On the Role of Dimensionality Reduction

Nadia Abd-Alsabour*
Cairo University, Cairo, Egypt.

* Corresponding author. Email: nadia.abdalsabour@cu.edu.eg
Manuscript submitted May 5, 2017; accepted July 20, 2017.
doi: 10.17706/jcp.13.5.571-579

Abstract: Due to the need for the dimensionality reduction in many applications such as real-world and large scale applications, this paper demonstrates it, and covers it from all of its perspectives such as its components, its importance, and where it is needed. Moreover, this paper demonstrates the impact of irrelevant features on the performance of many predictors such as decision tree, Naïve Bayes, nearest neighbor, and support vector machines. This is because the performance of these predictors degenerates when provided with data containing irrelevant features.

Key words: Dimensionality reduction, feature selection, feature extraction, instance reduction.

1. Introduction

The input to a data mining task is usually a dataset that is a collection of features (dimensions, attributes, or variables) with their values. Each dataset describes an application and each line in it describes an instance (object, individual, transaction, or entity) [1]. Some datasets are so large that makes many algorithms have problems to just enter this data. One reason for having a large number of features is that the collected data are not solely for one particular task. Another reason is that a data mining task may require data from different sources (such as databases, data cubes, or files) and the integration of these sources (to constitute a dataset) can be large [2], [3]. Fig. 1 [2] shows this process.

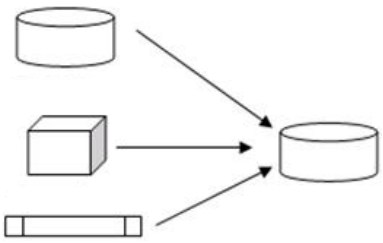

Fig. 1. Data integration [2].

In theory, having more features should result in more discriminating power but this is not always valid in practice [4]. There are many examples that show that not all of the features in a dataset are necessary for accurate prediction. Moreover, including these features can affect the prediction accuracy. For example, a feature may be irrelevant (i.e., does not contribute to the prediction accuracy) such as the month of a person birth for many applications and the day of the week on which a loan application was completed. Besides, a feature may be redundant (i.e., more than one feature contain the same predictive information that is a feature is redundant when it can be derived from another feature or set of features). An example of

redundant features is the annual revenue. Another example of redundant features is the income feature before and after the tax (they are highly correlated). These features may mislead the learning step (which results in poor quality of the discovered patterns). In addition, the added volume (because of the irrelevant features) can slow down the mining process. Moreover, mining on a reduced set of features leads to reducing the number of the discovered patterns which helps understand these patterns more easily. The aforementioned notes raise the necessity of adding the phase of *features reduction* (also called column reduction) in order to improve the prediction accuracy [1]-[3], [5]-[7]. Besides features reduction, dimensionality reduction also includes instance reduction (deleting a row from the given dataset) [8]. Dimensionality reduction helps all phases of the data mining process as it aims at the sake of computation efficiency. Subsequently, it has to be started in the preprocessing phase although in some applications features reduction can be part of the data mining algorithm [5]. The resulted reduced dataset should maintain the integrity of the original data [2].

In many applications, there are large number of samples with different types of features and these samples are often high dimensional (have large number of features). This results in the *curse of dimensionality* problem which means the amount of the needed data increases exponentially with the dimensionality [5], [6]. The reason for that is when the number of the input features increases, the spread of the data also increases which leads to the difficulty of processing such data [9], [10]. That is why instance reduction is considered one way for dimensionality reduction. Predictors that perform efficiently in low dimensions cannot give meaningful results when the number of instances goes beyond the modest size of ten features. An example of these predictors is the nearest neighbors classifier [11].

Therefore, the three characteristic steps of the data mining process can be defined as follows [9]:

- Exploring (preparing/preprocessing) the data. Dimension reduction comes here besides the other preprocessing steps,
- Building and validating the model, and
- Applying the model to new data.

In this paper, dimensionality reduction is explained as well as its components and the differences among them, its applications, and the situations need it. The negative impact of irrelevant features on the performance of some of the most known predictors is demonstrated as well.

