Unsupervised Ensemble-Kernel Principal Component Analysis for Hyperspectral Anomaly Detection

Nicholas Merrill
U.S. Naval Research Laboratory, Naval Research Enterprise Internship Program
nmerrill@vt.edu

Colin C. Olson
U.S. Naval Research Laboratory
colin.olson@nrl.navy.mil

Abstract

Unsupervised anomaly detection–which aims to identify outliers in data sets without the use of labeled training data–is critically important across a variety of domains including medicine, security, defense, finance, and imaging. In particular, detection of anomalous pixels within hyperspectral images is used for purposes ranging from the detection of military targets to the location of invasive plant species. Kernel methods have frequently been employed for this unsupervised learning task but are limited by their sensitivity to parameter choices and the absence of a validation step. Here, we use reconstruction error in the kernel Principal Component Analysis (kPCA) feature space as a metric for anomaly detection and propose, via batch gradient descent minimization of a novel loss function, to automate the selection of the Gaussian RBF kernel parameter, σ. In addition, we leverage an ensemble of learned models to reduce computational cost and improve detection performance. We describe how to select the model ensemble and show that our method yields better detection accuracy relative to competing algorithms on a pair of data sets.

1. Introduction

Typical (three-color) cameras lack the spectral sensitivity required for accurate material detection and/or identification in remote sensing applications. Instead, imagers that capture multispectral and hyperspectral imagery (MSI/HSI) seek to generate finer samplings of the spectrum such that useful information about observed materials is not averaged out by the sampling process.

Given such data, the goal of a spectral anomaly detection method is to detect unusual samples within a data set dominated by the presence of ordinary background pixels. Anomalies are by definition rare and are often generated by different underlying processes [1, 2] and, as such, unsupervised techniques are used to detect such anomalies in the absence of labeled training data. Numerous algorithms have been devised for this goal, the results of which have been applied to a variety of fields to improve upon domain-specific, rule-based detection methods. Unsupervised anomaly detection has applications in medicine, fraud detection, fault detection, and remote sensing, among others [1, 3, 4] and a recent, comprehensive review of the most successful of these techniques and their applications can be found in Goldstein and Uchidal [1].

We apply our algorithms here to improve the detection of anomalies in hyperspectral images. In particular, we use reconstruction error in the kernel Principal Component Analysis (kPCA) feature space as a metric for anomaly detection and demonstrate the use of a batch gradient descent minimization of a novel loss function to automate the selection of the Gaussian RBF kernel parameter, σ. In addition, we describe how a data subsampling process reduces the computational cost of developing a background model and exploits the assumed rarity of anomalous pixels to ensure the model truly represents background data.

2. Background

The hyperspectral anomaly detection problem assumes there is a rare pixel class within the hyperspectral image. More concretely, given N data points, $\mathbf{x}_i \in \mathbb{R}^D$, comprising the data set, $\mathbf{X} \in \mathbb{R}^{D\times N}$, the goal is to determine if a test pixel, $\mathbf{x}'$, drawn from $\mathbf{X}$ is anomalous. Due to the rarity of anomalies, a common approach is to model background pixels and identify pixels which are poorly fit by the model as anomalous. Essentially, the model fits abundant examples and those that are poorly fit are interesting.

In the context of unsupervised learning, there is no dis-

tinction between training and test data which makes the total number of parameters and their sensitivity crucial considerations in model selection. Data-driven or theory-motivated heuristics are often used for selecting parameters [1, 5] but do not always produce satisfactory results. While techniques to approximate accuracy without labels exist, they are limited and computationally expensive [6]. Ultimately, there is no guaranteed means for verifying that a specific parameter selection is an optimal or even reasonable choice and therefore methods that are insensitive to parameter selection are desirable.

Statistical methods rely on the accuracy of assumptions about the data distribution and seek to estimate the parameters of that assumed model. For example, in the remote sensing literature, the global *RX* detector (GRX) introduced by Reed and X. Yu [7] seeks to model the background using a multi-variate Gaussian with mean, μ, and covariance, Σ, calculated from all image pixels. Under this assumption the Mahalanobis distance, $D_{RX}(\mathbf{x}') = (\mathbf{x}' - \mu)\Sigma^{-1}(\mathbf{x}' - \mu)^T$, of a test pixel, $\mathbf{x}'$, is used as a detection statistic. Pixels with values of D_{RX} that fall below a threshold, γ, are background while pixels with larger values are anomalous.

Data-driven methods often use distance measures (nearest-neighbor), clustering techniques, or statistical measures independently or in some combination. These also suffer from various limitations including parameter selection sensitivity and reliance on assumed, but often inaccurate, statistical models [1]. Nevertheless, kernel-based methods are a promising family of techniques motivated by the idea that a better data model may be formed in a transformed, non-linear, feature space, $\mathcal{F}$, where a kernel function, $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$, allows the efficient computation of inner products between each datum $\boldsymbol{\Phi}(\boldsymbol{x}_i)$ in $\mathcal{F}$ without explicit calculation of the mapping $\mathbf{x}_i \rightarrow \boldsymbol{\Phi}(\boldsymbol{x}_i)$ from the ambient space to the feature space. The most popular amongst these methods for anomaly detection is the One-Class Support Vector Machine (OC-SVM), which separates the data from the origin in $\mathcal{F}$ [8].

Hoffmann [4] demonstrated, however, that the independent treatment of samples within the OC-SVM framework yields boundaries that do not to tightly model the data. Hoffmann used kernel PCA (kPCA) [9] to better model the preponderance of data and formulated an anomaly score as the reconstruction error between a given datum and its representation by a learned kPCA model in the feature space. This reconstruction error demonstrates better generalization, accuracy, and robustness over linear PCA, the Parzen density estimator, and OC-SVMs on a number of real-world and toy data sets.

