A least-squares approach to anomaly detection in static and sequential data

John A. Quinn[a], Masashi Sugiyama[b]

[a]*Department of Computer Science, Makerere University, PO Box 7062, Kampala, Uganda*
[b]*Department of Computer Science, Tokyo Institute of Technology, Tokyo 152-8552, Japan*

Abstract

We describe a probabilistic, nonparametric method for anomaly detection, based on a squared-loss objective function which has a simple analytical solution. The method emerges from extending recent work in nonparametric least-squares classification to include a "none-of-the-above" class which models anomalies in terms of non-anamalous training data. The method shares the flexibility of other kernel-based anomaly detection methods, yet is typically much faster to train and test. It can also be used to distinguish between multiple inlier classes and anomalies. The probabilistic nature of the output makes it straightforward to apply even when test data has structural dependencies; we show how a hidden Markov model framework can be incorporated in order to identify anomalous subsequences in a test sequence. Empirical results on datasets from several domains show the method to have comparable discriminative performance to popular alternatives, but with a clear speed advantage.

Keywords: Anomaly detection, Outlier detection, Robust classification, Sequential data, Kernel models

1. Introduction

Anomaly detection is useful in several practical situations where test data may be subject to unexpected regimes, for example due to sensor failures, malicious user behavior, or external changes to the system being modeled. In this letter we focus on the form of the problem in which training samples without anomalies are provided, and the task is to calculate anomaly scores for test data. This is distinct from the case in which a dataset contains a mixture of inliers and outliers, and the task is to separate them (often referred to as outlier detection, though note that some authors use the terms "anomaly detection" and "outlier detection" interchangeably).

We propose a novel nonparametric method for addressing this problem, based on the recently introduced least-squares probabilistic classifier (LSPC) (Sugiyama, 2010). As well as having the flexibility and discriminative power of a kernel model, our method is fast at training time, due to the convexity of ℓ_2 loss, and very fast at test time, simply requiring a weighted average of kernel basis functions for inference. If training data is labeled with multiple inlier classes, the method can also be used for robust classification, i.e. for each test datapoint we can calculate the probability of that point belonging to each of the inlier classes as well as to the outlying, anomaly class. Furthermore, being a probabilistic method it is straightforward to incorporate into models where the test data has structural dependencies; we demonstrate how it can be incorporated into a hidden Markov model framework in order to apply it to anomaly detection in sequences.

In the remainder of this letter, we first review related work for anomaly detection and the least-squares approach for probabilistic classification, then show in Section 4 how the least-squares formulation can be extended to assign a probability to a test input of it being anomalous. In Section 5 we explain how this can be incorporated into a hidden Markov model (HMM) framework in order to identify anomalies in sequential data. We give exper-

Email addresses: jquinn@cit.ac.ug (John A. Quinn), sugi@cs.titech.ac.jp (Masashi Sugiyama)

imental results for the static anomaly detection method in Section 6 on several standard datasets, showing it to have competitive accuracy and superior speed compared to alternative methods, and illustrate sequential anomaly detection on time series from medicine and engineering.

The Python implementation of the method, including demonstrations and code to recreate the experiments described here, is available at `http://cit.mak.ac.ug/staff/jquinn/software/lsanomaly.html`.

2. Related work

There are many existing methods for anomaly detection, for which an extensive review can be found in Chandola et al. (2009). Different assumptions might be made about the distribution of anomalous points relative to the training, inlier points, which yield different methods. For instance, an assumption that anomalous datapoints have a large distance from any of the training points leads to the use of k-nearest neighbor methods for anomaly detection. An alternative is to make assumptions regarding clusters in the data, e.g. that normal data points belong to clusters, whereas anomalous data points do not, or that normal data points are usually closer to the nearest cluster centroid than anomalous data points. Statistical assumptions might also be made, e.g. that normal data points occur in high-probability regions of the data space (according to some stochastic model), whereas anomalous data points occur in low-probability regions.

Kernel models have been used in a number of anomaly detection schemes. For example, kernel density estimation can be applied to data from the normal regime; a low estimated density for test points indicates anomaly. Kernel recursive least-squares has been used for anomaly detection by Ahmed et al. (2007), in order to calculate a codebook of vectors which represent the support of the normal regime. Multi-scale kernel regression for anomaly detection was proposed by Gao et al. (2010), in which the length scales in the kernel model of normality are varied according to the distances between training samples. Clustering in kernel space can also be used to characterize the normal regime, providing stability improvements over standard methods (Filippone et al., 2010). Gaussian process models can also be used for kernel-based outlier detection (Kemmler et al., 2010).

