Reports of the Department of Mathematical Information Technology
Series B. Scientific Computing
No. B 6/2013

Gear Classification and Fault Detection Using a Diffusion Map Framework

Tuomo Sipola Tapani Ristaniemi
Amir Averbuch

University of Jyväskylä
Department of Mathematical Information Technology
P.O. Box 35 (Agora)
FI–40014 University of Jyväskylä
FINLAND
fax +358 14 260 2771
http://www.mit.jyu.fi/

ISBN 978-951-39-5464-2
ISSN 1456-436X

Gear Classification and Fault Detection Using a Diffusion Map Framework[*]

Tuomo Sipola[†] Tapani Ristaniemi[‡] Amir Averbuch[§]

Abstract

A system health monitoring scheme using diffusion map is proposed. Diffusion map reduces the dimensionality of measurement data. This facilitates the comparison of newly arriving measurements to the known training data. The method is trained and tested with real gear monitoring data. The results show that data recordings can be classified as working or broken using dimensionality reduction.

1 Introduction

Modern industry monitoring systems produce high-dimensional data that are difficult to analyze as a whole without dimensionality reduction. The goal of the study is to estimate whether the proposed dimensionality reduction scheme effectively distinguishes working gears from broken ones. System health management has multiple sensors that measure vibration, temperature and oil properties. The early detection of anomalous gear behavior using this sensor data reduces the risk of severe damage. Sensor data are then used to monitor the health of the system, to detect anomalies and to predict problems [3, pp. 15–16].

Anomaly detection methods try to find deviant or atypical measurements from a large datamass [3]. In this study known anomalies are in the training so that they can be contrasted with the normal behavior. An ideal indicator would tell with certainty that a machine works or is going to fail. However, in reality the non-working state is ambiguous and it can be difficult to classify.

Spectral dimensionality reduction methods include principal component analysis (PCA), kernel PCA, multi-dimensional scaling (MDS), Laplacian eigenmaps,

[*] This research was partially supported by the Foundation of Nokia Corporation and the Finnish Foundation for Technology Promotion.

[†] Department of Mathematical Information Technology, University of Jyväskylä, PO Box 35 (Agora), FI-40014 University of Jyväskylä, Finland, `tuomo.sipola@jyu.fi`

[‡] Department of Mathematical Information Technology, University of Jyväskylä, PO Box 35 (Agora), FI-40014 University of Jyväskylä, Finland, `tapani.ristaniemi@jyu.fi`

[§] School of Computer Science, Tel Aviv University, Israel, `amir@math.tau.ac.il`

isomap and locally linear embedding (LLE). These methods facilitate the analysis of high-dimensional data by mapping the high-dimensional coordinates to a lower dimension. The spectral approach also leads to the concept of spectral clustering [2, 19]. Spectral methods have been used to analyze system operational states [15], motor fault detection [14] and anomaly detection for spacecraft [7].

This study uses diffusion map, which is another spectral dimensionality reduction method. Its mathematical foundation is random walk on Markov transition matrix of the graph of the data [4]. Diffusion map can be classified as a nonlinear distance-preserving dimensionality reduction method that preserves global properties [18]. Furthermore, the Nyström method is used to extend new points, although newer methods such as geometric harmonics exist [6, 5]. A similar study using diffusion map has been made concerning machine condition monitoring [8]. This study presents a way to detect faults in gears by devicing an index to describe how close to the faulty state a gear is. Besides gear fault detection, this method can also be used with other collections of high-dimensional time series data.

2 Method

This method trains a diffusion map that describes the good and bad state of the gears. It then extends newly arriving test measurements to the model and classifies the gear as good or bad. Most of the preprocessing is domain specific, but the dimensionality reduction and classification, that are more universally applicable, are presented here. Figure 1 introduces the overall data processing architecture. The equations are in matrix form. The details behind them are discussed elsewhere [12, 6, 1].

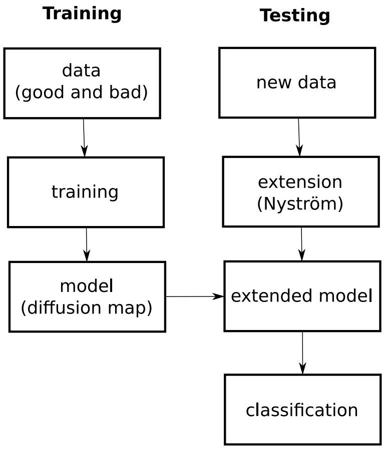

Figure 1: Data processing block diagram.