The rest of this paper is organized as follows. Section two explains dimensionality reduction and its components. Section three demonstrates the effect of irrelevant features on many predictors. Finally, Section 4 concludes this paper and highlights the future work in this research area.

2. Dimensionality Reduction

In real world data mining applications, much effort is made to preprocess (prepare) the data than to apply data mining methods. The beginning of preparing the given data is to assure their quality. This is because data may contain noise, duplicate data, missing values, anomalies/outliers, incorrectly recorded data, expired data, etc. There are specific methods for each type of these quality problems. For example, for dealing with extreme values, one approach is to remove them; another approach is to use alternative methods that are not sensitive to these values. Another example is dealing with missing values; there are many approaches for this quality problem such as eliminating the objects with missing values, estimating these values, replacing them with other values (mean/median …), etc. [9]. These methods are not mutually exclusive i.e., they may work together [2]. After that, dimensionality reduction can be performed on the resulted data.

Usually, the above-mentioned pre-processing steps are sufficient. However, when facing large datasets, adding the additional step of dimensionality reduction (also called dimension reduction) becomes

necessary and this should be prior to applying the used data mining method. Dimensionality reduction is one aspect of preparing the initial data (which is considered one of the most crucial steps in any data mining task) [5]. It leads to reducing the amount of memory and time required for data processing, eliminating the irrelevant features, and decreasing the noise in such data. Having more data (features or samples) requires much time to build and apply a model to new data. Besides, some features are more predictive than others. Noisy and/or non-predictive features can negatively affect the quality of the models that were built using them [9]-[12].

The importance of dimensionality reduction comes from being a necessary step in many real-world applications such as text categorization, customer relationship management, gene expression microarray analysis, intrusion detection, content based information retrieval. For example, the difficulty of text categorization (aims at assigning the new text documents to predefined classes) comes mainly from its high dimensionality. This is because of the massive on-line text (from many sources such as WWW, Emails, digital libraries, etc.) and each unique term (word or phrase) represents a feature in the original set of features. This means for a moderate-sized text collection, we will have hundreds or thousands of unique terms. Therefore, it is desirable to reduce the original feature space without affecting the prediction accuracy. Another example that shows the need for dimensionality reduction is gene expression microarray analysis (where the aim is to classify the new instances into predefined disease types). This is because we usually have thousands of genes and few samples in such applications. Recently, new applications of dimensionality reduction have emerged in many fields such as social networks analysis, recommendations systems, and product reviews summarizations [13]-[17].

There are three main dimensions of pre-processed datasets. These are: columns (features), the values of these features, and rows (instances). Therefore, the main three dimension reduction operations are decreasing the number of the columns (feature reduction), decreasing the number of the values of a feature (smoothing a feature), and deleting a row (instance reduction) as shown in Fig. 2. All of these operations delete unimportant data. Therefore, they preserve the characteristics of the original data. These operations will be explained in the following subsections.

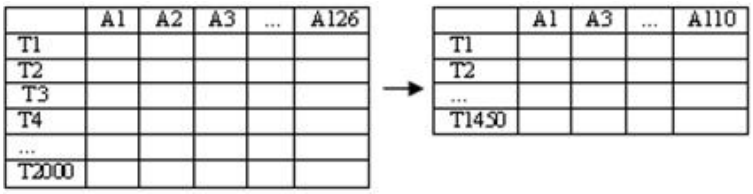

Fig. 2. Data reduction [2].

There are other dimensionality reduction operations but their results are not recognizable when comparing them with the original data. An example of these operations is to replace a set of features with only one new feature such as replacing the person-height feature and the person-weight feature with only one feature which is body-mass-index (according to the available background in this particular medical domain) [5]. This is known as feature extraction (also called feature composition, feature creation, or feature construction).

As it is known, no single data-reduction operation can suite all applications. One can decide on which suitable operation based on the available knowledge about the given application and the other constraints such as the required time for the final solution. For example, for a given application we can decide on extracting new features from the original set of features (such as exploratory analysis applications) but for another application (such as gene structure discovery) we can decide on selecting a subset of the original

set of features). Regardless of the used dimension reduction operation, the reduced data should not reduce the quality of the obtained results [5].

