Norwegian Artificial Intelligence Symposium, Gjøvik, 22 November 2010

The Use of Game Theory in Feature Selection

Gleb Sizov, Pinar Öztürk, Kjetil Valle
Department of Computer Sciences
Norwegian University of Science and Technology
Trondheim, Norway

Abstract—**This paper investigates the use of the Shapley value, a game theory notion, for feature selection in document classification. Feature selection in statistical natural language processing and some of the related theoretical issues from game theory are discussed. We propose a 2-stage feature selection algorithm that utilizes an approximation of the Shapley value to compute the importance of features in coalition with other features. Several optimizations are introduced into the algorithm to reduce the computational cost of the approach. Our algorithm demonstrates superior performance compared to the baseline algorithm that relies on term frequency-inverse document frequency (TF-IDF) weighting scheme.**

1. Introduction

Feature selection is an important issue in human cognitive processes and consequently in artificial intelligence. Depending on the peculiarities of the task, whether it is categorization of plants or retrieval of documents, only a subset of all available features would be relevant. For example, the height of a student is hardly a good predictor of the progress in their studies. Determination of the relevant feature subset is called feature selection.

The most important reasons for conducting feature selection are:

- Size of the original feature set might be very large leading to high storage and computational costs.
- Sparseness of data in a high dimensional feature space may degrade the performance of similarity-based algorithms.
- Irrelevant features introduce noise which may have aversive effects on the performance of the learning algorithm.
- Large number of features makes it difficult to visualize, analyze and understand data.

Hence, the goal of feature selection is to reduce the feature space while increasing the performance, through eliminating the noisy and redundant features. Feature selection has been investigated in various text related tasks such as information retrieval [5], automatic text summarization [11], word sense disambiguation [9] and text categorization [4], [13]. The majority of predominantly used feature selection techniques rely on statistical methods that exploit word frequencies in documents and classes.

This paper follows a different approach and relies on the notion of the Shapley value for feature selection. Originating from game theory, Shapley value serves as a means to determine a fair allocation of collectively gained profits between collaborating players [12].

In this paper we investigate feature selection in the context of classification task, more precisely, document classification. In a classification task, typically a classifier is learned on the basis of a data set, e.g., a document collection in text categorization, which is represented in terms of relevant features. Once a classifier is learned, it can predict the category of a previously unseen text. The main objective is to find the feature subset that provides the best classification accuracy.

The key rationale behind using the Shapley value for this purpose is that it makes it possible to compute how the performance of a classifier is affected by the presence of a certain feature. Unlike most of the commonly used feature selection approaches, the Shapley value allows the evaluation of features based on their performance in coalitions with other features, rather than based on individual measures. However, when the original feature set is very large the use of Shapley values is a costly method. In this paper we propose and test a Shapley-based, 2-stage feature selection algorithm. In order to reduce the computational complexity the majority of features is eliminated by a simple method during the first stage and then a Shapley-based method is used to discover the relevance of the remaining features. The results obtained by the proposed algorithm are compared to the method based on term frequency-inverse document frequency (TF-IDF) weights, which we use as the baseline.

The rest of the paper is organized as follows. In the

next section the problem of feature selection in natural language processing is discussed. Section 3 introduces background concepts directly related to Shapley value and its adaptation for feature selection. In section 4 the proposed feature selection algorithm is described. Section 5 provides the details of our experiments. The results of the experiments are presented in section 6. Section 7 wraps up the conclusions and the future research directions.

2. Feature Selection in Natural Language Processing

Natural language processing (NLP) makes extensive use of machine learning techniques. Feature selection plays a major role in NLP because language representation models tend to be of high dimensionality due to the richness and flexibility of natural language. The most common representation model used in statistical NLP is the vector space model (VSM), where a document or some other textual unit is represented by a vector of which elements contain the frequency of terms in the document. A simple VSM representation that is often utilized in document classification and information retrieval is a term-document matrix where rows correspond to terms and columns to documents. An element in this matrix represents the frequency of a term (e.g., word) in the corresponding document. Terms in the described representation constitute features. Original term-document matrix contains rows for all the unique words (often after stop word removal) in a corpus and might be quite large. For example, the well-known Reuters-21578 corpus contains 21578 documents and 55377 unique words. After stop words removal and stemming this number is reduced to 23706 which still produces a huge term-document matrix. Such feature space representations lead to sparse matrices where only a small percentage of elements have non-zero values. This property of the representation is described by Zipf's law stating that a very small percentage of words accounts for a large document collection. Feature-space and the sparsity can be reduced through feature selection and extraction techniques.

