Problematic Projection to the In-Sample Subspace for a Kernelized Anomaly Detector

James Theiler and Guen Grosklos

Abstract—We examine the properties and performance of kernelized anomaly detectors, with an emphasis on the Mahalanobis-distance-based kernel RX (KRX) algorithm. Although the detector generally performs well for high-bandwidth Gaussian kernels, it exhibits problematic (in some cases, catastrophic) performance for distances that are large compared to the bandwidth. By comparing KRX to two other anomaly detectors, we can trace the problem to a projection in feature space, which arises when a pseudoinverse is used on the covariance matrix in that feature space. We show that a regularized variant of KRX overcomes this difficulty and achieves superior performance over a wide range of bandwidths.

Index Terms—Adaptive signal detection, algorithms, covariance matrices, data models, detectors, multidimensional signal processing, pattern recognition, remote sensing, singular value decomposition, spectral analysis.

I. INTRODUCTION

ANOMALY detection is the unsupervised identification of data samples (e.g., pixels in a hyperspectral image) that are unusual with respect to the rest of the data [1], [2]. For instance, if the data are modeled by a Gaussian distribution with mean \( \mu \) and covariance \( C \), then the Mahalanobis distance [3]

\[ A(r) = (r - \mu)^T C^{-1} (r - \mu) \]  

(1)

provides a simple measure of how anomalous the point \( r \) is; this measure monotonically increases for decreasing likelihood that \( r \) is drawn from the distribution. The use of Mahalanobis distance for multispectral and hyperspectral anomaly detection was popularized by Reed and Yu [4] and is commonly referred to as the RX algorithm. A kernelization of the Mahalanobis distance was proposed by Cremers et al. [5] and adopted for hyperspectral anomaly detection by Kwon and Nasrabadi [6]. In this approach, a data sample is mapped to a feature space, and Mahalanobis distance is computed in that feature space. Where RX effectively assumes that the data samples are drawn from an elliptically contoured distribution, kernel RX (KRX) can accommodate more convoluted contours.

In this letter, we identify a property of KRX—an implicit projection in feature space—that leads to diminished performance, particularly at small bandwidths, and we show that a simple regularization scheme alleviates the problem. Section II derives a family of four kernelized anomaly detectors, two of which employ a projection to the in-sample subspace and two of which do not. These derivations clarify the role of this projection in kernelized anomaly detectors. Section III deploys these anomaly detectors first on simple 1-D and 2-D problems (where the distinctions between the detectors are pronounced and conspicuous) and then on real hyperspectral image data. In Section IV, we briefly conclude.

II. KERNELIZED ANOMALY DETECTION

In this section, we derive KRX and a regularized variant, regularized KRX (KRX-reg), but before we do that, we derive two other kernelized anomaly detectors, one that is standard (kernel density estimation (KDE) [7], [8]) and one that is new [flattened KDE (KDE-flat)]. Although the KDE-flat detector is our own invention, we do not advocate its use in practice because its performance is poor. We introduce this detector to illuminate the problem caused by projection to the data-defined subspace of feature space. This is a problem that it shares with KRX.

A. Notation

The background data set is composed of \( N \) samples (pixels, usually, in hyperspectral applications), with each sample being a \( d \)-dimensional point: \( x_n \in \mathcal{X} \subseteq \mathbb{R}^d \) for \( 1 \leq n \leq N \). Our aim is to estimate a function \( \mathcal{A} : \mathcal{X} \rightarrow \mathbb{R} \) that characterizes the relative anomalousness of points in the data space \( \mathcal{X} \).

If the data are drawn from a probability density function \( p(x) \), then anomalies occur when \( p(x) \) is small, so what we seek from an anomaly detector \( \mathcal{A}(x) \) is some negative monotonic function of \( p(x) \).

