Bogazici University
Department of Computer Engineering

Graduation Project

Developing a Centroid-Based Classification Model for Text Categorization

Batuhan TUNA

Advisor:
Tunga GÜNGÖR

Spring 2017

Contents

1. Introduction and Motivation

1.1 Text Categorization

Great developments in the area of artificial intelligence has created different research areas in computer science. Text categorization or text classification is one of these areas. Many researches have been done for many years and these scientific studies still continue. Because, text classification is a subfield of machine learning and because of the optimization aim of machine learning techniques, it is impossible to find a generic method which solves this classification problem in a perfect way according to "No Free Lunch Theorem"[1].

Due to the great expansion of Internet, anyone can reach enormous number of texts. For finding the wanted document, documents should be classified. Additionally, we are living in the era of information and many information is still hidden inside documents and books. To retrieve useful information from them, we need to know which documents or text will be helpful. For this purpose, text categorization can be used because with text categorization we can learn the category of a document.

Text classification is not only used for separating the documents according to their topics. It is also used for different areas like spam filtering or language identification. The key point is the representation of documents or texts. Bag-of-words[2] and vector space model[3] is the most popular ones. Bag-of-words model is based on number of usages of the words. It gives the general information about which document contains which document. On the other hand, vector space model vectorizes the texts based on the frequency of the words. This vectorization process allows the comparison of documents and talking about similarity. This model is also used in centroid based classification models that will be explained later on this paper.

1.2 Simulated Paper

Chuan Liu and his colleagues tried to find a text categorization method that works well with the skewed data in their paper called "A New Centroid Based Classification Model for Text Categorization"[4]. They have proven that most of popular methods like Centroid Based Classifier(CBC) do not perform well if the dataset is skewed. In skewed datasets, some classes have high amounts of documents but some classes only have couple of documents. They proposed a method called "Gravitational Model" which considers the power of classes and asymmetry between different class centroids. They simulated this method with different datasets and compared its results with other text classification methods.

1.3 Aim of This Research

Chuan Liu and his colleagues proposed that Gravitational Model is successful but not enough. Because, there is not any mechanism about finding "good" centroids in this method. Good centroid means a centroid which gives the best information about the characteristics of a class and the least error when documents are classified. Gravitational Model uses tfidf method for obtaining centroids but tfidf does not give enough importance to term distributions among the corpus or does not adjust the positions of centroids by looking at misclassifications. So, in this research, I created a new method and compared it with both centroid based classifier and gravitational model.

2. State of the Art

Text categorization is a subfield of machine learning. So, applying known machine learning methods for text classification is possible. These applications must be done after representing the documents in a model like vector space model. After converting documents into classifiable types, common machine learning methods can be implemented. However, there are some important issues for choosing the best method.

2.1 Size of Feature Space

Size of the feature space eliminates most of common machine learning methods. A car can be represented by a few features like color, engine power, production year, etc. However, for representing the document words or word groups are used. This situation creates an enormous feature space comparing to feature space of car. Size of the feature space is maximum 10 or 15 for a car but it could be thousands or ten-thousands depending upon the number and length of documents. So, using machine learning methods like neural networks or decision trees will not perform well on document categorization. Because, these methods consider features one by one. However, doing it while text classification would cause poor performance which cannot be passed over.

2.2 Importance of Terms

Terms are the words which constitute the feature space for documents. They are creating the dimensions for each document if we are using vector space model. Normally, there is not any difference between the dimension in terms of importance. However, this should not be the case for document classification. Because, importance of the words is different. For example, if we see "the" in a text, we cannot receive any information about context but if "debugging" occurs in a text we can say that it can be about programming.

So, considering these two words in the same importance will cause misclassification. To overcome this problem, common words can be taken out from the feature space or a mechanism that considers the importance of words should be adapted.

2.3 Rarity of Classes

In text categorization, we consider multiple classes. Our task is classifying the text into one of these classes. However, these classes have different characteristics. Some classes may have only a couple texts, some classes may involve many documents. So, we should not approach to all classes in the same way. Classes which have small amounts of documents are called rare classes, and the others are called common classes. To get rid of rarity problem, multiple things can be done. Maybe rarest classes and their documents can be removed from the dataset. Another and better approach may be applying a classification mechanism that also considers the rarity of classes.

