Schedae Informaticae Vol. 26 (2017): 69–78
doi: 10.4467/20838476SI.17.006.8152

Short Review of Dimensionality Reduction Methods for Failure Detection

Agnieszka Pocha[1], Krzysztof Misztal[1], Paweł Morkisz[2,3]
[1]Faculty of Mathematics and Computer Science
ul. Łojasiewicza 6, 30-348 Kraków, Poland
e-mail: *agnieszka.pocha@doctoral.uj.edu.pl, krzysztof.misztal@uj.edu.pl*
[2]AGH University of Science and Technology
Faculty of Applied Mathematics
al. Mickiewicza 30, 30-059 Kraków, Poland
e-mail: *morkiszp@agh.edu.pl*
[3]Reliability Solutions
ul. Lublańska 34, 31-476 Kraków, Poland

Abstract. Size of a dataset is often a challenge in real-life applications. Especially, when working with time series data, when the next sample is produced every few milliseconds and can include measurements from hundreds of sensors, one has to take dimensionality of the data into consideration. In this work, we compare various dimensionality reduction methods for time series data and check their performance on a failure detection task. We work on sensory data coming from existing machines.

Keywords: dimensionality reduction, time series, failure detection

1. Introduction

Dimensionality reduction has become an integral part of data analysis nowadays as the amount of data that we are capable of collecting is much greater than what we can actually process. However, a great part of the collected data might be redundant or unnecessary, thus, dimensionality reduction methods are of major importance,

Received: 25 September 2017 / Accepted: 20 October 2017

especially for practical, industrial applications. According to report by Grand View Research, Inc. [1], Industrial Internet of Things is rapidly growing with CAGR at level about 27%. That directly implies humongous data sets collected by the industry.

One of the industrial applications is predictive maintenance, a machinery maintenance strategy used to predict the oncoming failure. Such knowledge gives not only huge financial benefits, as it can dramatically reduce operational costs but sometimes also prevents environmental catastrophes or multiple casualties.

The choice of an appropriate dimensionality reduction model for a specific failure detection task is not an easy problem. On the one hand, one must consider all the sensors available as often a failure of a particular machine has roots in some process problems of the connected machines. For large industrial applications such as power plants, oil rigs or chemical plants, that often means measurements from tens of thousands of sensors collected every minute. On the other hand, feeding mathematical models with such huge data portions without proper preprocessing usually leads to overfitting or generates false positives. Hence, tools that reduce the dimensionality of data, either by selecting relevant features or by transforming the data to a lower-dimensional space, play a major role in the initial data transformations.

Our goal in this paper is to investigate the value of dimensionality reduction methods for the task of failure prediction.

The paper is organized as follows: in section 2. we describe the current state of the art, in section 3. we give a brief summary of models used in this work, in section 4. we present what experiments were performed, in section 5. we present the results, and, finally, in section 6. we enclose our conclusions.

2. State of the art

In this section, we want to draw attention to two areas of research: in section 2.1. to dimensionality reduction models and in section 2.2. to sequential data. Moreover, in section 2.3. we give some examples of why using real-life datasets might be a challenge.

2.1. Dimensionality reduction

Dimensionality reduction methods are in focus of researchers in many areas of real-world applications for which datasets contain tens, hundreds, or millions of variables. A large number of algorithms have been proposed to deal with such datasets [2].

Dimensionality reduction methods can be divided into two groups: models that select most informative features, and models that project data to some lower dimensional space. Selection-based models are more computationally effective after being trained (selecting features is faster than applying any algebraic transformation) and

do not cause loss of interpretability of data. However, transformation-based models might yield better results in some applications.

Selection-based algorithms might be further divided into following classes:

- filter approach models consider statistical characteristics of the input data, ex. select the features for which the correlation between the feature and the target variable exceeds a correlation threshold. These methods are computationally inexpensive;
- wrapper models train a chosen estimator on the original data and select relevant features based on the performance of the learning algorithm. These methods are computationally expensive;
- embedded approach adjusts existing methods and as a result feature selection is built into the target model, examples of this approach are linear regression with LASSO or regularized random forests.