For kernel-based methods, the data samples are mapped to a feature space \( \mathcal{F} \) by a function \( \Phi \), i.e., \( \Phi : \mathcal{X} \rightarrow \mathcal{F} \), and in particular, \( \Phi(r) \in \mathcal{F} \) is the mapping of the point \( r \in \mathcal{X} \) into feature space. This feature space has the property that dot products can be expressed as a scalar function of points in the original data space: \( k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R} \); more specifically

\[ k(r, s) = \Phi(r)^T \Phi(s) \in \mathbb{R}. \]  

(2)

The “kernel trick” is the recognition that, by specifying the kernel function \( k(r, s) \), one may not need to explicitly evaluate...
$\Phi(r)$ or $\Phi(s)$. A popular choice, and the one that we consider here, is the Gaussian radial basis function kernel

$$k(r,s) = \exp\left(-\|r-s\|^2/2\sigma^2\right)$$

where the parameter $\sigma$ is called the bandwidth.

It is useful to consider the centroid of the data set $X = \{x_1, \ldots, x_N\}$ in the feature space, $\mu_\Phi = (1/N) \sum_{n=1}^{N} \Phi(x_n)$, and in terms of this centroid, to define a centered feature map $\Phi_c(r) = \Phi(r) - \mu_\Phi$. This new feature map can then be used to define a centered kernel function

$$k_c(r,s) = \Phi_c(r)^T \Phi_c(s) = k(r,s) - \frac{2}{N} \sum_{n} k(r,x_n) + \frac{1}{N^2} \sum_{n,m} k(x_n,x_m).$$

\[ (4) \]

### B. Kernel Density Estimation

In traditional KDE [8] (also called Parzen windows [7]), a probability density $\hat{p}(r)$ is estimated in terms of the data set $X = \{x_1, \ldots, x_N\}$

$$\hat{p}(r) = \frac{1}{N} \sum_{n} c_0^{-1} \exp\left(-\|r-x_n\|^2/2\sigma^2\right)$$

where $c_0 = (2\pi\sigma^2)^{d/2}$ is a constant that normalizes the density function. Note that $\hat{p}(r)$ is a fixed kernel density estimator; there is a large family of variable kernel density estimators for which the kernel itself is different for different points $r$, e.g., [8]–[11].

Following an appendix in [5], we derive the KDE detector as a Euclidean distance in feature space:

$$A_{\text{KDE}}(r) = \Phi_c(r)^T \Phi_c(r) = k_c(r,r)$$

where $\Phi_c(r)$ is the centered feature map and $k_c$ is the centered kernel function. From (4), we can write the KDE detector

$$A_{\text{KDE}}(r) = k(r,r) - \frac{2}{N} \sum_{n} k(r,x_n) + \frac{1}{N^2} \sum_{n,m} k(x_n,x_m).$$

The third term is a constant, and for a radial-basis kernel, $k(r,r)$ is also a constant, so

$$A_{\text{KDE}}(r) = c_1 - \frac{2}{N} \sum_{n} k(r,x_n) = c_1 - 2c_0 \hat{p}(r)$$

with $c_0$ and $c_1$ being constants. Thus, the KDE anomaly detector is a negative monotonic function of the probability density $\hat{p}(r)$ that was estimated using KDE.

\[ (8) \]

### C. KDE-Flat

In this section, we derive a variant of KDE that includes a “flattening” of the input, i.e., a projection to the subspace of $\mathcal{F}$ spanned by the training data. As we will see in Section III, the effect of this projection is significant.

To begin, define the data matrix in centered feature space

$$X_\Phi = [\Phi_c(x_1) \cdots \Phi_c(x_N)].$$

Let $r$ be the rank of this matrix (observe that $r \leq N - 1$ since the centroid has been subtracted). Express $X_\Phi$ with a singular value decomposition

$$X_\Phi = V_\Phi \Lambda_\Phi^{1/2} W^T.$$ 

Here, $V_\Phi$ is an orthogonal matrix with $r$ columns (so $V_\Phi^T V_\Phi = I$), $\Lambda$ is a diagonal $r \times r$ matrix with positive entries, and $W$ is an orthogonal $N \times r$ matrix (for which $W^T W = I$).

Note that the columns of $V_\Phi$ are eigenvectors of the covariance matrix $C_\Phi = X_\Phi^T X_\Phi$, and the columns of $W$ are eigenvectors of the centered Gram matrix

$$K_c = X_\Phi^T X_\Phi = \begin{bmatrix} k_c(x_1,x_1) & \cdots & k_c(x_1,x_N) \\ \vdots & \cdots & \vdots \\ k_c(x_N,x_1) & \cdots & k_c(x_N,x_N) \end{bmatrix}. $$

Also, the diagonal elements of $\Lambda$ are the positive eigenvalues of both $C_\Phi$ and $K_c$.

