

# Project Summary: Document Classifier

Arjun Patel<sup>1</sup>, Harshil Prajapati<sup>1</sup>, and Bowen Song<sup>1</sup>

<sup>1</sup>Department of Electrical and Computer Engineering, Boston University

## 1. Problem Definition

This project explores machine learning algorithms for document classification. The specific application of the project is sensitive to datasets. The contribution of the project is to compare and contrast algorithm performances based on results and computation efficiency with respect to different types of datasets.

## 2. Literature Review

Document Classification is a traditional problem where usually a bag-of-words approach is to extract features and are used for supervised classification using Naive Bayes or Support Vector Machine (SVM) algorithms [1].

In classical papers, the order of words is ignored, but the order can be included by using Term Frequency - Inverse Document Frequency (TF-IDF) they can be incorporated [2]. Latent semantic indexing was introduced in [3] which uses Singular Value Decomposition onto TF-IDF space. This was modified by [4] to fitting the model to data using maximum likelihood or Bayesian methods known as Probabilistic latent semantic indexing (pLSI). Latent Dirichlet allocation (LDA) uses mixture models considering the exchangeability of both words and documents [5]. Kernel methods (KMs) are an effective alternative to explicit feature extraction. Another way to efficiently extract features is based on finding conditional probability using n-gram models [6].

More over the traditional supervised learning techniques, research has been done to improve prediction in k-NN by using pyramidal decomposition [7], skip-gram and paragraph vectors-distributed bag of words (PV-DBOW) with multiple discriminant analysis [8], expansion method and Powerset-label mechanism for the short hierarchical classification using the Support Vector Machine (SVM) classifier [9], Bayes-optimum decision with the maximum margin principle yields kernels for SVMs [10]. Bayes-optimum decision with the maximum margin principle yields kernels for SVMs [10].

## 3. Proposed Work

This projects aims at implementation of various document classification algorithms discussed in Section 3.1 This project also includes stop-words filtering based on stop-words vocabulary list and TF-IDF in attempt to improve performances for each model. The project learns from different datasets; the overview of the datasets is included in later chapters. In addition to discussing algorithm performances, the project also concerns the importance of preserving word ordering within a document with respect to classification accuracy.

### 3.1. Model and Algorithms

- *Preprocessing using TF-IDF*  
A way to do thresholding aside from just word frequency is by determine how important a word is in determining the class of a document. To determine how important a word is we will be using TF-IDF weighting, which takes into consideration word frequency as well as document frequency. The weight of the word is the frequency of a word multiplied by the inverse document frequency. A word has the highest weight when with a high word frequency and a low document frequency. Using the TF-IDF, we can set a threshold, which would decrease the feature size and remove irrelevant words [11].
- *Bag-of-words model with Naive Bayes assumption*  
The Naive Bayes classifier with bag-of-words model is considering all features as conditionally independent given the class label and evaluated using MPE decision rule. [12] The prediction performance considers to be the baseline for our prediction accuracy. The decision is based on MPE rule:

$$h_{MPE} = P(Y_j|\mathbf{x}_j) \propto P(Y_j) \prod_{i=1}^{d_j} P(w_{i,j}|Y), \quad j = 1, \dots, n$$

- Multi-class logistic regression (Maximum Entropy Classifier)

The logistic regression of multiples class is a supervised learning by maximizing the conditional likelihood.

$$\hat{\theta}(\mathbb{D}) = \arg \max_{\theta} \sum_{j=1}^n p(y_j | x_j, \theta), \quad j = 1, \dots, n$$

Without a closed form, best  $\theta$  is learnt via gradient descent with respect to the following questions.

$$\nabla_{\mathbf{w}_y} NLL(\theta) = \sum_{j=1}^n \left( \frac{e^{\mathbf{w}_y^T \mathbf{x}_j^{ext}}}{\sum_{k=1}^m e^{\mathbf{w}_k^T \mathbf{x}_j^{ext}}} - 1(y_j = y) \right) \mathbf{x}_j^{ext}$$

$$\nabla_{\mathbf{w}_y} f(\theta) = \nabla_{\mathbf{w}_y} NLL(\theta) + \lambda \mathbf{w}_y, \quad y = 1, \dots, m.$$

As a MPE/MAP Classifier, the decision is based on MPE rule where we maximize  $f(\theta)$ .

