Robust Subspace Mixture Models for Anomaly Detection in High Dimensions

Oren Barkan and Amir Averbuch

Abstract—Robust subspace mixture models for density estimation of high dimensional data with an application for anomaly detection, are described. The essence of the proposed method is as follows: First, we learn a low dimensional representation using Whitened Principal Component Analysis (WPCA), then the source multidimensional data is projected onto this subspace and the density of its data points is estimated using a robust mixture model. Finally, the likelihood of the multidimensional data points is ranked according to the learnt density. Then, a method for online model adaption and parameter identification in the original feature space that accounts for the occurrence of a specific anomaly, is described.

Index Terms— anomaly detection, outlier detection, mixture models, robust statistics, model adaptation, parameter rating.

1 Introduction

In the last decade, with the explosion of data, the demand for data driven based methods for anomaly detection has increased substantially. Cyber security [10, 11], fraud detection [12] and industrial damage control [13, 14] are all examples for anomaly detection applications that operate on high dimensional big data. Several factors make anomaly detection in high dimension a challenging task: the lack of labeled data is a major issue, learning high dimensional distributions is a difficult problem, the boundary between normal and abnormal behavior is sometimes vague and in general depends on the specific domain. Furthermore, many scenarios exhibit data that evolve in time which means that what is currently considered as a normal behavior might be abnormal in future and vice versa [1].

During the years, many techniques have been developed and adopted for anomaly detection among them are Neural Networks [2, 3], Support Vector Machines [4], Rule-based systems [5-8], Clustering [9, 31], Information Theoretic methods [15] and Nearest Neighbor based methods [16]. In this paper, we focus on an unsupervised anomaly detection and propose a method for parametric density estimation. The method relies on the theoretical foundation of subspace mixture models [17] that does not necessitate domain expertise rules.

Mixture models for anomaly detection have been studied. Gaussian Mixture Models (GMM) were used for intrusion detection [18], anomaly detection in medical data [20] and in biomedical signal data [21]. Our work differs from previous ones in several aspects: First, we do not learn mixture models in the original feature space, i.e. we first find a low dimensional subspace that preserves most of the intrinsic information and estimates the density parameters in this subspace (Sections 2 – 4). Secondly, we describe a technique for adapting the mixture model parameters over time (Section 4.1). This technique is based on Maximum A-Posteriori (MAP) adaptation [22]. Third, we show how to identify the actual parameters in the original feature space that accounts for the occurrence of the anomaly. This is done by parameter rating (Section 5). Fourth, we propose a tradeoff between a point and a full posterior estimation by introducing mixture of models (Section 6). Lastly, we propose to use Diffusion Maps [23] and mixture of Student-t distributions [24] for robust anomaly detection (Section 7). In Section 8, we present experimental results that demonstrate the performance of the described methods.

- *Oren Barkan is with Tel Aviv University, Israel. E-mail: orenbarkan@post.tau.ac.il*
- *Amir Averbuch is with Tel Aviv University, Israel. E-mail: amir1@post.tau.ac.il*

2 Gaussian Mixture Model

A Gaussian Mixture Model (GMM) defines a Probability Density Function (PDF) of a d-dimensional vector $x \in \mathbb{R}^d$ as additive composition of multivariate Gaussians as follows:

$$p(x;\Theta) = \sum_{i=1}^{M} w_i N(x;\mu_i,\Sigma_i) \qquad (2.1)$$

where

$$\Theta = \{\Theta_i\}_{i=1}^{M},\ \Theta_i = \{w_i,\mu_i,\Sigma_i\}, \qquad (2.2)$$

w_i is the relative weight of the i-th mixture component with the constraint $\sum_{i=1}^{M} w_i = 1$ and $N(x;\mu_i,\Sigma_i)$ is the evaluation of the normal distribution with a mean vector $\mu_i \in \mathbb{R}^d$ and a covariance matrix $\Sigma_i \in \mathbb{R}^{d\times d}$ for the observation vector x s.t.

$$N(x;\mu_i,\Sigma_i) \triangleq \frac{1}{(2\pi)^{d/2}\left|\Sigma_i\right|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right).$$

The set of the GMM parameters Θ is estimated using the

xxxx-xxxx/0x/$xx.00 © 200x IEEE Published by the IEEE Computer Society

Expectation-Maximization (EM) algorithm [25] based on either Maximum Likelihood (ML) or MAP estimation.

Given a training set $X = \{x_k\}_{k=1}^{K} \subseteq \mathbb{R}^n$, we apply ML estimation which results in the following update steps:

$$w_i^{[t+1]} = \frac{\sum_{k=1}^{K} r_{ki}^{[t]}}{\sum_{j=1}^{M}\sum_{k=1}^{K} r_{kj}^{[t]}},$$

$$\mu_i^{[t+1]} = \frac{\sum_{k=1}^{K} r_{ki}^{[t]} x_k}{\sum_{k=1}^{K} r_{ki}^{[t]}}, \qquad (2.3)$$

$$\Sigma_i^{[t+1]} = \frac{\sum_{k=1}^{K} r_{ki}^{[t]} (x_k - \mu_i^{[t+1]})(x_k - \mu_i^{[t+1]})^T}{\sum_{k=1}^{K} r_{ki}^{[t]}}$$

where

$$r_{ki}^{[t]} = p(c = i \mid x_k; \Theta^{[t]}) = \frac{w_i^{[t]} N(x_k; \mu_i^{[t]}, \Sigma_i^{[t]})}{\sum_{j=1}^{M} w_j N(x_k; \mu_j^{[t]}, \Sigma_j^{[t]})}$$

and c is a categorical random variable which represents the probability of the mixture components.