Our work begins with similar assumptions about the nature of outliers as used in the one-class support vector machine (Schölkopf et al., 1999) and the kernel Fisher discriminant method for outlier detection (Roth, 2006), as we describe in Section 4, though our choice of loss function leads to a method which is comparable in terms of empirical performance on benchmark data but usually faster to train and test.

3. Least-squares probabilistic classification

We now give a brief review of least-squares probabilistic classification (Sugiyama, 2010). Given labelled training data of the form $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ where $\mathbf{x}_i \in \mathbb{R}^d$ is an input point in the data space, $y_i \in \mathcal{Y}$ is the corresponding class label and $\mathcal{Y} = \{1, \ldots, c\}$ is the set of possible classes, we wish to be able to estimate the class-conditional probability $p(y|\mathbf{x})$. It is possible to construct functions $q(y{=}i|\mathbf{x}, \boldsymbol{\theta}_i)$ to estimate $p(y{=}i|\mathbf{x})$ for each $i \in \mathcal{Y}$, using an approximation of the form

$$q(y{=}i|\mathbf{x}, \boldsymbol{\theta}_i) = \boldsymbol{\theta}_i^\top \boldsymbol{\phi}(\mathbf{x}) \,,$$

where

$$\boldsymbol{\theta}_i = (\theta_{i,1}, \ldots, \theta_{i,B})^\top \in \mathbb{R}^B$$

for some number of parameters B, and

$$\boldsymbol{\phi}(\mathbf{x}) = (K(\mathbf{x}, \mathbf{x}_1), \ldots, K(\mathbf{x}, \mathbf{x}_B))^\top \in \mathbb{R}^B$$

is a vector of kernel basis functions. We can set $B = N$ to have a kernel basis function at every training point, or for $B < N$ use some random subset of the training points. In this work we use the squared exponential kernel $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{\sigma^2}\|\mathbf{x} - \mathbf{x}'\|^2\right)$.

We fit this model using squared loss:

$$J_i(\boldsymbol{\theta}_i) = \frac{1}{2} \int \left(q(y{=}i|\mathbf{x}, \boldsymbol{\theta}_i) - p(y{=}i|\mathbf{x})\right)^2 p(\mathbf{x}) d\mathbf{x} \,.$$

Expanding and using $p(y|\mathbf{x}) = p(\mathbf{x}|y)p(y)/p(\mathbf{x})$ we obtain

$$\begin{aligned} J_i(\boldsymbol{\theta}_i) = & \frac{1}{2} \int q(y{=}i|\mathbf{x}, \boldsymbol{\theta}_i)^2 p(\mathbf{x}) d\mathbf{x} \\ & - \int q(y{=}i|\mathbf{x}, \boldsymbol{\theta}_i) p(\mathbf{x}|y{=}i) p(y{=}i) d\mathbf{x} + C \,. \end{aligned}$$

Empirically, we can approximate the expectations by sample averages, and the prior $p(y{=}i)$ by sample ratios.

Ignoring the constant C, factor $1/N$ and including an ℓ_2-regularizer, we have the following training criterion:

$$\widehat{J}_i(\boldsymbol{\theta}_i) = \frac{1}{2}\boldsymbol{\theta}_i^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi}\boldsymbol{\theta}_i - \boldsymbol{\theta}_i^\top \boldsymbol{\Phi}\mathbf{m}_i + \frac{\rho}{2}||\theta_i||^2 ,$$

where $\boldsymbol{\Phi} = (\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_N))^\top$ and $\mathbf{m}_i$ is a column vector indicating membership of class i such that the jth element is one if $y_j = i$ and zero otherwise. $\widehat{J}_i(\boldsymbol{\theta}_i)$ is minimized by

$$\widehat{\boldsymbol{\theta}}_i = \left(\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \rho \mathbf{I}_B\right)^{-1} \boldsymbol{\Phi}\boldsymbol{m}_i , \tag{1}$$

which is essentially kernel ridge regression. We select ρ and σ with cross validation. Because of the nature of the estimator, it is sometimes possible to obtain estimates of posteriors which are negative. We simply round up to zero in such cases,

$$q(y{=}i|\mathbf{x}, \widehat{\boldsymbol{\theta}}_i) = \max\left(0, \widehat{\boldsymbol{\theta}}_i^\top \boldsymbol{\phi}(\mathbf{x})\right) . \tag{2}$$