2.1. Smoothing a Feature

It reduces the number of distinct values for a given feature (feature's cardinality). An example of data smoothing techniques is binning methods that smooth a sorted data value by consulting its neighborhood (the values around it). The sorted values are distributed into bins or buckets. The importance of smoothing a feature is that some predictors (such as Naïve Bayes) work best when provided with a small number of distinct values per a feature [12].

Another example is the normalization (applied only to numerical data) which compresses/normalizes the scale of feature's values. It allows avoiding the existence of a feature overly affects an algorithm's processing because it contains large numbers. Normalization is very important for many methods such as neural networks which requires the input data to be normalized to ensure that a feature (for example, with a range from 0 to 1,000,000) doesn't eclipse another feature (for example, with a range from 0 to 100) as it relatively crushes the values into a uniform range (typically from 0 to 1 or from -1 to 1) [12].

2.2. Instance Reduction

The number of the instances in a dataset is determined by the number of the features and the number of the values that each feature can take. For example, if we have a dataset that consists of 2 binary features, then the number of instances will be no more than four [3].

Usually, the largest dimension in a dataset is the number of samples. Instance reduction is considered the most complex task in data reduction. However, if the number of samples in the prepared dataset can be managed by the used data mining methods, there is no reason for performing it [5], [10].

2.3. Feature Reduction

Dealing only with relevant features is more efficient and effective than with including irrelevant and redundant ones. The aim of feature reduction is to choose features that are relevant to the given data mining application in order to achieve the maximum performance with the minimum processing cost [5]. Feature reduction includes methods seek to form new features out of the original set of features (feature extraction) and methods seek merely a smaller subset of the original set of features (feature selection) as explained below [10].

2.3.1 Feature selection

Because of the negative effect of irrelevant features on most predictors, it is common to precede the learning step with the feature selection stage to find a subset of features (after detecting and removing irrelevant, weakly relevant, or redundant features) with predictive performance comparable to the original set of features [2], [4], [8] that is the reduced set should be having potential for good solutions.

Feature selection algorithms can be generally classified into two categories. These are feature-ranking algorithms and minimum subset algorithms:

- **Feature-ranking algorithms**: their output is a ranked list of features that are ordered according to a specific evaluation measure such as the accuracy of available data, information content, consistency, distances between samples, and statistical dependencies between features. These algorithms indicate the relevance of a feature compared to others. Since these algorithms rank the features from the most relevant features to the least ones, a decision is to be made in order to decide on which percentage of the top features to be included and which ones to be removed [12]. Top ranked features can be selected above a threshold criterion, and other irrelevant features are dropped. Then, if there are still a large number of features, it may be appropriate to build models on different subsets of the top

features to decide on which subset produces the best model [2], [12].

- **Minimum subset algorithms**: their output is the minimum feature subset whose features have the same ranking and considered relevant for the given data mining task (the other features are considered irrelevant). In both algorithms, it is essential to set a feature-evaluation scheme in which the features are evaluated and then ranked (in feature-ranking algorithms), or added to the selected feature subset (in minimum subset algorithms) [5]. The importance of these methods comes from being general methods (as they can be applied to many applications without the need for studying the data characteristics of the application at hand).

As *a search problem*, there are two types of search strategies that can be used for solving feature selection problems. These are:

- **Exhaustive search:** It analyzes all of the candidate solutions (different feature subsets) in any order. This is trivial if the search space (consists of all of the combinations of the feature subsets which equals 2^n-1 different non-empty feature subsets; where *n* is the number of features in the given dataset) is small and the search will get completed in a short time. This can be achieved easily in short time if the original set of features consists of a small number of features. Implementations examples of this search strategy are depth-first search and breadth-first search as shown in Fig. 3 and Fig. 4 respectively. Their essence is the way of performing the systematic search. The depth-first search goes down one branch (backtracks to another branch until checking all of the braces). The systematic search in breadth-first search is performed layer by layer (checking all of the attribute subsets that have one attribute then all of the attribute subsets that have 2 attributes and so on) [3].

