Text Classification by Aggregation of SVD Eigenvectors*

Panagiotis Symeonidis, Ivaylo Kehayov, and Yannis Manolopoulos

Aristotle University, Department of Informatics, Thessaloniki 54124, Greece
{symeon,kehayov,manolopo}@csd.auth.gr

Abstract. Text classification is a process where documents are categorized usually by topic, place, readability easiness, etc. For text classification by topic, a well-known method is Singular Value Decomposition. For text classification by readability, "Flesch Reading Ease index" calculates the readability easiness level of a document (e.g. easy, medium, advanced). In this paper, we propose Singular Value Decomposition combined either with Cosine Similarity or with Aggregated Similarity Matrices to categorize documents by readability easiness and by topic. We experimentally compare both methods with Flesch Reading Ease index, and the vector-based cosine similarity method on a synthetic and a real data set (Reuters-21578). Both methods clearly outperform all other comparison partners.

1 Introduction

Data mining algorithms process large amount of data to find interesting patterns from which useful conclusions can be extracted. Several popular data mining techniques (Naive Bayes classifier, TF/IDF weights, Latent Semantic Indexing, Support Vector Machines, etc.) have been used for text categorization. Most of them are based on the vector space model for representing documents as vectors. These methods categorize documents by topic, place or people interests.

In this paper, we are interested in categorizing documents by readability easiness. In particular, we exploit the vector space model for representing documents. Then, we use Singular Value Decomposition (SVD) for dimensionality reduction to categorize documents by readability easiness. Our advantage over other methods relies on the fact that SVD reduces noise, which is critical for accurate text categorization. In addition, we propose a new method, denoted as Aggregated SVD, which creates distance matrices that contain distances/dissimilarities between documents. These distance matrices are combined using an aggregation function, creating a new distance/dissimilarity matrix. As will be experimentally shown, this aggregated new matrix boosts text categorization accuracy.

* This work has been partially funded by the Greek GSRT (project number 10TUR/4-3-3) and the Turkish TUBITAK (project number 109E282) national agencies as part of Greek-Turkey 2011-2012 bilateral scientific cooperation.

T. Morzy, T. Haerder, and R. Wrembel (Eds.): ADBIS 2012, LNCS 7503, pp. 385–398, 2012.

The contribution of this paper is two-fold: First, it proposes a new text categorization technique (Aggregated SVD). Second, it compares the accuracy performance of several methods for text categorization by readability easiness.

The rest of the paper is structured as follows. In Section 2, we present the related work. In Section 3, we present basic information for text categorization by readability easiness. In Section 4, we present how classic SVD is applied in text categorization. In Section 5, we describe our new proposed method, denoted as Aggregated SVD. Experimental results are presented in Section 6. Finally, Section 7 concludes this paper.

2 Related Work

The application of SVD in a document-term vector space model has been proposed in [4] in the research area of information retrieval (IR). Documents and queries are represented with vectors and SVD is applied for reducing the dimensions of these vectors. Yang Yu [14] published a comparative evaluation of 12 statistical approaches to text categorization. Moreover, Joachims [7] explored the usage of Support Vector Machines for learning text classifiers and identified the benefits of SVMs for text categorization.

Sarwar et al. [10] compared experimentally the accuracy of a recommender system that applies SVD on data with the one that applies only collaborative filtering. Their results suggested that SVD can boost the accuracy of a recommender system.

In [13] the method of Latent Semantic Indexing (LSI) has been applied on feature profiles of users of a recommender system. These feature profiles are constructed by combining collaborative with content features. Dimensionality reduction is applied on these profiles by using SVD, to achieve more accurate item recommendations to users.

Guan et al. [5] proposed Class-Feature-Centroid classifier for text categorization, denoted as CFC classifier. It adopts the vector space model and differs from other approaches in how term weights are derived, by using inner-class and inter-class term indices.

Hans-Henning et al. [6] proposed a method that creates self-similarity matrices from the top few eigenvectors calculated by SVD. Then, these matrices are aggregated into a new matrix by applying an aggregating function. Our method differentiates from their work as follows. Their method has been used for boosting clustering of high dimensional data. In contrast, we are exploiting Aggregated SVD for text categorization purposes.