A posterior estimate is then obtained by normalizing over all classes,

$$\hat{p}(y{=}i|\mathbf{x}) = \frac{q(y{=}i|\mathbf{x}, \widehat{\boldsymbol{\theta}}_i)}{\sum_{j \in \mathcal{Y}} q(y{=}j|\mathbf{x}, \widehat{\boldsymbol{\theta}}_j)} .$$

This least-squares approach is a consistent estimator and is very fast to compute in practice, finding a global optimum in a single step with no iterative parameter search required. Consistency is guaranteed even in the case where estimates are rounded up to zero, as discussed in (Sugiyama, 2010, §2.2). This formulation is therefore an alternative to kernel logistic regression, providing similar theoretical guarantees and empirical accuracy, but with a speed increase of orders of magnitude (Sugiyama, 2010, §3).

4. Anomaly model

We now consider the case in which other classes $\{c+1, c+2, \ldots\}$ might be represented in the test data but not in the training data. We use $y{=}*$, $* \notin \mathcal{Y}$ to denote any such anomaly class. The supervised anomaly detection problem is to assign a value to the estimate $\hat{p}(y{=}*|\mathbf{x})$ for some test data $\mathbf{x}$ given training data only from classes in $\mathcal{Y}$. Although we do not have explicit training data, we are free to make assumptions about the possible distribution of such data relative to the "known" classes, yielding estimators consistent with those assumptions.

The method we propose is similar in essence to the one-class support vector machine (Schölkopf et al., 1999). These methods begin with the assumption that outliers occupy low-density regions of the data space and that a kernel model can be used to characterize the high-density regions given training data. Any given significance threshold can then be used to separate the inlier and outlier level sets.

With some abuse of notation, we estimate the conditional probability of an outlier $p(y{=}*|\mathbf{x}, \boldsymbol{\theta}_i)$ with

$$q(y{=}*|\mathbf{x}, \boldsymbol{\theta}_*) = 1 - \boldsymbol{\theta}_*^\top \boldsymbol{\phi}(\mathbf{x}) . \tag{3}$$

The problem of identifying outliers can then be equated with learning $\boldsymbol{\theta}_*$ such that Eq. (3) is close to zero when $\mathbf{x}$ is within a region in which training data has high density, and is close to one anywhere else. To achieve this we minimize the following loss function:

$$J_*(\boldsymbol{\theta}_*) = \frac{1}{2}\int \left(1 - \boldsymbol{\theta}_*^\top \boldsymbol{\phi}(\mathbf{x})\right)^2 p(\mathbf{x}) d\mathbf{x} + \frac{\rho}{2}||\boldsymbol{\theta}_*||^2 . \tag{4}$$

The integral term specifies the first part of the objective, that Eq. (3) should be close to zero for inlying regions. For $\mathbf{x}$ in highly outlying regions where $\boldsymbol{\phi}(\mathbf{x})$ approaches the origin, Eq. (3) approaches one for any choice of $\boldsymbol{\theta}_*$. However, the term $\frac{\rho}{2}||\boldsymbol{\theta}_*||^2$ rewards choices of $\boldsymbol{\theta}_*$ for which Eq. (3) approaches one in outlying regions more quickly. The objective function in this form is analogous to that in Schölkopf et al. (1999), which uses a support vector machine to separate training data from the origin with maximum margin.

Expanding Eq. (4) and approximating empirically for finite training data gives

$$\widehat{J}_*(\boldsymbol{\theta}_*) = \frac{1}{2} + \frac{1}{2}\boldsymbol{\theta}_*^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi}\boldsymbol{\theta}_* - \sum_{i=1}^{N} \boldsymbol{\theta}_*^\top \boldsymbol{\phi}(\mathbf{x}_i) + \frac{\rho}{2}||\boldsymbol{\theta}_*||^2 , \tag{5}$$

which is minimized by

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_* &= \left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \rho\mathbf{I}_B\right)^{-1}\sum_{i=1}^{N}\boldsymbol{\phi}(\mathbf{x}_i)\\ &= \left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \rho\mathbf{I}_B\right)^{-1}\sum_{j\in\mathcal{Y}}\boldsymbol{\Phi}^\top\boldsymbol{m}_j\\ &= \sum_{j\in\mathcal{Y}}\widehat{\boldsymbol{\theta}}_j\,.\end{aligned}$$

