RODS: Rarity based Outlier Detection in a Sparse Coding Framework

Jayanta K. Dutta, Bonny Banerjee, *Member, IEEE*, and Chandan K. Reddy, *Senior Member, IEEE*

Abstract—Outlier detection has been an active area of research for a few decades. We propose a new definition of outlier that is useful for high-dimensional data. According to this definition, given a dictionary of atoms learned using the sparse coding objective, the outlierness of a data point depends jointly on two factors: the frequency of each atom in reconstructing all data points (or its negative log activity ratio, NLAR) and the strength by which it is used in reconstructing the current point. A **R**arity based **O**utlier **D**etection algorithm in a **S**parse coding framework (RODS) that consists of two components, NLAR learning and outlier scoring, is developed. This algorithm is unsupervised; both the offline and online variants are presented. It is governed by very few manually-tunable parameters and operates in linear time. We demonstrate the superior performance of the RODS in comparison with various state-of-the-art outlier detection algorithms on several benchmark datasets. We also demonstrate its effectiveness using three real-world case studies: saliency detection in images, abnormal event detection in videos, and change detection in data streams. Our evaluations shows that RODS outperforms competing algorithms reported in the outlier detection, saliency detection, video event detection, and change detection literature.

Index Terms—Anomaly detection, saliency detection, abnormal event detection, change detection, data streams

1 Introduction

The era of Big Data has ushered in an unprecedented interest in efficient detection of abnormal data in multiple scientific disciplines. Outlier detection algorithms play a crucial role in detecting the abnormal patterns that significantly deviate from the norm. Over the last few decades, the problem of outlier detection has continued to garner interest from both academic researchers and practitioners working on real-world applications. Outlier detection is fundamental to many real-world applications, such as fraud detection [1], network intrusion [2], clinical diagnosis [3], customer relations management [4] and biological data analysis [5], [6].

In spite of the rich literature addressing this problem, there has been relatively less effort in transforming the feature space and extracting outliers through sparse representation of the data. In this paper, we present a fast unsupervised algorithm for outlier detection using a sparse coding based reconstruction framework. A linear model for the data is assumed whereby each data point $\vec{x} \in \mathbb{R}^m$ is represented as the linear combination of a dictionary $\mathbf{D} = [\vec{d}_1, \ldots, \vec{d}_k] \in \mathbb{R}^{m \times k}$ of non-orthogonal bases (or atoms). That is, $\vec{x} = \sum_j \vec{d}_j \gamma_j$ where $\gamma_j \in \mathbb{R}$ is the coefficient corresponding to $\vec{d}_j$. The absolute value of γ_j signifies the strength of $\vec{d}_j$ in representing $\vec{x}$. Informally, we define a data point as an outlier if it consists of one or more atoms with significant strength that rarely occur in the other observed data points.

The intuition behind our definition of outlier is as follows. A linear generative model assumes that a data point or observation is the effect of a linear combination of multiple causes, some of which play a stronger role than others in generating that point. Each cause, represented as a dictionary atom, has the same dimension as a data point. If a cause rarely plays a strong role in generating the observed data points, as captured by its *rarity*, its strong involvement in generating a particular point renders that point an outlier. Our definition of outlier, which leads to the **R**arity based **O**utlier **D**etection in a **S**parse coding framework (**RODS**) algorithm, is illustrated in Fig. 1 using data points in 2D space.

In Fig. 1a, data points are shown with red dots. There are four clusters. Six dictionary atoms learned using a sparse coding objective are shown with black circles and labeled as $\vec{d}_1, \vec{d}_2, \ldots, \vec{d}_6$. The space is partitioned based on the absolute cosine similarity between the data points and the atoms. Each partition belonging to an atom is marked with its corresponding label. The number of data points in the partitions belonging to an atom determines the atom's rarity using negative log activity ratio (or NLAR). More the number of points in the partition of an atom, lower is its NLAR, as shown in Fig. 1b. An outlier score of a point is the weighted sum of the NLARs of the atoms where the weights are absolute coefficients required to reconstruct the point using these atoms. To illustrate the difference between our notion of "outlier" and traditional definitions using clustering with Euclidean norm, two data points, $\vec{x}_1$ and $\vec{x}_2$, are considered in Fig. 1a. $\vec{x}_1$ is reconstructed using $\vec{d}_1$, $\vec{d}_6$ and $\vec{d}_3$, in decreasing order of their strength. Since none of these three atoms is very rare (as shown in Fig. 1b), $\vec{x}_1$ is not an

- *J.K. Dutta is with the Department of Electrical & Computer Engineering, University of Memphis, Memphis, TN 38152. E-mail: jkdutta@memphis.edu.*
- *B. Banerjee is with the Institute for Intelligent Systems, and the Department of Electrical & Computer Engineering, University of Memphis, Memphis, TN 38152. E-mail: bbnerjee@memphis.edu.*
- *C.K. Reddy is with the Department of Computer Science, Wayne State University, Detroit, MI 48202. E-mail: reddy@cs.wayne.edu.*

Manuscript received 17 Mar. 2015; revised 25 June 2014; accepted 7 Aug. 2015. Date of publication 2 Sept. 2015; date of current version 6 Jan. 2016. Recommended for acceptance by S. Yan.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2015.2475748

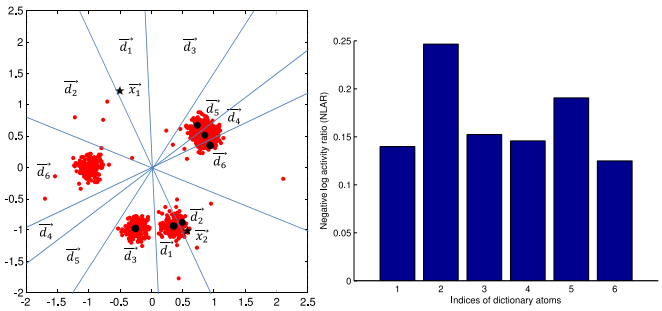


Fig. 1. Geometric interpretation of the proposed RODS algorithm.

outlier by our definition. $\vec{x}_2$ is reconstructed using $\vec{d}_2$. Since $\vec{d}_2$ is rarely used (as shown in Fig. 1b), $\vec{x}_2$ is an outlier by our definition. In contrast, $\vec{x}_1$ would be judged as an outlier and $\vec{x}_2$ not an outlier if a clustering based approach for outlier detection with Euclidean norm was used.

The key contributions of this paper are as follows:

1) A *new definition of outlier* by which a data point is deemed an outlier if it is strongly constituted of non-orthogonal bases (or dictionary atoms) that rarely constitute other data points. This definition is particularly useful for high-dimensional data. An outlier scoring function, based on the Shannon information content for the activity of the atoms that assigns higher scores to data points strongly constituted of rarer atoms, is introduced.
2) *An outlier detection algorithm, RODS, consisting of two components—rarity learning (as NLAR) and outlier scoring.* The learning algorithm is unsupervised; both offline and online variants are presented. The algorithm has very few parameters, and operates in time linear in size of the dataset for the offline case and constant time per input for the online case applicable to streaming data.
3) *Superior performance of the proposed RODS algorithm in comparison with various state-of-the-art outlier detection algorithms on a number of benchmark datasets.*
4) *Effectiveness of the proposed RODS algorithm in real-world problems, such as saliency detection in images, abnormal event detection in videos and change detection in data streams.* Performance evaluations show that the proposed algorithm outperforms various competing methods.

The rest of this paper is organized as follows. In Section 2, a brief review of the related literature is provided. Section 3 introduces the notations and definitions that are necessary to comprehend our proposed algorithm. The proposed algorithms are described in Section 4 and are evaluated on various benchmark datasets including image collections and video streams in Section 5. Finally, our discussion concludes in Section 6.

2 Related Work

Comprehensive survey articles describing outlier detection algorithms for different kinds of data and applications are available in the literature (see [7], [8] for example). Here we briefly review prior works related to basic outlier detection methods, methods for high-dimensional data and data streams.

2.1 Outlier Detection Methods

Traditionally, approaches relied on robust statistics for outlier detection [9]. Subsequently a number of methods were developed for detecting outliers using the notion of proximity (distance) of a given data point with reference to the other points [10]. One of the earliest approaches [11], using proximity information, estimates the outlierness of each sample based on the average distance to the k-nearest neighbors. The proximity-based methods can be broadly classified into density-based and clustering-based methods. Instead of relying on the neighborhood information, the density-based methods aim at estimating the density surrounding each data point; the outliers are assumed to occur in extremely low density regions. One of the representative methods in this category is the Local Outlier Factor (LOF) method [12]. In clustering-based methods, such as [13], the distance of a point from its closest cluster centroid is usually chosen as the metric for the outlierness of the point. One of the major limitations of such clustering-based methods is their assumption regarding the shape and form of the clusters. Other machine learning algorithms, such as active learning [14] and ensemble methods [15], [16], can sometimes be more efficient and produce robust outlier detection results. They are mostly based on different notions of proximity (or distance). Such distance-based assumptions might yield suboptimal results in high-dimensional feature spaces and cause a dilemma on the choice of metrics and threshold values to use.