2.1 Training dimensionality reduction

The underlying assumption in manifold learning methods is that the data is situated on a lower-dimension manifold in the high-dimension measurement data [3, p. 37]. We try to create a function that maps the behavior of high-dimensional points to lower dimensions. Then new measurement points are mapped from high dimensions to this low-dimensional presentation.

Let $x_i \in \mathbb{R}^n, i = 1 \dots N$ be a measurement in n-dimensional space. The kernel matrix W includes the pairwise distances of these points. The used kernel is the Gaussian kernel using Euclidian distance measure. This is the most computationally intensive step because each point is compared to other points:

$$W_{ij} = \exp\left(-\frac{||x_i - x_j||^2}{\epsilon}\right). \tag{1}$$

Determining ϵ is a problem in itself. The chosen estimation is the median of the distances between the points, $\epsilon = \text{median}\{||x_i - x_j||\}_{x_i, x_j \in \mathbb{R}^n}$ [16]. Depending on the problem, changing this parameter might give more meaningful results.

Matrix $D_{ii} = \sum_{j=1}^{N} W_{ij}$ has the degree of each point on its diagonal. The degree of a point is the sum of weights that connect to other points. This is equal to the sum of kernel matrix rows.

The rows are normalized by these sums. The result can also be understood as transition probabilities between points. These probabilities are collected in matrix P,

$$P = D^{-1}W. \tag{2}$$

However, future calculations on P become easier if a similarity transformation symmetricizes the matrix:

$$\tilde{P} = D^{\frac{1}{2}} P D^{-\frac{1}{2}}. \tag{3}$$

These last two steps can be combined. Substituting P with $D^{-1}W$ yields:

$$\tilde{P} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}. \tag{4}$$

Such normal matrix is decomposed as:

$$\tilde{P} = U \Lambda U^*. \tag{5}$$

This decomposition is done using singular value decomposition (SVD). The columns of matrix U contain eigenvectors u_k of matrix $\tilde{P}$. Likewise, the diagonal of Λ contains its corresponding eigenvalues. However, the real interest is in the eigenvectors of the transition matrix P. The eigenvalues of P are the same, but the eigenvectors are obtained from V:

$$V = D^{-\frac{1}{2}} U. \tag{6}$$

Recall that the eigenvalues λ are in the diagonal of Λ. The eigenvector v are columns of V. An original data point x_i has a corresponding value on the ith row of the eigenvector. For example, $v_2(x_{236})$ would signify the second eigenvector and its 236th row, corresponding to the 236th sample x_{236} of the original dataset.

The diffusion map itself is a function in the form $\Psi : \mathbb{R}^n \to \mathbb{R}^d$, when $d \ll n$. We multiply the eigenvectors and eigenvalues to get the diffusion coordinates of the training points:

$$\Psi = V\Lambda. \tag{7}$$

The first eigenvector is constant, so only the following eigenvectors and eigenvalues are used. This way we get the following function that maps the original data points to a lower-dimensional space:

$$\Psi_d : x_i \to \begin{pmatrix} \lambda_2 v_2(x_i) \\ \lambda_3 v_3(x_i) \\ \lambda_4 v_4(x_i) \\ \vdots \\ \lambda_{d+1} v_{d+1}(x_i) \end{pmatrix}. \tag{8}$$

It has been shown that the diffusion distance in the original space equals to the Euclidean distance in the diffusion space [4]. Thus, the distance measurements in the diffusion space are actually meaningful and can be used in further analysis in this lower-dimensional space.

Later analysis uses only the first few diffusion coordinates. Fast decay of eigenvalues leaves most of the diffusion coordinates rather small compared to the first few. The overall reconstruction of P does not differ much from a reconstruction that uses only the first coordinates. These coordinates capture most of the differences between the data points [4, 11].

2.2 Extension of new measurements

New measurements that are not part of training are extended to the model with Nyström method [6, 1]. The features selected during training are the only ones needed. These new measurements are normalized using the same normalization as during training.