Therefore

$$\begin{aligned}q(y{=}*\,|\mathbf{x},\widehat{\boldsymbol{\theta}}_1,\ldots,\widehat{\boldsymbol{\theta}}_S) &= 1-\sum_{j\in\mathcal{Y}}\widehat{\boldsymbol{\theta}}_j^\top\boldsymbol{\phi}(\mathbf{x})\\ &= 1-\sum_{j\in\mathcal{Y}}q(y{=}j|\mathbf{x},\hat{\theta}_j)\,.\end{aligned}$$

We round up to zero as before in case of negative estimates,

$$q(y{=}*\,|\mathbf{x},\widehat{\boldsymbol{\theta}}_1,\ldots,\widehat{\boldsymbol{\theta}}_c) = \max\left(0, 1-\sum_{j\in\mathcal{Y}}q\left(y{=}i|\mathbf{x},\widehat{\boldsymbol{\theta}}_j\right)\right). \tag{6}$$

An intuition for this result is that since $q\,(y_t{=}i|\mathbf{x}_t)$ is an estimator of the conditional probability $p\,(y_t{=}i|\mathbf{x}_t)$, the sum of these terms for all $i \in \mathcal{S}$ can be interpreted as an approximation to $p(y_t \in \mathcal{S}|\mathbf{x}_t)$ the conditional probability of being in any one of the known states. The complement of the sum could therefore be viewed as an estimate of $p(y_t \notin \mathcal{Y}|\mathbf{x}_t)$, the conditional probability of an unknown state given the assumptions about classwise distribution of $\mathbf{x}$ encoded in the choice of kernel parameter σ and regularization parameter ρ.

In this method, the effect of increasing the size of parameter ρ is both to regularize (intuitively, to model the densities of known classes as more of a smooth "ball" around the most inlying training points) and to increase the sensitivity to outliers. Figure 1 plots the contours of $\hat{p}(y = *|\mathbf{x})$ for different settings of ρ in a one-class problem where inlying training data is drawn from a 2D Gaussian distribution to illustrate this. The choice of the kernel length scale parameter σ is equivalent to that in any kernel modeling problem, for which a number of heuristics have been found to be successful, such as using the median distance between pairs of training data points or the average distance between kth nearest neighbors in the training data.

In summary, the assumptions we make in this section lead us to a method for assigning probabilities to test data instances that they belong to an anomaly class which occupies a region of data space with low density in the training data. To assign such probabilities, we do not need to do any extra parameter estimation than was already carried out in the conventional supervised learning process described in Section 3. This anomaly detection method therefore shares the advantages of LSPC in terms of speed of training and inference and the flexibility of a nonparametric kernel-based method.

5. Sequential anomaly detection

Anomaly detection is often applied in a sequential setting, where we have temporal information to aid the inference process (Chandola et al., 2012; Quinn and Williams, 2007; Song et al., 2013; Tan et al., 2011). For example, inlier or outlier states might be expected to have long durations relative to the frequency of observations, such that if we knew there was an anomaly at time t, we may expect an anomaly a time $t + 1$ with high probability before having seen any data.

In this section, we therefore consider the estimation of a latent sequence of class labels $y_{1:T} = \{y_t \in \mathcal{Y} \cup *\}_{t=1}^{T}$ from a sequence of observed vectors $\mathbf{x}_{1:T} = \left\{\mathbf{x}_t \in \mathbb{R}^d\right\}_{t=1}^{T}$. We assume that each observation is independently drawn from the distribution $p(\mathbf{x}_t|y_t)$, and that y_t has first order Markovian dynamics such that $p(y_t|y_{1:t-1}) = p(y_t|y_{t-1})$. The values y_t can take on are as above in the static classification problem. Thus $\mathcal{Y}$ might contain a single inlier class and the corresponding inference problem is that of identifying anomalous subsequences in $\mathbf{x}_{1:T}$. There could also be multiple inlier classes, $\mathcal{Y} = \{1,\ldots,c\}$, in which case we may be interested in determining at each time frame t whether y_t is one of these known inlier classes (and if so, which one) or whether $y_t = *$.