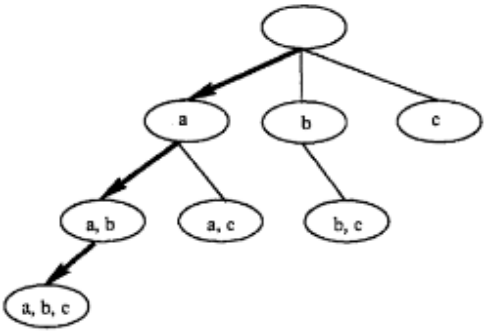

Fig. 3. Depth-first search [3].

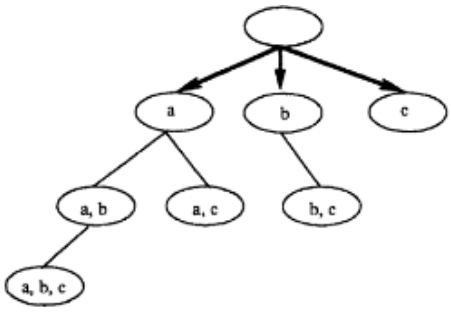

Fig. 4. Breadth-first search [3].

- **Heuristic search:** The search space is not usually small in practice i.e., for practical reasons, complete search is not feasible (except for very small *n*). In this case, examining all of the feature subsets will be replaced by heuristic search [5], [6]. Heuristic search does not examine all of the possible subsets of the original set of features and finds a near optimal subset (acceptable, reasonable, and timely solution)

[3], [5]. Being solved by heuristic methods is considered an advantage of feature selection problems. This is because recently researchers have developed many new metaheuristics (and more recently hybrid metaheuristics) that have been successfully applied to many optimization problems. Examples of metaheuristics that have been used successfully for the feature selection problem are ant colony optimization algorithms and genetic algorithms. Here, we can say that these algorithms for the feature selection besides the used predicator together constitute a hybrid method which is considered the second form of hybrid metaheuristics (metaheuristics with other algorithms). For more details, see [18].

Feature selection is necessary in many situations such as:

- It results in a more easily interpretable representation of the original data that makes the users focus on the more relevant features [4].
- Important decisions are taken based on the final selected feature subset in many applications such as a doctor may settle on a choice in view of the chosen subset whether a hazardous surgery is important for treatment or not [19].
- Having a substantial number of features can prompt considering the available instances not enough as in real-world problems where the number of features can be several hundreds. In this case, if the number of the given instances is a few hundreds, then dimensionality reduction is required in order for any reliable model to be of any practical use. Moreover, data overload (as a result of high dimensionality) can make data mining methods are not applicable [5]. This is obvious in drug datasets that are known as hard-modeled as a result of having large number of features and a small number of samples. Despite of this difficulty, getting good results with these datasets can lead to time and financial savings in many pharmaceutical fields [20].
- In many applications, the given dataset has many rows than columns. For instance, customers (rows in a dataset) may increase but the number of features for each customer may not increase much. Another example is sales that may increase but the number of features per sale may not increase as well. Thus, feature selection (deleting a column) has a more crucial effect than instance reduction [8].

All of these benefits encourage researchers to sacrifice the increased computation involved in adding the phase of feature selection to gain such benefits.

2.3.2 Feature extraction

It transforms the original p features into a new set of features (artificial features) p` such that $p$` is much smaller than p and p` is more fundamental to the given task i.e., they can better capture important information from the given data than the original set of features [9]. Therefore, this smaller set of new features can result in deeper understanding of the original features and improve the quality of the model by providing them with fewer features that have richer content [12]. These new features are called principal components, basis functions, latent features, factors, etc. depending on the used methods to derive them and also the specific objectives. Examples of feature extraction methods are neural networks, principal components analysis, wavelet transforms, and projection pursuit [5], [6], [9].