3 Whitened Principal Components Analysis

Principal Component Analysis (PCA) [26] is a common method for subspace learning. For a given set $X = \{x_k\}_{k=1}^{K}$ ($x_k \in \mathbb{R}^n$), the learnt subspace is spanned by a set of orthogonal vectors $U = [u_1, ..., u_d]^T$ that accounts for the directions in space with most variability in X. U is derived by solving the eigenproblem $CU = \Lambda U$, where $C = K^{-1}\sum_{k=1}^{K}(x_k - \mu)(x_k - \mu)^T$ is the sample covariance matrix and $\mu = K^{-1}\sum_{k=1}^{K} x_k$ is sample mean. Then, by projecting X onto U, we obtain a d-dimensional representation $Y = \{y_k\}_{k=1}^{K}$ ($y_k \in \mathbb{R}^d$), where $y_k = \mathrm{U}\, x_k$ with a diagonal sample covariance matrix Λ. Whitened PCA (WPCA) is the application of PCA followed by whitening of the representation outcome. Since Λ is diagonal, we apply a subsequent normalization $\Lambda^{-1/2}$ in order to obtain a sample unit covariance matrix. Hence, the final representation for x_k is given by a mapping $H : \mathbb{R}^n \to \mathbb{R}^d$

$$H(x_k) = \Lambda^{-1/2} U (x_k - \mu). \qquad (3.1)$$

4 GMM Based Anomaly Detection

It has been shown [22] that a GMM with sufficient number of components is able to approximate any distribution type. However, for high dimensional data, the number of parameters that have to be estimated is quadratic in the original feature space dimension. Moreover, when the features are highly correlated, the inversion of the sample covariance matrices becomes ill conditioned.

We alleviate the above mentioned problems by learning a low dimensional WPCA subspace and projecting the source data onto this subspace. In this way, the number of parameters to be estimated is substantially reduced and the projected data is uncorrelated with the unit covariance. The intrinsic dimension of the training data is determined from the eigenvalues decay of the sample covariance matrix. Specifically, we apply a modified version of the profile likelihood method [27]. Finally, given a test set we rank the data points according to their log likelihood using Eq. (2.1) s.t.

$$\ell(x) = \log p(x; \Theta). \qquad (4.1)$$

4.1 Model Adaptation

In big data processing, there is a need to handle an extremely large amount of data which increases the demand for computational resources. Moreover, in some scenarios, the data is streamed over time. As a result, a learnt model must be consistently updated according to the newly arriving data points. A naïve solution would be to apply the EM procedure from the beginning to new and old data points together. However, the computational complexity of this solution increases over time. Therefore, we suggest a model adaptation method whose computational complexity depends only on the newly arrived data points. This is done by MAP adaption [22] of Θ.

Like the EM algorithm, MAP adaptation consists of two steps estimation process: the first step is identical to the expectation step in EM, in which sufficient statistics of the training data are computed for each Gaussian mixture. In the second step, however, both old and new sufficient statistics are incorporated into the estimation of the adapted GMM parameters using a data dependent mixing coefficients. The data dependent mixing coefficients are designed in such way that a mixture with high counts of new data points relies more on the new sufficient statistics Θ^{new} for parameter estimation, while a mixture with low data counts relies more on the old sufficient statistics Θ for parameter estimation. Next, we describe in detail the MAP adaptation process.

Given a trained GMM and a set $S = \{x_k\}_{k=1}^{K}$ of new data points, we first compute the responsibilities r_{ki} (Eq. 2.3) for each mixture component i and a data point x_k. Then, we use these responsibilities to compute the new sufficient statistics for Θ^{new} according to Eq. (2.3). Finally, we combine both the old and the new sufficient statistics to create

the adapted set of parameters $\tilde{\Theta}$ as follows:

$$\begin{aligned}\tilde{w}_i &= \left(\alpha_i^w w_i^{new} + (1-\alpha_i^w)w_i\right)\gamma, \\ \tilde{\mu}_i &= \alpha_i^\mu \mu_i^{new} + (1-\alpha_i^\mu)\mu_i, \\ \tilde{\sigma}_i &= \alpha_i^\sigma \sigma_i^{new} + (1-\alpha_i^\sigma)(\sigma_i + \mu_i^2) - \tilde{\mu}_i^2\end{aligned} \tag{4.2}$$

where γ is a normalization factor which ensures that the adapted weights sum to unity, α_i^ρ and $\rho \in \{w, \mu, \sigma\}$ are the data dependent adaptation coefficient that are computed by $\alpha_i^\rho = \frac{n_i}{n_i + \zeta^\rho}$, where $n_i = \sum_{k=1}^{K} N(x_k; \mu_i, \Sigma_i)$ is the data counts for the i-th mixture component and ζ is a relevance factor which is fixed for a parameter ρ. We further assume of having diagonal covariance matrices $\Sigma_i = diag\{\sigma_i\}$, where σ_i is the variance vector for the i-th mixture component. The proposed algorithm is described in Fig. 1.

Algorithm 1

Input:

Θ - *GMM parameters set (Eq. (2.2))*

$\zeta^\rho, \rho \in \{w, \mu, \sigma\}$ - *Relevance factors*

$S = \{x_k\}_{k=1}^K$ - *New data points*

Output:

$\tilde{\Theta}$ - *Adapted GMM parameter set.*

1. Apply a single estimation step to compute a set of new model parameters Θ^{new} using Eq. (2.3)

2. For $i \leftarrow 1$ to M

2.1. $n_i \leftarrow \sum_{k=1}^{K} N(x_k; \mu_i, \Sigma_i)$ (using Eq. (2.1))

2.2. $\alpha_i^\rho \leftarrow \frac{n_i}{n_i + \zeta^\rho}$, $\rho \in \{w, \mu, \sigma\}$

3. Compute the adapted GMM parameters set $\tilde{\Theta}$ using Eq. (4.2)

Fig. 1. Algorithm 1 – GMM model adaptation.

5 Parameter Rating

The method described in Section 4 provides a way to trace anomalies in the learnt subspace by using Eq. (4.1). However, in many scenarios, there is an interest to understand the source for the anomalies occurrences in the original feature space. We propose a method for parameter rating according to the learnt subspace.