There are two types of feature selection techniques utilized in NLP: task-specific and task-free. Task-free methods select informative words without considering the task at hand or using task specific information.

Among task-free techniques are preprocessing routines such as stop words removal, stemming and weighting schemes such as document frequency thresholding (DF) and term strength (TS). DF feature selection simply ranks the features based on the number of documents they occur in. Then all the features below some predefined threshold are discarded. TS feature selection is based on the assumption that terms that frequently occur in similar documents are informative and thus should be retained. Document similarity is measured using the cosine of two document vectors.

Task-specific methods such as information gain (IG), mutual information (MI) and χ^2 statistics (CHI) make use of additional information such as class labels for pre-classified documents. IG is the amount of information (measured in bits) provided by a feature for prediction of a document category. It is calculated as the change in information entropy before and after introduction of a certain feature. MI measures the association or dependence between the value of a feature and a document category. Higher association indicates more informative features. CHI statistics is similar to MI in a way that it measures the dependence between document classes and terms. The major difference however is that CHI provides a normalized value which makes term values comparable for the same category.

A comparative analysis of the mentioned feature selection techniques in text categorization was presented in [13]. The obtained results indicate that IG, CHI and DF achieve the best performance. Among these three techniques DF is the simplest and the computationally least expensive one. It has also the advantage of being task-free. DF will, therefore, be used further in this paper to compare it with our baseline algorithm.

3. The Shapley Value

The Shapley value has been shown to be promising as a feature selection mechanism in document classification [3], [8]. This section introduces some concepts from game theory that relates to the Shapley value, and describes their relevance to feature selection for document classification.

In cooperative games players form coalitions to obtain a high total profit. Formally a coalition game is defined by a set of players N of size n and a payoff function $v(S)$ that assigns the total payoff for a coalition S, which is a subset of players from N. The payoff function $v(S)$ has the following properties:

1) The total payoff for empty coalition is null: $v(\emptyset) = 0$
2) The payoff function is superadditive: $v(S \cup T) \geq v(S) + v(T)$, where $S \cap T = \emptyset$

Contributions of players in the coalition varies since the role and the importance of each player is often different. Shapley value is a measure used to determine the payoff of each individual player according to its contribution to the total payoff gained by the coalition. For example, Shapley value could be used to determine the prices or values of football players. During several seasons football players can be transfered from one team to another. In addition, composition of football teams may change from match to match. This way players become members of different teams (i.e., coalitions) and thus contribute to the final score in each match. Based

on these scores the Shapley value might be used to determine fair prices of football players according to their contributions over several seasons.

Formally, marginal contribution Δ_i of a player i to a coalition S is defined as follows:

$$\Delta_i(S) = v(S \cup \{i\}) - v(S) \tag{1}$$

Then, the Shapley value Φ_i for a player i is defined as a mean of marginal contributions to all possible coalitions of players in N

$$\Phi_i = \frac{1}{n!} \sum_{\pi \in \Pi} \Delta_i(S_i(\pi)) \tag{2}$$

where Π is the set of permutations over N and $S_i(\pi)$ is a set of players from π that appear before player i in the permutation.

The feature selection problem can be easily represented in terms of a coalition game where features cooperate to achieve optimal performance in a certain task such as document classification. This way a set N contains all the features and $v(S)$ is the accuracy achieved by the classifier using a subset of features S.

Evaluation of features using the Shapley value requires testing on all possible subsets of features. Considering that the number of features n could reach thousands, using all $n!$ subsets is not computationally tractable. Instead, an approximation of the Shapley value must be used. An approximation proposed by Keinan et al. [7] utilizes uniformly sampled feature subsets instead of the full set of subsets. The number of sampled subsets is assumed to be much less than $n!$ but still large enough to provide a robust estimation. In the original version of the estimator the size of the permutations equals the number of features n. However, as reasonably proposed by Cohen et al. [3] the size of permutations could be bounded to some constant value that is smaller than n because the number of significant interactions between features is much smaller than the number of features. Further in this paper such permutations will be referred to as d-bounded permutations, where d is the size of a permutation. Formally, approximation of the Shapley value using d-bounded permutations is defined as

$$\phi_i = \frac{1}{|\Pi_d|} \sum_{\pi \in \Pi_d} \Delta_i(S_i(\pi)) \tag{3}$$

where Π_d is a set of d-bounded permutations. The value of d accounts for the number of interactions considered in evaluation of features. When $d = 1$, features are evaluated individually and interactions between features are ignored completely. In the original paper on feature selection based on coalition game by Cohen et al. d is set to $\sqrt{n}$, which is borrowed from the equivalent value in feature selection by random forests [1].

