Hierarchial sensor fusion with applications for detection of anomaly trends in dynamically evolving systems

N. Rabin, A. Averbuch

Yale University, Applied Math Dept. neta.rabin@yale.edu
Tel Aviv University, Computer Science Dept. amir@math.tau.ac.il

Abstract

In this paper, we describe a new approach for sensor fusion that enables to learn and track the behavior of dynamical systems. A high-dimensional dataset, which describes the measured/observed sensors of a dynamical system, is embedded into a lower-dimension space by the application of the Diffusion Maps to it. Then, the system behavior and sought after anomalies are studied and detected in this lower-dimension embedded space. To achieve it, the Diffusion Maps methodology was extended to provide hierarchal (multi-scale) processing. The frequency appearance of each point in the embedded space quantitatively measures the systems state at each given time point. In addition, the data was reformulated to extract its hidden underlying oscillatory behavior which turns out to be a major source for failure of dynamical systems. The presented algorithms has two sequential steps: training and detection. The classification of the status of each newly arrived data point as normal or abnormal depends on its location coordinates. These coordinates are determined by the application of a multi-scale Gaussian approximation procedure.

1 Introduction

Representation and understanding of real world datasets are the essence of machine learning and data mining methodologies. In this work, we introduce a novel framework for modeling high-dimensional heterogenous datasets. This framework addresses the following questions:

1. How to represent and process heterogenous dataset?
2. How to understand and query the data? How to fuse the data? How to find anomalies in data?

The core of the presented method is based upon diffusion processes, diffusion geometries and other methodologies for tracking and detecting meaningful geometric descriptions of geometric patterns that deviate from normality in the inspected data. It offers behavioral analysis of heterogeneous complex data to maintain and preserve systems’ health. Unsupervised anomaly detection techniques track and detect anomalies in an unlabeled dataset under the assumption that majority of the instances in the dataset are normal while achieving low false alarm rate.

Transforming a heterogenous dataset, which consists of a number of parameters (extracted from measurements and sensors) that are measured in different scales into a uniform workable setup, is done by the application of the Diffusion Maps. The Diffusion Maps algorithm, which was originally proposed as a dimensionality reduction scheme, is used to transform each parameter or sensor, into a new set of coordinates that reflect its geometric structure. This novel step of using the Diffusion coordinates as a re-scaling method allows us to construct a reliable and comparable representation of the dataset.

The Diffusion Maps framework [2] unifies ideas arising in a variety of contexts such as machine learning, spectral graph theory and kernel methods. Diffusion Maps have been applied to clustering and classification problems [13, 21]. In [12, 10], Diffusion Maps were used for data fusion and multicue data matching. Image processing and signal de-noising applications were implemented by diffusion methods in [19, 20]. Tomographic reconstruction was achieved in [3] by utilizing Diffusion Maps. There are only several published examples of utilizing Diffusion Maps to time series data. Classification of fMRI Time Series was done in [18, 14]. Anomaly detection and classification in hyper-networks by diffusion methodologies was presented in [5]. Recently, Diffusion Maps were used in [1] to model hurricane tracks and for detection of moving vehicles [17]. This work shows how to utilize the Diffusion Maps methodology to process in an unsupervised way high dimensional time series datasets, which dynamically change all the time.

The method is demonstrated on two different datasets, an synthetic one and a dataset that was recorded by a performance monitor of a transaction based system. The reliability of detection and tracking of failures in computerized systems and in transaction oriented systems in particular, influences the quality of service for short and long terms. Some expected failures can be fixed online by providing an advance warning through tracking of the
emerged problem. In other cases, post-processing of the problem can reveal the importance of each parameter to the operation of the entire system and to the relation between the parameters. Classification of failures is also possible since the parameters that caused the problem can be singled out.

