Text Classification using Support Vector Machine

Karuna P. Ukey
Department of IT
PRMIT & R, Badnera
Amravati, India

Dr. A.S. Alvi
Department of I.T.
PRMIT & R, Bandera
Amravati, India

Abstract— **Text categorization is the task of automatically sorting text documents into a set of predefined classes. Text categorization algorithms usually represent documents as bags of words and consequently have to deal with huge numbers of features. Semantic Analysis will be used for feature extraction, eliminating the text representation errors caused by synonyms and polysemes, and reducing the dimension of text vector.**

Keywords-bag of words;feature extraction;support vector machine.

I. INTRODUCTION

TC is the task of assigning documents expressed in natural language into one or more categories belonging to a predefined set. As more and more information is available on the internet, there is an ever growing interest in assisting people manages the huge amount of information. Information routing/filtering, identification of objectionable materials or junk mail, structured search/browsing, and topic identification, etc, these are all hot spots in current information management. The assignment of texts to some predefined categories based on their content, namely Text Categorization (TC), is an important component among these tasks

Text representation is a necessary procedure for text categorization tasks. Currently, bag of words (BOW) is the most widely used text representation method but it suffers from two drawbacks. First, the quantity of words is huge; second, it is not feasible to calculate the relationship between words. Semantic analysis (SA) techniques help BOW overcome these two drawbacks by interpreting words and documents in a space of concepts.one advantage that SVMs offer for TC is that dimensionality reduction is usually not needed, as SVMs tend to be fairly robust to overfitting and can scale up to considerable dimensionalities. Recent extensive experiments also indicate that feature selection tends to be detrimental to the performance of SVMs. For application developers, this interest is mainly due to the enormously increased need to handle larger and larger quantities of documents, a need emphasized by increased connectivity and availability of document bases of all types at all levels in the information chain. But this interest is also due to the fact that TC techniques have reached accuracy levels that rival the performance of trained professionals, and these accuracy levels can be achieved with high levels of efficiency on standard hardware / software resources. This means that more and more organizations are automating all their activities that can be cast as TC tasks.

II. PHASES IN LIFE CYCLE OF TEXT CLASSIFICATION

There are three phases in the life cycle of text classification. These are document indexing, classifier learning and classifier evaluation.

Document Indexing

Document indexing denotes the activity of mapping a document *dj* into a compact representation of its content that can be directly interpreted (i) by a classifier building algorithm and (ii) by a classifier, once it has been built. An indexing method is characterized by (i) a definition of what a term is, and (ii) a method to compute term weights. Concerning (i), the most frequent choice is to identify terms either with the *words* occurring in the document or with their *stems*. A popular choice is to add to the set of words or stems a set of *phrases*, i.e. longer (and semantically more significant) language units extracted from the text by shallow parsing and/or statistical techniques. Concerning (ii), term weights may be binary-valued or real-valued, depending on whether the classifier-building algorithm and the classifiers, once they have been built, require binary input or not. When weights are binary, these simply indicate presence/absence of the term in the document. When weights are non-binary, they are computed by either statistical or probabilistic techniques, the former being the most common option.

Classifier Learning

A text classifier for *ci* is automatically generated by a general inductive process (the *learner*) which, by observing the characteristics of a set of documents preclassified under *ci* or $\bar{c}i$, gleans the characteristics that a new unseen document should have in order to belong to *ci*. In order to build classifiers for *C*, one thus needs a set Ω of documents such that the value of $\Phi(dj, ci)$ is known for every $(dj, ci) \in \Omega \times C$.

Classifier Evaluation

Training efficiency (i.e. average time required to build a classifier $\hat{\Phi}i$ from a given corpus Ω), as well as *classification efficiency* (i.e. average time required to classify a document by means of $\hat{\Phi}i$), and *effectiveness* (i.e. average correctness of $\hat{\Phi}i$'s classification behaviour) are all legitimate measures of success for a learner.

In TC *research*, effectiveness is usually considered the most important criterion, since it is the most reliable one when it comes to experimentally comparing different learners or different TC methodologies, given that efficiency depends on too volatile parameters (e.g. different sw/hw platforms). In TC *applications*, however, all three parameters are important, and one must carefully look for a tradeoff among them, depending on the application constraints. For instance, in applications involving interaction with the user, a classifier with low classification efficiency is unsuitable.

III. SUPPORT VECTOR MACHINE

SVM is an effective technique for classifying high-dimensional data. Unlike the nearest neighbour classifier, SVM learns the optimal hyper plane that separates training examples from different classes by maximizing the classification margin. It is also applicable to data sets with nonlinear decision surfaces by employing a technique known as the kernel trick, which projects the input data to a higher dimensional feature space, where a linear separating hyperplane can be found. SVM avoids the costly similarity computation in high-dimensional feature space by using a surrogate kernel function. It is known that support vector machines (SVM) are capable of effectively processing feature vectors of some 10 000 dimensions, given that these are sparse. Several authors have shown, that support vector machines provide a fast and effective means for learning text classifyers from examples. Documents of a given topic could be identifyed with high accuracy

Support Vector Machine (SVM) is supervised learning method for classification to find out the linear separating hyperplane which maximize the margin, i.e., the optimal separating hyperplane (OSH) and maximizes the margin between the two data sets. An optimal SVM algorithm via multiple optimal strategies is developed in presented latest technique for documents classification. Among all the classification techniques SVM and Naïve Bayes has been recognized as one of the most effective and widely used text classification methods provide a comprehensive comparison of supervised machine learning methods for text classification.