2.4 Multi-class vs Single-class

Documents may have multiple categories. For example, if we are considering news articles, an article about Minister of Economy's statement is a part of both politics and economics categories. In this case, different approaches should be applied. Because, methods which are for single labeled documents, generally finds a class for document and quits. However, in multi labeled texts, covariances between classes should also be considered. Additionally, talking about centroids in multi-class data is also hard. In this paper, the main target will be single-class data so we do not need to consider these problems.

3. Methods

3.1 Centroid Based Classifier (CBC)

Centroid based classifier[5] is a classification algorithm that is based on centroids in vector-space model of documents. For this algorithm, all documents should be converted into vector in the feature space. Feature space is created by the terms which are contained in training documents. After finding feature space, all documents can be converted into vectors which have same dimensions by finding the frequencies of words in documents. These vectors are called term-frequency(tf). Finding term frequencies is not enough for deciding characteristics of a document. Because, some terms may give more information about the class of the document, like explained in part 2.2 of this paper. So, another mechanism steps in which is called inverse document frequency(idf). idf value is changed by each term. By combining these two, we find "tfidf" for each term and document pair. It is calculated like this:

$$tfidf(t_k, \mathbf{d}_i) = tf(t_k, \mathbf{d}_i) \times \log \frac{|D|}{|D(t_k)|} \tag{3.1}$$

where $tf(t_k, \mathbf{d}_i)$ is the term frequency of term t_k in document $\mathbf{d}_i$, $|D|$ is the number of training documents and $|D(t_k)|$ is the number of documents containing term t_k. After finding tfidf values, centroids are found by this formula:

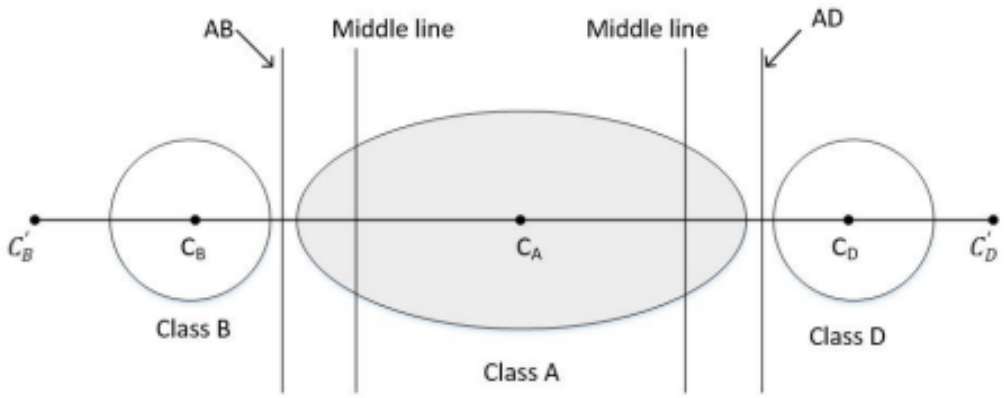
$$\mathbf{c}_j = \frac{\sum_{\mathbf{d}_i \in C_j} \mathbf{d}_i}{||\sum_{\mathbf{d}_i \in C_j} \mathbf{d}_i||_2} \tag{3.2}$$

After finding centroids, testing phase is started. For each document, similarity between the document vector and the centroid vector of each class is considered. The document is classified into the class which gives the highest similarity value. Similarity is measured by cosine function which is calculated by this formula:

$$\cos(\mathbf{d}, \mathbf{c}_j) = \frac{\mathbf{d} \cdot \mathbf{c}_j}{||\mathbf{d}||_2 \times ||\mathbf{c}_j||_2} \tag{3.3}$$