Let a new data point be $y_j \in \mathbb{R}^n$. Then the distance between the new points and each training point are collected in a matrix $\bar{W}$. This function uses the same ϵ as the one in training phase:

$$\bar{W}_{ij} = \exp\left(-\frac{||x_i - y_j||^2}{\epsilon}\right). \tag{9}$$

Diagonal matrix $\bar{D}_{ii} = \sum_{i=1}^{N} \bar{W}_{ij}$ contains the column sums of $\bar{W}$. Now we can create the transition probability matrix B:

$$B = \bar{W} * \bar{D}^{-1}. \tag{10}$$

The following matrix multiplication produces new eigenvectors for the new point. The eigenvectors V and eigenvalues Λ are the same as in training:

$$\bar{V} = B^T V \Lambda^{-1}. \tag{11}$$

These new eigenvectors now extend the new point to the diffusion coordinates:

$$\bar{\Psi} = \bar{V} \Lambda. \tag{12}$$

The last two steps can be combined:

$$\bar{\Psi} = B^T V. \tag{13}$$

Matrix $\bar{\Psi}$ now contains the extended eigenvectors in its columns for the new points y_j.

2.3 Classification of new measurements

Low-dimensional presentation of the data facilitates clustering. The clustering approach here is spectral clustering and it reveals the normal and anomalous areas [19, 9]. Any other clustering, for example k-means, can be used if they provide better results [13, 10, 17]. The used algorithm simply tests whether the sample is to the left or to the right of 0 on the dimension corresponding to the 2nd eigenvector. This provides a classifier that discriminates two states: working or broken.

2.4 Warning levels

For more warning levels, different thresholds can be applied. There are three warning levels: note, warning and damage. These describe the severity of the problem in the gear.

Note means that there is an unusual measurement in the data, but the gear is still in operational state. The sample is not inside the good cluster but is still closer to it than to the bad.

$$\theta_{note} = \min\{\Psi_{1,good}\} \tag{14}$$

Warning level is at $\theta_{warning} = 0$. It describes the border between good and bad clusters. The sample is closer to the bad cluster. This can be seen as a predictive sign that the gear has problems. If the bad cluster goes beyond 0, the middle point between the two clusters can be used.

Damage level is at $\theta_{damage} = \max\{\Psi_{1,bad}\}$. This means that the sample is within the bad cluster.

3 Results

This study uses a dataset consisting of gear monitoring recordings of multiple features. It consists of recordings of 18 good and 20 bad machines labeled by domain specialists. The gears come from different locations where the operational environment varies. However, each gear is of the same type and includes same features. Two of the gears are discarded because they contain empty data due to instrument failures. The dataset is divided to training and testing sets. The training set includes five good and five bad gears. The testing set includes the rest of the gears.

3.1 Preprosessing

The data are sampled at an approximate frequency of one sample per 30 minutes. The recordings last for months. Because there were times when no data were available, linear interpolation is used. This data formed the samples × features matrix.

Instrument failures give unrealistic or missing measurements. Because it is difficult to compare such measurements to ones that do not have unrealistic values, measurements containing missing values are discarded. However, this process might lose some usable information.

3.1.1 RPM filtering

Samples whose rotations per minute (RPM) value is too small are filtered out, because only higher values represent the actual working state of a gear. Lower values are associated with idle state, and those measurements are not interesting when monitoring actual working gears. The RPM values are clustered into two clusters using k-means clustering. The threshold value,

$$threshold_{RPM} = \max\{\min\{RPM_{cluster\ 1}\}, \min\{RPM_{cluster\ 2}\}\}, \tag{15}$$

is calculated and all the samples whose RPM value is below this threshold are removed.

3.1.2 Data scaling

All the data are normalized with logarithm. Other normalizations, like dividing by maximum or dividing by norm, do not give as good separation for this dataset.

3.1.3 Feature selection

There are 136 features. The initial feature selection reduced their number to 20. Some features separate more clearly the two groups from each other. A preliminary feature selection in the original feature space gives these features. One feature is left out at a time. The average Mahalanobis distance between the good and bad machines shows how much that feature describes the difference. The features with smallest averaged Mahalanobis distances are most useful. Small distance reveals that leaving the feature out affects negatively the separation of good and bad. Thus, using the feature separates the groups well in the feature space.