Despite demonstrated success on anomaly detection problems, kernel methods require calculation of a distance (adjacency) matrix comprised of all pairwise similarity measures between each of the N data points comprising the data set. The cost of this calculation scales poorly, however, with increasing D and N which can be limiting for many applications. Uniform sub-sampling of the data in order to reduce the cost of calculating the adjacency matrix has been proposed for both classification [10] and anomaly detection [11] where out-of-sample data [12, 13] are represented in the model space learned from the sampled subset using techniques such as the Nyström extension [14, 15].

Even with manageable computational cost, however, all kernel-based methods are inherently sensitive to parameter selection. For the Gaussian RBF kernel,

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/(2\sigma^2)), \tag{1}$$

this parameter is the kernel bandwidth, σ. Although heuristics exist for selecting an appropriate σ, they often fail to generalize [5, 16, 1]. Here, we leverage Hoffmann's kPCA-based reconstruction error and a subsampling process to introduce Unsupervised Ensemble Kernel Principal Component Analysis (UE-kPCA)–a stable scheme for automating selection of the kernel bandwidth via batch gradient descent minimization of a custom loss function.

3. Anomaly Detection Using kPCA

To begin, we first review how kPCA is used to perform PCA in the feature space and how the reconstruction error is calculated from linear combinations of kernel functions operating on pairwise distances among ambient data points.

3.1. Kernel PCA

The kPCA algorithm is designed to calculate the non-linear mapping $\mathbf{x}_i \rightarrow \boldsymbol{\Phi}(\boldsymbol{x}_i)$ of a datum in $\mathbf{X}$ from the original D-dimensional (ambient) space into the theoretically infinite-dimensional (for an RBF kernel with infinite support) feature space $\mathcal{F}$ (although in practice the maximum dimensionality is N). After this non-linear mapping, the data in $\mathcal{F}$ are centered via the transformation

$$\tilde{\boldsymbol{\Phi}}(\boldsymbol{x}_i) = \boldsymbol{\Phi}(\boldsymbol{x}_i) - \boldsymbol{\Phi}_0, \tag{2}$$

where

$$\boldsymbol{\Phi}_0 = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\Phi}(\boldsymbol{x_n}). \tag{3}$$

is the mean of the data distribution in $\mathcal{F}$. Linear PCA is then performed on the centered data to find the M-dimensional subspace $M \leq N$ associated with the M principal components representing the greatest variance of the data in $\mathcal{F}$.

The principal components of $\mathbf{X}$ in $\mathcal{F}$ are the eigenvectors corresponding to the largest eigenvalues of the covariance matrix formed in $\mathcal{F}$. More directly, we are interested in the eigenvectors $\mathbf{V} = \{V^1, V^2, \ldots, V^M\}$ and corresponding

eigenvalues $v_1 \geq v_2 \geq \ldots \geq v_M$ of

$$\tilde{\boldsymbol{\Sigma}}_{\mathcal{F}} = \frac{1}{N}\sum_{i=1}^{N} \tilde{\boldsymbol{\Phi}}(\boldsymbol{x}_i)\tilde{\boldsymbol{\Phi}}(\boldsymbol{x}_i)^T. \tag{4}$$

But the covariance matrix, $\tilde{\boldsymbol{\Sigma}}_{\mathcal{F}}$, and therefore the principal components, $\boldsymbol{V}$, cannot be explicitly computed, as $\boldsymbol{\Phi}(\boldsymbol{x}_i)$ is never available. Instead of explicitly finding $\boldsymbol{V}$, the projections of $\boldsymbol{\Phi}(\boldsymbol{x}_i)$ onto $\boldsymbol{V}$ are found. Because $\boldsymbol{V^k}$ is one eigenvector of $\tilde{\boldsymbol{\Sigma}}_{\mathcal{F}}$ it can be expressed as a linear combination of points $\boldsymbol{\Phi}(\boldsymbol{x}_i)$,

$$\boldsymbol{V^k} = \sum_{i=1}^{N} \boldsymbol{\alpha}_i^k \tilde{\boldsymbol{\Phi}}(\boldsymbol{x}_i), \tag{5}$$

where each of the α_i^k is a component of a vector $\boldsymbol{\alpha}^k$, which is an eigenvector of the $N \times N$ kernel adjacency matrix $\tilde{K}_{ij} = \tilde{\boldsymbol{\Phi}}(\boldsymbol{x}_i) \cdot \tilde{\boldsymbol{\Phi}}(\boldsymbol{x}_j)$. Using the kernel trick, this matrix may in turn be expressed solely as a function of ambient data,

$$\tilde{K}_{ij} = K_{ij} - \frac{1}{N}\sum_{q=1}^{N} K_{iq} - \frac{1}{N}\sum_{p=1}^{N} K_{pj} + \frac{1}{N^2}\sum_{p,q=1}^{N} K_{pq}, \tag{6}$$

where $K_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The eigenvectors $\boldsymbol{\alpha}^k$ and corresponding eigenvalues λ_k are then found by the eigendecomposition of $\tilde{K}_{ij}$ ultimately yielding N eigenvectors $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \ldots, \boldsymbol{\alpha}^N\}$. A scaling of each $\boldsymbol{\alpha}^k$ is performed so that each $\boldsymbol{V^k}$ has unit length, $||\boldsymbol{\alpha}^k||^2 = 1/\lambda_k$. Additional details may be found in Hoffmann [4].