Although network performance monitoring and automatic fault detection and prediction is somewhat related to the field of network intrusion detection, only a few research papers have been published in this field during the last few years. An automatic system for detection of service and failure anomalies in transaction based networks was developed in AT&T [6, 7, 9, 8]. The transaction-oriented system performs tasks that belonged to a few service classes, which are mutually dependent and correlated. Probability functions, which describe the average process time of the system’s transactions, were a basis for learning the system’s normal traffic intensities. Dynamic thresholds are generated and an anomaly is detected by deviation from these thresholds.

The proposed methods enhances the basic graph construction in the Diffusion Maps methodology. They provides an hierarchial (multi-scale) representation of the inspected data. This hierarchial representation provides better understanding of the data, more options how to group the data, more options for scaling the data, which is a critical pre-processing step. The unsupervised learning algorithm is a generic process that forms a base for studying complex dynamic systems of different types.

In this paper the work of [16] is extended by providing the following contributions:

- The sensor fusion model is simplified to a version that is more natural and does not require any re-scaling procedures.
- The anomaly detection procedure is modified to be independent of the fusion process.
- A novel multi-scale approximation for out-of-sample extension of the data is introduced.
- Empirical results on both synthetic and real data are presented.

2 Hierarchial Sensor Fusion

This section describes how to process and fuse a set of sensors, which collect homogenous or heterogenous data. Once the data is represented by an hier-
archial structure, we explain how to query the data for detecting abnormal dynamic trends.

2.1 Sensors Representation and Embedding

Let \( T = \{T_1, T_2, \ldots, T_K\} \) be the data collected by \( K \) sensors over a pre-determined number of time intervals \( t = N \). \( T \) is a matrix of size \( N \times K \). The column vector \( T_i \in T \) holds data from one sensor, which measures a specific activity in the system. The dataset \( T \) is processed in a bottom-up approach. It is represented by an hierarchical embedding tree. Figure 2.1 shows a four-level hierarchical tree, which fuses 11 input sensors. The fusing process combines groups of sensors in each level. The top level embeds the entire input.

![Hierarchical four level embedding model](image)

Figure 2.1: An example for an hierarchical four-level embedding tree that fuses 11 sensors.

The single sensors \( T_i \in T \), \( i = 1 \ldots K \) correspond to the \( K \) bottom level nodes of the tree. Each bottom level node will hold a low-dimensional embedding of the single sensor. The intermediate level nodes fuse the embedded data from the bottom level. These nodes describe sub-systems or sub-activities that span the dynamic process. The top level tree node fuses
data from the intermediate nodes. The top level node forms an embedding, denoted as the super-graph, which describes the entire system behavior in a low dimensional space. The super-graph gives a full description of the dynamic process. The hierarchical embedding tree provides a flexible structure, which can be adapted to the input data.

The processing in each node is done by the application of Diffusion Maps (DM) [2], these are described in Algorithm 1 and Algorithm 2.
Algorithm 1: Bottom level processing

**Input:** Sensor data \( T_i \in T \)

**Output:** Diffusion coordinates \( \Psi_i(x) \) that embed the dynamics of \( T_i \)

1: Define \( T_i \triangleq (t_i^1, t_i^2, \ldots, t_i^N)^T \).
2: By using a sliding window of length \( \mu \), which moves along \( T_i \), form \( N - \mu + 1 \) dynamic-paths \( T_i^r \)

\[
T_i^r = (t_i^r, t_i^{r+1}, \ldots, t_i^{r+\mu-1}) \quad r = 1, \ldots, N - \mu + 1 \quad (2.1)
\]

3: Construct the dynamic-matrix \( \tilde{T}_i \), which consists of \( r \) dynamic-paths as its rows, as

\[
\tilde{T}_i = \begin{pmatrix}
T_i^1 \\
\vdots \\
T_i^{N-\mu+1}
\end{pmatrix} \in \mathbb{R}^{(N-\mu+1) \times \mu} \quad (2.2)
\]