4 Classification results

Five good and five bad gears were used in training. The data has 136 features, 20 of which are used after preliminary feature selection. All the gears, including training gears, were then tested as new incoming data. Table 1 shows that each of the broken test gears had alerts. Table 2 shows that no working gear had warnings, although some of them had notes.

gear	alerts
OO03	2.5703%
*OO06	14.4068%
OO08	42.6573%
OO09	6.6667%
AH01	16.835%
AH02	16.7431%
AH06	2.8777%
*AH11	14.916%
AH18	7.0941%
FE09	5.3495%
FE10	4.4068%
*FE12	16.0083%
CA03	22.7599%
CA04	7.7089%
QU23	6.0469%
*QU32	21.6535%
ET104	0.32841%
*ET403	3.3597%
PH05	9.3694%

Table 1: Broken gear units (alert threshold 0). Asterisk marks training gears.

gear	alerts
*AH10	0%
AH16	0%
AH18	0%
FE01	0%
*FE02	0%
FE03	0%
CA01	0%
LS04	0%
*LS05	0%
MB08	0%
QU32	0%
*PH01	0%
PH03	0%
PH05	0%
PH08	0%
*PH09	0%
PH13	0%

Table 2: Working gear units (alert threshold 0). Asterisk marks training gears.

The following figures illustrate the behavior of broken gears. Normal state does not produce figures of interest because there are no alerts. Figure 2 shows how the newly incoming data is situated in low-dimensional space. Figure 3 shows the alert index, while Figure 4 indicates the accumulating number of alerts. The alerts themselves are in Figure 5. Figures 6, 7, 8, 9 show the same measurements for another gear. It breaks down more slowly but the high number of notes can be seen.

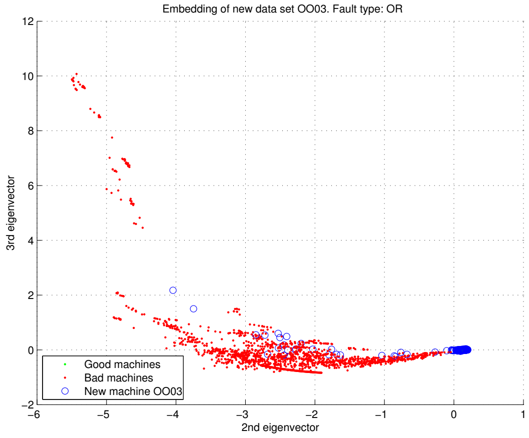

Figure 2: Samples of a broken gear in low-dimensional space.

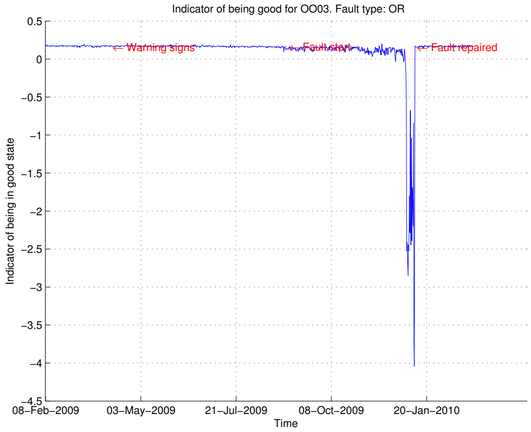

Figure 3: Alert level index of a broken gear. Above 0 is considered normal working state.

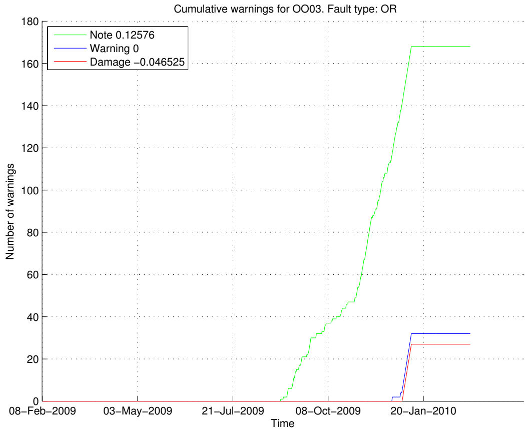

Figure 4: Number of alerts.

