Submitted in part fulfillment of the requirements for the Specialization Program of
Big Data & Business Analytics

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Evaluation of Dimension Reduction Techniques for Text Classification and Sentiment Detection in Voice Recordings

by

Asterios Stergioudis

Department of Management Science and Technology

Athens University of Economics and Business

Supervisor: Dr. Harris Papageorgiou

Scientific Coordinator: Dr. Damianos Chatziantoniou

January 2015

Abstract

The purpose of this study is to evaluate and compare the most common dimensionality reduction algorithms. Their application has become essential due to the explosion of data nowadays. Given their variety and the fact that feature selection and feature transformation algorithms are usually studied separately, this study aims to give directions towards the methods that are performing the best on two different tasks; text classification and sentiment detection in voice recordings.

Three well-known text datasets were examined and 11 dimensionality reduction algorithms were evaluated based on the F1-score achieved in the classification on single test datasets. For the voice recordings dataset, four methods were applied since some of the other techniques were not applicable on this kind of data.

The results indicate that Singular Value Decomposition was usually the best performing method and 200 to 500 dimensions could be a logical value that provided increased predictive performance with relatively low computational overhead. However, other methods such as Gaussian Random Projections could be a viable alternative that is easily scalable to very large datasets and leads to small or no sacrifices on classification performance.

Table of Contents

1. Introduction

Text classification is the problem of automatically assigning predefined categories to text documents. Given the more and more textual information available, mainly online, text categorization is one solution to news filtering, document routing and personalization. Improved document classification techniques can also lead to better spam e-mail filtering systems, improved web search results and better translations between languages.

In text classification, 'bag of word' models are typically used where each position in the feature vector corresponds to a given word or phrase. The number of potential words often exceeds the number of training documents by many orders of magnitude. Therefore, a major difficulty faced in text classification problems is the high dimensionality of the feature space. This can be prohibitive for many machine learning algorithms (e.g. neural networks) since the computations needed cannot be completed in a reasonable amount of time. Dimensionality reduction not only helps conserve computation time, storage and network resources but can also improve categorization accuracy since irrelevant and/or redundant features can hinder the performance of learning algorithms due to the "curse of dimensionality".

Correctly indentifying the relevant features in a text is of paramount importance for text classification. However, this should be performed without sacrificing categorization accuracy and without any manual definition or construction of the features. Although one could try to find the combination of features that result in the highest classification accuracy using exhaustive search over the feature space or via search algorithms (e.g. genetic algorithms, simulated annealing), this task would be computationally expensive, if not impossible.

Sentiment detection in voice recordings is a relatively new research area. The difficulties encountered in this task refer to two issues; firstly, the data collected are often noisy since they have been obtained in a mechanical, and usually non-optimal, way and secondly, many of the statistics calculated on these data are usually correlated and/or redundant. For both of the above two reasons, dimensionality reduction techniques are considered a necessary preprocessing step before the application of any classification algorithm.

There are two general ways of automatic dimensionality reduction; feature selection and feature extraction. The former methods refer to the removal of non-informative terms according to corpus statistics; some of these methods are supervised (i.e. they take into account the text classes) while others are not. Most of these methods are applicable only to count data (e.g. how many times a word appears in a document) and will therefore be evaluated only on the text datasets. The latter methods construct new features that combine the original ones. Although these two technique categories are usually examined separately, the focus of this study is to evaluate and compare algorithms from both categories regarding their performance in text classification tasks.

The choice of the learning algorithm is not an object of this study. Support Vector Machines (SVM) have been shown to be a consistent top performer (Dumais, Platt, Heckerman, & Sahami, 1998). Therefore, a linear kernel SVM with default parameters will be used; the exploration of interactions between feature selection and model tuning could be an idea for further studies.

The following chapter firstly describes the different ways documents can be represented. Following, several methods of supervised and unsupervised feature selection are described. Lastly, four commonly used feature extraction algorithms are presented. Chapter 3 describes the methods that will be in-scope for the comparative study as well as the datasets that will be used. The results are shown in chapter four while the conclusions of this study are featured in the last chapter.

2. Literature Review

2.1 Document Representation

Documents, in their original form, cannot be used with the existing classification algorithms which require as input a matrix format. Therefore, in order to transform a document, or a set of documents, in a numeric matrix several approaches have been devised with the most common one being the bag-of-words (BoW) model. In the BoW model, each text (document or sentence) is represented as a bag of its terms. However, the grammar rules and the word order are completely disregarded. Moreover, no information is captured about the actual meaning of the words.

Since each text is represented as a vector of numbers, this representation is often referred to as a vector-space model. "Vector-space models rely on the premise that the meaning of a document can be derived from the document's constituent terms" (Berry, 1996). A document is represented as a vector where each dimension corresponds to a separate term. If a term is present in a document, the respective dimension will have a non-zero value. The terms are usually single words but generally they can be parts of words or multiple-word phrases. There are several ways of calculating these values; the three most common ones are presented below.

2.1.1 Binary Representation

In the case of Binary representation, each dimension's value is either 1 or 0, indicating the presence or absence of a term. In this way, a set of documents is transformed into a document-term (or term-document) matrix where each row corresponds to a document and each column to a term. If the terms are chosen to be individual words, then the total number of columns (i.e. the dimensionality of the matrix) is the number of distinct words present in the corpus.

2.1.2 Term Frequency Representation

An improvement over the Binary representation model would be to find a method that assigns different weights to each word in a document; in a way that 'more important' words are given bigger weights. A representation exploiting this idea is the Term Frequency representation. In this case, the value of the document vector in each dimension does not only represent the presence or absence of the respective term but it also gives a higher weight to the words that appear more often in a document. Essentially, each dimension is the frequency of occurrence of the term (Technical University of Denmark). The term frequency is intuitively content-descriptive for the documents and it is a common weighting factor for document vectors.

2.1.3 Term Frequency - Inverse Document Frequency

The third, and most common, representation of documents is the so-called 'term frequency - inverse document frequency' (or tf-idf) representation. This method is based on two assumptions (Brazdil). The first one is similar to that of the Term Frequency representation; a term appearing more often within a document is usually more important than a term appearing rarely. Tf-idf goes one step further though claiming that if a term appears in many documents it will probably be less important (e.g. articles). This aspect is captured by the 'idf' part. The tf-idf weighting scheme assigns to term *t* a weight in document *d* given by the formula:

$$\text{tf-idf}_{t\,=}\ \text{tf}_{t,d}\ \text{x}\ \text{idf}_{t.}$$

The first part ($\text{tf}_{t,d)}$ is just the frequency of the word *t* in document *d*. The second part is calculated as follows:

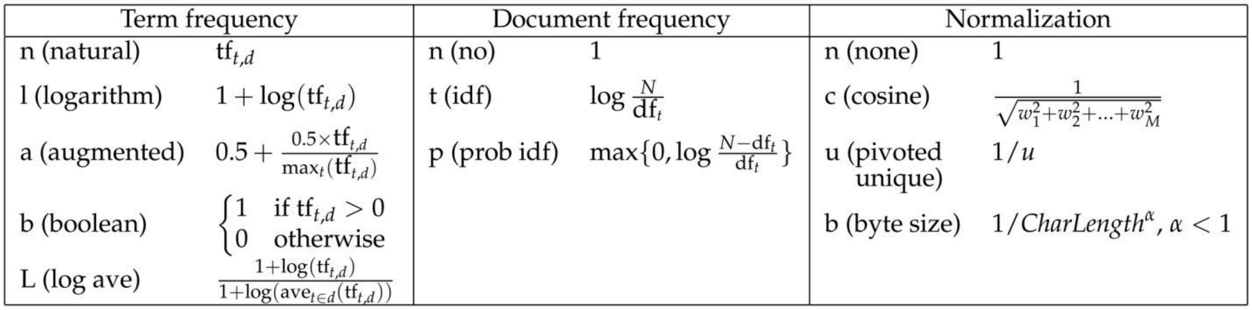
$$\text{idf}_t = \log_2(N/df(t))$$

where N is the number of documents in the corpus and df(t) is the number of documents in which the term *t* occurred (Brazdil).

There are various alternatives to the above weighting scheme. According to Manning et. al (Manning, Raghavan, & Schütze, 2009) one modification is based on the observation that a term appearing twenty times in a document, is unlikely to carry twenty times the significance of a single occurrence. Therefore, the logarithm of the term frequency could be used instead in the 'tf' part of 'tf-idf' weighting scheme.

Another observation is that term frequency scores are generally higher in longer documents just because longer documents tend to repeat the same words many times. Thus, term frequencies for each document could be normalized based on the maximum tf on that document. Usually a smoothing term is used that controls the scaling of the contributions of the terms in the document's vector.

The most common weighting schemes are shown in table 1 below. The table shows separately the term frequency, document frequency and normalization parts. The parts can be combined to create a large set of possible weighting schemes.



Term frequency		Document frequency		Normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1+\log(\mathrm{tf}_{t,d})$	t (idf)	$\log \frac{N}{\mathrm{df}_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5+\frac{0.5\times\mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$	p (prob idf)	$\max\{0,\log \frac{N-\mathrm{df}_t}{\mathrm{df}_t}\}$	u (pivoted unique)	$1/u$
b (boolean)	$\begin{cases} 1 & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$			b (byte size)	$1/CharLength^{\alpha}, \alpha < 1$
L (log ave)	$\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t\in d}(\mathrm{tf}_{t,d}))}$				

Table 1: tf-idf variants (adapted from Manning, Raghavan, & Schütze, 2009)

2.2 Feature Selection

Feature selection is an essential task that needs to be performed before the application of any classification algorithm. While this is true for any other classification task, it is fundamental in text classification due to the high dimensionality of text features and the existence of noisy features. The task of feature selection is to keep only those ones that are more probable to be relevant with the classification task and reject the others. This action not only makes the dataset more manageable and smaller in size but has also been proven to improve classification accuracy (Aggarwal & ChengXiang, 2012).

2.2.1 Text Pre-processing

The most common feature selection method is text pre-processing which effectively reduces the dimensionality of the term-document matrix by combining some features and by rejecting others. This is mainly achieved from a processing pipeline where text is firstly turned to lowercase characters. This way, for example, the terms 'Hotel' and 'hotel' become one and the same.

In addition, common words, called stop-words, are usually removed entirely. Such words are not specific or discriminatory to different classes (e.g. the word 'the'). The removal of stop-words is performed using an appropriate stop-word list. Besides stop-words, punctuation is sometimes removed while numbers are often removed as well (Aggarwal & ChengXiang, 2012).

A more technically difficult approach is word stemming. This technique groups words with common stem together. As an example, the words 'reads, 'read', 'reading' and 'readable' are all represented as one feature. This procedure is performed by removing suffixes and prefixes either with the help of a lexicon or using stemming algorithms, or stemmers. The resulting terms are usually conflated to avoid mismatches that may affect classification accuracy (Sandhya, Sri Lalitha, Sowmya, Anuradha, & Govardhan, 2011). Other approaches would be to replace synonym words with one term using the help of a lexicon.

Document frequency thresholding is a very simple technique for reducing the input dimensions. The frequency in each unique term is computed and those terms that are below a predefined threshold are removed from the feature space (Yang & Pedersen). Frequent words are more probable to be present in future test cases (Forman, 2003). This method is easily scalable however, it is considered an ad-hoc approach to feature selection and not a principled criterion for selecting predictive features.

The approaches described above are usually performed in both supervised and unsupervised contexts. However, "in the case of the classification problem, it makes sense to supervise the feature selection process with the use of the class labels" (Aggarwal & ChengXiang, 2012). Therefore, the methods presented below exploit the class labels in order to select a subset of the initial features giving each of them a score and keeping only those with the highest ones.

2.2.2 χ^2 statistic (Chi-square)

A common approach to feature selection for text classification tasks is the Chi-squared statistic which measures the lack of independence between a term and the category (Rogati & Yang). It actually measures the divergence from the expected distribution if one assumes that the feature occurrence is actually independent of the class value (Forman, 2003). Chi-square statistic is not considered appropriate for very small expected counts; something common in text classification because of having rarely occurring words and/or imbalanced classes. The features kept are the ones that satisfy a predefined significance level or the terms with the top-k Chi-square scores.

The Chi-square statistic for each term *t* is calculated as follows:

$$\chi^2(\mathrm{t,c}) = \frac{N\ x\ (p(t,c)x\ p(\bar{t},\bar{c}) - p(t,\bar{c})x\ p(\bar{t},c))^2}{p(t)x\ p(\bar{t})x\ p(c)x\ p(\bar{c})}$$

$$\chi^2(\mathrm{t}) = \mathrm{avg}_i\{\chi^2(\mathrm{t}, \mathrm{c_i})\}$$

where c are the possible classes of the classification task, N is the total number of documents and $\bar{t}$ or $\bar{c}$ is the negation of the term or class (e.g. $p(t, \bar{c})$ is the probability that the term t is not present in the class c). The final score of each term is the average of the respective scores for each possible class.