4. Feature Selection Algorithm

Feature selection might be viewed as a coalition game where features cooperate to obtain optimal classification accuracy. This section presents a two-stage feature selection algorithm based on the approximation of the Shapley value to select the most important features.

The inputs to the algorithm are a dataset represented by a term-document matrix of which entries are TF-IDF values, and a list of all the documents coupled with the corresponding class labels. The algorithm consist of two stages:

1. stage:
   Reduce the number of features on the basis of TF-IDF weights.
2. stage:
   Use Shapley values to select a feature subset of the targeted size from the reduced set of features from stage 1.

The motivation behind this two-stage approach is to reduce the computational cost of calculating the Shapley value for a large set of features. The first stage effectively eliminates the major part of features before the second stage, which is much more computationally expensive. The number of features selected during the first stage is approximately 5 times larger then the number of features selected in the second stage.

The first stage utilizes TF-IDF weights, which is a classical measure of term importance in automatic text analysis [10]. It is computed as the product of a document frequency $\mathrm{tf}_{t,d}$ of a term t in a document d and inverse document frequency idf_t:

$$\text{tf-idf}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t \tag{4}$$

$$\mathrm{tf}_{t,d} = \frac{n_{t,d}}{|d|} \tag{5}$$

$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t} \tag{6}$$

where $n_{t,d}$ is the number of times t appears in d, $|d|$ is the total number of words in d, df_t is the number of documents that contain t and N is the total number of documents in the collection. According to this definition, terms that appear frequently in few documents get high TF-IDF values and are considered more important than terms that appear in many of the documents and consequently get low values. Based on this intuition the first stage of our feature selection algorithm includes the following steps:

1) TF-IDF weights are calculated for each cell in the term-document matrix.
2) The TF-IDF weights are summed up for each term (all entries in a row).
3) The resulting values are sorted in descending order and the predefined number of terms with the highest values are selected.

The algorithm for evaluation of term features in the second stage is our implementation of the Shapley value approximation, which is slightly different from the approximation described in section 3. Our algorithm utilizes randomly sampled feature subsets of the predefined size m instead of d-bounded permutations. Unlike the approximation defined by equation 3, in our version all the subsets evaluated by the algorithm are of the same size m, which is equal to the targeted number of features. The underlying assumption is that this approximation of feature contributions is more appropriate because it focuses on the feature subsets that have the same number of features we want to finally select and hence, deals with subsets that are more likely to capture the feature interactions we are most interested in.

for all $i \in I_1$ **do**
  for all $S \in$ {random subsets of size m sampled from I_1/i} **do**
    $\Delta_i \leftarrow v(S \cup \{i\}) - v(S)$
    $\Phi_i \leftarrow \Phi_i + \Delta_i$
  end for
end for

where I_1 is a set of features selected in the first stage and the payoff function $v(S)$ is the calculated accuracy of the the classifier based on features in S. The following routine calculates $v(S)$:

1) Construct the term-document matrix using term features in S.
2) Split the matrix into training and testing parts corresponding to 60% and 40% of the documents in the dataset.
3) Train a classifier on training documents.
4) Return the accuracy $v(S)$ achieved by the classifier on testing documents.

The accuracy of the classifier is calculated as the fraction of labels assigned by the classifier that are the same as the corresponding gold standard labels.

$$v(S) = \frac{|\{\text{correctly classified testing documents}\}|}{|\{\text{testing documents}\}|} \quad (7)$$

After the contributions for features in I_1 are obtained, the terms with the highest values are selected as features.

5. Experiments

This section describes the experiments we conducted in order to test the algorithm described in section 4. The algorithm is tested on two datasets constructed from the well-known Reuters-21578 text categorization test collection [2]. The datasets are of different sizes and contain documents from different number of classes. The details for the datasets are shown in table I.

The following preprocessing routines have been employed:

Name	Size	Classes	Features
Reuters 1	232	5	4388
Reuters 2	1000	10	8798

TABLE I
DATASETS USED IN THE EXPERIMENTS.

1) Removal of punctuations.
2) Stop words removal.
3) Removal of terms that occur less than 4 times in the dataset or in less than 2 documents.