4: Construct a Gaussian kernel \( w(x, y) = e^{-\|x - y\|^2 / \sigma} \) to measures the pairwise similarity between the points in \( \tilde{T}_i \).
5: Normalize the kernel by

\[
\hat{w}(x, y) = \frac{w(x, y)}{q^{0.5}(x) q^{0.5}(y)}, \quad q(x) = \sum_{y \in T_i} w(x, y). \quad (2.3)
\]

6: Construct a transition matrix \( P \) such that

\[
p(x, y) = \frac{\hat{w}(x, y)}{d(y)}, \quad d(y) = \sum_{x \in \tilde{T}_i} \hat{w}(x, y). \quad (2.4)
\]

7: Compute the eigendecomposition of \( P \): If \( \{\phi_k\} \) and \( \{\psi_k\} \) are the corresponding left and right eigenvectors of \( P \), then, the eigendecomposition of the transition matrix is given by

\[
p(x, y) = \sum_{k \geq 0} \lambda_k \psi_k(x) \phi_k(y). \quad (2.5)
\]

8: Define the family of DM, which embeds \( \tilde{T}_i \), by

\[
\Psi_i(x) = (\lambda_1 \psi_1(x), \lambda_2 \psi_2(x), \lambda_3 \psi_3(x), \cdots). \quad (2.6)
\]

The diffusion coordinates \( \Psi_i(x) \) embed \( \tilde{T}_i \) into an Euclidean space. Usually, a small number of diffusion coordinates is sufficient to describe the
behavior of a single sensor. The coordinates $\Psi_i(x)$ organizes the sensor’s short-time dynamics. Frequent dynamic paths are embedded close together while abnormal paths have a small number of neighboring points in the embedded space.

The diffusion coordinates $\Psi_i$, which embed each sensor $T_i \in T_i$, consist of the set of eigenvectors $\{\psi_1(x), \psi_2(x), \psi_3(x), \ldots\}$, which have a norm that is equal to 1, multiplied by the corresponding eigenvalues $\{\lambda_1, \lambda_2, \lambda_3, \ldots\}$. The eigenvalues can be seen as weights for each diffusion coordinate. This makes the diffusion coordinates that embed the different sensors comparable. This process bypasses the need to scale the sensors (parameters, features) if heterogenous datasets are processed.

Algorithm 2 describes how to fuse embedded single sensors from the bottom level nodes to embeddings that describe the mutual behavior of a group of nodes according to the structure of the hierarchical tree. If the sensors are naturally separated into groups, like seen in Fig. 2.1 where there are 4 separate groups, Algorithm 2 is first applied to each group to form the embedding of the intermediate level nodes and then repeated for integrating all the sensors.

**Algorithm 2: High level processing**

**Input:** The embedding coordinates $\Psi_i, i = 1,\ldots, K$ that embed the bottom level nodes.

**Output:** Diffusion coordinates $\Phi(x)$ that embed the fusion of the sensors.

1: For each parent node, gather the diffusion coordinates that embed its child nodes into an input matrix $V = \{\Psi_1, \Psi_2, \Psi_3, \Psi_4, \ldots\}$.
2: Apply the DM to the matrix $V$.
3: The embedding coordinates $\Phi(x) = \{\nu_1\phi_1(x), \nu_2\phi_2(x), \ldots\}$ describe the mutual dynamic behavior of the group of sensors that is gathered by the high-level node.

