support vectors and decision boundary (a line in 2D) for linear classification is shown in Fig 5.2.1.

For Linear classification – :

If we have two labels negative and positive.

$$y_i = sign(w^T x_i + b) = \begin{cases} 1\,;if\ w^T x_i + b \geq 0 \\ -1\,;if\ w^T x_i + b < 0 \end{cases} \qquad \text{eq 5.5}$$

w	Normal Vector to Hyper Plane
b	defined as $\frac{\lvert b \rvert}{\lVert w \rVert}$ is perpendicular distance from hyper plane to origin
x_i	Feature vector

i. Decision boundary $y_i(w^T x_i + b) = 0$.
ii. For negative objects $y_i(w^T x_i + b) \leq -1$.
iii. For positive objects $y_i(w^T x_i + b) \geq 1$.

For non-linear classification of Inseparable data

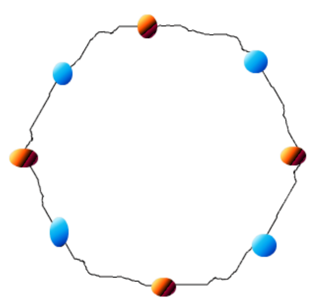

Fig5.2.2: Non-Linear Separable Data

(Source: http://www.quora.com/What-are-Kernels-in-Machine-Learning-and-SVM)

Suppose we have objects of two classes as shown in Fig5.2.2. Using Linear SVM we cannot find a Decision Hyper Plane because these Objects are not linearly separable, hence kernel functions are used. Kernel functions can do separation of multiclass objects without affecting dimensionality because it works on dot product in a feature space. Kernel simply maps our feature vector to higher dimensions but it doesn't transform feature vector to high dimension. If we have high dimensional feature vector then simply using kernel may cause infinite dimensional mapping, for avoiding this we use Kernel functions with kernel trick.

Popular Kernel functions – :

Linear kernel: $k(x_i, x_j) = x_i^T x_j$.

Polynomial: $k(x_i, x_j) = \gamma(x_i^T x_j + r)^d, \gamma > 0$.

Radial basis function (RBF): $k(x_i, x_j) = \exp -\gamma(\left\|x_i - x_j\right\|^2), \gamma > 0$.

Sigmoid: $k(x_i, x_j) = \tanh \gamma(x_i^T x_j + r)$.

VI. TOOLS AND FRAMEWORKS

- Python 3.4, Natural Language Toolkit 3.0.

  Basic NLP tasks are performed using NLTK 3.0 such as: Stemming, Tokenization, Corpus Reading, Stop Words Removal etc.

- LIBSVM v3.20 (A library for Support Vector Machines).

  Used Bag of Words Representation of Review for Feature Vector with frequency omitted.

VII. Result

Approaches	Accuracy (%)
Naïve Dictionary Lookup	54
Naïve Bayes Classification	79
Linear Support Vector Machine	78
Support Vector Machine with polynomial kernel	81
Support Vector Machine with Radial kernel(RBF)	83

Table 7.1: Result

Kernel Parameters	Value
C (cost)	0.5
γ (Gamma)	0.002
R (Coef0)	78.8889

Table 7.2: Kernel Parameters Used

In Naïve Dictionary Lookup Approach we achieved very less efficiency, because it is a simple and naïve approach which classify text according to number of occurrence of positive and negative words. So we moved to another Machine Learning approaches for getting better accuracy.

In Naïve Byes classification we achieved 80% accuracy which is acceptable but not much satisfactory. So we will switch to another Machine Learning approach works using support vectors which is SVM.

In support Vector Machine approach, if we use linear kernel we got only 78% accuracy which is less than Naïve Bayes classification, because our generated data is not linearly separable. SVM with polynomial kernel gives efficiency more than naïve Bayes and then we used SVM with RBF kernel. SVM with RBF kernel gives accuracy of 83%, higher than all other implemented approaches.

VIII. References

[1] *Sentiment Analysis*, http://en.wikipedia.org/wiki/Sentiment_analysis

[2] J. Reilly and L. Seibert, "*Language and Emotion*," Handbook of Affective Science, R.J. Davidson, K.R. Scherer, and H.H. Goldsmith, eds., pp. 535-559, Oxford Univ. Press, 2003.

[3] Bo Pang and Lillian Lee, "*A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts"*, Proceedings of the ACL, 2004.

[4] Bo Pang and Lillian Lee, "*Opinion mining and sentiment analysis*," Foundations and Trends in Information Retrieval 2(1-2), pp. 1–135, 2008.

[5] Kushal Dave, Steve Lawrence, and David M. Pennock. "*Mining the peanut gallery: Opinion extraction and semantic classification of product reviews*". In Proceedings of WWW, pages 519–528, 2003.

[6] Chris Manning and Hinrich Schütze, (1999). "*Foundations of Statistical Natural Language Processing"*, MIT Press. Cambridge, MA: May 1999.

[7] Bo Pang, Lillian Lee and Shivakumar Vaithyanathan (2002). "*Thumbs up? Sentiment Classification using Machine Learning Techniques*". Proceedings of EMNLP 2002, pp. 79– 86.

[8] Peter Turney (2002). "*Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews*". Proceedings of the Association for Computational Linguistics. pp. 417–424, 2002.

[9] *Smoothing*, http://en.wikipedia.org/wiki/Smoothing

[10] *Tokenization*, http://en.wikipedia.org/wiki/Tokenization_%28lexical_analysis%29

[11] *Stemming*, http://en.wikipedia.org/wiki/Stemming

[12] *Stop Word,* http://en.wikipedia.org/wiki/Stop_words

[13] Neviarouskaya, Alena, Helmut Prendinger, and Mitsuru Ishizuka. "*SentiFul: A lexicon for sentiment analysis*." Affective Computing, IEEE Transactions on 2.1 (2011): p22-36.

[14] Cambria, Erik, et al. "S*tatistical approaches to concept-level sentiment analysis.*" IEEE Intelligent Systems 28.3 (2013): 6-9.