3.2. Anomaly Score

The anomaly score for a point $\boldsymbol{x}'$ is found by determining the reconstruction error in $\mathcal{F}$. Conceptually, the reconstruction error is

$$d_E(\boldsymbol{x}') = \tilde{\boldsymbol{\Phi}}(\boldsymbol{x}') \cdot \tilde{\boldsymbol{\Phi}}(\boldsymbol{x}') - W\tilde{\boldsymbol{\Phi}}(\boldsymbol{x}') \cdot W\tilde{\boldsymbol{\Phi}}(\boldsymbol{x}'), \tag{7}$$

where W contains M rows of principal components $\boldsymbol{V^k}$ corresponding to the M largest eigenvalues. The first term is the spherical potential of $\boldsymbol{x}'$ found by taking the scalar product

$$d_p(\boldsymbol{x}') = \tilde{\boldsymbol{\Phi}}(\boldsymbol{x}') \cdot \tilde{\boldsymbol{\Phi}}(\boldsymbol{x}'), \tag{8}$$

which is simply the squared distance of $\tilde{\boldsymbol{\Phi}}(\boldsymbol{x}')$ from the data mean $\boldsymbol{\Phi}_0$ in $\mathcal{F}$. Again we avoid working with $\tilde{\boldsymbol{\Phi}}(\boldsymbol{x}')$ directly and by substituting (3) into (8) and applying the kernel trick we obtain

$$d_p(\boldsymbol{x}') = \kappa(\boldsymbol{x}', \boldsymbol{x}') - \frac{2}{N}\sum_{i=1}^{N} \kappa(\boldsymbol{x}', \boldsymbol{x}_i) + \frac{1}{N^2}\sum_{i,j=1}^{N} \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{9}$$

Next we define $f_k(\boldsymbol{x}')$, the projection of $\boldsymbol{x}'$ in $\mathcal{F}$ onto $\boldsymbol{V}^k$, as $f_k(\boldsymbol{x}') = (\tilde{\boldsymbol{\Phi}}(\boldsymbol{x}') \cdot \boldsymbol{V}^k)$. This projection can be written as a function of only the ambient data by applying (5) and the kernel trick,

$$f_k(\boldsymbol{x}') = \sum_{i=1}^{N} \alpha_i^k [\kappa(\boldsymbol{x}', \boldsymbol{x}_i) - \frac{1}{N}\sum_{q=1}^{N} \kappa(\boldsymbol{x}_i, \boldsymbol{x}_q) - \frac{1}{N}\sum_{q=1}^{N} \kappa(\boldsymbol{x}', \boldsymbol{x}_q) + \frac{1}{N^2}\sum_{p,q=1}^{N} \kappa(\boldsymbol{x}_p, \boldsymbol{x}_q)]. \tag{10}$$

Finally, the reconstruction error-based anomaly can be directly computed by:

$$d_E(\boldsymbol{x}') = d_p(\boldsymbol{x}') - \sum_{k=1}^{M} f_k(\boldsymbol{x}')^2, \tag{11}$$

which is the reconstruction error between $\tilde{\boldsymbol{\Phi}}(\boldsymbol{x}')$, the centered projection of $\boldsymbol{x}'$ into $\mathcal{F}$, and its representation in $\mathcal{F}$ as a projection onto the largest M principal components of the PCA model learned from the data. If $M = N$ then $d_E(\boldsymbol{x})$ will be zero for all $\boldsymbol{x}$ because the representation of $\tilde{\boldsymbol{\Phi}}(\boldsymbol{x})$ in PCA coordinates is identically $\tilde{\boldsymbol{\Phi}}(\boldsymbol{x})$. When $M < N$ then $d_E(\boldsymbol{x})$ will remain smaller for non-anomalous points because the learned PCA model better represents the background and the error associated with dropping low-eigenvalue eigenvectors will remain smaller as M decreases.

4. Unsupervised Ensemble Kernel Principal Component Analysis

In practice, it is difficult to implement kPCA for unsupervised anomaly detection for two main reasons: 1) The sensitivity to the parameter settings of the Gaussian kernel 2) the cubic time complexity of the eigendecomposition of the kernel matrix, $\tilde{K}$ (6). We outline these issues and then describe our proposed solutions in 4.1 and 4.2.

First, it is worth exploring parameter limits and general properties of the kernel matrix as a function of the bandwidth. To begin, a Gaussian kernel matrix will always have a diagonal containing all ones, as the diagonal represents the self distance term, e.i. $||\boldsymbol{x}_i - \boldsymbol{x}_i|| = 0$. As σ approaches an arbitrarily large value, the argument of the kernel for any value of $\boldsymbol{x}$‘ approaches 0 as the argument of the exponent in (1) approaches negative infinity. Explicitly,

$$\lim_{\sigma \to 0} \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0 \tag{12}$$

In this case $\tilde{K}$ approaches the identity matrix. This indicates that all data vectors in feature space become orthogonal to one another and the principal components become

meaningless. Alternatively, as the width of σ increases the off diagonal tend toward 1,

$$\lim_{\sigma \to \inf} \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1. \tag{13}$$

In short, too small a bandwidth leads to over-separation of points in $\mathcal{F}$, a form of over-fitting. Conversely, too large a bandwidth forces all points to be mapped to similar locations in $\mathcal{F}$, a type of under-fitting [17]. Otherwise stated, in the former case all points appear to be anomalous in $\mathcal{F}$, while in the latter, all points appear normal. In the unsupervised setting it is not possible to perform a parameter search because there is no conventional sense of a hold-out validation set given the lack of labels. Section 4.1 describes a proposed solution to determining a nearly optimal kernel choice in the unsupervised setting.