## 2.2 Detection of Abnormal behavior

Section 2.1 described a method for representing the behavior of dynamically evolving systems that are monitored by a set of sensors. This was done by repeated embedding of the data into a low-dimensional space and fusing the embedding coordinates to form new embeddings that represent the joint behavior of the sensors. The DM coordinates form a reliable space for tracking and detecting abnormal system behavior. Normal system behavior is characterized by a large number of dynamic-paths (see Eq. (2.1)) that ap-
pear frequently. In the embedded space, these dynamic-paths correspond to points that are embedded close to one another. Abnormal dynamic-paths, which we wish to detect and track, have a small number of neighbors in the embedded space. In order to evaluate the appearance probability of the embedded points, a frequency score function is defined on the embedded space. Once evaluated, this score function will be used as a detector to measure the status of each sensor or group of sensors. We recall that the diffusion distance as was defined in [15, 2]. For a given dataset $\Gamma$, the diffusion distance between two data points $x$ and $y$, which belong to $\Gamma$, is the weighted $L^2$ distance

$$D^2(x, y) = \sum_{z \in \Gamma} \left( \frac{p(x, z) - p(z, y)}{\phi_0(z)} \right)^2.$$  

This distance reflects the geometry of the dataset. The value of $\frac{1}{\phi_0(x)}$ depends on the points’ density. Two points in the original space that have a large number of paths that connect between them, will be embedded close to each other in the low-dimensional space. Substituting Eq. (2.5) into Eq. (2.7) together with the biorthogonality property, then, the diffusion distance with the right eigenvectors of the transition matrix $P$ is expressed as

$$D^2(x, y) = \sum_{k \geq 1} \lambda_k^2 (\psi_k(x) - \psi_k(y))^2.$$  

The Euclidean distance between two points in the embedded space represents the distances between the two high-dimensional points as defined by a random walk.

Frequency score functions, which are denoted by $d(x)$ are constructed on the embedding coordinates that belong to each node of the hierarchical embedding tree. This construction results in an hierarchical score tree that detects the system’s state at each time point in different resolutions.

The score function is defined by

$$d(x) = \sum_{y \in S} ||\Psi(x) - \Psi(y)||, \quad S = \{l \text{ nearest neighbors of } x \text{ in } \Psi(x)\}.$$  

Normally behaved points will receive scores that are close to 0. Abnormally behaved points are detected by values that are rising above the average value of this function for the tested dataset. In general, different types of functions can be constructed on the embedding manifolds for different tasks. The constructed functions can emphasize and stress properties in the data that were uncovered by the DM coordinates.
3 Processing new sensor data

The hierarchical embedding structure can be extended to newly arrived data points. We introduce a general scheme for multi-scale function approximating and extension. The scheme will be used for extending the embedding coordinates in each node to newly arrived sensor data and for extending the score functions to handle out-of-sample extension. This multi-scale approximation and extension procedure is iterative. It uses Gaussians that become thinner at each iteration. The first iteration builds a coarse approximation of the given function. The next iterations approximate the residual with thinner Gaussian.

3.1 Multi-scale approximation and out-of-sample extension by Gaussians

Let $\Gamma = \{x_1, x_2, \ldots, x_n\}$ be a set of $n$ data points in $\mathbb{R}^N$. The function $f$ is defined on $\Gamma$. $f$ is extended to a newly arrived data point $y$, $y \in \mathbb{R}^N \setminus \Gamma$ (out-of-sample extension).

1. Set $\sigma$ to 1. For each point $x_k \in \Gamma$, $k = 1, \ldots, n$, define

$$f_0(x_k) = \frac{n}{s(x_i)} e^{-||x_k - x_i||^2 / \sigma^2} f(x_i) \quad \text{where} \quad s(x_i) = \sum_{j=1}^{n} e^{-||x_i - x_j||^2 / \sigma^2}$$

and $f_0$ is a coarse approximation of $f$.