2.2 High-Dimensional Outlier Detection

Spectral methods specifically address outlier analysis in high-dimensional data by projecting them to lower-dimensional spaces [17]. They operate on the assumption that a lower dimensional manifold exists in which, if the data is embedded, the normal and anomalous instances will appear significantly different [7]. Thus, spectral methods are useful only when such manifolds truly exist. Also, they suffer from high computational complexity. Nevertheless, this topic has received a lot of attention lately; see [18] for an extensive survey on various subspace methods related to outlier detection. The LOCI method [19] is based on the idea of estimating the local behavior of the sample space using correlation integrals. One of the simplest and intuitive approaches for high-dimensional data uses the idea of principal components to estimate the outlierness of the data [20]. Such projection based methods, where the original space is transformed into a lower-dimensional feature space using certain linear transformation, are not suitable for detecting outliers in complex feature spaces. Non-linear projection methods would be more suitable for identifying outliers. However, such non-linear transformations are often difficult to learn in real-world scenarios. Also, some of these methods aim at identifying outliers in a low-dimensional subspace which correspond to certain subsets of the original feature space. Such feature subset based outlier detection methods are not applicable for images and videos. In contrast to the above methods, our proposed method

projects the data to a higher dimensional manifold using a linear transformation such that the data representation becomes sparse.

2.3 Feature Space Transformation Methods

There has been relatively less work in transforming the feature space and detecting outliers based on the rarity of the transformed indices used for representing the data. The only work closely related to the proposed method is in the context of unusual event detection in videos [21], [22]. A framework for simultaneous sparse coding and anomaly detection by adding an extra term to the sparse coding objective function is presented in [21]. Outliers have nonzero columns in this extra term which signifies a deviation from the model. The approach is offline, all the data vectors are needed to determine the outliers, and the maximum number of outliers to be captured is a parameter that is difficult to estimate. In [22], anomalous events in videos are detected from the error in reconstructing each 3D event (first two dimensions represent space while the third dimension represents time); the reconstruction utilizes a sparse coding approach. While the proposed method also represents the data using sparse coding, the outlierness of a data point is a function of the rarity of the dictionary atoms and not the reconstruction error.

In sparse coding, each data point is represented as a linear combination of dictionary atoms where the coefficient vector is allowed a sparse number of nonzero entries. Thus, sparse coding can be viewed as a generalization of the vector quantization (VQ) or clustering objective where each signal is typically represented by a single atom. Sparse coding enables a more accurate representation of the input signal with same memory requirements as VQ/clustering but slightly higher computational cost. Sparse coding has been a better performer in many real-world applications including image denoising [23], [24], [25], texture synthesis [26], edge detection [27], image super-resolution [28], audio processing [29], [30], image classification [31]. It has been extensively used for handling high-dimensional data effectively [23], [24]. The idea of dictionary learning has produced promising results in the context of classification [32]. However, it has seldom been investigated for the problem of outlier detection, especially for datasets other than images and videos. A number of methods for efficient learning of sparse dictionaries have been reported, such as K-SVD [23] and ODL [24] (also see [33] for recent work on reducing the complexity of the sparse coding problem). Due to its effectiveness and simplicity, we chose the K-SVD algorithm for dictionary learning.

2.4 Outlier Detection for Data Streams

Very few algorithms have been reported in the literature that produce state-of-the-art outlier detection results in both static and streaming data which is one of the goals of the proposed RODS algorithm. Making static outlier detection algorithms applicable to the streaming data context requires non-trivial enhancements. See [34], [35], [36], [37] for examples of work along this line. The goal of some of these methods is to efficiently calculate the local outlier factors for streaming data through an element of sliding window-based approach and making proximity-based assumptions regarding the outliers. Such methods suffer from the drawbacks of the proximity-based methods mentioned earlier in this section. In this paper, an online variant of the static RODS algorithm is proposed for streaming data where the NLARs are updated as each new data point arrives.

TABLE 1
Notations Used in This Paper

Notation	Description
$\mathbf{X}$	Dataset
N	Number of data points in $\mathbf{X}$
m	Dimension of each data point or atom
$\mathbf{D}$	Dictionary of atoms
k	Number of atoms in $\mathbf{D}$
Γ	Coefficient matrix for representation of $\mathbf{X}$
κ	Maximum of ℓ_0 norm for sparse Γ
$\vec{p}$	Activity ratio of atoms
$\vec{q}$	Summary activity ratio of atoms
$\vec{\theta}$	Negative log activity ratio of atoms
$\vec{s}$	Outlier score of dataset
ζ	Outlier scoring function
ω	Outlier decision threshold
I	Index vector

3 Preliminaries

In this section, we describe the notations and definitions needed to understand our proposed algorithm.

3.1 Notations Used

The notations used in this paper are provided in Table 1. Matrices are denoted by bold uppercase letters (e.g., $\mathbf{Z}$), while lowercase letters with vector sign denote column vectors (e.g., $\vec{z}$). The columns of a matrix are represented by corresponding lowercase letters (e.g., $\mathbf{Z} = [\vec{z}_1, \ldots, \vec{z}_k]$). The elements of a vector are denoted by letters without vector sign (e.g., $\vec{z} = (z_1, \ldots, z_n)$). $\vec{0}$ denotes a zero vector with size depending on the context. The elements of a matrix are denoted using subscript where row and column indices are separated by a comma (e.g., $\mathbf{Z}_{i,j}$ denotes the element corresponds to the ith row and jth column). Iteration numbers in a loop are indicated by superscripts (e.g., $\mathbf{Z}^c$). Time indices are denoted in a parenthesis after the variable (e.g., $\mathbf{Z}(t)$).

3.2 Problem Statement

Given a set of data points $\mathbf{X} = [\vec{x}_1, \ldots, \vec{x}_N] \in \mathbb{R}^{m \times N}$, an outlier data point $\vec{x}_c$ is defined as:

$$\vec{x}_c \in \mathbf{X}, \quad \zeta(\vec{x}_c|\mathbf{X}) \geq \omega, \tag{1}$$

where $\vec{x}_i \in \mathbb{R}^m$, m is the dimension of each data point, $\zeta : x \rightarrow \mathbb{R}^+$ is a scoring function, $\mathbb{R}^+$ is the set of non-negative real numbers and ω is a threshold. ζ assigns an *outlier score* to each data in $\mathbf{X}$ based on its frequency of occurrence in comparison to that of the other observed data. Rarer data points are assigned higher scores. A data could be a pixel, region, object or event, depending on the nature of the data and the goal. For example, if the data is a video sequence, $\mathbf{X}$ is a set of events $\vec{x}_i$ defined at each point (x_i, y_i, t_i) in the video where (x_i, y_i) refers to the spatial location in a frame

and t_i is the index of the frame. The crux of the problem is to discover the function ζ such that unusual or rare data in a dataset can be detected.

Our goal is to build a fast and accurate outlier detection algorithm for practical real-world applications. It is desirable that the proposed method quickly builds a model of normality and detects outliers while incrementally updating itself in an unsupervised manner as new normal patterns are observed.

3.3 Definitions

We now define the terms and concepts relevant to our algorithm.

Definition 1 (Sparse representation). *Let $\vec{x} \in \mathbb{R}^m$ be a data point. It admits a sparse representation $\vec{\gamma} \in \mathbb{R}^k$ over a dictionary of k atoms, $\mathbf{D} \in \mathbb{R}^{m \times k}$, if $\vec{x}$ can be represented as a linear combination of κ atoms in $\mathbf{D}$ and $\kappa \ll k$.*

Definition 2 (Dictionary learning). *The dictionary learning task is to compute a dictionary such that it is well adapted for reconstructing a set of data points. Given a dataset of size N, $\mathbf{X} = [\vec{x}_1, \ldots, \vec{x}_N]$, a dictionary of k atoms with a sparsity constraint can be learned by solving the following optimization problem:*

$$\min_{\Gamma, \mathbf{D}} \frac{1}{2} \sum_{i=1}^{N} \|\vec{x}_i - \mathbf{D}\vec{\gamma}_i\|_2^2 \quad \text{subject to} \quad \|\vec{\gamma}_i\|_0 \leq \kappa \; \forall i, \tag{2}$$

where $\Gamma = [\vec{\gamma}_1, \ldots, \vec{\gamma}_N] \in \mathbb{R}^{k \times N}$ is a sparse representation matrix, $\|.\|_0$ denotes the ℓ_0 pseudo-norm, the number of non-zero elements. κ is the maximum number of non-zero elements allowed in each $\vec{\gamma}_i$ and $\kappa \ll k$. Each element $\vec{d}_j \in \mathbf{D}$ $(j = 1, \ldots, k)$ is constrained to have a unit ℓ_2 norm.