There are techniques outside the field of data mining called *readability tests* which are formulas used for determining readability of text and usually contain counting of syllables, words and sentences. Such technique is Flesch Reading Ease index [8] which is presented in section 3 and is one of the evaluated methods in this work.

McLaughlin G. H. in [9] proposed SMOG Grade readability formula that estimates the years of education needed to understand a piece of writing. It yields a 0.985 correlation with a standard error of 1.5159 grades with the grades of readers who had complete comprehension of test materials.

Dale et al. [3] proposed Dale-Chall readability formula that provides a numeric gauge of the comprehension difficulty that readers will have when reading a text. The authors used a list of 763 words that 80

Coleman et al. [1] proposed Cole-Liau index which relies on counting characters instead of syllables per word, unlike most other methods. This design allows easy implementation of Cole-Liau index in computer programs since words are not analyzed but just their length is measured. Similar technique is Automated Readability index presented in [11].

Spache in [12] introduced Spache readability formula for texts in English. Its output is grade level computed by comparing words in a text to a set list of common everyday words. Spache formulas results are more accurate when applied on texts for children up to fourth grade.

3 Preliminaries in Text Categorization by Readability Easiness

In this Section, we provide the basic methodology, which is usually followed for text categorization by readability easiness. More specifically, we divide documents to a train and a test set. Then, we compare each document belonging in the test set to each document belonging in the train set and decide in which category by readability easiness (easy, medium, advanced) it belongs. To compare documents to each other, the documents must be represented by vectors. The vectors of all documents have the same length, which is the dictionary length. The dictionary is a vector containing all the unique words of all documents in the train set. The value of each dimension in a document's vector is the frequency of a specific word in that document. For example, the value of the fifth dimension in a document's vector is the appearance frequency of the word of the dictionary's fifth dimension. The words are also called "terms", the dimensions' values are called "term frequencies" and the documents' vectors are called "frequency vectors". In the following, we present how vector space model is applied in the following three documents:

D1: Sun is a star.
D2: Earth is a planet.
D3: Earth is smaller than the Sun.

The dictionary is the vector:

Dictionary = [a, Earth, is, planet, smaller, star, Sun, than, the]

As depicted, the vector length is 9. The frequency vectors of the three documents are:

D1 = [1, 0, 1, 0, 0, 1, 1, 0, 0]
D2 = [1, 1, 1, 1, 0, 0, 0, 0, 0]
D3 = [0, 1, 1, 0, 1, 0, 1, 1, 1]

Frequency values are either 0 or 1 because documents are very small. Some terms, like "a" and "is", are found in every document or in the majority of them. These terms do not offer any useful information that helps the document categorization and are considered as noise. If we can remove them somehow, then the categorization will be done more effectively. For this purpose, stop words can be used, which is a list of words that will be ignored during the creation of the dictionary and the frequency vectors. After cleaning our data with the stop words, we additionally remove the noise by performing also dimensionality reduction of the frequency vectors through SVD.

A widely used method that does not belong in the data mining field is *Flesch Reading Ease index* [8]. It is based on three factors: (i) total number of syllables, (ii) total number of words, and (iii) total number of sentences in the document. Based on these three factors, Flesch Reading Ease index calculates a score from 0 to 100 for each document. The higher the score, the easier the document to be read. The formula for the English language is, as shown in Equation 1:

$$206.835 - 1.015 * \left(\frac{total\ words}{total\ sentences}\right) - 84.6 * \left(\frac{total\ syllables}{total\ words}\right) \quad (1)$$

As we can see, more syllables per word as well as more words per sentence mean higher reading difficulty and vice versa. Notice that we categorize documents by their reading easiness into three categories: easy, medium and advanced. Table 1 shows how the Flesch Reading Ease score corresponds to the three categories.