The densities $p(y_t|y_{t-1})$ and $p(\mathbf{x}_t|y_t)$ and the initial class probabilities $p(y_1)$ together define a Hidden Markov Model (Rabiner, 1989). The HMM forward-backward algorithm can be used to estimate the probability of each state at each time frame, $p(y_t{=}i|\mathbf{x}_{1:T})$. We briefly review the algorithm here, using the following shorthand: $\mathbf{A}$ de-

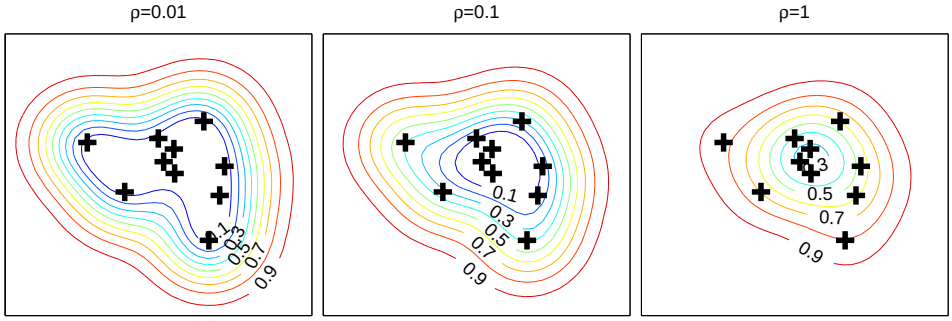


Figure 1: Contours of outlier probability density at different regularization levels, given 10 inlier samples from a 2D Gaussian distribution. Larger values of ρ increase the sensitivity to outliers.

notes a $(c+1)\times(c+1)$ matrix of class transition probabilities, such that $\mathbf{A}_{ij} = p(y_t{=}j|y_{t-1}{=}i)$ and $\pi \in \mathbb{R}^{c+1}$ is a vector of initial state probabilities such that $\pi_i = p(y_1{=}i)$. These parameters are straightforward to estimate if labeled training data is available (Rabiner, 1989), and domain knowledge can be used to choose suitable values for π_*, $\{A_{i*}\}_{i=1}^{c+1}$,$\{A_{*i}\}_{i=1}^{c}$ with respect to the anomaly class.

The first stage of the algorithm involves recursively calculating forward messages $\alpha_t(i)$ for each of the states $i \in \mathcal{Y} \cup *$:

$$\alpha_i(1) = \pi_i p(\mathbf{x}_1|y_1{=}i)\,, \tag{7}$$

$$\alpha_i(t) = \left[\sum_{j\in\mathcal{Y}\cup *} \alpha_j(t-1)\mathbf{A}_{ji}\right] p(\mathbf{x}_t|y_t{=}i)\,, \quad t = 2,\ldots,T. \tag{8}$$

The forward messages can be interpreted as the posterior probability of each class given observations up to that time frame, and the process of calculating them is known as *filtering*. For numerical stability or to interpret these quantities as probability measures it is necessary to normalize the messages $\alpha_1(t),\ldots,\alpha_{c+1}(t)$ so that they sum to 1.

To carry out *smoothing* (calculation of $p(y_t{=}i|\mathbf{x}_{1:T})$) the backwards messages $\beta_t(i)$ must first be similarly calculated:

$$\beta_i(T) = 1\,, \tag{9}$$

$$\beta_i(t) = \sum_{j\in\mathcal{Y}\cup *} \mathbf{A}_{ij} p(\mathbf{x}_{t+1}|y_{t+1}{=}j)\beta_j(t+1)\,, \quad t = T-1,\ldots,1. \tag{10}$$

The two types of messages are then combined to give the final result

$$\gamma_i(t) = p(y_t{=}i|\mathbf{x}_{1:T}) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j\in\mathcal{Y}\cup *}\alpha_j(t)\beta_j(t)}\,. \tag{11}$$

In order to carry out these inference steps in practice, we have to calculate the $p(\mathbf{x}_t|y_t{=}i)$ terms. To do this, we use the relationship $p(\mathbf{x}_t|y_t{=}i) \propto p(y_t{=}i|\mathbf{x}_t)/p(y_t{=}i)$. At test time, to compute Eqs. (7,8,10) we therefore substitute $q(y_t{=}i|\mathbf{x}_t)/\pi_{y_t}$ for $p(\mathbf{x}_t|y_t{=}i)$, calculating $q(y_t{=}i|\mathbf{x}_t)$ with Eq. (2) for inlier classes $i \in \mathcal{Y}$ and with Eq. (6) for the anomaly class $i = *$. This means that to estimate parameters we only have to calculate $\widehat{\boldsymbol{\theta}}_1,\ldots,\widehat{\boldsymbol{\theta}}_c$ from training data exactly as is done for the classifier in Section 3 (in addition to choosing $\mathbf{A},\pi$ as above).

