SVM based classifier for Blogs

Kaushik Subramanian
Department of Chemical and Biological Engineering

Abstract

In this project a classifier is developed that can classify a blog into different categories based on the topics that author frequently writes about. Treating each blog as a document and each topic category as a class, a SVM based multi-class classifier is developed. The impact of feature selection has been studied by using different methods to generate the feature vector from the documents.

Introduction

Blogs are increasingly becoming a major component of user interaction with the world wide web. Estimates made in 2005, indicate that there were more than 50 million blogs worldwide and this number is growing. Many businesses are recognizing that blogs are an important avenue for interacting with potential customers. Natural language processing based tools can be used cluster, classify and analyze these blogs to tailor business requirements like placing advertisements etc.

In this project, data from "Indie-bloggies", an online competition to find the best Indian blogger. As a part of the nomination process every contestant submits his/her blog along with the most relevant category it belongs to (Humor, Business, Travel, Science, Humanities, Food, Personal, Entertainment). This gives us a data set in which blogs have been tagged with their categories. Treating each blog as an document and each category as a class, a SVM based classifier is trained and tested. Different methods of selecting the feature space is studied to obtain a good classifier.

Tools

Support Vector Machines

Support vector machines are popular tools for designing classifiers.The multi-class SVM (Crammer and Singer 2001), finds a classifier $H : \mathcal{X} \rightarrow \mathcal{Y}$ of the form

$$H_M(x) = arg\max_{r \in \mathcal{Y}}(M_r.x)$$

in which M is a $k \times n$ matrix and M_r is the r^{th} row of M. k is the number of classes in the problem and n is the dimension of the feature space. $\mathcal{X} \subseteq \underline{\mathrm{R}}^n$ is the feature-space, while $\mathcal{Y} == \{1, 2, \ldots k\}$ is the set of classes. Given a training set $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, H_M is the matrix that minimizes the empirical error for multi-class problem in the training set S

$$\epsilon_S(M) = \frac{1}{m}\sum_{i=1}^{m} \|H_M(x_i) \neq y_i\|$$

in which $\|H_M(x_i) \neq y_i\|$ is 1 if x_i is misclassified by the matrix H_M

Feature Selection

Selecting features that contain maximum information about the training data is essential in finding a good classifier. In the case of documents, Bag of Words (BOW) is a very popular feature used. In this feature, the feature space is all the words that appear in the training. Sparse features (features that appear in very few documents) are not good for classification, as they have poor recall, while features that appear in every document have poor precision. Thus, a good feature space consists of words that appear in quite a few documents, with moderate frequency.

Term Discrimination (Willett 1985) is a systematic procedure to remove sparse features. Consider a data matrix $\mathcal{A} \in \underline{\mathrm{R}}^{n \times k}$in which each column is a representation of one document. There are k documents and each row is a feature. Let $m \in \underline{\mathrm{R}}^k$ is the distance of each document from the mean document $c \in \underline{\mathrm{R}}^n$.

$$m(i) = |A(:, i) - \sum_{j=1}^{k} A(i, :)/k|_2 \qquad \forall i = 1, 2, \ldots k$$

We now remove word j from the corpus. A_j is the data matrix without the j^{th} row.and calculate $\tilde{m} \in \underline{\mathrm{R}}^k$, the distance of each document from the mean document in the reduced space $\underline{\mathrm{R}}^{n-1}$. Term Discrimination for the word j is $\frac{1}{k}\sum_{l=1}^{k} m(l) - \tilde{m}(l)$.We select the top r words to form a reduced $\underline{\mathrm{R}}^r$ dimensional feature space

TF-IDF based methods are another popular tool to select features. In this method, each word in a document is assigned a $TF_{i,j}.IDF_j$ score. TF denotes term frequency,

and is the number of times word j occurs in a document, while $IDF_j = \log(H/d_j)$ is the inverse document frequency, the number of documents in which the word j occurs. The $TF.IDF$ score weights both frequency in a document and frequency of appearance of the word in the dataset. In this project, the words were ranked by a $TF.IDF$ based score is used to generate the top r words and the following three approaches were used to generate the features:

- Approach-1: BOW feature vector for the in the $\underline{\mathrm{R}}^r$ space.
- Approach-2: TF-IDF feature vector in the reduced dimensional space.
- Approach-3: Modified TF-IDF feature vector, with less stress on term frequency (S.M.Rüger and S.E.Gauch 2000).

$$v_{ij} = \log(1 + t_{ij}).\log(|D|/d_j)$$
$$\forall j = 1, 2, \ldots r, \forall i = 1, 2, \ldots k$$

Methods

370 blogs, categorized into 8 categories were downloaded. The data set was divided into a training set consisting of 250 blogs and a test set consisting of 120 blogs. Each blog was downloaded, and the blog-posts were extracted, discarding all the other parts of the blog (comments, links etc). Stop words were removed and a corpus was made of all the distinct words. This corpus had length 456300. A smaller corpus consisting of 23607 words was created by applying a criterion that a word must occur in at-least 10 documents. Different smaller corpuses of lengths 1000,2000,5000 and, 10000 words were made using the feature selection methods discussed in section .

A multi-class SVM was trained for features generated by each corpus and tested against the test set. The results of the classification exercise are presented in the next section.