Table 1. Explanation of Flesch Reading Ease Score

Score	Readability Level
71-100	Easy
41-70	Medium
0-40	Advanced

4 The Classic SVD Method

By applying the SVD technique to a $m \times n$ matrix A, it can be analyzed to a product of 3 matrices: an $m \times m$ orthogonal matrix U, a $m \times n$ diagonal matrix S and the transpose of an $n \times n$ orthogonal matrix V [13]. The SVD factorization is shown in Equation 2:

$$A_{m \times n} = U_{m \times m} * S_{m \times n} * V^T_{n \times n} \quad (2)$$

The columns of U are orthonormal eigenvectors of AA^T, S is a diagonal matrix containing the square roots of eigenvalues from U or V in descending order,

whereas the columns of V are orthonormal eigenvectors of A^TA. To remove some noise from the data, dimensionality reduction should be applied. This is done by removing rows from the bottom of matrices U and S and left columns from matrices S and V^T.

Next, we will use SVD to categorize documents by readability easiness by using a running example. We categorize the documents into three categories: easy, medium and advanced. Table 2 contains the document frequency vectors of the train set.

Table 2. Frequency vectors of the documents

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
D_1	4	6	2	3	3	1	1	1	0	1
D_2	5	5	3	1	1	2	3	3	2	1
D_3	2	3	4	0	0	0	2	1	2	0
D_4	2	2	3	5	6	4	3	2	1	0
D_5	1	0	1	2	3	2	2	0	2	1
D_6	3	2	0	5	6	5	4	3	0	0
D_7	0	0	2	3	2	3	5	4	6	1
D_8	2	3	3	0	0	3	5	5	4	0
D_9	1	1	0	3	3	2	4	3	3	1

Each table row is a frequency vector. There are 9 documents and 10 unique terms. The first three documents (D1-D3) belong to the category "easy", the next three (D4-D6) to category "medium" and the last three (D7-D9) to category "advanced". The dictionary length and therefore the length of the vectors is 10 due to the number of the unique terms. The first three terms (T1-T3) have higher frequency values in the first three documents because these terms define the category "easy". For the same reason, the second three terms (T4-T6) define the category "medium" and the next three (T7-T9) the category "advanced". The last term (T10) does not describe any category and is considered as noise.

In the beginning, we can apply SVD factorization on the data of Table 2 without performing any dimensionality reduction, as shown in Equation 3:

$$A_{9\times10} = U_{9\times9} * S_{9\times10} * V^T_{10\times10} \tag{3}$$

Let's assume that we want to categorize a new inserted document in our running example, as shown in Equation 4:

$$test = [4, 4, 5, 3, 2, 2, 0, 3, 0, 1] \tag{4}$$

The document *test* belongs in the category "easy" and this can be confirmed by the first 3 dimensions of the vector, which present the highest values. To find out, in which category SVD will assign the new document, it should be compared with all the documents of the train set. However, before this, the *test* document should be represented in the new dimensional space, so that we will be able to

compare it with the documents represented in the U matrix. We call it *test_new* and calculate it by multiplying with matrix V and the inverse of S, as shown in Equation 5:

$$test\_new_{1\times 9} = test_{1\times 10} * V_{10\times 9} * S^{-1}_{9\times 9} \quad (5)$$

Next, the new vector *test_new* is compared to the 9 documents of the train set; more specifically, it is compared to every row of matrix $U_{10\times 9}$ using cosine similarity. The first vector (row) of matrix U corresponds to document D1, the second to document D2, etc. The calculated cosine similarities are shown in Table 3.

Table 3. Similarity of document test with the 9 documents (before removing noise)

Document	**Similarity with document "test"**
D_1	-0.14
D_2	0.53
D_3	-0.33
D_4	0.57
D_5	-0.32
D_6	-0.38
D_7	0.02
D_8	-0.13
D_9	0.03

Based on Table 3, the test document is more similar with D4, which belongs to the second category. The categorization is not correct as the document test belongs in the first category. The result could be improved by using the majority vote, which takes into account the 3 or the 5 highest similarities. However, in our running example, if we consider the majority vote from the 3 higher similarities (D2, D4 and D9), then the text categorization result is again incorrect.