Feature extraction is an important factor in determining the quality of the used data mining method [5]. Moreover, it is essential where there are many features (hundreds or thousands) and each feature has on its own weak (may be ambiguous) predictability. But when these features are taken in combination, they produce meaningful patterns. This occurs in many domains such as text mining and genomics [12].

Unfortunately, human assistance is required to find the best set of transformations for a given data mining task and these transformations cannot be generalized to another application. Besides, it is highly much more application and domain dependent than is proper classification as (in most instances) it depends on the knowledge of the application at hand (and thus requires knowledge of the domain). For

example, a good feature extraction algorithm for sorting fish problem will certainly be of little use for other applications such as classifying blood cells' photomicrographs, or identifying fingerprints [5], [10], [11].

3. The Negative Effect of Irrelevant Features on Predictors

Predictors are designed in a way that makes them learn which features from the provided set of features are most appropriate to use for making their decisions. A good example of that is the decision tree method that should at each point choose the most promising features to split on and should not select irrelevant ones [4]. But, if the given set of features contains irrelevant features, these features can confuse the predictors. This emphasizes the fact that irrelevant features have negative effect on most predictors, in general, and more particularly on decision tree and rule learners as these features decrease the performance of state-of-the-art decision tree and rule learners as explained below [4], [9].

In the reminder of this section, we mention examples of this effect on some of the most known machine learning algorithms.

- Experiments with a decision tree showed that adding a random attribute influenced the prediction accuracy as it became worsen by 5% to 10% in the tested situations [4].
- Divide-and-conquer tree and separate-and-conquer rule methods both suffer from the same bad effect of irrelevant features as decision tree method [4].
- Naïve Bayes robustly disregard irrelevant features as it supposes (by its design) that all of the features are independent of each other. But, on the other side, it pays a substantial cost as its operation is harmed by including redundant features [4], [9].
- The accuracy of the nearest neighbor can be severely decreased by the existence of irrelevant or noisy features, or if the feature scales are not consistent with their importance causing it to perform poorly. Therefore, its performance can be enhanced by adding the phase of feature selection [4], [9]. This is because in nearest neighbor classifiers, all of the attributes have equal least and most potential effect on distance computations which handicaps these classifiers since they allow irrelevant, noisy, or redundant attributes to have as much influence on distance calculations as other attributes [11]. Moreover, in nearest neighbor classifiers, the sample complexity increases exponentially with the dimension. Therefore, this method strongly depends on feature reduction [21]. This illustrates the fact that high dimensional data may prompt poor generalization abilities of the learning algorithm [8].
- Radial basis functions (RBF) networks assign the same weight for every feature as all of the features are handled evenly in the distance calculation. Therefore, they cannot deal effectively with irrelevant features (in contrast to multilayer perceptrons). This is considered a disadvantage of radial basis functions (RBF) networks.
- The pervious disadvantage of radial basis functions is present with support vector machines (SVM). This is because support vector machines with Gaussian kernels (i.e., RBF kernels) is a particular type of RBF networks [4]. Besides, feature selection plays an important role in building support vector machines.
- Besides, many researchers have applied evolutionary algorithms for the solution of the feature selection problem and they all tested the effect of adding the phase of feature selection on the performance of the used predictor by performing two types of experiments:

1. the used predicator that uses the entire set of features (without the phase of feature selection), and
2. the used predicator that uses a subset of features selected by the used algorithm for performing the feature selection. An example of that is the work of Abd-Alsabour and Randall [22] where they used the support victor machine classifier with real-world and artificial datasets. Abd-Alsabour and Randall [22] also listed other results (available from literature) of different classifiers (nearest neighbor and

DistAI classifiers) with the use of different evolutionary algorithms (genetic algorithms, particle swarm optimization, and chaotic binary particle swarm optimization) for performing the feature selection phase.

4. Conclusion and Future Work

This paper studied the impact of irrelevant features on the performance of many predictors. Besides, it demonstrates dimensionality reduction from many perspectives such as its applications, its branches and the differences among them, its importance, and the algorithms used for tackling it.