Since ℓ_0 minimization is NP-hard, the ℓ_1 norm is widely used. Minimizations of both norms are equivalent if the solution is sufficiently sparse [38]. The dictionary learning objective is not jointly convex, but convex with respect to each of the two variables $\mathbf{D}$ and $\mathbf{\Gamma}$. Hence it is minimized by alternating between the two variables, minimizing over one while keeping the other fixed until the dictionary reaches a stable state [23], [24]. A series of alternate minimization steps reduce the mean square error of the overall objective function, and therefore, convergence to a local minimum is guaranteed under necessary conditions. The proofs and necessary conditions for convergence of different dictionary learning algorithms can be found in the literature (e.g., K-SVD [23], ODL [24]).

Definition 3 (Sparse coding). *Given a fixed dictionary $\mathbf{D} \in \mathbb{R}^{m \times k}$ and a data point $\vec{x} \in \mathbb{R}^m$, the sparse linear representation $\vec{\gamma} \in \mathbb{R}^k$ can be obtained by solving the following sparse approximation problem:*

$$\min_{\vec{\gamma}} \frac{1}{2} \|\vec{x} - \mathbf{D}\vec{\gamma}\|_2^2 \quad \text{subject to} \quad \|\vec{\gamma}\|_0 \leq \kappa. \tag{3}$$

This sparse approximation problem can be efficiently solved using Orthogonal Matching Pursuit (OMP) [33] which is a greedy forward selection algorithm that starts with an empty list and includes at each iteration the atom most correlated with the current residual. The residual starts with the input and the coefficients of the selected atoms are updated by computing the orthogonal projection of the input onto the linear subspace spanned by the atoms selected so far. Then the residual is recomputed. This procedure continues until κ atoms have been used or the ℓ_2 norm of the residual becomes smaller than a small predefined constant ϵ. The batch version of OMP [33] is used for the offline case, which speeds up the process by a considerable amount.

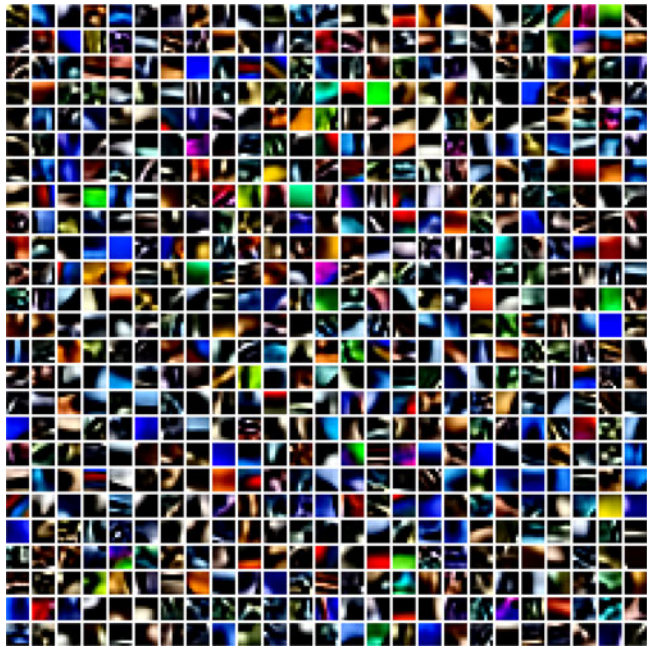


Fig. 2. Dictionary atoms learned using sparse coding. Atoms of size 8×8 pixels learned from natural images in the Toronto dataset.

Definition 4 (Dictionary update). *Given the sparse representations $\vec{\gamma}_i$ of the data points $\vec{x}_i$, the optimal dictionary $\mathbf{D}$ is the solution of the following optimization problem:*

$$\mathbf{D} = \underset{\mathbf{D}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{2} \|\vec{x}_i - \mathbf{D}\vec{\gamma}_i\|_2^2 \right). \tag{4}$$

This optimization problem can be solved using K-SVD [23]. However, the exact solution can become computationally difficult as the size of the data increases. So we use the approximate version of K-SVD [33] for learning the dictionary where a single iteration of the alternate minimization between sparse representation and dictionary update is generally sufficient to provide very close results to the full computation. Dictionary of small image patches learned using the dictionary learning algorithm is shown in Fig. 2.

4 The Proposed RODS Algorithm

In this section, we describe the batch and online versions of the proposed RODS algorithm.

4.1 Batch-RODS Algorithm

Given the dataset $\mathbf{X} = [\vec{x}_1, \ldots, \vec{x}_N] \in \mathbb{R}^{m \times N}$, first the dictionary $\mathbf{D} \in \mathbb{R}^{m \times k}$ is learned and the sparse linear coefficient vectors $\Gamma = [\vec{\gamma}_1, \ldots, \vec{\gamma}_N] \in \mathbb{R}^{k \times N}$ are computed using Eqs. (2) and (3) respectively (using the Batch-OMP and approximate K-SVD). This is a very well-established procedure for matrix

factorization and has been adopted in this paper. However, rest of the proposed RODS algorithm may be suitable for other kinds of dictionary learning and encoding procedures, such as using principal/independent components, clustering, etc. as well. Also, it is important to note that the encoding technique is much more important than the quality of the learned dictionary [39], hence it is acceptable to start the RODS algorithm with a given or randomly-chosen dictionary provided the encoding scheme to be used is known. The data points admit sparse representations when the dictionary is overcomplete [23], [24].

Next, the activity ratio, p_j, for jth atom ($j = 1, \ldots, k$) over the dataset which corresponds to the probability of using the jth atom based on the dataset is computed as follows:

$$p_j = \frac{\sum_{n=1}^{N} |\mathbf{\Gamma}_{j,n}|}{\sum_{i=1}^{k}\sum_{n=1}^{N} |\mathbf{\Gamma}_{i,n}|}. \tag{5}$$

The activity ratio will be lower for rarely used atoms and higher for frequently used ones. Rarity of the atoms is inversely proportional to their corresponding activity ratio. The rarity of dictionary atoms is computed using NLAR. The NLAR of the jth atom, denoted θ_j, is defined as:

$$\theta_j = -log_2(p_j). \tag{6}$$

NLAR of a dictionary atom is the Shannon information content for the activity of that atom. Information content represents the "surprise" of the activity, hence also referred to as *surprisal*. Use of a rarely-used atom in reconstructing a data point is improbable and evokes surprise when it is actually used. Thus, information is a function of the probability of the atom's activity; the smaller its probability of use, larger its information content. It is noteworthy that information content is a proper scoring of the activity of atoms, i.e. its objective is to track the true probability distribution of the atoms' activities. It is the one out of many possible scoring functions that has been widely studied in different fields and produced state-of-the-art results in our experiments with different kinds of benchmark datasets.

After calculating NLAR score vector $\vec{\theta}$, it is normalized to convert to a probability distribution function by dividing the NLAR of each element by the sum of NLARs of all elements in the vector.

Finally, given the data $\mathbf{X}$, the data point $\vec{x}_i$ and its coefficient vector $\vec{\gamma}_i$ with respect to a learned dictionary $\mathbf{D}$, the outlier score s_i is defined as:

$$\zeta(\vec{x}_i|\mathbf{X}) = s_i = |\vec{\gamma}_i|^T\vec{\theta}. \tag{7}$$

Thus, a data point is an outlier if it consists of one or more dictionary atoms with significant strength that rarely occur in all the observed data points. This score assigns a probabilistic degree of being an outlier to each data point. The overall algorithm is presented in Algorithm 1. The time complexity of this algorithm is dominated by that of Batch-OMP which is $O(N(2mk + \kappa^2 k + 3\kappa k + \kappa^3) + mk^2)$ [33] where m, k and κ are constants with respect to the size of the dataset (N). Thus, the algorithm runs in time linear in the size of the dataset.