5.1 The Proposed Parameter Rating

Denote the mapping from the original feature space X to the embedded space Y by $H: X \to Y$ and let $S = \{x_k\}_{k=1}^K \subseteq \mathbb{R}^n$. As described in Section 4, a data point $y_a = H(x_a)$ is a suspected anomaly when $\ell(y_a)$ (Eq. (4.1)) is extremely low. We further assume that y_a originated from a mixture component

$$i^* \triangleq \arg\max_i \{q_i(y_a) \mid 1 \le i \le M\} \tag{5.1}$$

where

$$q_i(y_a) \triangleq p(c = i \mid y_a; \Theta) \tag{5.2}$$

is the responsibility of the i-th mixture component for y_a We define the explanatory vector, which represents the parameters that accounts for the anomaly x_a, by

$$\bar{x}_a = \phi_{i^*}(x_a) \triangleq \Sigma_S^{-1/2} \left| x_a - \mathbf{E}_{q_{i^*}}[S] \right| \tag{5.3}$$

where $\mathbf{E}_{q_{i^*}}[S] = \sum_{k=1}^{K} q_{i^*}(y_k) x_k$, i^* is defined in Eq. (5.1) and Σ_S is the sample diagonal covariance matrix induced by S. In this way, $\bar{x}_a$ represents the scaled geometric difference vector between the anomaly x_a and the mean of the data points in the original feature space that is taken with respect to the association of the corresponding embeddings with the mixture component i^*. Then, we rate the parameters by their responsibility for the anomaly occurrence by sorting the entries in $\bar{x}_a$ in a descending order.

5.2 Soft Parameter Rating

The GMM assumes that each data point is generated from a single mixture component. However, when a suspected anomaly is examined, there are cases that we have a low confidence in the responsibility of a specific mixture component. Therefore, instead of aligning the suspected anomaly with a single component, we suggest the following alternative of computing $\bar{x}_a$

$$\bar{x}_a = \mathbf{E}_q\left[\phi(x_a)\right] = \sum_{i=1}^{M} q_i(y_a)\phi_i(x_a) \tag{5.4}$$

where M is the number of mixture components and q_i and ϕ_i are defined in Eqs. (5.2) and (5.3), respectively. Then, we continue in the same way as described in Section 5.1.

In this work, we learnt a mixture model in the WPCA subspace using Eq. (3.1).
This algorithm is described in Fig. 2.

Algorithm 2

Input:

y_a - *A detected anomaly.*

Output:

$\bar{x}_a$ - *The parameter vector.*

z - *The corresponding indices sorted in a descending order according to the entries in* b.

1. Compute the explanatory vector $\bar{x}_a$ using Eq. (5.3) or (5.4).

2. Sort the entries of $\bar{x}_a$ in a descending order and save the corresponding indices to z.

Fig. 2. Algorithm 2 – Parameter rating.

6 Mixture of Parametric Model

The EM algorithm results in maximum likelihood estimates and convergence is guaranteed. However, the EM algorithm does not guarantee to find a global optimum since the problem is non-convex and the final solution depends on the initial parameter values. Moreover, when fitting a mixture model using EM, we must specify a-priori the number of components.

A Bayesian approach for the predictive density is to assume that the parameters are drawn from a distribution and integrate over this distributions as follows:

$$p(x \mid S) = \int p(x \mid \Theta) p(\Theta \mid S) d\Theta \qquad (6.1)$$

where x is a new data point, $S = \{x_k\}_{k=1}^{K}$ are the observations and Θ are the model parameters. Usually, for complex models like GMM, there is no closed form solution for Eq.(6.1) which raises the need for having approximation methods like Variational Bayes [28] or Monte Carlo Markov Chains [29].

In this section, we describe a method that approximatef the predictive density in Eq. (6.1). Instead of training a single model, we train a set of independent mixture models $\Omega_M = \{\Theta^i\}_{i=1}^{L}$, where M is the number of components in each model. Then, we approximate the predictive density in Eq. (6.1) with $\tilde{p}(x \mid S) = \kappa^{-1} \sum_{i=1}^{L} p(x \mid \Theta^i) p(\mathrm{S} \mid \Theta^i)$, where, $\kappa = \sum_{i=1}^{L} p(\mathrm{S} \mid \Theta^i)$ is a normalization factor. Note that $\tilde{p}(x \mid S)$ is a valid PDF since it is a summation over PDFs with a proper normalization. The approximation is both due to the discretization of Θ over a finite parameter space and the approximation of the true posterior $p(\Theta^i \mid \mathrm{S}) = p(\mathrm{S} \mid \Theta^i) p(\Theta^i) / p(\mathrm{S})$ with the likelihood

$$p(\mathrm{S} \mid \Theta^i) = \prod_{k=1}^{K} p(x_k \mid \Theta^i). \qquad (6.3)$$

It is worth noting that the proposed method is easily parallelized (due to model independence) in the expense of computational resources. Furthermore, applying model adaptation (Section 4.1) is straightforward.

It is important to clarify that if we do not fix M, we cannot use the likelihood to approximate the posterior since it favors complex models and increases as $M \to \infty$. In this case, we can assume a uniform posterior and have a mixture of models with various complexities. An improvement would be to assume a prior $p(\Theta)$ and apply EM based MAP estimation [25]. This algorithm is described in Fig. 3.

Algorithm 3

Training phase

Input:

$S = \{x_k\}_{k=1}^{K}$ - *Training set.*

L - *Number of models to be trained.*

M - *Number of component in the model.*

Output:

$\Omega_M = \{\Theta^i\}_{i=1}^{L}$ - *A set of trained models parameters.*

$\Xi = \{\varphi_i\}_{i=1}^{L}$ - *Normalized likelihood of the estimated parameters.*

1. $\kappa \leftarrow 0$

2. For $i \leftarrow 1$ to L

2.1. Train a M -components mixture model using S and save the set of estimated parameters in Θ^i.

2.2. $p_i \leftarrow p(\mathrm{S} \mid \Theta^i)$ (Eq. (6.2)).

2.2. $\kappa \leftarrow \kappa + p_i$.

3. For $i \leftarrow 1$ to L

3.1. $\varphi_i \leftarrow \kappa^{-1} p_i$.

Test phase

Input:

x - *Test instance.*

$\Omega_M = \{\Theta^i\}_{i=1}^{L}$ - *A set of trained models parameters.*

$\Xi = \{\varphi_i\}_{i=1}^{L}$ - *Normalized likelihood of the estimated parameters.*

Output:

$\tilde{p}$ - *The approximated density value for* x.

1. $\tilde{p} \leftarrow \sum_{i=1}^{L} p(x \mid \Theta^i) \varphi_i$

Fig. 3. Algorithm 3 – Mixture of parametric models.