Transform-based models project existing features into new, lower-dimensional feature space, which, we hope, better explains our observations. These methods can be divided into two classes:

- linear: PCA (Principal Component Analysis) [3]; Singular Value Decomposition (SVD) [4]; ICA (Independent component analysis) [5]; LDA (Latent Dirichlet Allocation) [6]; Latent semantic indexing [7]; Piecewise Linear Representation [8]; Genetic Programming [9];
- non-linear: NPCA (nonlinear PCS) [10] or KPCA (kernel principal component analysis) [11]; NLDA or KLDA [12]; MDS (Multidimensional scaling) [13]; Principal curves; Neural networks [14]; Genetic Programming [15].

2.2. Sequential data

A common practice when working with sequential data is to represent it as a combination of several factors. Two types can be distinguished:

- systematic – components of the time series that have consistency or recurrence and can be described and modeled;
- unsystematic – components of the time series that cannot be directly modeled.

Thus the data series is thought to be an aggregation or combination of following four components:

- Trend component (long-term trend) – A trend exists when there is a long-term increase or decrease in the data. It does not have to be linear. Sometimes we will refer to a trend changing direction when it might go from an increasing trend to a decreasing trend.

- Seasonal component (seasonal variation) – A seasonal pattern exists when a series is influenced by seasonal factors (e.g. a quarter of a year, or a day of the week). Seasonality is always of a fixed and known period.

- Cyclical component (repeated but non-periodic fluctuations) – A cyclic pattern exists when data exhibit rises and falls that are not of fixed period.

- Irregular component (the residuals) – This component contains the volatile part of the series (noisy or random) and tends to be the least predictable of all the elements.

Currently, the most successful models for sequential data are recurrent neural networks, especially their modifications Long Short-Term Memory (LSTM) networks [16] and Gated Recurrent Unit (GRU) networks [17]. However, in this work, we decided to concentrate on such models that are feasible for embedded systems, what requires that their training and inference time is short. Therefore, we had to abandon recurrent neural networks approach.

2.3. Challenging character of real-life data

Real-world datasets are often challenging due to many factors such as unknown nature of noise, little or no knowledge of how are features correlated with the target variable and if they are indeed correlated. In many cases, the data on which the analysis is performed has been previously anonymized and as a result, any guess of what the features represent is impossible. Anonymizing techniques such as hashing or normalizing the data make it also more difficult to draw conclusions about the original nature of the data, ex. which features are more noisy and which are more reliable.

For failure prediction task two maybe most important challenges are high inconsistency of features, which might be caused by varying operating conditions and asses-to-asset variations, and low sensitivity of features to faults or degradation.

To answer the above problem one might try using such techniques as normalization of the data, finding generalizable features, clustering samples depending on operating conditions, or using local models.

3. Review of used methods

In this section, we briefly describe dimensionality reduction models, which we use in the experimental part of this work. For each method, we define acronyms used in the further part of this article.

3.1. Selection-based models

The problem of feature selection is defined as follows: given a set of available features, select such a subset that performs best under some classification system or other appropriate relevance measure ex. time or memory consumption.

- **CHI2** – the χ^2 statistics between the target and each feature is calculated and then the desired number of features with the best χ^2 score is selected. Features that are not correlated with the target have χ^2 statistics close to zero and are not selected.
- **CLASSIF** – ANOVA F-value [18] between the target vector and each feature is calculated. The higher F-value is, the more proportion of variance the feature or groups of features can explain in the target data. Features with high F-value should be selected.
- **REGR** – A method using F-value between label/feature for classification tasks. The correlation between each regressor and the target is computed and the result is transformed into F score then to a p-value – the best are described by best features.
- **MUTUAL** – Mutual information quantifies the dependence between two random variables in terms of information communicated about the value of one variable given knowledge of the other. If mutual information is close to zero both variables are independent. Variables are compared using this criterion.

3.2. Transformation-based models

Transformation-based models project data onto a lower-dimensional subspace which best fits the data. There are several criteria that define a "good fit" and therefore there exist several different algorithms for finding a subspace that meets the chosen criterion.

- **PCA** – based on the covariance matrix of the features the most significant principal components are chosen to form the directions of the new coordinate system;
- **TSVD** – a variant of singular value decomposition (SVD) that computes only the k largest singular values. This method is similar to PCA;
- **FICA** – uses FastICA algorithm to find independent components which form a base of the new coordinate system
- **FAC_ANA** – Factor Analysis performs a maximum likelihood estimate of the so-called loading matrix, the transformation of the latent variables to the observed ones, using expectation-maximization (EM).