Results

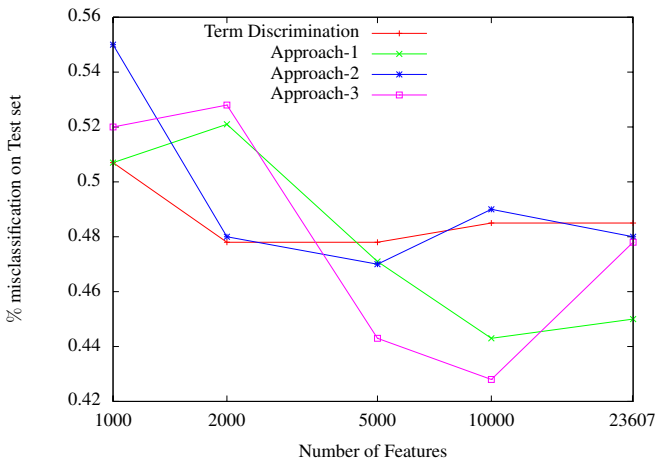

Figure 1: Accuracy as a function of number of features

Figure 1 shows the % misclassification as a function of the number of features for the training set. Analysis of the

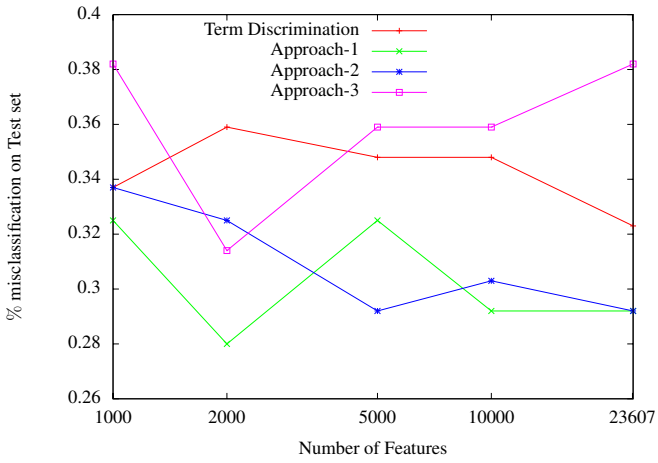

Figure 2: Accuracy as a function of number of features

results showed that classifying blogs categorized "Personal" was the hardest. This can be attributed to personal blog authors writing about different topics (a post about a recent holiday, followed by a post about government policy). Figure 2 shows % misclassification when Personal blogs were left out of the training and test data sets.

These figures do not provide any discernible pattern, but it can be seen that the TF-IDF feature space performs better than the Term discrimination feature space for the full data-set. This behavior was expected because the Term-discrimination algorithm removes the sparse rows, and the top words are the most frequent words. Analysis of the top few words from the TF-IDF method and Term discrimination method reveal this effect. The top few words from TF-IDF based ranking are *wisdom, washing, waking, voices, viewing, tension, stare, someday, shifted*, while the top few words from Term discrimination based ranking are *like, just, time, people, u, india, good, life, movie, add*. The term discrimination based ranking ranks very frequent words, and can be used as an extended list of stop words in conjunction with the TF-IDF based approaches.

The patterns seem to be quite different for the classification exercise without the "Personal blogs". It is seen that a much smaller feature space captures more information in this case. This fits the observation that maximum words generated in the corpus were from the personal blogs. Term discrimination also seems to perform better because, it could find top few words from across different categories, as opposed to the previous case, when many of the top words were the most frequent words in the personal category. It is very surprising to see Approach-3 perform so poorly in Fig 2 as compared to Fig 1. We cannot comment on the stability of the scores generated in Approach-3 because of this fluctuating behavior of the feature generated by Approach-3.

Conclusions

This project deals with some issues in feature selection for classification of blogs and compares two such methods for feature selection. It was seen that the TF.IDF based approach

for feature selection perform better than term discrimination based approach.

Blog classification is an interesting exercise, and with lots of possible applications. The potential application of this type of classifier is a "blog suggester". The application can be built in the following way. A user submits her blog and each post of the user is categorized using the classifier. Then weightage to each category is given based on the classification of every individual post. The blog can now be represented in a $\underline{R}^8$ dimensional space of weightage to each score. Similar blogs can be searched for in this space (using distance, cosine similarity etc.) and the similar blogs suggested to the user.

The classifier built in this report may be used, but better classifiers can be potentially designed by using the "label" or "tag" feature in most common blogging platforms, as they provide information on the topic of the post.

References

Crammer, K., and Singer, Y. 2001. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research* 2:265–292.

Indie-bloggies 2009. `http://ibjury2008.indibloggies.org/`.

Python programing language. `http://python.org/`.

S.M.Rüger, and S.E.Gauch. 2000. Feature reduction for document clustering and classification. `http://www.doc.ic.ac.uk/research/technicalreports/2000/DTR00-8.pdf`.

SVM multiclass classifier. `http://www.cs.cornell.edu/People/tj/svm_light/svm_multiclass.html`.

Willett, P. 1985. An algorithm for the calculation of exact term discrimination values. *Information Processing and Management*.

Zhu, J. 2009. Class notes, CS-769. `http://pages.cs.wisc.edu/~jerryzhu/cs769.html`.