2. Extend $f_0$ to the new point $y$ by

$$f_0(y) = \frac{n}{s(x_i)} e^{-||y - x_i||^2 / \sigma^2} f(x_i), \quad s(x_i) = \sum_{j=1}^{n} e^{-||x_i - x_j||^2 / \sigma^2}.$$  \hspace{1cm} (3.2)

3. Compute the residual $f_1 = f - f_0$ on $\Gamma$.

4. Extend $f_1$ to the new point $y$ by interpolating $f_0$ with a thinner Gaussian

$$f_1(y) = \frac{n}{s(x_i)} e^{-||y - x_i||^2 / \sigma^2} f_0(x_i), \quad s(x_i) = \sum_{j=1}^{n} e^{-||x_i - x_j||^2 / \sigma^2}.$$  \hspace{1cm} (3.3)
5. Compute the residual $f_2 = f - f_0 - f_1$ on $\Gamma$. If the residual is large, repeat steps 3 and 4 to approximate the new residual. Use a thinner Gaussian for each iteration. Otherwise, $f$ is approximated by $f_0 + f_1$.

6. The extension of $f$ to the point $y$ is given by $f(y) = f_0(y) + f_1(y) + f_2(y) + \cdots$.

### 3.2 Extension of the hierarchial embedding structure

The multiscale extension algorithm via summation of Gaussian, which was described in section 3.1, is applied to extend the hierarchial embedding structure, which was described in sections 2.1 and 2.2, to the newly arrived data points.

Let $t = \{t_1, t_2, \ldots, t_K\}$ be the new data points that were collected from the $K$ sensors $T = \{T_1, T_2, \ldots, T_K\}$, respectively. For each $t_i \in t$, $i = 1, \ldots, K$, construct a dynamic-path for $t_i$ by using the previous $\mu - 1$ sample points that were collected by the sensor $T_i$. Denote these new dynamic-paths by $y = \{y_1, y_2, \ldots, y_K\}$, $y_i$ is of size $1 \times \mu$. The multiscale algorithm from Section 3.1 is first applied to extend the embedding coordinates $\Psi_i$, which were generated for the sensor $T_i$, to $y_i$. Once $\Psi_i(y_i)$ are determined, the frequency score functions, which are defined on $\Psi_i$, are extended to the newly arrived points $y_i$.

Extending $\Psi_i$ to the new dynamic-path $y_i$, is carried out by setting $\Gamma$ (see section 3.1) to be the dynamic-matrix $\tilde{T}_i$ and iteratively compute $\Psi_i(y_i)$. Extending the frequency score function $d_i$ that is defined on $\Psi_i$ can be done directly by using Eq. (2.9).

### 4 Experimental Results

#### 4.1 Synthetic Example

The framework, which was described in sections 2 and 3, is first demonstrated on synthetic data. This example simulates a process with a typical behavior and with a particular structure. The process is sampled by 3 sensors, which are unaware of the structure of the generated data. The information from the 3 sensors is fused. Although this dataset is not a dynamic one, it demonstrates the strength of the method.

This process generates a collection of 12 numbers $\{c_1, \ldots, c_{12}\}$ in each
time step. The numbers are generated with the following structure:

$$\sum_{i=1}^{12} c_i = 100, \text{ and } c_1 = c_7, c_2 = c_8, \ldots, c_6 = c_{12}. \quad (4.1)$$

The generated dataset contains 3000 rows that obey this structure. Ten additional data rows are randomly generated by following the structure

$$\sum_{i=1}^{6} c_i = 50, \quad \sum_{i=7}^{12} c_i = 50. \quad (4.2)$$

These 10 rows are anomalous since the first 6 numbers are not a copy of the last 6 numbers. Figure 4.1 shows an example of a few rows from this synthetic dataset.

Figure 4.1: Samples from the synthetic dataset. The first 3 rows sum to 100 and the first (right) 6 numbers are equal to the last (left) 6 numbers. The last row is an example of an anomalous row. The row’s sum is equal to 100 while the distribution in the first 6 columns is not identical to that of the last 6 columns.