3.2 Gravitational Model (GM)

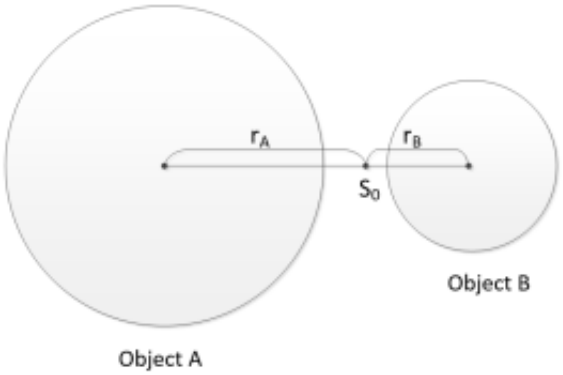
Gravitational model is an algorithm that is proposed by the paper[4] of Chuan Liu. In that paper, the authors have stated that centroid based classifier method has a huge problem. This classifier, also its improvements like DragPushing or LMDP, shows poor accuracy if the dataset is skewed. Skewed dataset is a situation related to the difference between rarities of classes. Let's say we have two different classes, class A and B, and A is a more common class than B. After applying the centroid based classifier we find the centroid which is the closest (most similar) to the document. If B is closer to the document, we expect that B must be a member of class B, but A is a more common class so separator between these two classes may not be in the halfway of them. This separator should be closer to the B, because A's region must be wider than B. Even though the document is closer to B, it may be in the A's region so this document is misclassified. Therefore, with CBC there is a huge risk of misclassification.



Because of this risk, Gravitational Model(GM) is proposed. This algorithm is based on Newton's gravitational law. As we know, two objects apply to each other an attractive force which is:

$$F = \frac{G * m_1 * m_2}{r^2} \tag{3.4}$$

where G is gravitational constant, m_1 is the mass of the first object, m_2 is the mass of the second object and r is the distance between these two objects. Let's say we put an object which has the mass of 1, between these two object. At some point, they will apply the same amount of attractive force to this object. This point is called equilibrium point.



In gravitational model text classification, every class centroid is accepted as objects which have masses. After finding centroids in the same way as centroid based classifier, masses of centroids are adjusted. Initially, all masses are equal(generally 1). For each training document, classification is made by calculating the attractive forces applied to it by classes. The class which applies the maximum force is found as the label of that document. Masses are adjusted, when there is a misclassification. Let's say document d is classified into class A but its real class is B. In this case, these adjustments are applied:

$$M_A := M_A - \xi \tag{3.5}$$

$$M_B := M_B + \xi \tag{3.6}$$

where M_A is the mass of class A, M_B is the mass of class B and ξ is a constant value used for adjustments. This adjustment is done until number of misclassified documents are not decreasing anymore. After finding centroids and adjusting masses, testing phase starts. Documents are classified into the classes which apply the maximum attractive force to them.

3.3 Class Feature Centroid (CFC)

Class feature centroid is a method that focuses on finding better centroids. Additional to CBC's term frequency and inverse document frequency, this algorithm also considers the inner-class and intra-class term index. In CBC's tfidf, main focus were the documents but in CFC main focus is the classes. Rather than calculating values for term and document pairs, CFC calculates values for term and class pairs. In this case, centroids are directly calculated by a single formula. Weight of term t_i of class j is calculated like this:

$$w_{ij} = b^{\frac{DF_{t_i}^j}{|C_j|}} \times \log \frac{|C|}{CF_{t_i}} \tag{3.7}$$

where b is a constant number($b > 1$), $DF_{t_i}^j$ is t_i's documents frequency in class j, $|C_j|$ is number of documents in class j, $|C|$ is total number of classes and CF_{t_i} is number of classes containing t_i. $b^{\frac{DF_{t_i}^j}{|C_j|}}$ is called the inner-class term index. It shows how common this term occurs in this class. $\log \frac{|C|}{CF_{t_i}}$ is called intra-class index. This part shows how common containing this term for a class. By considering both parts, better centroids are obtained. Also, this algorithm gives some elasticity because b can be changed according to the dataset.

3.4 Gravitational Class Feature Classifier (GCFC)

Gravitational model is a model which solves the skewed data problem. It is more like an improvement for centroid based classifier method. After finding the centroids, masses for those centroids are adjusted. However, as we know, centroid based classifier is not the best method for finding good centroids. Because of this situation, Chuan Liu said that "to choose good initial centroid vectors before learning the mass factors[4]" is crucial for observing better results. So, combining gravitational model with an algorithm that finds better centroids is a good idea.

By combining gravitational model (GM) with class feature centroid, I created gravitational class feature classifier (GCFC) method. In gravitational model, class centroids are determined by taking average of tfidf vectors for each document belongs to that class. However, in gravitational class feature classifier, they are determined by the cross product of inner-class and intra-class term indexes(Formula 3.7). After finding the centroids, mass factors are learned for each class. In testing phase, documents are classified according to attractive forces applied by centroids. Psuedocode for GCFC is given on the next page. This method combines the best parts of gravitational model and class feature centroid, so we expect better results from this method.