Up to this point, we have applied SVD but have not really taken advantage of it because we have kept all the information of the original matrix (which contains noise). As stated earlier, noise removal can be performed by dimensionality reduction [13]. Next, we keep 80% of the total matrix information (i.e. 80% of the total sum of the diagonal of matrix S). That is, we reduce the 9×10 matrix S to keep the $c = 3$ largest singular values. Thus, S matrix becomes $S_{3\times 3}$. Then, the reconstructed A^* matrix is the closest rank-3 approximation of the initial matrix A, as shown in Equation 6:

$$A^*_{9\times 10} = U_{9\times 3} S_{3\times 3} V^T_{3\times 10} \quad (6)$$

Next, we insert again the *test* document in the new dimensional space, as shown in Equation 7.

$$test\_new2_{1\times 3} = test_{1\times 10} * V_{10\times 3} * S^{-1}_{3\times 3} \quad (7)$$

As shown, *test_new*2 has fewer dimensions. Next, we recalculate the cosine similarity of the *test_new*2 document with the 9 documents of the $U_{10\times 3}$ matrix. The

new similarities are shown in Table 4. As shown, the categorization is now correct. The test document is most similar with D1, which belongs to the same category, i.e. "easy". Also, the 3 highest similarities are with D1, D2 and D3. All of them belong to the category "easy" just as the *test* document. That is, the 20% of information that was discarded from the original matrix was actually noise.

Table 4. Similarity of document test with the 9 documents (after removing noise)

Document	**Similarity with document** *test*
D_1	0.97
D_2	0.87
D_3	0.69
D_4	0.37
D_5	-0.8
D_6	0.26
D_7	-0.67
D_8	0.2
D_9	-0.16

5 The Aggregated SVD Method

In this Section, we adjust the aggregated SVD [6] method, which is initially proposed for clustering of data, to run it for text categorization. Clustering of high dimensional data is often performed by applying SVD on the original data space and by building clusters from the derived eigenvectors. Hans-Henning et al. [6] proposed a method that combines the self-similarity matrices of the top few eigenvectors in such a way that data are well-clustered. We extend this work by also applying SVD to a matrix that contains document frequency vectors. Then, a distance matrix $D^i_{m\times m}$ is produced from every column of matrix $U_{m\times m}$. It is called *distance matrix* because it contains the distances (i.e. the dissimilarities) from every value of the column to the rest of the values in the same column. If matrix $U_{m\times m}$ has m columns then the total distance matrices are m $(i = m)$. Every matrix D is symmetrical and its diagonal consists of zeros. The values of each distance matrix are normalized as shown in Equation 8:

$$x_{normalized} = \frac{x}{\sqrt{(x_1^2 + x_2^2 + ... + x_N^2)}}, \tag{8}$$

where x is the value normalized and $\{x_1, x_2, ..., x_N\}$ the values of the distance matrix that is being normalized. When all selected distance matrices are computed and normalized, they are aggregated into a new matrix M with an aggregation function. Such functions are minimum value, maximum value, average, median and sum. In our running example, we used the sum function. That is, a cell in matrix M is the sum of the values in the corresponding cells of all distance matrices. Thus, M is a symmetric matrix (after all distance matrices are

symmetrical) that shows the distances between documents taking into account all columns of U. When a document is to be categorized, the same steps are followed as described in the previous section (with or without dimensionality reduction).

Next, we will perform aggregated SVD on our running example. If we apply SVD on the original frequency matrix and keep 80% of the information, then matrix U remains with 3 columns. Based on these 3 columns, we can build 3 distance matrices $D^i_{9\times 9}$, where $(i = 1, 2, 3)$. By aggregating the 3 matrices we conclude with matrix M of Table 5. As we mentioned before, M is symmetric and there are only zeros on its diagonal. As far as the remaining values are concerned, the closer a value is to 0, the smaller the distance of the two documents is (i.e. more similar).