2.2.3 Gini Index

Another approach to the quantification of discrimination level of a feature is the so-called Gini Index. The intuition behind Gini Index is that it tries to measure the "purity" of a feature for categorization (Zhu & Lin, 2013).

The calculation of Gini Index is as follows. If the number of classes is k and $p_c(t)$ is the conditional probability that a document belongs to class c given that it contains the term t, then it true that:

$$\sum_{c=1}^{k} p_c(t) = 1$$

Then the Gini Index for the term t is defined as:

$$G(t) = \sum_{c=1}^{k} p_c(t)^2$$

It should be noticed that the value of $G(t)$ lies always between (1/k, 1) with higher values indicating greater discriminative power of the term t. The maximum value of 1 is obtained when all documents which contain the term t belong to the same class. In contrast, when the value of $G(t)$ equals 1/k when the distribution of documents containing the term t is uniform among the k different classes (Aggarwal & ChengXiang, 2012).

When the class distribution is skewed, then the above measure may not reflect the discriminative power of each term. Therefore, sometimes a normalized Gini Index is calculated which takes into account the distribution of the classes. If P_c is the global probability that a document belongs to the class c, then we can calculate the following measure:

$$p_c^{'}(t) = \frac{p_c(t)/P_c}{\sum_{j=1}^{k} p_j(t)/P_j}$$

and the normalized Gini Index for the term t is:

$$G(t) = \sum_{c=1}^{k} p_c^{'}(t)^2$$

In the case of biased class distributions in the whole corpus, the use of the global probabilities P_c ensures that the Gini Index reflects more accurately the class discrimination (Aggarwal & ChengXiang, 2012). It should be noted that sometimes the formula is written as:

$$G(t) = 1 - \sum_{c=1}^{k} p_c(t)^2$$

where lower values indicate higher "purity" of the attribute for categorization.

2.2.4 Information Gain

Information Gain "is a method that measures the decrease in entropy when the feature is given rather than not given" (Nicolosi, 2008). Information Gain is defined as follows:

$$IG(t) = -\sum_{i=1}^{k} P(c_i) \log P(c_i) + P(t) \sum_{i=1}^{k} P(c_i|t) \log P(c_i|t) + P(\bar{t}) \sum_{i=1}^{k} P(c_i|\bar{t}) \log P(c_i|\bar{t})$$

In machine learning, this metric is frequently used as a term goodness of fit criterion. It measures the numbers of bit of information obtained for category prediction by the presence or absence of the term in a document (Liu, Liu, Chen, & Ma, 2003).

The features that are selected are those that result in the largest decrease in entropy when they are removed from the set of all possible features.

2.2.5 Mutual Information

Another statistic that is used for feature selection in the Mutual Information. It compares the probability of observing term t and category c together (the joint probability) with the probabilities of observing t and c independently (i.e. by chance). The Mutual Information is defined as:

$$MI(t,c) = \log_2 \frac{p(t,c)}{p(t)p(c)}$$

As (Xu, Jones, Li, Wang, & Sun, 2007) point out, if the association between t and c is genuine, then the joint probability $\mathrm{p(t,c)}$ will be much larger than chance $\mathrm{p(t)p(c)}$ and, therefore $\mathrm{MI(t,c)}$ will be much larger than 0. If there is no significant relationship between t and c, then their joint probability will be almost equal to chance and thus $\mathrm{MI(t,c)}$ will be close to 0. On the other hand, if t and c are in complementary distribution then $\mathrm{p(t,c)}$ will be much less than $\mathrm{p(t)p(c)}$, forcing $\mathrm{MI(t,c)}$ to be much less than 0.

For global feature selection, the measure of goodness of a term can be calculated in two alternate ways (Yang & Pedersen):

$$MI_{avg}(t) = \sum_{i=1}^{k} P(c_i) MI(t,c_i)$$

$$MI_{max}(t) = \max_{i \in k}\{MI(t,c_i)\}$$

It should be noted that the above definition of the Mutual Information is not the same as the one used in information theory where the term refers to two random variables. While in information theory this measure is always non-negative, the above criterion can be negative.

2.2.6 Sampling-based Feature Selection

Feature selection can also be performed by 'randomly' selecting a subset of the original features. Specifically, the features are randomly sampled according to a probability distribution over the set of features. Drineas, Mahoney & Muthukrishnan (2006) show the theoretical properties of such column-based low-rank matrix approximations.

Three ways of assigning a probability to each feature have been proposed (Dasgupta, Drineas, Harb, Josifovski, & Mahoney, 2007). The simplest one, called uniform sampling, assigns equal probability to each feature i.e. $p_i = {}^{1}\!/_{n}$ for all $i = 1, \ldots, n$.

A second way, called Weight-based sampling, assigns a probability to each feature proportional to the squared length of the corresponding column of the matrix A, i.e. $p_i = {}^{\|A_i\|_2^2}\!/_{\|A\|_F^2}$ where the nominator is the square length of the column i and the denominator is the sum of the square length of all columns.

A similar sampling technique is called subspace sampling. In this method, the probability of choosing each feature is proportional to the squared length of the corresponding column of the matrix V_k consisting of the top k right singular vectors of A, i.e. $p_i = {}^{\left\|V_{k(i)}\right\|_2^2}\!/_{k}$ (please refer to section 2.3.1 below).

The above feature selection strategies are closely related to the so-called CUR decomposition of a matrix where the original matrix A is approximated by the multiplication of three other matrices as follows:

$$A = CUR$$

where matrix C contains r columns from A, R contains r rows from A and U is calculated based on the Singular Value Decomposition of the intersection of the chosen columns of C and the chosen rows of R (Rajaraman & Ullman, 2011).

2.2.7 Regularized Regression

Although not a feature selection technique per se, regularized, or penalized, regression models are classifiers that at the same time reduce the dimensionality of the data. In a supervised learning task, Logistic Regression models the probability distribution of the class label y given a feature vector x as follows:

$$p(y=1|x;\theta)=\frac{1}{1+\exp(-\theta^{T}x)}=h_{\theta}(x)$$

where θ are the parameters of the model. In this way, the log odds of an event happening, are modeled a linear function of the features (Kuhn & Johnson, 2013). The optimization problem for the Maximum Likelihood Estimate (MLE) of the parameters θ for Logistic regression can be written as (Lee, Lee, Abbeel, & Ng, 2006):

$$\min_{\theta}\sum_{i=1}^{M}-\log p(y=1|x;\theta)$$

The unregularized logistic regression estimates are unbiased and, of all unbiased linear techniques, this model also has the lower variance. However, it is possible to produce models with higher accuracy by allowing the parameter estimates to be biased. Regularization of the parameter estimates can be accomplished by adding a penalty term to the above minimization problem.

The Least Absolute Shrinkage and Selection Operator model, also called LASSO, adds a penalty on the absolute value (i.e. L1-norm) of the parameter estimates as follows:

$$\min_{\theta}\sum_{i=1}^{M}-\log p(y=1|x;\theta)+\lambda\sum_{j=1}^{P}\left|\theta_j\right|$$

As Hastie, Tibshirani & Friedman (2009) mention, the above form of the optimization problem is called its Lagrangian form and can also be stated as:

$$\min_{\theta}\sum_{i=1}^{M}-\log p(y=1|x;\theta)$$

$$\text{subject to } \textstyle\sum_{j=1}^{P}\left|\theta_j\right|\leq t$$

This addition in the minimization function has a significant impact on the parameters since some of them are shrunk towards 0. The higher the regularization parameter λ, the more parameter estimates become 0. Those parameters are effectively excluded from the model and that's the reason LASSO performs feature selection.

A small but important change of the optimization function leads to the so-called ridge regression:

$$\min_{\theta} \sum_{i=1}^{M} -\log p(y = 1|x; \theta) + \lambda \sum_{j=1}^{P} \theta_j^2$$

This slight modification leads to parameter estimates that are shrunk but never are exactly 0, thus the dimensionality of the input vectors is not reduced. The advantage of this change though is that it is more suitable for problems where the predictors are expected to be highly correlated. It can be proved that ridge regressions applies a greater amount of shrinkage to the directions of the principal components of the feature space X that refer to the lowest singular values (please refer to section 2.3.1 below). In essence, it shrinkages the 'least important' dimensions of the input space.

LASSO and ridge regression can be combined in order to gain the advantages of both methods. An intuitive approach would be to raise the parameter vector of the penalty term to a power between 1 (lasso) and 2 (ridge). However, this selection does share the ability of LASSO to set coefficients exactly to 0. The solution has been proposed by Zou and Hastie (2005) who introduced the Elastic Net penalty:

$$\lambda \sum_{j=1}^{P} (a\theta_j^2 + (1 - a) \left|\theta_j\right|)$$

which is a different compromise between ridge and lasso. The Elastic Net penalty has the effect of selecting variables and also shrinking the coefficients of correlated predictors (Hastie, Tibshirani, & Friedman, 2009).

2.3 Feature Transformation

Dimensionality reduction can be achieved either by selecting a subset of the original features with one of the methods described above, or the input space can be transformed to a low dimensional space. The latter way is called Feature Transformation, Feature Extraction, Low Rank Approximation or Subspace Learning. Using the former way, the original representation of the variables is not changed; it is typically preferred when one needs to keep the original meaning of the features and needs to determine which of the features are important (Masaeli, Fung, & Dy, 2010).

Most approaches exploit the matrix representation of the text (i.e. the term-document matrix) and try to summarize this matrix by finding a 'narrower' one that is, in some sense, close to the original. Apart from requiring less storage, these new matrices can be used much more efficiently but have also been proven to provide a more efficient representation of the relationship between the data elements. The Low Rank Approximation identifies the most essential components of the data and ignore components attributed to noise or inconsistencies. Several Low Rank Approximations are available for a given matrix: QR, URV, SVD, SDD, PCA, ICA, NMF, CUR etc. (Langville, Meyer, & Albright, 2006). Below, the most commonly used dimension reduction techniques for text classification are described.

2.3.1 Singular Value Decomposition

Singular Value Decomposition (SVD) is a method from linear algebra that decomposes a matrix into three matrices that, when multiplied, give as a result the original matrix. SVD in the context of text analysis, is often referred to as Latent Semantic Indexing (LSI) or Latent Semantic Analysis (LSA).

Given an $m\, x\, n$ term-document matrix A with m rows (i.e. documents) and n columns (i.e. terms), the SVD of A is defined as:

$$A = U\Sigma V^T$$

where $U = [u_{ij}]$ is an $m\, x\, n$ column-orthonormal matrix whose columns are called left-singular vectors; that is, each of its columns is a unit vector and the dot product of any two columns is 0.

$\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_v)$ is an $n\ x\ n$ diagonal matrix whose diagonal elements are called non-negative singular values and are sorted in descending order, and $V = [v_{ij}]$ is an $n\ x\ n$ column-orthonormal matrix whose columns are called right singular vectors. If the rank of the matrix A is r $(rank(A) = r)$ then Σ satisfies (Steinberger & Ježek, 2004):

$$\sigma_1 \geq \sigma_2 \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_v = 0$$

The rank of a matrix is defined as the size of the largest number of rows, or equivalently columns, we can choose for which no non-zero linear combination of them is the all-zero vector. Schematically, the Singular Value Decomposition of a matrix A can be represented as:

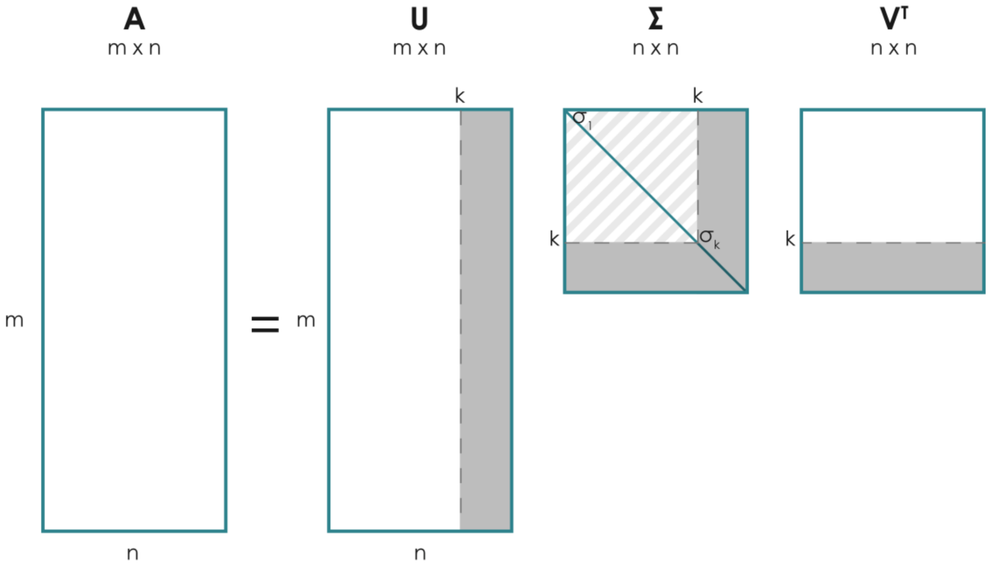


Figure 1: Singular Value Decomposition

SVD is a method closely related to Principal Component Analysis (PCA); in fact, they are so intimately related that the names are often used interchangeably. PCA seeks to find linear combinations of the predictors, know as Principal Components (PCs), which capture the most possible variance. The first PC can be defined as "the linear combination of the predictors that captures the most variability of all possible linear combinations" (Kuhn & Johnson, 2013).