7 Robust Anomaly Detection

We assumed, so far, that the training data is noise free. However, real data is usually contaminated with outliers to some extent. A Gaussian is sensitive to outliers due to

its rapidly decaying tail shape which is a result of quadratic penalty in the exponent function. In order to improve the robustness to outliers, a heavy tail distribution is considered. A possible choice is the Laplace or Student t distribution [24]. In this work, we use the Student t distribution for two main reasons: it is closer to the shape of the normal distribution since it can be written as an infinite mixture of normal distributions with mutual mean and converges to a normal distribution for a specific parameter value.

In this section, we propose an algorithm for robust anomaly detection. The outline of the algorithm is as follows: first, we apply a preprocessing step that aims to roughly discover the manifold the data reside in. We assume that the discovered manifold contains high percentage of inliers. We discover the manifold by the application of Diffusion Maps [23] (see Section 7.2). This step enables us to find an initial coarse filtration of existing outliers. Then, for the subset of the revealed potential inliers we learn a Student t distribution mixture model (SMM). Finally, we use the parameters of the estimated SMM in order to derive a robust GMM parameter estimation.

7.1 Student t Distribution Mixture Model (SMM)

The multivariate Student t distribution is defined by

$$S(x;\mu,\Sigma,\nu) \triangleq \frac{\Gamma\left((\nu+d)/2\right)}{(\nu\pi)^{d/2}\left|\Sigma\right|^{1/2}\Gamma(\nu/2)}\left(\frac{(x-\mu)^T\Sigma^{-1}(x-\mu)}{\nu}+1\right)^{-\frac{\nu+d}{2}}$$

where $\mu \in \mathbb{R}^d$ is a mean vector, $\Sigma_i \in \mathbb{R}^{d\times d}$ is a positive definite scale matrix, $\Gamma(\cdot)$ is the Gamma function and $\nu \in [0,\infty)$ is the degrees of freedom parameter which controls the length of the tails. As ν goes to infinity the distributions tends to the normal distribution.

A more intuitive way to understand the multivariate Student t distribution is by looking at an infinite mixture of Gaussians with a shared mean and scaled covariances as follows:

$$S(x;\mu,\Sigma,\nu) = \int N(x;\mu,\Sigma/y)G(y;\nu/2,\nu/2)dy$$

where

$$G(y;\alpha,\beta) \triangleq \frac{\beta^\alpha}{\Gamma(\alpha)}\exp(-\beta y)\,y^{\alpha-1}.$$

This is a generative formulation: first, we scale the covariance by drawing y from the Gamma distribution and then generate x from the corresponding normal distribution.
The multivariate Student t distribution mixture model (SMM) is defined by

$$p(x;\Theta) = \sum_{i=1}^{M} w_i S(x;\mu_i,\Sigma_i,\nu_i)$$

where w_i is the relative weight of the i-th mixture component with the constraint $\sum_{i=1}^{M} w_i = 1$. The set of the SMM parameters, $\Theta = \{w_i,\mu_i,\Sigma_i,\nu_i\}_{i=1}^{M}$ is estimated using the Expectation-Maximization (EM) algorithm [24]. For the sake of completeness, we briefly describe the main update steps of the EM for SMM: Given a training set $\{x_k\}_{k=1}^{K}$, the responsibilities are computed by

$$r_{ki}^{[t]} = p(c=i \mid x_k;\Theta^{[t]}) = \frac{w_i^{[t]}S(x_k;\mu_i^{[t]},\Sigma_i^{[t]},\nu_i^{[t]})}{\sum_{j=1}^{M} w_j S(x_k;\mu_j^{[t]},\Sigma_j^{[t]},\nu_j^{[t]})}$$

and the expected scales are computed by

$$y_{ki}^{[t]} = \frac{\nu_i^{[t]}+d}{\nu_i^{[t]}+(x_k-\mu_i^{[t]})^T\left(\Sigma_i^{[t]}\right)^{-1}(x_k-\mu_i^{[t]})}.$$

Then, a new set of parameters is estimated according to the following equations:

$$\begin{aligned} w_i^{[t+1]} &= \frac{\sum_{k=1}^{K} r_{ki}^{[t]}}{\sum_{j=1}^{M}\sum_{k=1}^{K} r_{kj}^{[t]}}, \\ \mu_i^{[t+1]} &= \frac{\sum_{k=1}^{K} r_{ki}^{[t]} y_{ki}^{[t]} x_k}{\sum_{k=1}^{K} r_{ki}^{[t]} y_{ki}^{[t]}}, \\ \Sigma_i^{[t+1]} &= \frac{\sum_{k=1}^{K} r_{ki}^{[t]} y_{ki}^{[t]} (x_k-\mu_i^{[t+1]})(x_k-\mu_i^{[t+1]})^T}{\sum_{k=1}^{K} r_{ki}^{[t]}} \end{aligned} \tag{7.1}$$

where the degrees of freedom for the i-th component is computed as the solution for the following equation:

$$\log\left(\frac{\nu_i^{[t+1]}}{2}\right) - \psi\left(\frac{\nu_i^{[t+1]}}{2}\right) + 1 - \log\left(\frac{\nu_i^{[t]}+d}{2}\right) + \psi\left(\frac{\nu_i^{[t]}+d}{2}\right) + \frac{\sum_{k=1}^{K} r_{ki}^{[t]}(\log y_{ki}^{[t]} - y_{ki}^{[t]})}{\sum_{k=1}^{K} r_{ki}^{[t]}} = 0.$$

where $\psi(x) \triangleq \frac{\partial\Gamma(x)}{\partial x}$ is the digamma function. We refer the reader to [24] for a detailed derivation of the EM algorithm for SMM.

7.2 Diffusion Maps

Diffusion Maps (DM) [23] is a machine learning technique for non-linear dimensionality reduction. Different from other dimensionality reduction methods such as principle component analysis (PCA), DM is a non-linear method that focuses on discovering the underlying manifold that the data has been sampled from.

In DM, a graph affinity matrix is built that is used to generate a diffusion process on it. As the diffusion process advances, it integrates local geometries to reveal intrinsic geometric structures of the data at different scales. Based on the revealed geometry, one can measure the similarity between two data points at a specific scale. DM embed the high dimensional data into a low dimensional space D such that the Euclidean distance between data points in D approximates the diffusion distance in the original feature space. The dimension of D is determined from the underlying geometric structure of the data and the accuracy by which the diffusion distance is approximated.

In our setup, the original feature vectors reside in a g-dimensional space G. DM is applied to G to map the feature vectors to l-dimensional in a space D ($l \ll g$). In the rest of this section we explain the DM algorithm in more detail.