Table 5. Aggregation distance matrix M (smaller values are better)

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9
D_1	0	-0.7169	0.0412	0.1885	-0.0123	0.5678	0.7600	0.5862	-0.5465
D_2	-0.7169	0	-0.7048	0.0978	0.6455	0.4771	-0.0737	-0.9082	-0.0440
D_3	0.0412	-0.7048	0	1.0129	-0.5747	1.3922	0.8414	-0.1216	0.2780
D_4	0.1885	0.0978	1.0129	0	-0.0035	-1.8320	0.1911	0.1983	-0.3611
D_5	-0.0123	0.6455	-0.5747	-0.0035	0	0.2802	0.4123	0.2385	-1.0267
D_6	0.5678	0.4771	1.3922	-1.8320	0.2802	0	0.4747	0.4820	-0.0774
D_7	0.7600	-0.0737	0.8414	0.1911	0.4123	0.4747	0	-0.7662	-0.7722
D_8	0.5862	-0.9082	-0.1216	0.1983	0.2385	0.4820	-0.7662	0	-0.6238
D_9	-0.5465	-0.0440	0.2780	-0.3611	-1.0267	-0.0774	-0.7722	-0.6238	0

Next, we will categorize again the *test* document based on the aggregated SVD method. For simplicity purposes in our running example, let's assume that we consider only the 1-NN document. As previously shown in Table 4, the 1-nearest neighbor (1-NN) [2] of the *test* document is D1. Then, based on matrix M (see Table 5), the most similar documents of D1 are D3 and D5 (with values 0.0412 and -0.0123 respectively). Since documents D1-D3 belong to category "easy" and documents D4-D6 belong to category "medium", we have two documents from category "easy" and one from category "medium". That is, the *test* document is correctly assigned to category *easy*. The main advantage of the Aggregated SVD over classic SVD relies on the fact that it takes into consideration every individual singular value separately.

6 Experimental Evaluation

In this Section, we experimentally compare the accuracy performance of 4 different methods for text categorization. These methods are: (i) k Nearest Neighbor Collaborative Filtering algorithm, denoted as Cosine, (ii) Latent Semantic Indexing, denoted as SVD, (iii) the aggregation of similarity matrices of SVD-eigenvectors

method, denoted as AggSVD, and (iv) the Flesch Reading Ease index, denoted as Flesch. For SVD and AggSVD, we have run experiments with 3 different levels of dimensionality reduction. The information we keep in each level is 30%, 70% and 100% for each method, respectively. The last one (100%) means there is no reduction at all. The performance measures are computed as follows [14]:

$$Precision = \begin{cases} \frac{a}{a+b}, & \text{if a+b} > 0 \\ \text{undefined}, & \text{otherwise} \end{cases} \tag{9}$$

$$Recall = \begin{cases} \frac{a}{a+c}, & \text{if a+c} > 0 \\ \text{undefined}, & \text{otherwise} \end{cases} \tag{10}$$

$$F = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{11}$$

where a is the number of correctly assigned documents in a category, b is the number of incorrectly assigned documents in a category, and c is the number of documents that incorrectly have not been assigned in a category.

6.1 Data Sets

To evaluate the examined algorithms, we have used a synthetic and a real data set.

Synthetic Data Set. As far as the synthetic data set is concerned, we have created an English text generator [1]. Based on our generator, we have generated documents of the 3 readability categories (easy, medium and advanced) by adding also up to 50% controllable noise to each document. Our generator uses three text files (easy.txt, medium.txt and advanced.txt), which contain terms used for generating documents. The names of the files indicate the difficulty level of the terms inside (easy.txt contains low difficulty terms, medium.txt contains medium difficulty terms and advanced.txt contains advanced difficulty terms). The generator has 3 input parameters: (i) number of files per category to be generated, (ii) number of terms per file to be generated, and (iii) the amount of noise to be generated.