Then, subsequent PCs are derived such that these linear combinations capture the most remaining variability while also being uncorrelated with all previous PCs. It should be noted that PCA is an unsupervised feature extraction method and thus it seeks predictor-set variation without regard to the target variable. Since it maximizes variability, the method is drawn towards to summarizing predictors that have more variation. If the original predictors are on measurement scales that differ in orders of magnitude, then the first few PCs will focus on summarizing the higher magnitude predictors (Kuhn & Johnson, 2013). Therefore, it is suggested that the original data are normalized (i.e. centered around their mean and scaled based on their standard deviation) before PCA is applied.

PCA is directly related to the SVD when the PCs are calculated from the covariance matrix (Wall, Rechtsteiner, & Rocha, 2003). PCA performs an eigenvalue decomposition of the covariance matrix as follows:

$$AA^T = W\Lambda W^T$$

where W is an orthonormal matrix and Λ is a diagonal matrix. The columns of W are the eigenvectors of matrix AA^T and the diagonal elements of Λ are the respective eigenvalues (Madsen, Hansen, & Winther, 2004). If we calculate the covariance matrix from the SVD of matrix A it is evident that the square roots of the eigenvalues of AA^T are the singular values of A:

$$AA^T = U\Sigma^2 U^T$$

By changing (r-k) of the lower values of Σ to zero, a low-rank approximation of matrix A, called A_k, can be created through the truncated SVD as:

$$A_k = U_k \Sigma_k V_k^T$$

Only the first k columns of U and the k rows of $V^T$are retained. By applying the SVD on a term-document matrix, documents are represented in a vector space of artificial concepts. Each of the k reduced dimensions corresponds to a latent concept (Kumar, 2009).

The most important property of the truncated SVD is that it forms the best axis to project the data on in the sense that A_k is the best approximation of A in terms of the Frobenius norm (Liberty, 2012). Frobenius norm of a matrix is the square root of the sum of absolute squares of its elements. Therefore:

$$A_k = \min_{B \in R^{mxn}} \parallel A - B \parallel_F$$

According the above, the truncated SVD of a term-document matrix A, can be interpreted as follows. U_k is the matrix that shows the relationship between documents and concepts, V_k links concepts to terms while Σ_k shows the significance of each concept.

2.3.2 Non-Negative Matrix Factorization

An alternative to Singular Value Decomposition that has gained momentum recently is the Non-Negative Matrix Factorization (NMF). The name of the method comes from the fact that all entries in the original matrix should be non-negative. The resulting matrices are non-negative as well. Given a non-negative matrix A, NMF finds non-negative matrices W and H such that:

$$A \approx WH$$

where A is an $m\ x\ n$ matrix, W is an $m\ x\ k$ and H a $k\ x\ n$ matrix and $k < n$. It should be noted that the matrices W and H are not unique. Schematically, the Non-Negative Matrix Factorization of a matrix A can be represented as:

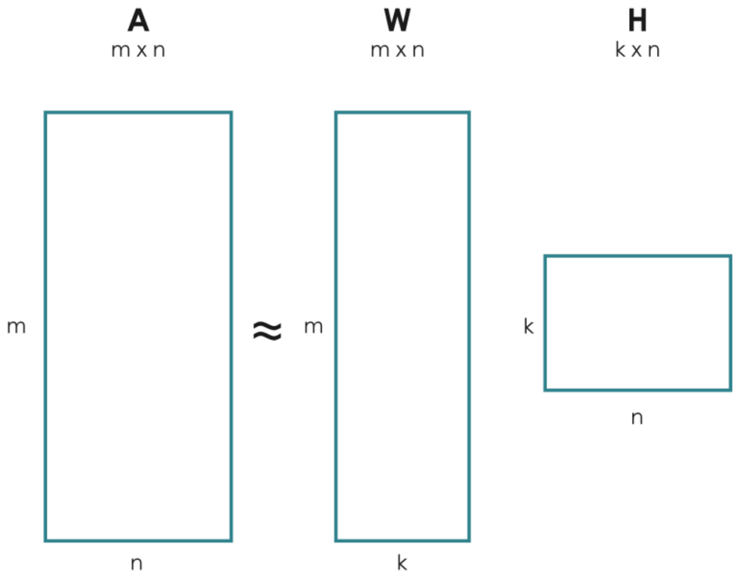


Figure 2: Non-negative Matrix Factorization

The motivation behind NMF is that besides the dimensionality reduction desired for many applications, when the underlying data is non-negative (e.g. term-document matrices) they can be better modeled and interpreted by means of non-negative factors (Boutsidis & Gallopoulos, 2007). NMF has gained much attention because its results can be interpreted straightforwardly i.e. each observation (e.g. document) can be explained by an additive linear combination of non-negative basis vectors. In contrast, the factors that result from the SVD of a matrix can be negative and therefore provide no interpretability. Moreover, NMF results in sparse matrices W and H if A is also sparse, as is the case for term-document matrices. This leads to reduced storage requirements.

SVD on the other hand, has the advantages of being able to find the optimal factors in terms of the Frobenius norm, it's computation is much less time-consuming, the factorization obtained is unique and the resulting factors are orthogonal and allow conceptualization of original data as vectors in space (Langville, Meyer, & Albright, 2006).

Finding the NMF is a non-convex problem and therefore iterative procedures are needed. W and H should be initialized; something that is usually performed randomly. However, other initialization methods have been proved to converge faster (Boutsidis & Gallopoulos, 2007). Some optimization algorithms need the initialization of only W or H with the non-initialized matrix satisfying some necessary properties. In order to find a good approximation to the matrix A, the cost function that quantifies the quality of the approximation should be defined. One possible selection of cost function for NMF is the Frobenius norm of the difference between the original matrix and the approximation:

$$\parallel A - WH \parallel_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\left|A_{ij} - WH_{ij}\right|^2}$$

subject to $W, H > 0$. The above cost function is lower bounded by zero and vanishes if and only if $A = WH$ (Lee & Seung, 2006). The second most used cost function for NMF optimization is the Kullback-Leibler Divergence defined as:

$$D_{KL}(A|WH) = \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij} \log\frac{A_{ij}}{WH_{ij}}$$

It should be noted that it is suggested that the matrix A as well WH are normalized before the application of the above optimization criterion. Lastly, the generalized Kullback-Leibler

Divergence (or I-divergence) criterion is also commonly used (Yang, Zhang, Yuan, & Oja, 2011) and is calculated as:

$$D_{KL}(A|WH) = \sum_{i=1}^{m}\sum_{j=1}^{n}(A_{ij}\log\frac{A_{ij}}{WH_{ij}} - A_{ij} + WH_{ij})$$

The interpretation of the matrix decomposition is similar to the one for SVD with the difference that the latent dimensions cannot be given a relative importance; in contract with SVD factors that are weighted based on the respective singular values. If A is an $m\,x\,n$ document-term matrix, then W is an $m\ x\ k$ matrix that can be interpreted as a topic-term matrix whose columns are the NMF basis vectors. The element j of row 1 of W measures the strength to which topic j appears in document 1 (Langville, Meyer, & Albright, 2006).

2.3.3 Random Projections

A set of dimension reduction techniques that have gained a lot of attention recently due to their simplicity, and mainly, due to their scalability are the Random Projections or the so-called embeddings. The idea is extremely simple; given a matrix A, the dimensionality of the data can be reduced by projecting it onto a lower-dimensional subspace formed by a set of random vectors. The whole idea is based on Johnson-Lindenstrauss lemma which states that for a set of m points in n-dimensional Euclidean space there exists a linear transformation of the data into a k-dimensional space (k<<n), $k \geq O({}^{\log m}/_{\varepsilon^2})$ that preserves pairwise distances up to a factor $1 \pm \varepsilon$ (Achlioptas, 2003). If A is an mxn matrix, then the projection of the data onto a k-dimensional subspace is calculated by:

$$A^{RP} = RA$$

where R in mxk random matrix. As Bingham & Mannila (2011) suggest, the above equation is not strictly speaking a projection because R is not orthogonal. However, based on a results presented by Hecht-Nielsen, "in a high-dimensional space, there exists a much larger number of almost orthogonal than orthogonal directions". Thus vectors having random directions might be sufficiently close to orthogonal.

Although Johnson & Lindenstrauss proved the existence of the above transformation, they did not present a way to get the linear transform. However Dasgupta (2000) shown that a random projection matrix can be obtained in several ways since certain random matrices satisfy the Johnson-Lindenstrauss lemma with high probability. A possible way to calculate the R matrix would be to generate a random matrix with i.i.d. Gaussian entries. This would lead to any two rows of the projection matrix being approximately orthogonal to each other and have approximately the same length. Thus there would be no need to normalize the vectors to unit length or orthogonalize the projection matrix (Mylavarapu & Kaban, 2013).

Achlioptas (2003) proved that there are, however, simpler ways of producing random projections. Specifically, the following two ways were presented:

$$r_{ij} = \begin{cases} 1\, with\, prob.\, 0.5 \\ -1\, with\, prob.\, 0.5 \end{cases}$$

$$\text{or } r_{ij} = \begin{cases} \sqrt{3}\, with\, prob.\, {}^{1}/_{6} \\ 0\, with\, prob.\, {}^{2}/_{3} \\ -\sqrt{3}\, with\, prob.\, {}^{1}/_{6} \end{cases}$$

These projections have the added benefit of being easy to implement and compute (Fradkin & Madigan, 2003). Bingham & Mannila (2011) state that practically all zero mean, unit variance distributions of r_{ij} would give a mapping that satisfies Johnson-Lindenstrauss lemma.

The above-described methods for the construction of the projection matrix can be integrated in the below formula:

$$r_{ij} = \sqrt{s} \begin{cases} 1\, with\, prob.\, {}^{1}/_{2s} \\ 0\, with\, prob.\, 1 - {}^{1}/_{s} \\ -1\, with\, prob.\, {}^{1}/_{2s} \end{cases}$$

where $s = 1$ or $s = 3$. Using $s = 3$, a threefold computation speedup can be achieved because only ${}^{1}/_{3}$ of the data need to be processed. For this reason, this method is also called Sparse Random Projections. It has also been suggested that s could be set to $\sqrt{n}$, were n is the dimensionality of the original dataset, with little loss of accuracy and with a $\sqrt{n}$-fold speedup

(Li, Hastie, & Church, 2006). The same authors also suggest that setting $s = \sqrt{n}$ can be robust against heavy-tailed data such as term-document matrices. They call this method very sparse random projections.

Lin & Gunopulos (2003) have also tried to combine SVD with Random Projections in order to mitigate the computational cost of SVD. Specifically, they firstly use Random Projections to reduce the dimensionality of the original dataset and then perform the SVD on the lower-rank matrix. However, their experimental results show that Random Projections prior to SVD do not always result in a faster algorithm than SVD alone.

2.3.4 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) refers to two distinct but related methods. The first one refers to classifier design where each class is modeled as Gaussian (with a covariance matrix and mean vector). Observations are classified to the class of the nearest mean vector according to the Mahalanobis distance (Torkkola, 2004).

In this thesis, LDA, also named Fisher Discriminant Analysis, refers to the second method where it is considered a discriminative feature transformation that is optimal for certain cases. LDA is a supervised feature reduction technique (i.e. the target variable of each sample is used) in contrast to the abovementioned feature extraction techniques.

The objective of LDA is to find directions in the feature space that preserve as much of the discriminatory information as possible. This is better illustrated with an example using two-dimensional data. LDA seeks to find a scalar y by projecting the samples x onto a line (i.e. direction) which maximizes the separability of the scalars. The following figure illustrates the two-dimensional case:

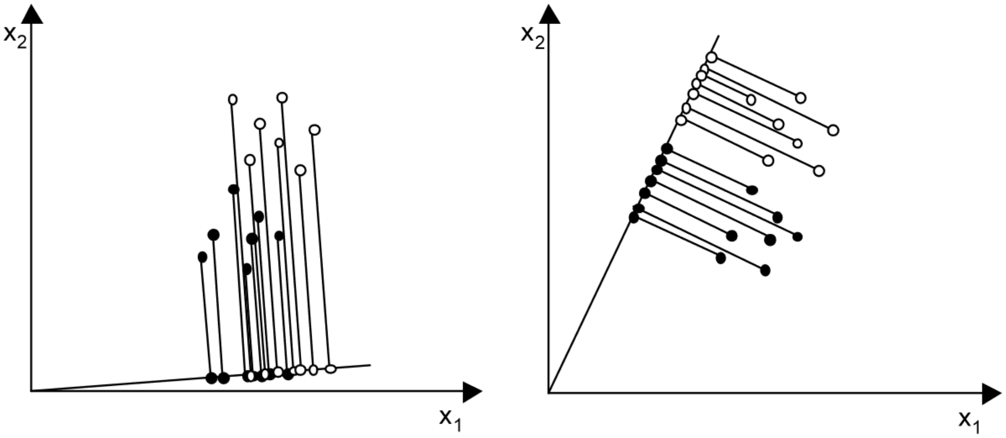


Figure 3: Two-dimensional LDA (source: Gutierrez-Osuna, 2002)

In order to find a good projection vector, a measure of separation between the classes should be defined. An intuitive objective function to maximize could be the distance between the projected means, however it is not a good measure since it does not take into account the standard deviation within the classes. Fisher proposed to maximize a function that represents the difference between the means, normalized by a measure of the within-class variability, also called scatter.