Given a training set $\{x_k\}_{k=1}^{K} \subseteq G$, the first step defines an affinity metric $c(x_i, x_j)$ over G. Then, this metric is converted to a similarity measure. A common approach for this type of conversion uses the Gaussian kernel, $\chi(x_i, x_j) = \exp\left(-c(x_i, x_j)^2 / \sigma\right)$ where the parameter σ determines the scale or size of the neighborhood which we trust our local similarity measure to be accurate in. A method for automatic configuration of σ was proposed in [30].

In this way, we can define a full undirected graph where the data points are the nodes and the weights of the edges are determined according to the chosen kernel. We then define a random walk on this graph by converting the similarity measure to a probability function by $p(x_i, x_j) = \chi(x_i, x_j) / \sum_{h=1}^{K} \chi(x_i, x_h)$. Next, we form a transition Markov matrix P in which the entry $P_{i,j} = p(x_i, x_j)$ is the probability of transition from node x_i to node x_j in a single step. In the same way, P^t is a matrix in which the entry $P_{i,j}^t$ is the transition probability from node x_i to node x_j in t steps.

Based on the above construction, the diffusion distance after t steps is defined by $Q_t(x_i, x_j) = \sum_{k=1}^{K} (P_{i,k}^t - P_{j,k}^t)^2$. By applying spectral decomposition to P we obtain a complete set of eigenvalues $1 = \lambda_0 \geq \lambda_1 \geq \ldots \geq \lambda_K$ while left and right eigenvectors satisfy $Pv_i = \lambda_i u_i$.

We then define a mapping $M_t : \{x_k\}_{k=1}^{K} \to D$ to be

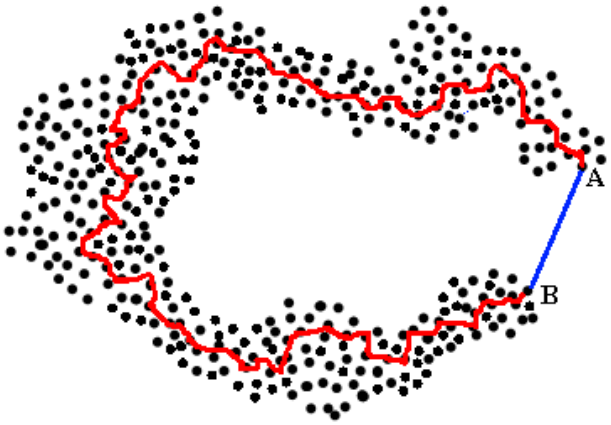

Fig. 4. As the diffusion process progresses, travelling along the red path (which follows the intrinsic geometry of the manifold) becomes more likely than travelling along the blue one.

$$M_t(x_k) = \left[\lambda_1^t v_{1k}, \ldots, \lambda_l^t v_{lk}\right]^T \qquad (7.2)$$

where v_{ik} indicates the k-th element of the i-th eigenvector of P and l is the dimension of the diffusion space D. It was shown [23] that for $l = g - 1$, $\left\|M_t(x_i) - M_t(x_j)\right\|_2^2 = Q_t(x_i, x_j)$ holds. Of course, in practice one should pick $l < g - 1$ according to the spectral decay of $(\lambda_k)_{k=1}^{K}$. This decay is determined from the complexity of the intrinsic dimensionality of the data and the choice of the parameter σ.

Figure 4 is an illustrative example for a diffusion process. Two alternative paths connect data points on the manifold. The red path is longer but it follows the geometric structure of the manifold while the blue path is short but does not follow the manifold structure. As the number of steps t in the diffusion process increases, the probability of travelling along the red path also increases, since it consists of many short distance jumps. However, the probability of travelling along the blue path stays always small (and becomes smaller and smaller as t increases) as it associated with long distance jumps.

7.3 Robust Mixture Model (RMM)

SMM in its nature is more robust to outliers than GMM. However, if a large number of outliers exists in the data, EM will produce an inferior SMM model. Moreover, for the task of anomaly detection, GMM, which is trained on clean data, will assign significant lower probabilities to outliers than SMM. Therefore, we look for a tradeoff between robust model and low tolerance for outliers. To this end, we propose to learn the weights, the means and the covariances of the GMM using SMM to ensure robust parameter estimation. Then, we use these sufficient statistics to form a GMM.

Although SMM is less sensitive to outliers than GMM, in scenarios where the training set contains high percentage of outliers, even a heavy tail distribution will be drastically biased towards the outliers and a poor model is likely to be generated. We propose to address this problem by applying a filtration step to the training set. This is done by measuring the belonging of each data point to the manifold and discarding all data points that are suspected as outliers. The measure is based on a learnt DM between the original feature space and the diffusion space. We assume that the manifold is connected and contains at least 50% of the data points i.e. less than 50% of the data points are outliers. DM provides a representation in which the data points are clustered according to their connectivity. In the diffusion space, inliers are expected to be clustered together, whereas outliers might be spread between several small clusters or scattered randomly. As long as the above assumption holds, we can identify the inliers by discovering the biggest connected component (the one that contains at least 50% of data points) in the diffusion space. To this end, we apply DM to the data points $X = \{x_k\}_{k=1}^{K}$ and obtain the embedding $Y = \{y_k\}_{k=1}^{K}$ using Eq. (7.2). Then, we define a graph on Y, where the vertices are Y and $A_{ij} = \mathbf{1}_{a(i,j)>\tau}$ is the adjacency matrix, where $\mathbf{1}$ is the indicator function with respect to the condition in the subscript,

$$a(i,j) = \frac{\left| \left\| y_i - y_j \right\| - \mu_\varsigma \right|}{\sigma_\varsigma}, \quad \mu_\varsigma = K^{-1} \sum_{k=1}^{K} \delta_{k\varsigma},$$
$$\sigma_\varsigma = \sqrt{K^{-1} \sum_{k=1}^{K} (\delta_{k\varsigma} - \mu_\varsigma)^2}, \quad \delta_{k\varsigma} = \sum_{\upsilon=1}^{\varsigma} \left\| y_k - y_{\Upsilon_{k\upsilon}} \right\|, \tag{7.3}$$

where $\Upsilon \in \{1,...,K\}^{K \times \varsigma}$ is an Euclidean nearest neighbor matrix in which the entry $\Upsilon_{k\upsilon}$ contains the index of the υ nearest neighbor of y_k, ς and τ are predetermined constants. Then, we run Breadth First Search (BFS) on Y in order to reveal the biggest connected component in the graph $Y_b \subseteq Y$ and consider Y_b as a set of inliers. Finally, $X_b = \{x_k \mid y_k \in Y_b\}$ is fed as an input into SMM ML parameter estimation (Section 7.1) and the relevant parameters are used to form a GMM. The reason we chose to output a GMM is due to the rapidly decaying tail of a Gaussian which ensures outliers are given extremely low PDF values. This algorithm is described in Fig. 5.