Algorithm 1: Gravitational Class Feature Classifier

Data:
Tr : whole training dataset with p classes and N samples
Te : whole testing dataset with M samples
ξ : step strength of updating mass factors
max_iter : maximum number of iterations
b : inner term index constant

```
find feature space;
calculate centroids c_j using formula (3.7) for each class j;
initialize masses for each class to 1;
prev_count = 0;
for iter = 1 to max_iter do
    wrong_count = 0;
    for each document d in Tr do
        classify document d calculating attractive force;
        if found label of d != class of d then
            i = found label of d ;
            j = class of d ;
            update M_i and M_j with formulas (3.5) and (3.6);
            wrong_count++;
        end
    end
    if wrong_count > prev_count then
        break;
    end
    prev_count = wrong_count;
end
for each document d in Te do
    classify document d calculating attractive force;
end
```

4. Experiments & Results

4.1 Datasets

In this project, two datasets are used: 20 Newsgroups and Reuters-21578.

4.1.1 20 Newsgroups

20 Newsgroups dataset is a popular dataset which is used on text categorization. Distribution of documents are nearly same, so there is not any rare or common categories. It consists 19,997 articles and 20 different categories.

4.1.2 Reuters-21578

Reuters-21578 is also a popular text classification dataset like 20 Newsgroups. In this dataset, there are 11,406 texts and 90 categories. It is a skewed dataset which contains rare and common classes.

20 Newsgroups dataset is used for observing overall performance of algorithms and Reuters-21578 dataset is used for observing performance in skewed datasets.

4.2 Preprocessing Steps

Before beginning the experiments, there are some preprocessing operations that should be done. Firstly, documents which have more than one label is removed from the dataset. Because, all of these methods are based on single-label texts. After that, rarest categories which have less than 10 documents are removed. Because, they can be accepted as outliers. Then, stop words are removed in order to reduce feature space. Because, these words don't give any information about the characteristics of a document or a class. After that, the terms which occur less than 5 are removed to reduce the size of feature

space. These words can also be accepted as outliers. Finally, word endings and punctuations are removed using Porter stemming algorithm. After all these steps summary of the dataset becomes like this:

Dataset	Classes	Documents	Features
20 Newsgroups	20	12368	17901
Reuters-21578	64	9130	7132

4.3 Performance Measurement

Measuring the performance of algorithms, F_1 values are used. There are two kinds of F_1 values: Micro-F_1 and Macro-F_1. Both of them are calculated using precision and recall. Precision is calculated by dividing number of correct positive predictions to number of positive predictions. Recall is calculated by dividing number of correct positive predictions to number of positive samples. F_1 value for a class is calculated like this:

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.1}$$

After finding recall, precision and F_1 values for each class, we can calculate Macro-F_1 and Micro-F_1 like this:

$$Macro - F_1 = \frac{\sum^{|C|} F_1}{|C|} \tag{4.2}$$

$$Micro - F_1 = \frac{2 * \sum^{|C|} Precision * \sum^{|C|} Recall}{(\sum^{|C|} Precision + \sum^{|C|} Recall) * |C|} \tag{4.3}$$

where $|C|$ is the number of classes. Macro-F_1 measurement considers all classes in the same way. On the other hand, micro-F_1 measurement is impacted by the performance of rare categories. Like using two datasets, using these two measurements give information about both overall and skewed data performance.

4.4 Experiment Results

In experiments, three different methods are applied to the datasets. These methods are CBC, GM and GCFC. Macro-F_1 and Micro-F_1 values for both 20 Newsgroups and Reuters-21578 are calculated. The results for 20 Newsgroups dataset are like these:

20 Newsgroups	CBC	GM	GCFC
MacroF_1	0.822	0.845	0.809
MicroF_1	0.831	0.850	0.812

Results of experiment that are applied to Reuters-21578 dataset are like these:

Reuters-21578	CBC	GM	GCFC
MacroF_1	0.634	0.649	0.718
MicroF_1	0.670	0.683	0.741

As expected, centroid based classifier is outperformed for both datasets and both F_1 values by gravitational model, which was also proven in the paper. Additionally, gravitational class feature classifier has shown an extraordinary result in Reuters-21578 dataset but for 20 Newsgroups dataset it is not the case. Even though the values are near, we can see that GCFC is not a good method for 20 Newsgroups dataset. Reason of this situation will be explained in conclusion and discussion part in details.