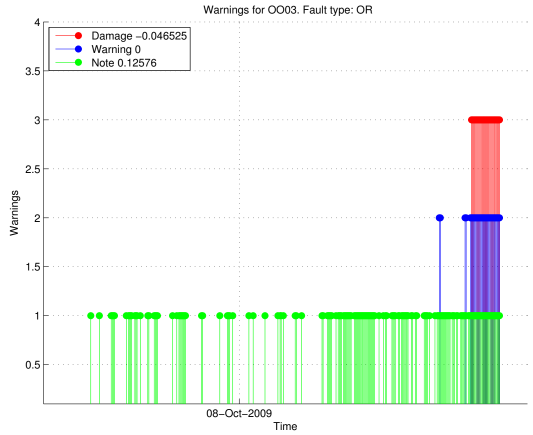

Figure 5: Alerts given by the method.

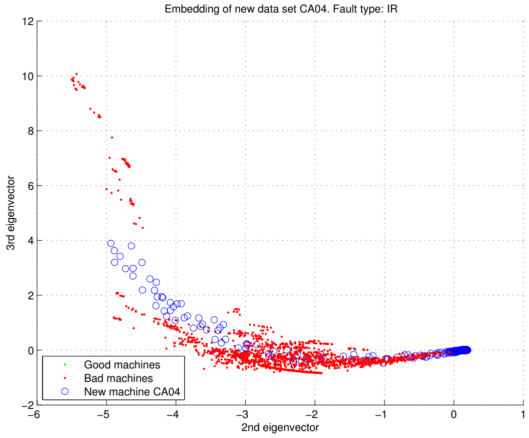

Figure 6: Samples of a broken gear in low-dimensional space.

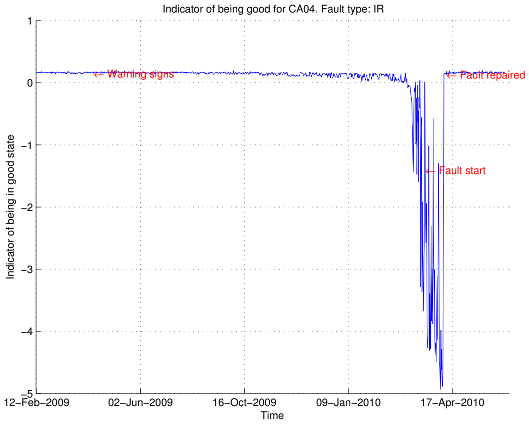

Figure 7: Alert level index of a broken gear. Above 0 is considered normal working state.

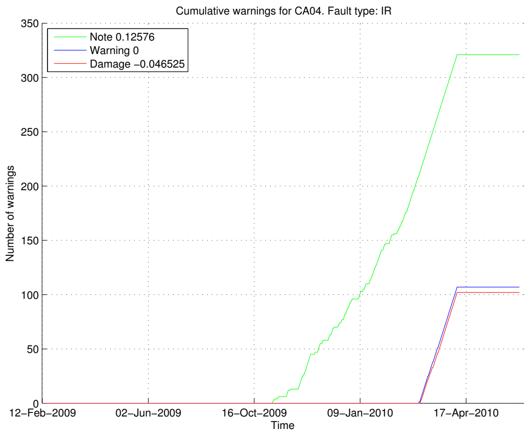

Figure 8: Number of alerts.

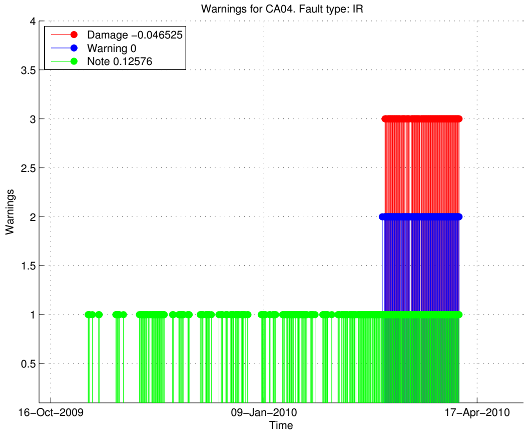

Figure 9: Alerts given by the method.

5 Discussion

The goal of this study is to estimate the usefulness of dimensionality reduction methods in gear fault detection. This goal is met since almost all the gears are classified correctly according to their labels. This proves that the training is successful and separates the good gears from the bad. More importantly, measurements from totally different gears can be extended into the model.