The quality of the selected feature subset, hence feature selection, is determined by the accuracy achieved by the classification algorithm. The classification algorithm used in the experiments is the k-nearest neighbors (k-NN) with $k = 1$. The similarity between document vectors is calculated using standard cosine similarity metric. Each dataset have been split into a training part and a testing part where the training part contains 60% of the documents in the dataset and the rest is used for testing. The accuracy on the testing set is obtained by taking the average of the accuracies in k-fold cross-validation with $k = 15$.

Feature selection based on the estimation of the Shapley value is not a fully deterministic process because it involves randomly sampled subsets of features. Therefore, features selected in different runs of the algorithm may vary, which consequently leads to differences in the obtained accuracies. To account for this effect the feature selection process is executed multiple times and the average of the accuracies is taken. The number of samples used for the estimation of the Shapley value was set to 500, which proved to provide consistent results over several runs. We have repeated the experiment for different number of selected features. The value of m varied accordingly.

In order to optimize the performance of the Shapley approximation in the second stage of our algorithm, the k-NN classifier utilized by the payoff routine (see section 4) was using class centroids calculated from the training instances instead of instances themselves. Given that the number of classes is much less than the number of training documents, the centroid-based classifier is faster, which is crucial for estimation of the Shapley value. Besides, there is some evidence that centroid-based classification may achieve better results than k-NN [6].

The results obtained by the Shapley-based feature selection algorithm are compared with the performance of the baseline, which is based on TF-IDF weights. The algorithm for the baseline is the same as the first stage of the algorithm described in section 4 but selects the targeted number of features. In the paper by Yang et al. [13] the document frequency (DF) algorithm demonstrated reasonable performance, on par with the more complex feature selection algorithms commonly used in

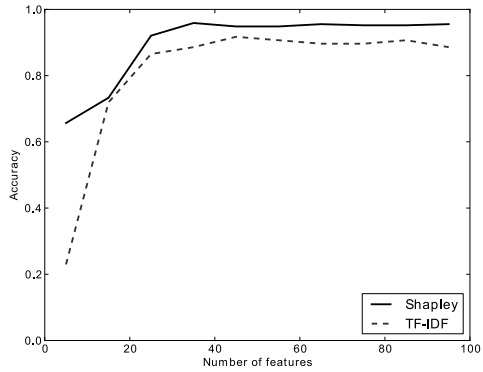

Fig. 1. Accuracy achieved on Reuters 1 dataset with gradual increase in the number of features from 5 to 95.

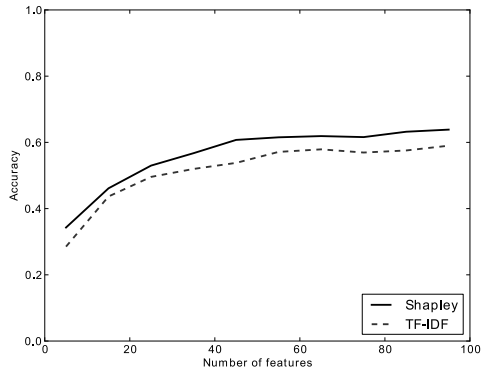

Fig. 2. Accuracy achieved on Reuters 2 dataset with gradual increase in the number of features from 5 to 95.

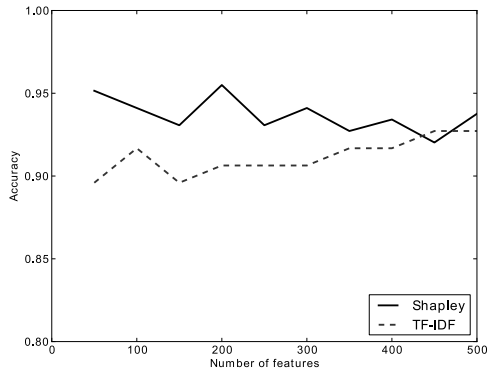

Fig. 3. Accuracy achieved on Reuters 1 dataset with gradual increase in the number of features from 50 to 500.

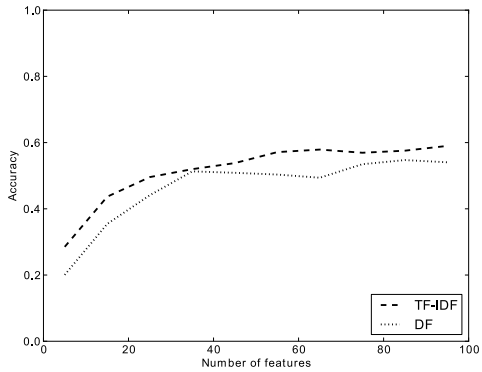

Fig. 4. Comparison between TF-IDF and DF feature selection on Reuters 2.