Additionally, we can see that GM is an improvement of CBC because it shows the same trend but GCFC has shown a different pattern. So, we can not say that GCFC is better than GM and CBC or vice versa. In some cases, GCFC may be better and in other cases GM may be better than GCFC.

In GCFC, b number which determines the effect of inner-term index, is set to 2. Because, in both datasets bigger b numbers were giving worse results. b should be higher than 1, so it could be minimum 2.

5. Conclusion and Discussion

All of these methods have shown better performances in 20 Newsgroups dataset. Because this dataset is more homogeneous dataset than Reuters-21578. Homogeneous datasets are easier to classify, because there are no problems like rare class effect or difference in regions. In those datasets, finding good centroids is enough to reach good performances, because effects of classes are nearly same. As we can see from the results, all of these methods gave similar results. GCFC is slightly worse than the others so we can come up with this conclusion: Finding tfidf for 20 Newsgroups is better than finding class features, when determining centroids.

In Reuters-21578 dataset, we can see that GCFC outperforms both GM and CBC. CBC's poor performance is expected, because there is not any mechanism in CBC for dealing with rarity problem. Therefore, CBC has shown a bad performance in a skewed dataset like Reuters-21578. GM is a better method, because it makes some adjustments for getting rid of rarity difference problem but not good enough as GCFC. Because, GCFC's main focus is dealing with this problem in both centroid finding phase and mass adjustment. By creating this method, my aim was good performance on skewed dataset. So, we can say that, GCFC is a successful method for skewed datasets.

In speed metrics, CBC worked quicker than the others because there is not any adjustment phase after finding centroids. If we compare the remaining two methods, GCFC is quicker than GM because GM's centroid finding phase is slower. In GM, tfidf values are found for all documents and then their average is taken for finding centroids. However, in GCFC we directly find the centroids via the formula.

Metric	CBC	GM	GCFC
Homogenous Dataset	2^{nd}	1^{st}	3^{rd}
Skewed Dataset	3^{rd}	2^{nd}	1^{st}
Speed	1^{st}	3^{rd}	2^{nd}

6. Future Work

One of the biggest problems for both gravitational model and gravitational class feature classifier is the speed issue. Mass adjustment is a costly operation that takes the longest time in the whole algorithm. It iterates the documents over and over again, so its complexity is nonlinear even though the centroid selection phase is linear for both GM and GCFC. In the future, a better mass adjustment mechanism can be proposed to increase the performance.

As we see in results chapter, GCFC solves the centroid selection problem for skewed data but fails at homogeneous data. So, centroids still are not perfect. A mechanism that adjusts the positions of centroids can also be added to GCFC. For example, there is a CBC improvement called DragPushing. It simply looks at the misclassifications on training data and adjust the centroids for found class and actual class of the misclassified document. This approach or a similar method can be adapted to GCFC but adding a new method can also bring complexity cost which is definitely not wanted if we consider this version's speed problem.

GCFC works better on skewed dataset but GM outperforms it in homogeneous dataset. So, a metric that shows how skewed the dataset may be implemented. If this metric is lower than a threshold GM is used, otherwise GCFC is used.

7. References

[1] Wolpert, David H., and William G. Macready. "No free lunch theorems for optimization." IEEE transactions on evolutionary computation 1.1 (1997): 67-82.

[2] Wallach, Hanna M. "Topic modeling: beyond bag-of-words." Proceedings of the 23rd international conference on Machine learning. ACM, 2006.

[3] Salton, Gerard, Anita Wong, and Chung-Shu Yang. "A vector space model for automatic indexing." Communications of the ACM 18.11 (1975): 613-620.

[4] Liu, C., Wang, W., Tu, G., Xiang, Y. and Konan, M.: A new centroid based classification model for text categorization.(2016)

[5] Han, Eui-Hong Sam, and George Karypis. "Centroid-based document classification: Analysis and experimental results." European conference on principles of data mining and knowledge discovery. Springer Berlin Heidelberg, 2000.

[6] Guan, Hu, Jingyu Zhou, and Minyi Guo. "A class-feature-centroid classifier for text categorization." Proceedings of the 18th international conference on World wide web. ACM, 2009.