For each class, scatter, which is equivalent to variance, is defined as:

$$\tilde{s}_i^2 = \sum_{y \in c_i} (y - \widetilde{\mu_i})^2$$

$\tilde{s}_i^2$ measures the variability within class c_i after projecting it on the y-space. Therefore, in the case of a binary classification problem, $\tilde{s}_1^2 + \tilde{s}_2^2$ measures the variability within the two classes and is called the within-class variability of the projected samples (Shiry Ghidary, 2010). The Fisher Linear Discriminant is defined as the linear function $w^T x$ that maximizes the below criterion function:

$$J(w) = \frac{|\widetilde{\mu_1} - \widetilde{\mu_2}|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

This criterion leads to a projection where samples from the same class are projected very close to each other and the projected means are as farther apart as possible.

In a multivariate feature space, the equivalent to scatter are the scatter matrices. If S_i is the covariance matrix of class c_i, the matrix $S_w = S_1 + S_2$ is called the within-class scatter matrix. Now the scatter of the projection y can be proved to be equal to:

$$\tilde{s}_i^2 = w^T S_i w$$

$$\text{and } \tilde{s}_1^2 + \tilde{s}_2^2 = w^T S_W w = \widetilde{S_W}$$

where $\widetilde{S_W}$ is the within-class scatter matrix of the projection y. Similarly, the difference between the means in the y-space can be expressed as follows:

$$(\widetilde{\mu_1} - \widetilde{\mu_2})^2 = w^T S_B w = \widetilde{S_B}$$

$$\text{where } S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

The matrix S_B is called the between-class scatter of the original vectors whereas $\widetilde{S_B}$ is the between-class scatter of the projected samples. Thus, the Fisher criterion can be defined as:

$$J(w) = \frac{|\widetilde{\mu_1} - \widetilde{\mu_2}|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{w^T S_B w}{w^T S_W w}$$

It can be proved that the optimal vector w that maximizes $J(w)$, is the eigenvector of the matrix ${S_W}^{-1} S_B$ corresponding to the largest eigenvalue (Shiry Ghidary, 2010). In the case of multiclass classification, the optimal projection matrix W is still the one whose columns are the eigenvectors corresponding to the largest eigenvalues of the matrix ${S_W}^{-1} S_B$ where $S_w = \sum_{i=1}^{C} S_i$ and $S_B = \sum_{i=1}^{C} N_i (\mu_i - \mu)(\mu_i - \mu)^T$.

LDA has not been used extensively as a feature transformation technique for document classification according to (Torkkola, 2004). This is due to two reasons. Firstly, labeled data are needed for LDA to work and secondly, the matrix S_W used in the formula ${S_W}^{-1} S_B$ is singular and therefore, not invertible. Moreover, if its pseudo-inverse is used, there are no efficient methods for its eigenanalysis. A solution to this problem is to first perform a dimensionality reduction with SVD, NMF or Random Projections with an order of magnitude starting from the original document-term matrix and then perform an LDA transformation for further dimension reduction (Jieping & Shuiwang, 2008).

However, LDA has some limitations. One important drawback is that it produces at most $C-1$ feature projections. Therefore, it may not be appropriate for datasets with just a few classes since the number of dimensions extracted may not be enough to capture the information included in a large dataset. LDA is also a parametric method since it assumes unimodal Gaussian likelihoods. Lastly, LDA will fail if discriminatory information is not in the mean but in the variance of the data (Gutierrez-Osuna, 2002). Variants of LDA are Non-parametric LDA, Orthonormal LDA, Generalized LDA and Multilayer Perceptrons.

3. Methodology

The above-described dimensionality reduction techniques have different characteristics; others have better theoretical statistical properties (e.g. chi-square), others have optimal algebraic properties (e.g. Singular Value Decomposition) while others are considered more computationally efficient without sacrificing the dataset's predictive information (e.g. Random Projections).

No method is better than the rest in all occasions. Therefore, in this study a variety of dimension reduction methods will be evaluated in various datasets. Although feature selection and feature extraction techniques are usually studied separately, this study will evaluate and compare both types of dimensionality reduction. Since the features created by each method cannot be assessed per se, the evaluation will depend on the classification results of the resulting datasets.

The feature selection algorithms that are in-scope for this study are the following; Chi-Square (CS), Gini Index (GI), Information Gain (IG), Mutual Information using the max across all classes (MI), Weight-based sampling (WBS) and Subspace Sampling (SS). For SS the 100 dimensions of the SVD matrix were kept.

Regarding feature extraction techniques, the following ones will be examined: Singular Value Decomposition (SVD), Non-Negative Matrix Factorization (NMF), Random Projections with Gaussian entries (GRP), Very Sparse Random Projections (SRP) and Linear Discriminant Analysis (LDA). The last one can only create projections that are less or equal to the number of classes of the problem minus one. Thus, it will can only be compared with the other techniques in those dimensions.

All datasets are pre-processed before the application of the above algorithms. Specifically, in the text datasets all words are turned to lowercase characters, stop-words are removed, the words are stemmed using the Wordnet stemmer (the last two functions use the implementation provided by the NLTK python package) and any terms that appear in less than three documents in the respective training set are removed. The remaining corpus is then turned to a document-term matrix using the tf-idf weighting scheme. Specifically, the natural frequency of each term is used along with cosine normalization. It should be noted that in order to prevent zero division, idf weights are smoothed by adding one to document frequencies as if an extra document was seen containing every term in the collection exactly once. In this way, terms with zero idf, i.e. that occur in all documents of a training set, will not be entirely ignored. For the voice recordings dataset, the data were transformed in order to be in the space [0, 1] using the formula:

$$(x_{ij} - \min_i x_j)/(\max_i x_j - \min_i x_j)$$

For each of the text datasets and using each of the above methods, the dimensionality of the dataset is reduced to 2, 5, 10, 50, 100, 200, 300, 500 and 1000. A linear Support Vector Machine (SVM) classifier is trained on each dataset and predictions for the respective test dataset are created. SVMs are very commonly used in the context of text classification and are considered appropriate tasks where the input space is of high dimensionality and the input vectors are sparse (Thorsten, 1998). A linear kernel is used since classes in such high-space classification problems are considered to be linearly separable and it is also more efficient to train the classifier. The LinearSVC class from python's library sklearn is used in this study and is based on the liblinear implementation. Multiclass classification tasks are handled according to a one-vs-rest scheme. No optimization on SVM parameters is performed. The default parameters are kept: C=1 penalty parameter with l2-norm penalization using the squared hinge loss function.

For the voice recordings dataset, only feature extraction algorithms were used since the feature selection methods are applicable to count data and not to numeric features. 50, 100, 200, 300 and 500 dimensions were extracted using SVD, NMF, GRP and SRP. LDA was not evaluated because it was not considered meaningful to compare the algorithms by extracting only one dimension (the maximum number of dimensions that LDA can produce in binary classification tasks). It should also be noted that the libsvm implementation of the SVM algorithm was used in this case, again with a linear kernel and default parameters, apart from the class weight which was set to be inversely proportional to the class frequencies in order to improve the classifier's performance.

The metrics used for the evaluation of the classifiers are the F1-score, classification accuracy, precision and recall. Each metric is calculated for each label and their average is taken, weighted by support (the number of true instances for each label); this takes into account the class imbalance. For sampling-based feature selection methods, namely WBS and SS, due to the randomness of the procedure, each iteration was repeated 10 times and the best results were kept. For NMF, GRP and SPR a random seed has been used so that the results are reproducible.

3.1 Dataset description

Three text datasets were firstly evaluated and were all obtained from (Cardoso Cachopo, 2007). All datasets had already a train/test split and these were used for evaluation of the SVM classifiers. Then, another dataset provided by the thesis supervisor was used and contained data captured from voice recordings. Telephone communications were broken-down to sub-second chunks and several statistics were calculated for each one of them. Finally, the dataset was manually annotated based on whether the speaker was angry or not at each time.

The first one is the '20 Newsgroups' dataset which is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. The articles are typical postings and thus have headers including subject lines, signature files, and quoted portions of other articles.

The other two datasets consist of documents contained in the 'Reuters-21578' collection appeared on the Reuters newswire and were manually classified by personnel from Reuters Ltd. This collection is very skewed, with documents very unevenly distributed among different classes. Due to the fact that the class distribution for these documents is very skewed, sub-collections of the dataset are usually considered for text categorization tasks. Moreover, many of these documents are classified as having no topic at all or with more than one topic. In this thesis, the so-called Reuters8 and Reuters52 datasets were used.

The last dataset, called Emotient, was highly skewed and therefore had to be preprocessed before applying any dimensionality reduction and/or classification technique. Thus, oversampling and undersampling was performed on the data. This was performed by thesis supervisor's team before being provided to the writer of this thesis. Thus, the resulting training dataset included 14.088 cases while the test dataset included 2.346 cases equally distributed between the two classes. The original dataset comprised of 768 dimensions with numerical data which where normalized in order to be in the same range (i.e. between 0 and 1) so no single feature can affect the SVM classifier due to its scale. Due to the high skewness of the original dataset, the accuracy is not considered as an appropriate metric. Therefore, precision and recall (and consequently F1-score) are deemed more appropriate.

4. Results

For each of the four datasets that were examined in this study, the F1-score of the classification after the pre-processing with each dimensionality reduction algorithm will be presented below. In order for the graphs to be more readable, dimension selection and feature extraction techniques will be presented separately. The results of the classification without any dimensionality reduction will also be presented as a benchmark.

The exact F1-scores along with the accuracy, precision and recall are presented in Appendix A while the code used is shown in Appendix B.

4.1 Newsgroup Dataset

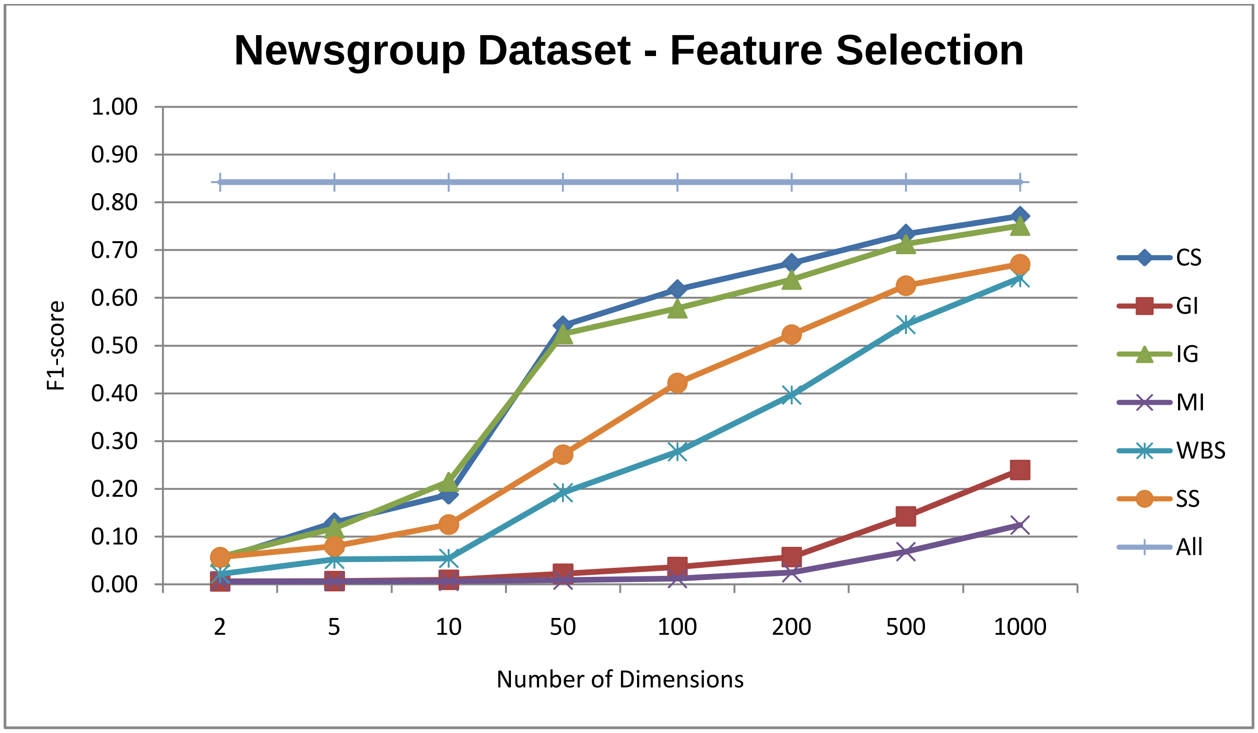


Figure 4: Newsgroup Dataset - Feature Selection results (source: this study)

The original dimensionality of the Newsgroup dataset was 27.637 dimensions, the largest dataset examined in this study, in terms of the number of features. The F1-score achieved by training the SVM classifier on the full dataset was 0.842 and none of the dimension reduction techniques managed to reach that score. It is evident that the score achieved by each feature selection algorithm increases as the number of features kept also increases. CS and IG were consistently the best-performing methods reaching a best score of 0.771 keeping 1000 features. It is interesting that although SS and WBS are sampling-based feature selection algorithms, they performed adequately well reaching 0.671. It should be noted that SS performed better across all dimensions showing that the corresponding low-rank right singular vectors capture some semantic information. GI and MI performed poorly and did not exceed 0.24. It is not clear why these methods did not perform well.