Algorithm 4

Input:

$X = \{x_k\}_{k=1}^{K}$ - *A training set*

M - *Number of components in the output model*

ς - *Number of nearest neighbors to consider*

τ - *Threshold parameter (standard deviation units)*

Output:

$\{w_i, \mu_i, \Sigma_i\}_{i=1}^{M}$ - *The GMM parameter estimates.*

1. Obtain $Y = \{y_k\}_{k=1}^{K}$ (Eq. (7.2)) by computing DM (Section 7.2) for X.

2. Compute the Euclidean nearest neighbor matrix $\Upsilon \in \{1,...,K\}^{K \times \varsigma}$ over Y.

3. For $i \leftarrow 1$ to K

3.1. $\delta_{k\varsigma} \leftarrow \sum_{\upsilon=1}^{\varsigma} \left\| y_k - y_{\Upsilon_{k\upsilon}} \right\|$

3.2. $\mu_\varsigma \leftarrow \mu_\varsigma + \delta_{k\varsigma}$

4. $\mu_\varsigma \leftarrow K^{-1} \mu_\varsigma$

5. $\sigma_\varsigma \leftarrow \sqrt{K^{-1} \sum_{k=1}^{K} (\delta_{k\varsigma} - \mu_\varsigma)^2}$

6. For $i \leftarrow 1$ to K

6.1 For $j \leftarrow i$ to K

6.1.1 $A_{ij} \leftarrow \mathbf{1}_{a(i,j)>\tau}$ (Eq. (7.3))

6.1.1 $A_{ji} \leftarrow A_{ij}$

7. Find the biggest connected component $Y_b \subseteq Y$ using the adjacency matrix A.

8. $X_b \leftarrow \{x_k \mid y_k \in Y_b\}$

9. Train a M - component SMM model (Eq. (7.1)) using X_b to obtain a set of parameters $\{w_i, \mu_i, \Sigma_i, \nu_i\}_{i=1}^{M}$.

Fig. 5. Algorithm 4 – Robust mixture model (RMM).

8 Experimental Results

In this section, we demonstrate the performance of the proposed algorithms. As an example, we used the synthetic 2D manifold (green points) in Fig. 6 (a) that is generated by drawing 200 points from a two GMM components. In Fig. 6 (b-d) we added 200 outliers (red points) to the data and trained two-component GMM (Section 2), SMM (Section 7.1) and RMM (Algorithm 4). We see that both GMM (b) and SMM (c) are affected by these outliers and exhibit poor PDF estimation. However, RMM (d) managed to capture the manifold structure by filtering most of the outliers and produces a PDF which is similar to the original one.

In Fig. 7(a), we present the embeddings Y after the application of DM to X. The points associated with the biggest connected component Y_b are marked in black. We see that all the green points are marked in black together with a few outliers. In Fig. 7(b), we present the original data points X, where the data points associated with X_b are marked in black. This filtration step managed to substantially reduce the number of outliers. Thus, by learning SMM on X_b, we are able to estimate robust sufficient statistics for a GMM (Algorithm 4).

In the next experiment, we continue to evaluate the performance of Algorithm 4, this time, on DARPA 1999 labeled dataset [32]. In this dataset, the supplied training set is clean (no attacks) and only the test sets include outliers (attacks).

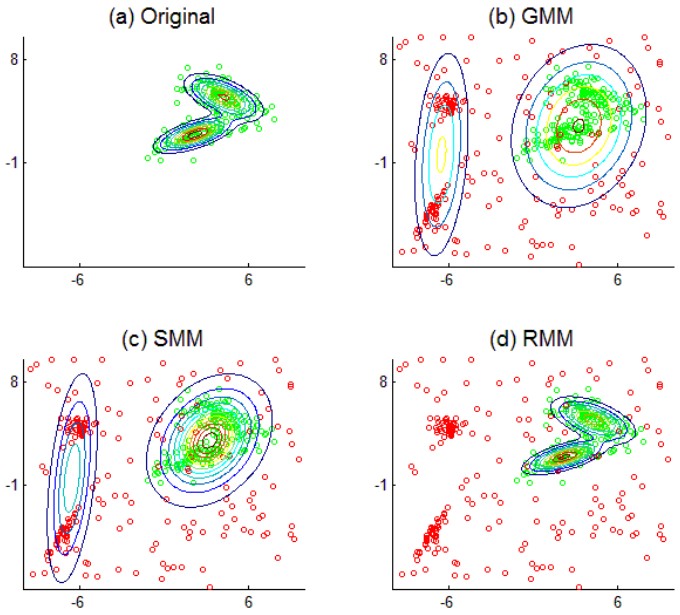

Fig. 6. (a) The original data points are generated by a two component GMM. (b-d) The contour of the PDF estimated by GMM, SMM and RMM, respectively, for the original 200 points and the addition of 200 outliers.

