Document Classification Using Topic Modelling

T Yashwanth Reddy ,P Revanth Rathan

***Abstract*—Most text categorization algorithms represent a doc ument collection as a Bag of Words (BOW).The BOW representation is unable to recognize synonyms from a given term set and unable to recognize semantic relationships between terms. In this approach, we appl y the topic-model approach to cluster the words int o a set of topics. Words assigned into the same topic a re semantically related. Our main goal is to compar e between the feature processing techniques of BOW an d the topic model. We also apply and compare between two Topic modelling techniques: Latent semantic analysis(LSA) and Latent Dirichlet analysis(LDA). Two text categorization algorithms: Decision Tree(DT), Suppor t Vector Machines (SVM), are used for evaluation. The experimental results showe d that the topic-model approach for representing th e documents yielded the best performance equal to 100with the LDA Topic selection technique.**

***Keywords*—LSA,LDA, SVM, Decision Tree.**

I. Introduction

The amount of text documents is increasing with an explosive rate. Text categorization has become one of the key techniques for managing and organizing those documents and also assists the information retrieval process in filtering the documents for a specific topic. Text categorization process usually adopts the supervised machine learning algorithms for learning the classification model . To prepare the term feature set, the bag of words (BOW) is usually applied to represent the feature space. Under the BOW model, each document is represented by a vector of weight values calculated from, for example, the term frequency inverse document frequency (TF-IDF), of a term occurring in the document. The BOW is very simple to create, however, this approach discards the semantic information of the terms (i.e., synonym). Therefore, different terms whose meanings are similar or the same would be represented as different features. As a result, the performance of a classification model learned by using the BOW model could become deteriorated. In this approach, we apply the topic model to cluster the words into a set of topics. The concept of topic model, the words (or terms) are clustered into the same topics. Given D is a set of documents composed of a set of words (or terms) W , T is a set of latent topics, that are created based on a statistical inference on the term set W.

In the approach, the topic model is applied based on the Latent Dirichlet Allocation (LDA) algorithm to produce a probabilistic topic model and Latent semantic analysis(LSA) from a BBC dataset. The topic model can help capture the synonyms, hypernyms and hyponyms of a given word such as the words human (hypernym) and John (hyponym) would be clustered into the same topic.

II. Data Set

We considered Dataset of BBC Documents which consist of mainly consist of five categories namely BUSINESS ,SPORTS ,TECH ,POLITICS ,ENTERTAINMENT . And each category contains roughly 500 document and total summing to nearly 2500 documents.

III. Pre-Processing Data

We processed data by removing all stop word using nltk python libraries which contians 500 stop word and then converted to Bag of words(BOW) representation ,we removed words that occurs only once in given document then we applied TF-IDF(Term frequency Inverse Document frequency).

IV. Topic modelling

we have used two topic modelling techniques for comparing namely

- Latent semantic analysis
- Latent Dirichlet analysis

A. Latent Semantic analysis

Latent Semantic analysis (Deerwester et al.,1990) tries to find a lower-dimensional subspace that takes into account association between terms and documents. The method applies SVD to estimate such subspace. Let X be a term-document matrix represented by a vector space model,X can be decomposed into the product of three matrices: X= TSD

T where T;D are orthonormal matrices and S is a diagonal matrix whose diagonal components cor- responds to singular values ordered in decreasing order. LSI approximates term-document matrix using bases that correspond to largest singular values. Let ^{S} be a matrix whose all but k-largest diagonal elements in S were set to zero, term-document matrix can be approximated as

X=TSDT Resulted matrix ^{X} is an optimal approximation of X in terms of mean-square error. LSI assumes that there is an underlying latent semantic structure in the word usage across documents, and the basic idea is that such structure can be uncovered by dropping small singular values in S . It has been empirically shown that LSI improves information retrieval performance (Deerwester et al., 1990; Dumais, 1995), and its effectiveness has been ex-plained theoretically by means of the Bayesian regression model (Story, 1996).

