Pattern Recognition Letters 40 (2014) 36–40

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

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

ARTICLE INFO

Article history:
Received 31 July 2013
Available online 6 January 2014

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

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.

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) [16]. 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 experimental 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 [3]. Different assumptions might be made about the distribution of anomalous points relative to the training, inlier points, which yield different

☆ This paper has been recommended for acceptance by D. Dembele.

* Corresponding author. Tel.: +256 773420920.

E-mail addresses: jquinn@cit.ac.ug (J.A. Quinn), sugi@cs.titech.ac.jp (M. Sugiyama).

0167-8655/$ - see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.12.016

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. [1], 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. [7], 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 [6]. Gaussian process models can also be used for kernel-based outlier detection [9].

Our work begins with similar assumptions about the nature of outliers as used in the one-class support vector machine [14] and the kernel Fisher discriminant method for outlier detection [13], 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 [16]. 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:

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

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

$$J_i(\theta_i) = \frac{1}{2}\int q(y = i|\mathbf{x}, \theta_i)^2 p(\mathbf{x})d\mathbf{x} - \int q(y = i|\mathbf{x}, \theta_i)p(\mathbf{x}|y = i)p(y = i)d\mathbf{x} + C.$$

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

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

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 [16, Section 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 [16, Section 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 [14]. 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 $p(y = *|\mathbf{x}, \theta_i)$ 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 [14], which uses a support vector machine to separate training data from the origin with maximum margin.