Three sensors, which sample the dataset, are defined. Denote the sensors as $S_1$, $S_2$ and $S_3$. The first sensor reads data columns 1, 2, 3, 4, the second sensor reads columns 5, 6, 7, 8 and the third sensor reads columns 9, 10, 11, 12. Figure 4.2 shows the information that is read by each sensor.
Although the three sensors do not capture the phenomena that governs the data directly, their fusion should express the equilibrium that exists in normal (legal) data rows. The abnormal rows should not stand out in each of the sensors, since their values are within the typical range of the sensor. The anomaly detection is seen after fusion process.

In order to make the problem even harder and to motivate the hierarchical construction and separate processing of each sensor at the bottom level, the following modifications of the first and third sensors were made:

- $S_1 = \log(S_1 + 3)$,
- $S_3 = -2 \times S_3 + \sin(\text{rand}([0,1]))$.

The diffusion based hierarchically learning Algorithms 1 and 2, which were described in Section 2.1, are applied to the three sensors. Since the data is not dynamic, Algorithm 1 starts at Step 4. A simple two-level hierarchical embedding tree is constructed to fuse the three sensors. An hierarchy of frequency score functions is constructed as described in Section 2.2. Anomaly detection is done by searching for outliers in the constructed frequency score functions. The anomaly that are constructed in this example should be detected by the score function that belongs to the top level node, which fuses the data.

Algorithm 1 is applied to each of the 4-dimensional sensors. The first three diffusion maps coordinates are stored. Denote the coordinated that
embed sensor $S_i$ as $\Psi_i(x) = \{\lambda_1\psi_1(x), \lambda_2\psi_2(x), \lambda_3\psi_3(x)\}$. A frequency score function (see Eq. (2.9)) is constructed based on the diffusion coordinates. The number of nearest neighbors, $l$ in Eq. (2.9), is set to 10. Figure 4.3 shows the application of the DM to the three sensors $S_1$, $S_2$ and $S_3$ and their fusion (super-graph) in the top level node.

Figure 4.3: The application of the hierarchical embedding process with the Diffusion Maps to the three sensors that capture the syntectic dataset. The top level node fuses the embeddings from the bottom level nodes.

The frequency score functions are constructed on each of the embeddings. Figure 4.4 shows the score functions that belong to each node. The $x$-axis is the time axis and the $y$-axis is the score axis. The last 10 points were anomaly points, like described in Eq. (4.2). The scores of the anomaly points in the bottom level nodes are within the normal range. When the sensors are fused, the anomalies can be detected due to their unbalanced structure between the sensors.
Figure 4.4: The frequency score functions that belong to the three sensors that capture the syntectic dataset. The top frequency score function reveals the anomalies that exist in the last 10 point of the data.

The proposed diffusion based hierarchically learning algorithm is compared with a similar algorithm that is applied with Multidimensional Scaling [4, 11]. In the hierarchical learning Algorithms 1 and 2, MDS replaces DM for embedding the sensor data.

Figure 4.5 shows the results from the hierarchical embedding process that uses MDS on the sensors. The bottom nodes hold the embeddings that were constructed from the three sensors. The top level is their fusion. The points are colored by their score values, red points are those that appear less frequently.
Figure 4.5: The application of the hierarchical embedding process with Multidimensional Scaling to the three sensors that capture the syntectic dataset. The top level node fuses the embeddings from the bottom level nodes.

Figure 4.6 shows the frequency score function that belongs to each node in the hierarchical structure. Compared to Figure 4.4, the application of MDS missed some of the anomaly points.
This example simulated a process that is sampled by three sensors. Like in many real life applications, the process is normal in an equilibrium state. The goal is to detect when the process gets out of balance. The application of the Diffusion Maps revealed the geometric structure of the data and the construction of the frequency score functions on this reliable embedded space detected the anomalous points.