In addition, computational efficiency, both in terms of time and space complexity, is an issue at several steps in the conventional deployment of kPCA and a major limiting factor for its applicability. Despite the demonstrated ability of kernel methods to fit non-linear patterns in data, calculating a distance matrix comprised of all pairwise similarity measures between each of the N data point in $\boldsymbol{X}$ is prohibitive. Specifically, the calculation of the adjacency matrix needed to form K, has $O(DN^2)$ tim e complexity and $O(N^2)$ space complexity. Of even greater concern is the $O(N^3)$ time complexity of the eigendecomposition of $\tilde{K}$ which is prohibitively expensive for large datasets. To address this problem we outline a process that greatly reduces computational cost without sacrificing detection accuracy in Section 4.2.

4.1. Learning the Kernel

For unsupervised tasks, a conventional grid search of any parameter space is not possible. Instead, a heuristic based on the nearest-neighbor distance, or some other distance adjacency metric is often used to select σ for kernel methods [18, 19, 20] . These heuristics are usually sub-optimal and tied to the dispersion of the data. Other methods such as [21, 22] require iterative evaluations of the kernel matrix or estimations of the error rate.

Evangelista and Embrechts [21] employ a powerful, general heuristic for selecting a near optimal value of σ without the need for labeled data. Their method is based on maximizing the coefficient of variance of the off-diagonal entries in the kernel matrix. Our proposed method extends their work to kPCA and significantly reduces the time and space complexity.

In the full $N \times N$ kernel matrix there are $N^2 - N$ off diagonal entries. Because of symmetry, half are duplicates, so there are only l *unique* off-diagonal entries, $l = (N^2 - N)/2$. Evangelista suggests the following *fundamental premise of pattern recognition*, that suggests a good model should follow,

$$\kappa(i,j)|(y_i = y_j) > \kappa(i,j)|(y_i \neq y_j), \tag{14}$$

which simply indicates that points that are closer in the ambient space will produce larger kernel values than distant points. For the Gaussian kernel, this can be seen as a consequence of

$$\lim_{\|\boldsymbol{x}_i - \boldsymbol{x}_j\| \to 0} \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1. \tag{15}$$

For anomaly detection, most pair-wise comparisons are of normal to normal data, i.e. $y_i = y_j$.

Algorithm 1: Batch Gradient Descent Optimization for σ

input : $\boldsymbol{X}$ -globally min-max normalized data
Given: N_b -batch sampling size, P -patience, σ_0 -initial σ
initialize β_0;
initialize $\mathcal{L}_{\min}$;
initialize $t = 0$ -batches since last $\mathcal{L}_{\min}$ update;
repeat
  randomly draw N_b samples from $\boldsymbol{X}$ to form a subsample $\boldsymbol{X}_b$;
  calculate K_b from $\boldsymbol{X}_b$;
  extract l_b off-diagonal unique entries from K_b;
  calculate $\mathcal{L}$ (17) ;
  apply gradient descent to update β;
  if $\mathcal{L} < \mathcal{L}_{\min}$ **then**
    $\mathcal{L}_{\min} = \mathcal{L}$;
    $t = 0$;
  else
    $t = t + 1$;
  end
until $t > P$;
output: $\bar{\sigma}$ during P

At first glance one might assume that simply tuning the matrix to take on high values will preserve the idea of adjacency in $\mathcal{F}$. This is misguided, however, and leads to a situation where anomalies are not pronounced due to under-fitting. Instead, the important metric is the dispersion of the data. Decomposing the "disperse" kernel matrix yields eigenvectors that are most representative of neighboring points (as they have proportionally higher values) while minimizing the impact of distant anomalies. This results in a good, non-linear model of the data.

The index of dispersion,

$$D = \frac{s^2}{\mu} \tag{16}$$

where s^2 is the variance and μ is the mean, provides a normalized measure of the spread of a distribution of values

relative to their mean (also called the coefficient of dispersion, coefficient of variation, relative variance, or variance-to-mean ratio). By applying this measure, it is possible to quantify the sparsity of the l off-diagonal kernel entries in $\tilde{K}$. Furthermore, the index of dispersion of the off-diagonal kernel entries exhibits a global maximum which is an ideal objective to optimize when seeking to determine σ [21].

One contribution of this work is to modify the objective to avoid the $O(iN^2)$ computational complexity associate with i iterative evaluations of the full kernel matrix. Instead of deploying the simple hill-climbing optimization used in [21], we instead formulate a loss function to fit the framework of mini-batch stochastic gradient descent. To begin, we uniformly draw N_b examples from $\boldsymbol{X}$ to form a batch, $\boldsymbol{X}_b$. We then apply the kernel to an adjacency matrix calculated from the batch to form K_b. Given the kernelized adjacency matrix for a batch, Equation 16 is inverted such that the objective becomes,

$$\mathcal{L}(\boldsymbol{X}_b, \sigma) = \frac{\mu_b}{{s_b}^2 + \epsilon}, \tag{17}$$

where s_b^2 is the variance of the off-diagonal entries of K_b, μ_b is the mean, and ϵ is a small term that prevents division by zero. Over the course of an optimization, precise notation would require that we indicate the i-th batch corresponding to the i-th iteration, $\boldsymbol{X}_{b,i}$, and all terms derived therefrom, but we drop the index corresponding to the iteration for the sake of simplicity and clarify if necessary.