For the first parameter, if the user inserts value 200 then 600 files will be generated, 200 for each category. For the last noise parameter, it is determined as a $1/x$ fraction. The user's input replaces the x in the fraction. For example, if user gives value 5, then the noise will be 1/5=20%. As noise the generator uses terms from the other 2 categories (i.e. if noise is 20%, then 80% of the terms of an easy level document will be derived from easy.txt and 20% from medium.txt and advanced.txt). Additionally, punctuation is added in the documents to use them for evaluating Flesch method. Recall that the number of sentences is a parameter for the calculation of the Flesch score. In the first category there is a 1/10=10% chance of adding a dot after each term. The result is sentences with

[1] `http://delab.csd.auth.gr/~symeon/generator1.zip`

average of 10 terms. In "medium" category this probability is 1/15=6.67% or 15 terms per sentence average. In the last category this number is 1/20=20%. As can be seen, the higher the difficulty level of the text, the lower the chance of dot appearance, because longer sentences mean lower readability easiness.

For our experiments, we have created 1200 documents (400 documents in each category) in the train set and 450 documents (150 documents in each category) in the test set. We have also created synthetic data set versions with different noise levels (i.e. 0%, 25%, 50%, and 75%). Here, we present only experiments with 50% controllable noise. Notice that our findings have been confirmed also with respect to the other synthetic data set versions.

Reuters 21578 Real Data Set. As far as the real data set is concerned, we have used the Reuters 21578 corpus. Reuters 21578 collection consists of 21578 news articles published during 1987 by Reuters news agency established in London, UK. In the Reuters data set the documents are categorized by topic. There are 5 topic super-sets: "exchanges", "orgs", "people", "places" and "topics". All topics in each of these sets are content related. For each document, a human indexer decided to which topic a document belongs. Table 6 shows the topic distribution across the 5 sets.

Table 6. Topic distribution of the Reuters 21578 collection

Category Set	Number of sub-Categories
exchanges	39
orgs	56
people	267
places	175
topics	135

The "topics" category of Table 6 concerns economic subjects. For instance, it includes subtopics such as "coconut", "gold", "inventories", and "money-supply". Typically a document assigned to a category from one of these sets explicitly includes some form of the category name in the document's text. However, these proper name categories are not as simple to assign correctly as might be seen. Notice that Flesch Reading Ease index was not tested with the real data set, because it calculates readability easiness score and it is not suitable for categorization by topic.

6.2 Algorithms' Accuracy Comparison on the Synthetic Data Set

In this Section, we test all four methods' accuracy performance on a synthetic data set with 50% controllable noise. That is, a 50% of the total document terms belong in the correct document category, whereas the other 50% are terms belonging in the other two categories. Moreover, the number of nearest neighbors

(k-nn) for AggSVD, SVD and Cosine methods is set to 5. Figure 1 shows the results for each readability easiness category (easy, medium, advanced).

As shown, for all three categories, AggSVD 30% has the best performance, followed by SVD 30%. Notice that the performance of both AggSVD and SVD is increased as we reduce the matrix dimensions (i.e. 100%, 70%, 50%, 30%). That is, the application of dimensionality reduction removes noise and focuses only on the important dimensions of the U matrix. Cosine similarity performs almost equal with SVD 50% and AggSVD 70%, but it can not follow the performance of the same methods when higher dimensionality reduction is applied. Finally, Flesch presents the worst performance. This can be explained due to the existence of high percentage of controllable noise in the data set, since Flesch can not capture it.