4.2 Online-RODS Algorithm

For the online case, at each time t, a data vector $\vec{x}(t)$ is drawn from the stream $\mathbf{X}$ and its outlier score is computed. First, the activity ratio at t is computed as:

$$p_j(t) = \frac{|\gamma_j(t)|}{\sum_{i=1}^{k} |\gamma_i(t)|}. \tag{8}$$

Algorithm 1. Batch RODS

Inputs: $\mathbf{X} \in \mathbb{R}^{m\times N}$, k, κ
Output: $\vec{s} \in \mathbb{R}^N$: outlier score
Compute $\mathbf{D}$ by solving Eq. (2)
Compute Γ by solving Eq. (3)
for $j = 1$ to k **do**
    $p_j \leftarrow \frac{\sum_{n=1}^{N} |\Gamma_{j,n}|}{\sum_{i=1}^{k}\sum_{n=1}^{N} |\Gamma_{i,n}|}$
    $\theta_j \leftarrow -log_2(p_j)$
end for
$\theta_j \leftarrow \theta_j / \sum_{i=1}^{k} \theta_i, \forall j = 1, \ldots, k$
for $i = 1$ to N **do**
    $s_i \leftarrow |\vec{\gamma}_i|^T\vec{\theta}$
end for
Return $\vec{s}$

Theorem 1. *Let the summary activity ratio at any time t be incrementally updated as follows:*

$$\vec{q}(t) = (1 - \alpha(t))\vec{q}(t-1) + \alpha(t)\vec{p}(t). \tag{9}$$

Then, there exists a function $\alpha : \mathbb{N}^+ \rightarrow (0, 1]$, where $\mathbb{N}^+$ is the set of positive integers, such that $\vec{q}$ converges to a stable solution.

Let α be a function of time, and $0 < \alpha(t) \leq 1, \forall t$. Thus, in order to determine the new estimate $\vec{q}(t)$, the prior estimate $\vec{q}(t-1)$ is weighted by $1 - \alpha(t)$, while the new outcome $\vec{p}(t)$ is weighted by $\alpha(t)$. If $\alpha(t) = 1/t$, $\vec{q}(t)$ is the mean of the activity ratio since the beginning of time. If $\alpha(t) = 1/t_1$ where t_1 is a constant, a positive integer, $\vec{q}$ is a soft moving average of the activity ratio for the last t_1 time instants. It does not discard everything before the last t_1 instants but assigns them much less weight in the estimation process. The latter case is particularly useful if the data distribution changes over time. However, choosing a small t_1 may not allow $\vec{q}$ to converge to a stable value any time even if the underlying data distribution is stationary. Eq. (9) can be expressed in differential form as:

$$\dot{\vec{q}} = \alpha(\vec{q} - \vec{p}). \tag{10}$$

$\vec{q}$ reaches a stable value when $\dot{\vec{q}} = 0$. Since the value of $\vec{p}$ is dependent on the instantaneous input, $\vec{p}$ is a stochastic variable and in general $\vec{p} \neq \vec{q}$. Hence, the condition for stability is $\alpha \rightarrow 0$ which is satisfied eventually if $\alpha(t) = 1/t$ or for a large constant t_1 if $\alpha(t) = 1/t_1$. The summary activity ratio of each dictionary atom is initialized to $1/k$. It is obvious because at the beginning, all the dictionary atoms are equally likely to be used for reconstruction.

Given the summary activity ratio at any time t, the NLAR at t is computed as $\theta_j(t) = -log(q_j(t))$. Hence the outlier score at t is $s(t) = |\vec{\gamma}(t)|^T\vec{\theta}(t)$. The online version of the RODS algorithm is presented in Algorithm 2. The time complexity of this algorithm is dominated by that of the OMP

algorithm which is $O(2mk\kappa + 2\kappa^2 m + 2\kappa(k+m) + \kappa^3)$ [33] per data point. Thus, the algorithm runs in constant time per input dominated by the maximum of ℓ_0 norm, κ.

Algorithm 2. Online-RODS

Inputs: Streaming data, k, κ, α : learning rate
Output: $s(t)$: outlier score at each time instant t
Compute **D** by solving Eq. (2) using a subset from the data stream
Initialization: $q_j(0) = 1/k \; \forall j$
for $t = 1$ to ∞ **do**
  Draw $\vec{x}(t)$ from **X**
  Compute $\vec{\gamma}(t)$ by solving Eq. (3)
  for $j = 1$ to k **do**
    $p_j(t) \leftarrow \frac{|\gamma_j(t)|}{\sum_{i=1}^{k} |\gamma_i(t)|}$
  end for
  $\vec{q}(t) \leftarrow (1-\alpha)\vec{q}(t-1) + \alpha\vec{p}(t)$
  for $j = 1$ to k **do**
    $\theta_j(t) \leftarrow -log_2(q_j(t))$
  end for
  $\theta_j \leftarrow \theta_j / \sum_{i=1}^{k} \theta_i, \forall j = 1, \ldots, k$
  $s(t) \leftarrow |\vec{\gamma}(t)|^T \vec{\theta}(t)$
  Output $s(t)$
end for

5 Experimental Results

In this section, we provide quantitative and qualitative evaluations of the proposed batch and online versions of the RODS algorithm using several benchmark datasets and compare the performance with various state-of-the-art outlier detection algorithms. We also discuss three real-world case studies: i. saliency detection in images, ii. abnormal event detection in videos iii. change detection in data streams. We demonstrate the superior performance of the batch RODS algorithm on image datasets and that of the online RODS algorithm on streaming datasets.

5.1 Scalability Experiments on Synthetic Data

Synthetic datasets of different sizes and dimensions were generated for the scalability experiments. As discussed in Section 3.3, a data point $\vec{x}$ admits a sparse representation $\vec{\gamma}$ over an overcomplete dictionary **D**. $\vec{\gamma}$ is the projection of $\vec{x}$ in the space spanned by the atoms in **D**.

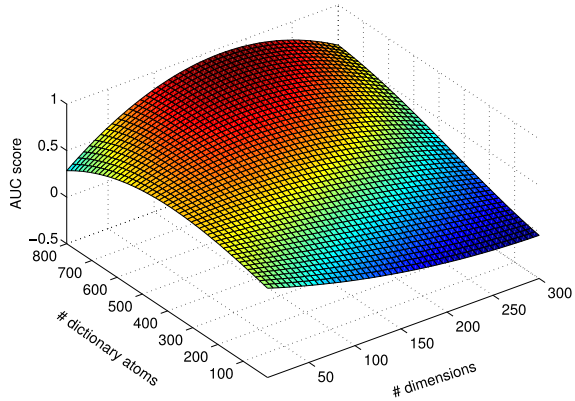
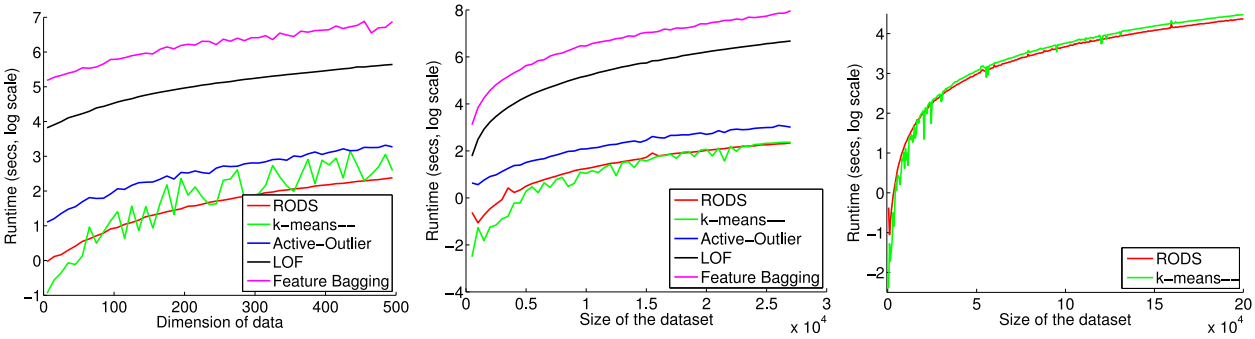


Fig. 3. Parameter sensitivity of the RODS algorithm.

In the proposed framework, a dictionary atom used infrequently to generate the data points has a high NLAR score; a data point generated by such an atom is an outlier. The synthetic dataset is generated from an arbitrary combination of randomly generated dictionary atoms. The set of outliers is generated in the same way, except that it is much smaller in size than the rest of the dataset. We learn k dictionary atoms from this entire data, calculate their NLARs, and compute the outlier score for each data point.

How to select k, the size of the dictionary, and κ, the maximum admissible ℓ_0 norm in the sparse representation, is an open problem. There is no well-defined rule for selecting these parameters and are usually selected via cross-validation. In general, overcomplete dictionary works better in case of sparse coding [23], [24]. We investigated the performance of the RODS algorithm with increasing dictionary size and data dimension. For this experiment, the data dimensions chosen are {5, 50, 100, 150, 200, 250, 300} and dataset size is 10,200 including 200 outliers. The AUC score is calculated with respect to the dictionary size in the range (3, 800). In all the cases, κ is selected as 10 percent of the dictionary size. Fig. 3 shows that the performance of our algorithm improves as the data dimension increases. For each dimension, the performance is optimal for a certain overcomplete range of dictionary size; the overcompleteness is with respect to the data dimension.