4.2 Detection of anomaly trends in a dynamically evolving system

In this section the framework, which was introduced in Sections 2.1 and 2.2, is applied to high-dimensional data that was captures from a transaction based system. A performance monitor, which resides inside the system, captures 13 parameters (sensors) that record different activities in the system. The sensors are of different types and scales and are not comparable without applying a re-scaling procedure. The transaction based system behaves normal most of the time. Normal behavior is characterized by an
unknown number of stable states. The analysis is carried out based only on the inputs from the sensors. Each of the sensors captures partial data regarding the process. The goal is to fuse the sensors in order to detect anomalous system behavior. The anomalies are detected when the equilibrium between the sensors is unusual compared to the baseline.

Prior to applying the hierarchical sensor fusion method, as was described in Section 2, we explore the characteristic and geometry of the dataset in Section 4.2.1 for emphasizing the strength of non-linear dimensionality reduction methods.

### 4.2.1 Characterizing the geometry and structure of the dataset

This section analyzes the geometric structure of the high-dimensional dataset, $T$, which was captured from a performance monitor application. The data, which consist of 13 sensors, is reduced to a lower dimension by the application of Diffusion Maps, PCA and MDS. The input sensors $T = \{T_i\}$, $i = 1, \ldots, 13$ are re-scaled by dividing each sensor by its norm. Each re-scaled sensor is denoted by $T^*_i = \frac{T_i}{\|T_i\|}$, $i = 1, \ldots, 13$. Next, a dynamic-matrix is constructed on the set $T^* = \{T^*_1, \ldots, T^*_3\}$. A row in the dynamic-matrix consists of $\mu$ consecutive rows from the matrix $T^*$. The parameter $\mu$ is set to 3. Figure 4.7 show 3 images of the embedding of the dataset to the first three Diffusion Maps coordinates. The images plot the same dataset, but are rotated for allowing a better view of the embedded data. The data is colored by time. It can be seen that the data lies on a non-linear manifold and that the Diffusion coordinates organize the different system behaviors.

![Figure 4.7: Embedding of the dynamic-matrix $T^*$ to a 3-dimensional space by using the top three Diffusion coordinates. The three images show different rotations of the diffusion coordinates. The points, which are colored by time, are organized by the diffusion coordinates in a way that enables to track both the system’s normal and abnormal behavior.](image)
Figure 4.8 shows the application of PCA and MDS to the dynamic-matrix $T^\star$. The data points are colored according to the time evolution. Both methods do not capture the non-linear high-dimensional structure of the data. Although outliers are seen, it is difficult to characterize the system’s normal dynamic behavior.

![Figure 4.8: Left: Projection of the the dynamic-matrix $T^\star$ to a 3-dimensional space by using the first three principal components. Right: Embedding of the the dynamic-matrix $T^\star$ to a 3-dimensional space by the application of MDS.](image)

Comparing the results from the three dimensionality reduction algorithms show the advantage of using the Diffusion Maps, which unfolds the data’s geometric structures, for analyzing the given dataset.

### 4.2.2 Sensor fusion for performance monitoring

The implementation of the learning phase, which was described in Sections 2.1 and 2.2, can be represented by an hierarchical three-level embedding model (see also [16]) that is shown in Fig. 4.9. In each node, the Diffusion Maps embedding coordinates and a frequency score function are constructed from the data.
The sensors are separated into three groups, which measure different activities in the system (see Fig. 4.9). The first group consists of six sensors. These sensors measure the average response time from different transactions that run in the system in the previous time step. The measured scale (time resolution) is in minutes while the quantization step is ten seconds long. The sensors in this group are usually correlated. The second group of sensors measures the percentage of executed transactions that wait for a specific system’s resource. At each time step, the performance monitor tracks all the running transactions in the system and calculates the percentage of the running transactions that wait for a particular system’s resource like I/O, database access, etc. Anomalous behavior in the system can be expressed, for example, by an unusual distribution profile of the running transactions on the resources. The last group holds two sensors that capture the capacity (in percentage) of two different memories that the system uses. A middle level was added to the hierarchical structure in order to first embed the joint behavior of the groups and then embed the entire system by fusing these groups.