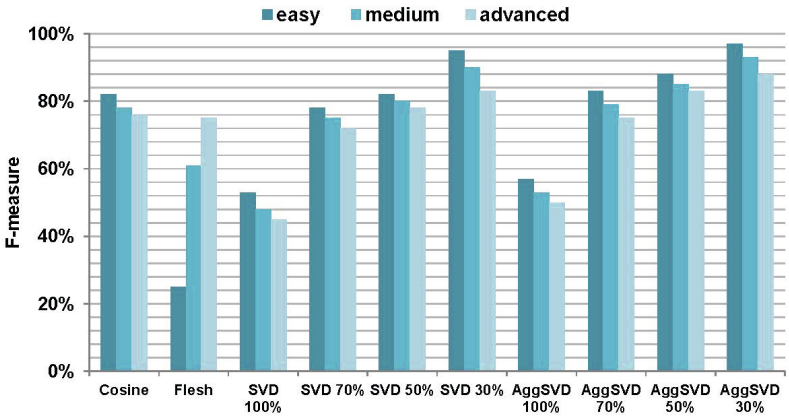

Fig. 1. F-measure diagram for the synthetic data set

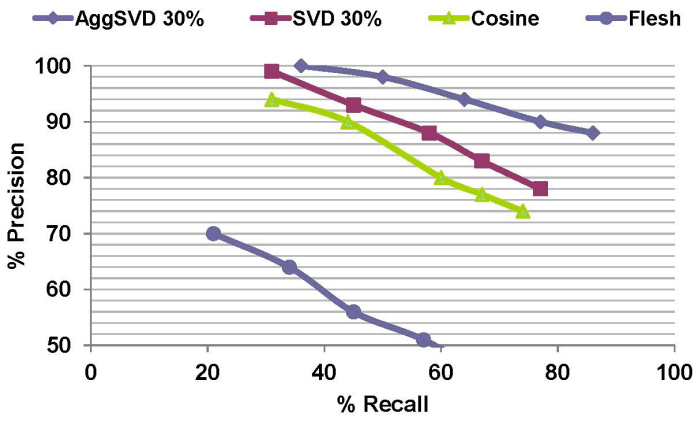

Fig. 2. Precision-Recall diagram for the synthetic data set

Next, we proceed with the comparison of all 4 algorithms by averaging their performance on all three categories (i.e."easy", "medium", "advanced"), in terms of precision and recall. This reveals the robustness of each algorithm in attaining high recall with minimal losses in terms of precision. As shown in Figure 2, the recall and precision vary as we increase the number of documents to be classified. AggSVD 30% precision value is almost 98% when we try to classify the first test document. This experiment shows that AggSVD 30% and SVD 30% are robust in categorizing correctly documents. The reason is that both of them perform dimensionality reduction, which removes noise from documents. In contrast, Cosine Similarity algorithm correlates documents without removing the noise. Finally, Flesch presents the worst results, because it does not take into account noise at all.

6.3 Algorithms Accuracy Comparison of Methods on the Real Data Set

In this Section, we test the comparison methods on a real data (Reuters 21578 collection). Notice that Flesch method is excluded from this experiment, since it is not suitable for topic classification. We have chosen 3 topic sub-categories to test our categorizations, i.e. "coffee", "gold" and "ship". As shown in Figure 3, both SVD and AggSVD perform almost equal. Notice that the best accuracy performance for both SVD and AggSVD is attained when we apply a 50% dimensionality reduction. That is, when we apply more than 50% dimensionality reduction in the data, we loss valuable information. This means that both SVD and AggSVD require appropriate tuning of the dimensionality reduction for each different data set.

Next, as shown in Figure 4, we plot a precision versus recall curve for all 3 algorithms. Once again, we re-confirm similar results with those of the synthetic data set. Both SVD and AggSVD outperform by far the Cosine Similarity method.