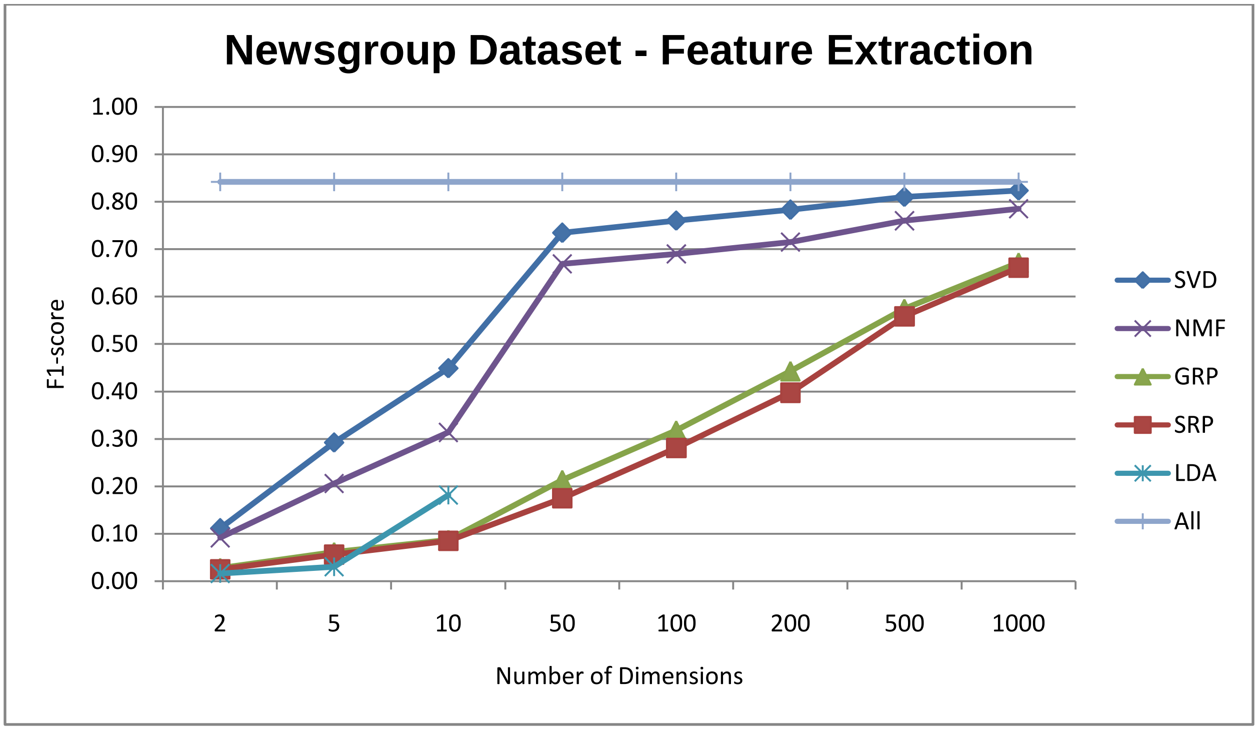


Figure 5: Newsgroup Dataset - Feature Extraction results (source: this study)

Feature extractions techniques proved to be better performing than the feature selection techniques. Specifically, SVD reached a score of 0.824 (score will all dimensions was 0.842). It is worth noting that SVD also had and F1-score equal to 0.734 with just 50 dimensions, which is higher than what IG and CS achieved with 500 dimensions. NMF proved to be performing well, however always lower than SVD. NMF also took considerable time for the algorithm to converge. Random Projections performed less well but were extremely fast to calculate so they could easily scale to much larger datasets. LDA performed poorly and seems not to be appropriate for textual data since it cannot produce enough dimensions to capture the information on a dataset with such a high dimensionality.

4.2 Reuters8 Dataset

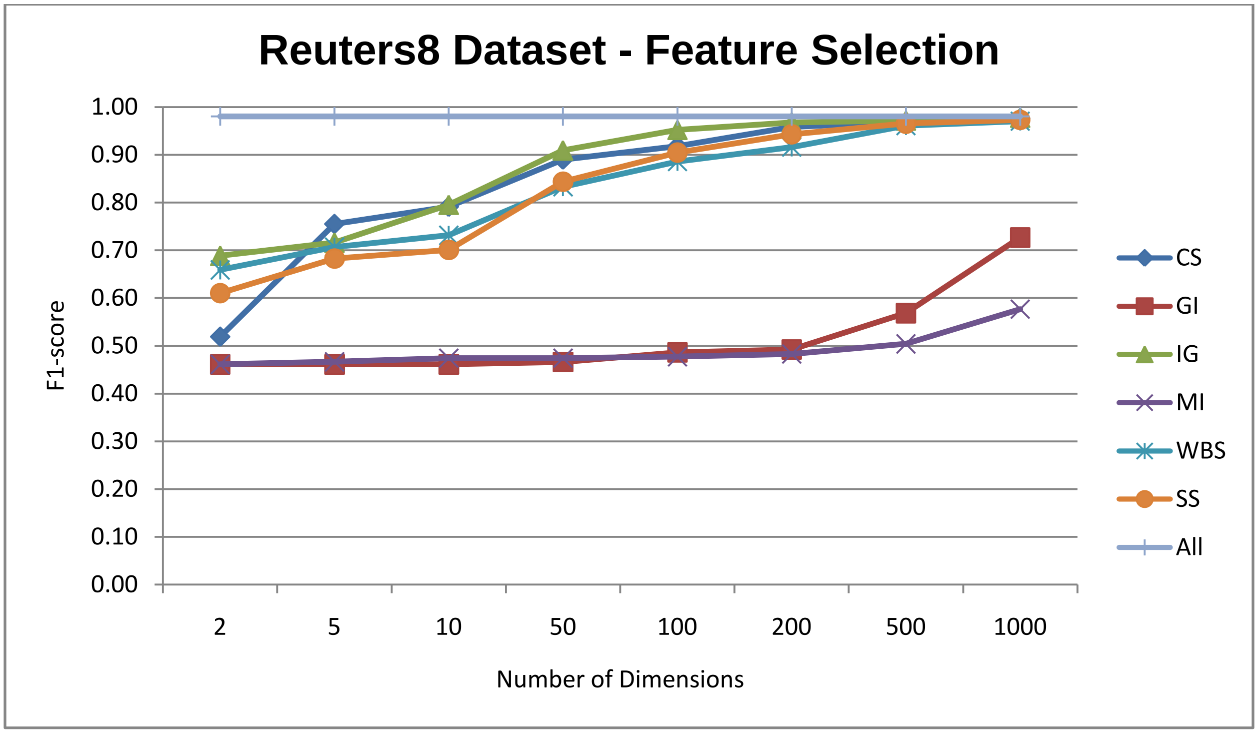


Figure 6: Reuters8 Dataset - Feature Selection results (source: this study)

Reuters8 dataset had a much lower dimensionality (6.820 features) and an SVM trained on all features achieved an almost perfect classification reaching 0.981 in F1-score. Many feature selection methods reached very close to this benchmark. Specifically, CS, IG, SS and WBS exceeded the score of 0.97 with just 1000 features kept and therefore it's not clear what the best method was. Although the difference was very small, IC and CS proved to be the best performing methods in this dataset as well. Similarly to what was observed on the Newsgroup dataset, GI and MI performed very poorly and showed no significant improvement when selecting up to 200 features.

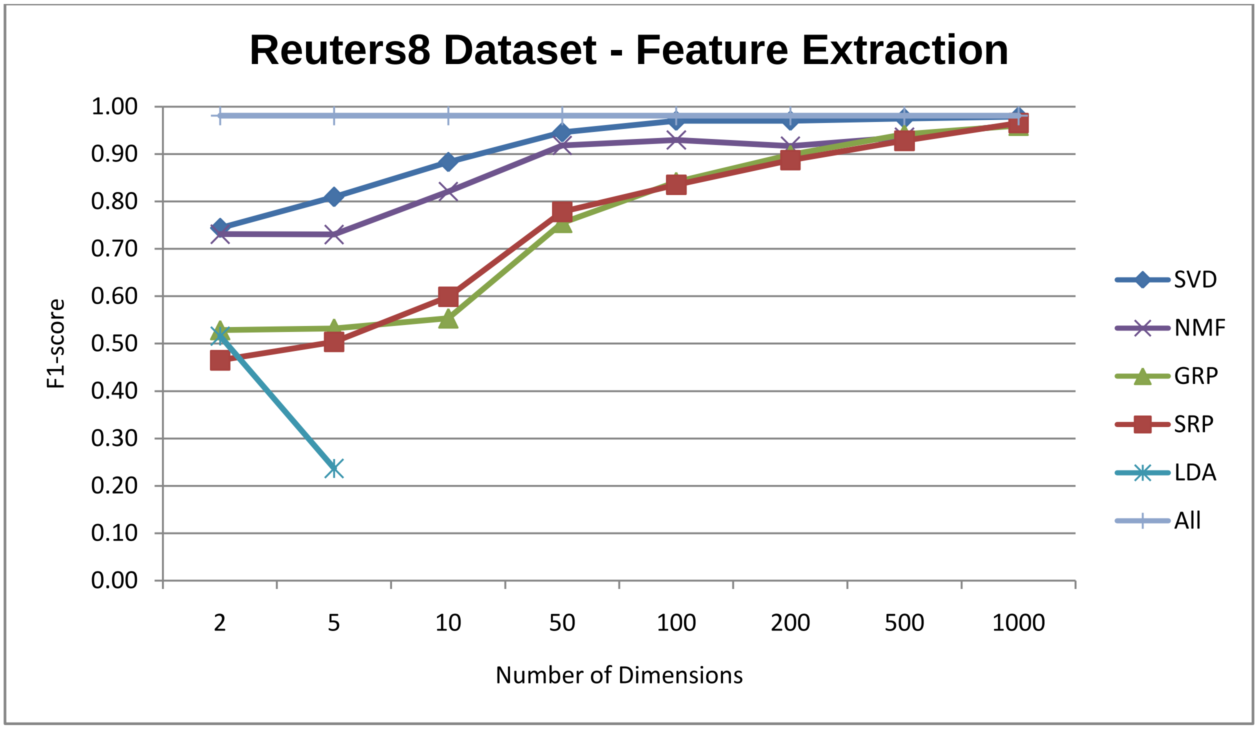


Figure 7: Reuters8 Dataset - Feature Extraction results (source: this study)

SVD was again the best performing dimension extraction method, reaching a score of 0.97 with just 100 dimensions. With 1000 dimensions, its performance was 0.979, just below the classification score without any dimensionality reduction. NMF performed very well in low dimensions but it was reached by SRP and GRP when more than 500 dimensions were kept. However, none of these 3 methods surpassed the 0.97 threshold and therefore performed somewhat worse that the best feature selection algorithms. It is surprising that a score equal to 0.744 was achieved using SVD by extracting only two dimensions. LDA had a poor performance in this dataset as well and seemed not to be appropriate in this case. The drop from 2 to 5 dimensions can probably be attributed to the particularity of the specific dataset.