- Sensing aware kernel SVM [10]

This kernel design of SVM combines Bayes-optimum decision boundary with Maximum margin principle. [10]

- N-gram

The N-grams model uses the probability of each word appearing. For unigram, it considers each word as independent of all other words. Therefore, for a sentence to appear, it would be the probability of each word appearing multiplied together. For bigram, it takes the probability of the first word appearing multiplied by the probability of the second word appearing given the first word and so. The probability of the word depends on the probability of the all the previous words that appear before it in a sentence. [13]

$$Unigram: P(w_{1,n}) = P(w_1).P(w_2) \cdots P(w_n) \quad (1)$$

and

$$Bigram: P(w_{1,n}) = P(w_1).P(w_2 | w_1) \cdots P(w_n | w_{n-1}) \quad (2)$$

In the n-gram models, the features can become very large i.e.  $D^N$  for N-gram model which can be reduced by setting a threshold frequency of appearance [6].

## 3.2. Code and Dataset

We have multiple datasets in which we want to implement these algorithms.

1. News Group (preprocessed) - 20 class, 11,314 training documents , 7,532 test documents.
2. Reuters21578 (preprocessed) - 65 class, 945 training documents , 2,347 testing documents [14]
3. Personalized Medicine: Redefining Cancer Treatment - 9 classes, 3,320 training documents , 5,667 testing documents [15]

## 3.3. Minimum Achievable Plan

We would like to implement and compare all the algorithms but if we aren't able to complete all the algorithms we will try to implement at least Naive Bayes, Logistic Regression, Traditional SVM and N-gram.

## 3.4. Code and Dataset

We have multiple datasets in which we want to implement these algorithms.

1. News Group (preprocessed) - 20 class, 11,314 training documents, 7,532 test documents.
2. Reuters21578 (preprocessed) - 65 class, 945 training documents, 2,347 testing documents [14]
3. Personalized Medicine: Redefining Cancer Treatment - 9 classes, 3,320 training documents, 5,667 testing documents [15]

## 3.5. Minimum Achievable Plan

We would like to implement and compare all the algorithms but if we aren't able to complete all the algorithms we will try to implement atleast Naive Bayes, Logistic Regression, Traditional SVM ,N-gram.

## 4. Conclusion

In this project, we will be exploring different document classification algorithms. We will be using the Nave Bayes, Logistic Regression, SVM (kernel sensing), and n-grams models. Each of these three algorithms will be tested on three datasets: newsgroup dataset, Reuters21578 dataset and Personalized Medicine: Redefining Cancer Treatment (Kaggle Project) dataset. Each of the algorithms will be taking into account TF-IDF to give each word value of importance. At the end we want to be able to compare the performances of each algorithm in terms of accuracy of classification and computational efficiency across all three datasets.

---

The equations used are from [13] and [12]

## Division of Labor

- Arjun Patel
  - Logistic Regression
  - TF-IDF
- Harshil Prajapati
  - Pre-Processing of Personalized Medicine: Redefining Cancer Treatment Dataset
  - N-gram
- Bowen Song
  - Naive Bayes
  - Sensing Aware Kernel SVM

## References

- [1] D. S. Sachan, M. Zaheer, and R. Salakhutdinov, "Investigating the working of text classifiers," *arXiv preprint arXiv:1801.06261*, 2018.
- [2] P. Maes, "Agents that reduce work and information overload," in *Readings in Human-Computer Interaction*. Elsevier, 1995, pp. 811–821.
- [3] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, p. 391, 1990.
- [4] T. Hofmann, "Probabilistic latent semantic analysis," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 289–296.
- [5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.
- [6] J. Fürnkranz, "A study using n-gram features for text categorization," *Austrian Research Institute for Artificial Intelligence*, vol. 3, no. 1998, pp. 1–10, 1998.
- [7] P. Heroux, S. Diana, A. Ribert, and E. Trupin, "Classification method study for automatic form class identification," in *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, vol. 1. IEEE, 1998, pp. 926–928.
- [8] P. Lauren, G. Qu, F. Zhang, and A. Lendasse, "Discriminant document embeddings with an extreme learning machine for classifying clinical narratives," *Neurocomputing*, vol. 277, pp. 129–138, 2018.
- [9] Z. F. Salih and S. Tiun, "Term expansion and powerlabel set for multi-label hierarchical on short document classification," *International Journal of Applied Engineering Research*, vol. 13, no. 1, pp. 539–544, 2018.
- [10] W. Ding, P. Ishwar, V. Saligrama, and W. C. Karl, "Sensing-aware kernel svm," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2947–2951.
- [11] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to information retrieval*. Cambridge University Press, 2008, vol. 39.
- [12] P. Ishwar, "Ec 503 - learning from data," January - May 2018.
- [13] F. Song and W. B. Croft, "A general language model for information retrieval," in *Proceedings of the eighth international conference on Information and knowledge management*. ACM, 1999, pp. 316–321.
- [14] "Collection of datasets," <http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html>.
- [15] "Kaggle data: Personalized medicine: Redefining cancer treatment," <https://www.kaggle.com/c/msk-redefining-cancer-treatment>.