Note further that $V_\Phi$ projects (flattens) vectors in the feature space into an $r$-dimensional subspace. We can thus modify our KDE anomaly detector by computing the Euclidean distance in this subspace (see [12, eq. (1)])

$$A_{\text{flat}}(r) = \Phi_c(r)^T V_\Phi V_\Phi^T \Phi_c(r).$$

From (10), we can write $V_\Phi = X_\Phi W \Lambda^{-1/2}$, so

$$A_{\text{flat}}(r) = \Phi_c(r)^T X_\Phi W \Lambda^{-1} W^T X_\Phi^T \Phi_c(r) = Z_c(r)^T K_c^{-1} Z_c(r)$$

where $Z_c(r)$ can be expressed in terms of the centered kernel

$$Z_c(r) = X_\Phi^T \Phi_c(r) = \begin{bmatrix} k_c(x_1,r) \\ \vdots \\ k_c(x_N,r) \end{bmatrix}. $$

\[ (14) \]

### D. Kernel RX

The KRX idea is to use a Mahalanobis distance instead of a Euclidean distance in the feature space. That is

$$A_{\text{KRX}}(r) = \Phi_c(r)^T C_\Phi^{-1} \Phi_c(r)$$

where the covariance matrix $C_\Phi$ is determined from the data in feature space

$$C_\Phi = \sum_{n} \Phi_c(r)^T \Phi_c(r) = X_\Phi X_\Phi^T = V_\Phi \Lambda V_\Phi^T$$

where $X_\Phi$ was defined in (9) and decomposed in (10). The problem with KRX, as it is expressed in (15), is that $C_\Phi$ is not invertible. Although not explicitly discussed in [6], the approach taken in [6] uses the pseudoinverse. That is

$$C_\Phi^{-1} = (V_\Phi \Lambda V_\Phi^T)^{-1} = V_\Phi \Lambda^{-1/2} V_\Phi^T. $$

The ambiguous left-hand side is simply replaced with the well-defined right-hand side. We can use $V_\Phi = X_\Phi W \Lambda^{-1/2}$, obtained from (10), to further simplify

$$C_\Phi^{-1} = (X_\Phi W \Lambda^{-1/2})^T \Lambda^{-1/2} W^T X_\Phi^T = X_\Phi W \Lambda^{-2} W^T X_\Phi^T = X_\Phi K_c^{-2} X_\Phi^T$$

\[ (18) \]
where $K_c$ is the centered Gram matrix defined in (11) and $K_c^{-2}$ refers to the pseudoinverse of $K_c^2$. Thus
\[
A_{KRX}(r) = \Phi_c(r)^T X_c K_c^{-2} X_c^T \Phi_c(r) = Z_c(r)^T K_c^{-2} Z_c(r)
\]
where $Z_c(r) = X_c^T \Phi_c(r)$ was defined in (14).

It is important to recognize that the pseudoinverse involves the projection of $\Phi_c(r)$ to $V_c^T \Phi_c(r)$. Therefore, it is more appropriate to think of KRX as the Mahalanobis variant not of KDE, but of KDE-flat.

\[e. krx-reg\]