4.3 Reuters52 Dataset

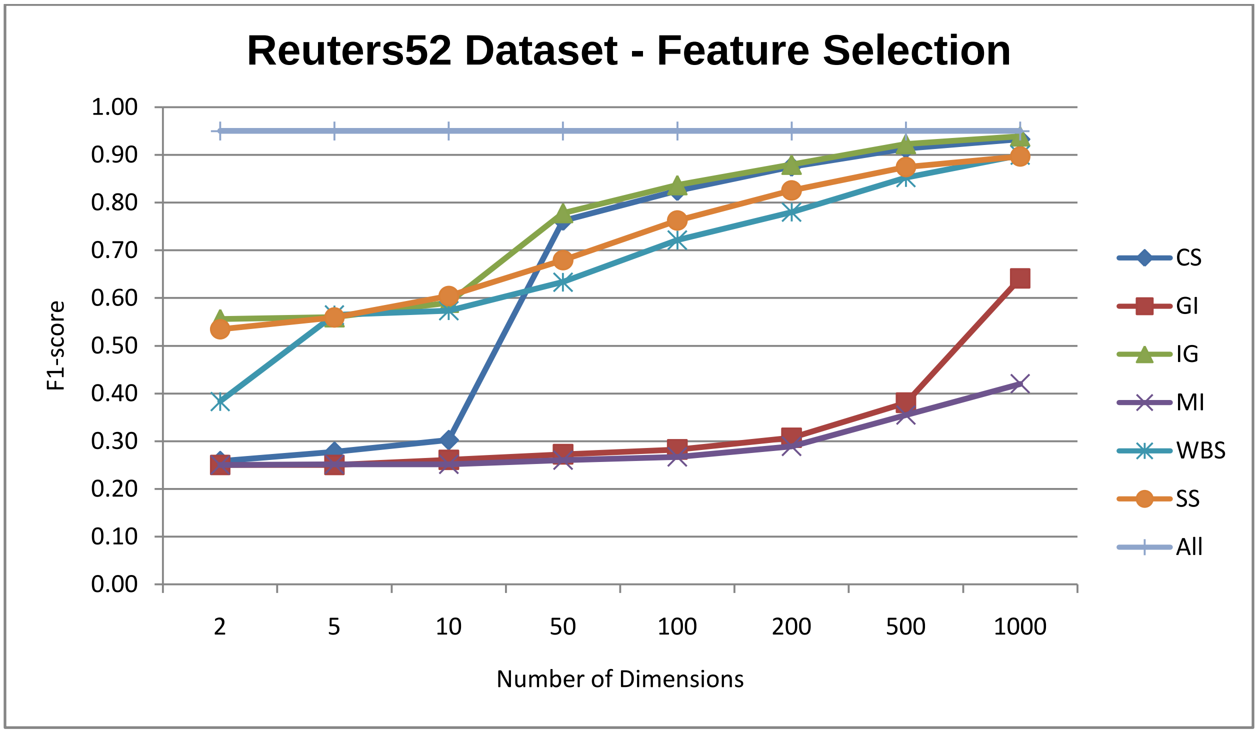


Figure 8: Reuters52 Dataset - Feature Selection results (source: this study)

This dataset is similar to Reuters8 dataset and had 7.855 features in total. The F1-score of the SVM classifier on the full dataset was 0.95 and again it was not reached by any of the dimensionality reduction techniques. The results were consistent with those of the other datasets; CS and IG again performed better than the other methods with IG reaching a score of 0.939 using 1000 features. SS and WBS performed lower with SS being again better than WBS while the highest score was just below 0.9. GI and MI performed poorly again. It is remarkable that MI with 1000 dimensions had a lower score than what other methods achieved with just 2 features.

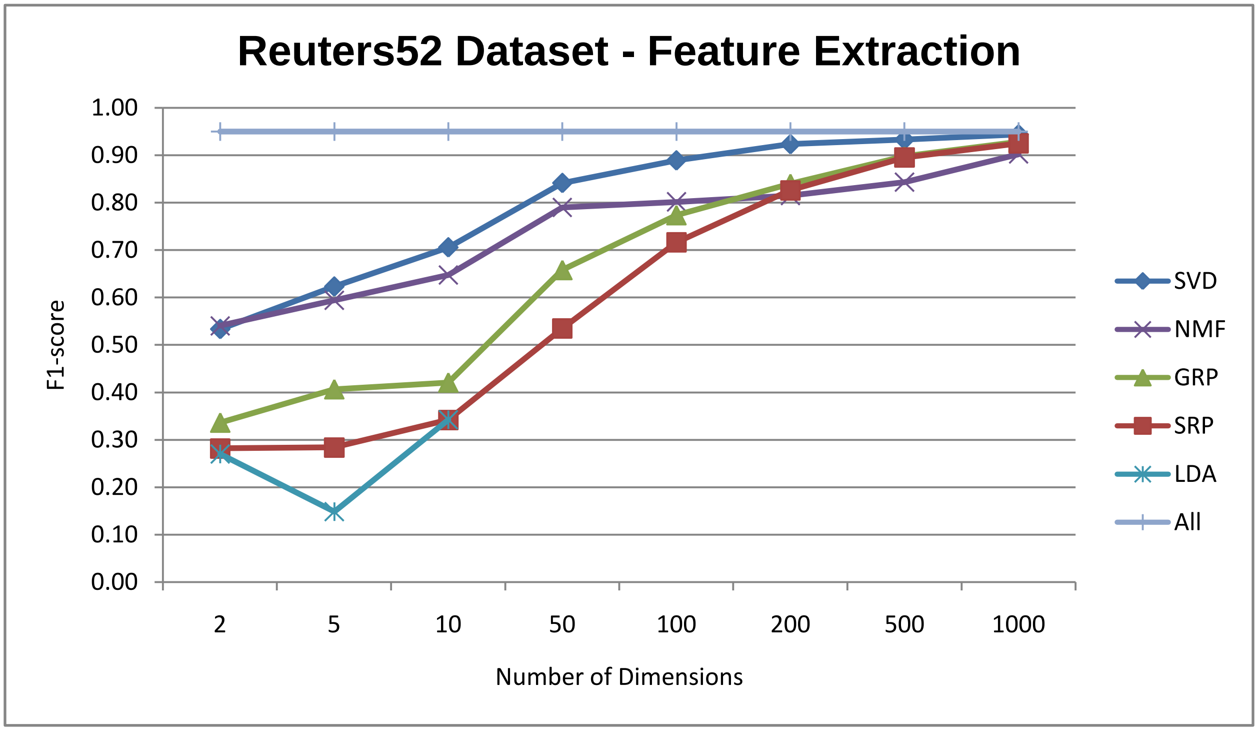


Figure 9: Reuters52 Dataset - Feature Extraction results (source: this study)

Similarly to Reuters8 dataset, SVD performed better than other methods reaching a score of 0.944. After extracting just 200 dimensions, it achieved a score of 0.923. NMF performed well in low dimensions but was again reached by the Random Projection methods as the dimensionality was increasing. In fact, both GRP and SPR surpassed the scores achieved by NMF for more than 500 dimensions. The performance of LDA was similar to what was observed in the previous datasets.

4.4 Emotient Dataset

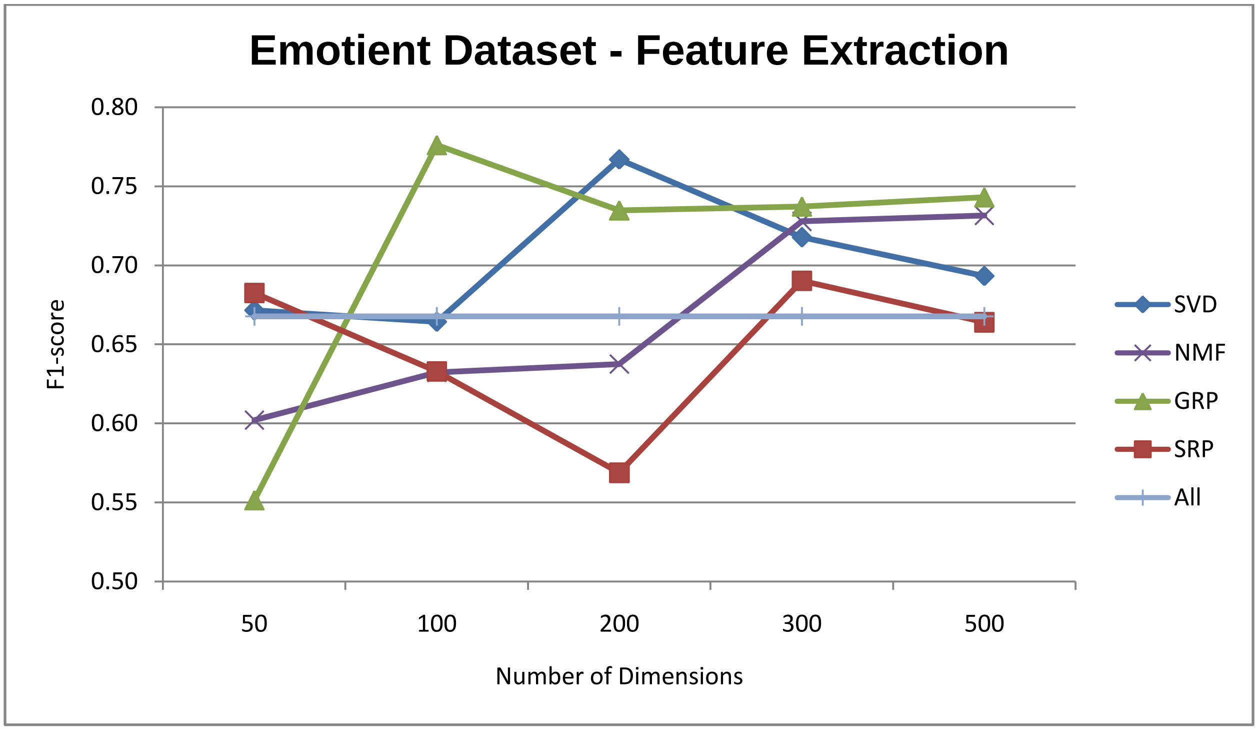


Figure 10: Emotient Dataset - Feature Extraction results (source: this study)

The Emotient dataset was different from the other three examined in this study since it was not referring to text data. Voice recordings and the various statistics calculated on each sub-second part of them, were expected to have increased redundancy. Therefore, feature extraction techniques were expected to provide increased classification performance.

The original dimensionality of this dataset was 768 features and the SVM classifier reached an F1-score of 0.668 (please note the different scale on the y-axis on the above chart - used for increased clarity). The results on this dataset had more variance (compared to the previous datasets) with no algorithm being the clear winner. It is interesting that GRP achieved the highest score (0.776) with just 100 dimensions while SVD was the second-best reaching 0.767 with 200 dimensions. In addition, all methods exceeded the score of the classifier using all features albeit with different dimensions kept. This shows that indeed there is increased redundancy in the data that was captured by the low-rank approximations. Notable is also the fact that the performance of NMF kept increasing with the additional dimensions whereas the other algorithms did not have a monotonic relationship with the number of dimensions.

5. Conclusions

This study firstly described the current methods used for representing textual data and then explained the main algorithms used for dimensionality reduction in text and voice data. All the techniques presented have been extensively used in the literature for effective reduction of the computed features and have led not only to faster computations of the following data processing (due to reduced size) but also sometimes to increased performance since noisy features are discarded or downgraded. This study also examined feature selection and feature extraction algorithms together while they are usually studied separately.

Through application of the dimension reduction methods to three text datasets, it was evident that effective classification of the text data can be performed by keeping a much lower dimensionality than the original. The best performing method proved to be the SVD which led to increased classification scores even in very low dimensions. The results indicate that using SVD and extracting between 200 and 500 features is an ideal trade-off between classification accuracy and computation time. However, depending on the dimensionality of the dataset, some faster to calculate methods such as Random Projections could provide an adequate classification score and can scale to truly large datasets. From the feature selection methods, CS and IG were the best performing ones but were usually performing worse than SVD when just few features were kept. However, no method performed better than using all features but it could be argued that keeping more dimensions could possibly lead to scores higher than the all-features classifier.

Voice data, on the other hand, proved to have increased redundancy and all feature extraction methods performed better than when all features were used. Surprisingly, GPR performed consistently high with SVD following. It should be noted however that since the test dataset had been made to have equal examples of both classes, the results may be different on skewed data.

With the increased data available nowadays, dimensionality reduction methods are necessary for today's systems to handle this volume. The results of this study can have practical implications to the design of text or voice classification systems and can indicate which feature reduction technique should be used given the needed trade-off between computational complexity and classification accuracy. The study also showed that algorithms with better theoretical foundations (e.g. SVD) can be considered a 'safe choice' but do not always have the best predictive performance; in some cases they could safely be substituted with much more scalable algorithms with little or no sacrifice in classification performance.

As proposals for further research, the presented methods should be tested to larger datasets. Furthermore, sentiment detection in voice recordings is still a research topic and much more experimentation is needed before concluding on the best approach to this task. Moreover, with the increased computational power of the current systems, the cloud computing and/or GPU computing infrastructure available, some new dimensionality reduction techniques should be explored. Specifically, a special type of deep neural networks, called autoencoders, provide hierarchical non-linear dimensionality reduction and have successfully been applied mainly to image recognition tasks. Although the implementation is much more difficult and the training time needed would be much higher than other methods, the predictive performance of the generated features could well justify its use. Lastly, other classification algorithms could be examined since the combined investigation of classifiers and dimension reduction methods could provide interesting results.

References

Achlioptas, D. (2003). Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences* (66), pp. 671-687.

Aggarwal, C., & ChengXiang, Z. (2012). A survey of text classfication algorithms. In C. Aggarwal, & Z. ChengXiang, *Mining Text Data* (pp. 163-222). Springer US.

Aggarwal, C., & Zha, C. A survey of text classfication algorithms.