Figure 4.10 displays the embedding results of different tree nodes. In the bottom level the embeddings that were generated from the sensors $T_1$, $T_8$ and $T_{13}$ are presented. The points are colored by their frequency score function values. Blue points correspond to paths that appear frequently in
the dynamic matrix while red points belong to uncommon dynamic-paths. The middle level nodes presents the embedding that fuse the diffusion coordinates of their child nodes from the bottom level. The top level node presents the embedding of the entire system. The red points are anomaly points.

Figure 4.10: Hierarchical three-level embedding using Diffusion Maps for the performance monitor dataset.

Detecting anomaly behavior is done by checking the frequency score functions that are constructed on each embedding node. Figure 4.11 shows the score functions that belong to the middle and top level nodes. A system breakdown that occurred at the end of the training set is clearly seen at the top level score function. High score values, which are seen approximately 5 time steps before the crash, indicate of the system’s anomaly behavior. Notice that although the score functions of the middle level nodes are not high, the unusual balance between the system components is detected by fusing the subsystems in the top level node.
4.2.3 Comparisons with MDS

The goal of Multidimensional Scaling (MDS) is to provide a visual representation of proximities pattern (i.e., similarities or distances) among a set of objects. MDS [4, 11] represents a collection of techniques that maps the high dimensional data representation into a low-dimensional representation while retaining the pairwise distances between the data points as much as possible. The dimensionality reduction method in the learning algorithm, which was described in Section 2.1, is replaced by the classical MDS, which uses the pairwise Euclidean distances between data points as its input. Figure 4.12 displays the results of applying MDS to the nodes of the hierarchical structure, which was displayed in Fig. 4.9. When comparing Figure 4.12 with Figure 4.10, it is clear that the Diffusion Maps embedding capture and embed the data in a more informative manner, which preserves and reflects its original structure.
Figure 4.12: Hierarchical three-level embedding using MDS for the performance monitor dataset.

The frequency score functions are constructed from the MDS embeddings. Figure 4.13 displays the score functions that belong to the middle and top level nodes.
Figure 4.13: Frequency score function from the hierarchical three-level embedding model for the performance monitor dataset, using the MDS for embedding each node.

The system breakdown, which happened at the last time step of this training set is not captured by the top-level frequency score function that was constructed on the MDS embeddings (see Fig. 4.13). Figure 4.14 compares between the Diffusions Maps top level frequency score function and the MDS top level frequency score function. It can be seen that the functions behave differently and that the system breakdown, which is circled in red, is detected by the Diffusion Maps based embeddings and missed by the MDS based embedding method.
Figure 4.14: Top: The top level frequency score function from the Diffusion Maps based hierarchical embedding structure. Bottom: The top level frequency score function from the MDS based hierarchical embedding structure. A buildup towards a system breakdown, which occurred at the last time step is detected by the Diffusion Maps frequency score function.

5 Conclusion

In this paper, we introduced a general un-supervised data mining approach for detection and prediction of anomaly that are suitable for a wide range of dynamically evolving systems. A training step is carried out on a designated training dataset. The high-dimensional dynamic data is embedded into a low-dimensional space by using an hierarchical tree structure. This system is decomposed into sub-systems with different resolutions. The top node of the hierarchical tree describes the entire system, middle level nodes describe different sub-systems and bottom level node describe separately the dynamic behavior of each input parameter. Diffusion Maps algorithm is applied to each tree node to embed the sub-system into a lower dimension space. This embedded space is described by the Diffusion Maps coordinates. A score function, which is defined on the points in their embedded space, provides a measure to identify input data points as normal or abnormal. A multi-scale method that evaluates the status of newly arrived data points is also introduced. The embedding, which were constructed in the training step, can be extended online to embed the newly arrived data points. This allows us to detect online if the newly arrived data points are normal or abnormal.
Finally, an illustrative example of a system crash shows how the proposed approach can be used to predict that the system state is unstable.

References