The proposed sampling method is beneficial because the full kernel matrix need never be computed or stored in memory. Forming K_b in this way is equivalent to randomly drawing a set of rows from K, applying the same indexing to the columns, and saving the entries of intersecting rows and columns. This process relies on the assumption that the index of dispersion of the samples, D_b, approximates the D of all l entries, so that using iterative draws of D_b as a metric for tuning yields the same near optimal result for σ.

To prevent negative values for σ, the optimization is instead performed relative to a bias, β, which is passed through an activation function,

$$\sigma = \log(1 + \exp(\beta)), \tag{18}$$

where an initial β_0 is set to correspond to $\sigma_0 = 1$. Early stopping is performed by tracking the lowest value of the loss. The number of training steps (batches) since the recorded lowest loss is tracked and if the number exceeds a set patience, P, training is halted, and the average bandwidth over that period, $\bar{\sigma}$, is returned. The Algorithm 1 outlines the steps for tuning σ and the correlation between the minimum activation and the best performing bandwidth are shown in Figure 1 for two test data sets.

The proposed extensions of [21] to the batch gradient descent framework allows for unsupervised, efficient, and near-optimal kernel tuning on very large data sets. The space complexity is reduced to $O({N_b}^2)$ space and to $O(i_b {N_b}^2)$ time complexity, where i_b is the number of batches drawn before convergences is declared. In practice, batches as small as $N_b = 100$ are generally sufficient (see Figure 3) and convergence typically occurs in under 2000 steps when applying a batch gradient descent optimizer with momentum.

4.2. Skeleton Ensembles

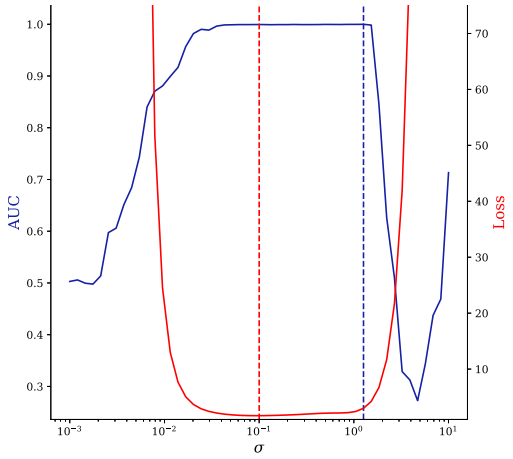
Even with an efficient means of computing an appropriate global bandwidth, the cubic time complexity of the eigenvalue decomposition of the $\tilde{K}$ makes performing kPCA on larger data sets infeasible. Here, we describe an alternative ensemble technique that avoids any full formations or decomposition of K. The key insight is that a reasonably small sampling of a collection of points has approximately the same principal components as the full data set.

Algorithm 2: Ensemble kPCA

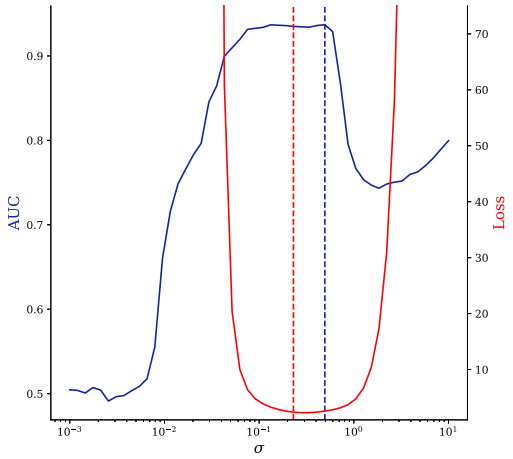
input : $\boldsymbol{X}$ -globally min-max normalized data, σ -rbf kernel parameter
Given: N_s -skeleton sampling size, N_m -number of models in the ensemble
for *model* in N_m **do**
  randomly draw N_s samples from $\boldsymbol{X}$ to form a skeleton subsample $\boldsymbol{X}_s$;
  form K_s from $\boldsymbol{X}_s$;
  decompose K_s to extract $\boldsymbol{\alpha}_s$;
  unit-norm $\boldsymbol{\alpha}_s$;
  calculate $d_E(\boldsymbol{x})$ for all $\boldsymbol{x} \in \boldsymbol{X}$ (11);
end
output: $\bar{d_E}(\boldsymbol{x})$ -average anomaly score for each example across all N_m models

The idea extends upon the concept of an out-of-sample extension [14, 15, 12, 13, 23], that is, a datum that was not originally used in the eigendecomposition of K can be still be projected onto the set of learned principal components. Some prior works have focused on finding a single good approximation of the kernel[11, 24] but in the unsupervised setting this can be problematic, as an "unlucky" random sampling may be strongly influenced by outliers. Our proposed method mitigates this problem with an ensemble of models that account for the errors produced by decomposing a lower-rank K.

We begin by uniformly drawing N_s samples from $\boldsymbol{X}$ to yield a *skeleton*, $\boldsymbol{X}_s$, which is used to produce an approximate low-rank kernelized adjacency matrix, K_s, from which the corresponding skeleton eigenvectors, α_s, that form an approximate model of the data are calculated. Given a single skeleton, the reconstruction error (11) for *all* points in $\boldsymbol{X}$ are found using the global bandwidth found