B. Latent Dirichlet

Latent Dirichlet Allocation (LDA) is a generative probabilistic model for a set of documents.The concept of this approach is that documents are represented as random mixtures over latent topics. Each topic is represented by a probability

distribution over the terms. Each article is represented by a probability distribution over the topics. LDA has also been applied for identification of topics in a number of different areas such as collaborative filtering [3], content-based filtering [7],[8] and classification [9-13]. In the generation process of LDA, a word w is generated from combination of topics z in a document d and sampling a word from topic-word distribution

V. Classification Techniques

In our approach we considered two classification techniques

- Support vector machines(SVM)
- Decision Tree

A. Multi-class Support Vector Machine

We then train a support vector machine. In [1], Crammer and Singer makes a derivation for a multi-class SVM which we are using.

Lets replace ourselves in the context of the course but with k classes this time. Let $(x^{(i)}, y^{(i)}), i = 1...m$ with $x^{(i)} \epsilon R^n$ but $y^{(i)} \epsilon \{1...k\}$. Crammer and Singer (2002) proposed the following multi-class approach by solving the following optimization problem:

$$\begin{aligned} \underset{w_1,\xi_i,l=1\ldots k,i=1\ldots m}{\text{minimize}} \quad & \frac{1}{2}\sum_{l=1}^{k} w_l^T w_l + C\sum_{i=1}^{m}\xi_i \\ \text{subject to} \quad & w_{y^{(i)}}^T x^{(i)} - w_l^T x^{(i)} \leq \delta_{i,l} - \xi_i,\ i = 1,\ldots,m. \end{aligned}$$

With the following decision function:

$$h_{w_1,\ldots,w_k}(x) = \underset{l\in\{1,\ldots,k\}}{\operatorname{argmax}}\, w_l^T x$$

We can derive the dual problem:

$$\begin{aligned} \underset{\alpha_i^l,l=1\ldots k,i=1\ldots m}{\text{minimize}} \quad & \sum_{1\leq i,j\leq m,1\leq l\leq k} \alpha_i^l \alpha_j^l \langle x^{(i)}, x^{(j)}\rangle + \sum_{1\leq i\leq m,1\leq l\leq} \\ \text{subject to} \quad & \sum_{l=1}^{k} \alpha_i^l,\ i = 1,\ldots,m. \\ & \alpha_i^l \leq \delta_{i,l} C, \forall i \in \{1,\ldots,m\}, \forall l \in \{1,\ldots k\} \end{aligned}$$

We can then solve the dual using the coordinate ascent method. It is clear that this extension from 2 classes to k classes works in the exact same way than before. We used LIBLINEAR SVM to implement this multi-class SVM.

B. Decision Tree

Decision tree is a classifier in the form of a tree structure, where each node is either:

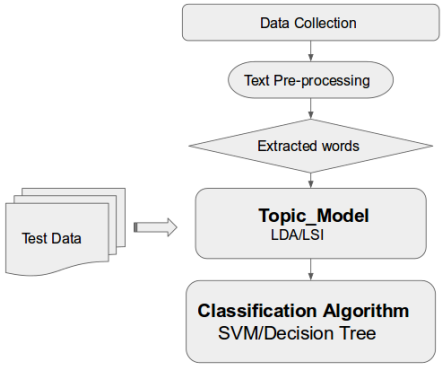
- a leaf node - indicates the value of the target attribute (class) of examples, or
- a decision node - specifies some test to be carried out on a single attribute-value, with one branch and sub-tree for each possible outcome of the test.

A decision tree can be used to classify an example by starting at the root of the tree and moving through it until a leaf node, which provides the classification of the instance. Decision tree induction is a typical inductive approach to learn knowledge on classification. The key requirements to do mining with decision trees are:

- Attribute-value description: object or case must be expressible in terms of a fixed collection of properties or attributes. This means that we need to discretize continuous attributes, or this must have been provided in the algorithm.
- Predefined classes (target attribute values): The categories to which examples are to be assigned must have been established beforehand (supervised data).
- Discrete classes: A case does or does not belong to a particular class, and there must be more cases than classes.
- Sufficient data: Usually hundreds or even thousands of training cases