Fig. 4 illustrates the scalability of the RODS algorithm in comparison to the state-of-the-art algorithms with respect to



(a) Scalability in data dimensionality m (b) Scalability in dataset size N (c) Scalability in dataset size N

Fig. 4. Scalability of the RODS algorithm in comparison to other algorithms. (a) Comparison of runtimes of the algorithms with respect to the data dimension. As the dimension increases, RODS emerges as the winner. (b) Comparison of runtimes of the algorithms with respect to the dataset size. RODS is significantly better than all competing algorithms except k-means–, to which it is very close. (c) Runtime comparison of RODS with k-means– over extended dataset size. As size increases, RODS emerges as the winner.

TABLE 2
Characteristics of the Benchmark Datasets Used for Performance Evaluation

Datasets	Normal class	Outlier class	# Dimensions	# Samples	# Outliers
Shuttle	1, 4, 5	2, 3, 6, 7	9	43,500	186
Ann-thyroid	3	1	6	3,581	93
WBC	2	4	9	464	20
Pendigits	all digits except 4	50% of digit 4	16	7,104	390
Glass	1, 2, 3, 4	5, 6, 7	9	214	51
Ionosphere	g	b	34	351	126
Arythmia	1, 2, 6, 10, 16	3, 4, 5, 7, 8, 9, 14, 15	279	452	66
KDD-CUP 1999	normal, neptune, smarf	others	38	494,021	8,752

increasing data dimension and dataset size. The algorithms used for comparison include k-means–[13], Local Outlier Factor [12], Active-Outlier [14] and Feature Bagging [15]. Fig. 4a shows the running time required with respect to increasing data dimension for all the algorithms. As the dimension increases, RODS emerges as the winner. k-means– and Active-Outlier perform better than LOF and Feature Bagging. Figs. 4b and 4c show the running time required with respect to increasing dataset size for all the algorithms. The runtime increases exponentially for all algorithms; however, it increases much faster for LOF and Feature Bagging than the others. RODS and k-means– seem close enough to be joint winners, but as the dataset size is increased further, as shown in (c), RODS performs better.

5.2 Experiments on Benchmark Data Sets

We performed our experiments using seven datasets from UCI machine learning repository [40] and the 1999 KDD-CUP dataset.[1]

5.2.1 Experimental Setup

We used seven UCI benchmark datasets: Shuttle, Ann-thyroid, Breast Cancer Wisconsin Diagnostic (WBC), Pendigits, Glass, Ionosphere and Arythmia. In each dataset, the data belonging to the smallest class was considered as the outlier set and the remaining data as the normal set. Details of the normal and outlier sets are presented in Table 2. The normal and outlier definitions are only used during the evaluation of the algorithms and not during the scoring phase since our algorithm is unsupervised.

The *KDD-CUP 1999* dataset [40] is a large-scale data containing a total of 494,021 instances describing the connections of sequences of TCP packets. The goal is to distinguish between normal and bad connections. We considered all of the 38 numerical attributes and discarded the categorical attributes in our experiments. Out of the 23 classes in the dataset, the three classes which contain 98.3 percent of the entire data are considered to be normal and the data from the other twenty classes will be the outliers. For all the datasets, we normalized each dimension of the data by subtracting the mean and dividing by the standard deviation for each data point so that each dimension has zero mean and unit variance.

We compared the RODS algorithm with several state-of-the-art outlier detection algorithms including k-means–[13], Local Outlier Factor [12], Active-Outlier [14] and Feature Bagging [15]. In terms of the implementation, the code for k-means– algorithm was obtained from the authors of [13] and the outlier detection toolbox[2] was used to obtain the results for the remaining algorithms. Among these algorithms, our algorithm, k-means–, LOF and Feature Bagging are unsupervised. Active-Outlier converts the outlier detection to a classification problem and hence supervision is required.

The ranges for the parameter values used for the UCI datasets are as follows. The number of dictionary atoms was set between 10 and 100 and maximum number of ℓ_0 norm was set between 5 and 20. Number of cluster centers was between 10 and 100 for k-means–, number of classifiers was set between 10 and 30 for Active Outlier, size of neighborhood was set between 3 and 25 for LOF and Feature Bagging algorithms. The general guideline is to assign higher values in the range for all parameters for a larger dataset and a lower values for a smaller dataset. The number of selected outliers was always set to the exact number for the k-means– algorithm. For KDD-CUP 1999 dataset, the dictionary size and maximum number of ℓ_0 norm were set to 200 and 20, respectively, for the RODS algorithm. Number of cluster centers were set to 200 for k-means– and number of classifiers was set to 30 for Active Outlier algorithm. It should be noted that most of these algorithms are not extremely sensitive to the particular parameter value that is being set and will produce a reasonable result when the parameters are chosen in the appropriate range depending on the size of each dataset.

5.2.2 Comparison Results

Performance of the algorithms was measured using the Area under the ROC curve (AUC) which is a standard metric used for evaluating the outlier detection algorithms. The ROC curve is a two-dimensional representation drawn as true positive rate (TPR) versus false positive rate (FPR). A perfect model will have a AUC score of 1 and random guessing will have score around 0.5. The AUC values (along with the standard deviations) of the algorithms are shown in Table 3.

The average AUC values for 50 different runs are being reported along with the standard deviations. In the case of LOF, there will not be any standard deviation because it produces the same result for every run and does not depend

1. kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

2. https://bitbucket.org/gokererdogan/outlier-detection-toolbox/

TABLE 3
Comparison of the AUC Scores on the Benchmark Datasets Using Various Outlier Detection Algorithms

Datasets	RODS	k-means– [13]	Active-Outlier [14]	LOF [12]	Feature Bagging [15]
Shuttle	**0.979 (±0.062)**	0.942 (±0.007)	0.949 (±0.081)	0.882	0.833 (±0.058)
Ann-thyroid	0.976 (±0.007)	0.977 (±0.002)	**0.979 (±0.003)**	0.869	0.873 (±0.036)
WBC	**0.978 (±0.002)**	0.973 (±0.002)	0.872 (±0.095)	0.811	0.850 (±0.052)
Pendigits	**0.661 (±0.160)**	0.556 (±0.065)	0.617 (±0.105)	0.492	0.507 (±0.027)
Glass	**0.899 (±0.024)**	0.888 (±0.022)	0.790 (±0.029)	0.799	0.858 (±0.048)
Ionosphere	**0.881 (±0.026)**	0.819 (±0.021)	0.834 (±0.018)	0.838	0.848 (±0.083)
Arythmia	**0.783 (±0.019)**	0.765 (±0.002)	0.553 (±0.017)	0.768	0.769 (±0.013)
KDD-CUP 1999	**0.979 (±0.007)**	0.966 (±0.009)	0.552 (±0.055)	–	–

TABLE 4
Comparison of the Time Taken (in Seconds) for Various Outlier Detection Algorithms to Obtain the Results on the Benchmark Datasets

Datasets	RODS	k-means– [13]	Active-Outlier [14]	LOF [12]	Feature Bagging [15]
Shuttle	**2.43**	9.25	9.07	775.55	4,369.94
Ann-thyroid	**0.59**	0.62	1.53	20.31	104.23
WBC	**0.04**	0.29	0.55	4.18	29.57
Pendigits	**0.54**	1.13	5.14	55.41	201.55
Glass	**0.02**	0.15	0.48	1.06	4.09
Ionosphere	**0.03**	0.31	0.93	1.89	7.44
Arythmia	**1.51**	1.53	1.76	4.89	5.10
KDD-CUP 1999	**359.55**	2,669.92	485.05	–	–

on the initialization unlike other algorithms. It can be clearly seen that the RODS outperforms other algorithms in almost all of the datasets (except in the Ann-thyroid dataset where RODS gives a competitive result).

In addition, the RODS also outperforms all other algorithms with respect to the computation time taken. Table 4 shows the average time taken by the algorithms on different datasets. All the experiments were performed using Matlab version R2013a on an Intel Core i7-2600 3.40 GHz CPU with 32GB RAM. Due to the quadratic complexity of the LOF and Feature Bagging algorithms, the memory requirements in the specified system were not sufficient for the KDD-CUP 1999 dataset. It can be observed that the proposed RODS algorithm is much faster and efficient compared to the other competing algorithms.

5.3 Saliency Detection in Images

To demonstrate the effectiveness of the proposed algorithm on high-dimensional data, we evaluate the performance on a real-world problem of saliency detection in image data.