As a research direction for future work in this area, listing recent results of different algorithms used for dimensionality reduction should be added. Moreover, comparisons among these algorithms should be included.

References

[1] Klosgen, W., & Zytkow, J. M. (2002). *Handbook of Data Mining and Knowledge Discovery.* Oxford University Press.

[2] Han, J., & Kamber, M. (2001). *Data Mining Concepts and Techniques* (1st ed.). Morgan Kaufmann Publishers.

[3] Liu, H., & Motoda, H. (1998). *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers.

[4] Witten, I., & Frank, E. (2005). *Data Mining-Practical Machine Learning Tools and Techniques.* Elsevier.

[5] Kantardzic, M. (2003). Data mining: Concepts, models, methods, and algorithms. *John Wiley & Sons.*

[6] Hand, D., Mannila, H., & Smyth, P. (2001). Principles of data mining. *Massachusetts Institute of Technology.*

[7] Cios, K. J., Pedrycz, W., & Wiswiniarski, R. (1998). *Data Mining Methods for Knowledge Discovery.* Kluwer Academic Publishers.

[8] Weiss, S., & Indurkhya, N. (1998). *Predictive Data Mining- A Practical Guide.* Morgan Kaufmann Publishers.

[9] Gorunescu, F. (2011). *Data Mining-Concepts, Models and Techniques.* Springer-Verlag Berlin Heidelberg.

[10] Duda, R., Hart, P., & David, S. (2001). *Pattern Classification* (2nd ed.). Wiley.

[11] Maimon, O., & Rokach, L. (2005). *Data Mining and Knowledge Discovery Handbook.* Springer.

[12] Hornick, M., Marcadé, E., Venkayala, S. (2007). *Java Data Mining: Strategy, Standard, and Practice.* Elsevier.

[13] Chen, J., Huang, H., Tian, S., & Qua, Y. (2009). Feature selection for text classification with naïve bayes. *Expert Systems with Applications, 36,* 5432-5435.

[14] Yu, L., Ye, J., & Liu, H. (2007). Dimensionality reduction for data mining — Techniques, applications and trends. *Proceedings of SIAM International Conference on Data Mining.*

[15] Li, Q., Jin, Z., Wang, C., & Zeng, D. (2016). Mining opinion summarizations using convolutional neural networks in Chinese microblogging systems. *Knowledge-Based Systems, 107,* 289-300.

[16] Zhang, L., Xia, Y., Mao, K., Ma, H., & Shan, Z. (2015). An effective video summarization framework toward handheld devices. *IEEE Transactions on Industrial Electronics, 62(2)*, 1309-1316.

[17] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Application of dimensionality reduction in recommender system — A case study. *Proceedings of the ACM WebKDD Workshop.*

[18] Abd-Alsabour, N. (2016). Hybrid metaheuristics for classification problems. In S. Ramakrishnan (Ed.), *Pattern Recognition*.

[19] Kim, Y. (2001). *Feature Selection in Supervised and Unsupervised Learning via Evolutionary Search.*

Unpublished Ph.D Thesis, The University of Iowa, USA.
[20] Abd-Alsabour, N. (2015). Binary ant colony optimization for subset problems. In S. Dehuri, A. K. Jagadev, & M. Panda (Eds.), *Multi-objective Swarm Intelligence — Theoretical Advances and Applications, Studies in Computational Intelligence* (pp. 105-121), Springer.
[21] Novakovic, J. (2010). The impact of feature selection on the accuracy of naïve bayes classifier. *Proceedings of the 18th Telecommunications forum TELFOR.*
[22] Abd-Alsabour, N., & Randall, M. (2010) Feature selection for classification using an ant colony system. *Proceedings of IEEE International Conference on e-Science and Grid Computing workshops* (pp. 86-91).

Nadia Abd-Alsabour is an assistant professor at Cairo University, Cairo, Egypt. Her research interests are swarm intelligence, optimization problems, data mining, and software engineering. Abd-Alsabour is a PC member, session chair, publicity chair, and a reviewer in many international conferences and journals.