The misclassification of good machine FE01 as bad is probably because of the data interpolation. Further domain analysis revealed that there actually had been a small problem with the gear, and thus raises the question whether it is labeled correctly. The misclassification of bad machine ET104 as good can be explained. Firstly, there are no training gears from this location. ET104 is too close to the good gears in diffusion space. Secondly, domain analysis reveals that this gear has only a small problem. Better training data and more detailed labeling could prevent this kind of misclassification. Vastly different operating environment and behavior of gears in ET1 might also cause this misclassification.

The problems of spectral methods in general need some addressing. The proposed method works because, after slight filtering, the good and bad gears are separable in the lower dimensions. However, the high computational cost could be a problem in a more real-time system. The classification of a gear time series itself is an ambiguous concept. However, this study shows that gears in normal condition and gears that are going to break down behave differently and can be separated from each other.

References

[1] Serge Belongie, Charless Fowlkes, Fan Chung, and Jitendra Malik. Spectral partitioning with indefinite kernels using the Nyström extension. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Computer Vision – ECCV 2002*, volume 2352 of *Lecture Notes in Computer Science*, pages 51–57. Springer Berlin / Heidelberg, 2002.

[2] Yoshua Bengio, Olivier Delalleau, Nicolas Le Roux, Jean-François Paiement, Pascal Vincent, and Marie Ouimet. *Feature Extraction*, chapter Spectral Dimensionality Reduction, pages 519–550. Studies in Fuzziness and Soft Computing. Springer Berlin, Heidelberg, 2006.

[3] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41:15:1–15:58, July 2009.

[4] Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

[5] Ronald R. Coifman and Stphane Lafon. Geometric harmonics: A novel tool for multiscale out-of-sample extension of empirical functions. *Applied and Computational Harmonic Analysis*, 21(1):31 – 52, 2006. Diffusion Maps and Wavelets.

[6] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nystrom method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):214 –225, 2004.

[7] Ryohei Fujimaki, Takehisa Yairi, and Kazuo Machida. An approach to spacecraft anomaly detection problem using kernel feature space. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 401–410, New York, NY, USA, 2005. ACM.

[8] Yixiang Huang, Xuan F Zha, Jay Lee, and Chengliang Liu. Discriminant diffusion maps analysis: A robust manifold learner for dimensionality reduction and its applications in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, 34(1):277–297, 2013.

[9] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad and spectral. *J. ACM*, 51:497–515, May 2004.

[10] Marina Meila and Jianbo Shi. Learning segmentation by random walks. In *NIPS*, pages 873–879, 2000.

[11] Boaz Nadler, Stephane Lafon, Ronald Coifman, and Ioannis Kevrekidis. Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 955–962. MIT Press, Cambridge, MA, 2006.

[12] Boaz Nadler, Stephane Lafon, Ronald Coifman, and Ioannis G. Kevrekidis. Diffusion maps – a probabilistic interpretation for spectral embedding and clustering algorithms. In Timothy J. Barth, Michael Griebel, David E. Keyes, Risto M. Nieminen, Dirk Roose, Tamar Schlick, Alexander N. Gorban, Balázs Kégl, Donald C. Wunsch, and Andrei Y. Zinovyev, editors, *Principal Manifolds for Data Visualization and Dimension Reduction*, volume 58 of *Lecture Notes in Computational Science and Engineering*, pages 238–260. Springer Berlin Heidelberg, 2008.

[13] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2001.

[14] Lucas Parra, Gustavo Deco, and Stefan Miesbach. Statistical independence and novelty detection with information preserving nonlinear maps. *Neural Computation*, 8(2):260–269, 1996.

[15] Markus Pylvänen, Sami Äyrämö, and Tommi Kärkkäinen. Visualizing time series state changes with prototype based clustering. In Mikko Kolehmainen,

Pekka Toivanen, and Bartlomiej Beliczynski, editors, *Adaptive and Natural Computing Algorithms*, volume 5495 of *Lecture Notes in Computer Science*, pages 619–628. Springer Berlin / Heidelberg, 2009.

[16] A. Schclar, A. Averbuch, N. Rabin, V. Zheludev, and K. Hochman. A diffusion framework for detection of moving vehicles. *Digital Signal Processing*, 20(1):111 – 122, 2010.

[17] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888 –905, 2000.

[18] L. J. P. van der Maaten, E. O. Postma, and H. J. van Den Herik. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, 10:1–41, 2009.

[19] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007.