5.3.1 Saliency Detection

An important goal of saliency detection in images is to understand the attentional mechanism of human visual system. In images, a salient region (or object) is one that stands out due to some property that occurs infrequently among the regions (or objects) in its neighboring spatial locations [41], [42]. Thus, saliency detection can be formulated using a reconstruction process and interpreted as an outlier detection problem [43]. We used Toronto dataset,[3] which is one of the most widely used datasets for performance evaluation in saliency detection algorithms for image data [44]. This dataset consists of 120 color images with a resolution of 511×681 pixels. It has both indoor and outdoor environments. The images were presented in a random order to 20 subjects for four seconds each, with a mask between each pair of images.

5.3.2 Performance Evaluation

Using the Toronto dataset, the saliency maps generated from the RODS algorithm are compared with those from k-means–, LOF, Feature Bagging and two other well known saliency detection algorithms including Itti[4] [45] and GBVS[5] [46]. It should be noted that the Active-Outlier algorithm cannot be used for this problem since it is supervised and hence was excluded from the comparison.

As part of data preprocessing, each image was downsampled to 60×80. Then each image was divided into 8×8 overlapping square patches at each pixel and converted to a column vector of size 192 (being a color image, it has three channels: R, G, B). These vectors collected from one image creates a dataset ($\mathbf{X} \in \mathbb{R}^{192 \times 3,869}$) for the RODS, k-means–, LOF and Feature Bagging. Then the task is to assign an outlier score to each patch in the image with respect to all other patches in that image. The patches with high outlier score will be more salient than the others. These outlier scores are finally arranged in a 2D grid for getting the saliency map. Saliency map is a gray scale representation where bright regions represent salient regions and dark regions represent non-salient regions. Also, the level of brightness indicates the level of saliency.

Fig. 5 shows some of the saliency maps generated from the models in comparison to human eye-tracking results. We can see that the saliency maps produced by the RODS

3. http://www-sop.inria.fr/members/Neil.Bruce/

4. ilab.usc.edu/toolkit/

5. www.vision.caltech.edu/∼harel/

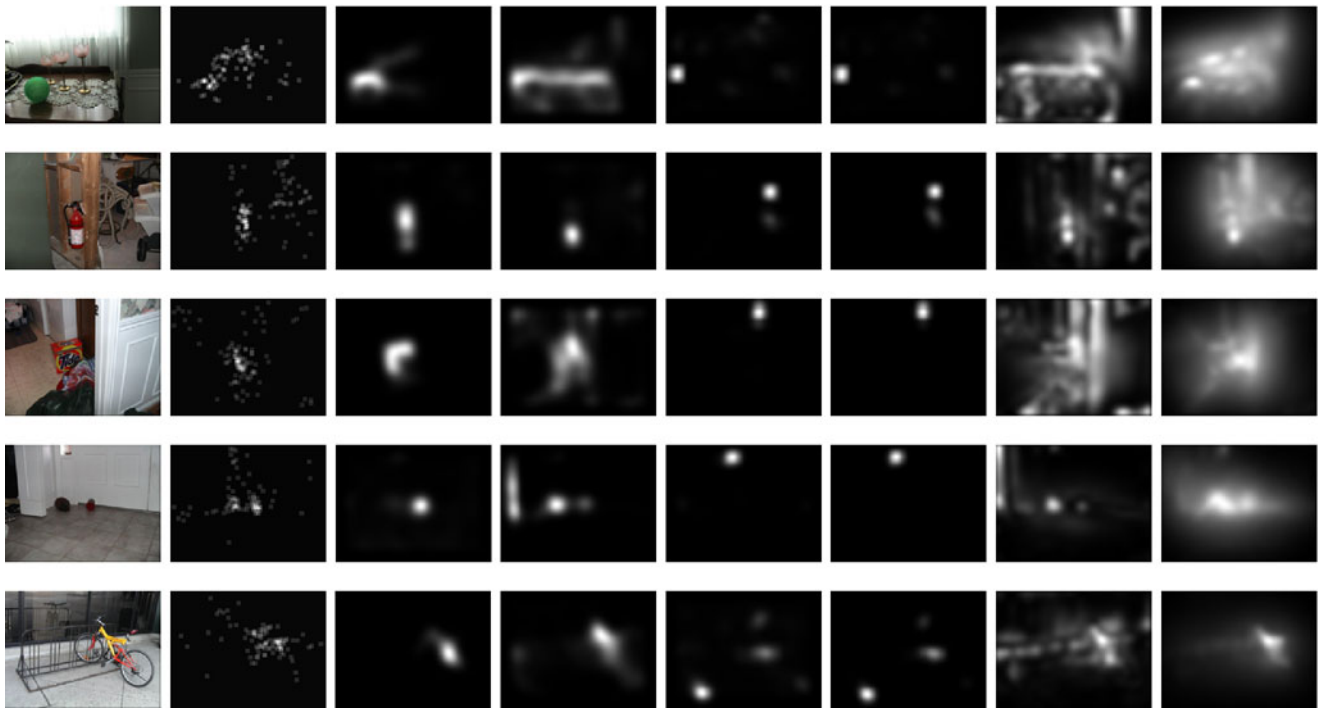


Fig. 5. Sample images from the Toronto dataset, the corresponding human fixation, and the saliency maps generated using RODS and various other state-of-the art algorithms.

algorithm are more accurate and less noisy compared to other algorithms which either produce imprecise and inaccurate results or contain a lot of noise. The AUC scores have also been used to evaluate the performance. Table 5 compares the mean AUC scores (averaged over all the images in the Toronto dataset) along with the average time taken to obtain the results for each algorithm. It can be seen that RODS not only produces the best AUC scores but also achieves it much faster than the competing algorithms.

5.4 Abnormal Event Detection in Video Streams

Here we demonstrate the performance of the online version of the RODS algorithm in the context of video streams. We used the popular UCSD dataset [47]. For our experiments, we used one of the datasets, namely, Ped1, which contains 34 training and 36 testing video clips, each with pixel resolution of 158×238. The training sets have all normal events and contain only pedestrians on the pedestrian walkway. Each testing video clip contains at least one abnormal event with the presence of bicyclists, skaters, small cars, and people in wheelchairs. These are considered to be outliers based on the context of the scene because they are rare events on the pedestrian walkway. The goal here is to detect these events in the testing dataset in an unsupervised setting.

TABLE 5
Comparison of Different Algorithms on the Toronto (Image) Dataset Using Mean AUC Values and Time Taken

Algorithm	Mean AUC	Average time taken (seconds)
RODS	**0.653**	**0.73**
k-means– [13]	0.624	47.09
LOF [12]	0.534	43.51
Feature bagging [15]	0.557	125.73
Itti [45]	0.635	0.78
GBVS [46]	0.634	3.65

5.4.1 Evaluation Metrics

The following two evaluation metrics which are widely used for measuring the accuracy of abnormal event detection in video streams are used for our comparisons [48].

- *Frame-level evaluation.* An algorithm determines the frames that contain abnormal events. The result is compared to the frame-level ground truth annotation of each frame and the number of true and false positive frames are calculated.
- *Pixel-level evaluation.* An algorithm determines the pixels that are related to abnormal events. If at least 40 percent of the truly anomalous pixels are detected for an abnormal frame, it is considered as an accurate detection.

For both the cases, true positive rate and false positive rate are calculated as follows:

$$\begin{aligned} \text{TPR} &= \frac{\text{number of true positive frames}}{\text{number of positive frames}}, \\ \text{FPR} &= \frac{\text{number of false positive frames}}{\text{number of negative frames}}. \end{aligned} \tag{11}$$

TPR and FPR is calculated for different threshold values. Then, ROC curve is drawn as the TPR versus FPR. Finally, the performance is summarized using the *equal error rate* (EER) which is the ratio of misclassified frames at which FPR is equal to 1-TPR in the ROC curve, for both frame-level and pixel-level criteria. A low EER value indicates a better performance.