VI. Approach Overiview

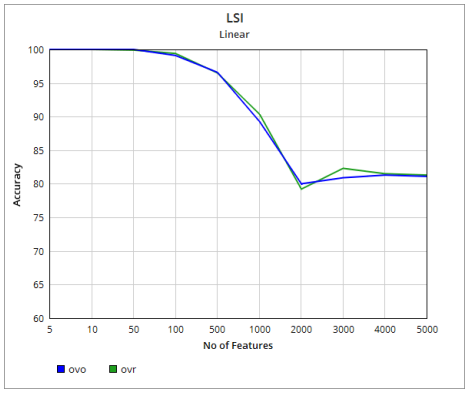
we have first pre processed data and converted into BOW representation ,then we have applied TFIDF to corpus which is then input to our topic model which is in our case both Latent semantic analysis and Latent Dirichlet analysis after training topic model we predicted the test document Topics using the above trained topic model which is then supplied to classification algorithm SVM and Decision Tree



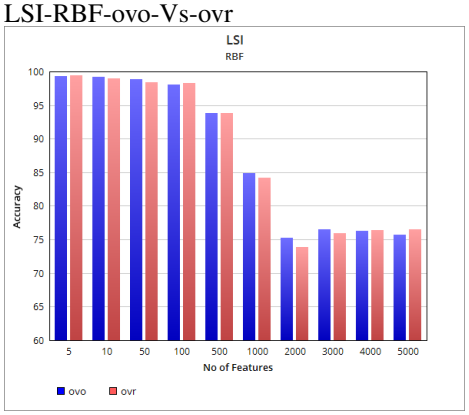
VII. Results

At first phase we have selected LSI from topic model and SVM for classification and run experiment with different topic size in topic model and different kernels in from SVM the following graphs show the performance of different combinations