We note that a mathematically equivalent way to view this inference is that the relationship $\frac{p(\mathbf{x}_t|y_t=i)}{p(\mathbf{x}_t|y_t=j)} = \frac{\pi_j p(y_t=i|\mathbf{x}_t)}{\pi_i p(y_t=j|\mathbf{x}_t)}$ can be used in order to apply the density ratio formulation of the HMM (Quinn and Sugiyama, 2013).

6. Experiments

We now give experimental results using the above methods when applied to several real-world datasets. First, we analyse the performance of the anomaly detection method for i.i.d. data compared to standard alternative methods, then we illustrate the sequential inference method on two times series monitoring applications.

6.1. Static anomaly detection

We compared anomaly detection performance of the least squares method on static data with standard anomaly

detection methods. These methods were the one-class support vector machine (Schölkopf et al., 1999), the distance to the kth nearest neighbor (Chandola et al., 2009, §5), and k-means clustering followed by finding the distance from test data to the nearest cluster centre (Chandola et al., 2009, §6). These methods were applied to 22 classification datasets made available on the libsvm website[1]. For each dataset, data points with the first class label were treated as inliers and data points with the second class label were treated as outliers. For multi-class datasets, any other classes were ignored. Anomaly detection was evaluated by carrying out a stratified 5-fold cross validation (i.e. keeping the proportion of inliers to outliers constant in each fold). For each fold, the anomaly detection methods were trained with only examples from the inlier class. On the test set, the anomaly scores were used to calculate AUC values.

For least squares anomaly detection, we set $\rho = 0.1$, $B = \min(N, 500)$, and σ to be the median distance between each data point and its kth nearest neighbour ($k = 7$) in a subset of the training data. For the one class support vector machine (OCSVM), we used the scikit-learn[2] implementation (based on libsvm) with its default values $\nu = 0.5$ and the same kernel width as for the least squares method. For k-NN we used $k = 10$, for k-means we used $k = 20$, and for both algorithms we again used the scikit-learn implementations.

The average AUC values are shown in Table 1. The least squares method had either the highest average AUC or performance not significantly different from the highest scoring method (calculated using a paired t-test with significance level 0.05) for 16 of the 22 datasets. Although the least squares method did not have significantly higher AUCs than the baseline methods across all datasets, it was much faster to run. For each method, the bottom row of Table 1 shows the total time required for running the experiments, relative to the time taken by our least squares method. All the baseline methods are significantly slower overall, and although the least squares was not the fastest in all cases (particularly for small datasets, in which case the k-NN algorithm has smaller overhead), the gain became larger with increasing dataset size. For example, when processing MNIST data, OCSVM was 8.1 times slower than least squares, k-NN was 4.1 times slower, and k-means was 23.3 times slower). Furthermore, the OCSVM used the highly optimised libsvm C code, whereas our least squares implementation ran in Python. The k-NN implementation was also heavily optimised, for example with heuristics to choose between KD-tree or ball-tree data structures as well as leaf size parameters based on the training data. We therefore conclude that the performance of our method on static data is competitive with alternative methods in terms of accuracy and superior in terms of speed for large datasets.

6.2. Sequential anomaly detection

We evaluated the sequential anomaly detection technique first using the NASA shuttle valve dataset (Ferrell and Santuro, 2005), which contains sequences of solenoid current readings from fuel control valves. Four sequences are measured under normal operating conditions, and eight sequences during which some fault was induced in the valve. Each sequence contains 1000 measurements taken at 1ms intervals.

We constructed a density ratio HMM with one state modeling normal dynamics and one anomaly state, setting $A = \begin{bmatrix} .999 & .001 \\ .1 & .9 \end{bmatrix}, \pi = [0.5, 0.5]$ and creating observation subsequences such that $\mathbf{x}_t = [x_t, x_{t+50}]$, where the scalar x_t is the measurement in the original dataset. The only preprocessing applied to the original data was a moving median filter of width 10 to remove transient spikes. The model was trained using single example sequences of normal dynamics. In each case, after training with one normal sequence, we applied the model to the remaining three normal sequences and eight abnormal sequences.