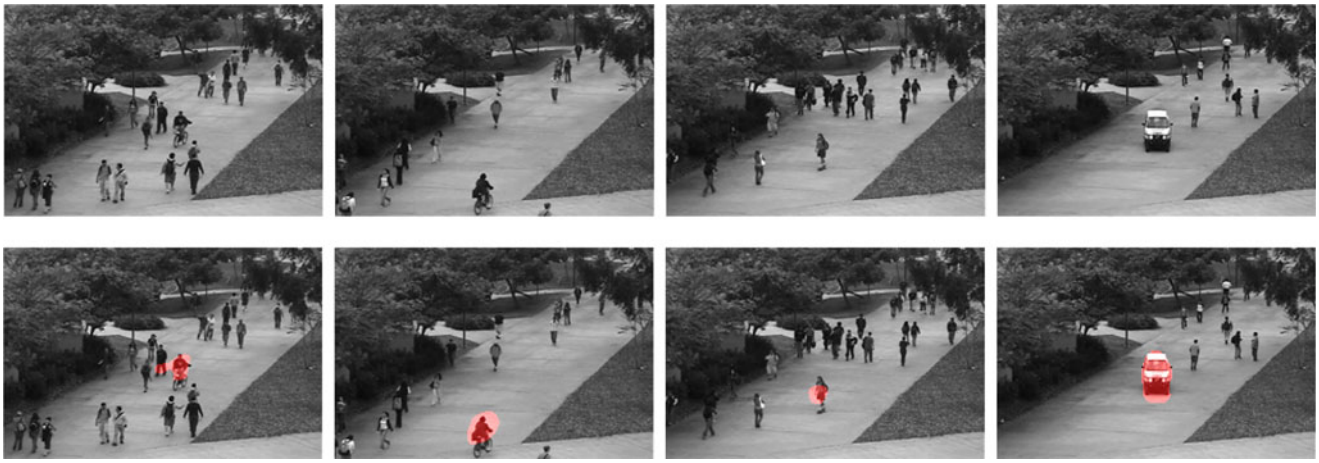


Fig. 6. Abnormal frames from UCSD Ped1 (first row) and the detection result from the RODS algorithm (second row). The bikers, skater, and car were detected as anomalous patterns (highlighted in red, best viewed in color).

Fig. 7. Examples of video frames with relatively minor changes from the chosen dataset. (From left to right) Top row shows frames 48, 49, 50, 51, while middle row shows frames 78, 79, 80, 81. Minor changes occur at frames 51 and 81. Bottom row shows frames 64, 65, 66, 67 where green color occurs due to camera artifact.

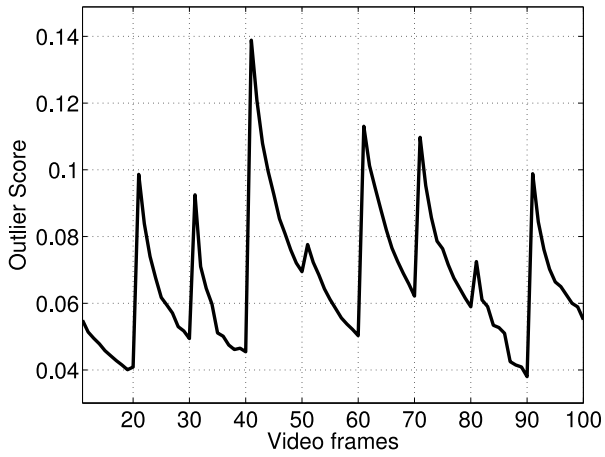
5.4.2 Performance Evaluation

The performance of RODS was compared with that of a number of state-of-the art abnormal event detection algorithms. Since most of the previously mentioned outlier detection algorithms are offline (only batch versions), it was not appropriate to compare with them, as the dataset becomes very large and high-dimensional for streaming videos. Hence, for comparison of performance evaluation, the following algorithms which are widely used for anomaly detection in streaming videos have been used: Sparse [48], MPPCA [49], Social force [50] and LMH [51].

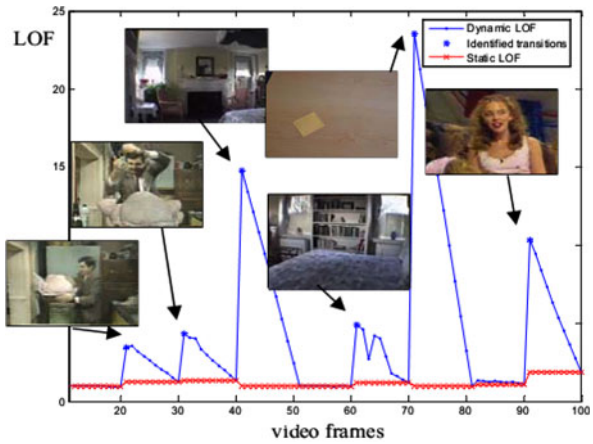
TABLE 6
Performance of Outlier Detection on the USCD Ped1 Datasets

Algorithm	Frame-level EER	Pixel-level EER
RODS	19.2	**31.7**
Sparse [48]	**19**	54
MPPCA [49]	35.6	76.8
Social force [50]	36.5	59.1
LMH [51]	38.9	67.4

The proposed framework uses local spatiotemporal volumes around the detected interest points in each clip as an input representation. Here, we adopt a spatiotemporal interest point detector [52] to extract cuboids which contain the spatiotemporally windowed pixels. Before learning the dictionary, each cuboid is converted to a vector and normalized to have a unit ℓ_2 norm. The inputs were cuboids of size $13 \times 13 \times 10$ pixels and 400 dictionary atoms were used.



(a)



(b)

Fig. 8. (a) Result of applying online RODS algorithm on video sequences from test movie. (b) Result of applying incremental (blue asterisks) and static periodic (red x) LOF algorithm (k=10) on video sequences from test movie. For the static LOF algorithm, LOF values that are shown for each frame t are computed when the latest data record is inserted. (b) is reproduced from [34] with permission.

The RODS algorithm could detect bikers, skaters, small cars as outliers or abnormal events. Some of the snapshots[6] of the results are shown in Fig. 6. Table 6 shows the quantitative performance comparison of the RODS model with other existing models. It can be clearly seen that RODS outperforms other algorithms.

5.5 Change Detection in Streaming Data

We have selected a dataset composed of 100 video frames[7] for the experiments on change detection. As in [34], the goal is to detect sudden changes in the video frames. Major changes occur in frames 21, 31, 41, 61, 71 and 91, while relatively minor changes occur in frames 51 and 81. The causes of the major changes include appearance of a new object (in frame 21), camera zooms into the objects (frame 31), and completely new content (frames 41, 61, 71, 91) [34]. Minor changes are caused by faster movement of the camera (see Fig. 7). We have used GIST features [53] to represent each video frame. As shown in Fig. 8, the proposed Online-RODS algorithm can detect all the major changes similar to Incremental LOF [34]. While Incremental LOF is completely oblivious to these minor changes, the RODS algorithm produces small increase in anomaly scores for these frames. Thus, RODS is more sensitive to changes than the Incremental LOF and produces a response proportional to the degree of change. However, due to the use of GIST features which represents global information, RODS is not susceptible to local noise as evident from its performance in frame 66 where it did not elicit a response but the Incremental LOF did (see Fig. 7).

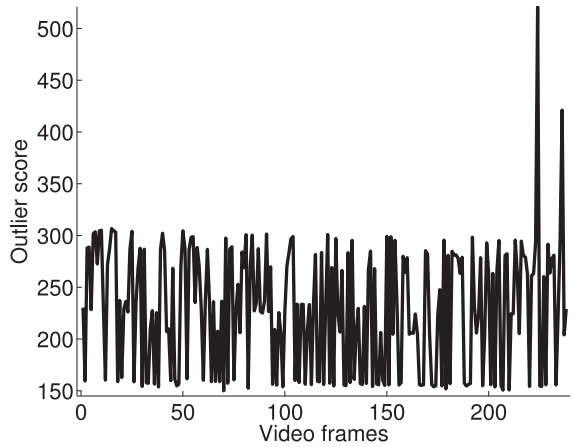


Fig. 9. Outlier score for each trajectory using Online-RODS algorithm.

The performance of our Online-RODS algorithm in detection of abnormal events is tested on a dataset of video motion trajectories.[8] The dataset consisted of 239 trajectories with only two trajectories (225, 237) identified as abnormal. Each trajectory is of 15 dimensions. The outlier score of each trajectory using Online-RODS algorithm is shown in Fig. 9. It can be seen that both the trajectories 225 and 237 have significantly higher outlier score than other trajectories which is very similar to the results produced by Incremental LOF [34].

6 Conclusions

We proposed a new definition of outlier whereby, given a dictionary of atoms learned using the sparse coding objective, the outlierness of a data point is a function of the frequency of each atom in reconstructing all data points (a.k.a. NLAR) and the strength by which it is used in reconstructing the current point. The RODS algorithm was developed which operates in time linear in size of the dataset. We also presented an efficient online extension of the algorithm which updates the NLARs of the atoms and computes the outlier scores for the data points in the streaming context. Compared to several state-of-the-art outlier detection methods, the RODS algorithm performs better on various benchmark datasets and real-world problems using standard evaluation metrics.

Acknowledgments

Bonny Banerjee was supported by the US National Science Foundation (NSF) CISE grant 1231620. Chandan K. Reddy was supported by NSF grants IIS-1231742 and IIS-1527827. Jayanta K. Dutta is the corresponding author.

6. A video demo of the results is made available at sites.google.com/site/jayantadutta05/tkdedemo

7. www.cis.temple.edu/~latecki/TestData/SimTest.zip

8. www.cs.umn.edu/~aleks/inclof

REFERENCES

[1] R. J. Bolton and D. J. Hand, "Unsupervised profiling methods for fraud detection," in *Proc.Credit Scoring Credit Control VII*, 2001, pp. 235–255.