To deal with the singular matrix $C_\phi$ in (16), KRX employs the pseudoinverse. A common alternative approach for inverting singular and near-singular matrices is to regularize them first. Thus, in place of the sample covariance defined in (16), we can employ a regularization operation
\[
C'_{\phi} = C_{\phi} + \lambda I
\]
for some small $\lambda$. Since $C_{\phi}$ is not of full rank, it is useful to employ singular value decomposition and write [see (16)] $C_{\phi} = V_{\phi} \Lambda V_{\phi}^T$. Thus
\[
C'_{\phi} = V_{\phi} (\Lambda + \lambda) V_{\phi}^T + \lambda (I - V_{\phi} V_{\phi}^T).
\]
Since the two terms in the sum are orthogonal to each other, we can invert the sum by inverting the individual components
\[
C'_{\phi}^{-1} = V_{\phi} (\Lambda + \lambda)^{-1} V_{\phi}^T + \lambda^{-1} (I - V_{\phi} V_{\phi}^T)
\]
and the anomalousness is given by the Mahalanobis distance with respect to this regularized covariance matrix
\[
A_{\text{reg}}(r) = \Phi_c(r)^T C'_{\phi}^{-1} \Phi_c(r)
\]
\[
= \Phi_c(r)^T V_{\phi} (\Lambda + \lambda)^{-1} V_{\phi}^T \Phi_c(r)
\]
\[
+ \lambda^{-1} \Phi_c(r)^T (I - V_{\phi} V_{\phi}^T) \Phi_c(r)
\]
\[
= A_{KRX}(r) + \lambda^{-1} [A_{KDE}(r) - A_{\text{flat}}(r)]
\]
where $A'_{KRX}$ is computed using
\[
A'_{KRX}(r) = Z_c(r) K_c^{-1/2} (K_c + \lambda I)^{-1} K_c^{-1/2} Z_c(r)
\]
and $K_c^{-1/2}$ is the matrix square root of the pseudoinverse of the Gram matrix. As a practical matter, we remark that $A'_{KRX}$ and $A_{KRX}$ behave nearly identically. However, $A_{\text{reg}}$ is considerably different from $A_{KRX}$ (as is indicated by the second term in (23), which, with the $\lambda^2$ prefactor, is large). We note that KRX-reg does not simply regularize the covariance matrix $K_c$ in the KRX formula, given in (19); the regularization is of $C_{\phi}$ and takes place in the feature space.

For the experiments reported here, we used the small but numerically relevant value $\lambda = 10^{-8} \max_i \Lambda_{ii}$. We note that, in the $\lambda \to 0$ limit, (23) is dominated by the regularization term; this term, $\Phi_c(r)^T (I - V_{\phi} V_{\phi}^T) \Phi_c(r)$, actually provides an anomaly detector in its own right [12], [13].

III. COMPARISON OF ALGORITHMS

To compare the kernel-based anomaly detectors derived in the previous section, we apply them both to artificial 1-D and 2-D examples and to real hyperspectral data.

A. One-Dimensional Example

We begin with a very simple 1-D example; $N$ points are drawn from a standard (zero mean and unit variance) normal distribution. For this distribution, we expect anomalousness to be minimal at position $r = 0$ (the peak of the probability distribution) and to increase monotonically with distance from zero. As seen in Fig. 1, this behavior is indeed observed with the KDE detector, but we see problems with KDE-flat and KRX, the two anomaly detectors that employ a projection to the in-sample subspace. Anomalousness does initially increase with increasing distance from zero for these detectors, but then, it reverses itself and gets smaller. One can try to address this problem by using a larger bandwidth $\sigma$, but however large the bandwidth is chosen to be [e.g., $\sigma = 5$ in Fig. 1(c)], there is a distance beyond which anomalousness decreases with increasing distance. This problem with KRX is fixed by KRX-reg.
Fig. 2. In these 2-D examples, $N = 50$ training points are drawn from a distribution. In the top row, this distribution is a mixture of two Gaussian distributions, one of unit variance at position $[3,3]$ and the other of smaller variance ($1/16$) at position $[-1,-1]$. In the bottom row, training points are drawn from points along a sine wave. Anomaly detectors are derived in terms of the training data, using bandwidth $\sigma = 5$ [above: (a), (b), (c), and (d)] and $\sigma = 0.5$ [below: (e), (f), (g), and (h)]. Anomalousness is scaled to range from zero to one and plotted as white to black on a square that ranges from $[-B,B]$ on both axes. $B = 15$ above, and $B = 1$ below. Contours indicate false alarm rates of (black) 0.05, (magenta) 0.01, and (cyan) 0.001. (a) KDE. (b) KDE-flat. (c) KRX. (d) KRX-reg. (e) KDE. (f) KDE-flat. (g) KRX. (h) KRX-reg.

Fig. 3. Volume enclosed by anomaly detection contour associated with a false alarm rate of 0.001, plotted against bandwidth values. (a) Multimodal Gaussian samples shown in the top row [(a), (b), (c), and (d)] of Fig. 2. (b) Sine wave samples shown in the bottom row [(e), (f), (g), and (h)] of Fig. 2. Smaller volumes are better.