Sample output is shown in Figure 2(a). For each of the four test sets, we calculated the total probability mass of the abnormal state in each of sequences. In all cases, this probability measure perfectly segmented the normal and abnormal test sequences, i.e. the probability mass for the normal test sequences was always lower than that of the abnormal test sequences. For comparison, previous work (Chan and Mahoney, 2005) has shown other anomaly detection methods unable to perfectly separate normal and abnormal sequences even when training on three normal sequences, and perfect separation to be pos-

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

[2] http://scikit-learn.org

Table 1: Results of anomaly detection on static data (average AUC after stratified k-fold cross validation). LSAD denotes least squares anomaly detection, OCSVM denotes the one-class support vector machine, KNN denotes k-nearest neighbour, and KM denotes k-means. Where there is a unique maximal average AUC value, it is indicated in bold. Italics indicate results which were not significantly different from the best scoring method according to a paired t-test at significance level 0.05.

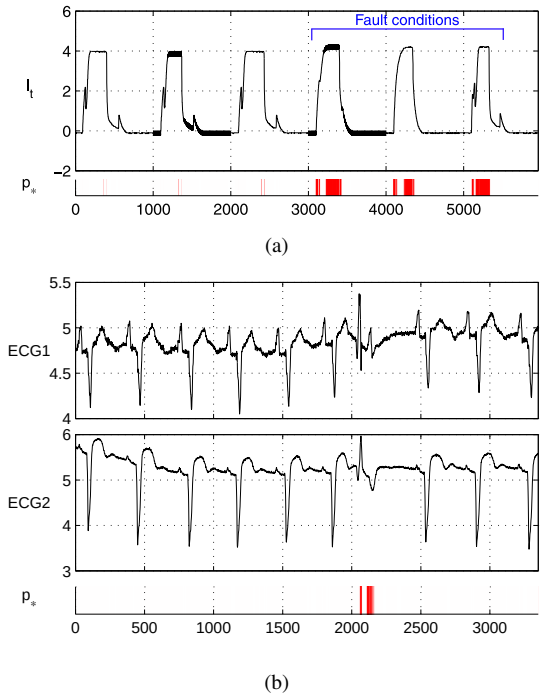
Dataset			Method			
	d	N	LSAD	OCSVM	KNN	KM
australian	14	690	*0.6911*	*0.6945*	**0.7027**	*0.6631*
breast-cancer	10	683	*0.9866*	*0.9867*	**0.9879**	0.9833
cod-rna	8	59535	0.8262	0.8408	**0.8488**	0.7702
colon-cancer	2000	62	*0.7000*	0.6987	*0.6450*	**0.7638**
diabetes	8	768	0.7042	0.7242	**0.7456**	*0.7246*
dna	180	1532	**0.7796**	0.7026	*0.7657*	*0.7637*
duke	7129	44	*0.5400*	*0.5400*	*0.5375*	*0.4525*
gisette	5000	7000	0.5199	0.4990	0.5324	**0.5385**
glass	9	146	**0.7961**	*0.7778*	0.6467	*0.7920*
heart	13	270	*0.6200*	0.6447	**0.7075**	0.6253
ijcnn1	22	49990	0.6947	**0.7300**	0.6172	0.6160
ionosphere	34	351	0.9621	**0.9658**	*0.9622*	*0.9547*
letter	16	1161	**0.9990**	*0.9988*	0.9921	0.9938
leukemia	7129	72	**0.7160**	0.6782	*0.6378*	*0.6747*
mnist	780	14780	*0.8348*	*0.8600*	0.8341	**0.8666**
mushrooms	112	8124	*0.9900*	*0.9908*	0.9881	**0.9924**
pendigits	16	1559	*0.9987*	*0.9989*	*0.9988*	**0.9992**
satimage	36	1551	*0.9999*	*0.9999*	*0.9998*	*0.9998*
sonar	60	208	*0.6746*	0.6524	0.6492	**0.7184**
usps	256	2199	0.9595	0.9618	0.9536	**0.9781**
vowel	10	180	**0.9963**	*0.9951*	0.6630	*0.9864*
wine	13	130	**0.9904**	*0.9892*	*0.9868*	*0.9868*
Time			1.00	4.21	1.98	13.80

sible when training on two normal sequences at a time using a 3-d feature vector and heuristic anomaly detection algorithm. Therefore the density ratio HMM not only requires less training data than previous methods to achieve perfect performance on this dataset, but requires very little feature engineering or parameter adjustment to do so.