[2] A. Lazarevic, L. Ertöz, V. Kumar, A. Ozgur, and J. Srivastava, "A comparative study of anomaly detection schemes in network intrusion detection," in *Proc. 3rd SIAM Int. Conf. Data Mining*, 2003, pp. 25–36.

[3] K. I. Penny and I. T. Jolliffe, "A comparison of multivariate outlier detection methods for clinical laboratory safety data," *J. Roy. Statistical Soc.: Series D (The Statistician)*, vol. 50, no. 3, pp. 295–307, 2001.

[4] Z. He, X. Xu, J. Z. Huang, and S. Deng, "Mining class outliers: concepts, algorithms and applications in CRM," *Expert Syst. Appl.*, vol. 27, no. 4, pp. 681–697, 2004.

[5] M. A. Rubin and A. M. Chinnaiyan, "Bioinformatics approach leads to the discovery of the TMPRSS2: ETS gene fusion in prostate cancer," *Laboratory Investigation*, vol. 86, no. 11, pp. 1099–1102, 2006.

[6] G. R. Abecasis, S. S. Cherny, W. Cookson, and L. R. Cardon, "GRR: Graphical representation of relationship errors," *Bioinformatics*, vol. 17, no. 8, pp. 742–743, 2001.

[7] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surveys*, vol. 41, no. 3, p. 15, 2009.

[8] C. C. Aggarwal, *Outlier Analysis*. New York, NY, USA: Springer, 2013.

[9] P. J. Rousseeuw and M. Hubert, "Robust statistics for outlier detection," *Wiley Interdisciplinary Rev.: Data Mining Knowl. Discovery*, vol. 1, no. 1, pp. 73–79, 2011.

[10] E. M. Knox and R. T. Ng, "Algorithms for mining distance based outliers in large datasets," in *Proc. Int. Conf. Very Large Data Bases*, 1998, pp. 392–403.

[11] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 427–438, 2000.

[12] M. M. Breunig, H. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, 2000.

[13] S. Chawla and A. Gionis, "k-means–: A unified approach to clustering and outlier detection," in *Proc. SIAM Int. Conf. Data Mining*, 2013, pp. 189–197.

[14] N. Abe, B. Zadrozny, and J. Langford, "Outlier detection by active learning," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 504–509.

[15] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2005, pp. 157–166.

[16] C. C. Aggarwal, "Outlier ensembles: Position paper," *ACM SIGKDD Explorations Newslett.*, vol. 14, no. 2, pp. 49–58, 2013.

[17] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," *ACM SIGMOD Rec.*, vol. 30, no. 2, 2001, pp. 37–46.

[18] A. Zimek, E. Schubert, and H. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Statistical Anal. Data Mining*, vol. 5, no. 5, pp. 363–387, 2012.

[19] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, "LOCI: Fast outlier detection using the local correlation integral," in *Proc. 19th Int. Conf. Data Eng.*, 2003, pp. 315–326.

[20] P. Filzmoser, R. Maronna, and M. Werner, "Outlier identification in high dimensions," *Comput. Statist. Data Anal.*, vol. 52, no. 3, pp. 1694–1711, 2008.

[21] A. Adler, M. Elad, Y. Hel-Or, and E. Rivlin, "Sparse coding with anomaly detection," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, 2013, pp. 1–6.

[22] B. Zhao, L. Fei-Fei, and E. P. Xing, "Online detection of unusual events in videos via dynamic sparse coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 3313–3320.

[23] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An Algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[24] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, 2010.

[25] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.* 2005, vol. 2, pp. 860–867.

[26] G. Peyré, "Sparse modeling of textures," *J. Math. Imag. Vis.*, vol. 34, no. 1, pp. 17–31, 2009.

[27] J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce, "Discriminative sparse image models for class-specific edge detection and image interpretation," in *Proc. 10th European Conf. Comput. Vis.*, 2008, pp. 43–56.

[28] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

[29] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng, "Shift-invariance sparse coding for audio classification," *arXiv preprint arXiv:1206.5241*, 2012.

[30] C. Févotte, N. Bertin, and J. Durrieu, "Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis," *J. Neural Comput.*, vol. 21, no. 3, pp. 793–830, 2009.

[31] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 1794–1801.

[32] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 3501–3508.

[33] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit," *CS Technion*, p. 40, 2008.

[34] D. Pokrajac, A. Lazarevic, and L. J. Latecki, "Incremental local outlier detection for data streams," in *Proc. IEEE Symp. Comput. Intell. Data Mining*, 2007, pp. 504–515.

[35] F. Angiulli and F. Fassetti, "Distance-based outlier queries in data streams: the novel task and algorithms," *J. Data Mining Knowl. Discovery*, vol. 20, no. 2, pp. 290–324, 2010.

[36] D. Yang, E. A. Rundensteiner, and M. O. Ward, "Neighbor-based pattern detection for windows over streaming data," in *Proc. 12th Int. Conf. Extending Database Technol.: Adv. Database Technol.*, 2009, pp. 529–540.

[37] M. Kontaki, A. Gounaris, A. N. Papadopoulos, K. Tsichlas, and Y. Manolopoulos, "Continuous monitoring of distance-based outliers over data streams," in *Proc. IEEE 27th Int. Conf. Data Eng.*, 2011, pp. 135–146.

[38] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[39] A. Coates and A. Y. Ng, "The importance of encoding versus training with sparse coding and vector quantization," in *Proc. Intl. Conf. Mach. Learn.*, 2011, pp. 921–928.

[40] K. Bache and M. Lichman. (2013). UCI machine learning repository [Online]. Available: http://archive.ics.uci.edu/ml

[41] J. K. Dutta and B. Banerjee, "Online detection of abnormal events using incremental coding length," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 3755–3761.

[42] B. Banerjee and J. K. Dutta, "SELP: A general-purpose framework for learning the norms from saliencies in spatiotemporal data," *Neurocomputing*, vol. 138, pp. 41–60, 2014.

[43] A. Borji and L. Itti, "Exploiting local and global patch rarities for saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* 2012, pp. 478–485.

[44] N. D. B. Bruce and J. K. Tsotsos, "Saliency based on information maximization," in *Proc. Adv. Neural Process. Syst.* , 2006, pp. 155–162.

[45] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[46] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proc. Adv. Neural Process. Syst.*, 2006, pp. 545–552.

[47] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 18–32, Jan. 2014.

[48] Y. Cong, J. Yuan, and J. Liu, "Abnormal event detection in crowded scenes using sparse representation," *Pattern Recog.*, vol. 46, no. 7, pp. 1851–1864, 2013.

[49] J. Kim and K. Grauman, "Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 2921–2928.

[50] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 935–942.

[51] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 555–560, Mar. 2008.
[52] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc. IEEE Int. Workshop Visual Surveillance Perform. Eval. Tracking Surveillance*, 2005, pp. 65–72.
[53] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

Jayanta K. Dutta received the BSc degree in computer science and engineering from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh, in 2011 and the MS degree in electrical and computer engineering from the University of Memphis (UM), Memphis, in 2013. He is currently working toward the PhD degree in electrical and computer engineering at UM. His research interest includes data mining, machine learning, and neural networks.

Bonny Banerjee received the MS degree in electrical engineering and the PhD degree in computer science and engineering, both from The Ohio State University, Columbus. He is currently a dual-appointed assistant professor in the Department of Electrical and Computer Engineering and the Research-Intensive Institute for Intelligent Systems at the University of Memphis (UM), Memphis, where he directs the Computational Intelligence Laboratory with research that focuses on brain-inspired data mining, machine learning, and artificial intelligence. Prior to joining UM, he led the research in a start-up company which resulted in 11 US and international patent applications within three years, besides attracting substantial investor funding and media coverage. He has published in the top journals in the field. He is serving as the special sessions co-chair and a program committee member of the inaugural INNS Conference on Big Data 2015. He is the PI on a current US National Science Foundation grant. He is a member of the IEEE.

Chandan K. Reddy received the MS degree from the Michigan State University and the PhD degree from Cornell University. He is an associate professor in the Department of Computer Science, Wayne State University. His primary research interests are in the areas of data mining and machine learning with applications to healthcare, bioinformatics, and social network analysis. His research is funded by the US National Science Foundation (NSF), NIH, DOT, and Susan G. Komen for the Cure Foundation. He has published more than 50 peer-reviewed articles in leading conferences and journals. He received the Best Application Paper Award at the ACM SIGKDD conference in 2010 and was a finalist of the INFORMS Franz Edelman Award Competition in 2011. He is a senior member of the IEEE and a life member of the ACM.