Berry, M. W. (1996). *The University of Tennesee.* Retrieved 9 14, 2014, from http://web.eecs.utk.edu/~mberry/lsi++/

Bingham, E., & Mannila, H. (2011, 8 17). *Random projection in dimensionality reduction: Applications to image and text data.* Retrieved 9 28, 2014, from Universidade Estadual de Campinas: http://www.ime.unicamp.br/~wanderson/Artigos/randon_projection_kdd.pdf

Boutsidis, C., & Gallopoulos, E. (2007, 4). *SVD based initialization: A head start for nonnegative matrix factorization.* Retrieved 23 9, 2014, from University of Patras: http://scgroup.hpclab.ceid.upatras.gr/faculty/stratis/Papers/HPCLAB020107.pdf

Brazdil, P. (n.d.). *Universidade do Porto.* Retrieved 9 14, 2014, from Representation of Documents and Information Retrieval: http://web.letras.up.pt/bhsmaia/EDV/apresentacoes/Bradzil_ReprDocs_InfRetr.pdf

Cardoso Cachopo, A. (2007, 10 8). Improving Methods for Single-label Text Categorization, PhD Thesis.

Chawla, N., Bowyer, K., Hall, L., & Kegelmeyer, P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research , 16*, pp. 321-357.

Dasgupta, A., Drineas, P., Harb, B., Josifovski, V., & Mahoney, M. W. (2007). *Feature Selection Methods for Text Classification.* Retrieved 9 30, 2014, from Proceedings of the 13-th Annual SIGKDD: http://www.cs.yale.edu/homes/mmahoney/pubs/kdd07.pdf

Dasgupta, S. (2000). *Experiments with random projection.* Retrieved 9 28, 2014, from Proceedings of the 16th Conference on Uncertainty in Artifical Intelligence: http://cseweb.ucsd.edu/~dasgupta/papers/randomf.pdf

Drineas, P., Mahoney, M., & Muthukrishnan, S. (2006). *Subspace Sampling and Relative-Error Matrix Approximation: Column-Based Methods.* Retrieved 9 30, 2014, from Proceeding of the 10-th Annual RANDOM: http://cs-www.cs.yale.edu/homes/mmahoney/pubs/random06.pdf

Dumais, S., Platt, J., Heckerman, D., & Sahami, M. (1998). *Inductive Learning Algorithms and Representations for Text Categorization.* Retrieved 10 8, 2014, from 7th International Conference on Information and Knowledge Management: http://research.microsoft.com/en-us/um/people/jplatt/cikm98.pdf

Forman, G. (2003, 3). An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Journal of Machine Learning Research* , pp. 1289-1305.

Fradkin, D., & Madigan, D. (2003). *Experiments with Random Projections for Machine Learning.* Retrieved 9 28, 2014, from Proceedings of KDD-03, The Ninth International Conference on Knowledge Discovery and Data Mining: http://dimacs.rutgers.edu/Research/MMS/PAPERS/rp.pdf

Gutierrez-Osuna, R. (2002). *Linear discriminant Analysis.* Retrieved 9 27, 2014, from Texas A&M University: http://courses.cs.tamu.edu/rgutier/cs790_w02/l6.pdf

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd Edition).* Springer-Verlag.

Jieping, Y., & Shuiwang, J. (2008). Discriminant Analysis for Dimensionality Reduction: An Overview of Recent Developments. In N. Boulgouris, K. Plataniotis, & E. Micheli-Tzanakou, *Biometrics: Theory, Methods & Applications* (pp. 1-17). IEEE/Wiley.

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling.* New York: Springer.

Kumar, A. (2009). Analysis of Unsupervised Dimensionality Reduction Techniques. *Computer Science and Information Systems/ComSIS , 11* (3), pp. 217-227.

Langville, A., Meyer, C., & Albright, R. (2006). *Initializations for the Nonnegative Matrix Factorization.* Retrieved 9 23, 2014, from NC State University: http://meyer.math.ncsu.edu/meyer/ps_files/nmfinit.pdf

Lee, D., & Seung, S. (2006, 11 2). *Algorithms for Non-negative Matrix Factorization.* Retrieved 9 23, 2014, from Massachusetts Institute of Technology: http://hebb.mit.edu/people/seung/papers/nmfconverge.pdf

Lee, S.-I., Lee, H., Abbeel, P., & Ng, A. (2006). *Efficient L1 Regularized Logistic Regression.* Retrieved 10 1, 2014, from Proceedings of the 21st National Conference on Artificial Intelligence (AAAI): http://web.eecs.umich.edu/~honglak/aaai06_L1logreg.pdf

Li, P., Hastie, T., & Church, K. (2006). *Very Sparse Random Projections.* Retrieved 9 28, 2014, from Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '06): http://web.stanford.edu/~hastie/Papers/Ping/KDD06_rp.pdf

Liberty, E. (2012). *Singular Value Decomposition (SVD) and Principal Component Analysis (PCA).* Retrieved 9 22, 2014, from Yale University: http://www.cs.yale.edu/homes/el327/datamining2012aFiles/06_singular_value_decomposition.pdf

Lin, J., & Gunopulos, D. (2003). *Dimensionality Reduction by Random Projection and Latent Semantic Indexing.* Retrieved 9 28, 2014, from Proceedings of the Text Mining Workshop, at the 3rd SIAM International Conference on Data Mining: http://www.cs.gmu.edu/~jessica/publications/lsi_sdm_workshop03.pdf

Liu, T., Liu, S., Chen, Z., & Ma, W.-Y. (2003). *An Evaluation on Feature Selection for Text Clustering.* Retrieved 9 20, 2014, from Proceedings of the Twentieth International Conference on Machine Learning: http://research.microsoft.com/en-us/people/zhengc/icml2003-15.pdf

Madsen, R. E., Hansen, L. K., & Winther, O. (2004, 2). *Singular Value Decomposition and Principal Component Analysis.* Retrieved 9 22, 2014, from Technical University of Denmark: www2.imm.dtu.dk/pubdb/views/edoc_download.php/4000/pdf

Manning, C., Raghavan, P., & Schütze, H. (2009). *An Introduction to Information Retrieval.* Cambridge, England: Cambridge University Press.

Masaeli, M., Fung, G., & Dy, J. G. (2010). *From Transformation-Based Dimensionality Reduction to Feature Selection.* Retrieved 9 20, 2014, from 27th International Conference on Machine Learning (ICML 2010): http://www.icml2010.org/papers/333.pdf

Mylavarapu, S., & Kaban, A. (2013). *Random projections versus random selection of features for classification of high dimensional data.* Retrieved 9 28, 2014, from Proceedings of the UK Workshop on Computational Intelligence: http://www.cs.bham.ac.uk/~axk/Sachin_ukci13.pdf

Nicolosi, N. (2008, 11 7). *Feature Selection Methods for Text Classification.* Retrieved 9 20, 2014, from B. Thomas Golisano College of Computing and Information Sciences: http://www.cs.rit.edu/~nan2563/feature_selection.pdf

Rajaraman, A., & Ullman, J. (2011). *Dimensionality Reduction.* Cambridge Press.

Rogati, M., & Yang, Y. (n.d.). *High-Performing Feature Selection for Text Classification.* Retrieved 9 18, 2014, from Carnegie Mellon University: http://www.cs.cmu.edu/~dgovinda/pdf/rogati-cikm02.pdf

Sandhya, N., Sri Lalitha, Y., Sowmya, V., Anuradha, K., & Govardhan, A. (2011, 9). Analysis of Stemming Algorithm for Text Clustering. *IJCSI International Journal of Computer Science Issues , 8* (5), pp. 352-359.

Shiry Ghidary, S. (2010, 9 12). *A Tutorial on Data Reduction.* Retrieved 9 27, 2014, from Amirkabir University of Technology: http://ceit.aut.ac.ir/~shiry/lecture/machine-learning/LDA%20Tutorial.pdf

Steinberger, J., & Ježek, K. (2004). *Using Latent Semantic Analysis in Text Summarization and Summary Evaluation.* Retrieved 9 20, 2014, from University of West Bohemia : http://www.kiv.zcu.cz/~jstein/publikace/isim2004.pdf

Technical University of Denmark. (n.d.). Retrieved 9 14, 2014, from http://cogsys.imm.dtu.dk/thor/projects/multimedia/textmining/index.html

Thorsten, J. (1998). *Text Categorization with Support Vector Machines: Learning with Many Irrelevant Features.* Retrieved 10 7, 2014, from Proceedings of the European Conference on Machine Learning (ECML): http://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf

Torkkola, K. (2004, 2 1). Linear Discriminant Analysis in Document Classification. *Formal Pattern Analysis & Applications , 6* (4), pp. 301-308.

Wall, M., Rechtsteiner, A., & Rocha, L. (2003, 3 3). *Singular value decomposition and principal component analysis.* Retrieved 9 22, 2014, from Washington University in St. Louis: http://www.cs.wustl.edu/~zhang/teaching/cs517/Spring12/CourseProjects/SVD.pdf

Xu, Y., Jones, G., Li, J., Wang, B., & Sun, C. (2007). A Study on Mutual Information-based Feature Selection for Text Categorization. *Journal of Computational Information Systems , 3* (3), pp. 1007-1012.

Yang, Y., & Pedersen, J. (n.d.). *A comparative study on feature selection in text categorization.* Retrieved 9 18, 2014, from University of Trento: http://disi.unitn.it/moschitti/Projects/yang97comparative.pdf

Yang, Z., Zhang, H., Yuan, Z., & Oja, E. (2011). *Kullback-Leibler Divergence for Nonnegative Matrix Factorization.* Retrieved 9 23, 2014, from Aalto University: http://users.ics.aalto.fi/rozyang/preprints/icann2011.pdf

Zhu, W., & Lin, Y. (2013). Using Gini-index for Feature Weighting in Text Categorization. *Journal of Computational Information Systems , 9* (14), pp. 5819-5826.

Zou, H., & Hastie, T. (2005, 4). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) , 67* (2), pp. 301-320.

Appendix A - Detailed Results

Newsgroup Dataset

Algorithm	Number of Features	F1-score	Accuracy	Precision	Recall
IG	2	0.057	0.116	0.054	0.116
IG	5	0.117	0.176	0.106	0.176
IG	10	0.215	0.265	0.237	0.265
IG	50	0.524	0.507	0.597	0.507
IG	100	0.578	0.570	0.620	0.570
IG	200	0.638	0.638	0.647	0.638
IG	500	0.713	0.715	0.715	0.715
IG	1000	0.751	0.753	0.753	0.753
CS	2	0.053	0.089	0.074	0.089
CS	5	0.130	0.152	0.186	0.152
CS	10	0.188	0.220	0.225	0.220
CS	50	0.542	0.525	0.656	0.525
CS	100	0.618	0.609	0.674	0.609
CS	200	0.673	0.662	0.710	0.662
CS	500	0.733	0.731	0.746	0.731
CS	1000	0.771	0.772	0.776	0.772
MI	2	0.006	0.053	0.036	0.053
MI	5	0.006	0.053	0.028	0.053
MI	10	0.007	0.054	0.027	0.054
MI	50	0.009	0.055	0.021	0.055
MI	100	0.012	0.057	0.021	0.057
MI	200	0.025	0.068	0.043	0.068
MI	500	0.068	0.102	0.122	0.102
MI	1000	0.124	0.154	0.202	0.154
GI	2	0.006	0.053	0.107	0.053
GI	5	0.007	0.054	0.261	0.054
GI	10	0.010	0.055	0.347	0.055
GI	50	0.022	0.062	0.434	0.062
GI	100	0.036	0.069	0.582	0.069
GI	200	0.057	0.082	0.613	0.082
GI	500	0.142	0.132	0.771	0.132
GI	1000	0.240	0.198	0.765	0.198
SVD	2	0.111	0.171	0.110	0.171
SVD	5	0.292	0.366	0.312	0.366
SVD	10	0.449	0.497	0.466	0.497
SVD	50	0.734	0.741	0.742	0.741