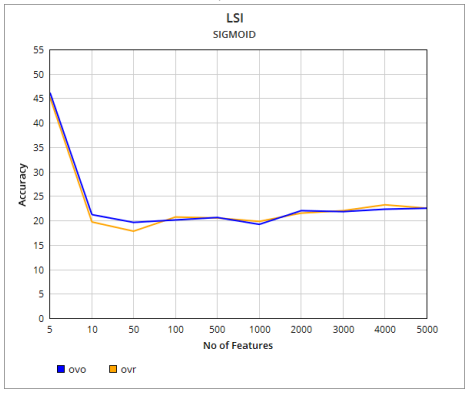
LSI-Linear(kernel)-OVO-vs-OVR



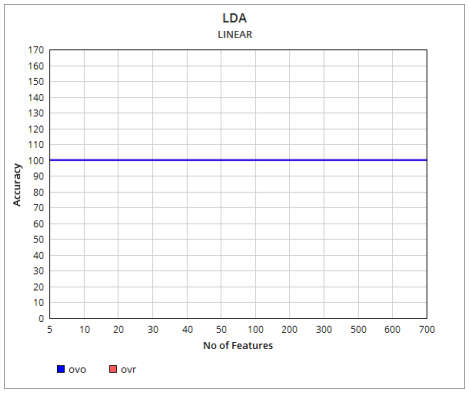
LSI-RBF-ovo-Vs-ovr



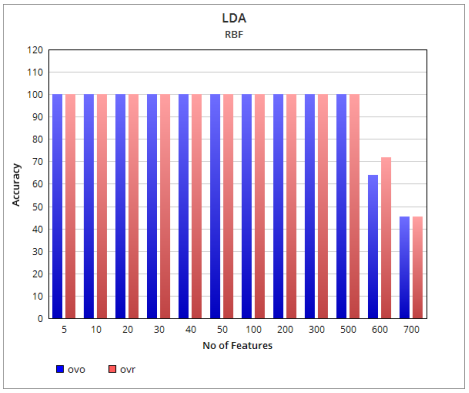
LSI-SIGMOID-ovo$_V s - OVR$



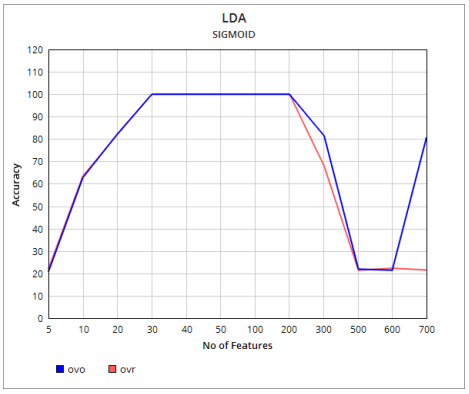
LDA-Linear(kernel)-OVO-vs-OVR



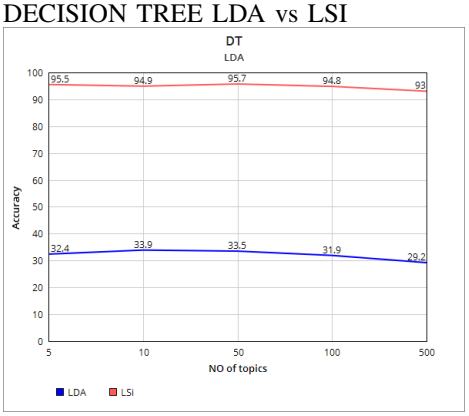
LDA-RBF-ovo-Vs-ovr



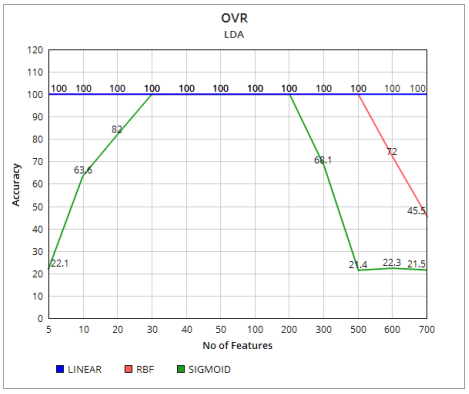
LDA-SIGMOID-ovo$_V s - OVR$



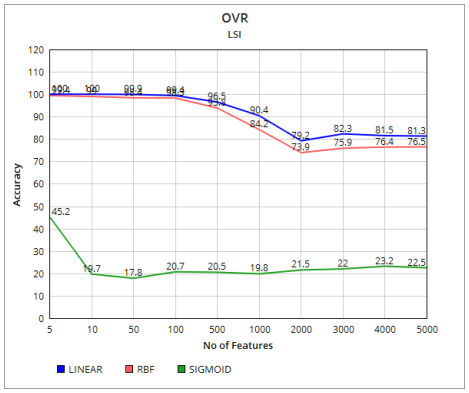
DECISION TREE LDA vs LSI



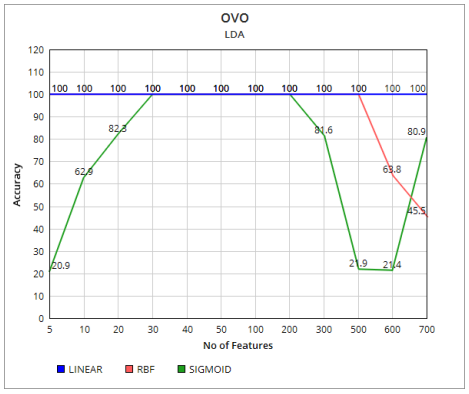
OVR-over different Kernels for LDA



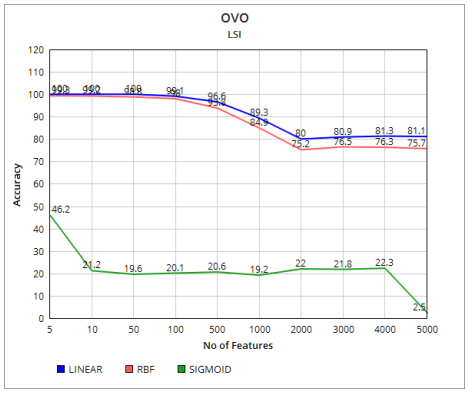
OVR-over different Kernels for LSA



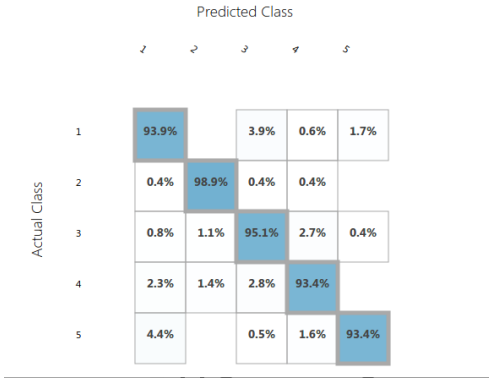
OVO-over different Kernels for LDA



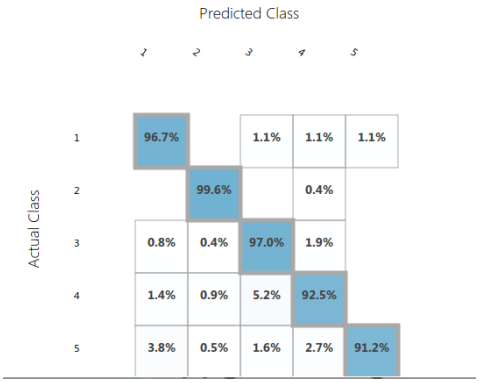
OVO-over different Kernels for LSA



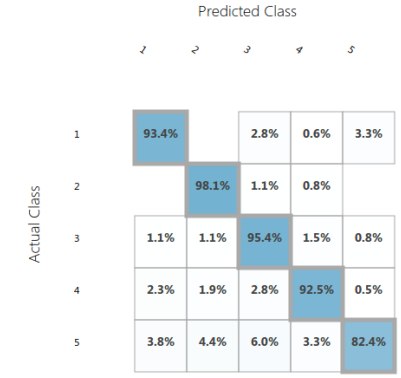
Decision Tree confusion matrix for topic size 5(LSI)