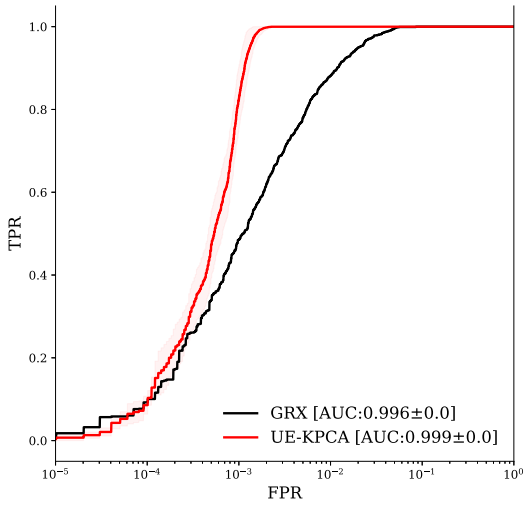


(a)

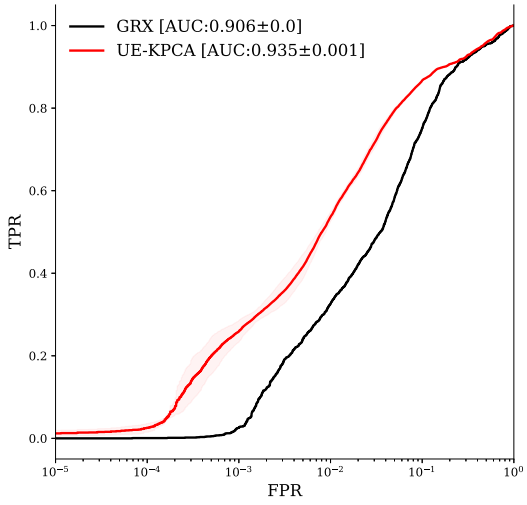


(b)

Figure 1: Detection performance (left vertical axis, blue curve) as measured by AUC (area under the curve) of a receiver operating characteristic (ROC) curve versus bandwidth parameter, σ, for the 4-class synthetic (a) and Forest Radiance (b) data sets. Value of the loss function given by Equation 17 is shown by the red curve with values on the right vertical axis. The loss function minimum (red vertical line) is either near the best-performing bandwidth (vertical blue line) for Forest Radiance or yields a bandwidth with nearly equivalent performance (4-class).



(a)



(b)

Figure 2: Receiver Operating Characteristic (ROC) curves showing true positive rate (TPR) versus false positive rate (FPR) for UE-kPCA (red) and GRX (black) on the 4-class (a) and Forest Radiance (b) data sets. Parameters were $N_b = 100$, $N_s = 256$, and $N_m = 100$ for each UE-kPCA curve. 30 ensemble curves were calculated and the red shading represents one standard deviation from the mean.

in Algorithm 1 and an out-of-sample extension. This process is repeated to form an ensemble of N_m approximate low-rank models where the same global σ is used for each model. The reconstruction errors across all N_m skeletons in the ensemble are averaged for each example in $\boldsymbol{X}$ to form a final anomaly score (although we note that other ensemble aggregation rules beside the average are possible). Algorithm 2 outlines the procedure.