SVD	100	0.761	0.765	0.764	0.765
SVD	200	0.783	0.786	0.786	0.786
SVD	500	0.810	0.812	0.812	0.812
SVD	1000	0.824	0.826	0.826	0.826
SPR	2	0.025	0.059	0.031	0.059
SPR	5	0.056	0.088	0.123	0.088
SPR	10	0.085	0.105	0.096	0.105
SPR	50	0.175	0.191	0.181	0.191
SPR	100	0.281	0.296	0.282	0.296
SPR	200	0.397	0.405	0.397	0.405
SPR	500	0.559	0.561	0.558	0.561
SPR	1000	0.661	0.662	0.663	0.662
GRP	2	0.027	0.062	0.019	0.062
GRP	5	0.062	0.083	0.084	0.083
GRP	10	0.087	0.105	0.096	0.105
GRP	50	0.213	0.230	0.223	0.230
GRP	100	0.318	0.332	0.322	0.332
GRP	200	0.443	0.448	0.442	0.448
GRP	500	0.575	0.577	0.574	0.577
GRP	1000	0.671	0.673	0.674	0.673
NMF	2	0.091	0.153	0.093	0.153
NMF	5	0.206	0.296	0.228	0.296
NMF	10	0.314	0.393	0.339	0.393
NMF	50	0.669	0.678	0.679	0.678
NMF	100	0.690	0.697	0.700	0.697
NMF	200	0.715	0.721	0.724	0.721
NMF	500	0.760	0.764	0.764	0.764
NMF	1000	0.785	0.789	0.788	0.789
LDA	2	0.016	0.058	0.012	0.058
LDA	5	0.030	0.060	0.027	0.060
LDA	10	0.181	0.214	0.237	0.214
LDA	50	0.170	0.202	0.235	0.202
LDA	100	0.182	0.216	0.232	0.216
LDA	200	0.161	0.194	0.238	0.194
LDA	500	0.159	0.189	0.248	0.189
LDA	1000	0.163	0.192	0.256	0.192
SS	2	0.057	0.105	0.091	0.105
SS	5	0.080	0.110	0.126	0.110
SS	10	0.125	0.139	0.282	0.139
SS	50	0.272	0.287	0.318	0.287
SS	100	0.422	0.416	0.465	0.416

SS	200	0.523	0.523	0.543	0.523
SS	500	0.625	0.625	0.634	0.625
SS	1000	0.671	0.672	0.675	0.672
WBS	2	0.022	0.067	0.061	0.067
WBS	5	0.052	0.087	0.104	0.087
WBS	10	0.055	0.102	0.096	0.102
WBS	50	0.192	0.203	0.277	0.203
WBS	100	0.278	0.282	0.311	0.282
WBS	200	0.397	0.400	0.412	0.400
WBS	500	0.544	0.545	0.554	0.545
WBS	1000	0.642	0.643	0.650	0.643
All Dimensions	27637	0.842	0.843	0.845	0.843

Reuters8 Dataset

Algorithm	Number of Features	F1-score	Accuracy	Precision	Recall
IG	2	0.689	0.720	0.712	0.720
IG	5	0.716	0.754	0.715	0.754
IG	10	0.795	0.807	0.811	0.807
IG	50	0.909	0.916	0.910	0.916
IG	100	0.952	0.953	0.955	0.953
IG	200	0.967	0.968	0.969	0.968
IG	500	0.972	0.972	0.973	0.972
IG	1000	0.978	0.978	0.978	0.978
CS	2	0.519	0.653	0.433	0.653
CS	5	0.755	0.757	0.813	0.757
CS	10	0.791	0.794	0.833	0.794
CS	50	0.890	0.888	0.902	0.888
CS	100	0.918	0.917	0.925	0.917
CS	200	0.959	0.959	0.961	0.959
CS	500	0.969	0.969	0.970	0.969
CS	1000	0.976	0.976	0.977	0.976
MI	2	0.461	0.609	0.371	0.609
MI	5	0.467	0.612	0.382	0.612
MI	10	0.474	0.619	0.385	0.619
MI	50	0.474	0.619	0.385	0.619
MI	100	0.477	0.621	0.393	0.621
MI	200	0.483	0.622	0.433	0.622
MI	500	0.504	0.630	0.555	0.630
MI	1000	0.577	0.671	0.662	0.671
GI	2	0.461	0.609	0.371	0.609
GI	5	0.461	0.609	0.371	0.609
GI	10	0.461	0.609	0.371	0.609
GI	50	0.466	0.609	0.497	0.609
GI	100	0.486	0.615	0.548	0.615
GI	200	0.493	0.615	0.533	0.615
GI	500	0.568	0.652	0.652	0.652
GI	1000	0.727	0.737	0.818	0.737
SVD	2	0.744	0.792	0.715	0.792
SVD	5	0.809	0.830	0.816	0.830
SVD	10	0.883	0.900	0.880	0.900
SVD	50	0.946	0.951	0.942	0.951
SVD	100	0.970	0.971	0.971	0.971
SVD	200	0.970	0.971	0.971	0.971
SVD	500	0.975	0.975	0.976	0.975

SVD	1000	0.979	0.979	0.980	0.979
SRP	2	0.465	0.606	0.454	0.606
SRP	5	0.504	0.603	0.472	0.603
SRP	10	0.599	0.640	0.577	0.640
SRP	50	0.778	0.786	0.780	0.786
SRP	100	0.835	0.835	0.840	0.835
SRP	200	0.887	0.888	0.891	0.888
SRP	500	0.928	0.928	0.932	0.928
SRP	1000	0.966	0.966	0.966	0.966
GRP	2	0.529	0.612	0.491	0.612
GRP	5	0.532	0.584	0.490	0.584
GRP	10	0.553	0.605	0.511	0.605
GRP	50	0.755	0.768	0.753	0.768
GRP	100	0.841	0.839	0.846	0.839
GRP	200	0.899	0.898	0.901	0.898
GRP	500	0.942	0.942	0.942	0.942
GRP	1000	0.960	0.960	0.960	0.960
NMF	2	0.731	0.771	0.724	0.771
NMF	5	0.730	0.774	0.713	0.774
NMF	10	0.821	0.844	0.815	0.844
NMF	50	0.918	0.925	0.918	0.925
NMF	100	0.930	0.934	0.937	0.934
NMF	200	0.917	0.923	0.929	0.923
NMF	500	0.936	0.938	0.941	0.938
NMF	1000	0.963	0.963	0.964	0.963
LDA	2	0.516	0.481	0.587	0.481
LDA	5	0.237	0.243	0.482	0.243
LDA	10	0.571	0.537	0.629	0.537
LDA	50	0.557	0.529	0.607	0.529
LDA	100	0.580	0.578	0.604	0.578
LDA	200	0.387	0.350	0.548	0.350
LDA	500	0.372	0.338	0.514	0.338
LDA	1000	0.206	0.226	0.340	0.226
SS	2	0.611	0.658	0.578	0.658
SS	5	0.683	0.721	0.672	0.721
SS	10	0.701	0.732	0.719	0.732
SS	50	0.844	0.860	0.855	0.860
SS	100	0.905	0.904	0.916	0.904
SS	200	0.943	0.947	0.947	0.947
SS	500	0.965	0.966	0.966	0.966
SS	1000	0.973	0.973	0.975	0.973

WBS	2	0.659	0.684	0.713	0.684
WBS	5	0.707	0.730	0.732	0.730
WBS	10	0.732	0.752	0.782	0.752
WBS	50	0.833	0.841	0.865	0.841
WBS	100	0.886	0.897	0.885	0.897
WBS	200	0.916	0.920	0.922	0.920
WBS	500	0.961	0.962	0.962	0.962
WBS	1000	0.970	0.971	0.971	0.971
All Dimensions	6820	0.981	0.981	0.981	0.981

Reuters52 Dataset

Algorithm	Number of Features	F1-score	Accuracy	Precision	Recall
IG	2	0.556	0.650	0.514	0.650
IG	5	0.560	0.657	0.511	0.657
IG	10	0.590	0.683	0.539	0.683
IG	50	0.778	0.822	0.756	0.822
IG	100	0.837	0.865	0.823	0.865
IG	200	0.879	0.899	0.870	0.899
IG	500	0.923	0.931	0.921	0.931
IG	1000	0.939	0.945	0.938	0.945
CS	2	0.258	0.428	0.187	0.428
CS	5	0.278	0.445	0.206	0.445
CS	10	0.302	0.465	0.230	0.465
CS	50	0.762	0.791	0.764	0.791
CS	100	0.825	0.840	0.841	0.840
CS	200	0.875	0.883	0.883	0.883
CS	500	0.913	0.918	0.917	0.918
CS	1000	0.933	0.938	0.934	0.938
MI	2	0.250	0.422	0.178	0.422
MI	5	0.251	0.423	0.179	0.423
MI	10	0.251	0.423	0.179	0.423
MI	50	0.260	0.428	0.193	0.428
MI	100	0.267	0.433	0.212	0.433
MI	200	0.289	0.444	0.504	0.444
MI	500	0.355	0.483	0.533	0.483
MI	1000	0.420	0.520	0.524	0.520
GI	2	0.250	0.422	0.178	0.422
GI	5	0.250	0.422	0.178	0.422
GI	10	0.261	0.427	0.226	0.427
GI	50	0.273	0.432	0.457	0.432
GI	100	0.283	0.436	0.455	0.436
GI	200	0.307	0.449	0.463	0.449
GI	500	0.381	0.490	0.517	0.490
GI	1000	0.641	0.691	0.698	0.691
SVD	2	0.533	0.633	0.469	0.633
SVD	5	0.623	0.709	0.573	0.709
SVD	10	0.706	0.766	0.673	0.766
SVD	50	0.841	0.872	0.830	0.872
SVD	100	0.889	0.907	0.887	0.907
SVD	200	0.923	0.933	0.920	0.933
SVD	500	0.933	0.940	0.932	0.940

SVD	1000	0.944	0.948	0.944	0.948
SRP	2	0.282	0.434	0.298	0.434
SRP	5	0.284	0.424	0.260	0.424
SRP	10	0.342	0.427	0.304	0.427
SRP	50	0.535	0.585	0.562	0.585
SRP	100	0.716	0.741	0.710	0.741
SRP	200	0.826	0.837	0.829	0.837
SRP	500	0.895	0.901	0.897	0.901
SRP	1000	0.925	0.930	0.926	0.930
GRP	2	0.336	0.456	0.303	0.456
GRP	5	0.407	0.509	0.342	0.509
GRP	10	0.421	0.518	0.367	0.518
GRP	50	0.658	0.693	0.662	0.693
GRP	100	0.774	0.793	0.774	0.793
GRP	200	0.840	0.851	0.837	0.851
GRP	500	0.898	0.905	0.899	0.905
GRP	1000	0.928	0.933	0.927	0.933
NMF	2	0.540	0.639	0.480	0.639
NMF	5	0.594	0.688	0.539	0.688
NMF	10	0.647	0.719	0.636	0.719
NMF	50	0.790	0.836	0.757	0.836
NMF	100	0.802	0.842	0.779	0.842
NMF	200	0.815	0.849	0.815	0.849
NMF	500	0.843	0.871	0.845	0.871
NMF	1000	0.902	0.916	0.907	0.916
LDA	2	0.270	0.349	0.222	0.349
LDA	5	0.149	0.123	0.298	0.123
LDA	10	0.342	0.382	0.383	0.382
LDA	50	0.393	0.410	0.398	0.410
LDA	100	0.245	0.216	0.310	0.216
LDA	200	0.379	0.402	0.388	0.402
LDA	500	0.360	0.371	0.368	0.371
LDA	1000	0.326	0.323	0.363	0.323
SS	2	0.534	0.611	0.532	0.611
SS	5	0.559	0.641	0.565	0.641
SS	10	0.604	0.672	0.605	0.672
SS	50	0.679	0.742	0.658	0.742
SS	100	0.763	0.799	0.748	0.799
SS	200	0.826	0.855	0.811	0.855
SS	500	0.875	0.892	0.865	0.892
SS	1000	0.897	0.908	0.893	0.908

WBS	2	0.383	0.454	0.337	0.454
WBS	5	0.565	0.643	0.547	0.643
WBS	10	0.574	0.639	0.575	0.639
WBS	50	0.634	0.704	0.610	0.704
WBS	100	0.721	0.773	0.691	0.773
WBS	200	0.780	0.822	0.762	0.822
WBS	500	0.852	0.873	0.844	0.873
WBS	1000	0.899	0.910	0.896	0.910
All Dimensions	7855	0.950	0.954	0.950	0.954

Emotient Dataset

Algorithm	Number of Features	F1-score	Accuracy	Precision	Recall
SVD	50	0.672	0.730	0.856	0.552
SVD	100	0.664	0.721	0.833	0.552
SVD	200	0.767	0.787	0.846	0.702
SVD	300	0.718	0.753	0.837	0.628
SVD	500	0.693	0.738	0.835	0.592
SRP	50	0.682	0.725	0.809	0.590
SRP	100	0.633	0.701	0.819	0.516
SRP	200	0.569	0.665	0.796	0.442
SRP	300	0.690	0.734	0.828	0.592
SRP	500	0.664	0.719	0.824	0.556
GRP	50	0.552	0.638	0.725	0.445
GRP	100	0.776	0.786	0.816	0.740
GRP	200	0.735	0.760	0.820	0.666
GRP	300	0.737	0.763	0.826	0.666
GRP	500	0.743	0.770	0.841	0.666
NMF	50	0.602	0.683	0.810	0.479
NMF	100	0.632	0.699	0.815	0.517
NMF	200	0.638	0.706	0.832	0.517
NMF	300	0.728	0.765	0.863	0.629
NMF	500	0.732	0.769	0.875	0.628
All Dimensions	768	0.668	0.723	0.836	0.556

Appendix B - Source Code

The python scripts used for the evaluation of the algorithms presented in this study, can be found on the following public repository:
https://github.com/asstergi/Big-Data-and-Business-Analytics-Thesis-AUEB

It should be noted that some results may not be reproducible since some of the algorithms examined depend on random sampling.