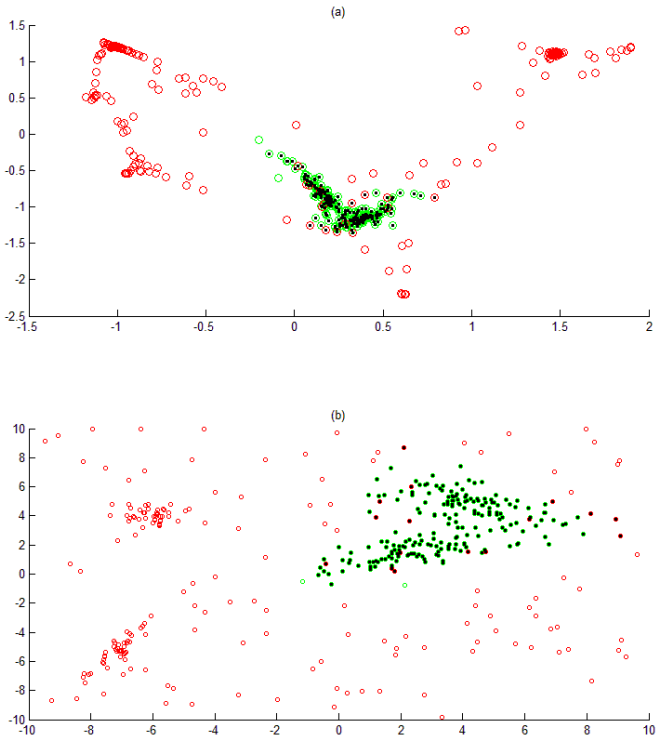

Fig. 7. Green and red points are the true inliers and outliers, respectively. Points associated with the biggest connected component are marked in black. (a) Diffusion space. Y_b are marked in black (b) Original feature space. X_b are marked in black.

Therefore, we learnt the test sets directly in order to demonstrate the ability of RMM to filter outliers (we observed that the samples that are labeled as inliers exhibit the same distribution for both training and test sets). In Fig.

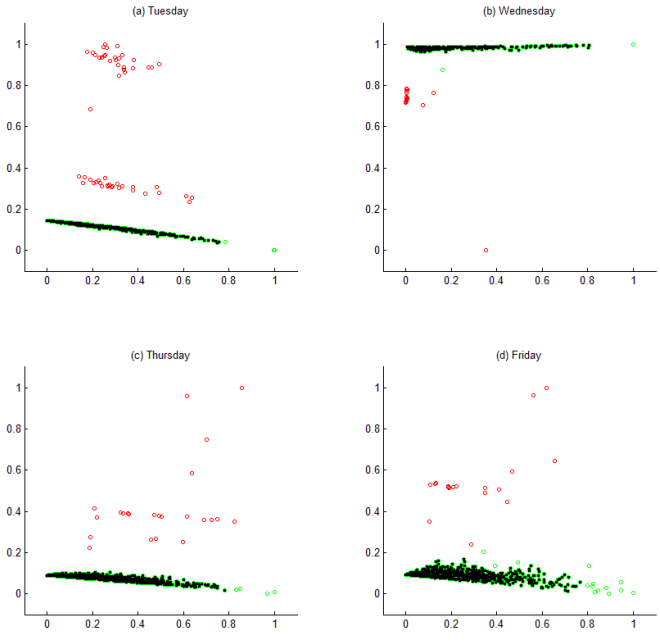

Fig. 8. DARPA 1999 test sets: Green and red points are the true inliers and outliers, respectively. Points associated with the biggest connected component are marked in black. (a-d) Diffusion space for various test sets (each belong to a different day), Y_b is marked in black.

8 we present the analysis of Algorithm 4 for four different test sets (Tuesday, Wednesday, Thursday and Friday attacks). The analysis is similar to the one that is described in Fig. 7(a). We see that RMM managed to identify most of the inliers (black markers) and filter most of the outliers (red circles). Since the original data points are of dimension 14, we do not present the RMM PDF contour. In all of the RMM experiments, we used the same parameter configuration $\varsigma = 1, \tau = 3$ (Eq. (7.3)).

Next, we evaluate the performance of Algorithm 2 for parameter rating. We drew 100 data points from a 10-dimensional GMM with two even weights, mixture components C_1, C_2 with means $-\vec{1}\cdot 20$ and $\vec{1}\cdot 20$, respectively, and identical covariance. Independently, from each component, we drew 50 data points. We generated two outliers by setting to 15 the second entry in the first data point and setting to zero the first four entries in the second data point. These data points were originated from C_1. Then, we applied WPCA (Eq. (3.1)) to reduce the dimension to three. In Fig. 9, we see the embedding of the data points in the WPCA subspace (only the first two dimensions are presented). Data points in green and red refer to inliers and outliers, respectively. By learning a two component GMM in the WPCA subspace and by examining the log likelihood (Eq. (4.1)) of the data points, we can trace the outliers $y_1 = (0,0)$ and $y_2 = (-24,15)$ that received the lowest log likelihood values. Applying Algorithm 2 to y_1 and y_2 results in

$$\bar{x}_1 = (3.6, 34, 1.8, 0.9, 6.6, 8.7, 6.1, 0.9, 6.9, 0.2),$$
$$\bar{x}_2 = (20.4, 19, 19.1, 19.1, 0.6, 0.5, 0.8, 1.4, 1.4).$$

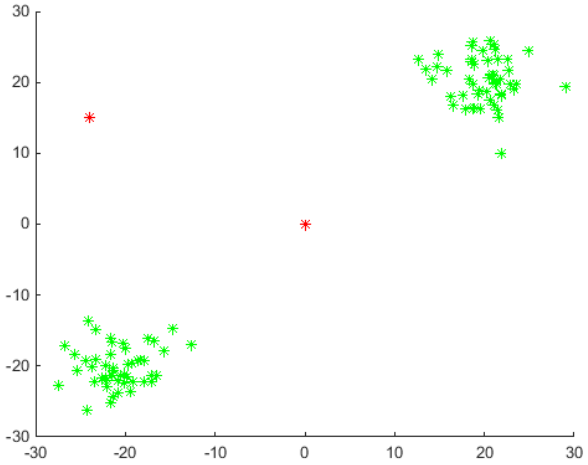

Fig. 9. A plot of the first two WPCA dimensions. Points in green and red are inliers and outliers, respectively.

We see that the largest entry in $\bar{x}_1$ is the second entry, 34, which corresponds to the second entry in x_1, 15. From examining Fig. 9, we wonder why the first entry in $\bar{x}_1$ is not as high as the second entry. This is due to the fact that x_1 is generated from C_1 and therefore $r_{11} \gg r_{12}$. In the same way, we see that the first four entries in $\bar{x}_2$ are significantly higher than the rest of the entries. This indicates the abnormal values in the first four entries of x_2.

9 Conclusion

In this paper, we propose several methods for density estimation which are applicable for anomaly detection in high dimensional data. We estimated the density in a low dimensional representation (WPCA) using the parametric mixture models GMM, SMM and RMM. We show that by using these models, we are able to apply a model adaptation and combined several different models in a probabilistic manner.