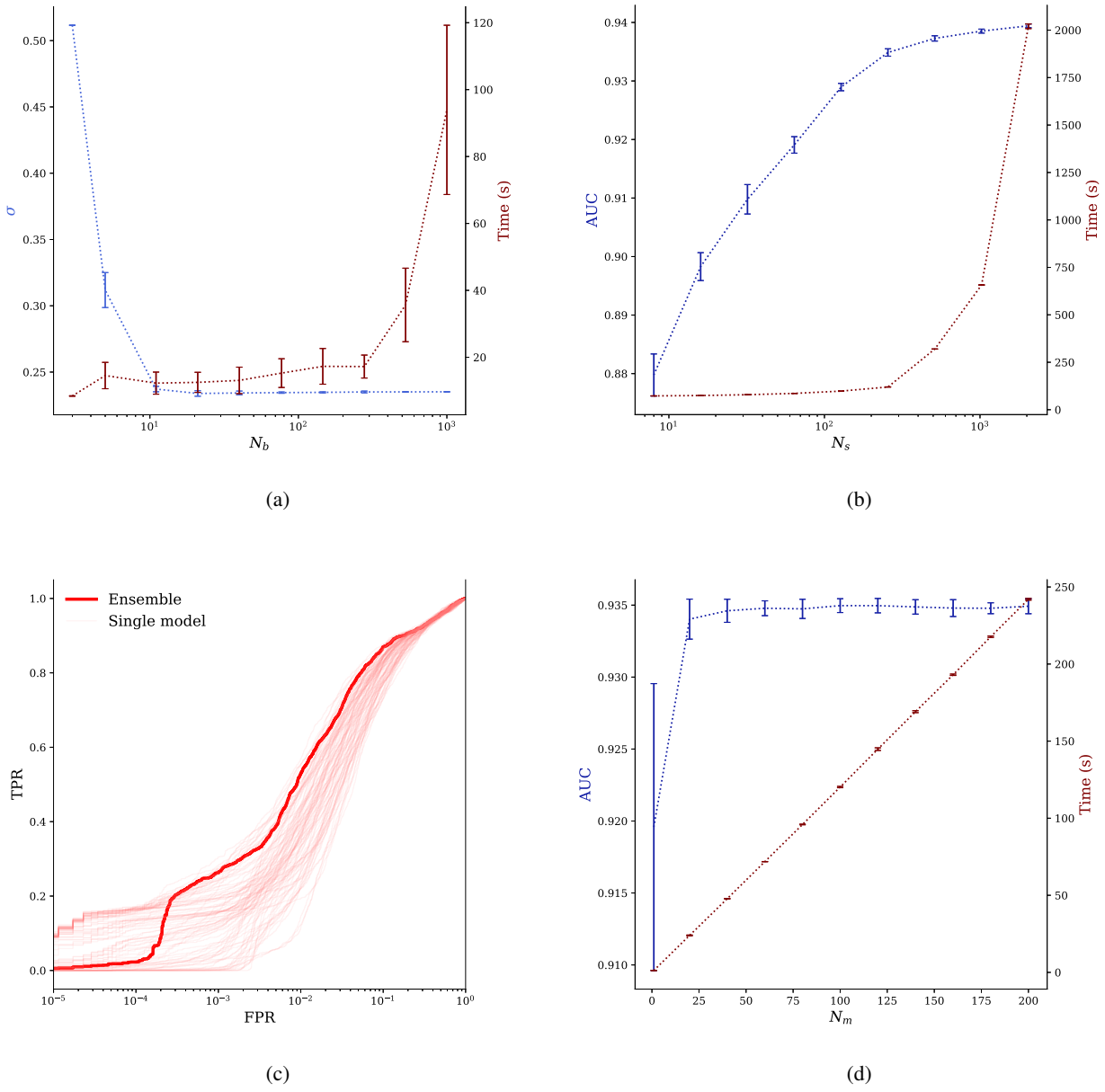


Figure 3: (a) Optimal kernel bandwidth (left axis, blue curve) and computation time (right axis, red) as a function of batch size. (b) AUC (blue) and computation time (red) as a function of skeleton size with $N_b = N_m = 100$. (c) Ensemble ROC curve (dark red) for $N_m = 100$ skeleton models with the ROC curve for each skeleton model (light red) where $N_s = 256$. (d) AUC (blue) and computation time (red) as a function of number of skeleton models with $N_b = 100$ and $N_s = 256$.

This procedure of model averaging, also known as bootstrap aggregation or *bagging*, is an ensemble method that has been primarily been primarily developed for decision tree methods, such as Isolation Forest [25, 26, 27]. Notably, this type of sampling does not work as well for many other anomaly detection methods such as OC-SVMs or distance-based approaches. As opposed to kPCA, where the Principal Components are comparable, the margins generated by OC-SVMs and nearest neighbor rankings vary significantly because the distance between the distances between points are much larger in the sample. Similar methods have applied KPCA ensembles to applications such as image denoising, however, this evaluation is the first to apply an ensemble version as a general approach to the problem of un-

supervised anomaly detection.

This sampling process greatly reduces the computational complexity of kPCA, even when accounting for the multiple evaluations necessary in the ensemble. The computational complexity of the eigendecomposition step is reduced from $\mathcal{O}(N^3D)$ to $\mathcal{O}(N_m{N_s}^3D)$. However, the scoring across all models requires a $\mathcal{O}(N_mN)$ time, but because only modest values of N_s and N_m are necessary to approach the accuracy of the full rank evaluation, the ensemble method quickly becomes the preferred approach as the cardinality increases (see Figure 3). Furthermore, each model's score can be calculated in parallel to reduce computation time.

5. Experimental Description and Results

5.1. Data

We use two datasets to compare performance betweem our algorithm and GRX: an HSI scene from the Forest Radiance I collect and a synthetic "bag-of-pixels" with no spatial information constructed from an AVARIS scene. Forest Radiance pixels are comprised of $D = 158$ bands pulled from a 600 x 300 segment of the original image (run05) as described in [19]. Synthetic data from the AVARIS collect ($D = 191$ bands) are generated by the subsample-and-average procedure described in [28]. The final dataset has 100,000 generated pixels representing four classes with an anomalous class abundance of 1.234%.

5.2. Results

We measure detection performance by calculating the area under the curve (AUC) of a receiver operating characteristic (ROC) curve. In essence, the ROC curve measures how well we have separated anomalous pixels from background for all threshold settings associated with a given algorithm. ROC curves corresponding to GRX and UE-kPCA for both data sets are provided in Figure 2. Given the random character of UE-kPCA we construct multiple ensembles and show the variance associated with the ensemble of ensembles as a shaded red region around the average ensemble performance (dark red curve).

Our method has similar performance to GRX at very low false positive rates for the 4-class synthetic data but achieves 100% detection at a false positive rate that is nearly two orders of magnitude lower than that of GRX. GRX performs poorly compared to our method for all false positive rates of interest on the Forest Radiance data set.

5.3. Parameter Study

Our UE-kPCA method introduces a number of nuisance parameters beyond the bandwidth parameter, namely: batch size, skeleton size, and number of skeleton models (N_b, N_s, and N_m, respectively). Here, we illustrate detection performance (AUC) and computation time for the Forest Radiance image as a function of those parameters in Figure 3.

In particular, from Figure 3a we see that computation time is not heavily affected by batch size over nearly two orders of magnitude from 10 to 1000 samples per batch. More importantly, the bandwidth parameter quickly converges to the global optimum when $N_b < 100$. Based on this (admittedly empirical) study we would recommend a batch size somewhat greater than 10 but less than 100. We set $N_b = 100$ for all ROC calculations shown in Figure 2.

Figure 3b illustrates the relative importance of skeleton size, N_s. Computation time is not a strong function of skeleton size until it reaches a critical threshold at which point it increases exponentially. Given bandwidth fixed at the optimum found by Algorithm 1 and $N_m = 100$, the AUC for a ROC curve calculated given a specific value of N_s increases approximately linearly from 10 to 100 before leveling off near the threshold where computational time begins to increase significantly. We set $N_s = 256$ for the ROC calculations in Figure 2.