4. Experiments

In this section, we describe the challenges that had to be faced while working with the data and the experiments we conducted. In section 4.1. we describe general problems that we encountered while working with the given data and in section 4.2. we describe the experiments.

4.1. Challenging character of the data

During our experiments, we had to face some challenges issued by the data characteristics. The most obvious one was an extreme class-imbalance - each dataset contained no more than 1% of samples labeled as failures. Moreover, there was no certainty that all failures will be similar, i.e. each failure might be very different to all others and that makes a classification task, on which we evaluated the dimensionality reduction models, very difficult. Furthermore, failures were not uniformly distributed within the datasets which made it a necessity to define splits manually so that each split would have similar class-balance and considerable size. Finally, the size of the datasets and the motivation to use the methods in embedded systems restricted the choice of both dimensionality reduction and classification models to computationally efficient representatives.

4.2. Setup

For the experiments, we used two datasets obtained from the company Reliability Solutions. The dataset RS1 had originally 49 dimensions, over 2.8M samples and contained only 16 samples labeled as failures. It was reduced to 18 dimensions. The dataset RS2 had originally 296 dimensions, over 2.2M samples and over 18K samples labeled as failures. It was reduced to 30 dimensions.

The quality of the concerned dimensionality reduction methods was measured on a classification task. We used logistic regression and random forests as representatives of linear and nonlinear methods. Grid search was performed to choose the best hyperparameters for each model.

During our experiments, we learned that splitting the datasets must be performed very carefully, which was already mentioned in section 4.1. Double cross-validation was performed using time split in such a way so that each split would have similar class balance and be of considerable size. Failures that belonged to one group were assigned to the same split.

The trend was removed from the data before running any necessary processing. Each of the considered dimensionality reduction models requires different preprocessing and in some cases, the data was processed after dimensionality reduction as well.

The details are presented below.

- CHI2 - before reducing the dimensionality each feature was scaled to $[0, 1]$
- CLASSIF - data was not altered in any way
- FAC_ANA - each feature was scaled and centered before and after reducing the dimensionality
- FICA - data was whitened before reducing dimensionality and each feature was scaled and centered after reducing the dimensionality
- MUTUAL - data was not altered in any way
- PCA - each feature was scaled and centered before and after reducing the dimensionality
- REGR - before reducing dimensionality each feature was scaled to $[-1, 1]$
- TSVD - each feature was scaled and centered before and after reducing the dimensionality

We report the mean value of Matthews Correlation Coefficient on four different testing sets.

5. Results

	CHI2	CLASSIF	REGR	MUTUAL	FICA	PCA	FAC_ANA	TSVD
				Logistic Regression				
rs1	0.35	0.07	0.25	0.00	0.08	0.10	0.36	0.13
rs2	0.12	0.05	0.32	0.11	-0.03	0.11	0.03	0.11
				Random Forest				
rs1	0.41	0.39	0.39	0.41	0.00	0.82	0.81	0.82
rs2	0.56	0.40	0.40	0.36	0.02	0.09	0.20	0.08

Table 1. MCC score for each dimensionality reduction method obtained using logistic regression and random forests on datasets rs1 and rs2.

Table 1 presents average MCC score obtained by logistic regression and random forests on datasets transformed by different dimensionality reduction methods. These

results suggest that the correlation between new features and the presence of a failure is a nonlinear one. There is no clear evidence for the superiority of any group of methods - depending on the dataset and classification model used either selecting or transforming methods achieved better performance. The obtained results are satisfactory considering the challenging character of the data.

In tables 2 and 3 we present an average time needed to reduce the dimensionality and to train and test the best performing models respectively. Dimensionality reduction and training of each model were performed on a single CPU unit. All dimensionality reduction methods but Mutual finished their task by 10 minutes. Training and prediction times spanned from few minutes up to an hour depending on the dimensionality reduction model and classification method used. These results suggest that some of the considered models are feasible for embedded systems.

	CHI2	CLASSIF	REGR	MUTUAL	FICA	PCA	FAC_ANA	TSVD
rs1	18s	15s	53s	49m 15s	1m 40s	46s	10m 3s	35s
rs2	56s	54s	1m 30s	3h 37m	4m 37s	3m 39s	9m 38s	1m 22s

Table 2. Average times needed for dimensionality reduction by different methods.