B. Two-Dimensional Examples

With the 2-D examples illustrated in Fig. 2, we can see how the more adaptive KRX is able to more compactly enclose the data and, at the same time, how the anomalousness decreases for distant outliers. This nonmonotonicity with distance is even more pronounced for KDE-flat, but the phenomenon is present in both KDE-flat and KRX.

While Fig. 2(a)–(d) uses bandwidth $\sigma = 5$, the corresponding Fig. 3(a) illustrates how performance (as measured by volume enclosed by a contour) varies with $\sigma$. It is clear that KDE works best at small values of $\sigma$, but KDE-flat and KRX are unusable at small values of $\sigma$. As $\sigma$ increases, the KDE contour becomes increasingly smooth and less able to effectively enclose the nonanomalous data. On the other hand, a larger $\sigma$ generally improves the performance of KRX.

Fig. 4. Cooke City HyMap hyperspectral data set [14] is $280 \times 800$ pixels, with 126 spectral channels. The kernelized anomaly detectors were trained with $N = 1500$ randomly chosen pixels, using $\sigma = 5000$. (a) First principal component, and (b)–(f) are the anomalousness maps for the different algorithms: (b) KDE, (c) KDE-flat, (d) RX, (e) KRX, and (f) KRX-reg. The most anomalous 0.1% of the pixels are shown in black, the top 1% are pale magenta, and the top 10% are an even paler cyan. The analysis was applied to the full data set, but what are shown here are $100 \times 175$ pixel insets.

This is also observed in Figs. 2(e)–(h) and 3(b), where KRX-reg strictly outperforms KDE-flat, KDE, and KRX over a wide range of bandwidth values $\sigma$. 
used Cooke City data set [14]. For numerical reasons, we
algorithms (along with the nonkernelized RX) to the widely
of the false alarm rate (i.e., the fraction of pixels in the image that would be declared anomalous at that threshold). This volume is estimated by uniformly
filling a large ellipsoid (covering all the data and a generous margin beyond that as well) with 20,000 random points and computing the fraction of them whose anomalousness is within the given threshold. (a) $\sigma = 500$; (b) $\sigma = 5000$; (c) $\sigma = 50,000$.

C. Hyperspectral Imagery

These projection effects are also evident in the application to hyperspectral imagery. In this section, we apply the kernelized algorithms (along with the nonkernelized RX) to the widely used Cooke City data set [14]. For numerical reasons, we employed a slight variant of the Gaussian kernel: $k(r, s; \sigma) = k(r, s; \sigma) + \epsilon^2 k(r, s, \sigma')$ with $k$ defined in (3) and $\epsilon = 0.1$. This puts a little extra weight on the tails but preserves the important properties of a kernel function [15].

Fig. 4 illustrates the differences that are observed when different anomaly detectors are applied to the same image. Comparing KRX to KRX-reg (and even more so, KDE-flat to KDE), we see that many of the most anomalous points lose their anomalousness in algorithms that employ a projection to the in-sample data subspace.

Of course, whether a given pixel is truly anomalous is a judgment call [16], so we also employed a more objective volume-based comparison of the different algorithms, as has been advocated previously [17], [18]. Because the volume of irregular contours is difficult to measure in high dimensions, we do the comparison using the first three principal components of the data set. In Fig. 5, three values of sigma are used, and as was also observed in Fig. 3, we find that KDE prefers small $\sigma$ while KRX prefers larger $\sigma$. When $\sigma$ is too small, the KRX volume diverges at a low false alarm rate. However, for adequately large $\sigma$, KRX outperforms RX and KDE. For all values of $\sigma$, however, the KRX-reg is observed to perform well.

IV. CONCLUSION

Because of its projection to the in-sample subspace, KRX exhibits spuriously low anomalousness for points far from the training data. This same problem is observed in a simpler context (and more dramatic fashion) by comparing KDE-flat to KDE. Using ridge regularization (instead of pseudoinverse) on the covariance matrix in the feature space avoids this projection of the data. The result is KRX-reg, a kernelized anomaly detector that is as good or better than KRX and KDE over a wide range of bandwidth values.

REFERENCES