ROC performance as a function of the number of skeleton models within an ensemble, N_m, is shown in Figure 3d. Clearly, ensemble performance plateaus quickly once a threshold number of models has been included while computational time increases linearly with no parallelization. ROC results in Figure 2 where calculated using $N_m = 100$. Figure 3c shows the ensemble ROC curve (dark red) calculated from 100 separate skeleton models where the light red curves illustrate ROC performance for each of the individual skeleton models ($N_s = 256$). Some of our initial studies for future work indicate that alternate ensemble aggregation methods may allow improvement at low false positive rates. Finally, we found UE-kPCA performance to be relatively insensitive to the number of retained principal components between 30 and 200 and therefore set $M = 75$ for all computations throughout.

6. Conclusion

We introduced here Unsupervised Ensemble-kernel Principal Component Analysis (UE-kPCA) for hyperspectral anomaly detection. By defining the reconstruction error in kPCA feature space as our anomaly score we showed that we can reliably select a Gaussian kernel bandwidth, σ, that yields nearly optimal detection performance. We formulated a novel loss function that can be minimized using minibatch gradient descent to consistently choose a σ that yields satisfactory detection performance. We then coupled this reliable bandwidth selection process with an ensemble sampling method that significantly reduces computation time. These two innovations enable application of unsupervised kPCA to data sets that were previously infeasible due to computational constraints.

References

[1] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PLoS ONE*, Apr 2016.

[2] G. Enderlein, "Hawkins, d. m.: Identification of outliers. chapman and hall, london – new york 1980, 188 s., £ 14, 50," *Biometrical Journal*, vol. 29, no. 2, pp. 198–198, 1987.

[3] G. O. Campos, A. Zimek, J. Sander, R. J. G. B. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle, "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study," *Data Mining and Knowledge Discovery*, vol. 30, pp. 891–927, Jul 2016.

[4] H. Hoffmann, "Kernel pca for novelty detection," *Pattern Recognition*, vol. 40, no. 3, pp. 863 – 874, 2007.

[5] P. F. Evangelista, M. J. Embrechts, and B. K. Szymanski, "Some properties of the gaussian kernel for one class learning," in *Artificial Neural Networks – ICANN 2007* (J. M. de Sá, L. A. Alexandre, W. Duch, and D. Mandic, eds.), (Berlin, Heidelberg), pp. 269–278, Springer Berlin Heidelberg, 2007.

[6] N. Goix, "How to evaluate the quality of unsupervised anomaly detection algorithms?," 07 2016.

[7] I. S. Reed and X. Yu, "Adaptive multiple-band cfar detection of an optical pattern with unknown spectral distribution," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, pp. 1760–1770, Oct 1990.

[8] D. M. Tax and R. P. Duin, "Support vector domain description," *Pattern Recognition Letters*, vol. 20, no. 11, pp. 1191 – 1199, 1999.

[9] B. Schölkopf, A. Smola, and K. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, July 1998.

[10] C. M. Bachmann, T. L. Ainsworth, and R. A. Fusina, "Exploiting manifold geometry in hyperspectral imagery," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 441–454, 2005.

[11] C. C. Olson and T. Doster, "A parametric study of unsupervised anomaly detection performance in maritime imagery using manifold learning techniques," in *SPIE Defense+ Security*, pp. 984016–984016, International Society for Optics and Photonics, 2016.

[12] S. Lafon, Y. Keller, and R. R. Coifman, "Data fusion and multicue data matching by diffusion maps," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 28, no. 11, pp. 1784–1797, 2006.

[13] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet, "Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering," in *Advances in Neural Information Processing Systems*, vol. 16, Cambridge, MA, USA: The MIT Press, 2004.

[14] C. T. H. Baker, *The Numerical Treatment of Integral Equations*. Oxford: Clarendon Press, 1977.

[15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. Cambridge: Cambridge University Press, 1988.

[16] M. Amer, M. Goldstein, and S. Abdennadher, "Enhancing one-class support vector machines for unsupervised anomaly detection," in *ODD '13*, 2013.

[17] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. USA: Cambridge University Press, 2004.

[18] F. Hallgren and P. Northrop, "Incremental kernel pca and the nyström method," 01 2018.

[19] C. C. Olson and T. Doster, "A novel detection paradigm and its comparison to statistical and kernel-based anomaly detection algorithms for hyperspectral imagery," in *Proc. CVPRW*, pp. 302–308, IEEE, 2017.

[20] A. Budynkov and S. Masolkin, "The problem of choosing the kernel for one-class support vector machines," *Automation and Remote Control*, vol. 78, pp. 138–145, 01 2017.

[21] P. Evangelista, M. Embrechts, and B. Szymanski, "Some properties of the gaussian kernel for one class learning," pp. 269–278, 09 2007.

[22] D. M. J. Tax and R. P. W. Duin, "Support vector domain description," *Pattern Recognition Letters*, vol. 20, pp. 1191–1199, 1999.

[23] C. Olson, K. Judd, and J. Nichols, "Manifold learning techniques for unsupervised anomaly detection," *Expert Systems with Applications*, vol. 91, pp. 374 – 385, 2018.

[24] C. C. Olson, M. Coyle, and T. Doster, "A study of anomaly detection performance as a function of relative spectral abundances for graph-and statistics-based detection algorithms," in *Proc. SPIE*, p. 101980X, ISOP, 2017.

[25] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[26] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, (Washington, DC, USA), pp. 413–422, IEEE Computer Society, 2008.

[27] C. C. Aggarwal and S. Sathe, "Theoretical foundations and algorithms for outlier ensembles," *SIGKDD Explor. Newsl.*, vol. 17, p. 24–47, Sept. 2015.

[28] T. H. Emerson, J. A. Edelberg, T. Doster, N. Merrill, and C. C. Olson, "Generative and encoded anomaly detectors," *10th IEEE Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2019.