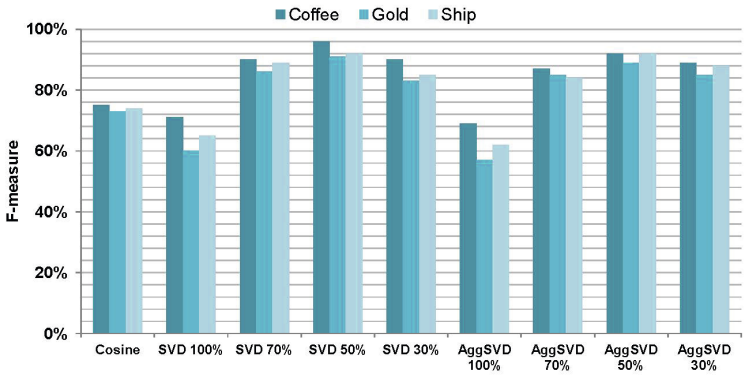

Fig. 3. F-measure diagram for the Reuters 21578 data set

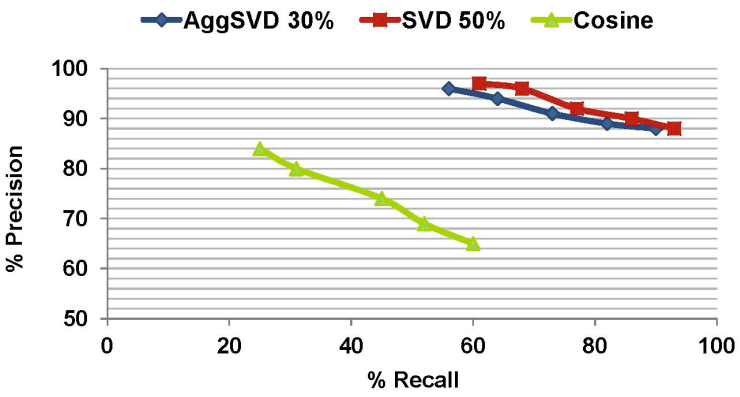

Fig. 4. Precision-Recall diagram for the Reuters 21578 data set

7 Conclusion

In this paper, we compared the performance of 4 methods in classifying text documents by readability easiness and by topic. In particular, we tested the k-NN collaborative filtering, the classic SVD, the Aggregated SVD and the Flesch Reading Ease index methods. We have shown that both classic SVD and Aggregated SVD techniques presented the best performance in a real and a synthetic data set. The results of the experiments showed us that dimensionality reduction improved the classification process in two ways: (a) better results, and (b) better efficiency. Better results means that more documents are categorized correctly. Better efficiency means lower computation cost because the computation of the similarities between documents is done using vectors with fewer dimensions. AggSVD is as a promising method, which needs further investigation in terms of other possible aggregation functions. Moreover, as a future work, we indent to compare our method with other state-of-the-art methods and with more real data sets for both readability and topic classification.

References

1. Coleman, M., Liau, T.L.: A computer readability formula designed for machine scoring. Journal of Applied Psychology 60, 283–284 (1975)
2. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13(1), 21–27 (1967)
3. Dale, E., Chall, J.: A Formula for Predicting Readability. Educational Research Bulletin 27, 11–20, 28 (1948)
4. Furnas, G.W., Deerwester, S., et al.: Information Retrieval Using a Singular Value Decomposition Model of Latent Semantic Structure. In Proceedings of SIGIR Conference, pp.465-480, Grenoble, France (1988)

5. Guan, H., Zhou, J., Guo, M.: A Class-Feature-Centroid Classifier for Text Categorization. In: Proceedings of WWW Conference, Madrid, Spain, pp. 201–210 (2009)
6. Hans-Henning, G., Spiliopoulou, M., Nanopoulos, A.: Eigenvector-Based Clustering Using Aggregated Similarity Matrices. In: Proceedings of ACM SAC Conference, Sierre, Switzerland, pp. 1083–1087 (2010)
7. Joachims, T.: Text Categorization with Support Vector Machines: Learning with many Relevant Features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
8. Kincaid, J.P., Fishburne, R.P., Rogers, R.L., Chissom, B.S.: Derivation of New Readability Formulas (Automated Readability Index, Fog Count, and Flesch Reading Ease formula) for Navy Enlisted Personnel. Chief of Naval Technical Training: Naval Air Station Memphis, Research Branch Report 8-75. Memphis, USA (1975)
9. McLaughlin, G.H.: SMOG Grading a New Readability Formula. Journal of Reading 12(8), 639–646 (1969)
10. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Application of Dimensionality Reduction in Recommenders Systems: a Case Study. In: Proceedings of ACM WebKDD Workshop, Boston, MA, pp. 285–295 (2000)
11. Smith, E.A., Senter, R.J.: Automated Readability Index. Wright Patterson AFB, Ohio. Aerospace Medical Division (1967)
12. Spache, G.: A New Readability Formula for Primary-Grade Reading Materials. The Elementary School Journal 53(7), 410–413 (1953)
13. Symeonidis, P.: Content-based Dimensionality Reduction for Recommender Systems. In: Proceedings of GfKl Conference, Freiburg, Germany, pp. 619–626 (2007)
14. Yang, Y.: An Evaluation of Statistical Approaches to Text Categorization. Information Retrieval 1(1-2), 69–90 (1997)