	CHI2	CLASSIF	REGR	MUTUAL	FICA	PCA	FAC_ANA	TSVD
				Logistic Regression				
rs1	2m 36s	2m 25s	1m 46s	2m 35s	2m 32s	2m 31s	2m 32s	2m 29s
rs2	1m 6s	2m 38s	2m 6s	2m 44s	1m 25s	1m 20s	3m 3s	1m 30s
				Random Forest				
rs1	5m 38s	8m 22s	7m 13s	6m 22s	9m 29s	8m 42s	13m 21s	8m 42s
rs2	15m 36s	20m 34s	20m 18s	16m 44s	48m 28s	19m 26s	27m 38s	19m 13s

Table 3. Average times for training and testing best performing models.

6. Conclusions

Our experiments show that when tackling time series data of enormous size one should consider dimensionality reduction methods based both on selection and on transformation. There is no clear indication whether to use selection- or transformation-

based models. At least some of the analyzed dimensionality reduction models can be effectively used in real-world applications using big datasets. Logistic regression and random forests have feasible training time even for big datasets. Random forests can achieve satisfactory results even on nonlinear tasks. The problem of classifying failures seems to be such a nonlinear problem.

7. References

[1] Grand View Research, I., *Industrial internet of things (iiot) market analysis by component (solution, services, platform), by end-use (manufacturing, energy & power, oil & gas, healthcare, logistics & transport, agriculture), and segment forecasts, 2018–2025.* Market Research Report, 2017.

[2] Van Der Maaten L., Postma E., Van den Herik J., *Dimensionality reduction: a comparative.* J Mach Learn Res, 2009, 10, pp. 66–71.

[3] Jolliffe I., *Principal component analysis.* Wiley Online Library, 2002.

[4] Howland P., Park H., *Generalizing discriminant analysis using the generalized singular value decomposition.* IEEE transactions on pattern analysis and machine intelligence, 2004, **26**(8), pp. 995–1006.

[5] Hyvärinen A., Karhunen J., Oja E., *Independent component analysis.* vol. 46. John Wiley & Sons, 2004.

[6] Blei D.M., Ng A.Y., Jordan M.I., *Latent dirichlet allocation.* Journal of machine Learning research, 2003, 3 (Jan), pp. 993–1022.

[7] Landauer T.K., *Latent semantic analysis.* Wiley Online Library, 2006.

[8] Chua L., Deng A.C., *Canonical piecewise-linear representation.* IEEE Transactions on Circuits and Systems, 1988, 35 (1), pp. 101–111.

[9] Yang J., Honavar V., Feature subset selection using a genetic algorithm. In: *Feature extraction, construction and selection.* Springer 1998 pp. 117–136.

[10] Monahan A.H., *Nonlinear principal component analysis by neural networks: theory and application to the lorenz system.* Journal of Climate, 2000, 13 (4), pp. 821–835.

[11] Schölkopf B., Smola A., Müller, K.R., *Kernel principal component analysis.* In: *International Conference on Artificial Neural Networks*, Springer, 1997, pp. 583–588.

[12] Mika S., Ratsch G., Weston J., Scholkopf B., Mullers K.R., *Fisher discriminant analysis with kernels.* In: *Neural networks for signal processing IX, 1999. Proceedings of the 1999 IEEE signal processing society workshop.*, IEEE, 1999, pp. 41–48.

[13] Venna, J., Kaski, S., *Local multidimensional scaling.* Neural Networks, 2006, 19 (6), pp. 889–899.

[14] Verikas, A., Bacauskiene, M., *Feature selection with neural networks.* Pattern Recognition Letters, 2002, 23 (11), pp. 1323–1335.

[15] Wu, Y.L., Tang, C.Y., Hor, M.K., Wu, P.F., *Feature selection using genetic algorithm and cluster validation.* Expert Systems with Applications, 2011, 38 (3), pp. 2727–2732.

[16] Hochreiter, S., Schmidhuber, J., *Long short-term memory.* Neural computation, 1997, 9 (8), pp. 1735–1780.

[17] Cho K., Van Merriënboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y., *Learning phrase representations using rnn encoder-decoder for statistical machine translation.* arXiv preprint arXiv:1406.1078, 2014.

[18] Cochran W.G., Cox G.M., *Experimental designs.*, 1950.