One remarkable property of SVMs is that their ability to learn can be independent of the dimensionality of the feature space. SVMs measure the complexity of hypotheses based on the margin with which they separate the data, not the number of features. This means that we can generalize even in the presence of many features, if our data is separable with a wide margin using functions from the hypothesis space. The same margin argument also suggests a heuristic for selecting good parameter settings for the learner (like the kernel width in an RBF network). The best parameter setting is the one which produces the hypothesis with the lowest VC-Dimension. This allows fully automatic parameter tuning without expensive cross-validation.

Why Should SVMs Work Well for Text Categorization?

To find out what methods are promising for learning text classifiers, we should find out more about the properties of text.

High dimensional input space: When learning text classifiers, one has to deal with very many (more than 10000) features. Since SVMs use overfitting protection, which does not necessarily depend on the number of features, they have the potential to handle these large feature spaces.

Few irrelevant features: One way to avoid these high dimensional input spaces is to assume that most of the features are irrelevant. Feature selection tries to determine these irrelevant features. Unfortunately, in text categorization there are only very few irrelevant features. All features are ranked according to their (binary) information gain. Then a naive Bayes classifier is trained using only those features ranked 1-200, 201-500, 501- 1000, 1001-2000, 2001-4000, 4001-9962. A classifier using only that \worst" feature has a performance much better than random. Since it seems unlikely that all those features are completely redundant, this leads to the conjecture that a good classifier should combine many features (learn a \dense" concept) and that aggressive feature selection may result in a loss of information.

Document vectors are sparse: For each document, the corresponding document vector contains only few entries which are not zero [5][7].

IV. PROPOSED SYSTEM DESIGN

The work will be carried out as follows.

1. Analysis of available text classification systems.
2. Implementation of text pre-processor.
3. Feature extraction using semantic analysis.
4. Vectorization of text.
5. Finally classification of text using SVM classifier.
6. Comparison of system with already available systems.
7. Performance Evaluation and result analysis.

To find out what methods are promising for learning text classifiers, we should find out more about the properties of text.

High dimensional input space: When learning text classifiers on has to deal with very many (more than 10000) features. Since SVMs use overfitting protection which does

Text pre-processing

Texts are unstructured and use the natural language of humans, which make its semantics difficult for the computer to deal with. So they need necessary pre-processing. Text pre-processing mainly segments texts into words.

LSA-based feature extraction and dimensionality reduction

LSA is used in this module for the feature extraction and the dimensionality reduction of word-document matrix of training set. K largest singular values and corresponding singular vectors are extracted by the singular value decomposition of word-document matrix, to constitute a new matrix for approximately representation of the original word-document matrix. Compared with VSM, it can reflect the semantic link between words and the impact of contexts on word meanings, eliminate the discrepancy of text representation caused by synonyms and polysemes, and reduce the dimension of text vectors.

Vectorization of text

In this model, each row vector of the word-document matrix represents a text that is the vectorization of text. During a testing procedure, after each test sample segmented into words, the initial text vectors are mapped to a latent semantic space in this module by LSA vector space model, to generate new text vectors.

IHS-SVM classifier and learning

Finally, the new text vectors are classified in IHS-SVM classification module. IHS-SVM is an improvement for HS-SVM, both of which will use a minimum enclosing ball to define each type of text. When determining categories, HS-SVM finds which hyper-sphere is the closest one to the test sample, and then the category it stands for is the one the test sample belongs to. However, the texts in overlapping regions cannot be classified correctly by this way. IHS-SVM divides samples into three types: those not in any hyper-sphere, those only contained in one, and those included in multiples. The classification of the first two types is same to HS-SVM. It compares the concentration of the test sample to each hyper-sphere, and then classes the sample to the highest one.

Feature Extraction and Dimensionality Reduction

The process of feature extraction is to make clear the border of each language structure and to eliminate as much as possible the language dependent factors, tokenization, stop words removal, and stemming. Feature Extraction is fist step of pre processing which is used to presents the text documents into clear word format. Removing stops words and stemming words is the pre-processing tasks. The documents in text classification are represented by a great amount of feature and most of then could be irrelevant or noisy. Dimension reduction is the exclusion of a large number of keywords, base preferably on a statistical criterision, to create a low dimension vector. Dimension Reduction techniques have attached much attention recently science effective dimension reduction make the learning task such as classification more efficient and save more storage space. Commonly the steeps taken please for the feature extractions are: Tokenization: A document is treated as a string and then partitioned into a list of tokens. Removing stop words: Stop words such as “the”, “a”, “and”… etc are frequently occurring, so the insignificant words need to be removed. Stemming word: Applying the stemming algorithm that converts different word form into similar canonical form. This step is the process of conflating tokens to their root form eg. Connection to connect, computing to compute etc.