We give another illustration of anomaly detection when applying our method to electrocardiogram (ECG) data from the MIT PhysioNet database (Goldberger et al., 2000). To demonstrate that the effectiveness of anomaly detection is not sensitive to parameter settings, we applied the same parameters A, π and subsequence construction $\mathbf{x}_t = [x_t, x_{t+50}]$ as we did to the shuttle data. The data was preprocessed by subtracting the results of a moving median filter of window size 300 on both channels in order to remove drift. Training the model with a sequence of length 2000 containing only normal heartbeats, we then applied it to the subsequent measurements in the recording containing an arrhythmia. The model correctly identifies the period of unusual dynamics, shown in Figure 2(b).

7. Discussion

In this letter we have introduced a novel method for anomaly detection, based on recent work in probabilistic classification using ℓ_2 loss. We have demonstrated that our method achieves practical anomaly detection performance comparable to currently popular methods, with significant improvements in training and testing time. Therefore we particularly recommend its application to



(a)

(b)

Figure 2: Anomaly detection examples. Panel (a) shows part of the shuttle dataset, with three normal sequences and three fault sequences concatenated. Here p_* denotes $p(y_t = * |\mathbf{x}_{1:T})$, the probability of the observation at time t being governed by an anomalous dynamical regime, where solid red indicates probability close to 1. These results were obtained after training on one separate normal sequence. Panel (b) shows an ECG recording, where the model identifies a cardiac arrhythmia.

the discovery of anomalies in large static and sequential datasets, or wherever CPU power available at test time is very limited. We have further shown that because the method is probabilistic, it is straightforward to incorporate into a hidden Markov model framework for sequential inference. This principle could be applied more generally, however, to other cases in which test data is subject to some kind of structural dependency. Specific examples of interest for future work are spatial and spatio-temporal modeling.

References

Ahmed, T., Coates, M., Lakhina, A., 2007. Multivariate online anomaly detection using kernel recursive least squares, in: Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM), pp. 625–633.

Chan, P., Mahoney, M., 2005. Modeling multiple time series for anomaly detection, in: Proceedings of the Fifth IEEE International Conference on Data Mining.

Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection: A survey. ACM Computing Surveys (CSUR) 41, 15.

Chandola, V., Banerjee, A., Kumar, V., 2012. Anomaly detection for discrete sequences: A survey. Knowledge and Data Engineering, IEEE Transactions on 24, 823–839.

Ferrell, B., Santuro, S., 2005. NASA Shuttle Valve Data. `http://www.cs.fit.edu/~pkc/nasa/data/`.

Filippone, M., Masulli, F., Rovetta, S., 2010. Applying the possibilistic c-means algorithm in kernel-induced spaces. Fuzzy Systems, IEEE Transactions on 18, 572–584.

Gao, J., Hu, W., Li, W., Zhang, Z., Wu, O., 2010. Local outlier detection based on kernel regression, in: Proceedings of the 20th International Conference on Pattern Recognition (ICPR), pp. 585–588.

Goldberger, A.L., Amaral, L.A.N., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.K., Stanley, H.E., 2000. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation 101, e215–e220.

Kemmler, M., Rodner, E., Denzler, J., 2010. One-class classification with Gaussian processes, in: Proceedings of the Ninth Asian Conference on Computer Vision (ACCV), pp. 489–500.

Quinn, J., Sugiyama, M., 2013. Density Ratio Hidden Markov Models. ArXiv 1302.3700.

Quinn, J., Williams, C., 2007. Known unknowns: Novelty detection in condition monitoring, in: Proceedings of the 3rd Iberian conference on Pattern Recognition and Image Analysis, pp. 1–6.

Rabiner, L., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77, 257–286.

Roth, V., 2006. Kernel fisher discriminants for outlier detection. Neural Computation 18, 942–960.

Schölkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J., 1999. SV estimation of a distribution's support, in: Advances in Neural Information Processing Systems.

Song, Y., Wen, Z., Lin, C.Y., Davis, R., 2013. One-class conditional random fields for sequential anomaly detection, in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence.

Sugiyama, M., 2010. Superfast-trainable multi-class probabilistic classifier by least-squares posterior fitting. IEICE Transactions on Information and Systems 93, 2690–2701.

Tan, S.C., Ting, K.M., Liu, T.F., 2011. Fast anomaly detection for streaming data, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, pp. 1511–1516.