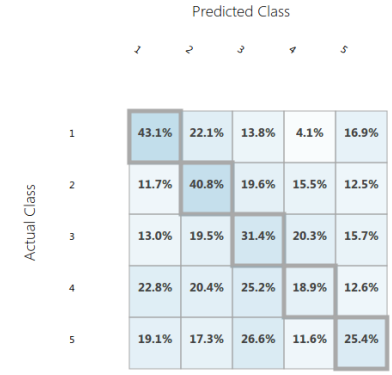
Decision Tree confusion matrix for topic size 50(LSI)



Decision Tree confusion matrix for topic size 500(LSI)

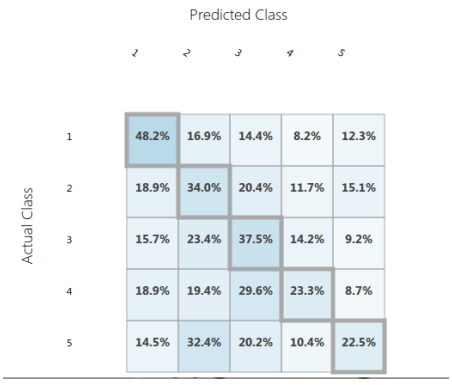


Decision Tree confusion matrix for topic size 5(LDA)

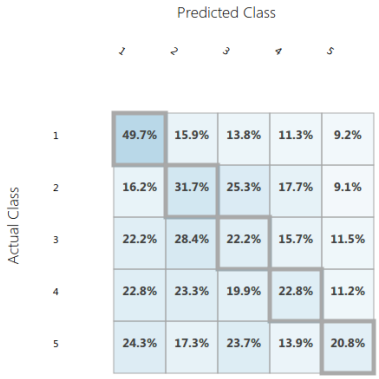


Decision Tree confusion matrix for topic size 50(LDA)

Decision Tree confusion matrix for topic size 50(LSI)



Decision Tree confusion matrix for topic size 500(LDA)



VIII. CONCLUSION

In every case SVM out performed Decision tree in both LSI and LDA topic modelling in some case where topic size is five(5) svm attained 100 accuracy.

In SVM mostly linear and RBF kernels gave better accuracy than sigmoid kernel.

Decision Tree gave comparable results w.r.t SVM in case of LSI but in case of LDA Decision tree failed where it gave accuracy about 40 where svm gave 95 accuracy.

LDA gave better results than LSI