VSM based on text keywords quantizes document vector with the weights of the words, having high efficiency and easy to use. However, it only counts the frequency of the words, while ignoring the semantic link among them and the impact of context on their meanings. Thus texts similarity depends only on the number of the same words they contained, which reduces the classification accuracy with the existence of polysemes and synonyms. In addition, the text matrixes constructed by VSM are generally high-dimensional sparse matrices, inefficient in training and classification and not suitable for handling large-scale text sets. However, LSA can effectively solve these limitations. It believes that there is a latent semantic structure between words of one text. And it hides in their context usage patterns. So, k largest singular values and their corresponding singular vectors are extracted by the singular value decomposition of word-document matrix, to constitute a new matrix for the approximate presentation of word-document matrix of the original documents set. Text presented by high-dimensional VSM is thus mapped into a low-dimensional latent semantic space. You can extract latent semantic structure without the impact of the correlation between the words to get high text representation accuracy. LSA is based on singular value decomposition. It maps texts and words form a high-dimensional vector space to a low one, reducing text dimensions and improving text representation accuracy.

Step1: Construct a word-document matrix A. In the LSA model, a text set can be expressed as a word-document matrix of $m \times n$ (m is the number of entries contained in a text, n is the number of texts).

Step2: Decompose singular value. A is decomposed into the mutiply of three matrices: U ‘, S ‘, V ‘. U ' and V ' are orthogonal matrices, S ' is a diagonal matrix of singular value. Retain the rows and the columns of S ' containing K largest single-values to get a new diagonal matrix. Then retain the same part of U ' and V ' to get U and V. Thus, construct a new word-document matrix $R = USV\ T$. For a text d, words are screened by singular value decomposition to form new vectors to replace the original text feature vectors. It ignores the factors of smaller influence and less importance. Key-words that don’t appear in the text will be represented in the new word-document matrix if they are associated with the text semantics. Therefore, the new matrix reflects the potential semantic relation among keywords from a numerical point of view. It is closest to the original term frequency matrix with the least-squares. Meaning of each dimension in vector space is greatly changed in process. It reflects a strengthened semantic relationship instead of simple appearance frequency and distribution relationship of entries. And the dimension reduction of vector space can effectively improve the classification speed of text sets.

CONCLUSION

It can be possible to develop a text classification system using support vector machine and semantic analysis for documents. By using the support vector machine and semantic analysis the system can give more accurate result.

Automated text classification is attractive because it frees organizations from the need of manually organizing document bases, which can be too expensive, or simply not feasible given the time constraints of the application or the number of documents involved. The accuracy of modern text classification systems rivals that of trained human professionals.

ACKNOWLEDGMENT

First & foremost, I would like to express my sincere gratitude towards my project guide, P**rof. A. S. Alvi** for his valuable guidance, encouragement, and optimism. I feel proud of his eminence and vast knowledge, which will guide me throughout my life.

REFERENCES

[1] Fabrizio Sebastiani, Text Categorization; In Alessandro Zanasied. Text Mining and Its Application.Southampton: WIT Press, pp.109-129, 2005.

[2] Evgeniy Gabrilovich, Shaul Markovitch. Text Categorization with Many Redundant Features: Using Aggressive Feature Selection to Make SVMs Competitive with C4.5. Proceedings of the 21 st International Conference on Machine Learning, Ban®, Canada, 2004.

[3]] C.H.Li, An efficient document categorization model based on LSA and BPNN, Sixth International Conference on ALPIT, pp.9-14, 2007

[4] An Overview of E-Documents Classification Aurangzeb Khan. , Baharum B. Bahurdin, Khairullah Khan Department of Computer & Information Science Universiti Teknologi, PETRONAS 2009 International Conference on Machine Learning and Computing IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore

[5] Text Categorization with Support Vector Machines: Learning with Many Relevant Features Thorsten Joachims Universit • at Dortmund Informatik LS8, Baroper Str. 301 44221 Dortmund, Germany.

[6]] Research of Text Classification Model Based on Latent Semantic Analysis and Improved HS-SVM Yu-feng Zhang Center for Studies of Information Resources Wuhan University Wuhan, China.

[7] Efficient Algorithm for Localized Support Vector Machine Haibin Cheng, Pang-Ning Tan, Member, IEEE, and Rong Jin, Member, IEEE

[8] A Graph-Based Approach for Multi-Folder Email Classification,Sharma Chakravarthy

[9] Department Of Comp. Sci. & Engg. Univ. of Texas at Arlington Arlington, TX, USA sharmac@uta.edu