We derived methods for parameter rating and demonstrated their performances. Finally, we introduced the RMM as a tool for robust anomaly detection when the training data is contaminated with outliers.

Acknowledgements

This research was partially supported by the Israeli Ministry of Science & Technology (Grants No. 3-9096 and 3-10898), by US - Israel Binational Science Foundation (BSF 2012282) and by the Blavatnik Computer Science Research Fund.

References

[1] Chandola, V., Banerjee, A. and Kumar, V. Anomaly detection: A survey. ACM Computing Surveys (CSUR) 41.3 (2009), 15-67.

[2] Markou, M. and Singh, S. Novelty detection: a review-part 2: neural network based approaches. Signal Processing 83, 12, (2003) 2499-2521.

[3] Ghosh, A. K., Wanken, J., and Charron, F. Detecting anomalous and unknown intrusions against programs. In Proceedings of the 14th Annual Computer Security Applications Conference. IEEE Computer Society, 259-267 (1998).

[4] Ratsch, G., Mika, S., Scholkopf, B., and Muller, K. R. Constructing boosting algorithms from svms: An application to one-class classification. IEEE Transactions on PatternAnalysis and Machine Intelligence 24, 9, (2002) 1184-1199.

[5] Wong, W.K., Moore, A., Cooper, G., and Wagner, M. Rule-based anomaly pattern detection for detecting disease outbreaks. In Proceedings of the 18th National Conference on Artificial Intelligence. MIT Press (2002), 217-223.

[6] Lee, W. and Stolfo, S. Data mining approaches for intrusion detection. In Proceedings of the 7th USENIX Security Symposium. San Antonio, TX (1998), 120 - 132.

[7] Lee, W., Stolfo, S., and Chan, P. Learning patterns from unix process execution traces for intrusion detection. In Proceedings of the AAAI 97 workshop on AI methods in Fraud and risk management (1997), 50-56.

[8] Lee, W., Stolfo, S. J., and Mok, K. W. Adaptive intrusion detection: A data mining approach. Artificial Intelligence Review 14, 6, (2000) 533-567.

[9] Allan, J., Carbonell, J., Doddington, G., Yamron, J., and Yang, Y. Topic detection and tracking pilot study. In Proceedings of DARPA Broadcast News Transcription and Understanding Workshop. (1998) 194-218.

[10] Mahoney, M. V. and Chan, P. K. Learning nonstationary models of normal network traffic for detecting novel attacks. In Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM Press, (2002) 376-385.

[11] Mahoney, M. V. and Chan, P. K. Learning rules for anomaly detection of hostile network traffic. In Proceedings of the 3rd IEEE International Conference on Data Mining. IEEE Computer Society, (2003) 601 - 612.

[12] Brause, R., Langsdorf, T., and Hepp, M. Neural data mining for credit card fraud detection. In Proceedings of IEEE International Conference on Tools with Artificial Intelligence, (1999) 103-106.

[13] Keogh, E., Lin, J., Lee, S. H., and Herle, H. V. Finding the most unusual time series subsequence: algorithms and applications. Knowledge and Information Systems 11, 1, (2006) 1-27.

[14] Keogh, E., Lonardi, S., and chi' Chiu, B. Y. Finding surprising patterns in a time series database in linear time and space. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM Press, New York, NY, USA, (2002) 550-556.

[15] Lee, W. and Xiang, D. Information-theoretic measures for anomaly detection. In Proceedings of the IEEE Symposium on Security and Privacy. IEEE Computer Society, (2001) 130 - 142.

[16] Ertoz, L., Eilertson, E., Lazarevic, A., Tan, P. N., Kumar, V., Srivastava, J., and Dokas, P. MINDS - Minnesota Intrusion Detection System. In Data Mining - Next Generation Challenges and Future Directions. MIT Press (2004).

[17] McLachlan, G., and Peel, D. Finite mixture models. John Wiley & Sons, 2004.

[18] Gonzalez, F. A. and Dasgupta, D. Anomaly detection using real-valued negative selection. Genetic Programming and Evolvable

Machines 4, (2003) 383-403.

[19] Kruegel, C., Mutz, D., Robertson, W., and Valeur, F. Bayesian event classification for intrusion detection. In Proceedings of the 19th Annual Computer Security Applications Conference. IEEE Computer Society, (2003) 14-23.

[20] Laurikkala, J., Juhola1, M., and Kentala., E. Informal identification of outliers in medical data. In Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology. (2000) 20-24.

[21] Roberts, S. Extreme value statistics for novelty detection in biomedical signal processing. In Proceedings of the 1st International Conference on Advances in Medical Signal and Information Processing. (2002) 166-172.

[22] Reynolds, D. A., Quatieri, T. F. and Dunn, R. B. Speaker verification using adapted Gaussian mixture models. Digital signal processing 10.1 (2000): 19-41.

[23] Coifman, R. R., and Lafon, S. Diffusion maps. Applied and computational harmonic analysis 21.1 (2006): 5-30.

[24] Peel, D. and McLachlan, G. Robust mixture modelling using the t distribution. Statistics and computing 10.4 (2000): 339-348.

[25] Dempster, A. P., Laird, M. N., and Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological) (1977): 1-38.

[26] Jolliffe, I. Principal component analysis. John Wiley & Sons, Ltd, 2005.

[27] Zhu, M. and Ghodsi, A. Automatic dimensionality selection from the scree plot via the use of profile likelihood. Computational Statistics & Data Analysis 51.2 (2006): 918-930.

[28] Attias, H. Inferring parameters and structure of latent variable models by variational Bayes. Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., (1999), 213-252.

[29] Gilks, W. R. Markov chain monte carlo. John Wiley & Sons, Ltd, 2005.

[30] Singer, A. From graph to manifold Laplacian: the convergence rate, Applied and Computational Harmonic Analysis, 21 (1), (2006) 135-144.

[31] David, G. and Averbuch, A. SpectralCAT: Categorical spectral clustering of numerical and nominal data. Pattern Recognition 45.1 (2012) 416-433.

[32] Lippmann, R., Haines, J. W., Fried, D. J., Korba, J., & Das, K. The 1999 DARPA off-line intrusion detection evaluation. Computer networks, 34(4), (2000) 579-595.