Algorithm	Reuters 1	Reuters 2
No feature selection	0.968 (4388 features)	0.601 (8798 features)
TF-IDF baseline	0.885 (95 features)	0.589 (95 features)
Shapley	0.955 (95 features)	0.638 (95 features)

TABLE II
CLASSIFICATION ACCURACY RATES.

classification. In our tests TF-IDF demonstrated even better results (see figure 4). Considering its simplicity, TF-IDF is a reasonable choice for the baseline feature selection algorithm so we chose it as the baseline.

6. Results

The accuracies obtained in the experiments are shown in table II. For both baseline and Shapley-based feature selection the top 95 features were selected and used for classification. Shapley-based feature selection demonstrated 7% and 4.9% performance improvement over the baseline algorithm for Reuters 1 and Reuters 2 datasets accordingly. For Reuters 2, which is a more complex dataset compared to Reuters 1, the Shapley-based algorithm performed 3.7% better than the classifier with the full set of features.

Figures 1 and 2 demonstrate the performance of two-stage Shapley-based feature selection compared to the baseline algorithm for different number of selected features. For both datasets Shapley provides better accuracies for the number of features from 5 to 95. Further increase in the number of features gradually eliminates the difference in performance. Figure 3 shows the performance of the algorithms with the number of features increased to 500. In addition, the results obtained for the large number of feature seems to be less consistent, which is probably due to the fact that the number of sampled feature subsets is insufficient relative to the total number of possible subsets of a larger size m.

7. Conclusion

The notion of Shapley value from game theory provides theoretical foundation for evaluation of agents in coalition game. This approach can be transfered to the problem of feature selection in text classification. The Shapley-based feature selection algorithm proposed in this paper has demonstrated its superior performance compared to the baseline algorithm on two datasets.

Although the experimental results demonstrated the benefits of utilizing the Shapley-value for feature selection, it is important to consider the computational costs associated with it. Each feature is evaluated in coalition with randomly sampled subsets of features, where the number of sampled subsets should be large enough for a robust evaluation. Evaluation of each feature involves training and testing of a classifier multiple times, which may be computationally expensive especially with complex classifiers. To overcome this problem the algorithm proposed in our paper does two-stage feature selection where the majority of features are discharged by computationally effective methods during the first stage. Another workaround is to simplify the method for evaluation of coalitions. In our experiments we used class centroids-based classifier instead of k-NN. With these optimizations applied, the algorithm was able to complete in reasonable time but still remains computationally expensive compared to commonly used feature selection approaches mentioned in section 2.

References

[1] L. Breiman. Random forests. *Machine learning*, pages 1–33, 2001.
[2] Carnegie Group Inc and Reuters Ltd. Reuters-21578 Text Categorization Test Collection.
[3] S. Cohen, E. Ruppin, and G. Dror. Feature selection based on the shapley value, 2005.
[4] G. Forman. An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Journal of Machine Learning Research*, 3(7-8):1289–1305, Oct. 2003.
[5] X. Geng, T.-Y. Liu, T. Qin, and H. Li. Feature selection for ranking. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '07*, (49):407, 2007.
[6] E.-h. S. Han and G. Karypis. Centroid-Based Document Classification : Analysis and Experimental Results. *Principles of Data Mining and Knowledge Discovery*, 1910:116–123, 2000.
[7] A. Keinan, B. Sandbank, C. C. Hilgetag, I. Meilijson, and E. Ruppin. Fair attribution of functional contribution in artificial and biological networks. *Neural computation*, 16(9):1887–915, Sept. 2004.
[8] J. Liu and S. Lee. Study on feature select based on coalitional game. In *Neural Networks and Signal Processing*, pages 445–450, 2008.
[9] R. Mihalcea. Instance based learning with automatic feature selection applied to word sense disambiguation. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, page 7. Association for Computational Linguistics, 2002.
[10] G. Salton and C. Yang. On the specification of term values in automatic indexing. *Journal of documentation*, 29(4):351–372, 1973.

[11] F. Schilder and R. Kondadadi. FastSum: fast and accurate query-based multi-document summarization. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, number June, pages 205–208. Association for Computational Linguistics, 2008.

[12] E. Winter. The shapley value. *Handbook of game theory with economic applications*, 101(406):644, May 2002.

[13] Y. Yang and J. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420. Citeseer, 1997.