Anomaly detection in information streams without prior domain knowledge

M. S. Beigi
S.-F. Chang
S. Ebadollahi
D. C. Verma

A key goal of information analytics is to identify patterns of anomalous behavior. Such identification of anomalies is required in a variety of applications such as systems management, sensor networks, and security. However, most of the current state of the art on anomaly detection relies on using a predefined knowledge base. This knowledge base may consist of a predefined set of policies and rules, a set of templates representing predefined patterns in the data, or a description of events that constitutes anomalous behavior. When used in practice, a significant limitation of information analytics is the effort that goes into defining and creating the predefined knowledge base and the need to have prior information about the domain. In this paper, we present an approach that can identify anomalies in the information stream without requiring any prior domain knowledge. The proposed approach simultaneously monitors and analyzes the data stream at multiple temporal scales and learns the evolution of normal behavior over time in each time scale. The proposed approach is not sensitive to the choice of the distance metric and hence is applicable in various domains and applications. We have studied the effectiveness of the approach using different data sets.

Introduction

Anomaly or outlier detection is of great importance when analyzing streaming data in many applications such as systems and network management for detecting faults and performance problems; sensor networks for detecting anomalous behaviors and activities; and security for detecting frauds and intrusions. As processors become faster and less expensive, more and more streaming data can be captured and made available for analysis. To manage the overload of the streaming data, one needs to create mechanisms for identifying only those time intervals that are informative and worthy of further high-level analysis by either machine or the human observer. For example, when analyzing sensor network data, one must segment the temporal data stream and identify the potential event bearing candidates for further analysis. These segments of data may be identified by an outlier or anomaly detector that scans the data streams at a high rate and outputs only the segments that need further processing.

A key challenge in anomaly detection is defining what is normal and identifying the boundary between normal behavior and the outlier or abnormal behavior. Use of rule- or template-based techniques for identifying normal or abnormal behavior is subject to the application domain and very much dependent on domain expertise. The nearest-neighbor or distance-based approaches rely heavily on the choice of the distance metric being used, which is highly dependent on the data type and application. It is almost impossible to choose a distance metric that performs well in all types of applications. Another challenge is that normal behavior and outliers frequently change over time. Therefore, the system needs to learn the evolution of normal behavior over time, yet another challenge is that the different types of anomalies may unfold at different temporal scales based on the application domain and the nature of the anomaly. Therefore, if the analysis is done at a coarse temporal scale, anomalies that span a short length of time might be missed. On the other hand, if the analysis is performed at a small temporal scale, the long-spanning anomalies may not be detected. Another

Digital Object Identifier: 10.1147/JRD.2011.2163280

challenge in anomaly detection is being able to process large volumes of data in real time. Hence, the anomaly detection algorithm needs to be fast and reactive in a timely manner.

In order to address the limitations of traditional knowledge-base-driven systems, we propose a data-driven approach that does not require any prior knowledge of normal behavior. The key observation that we make is that abnormality manifests itself in multiple temporal scales. Therefore, simultaneous examination of data at different time scales allows one to identify abnormal behavior in a nonparametric way, i.e., develop an algorithm that does not require any parameters that need to be defined manually. Nonparametric statistical approaches, which are also known as "distribution free," do not assume that the data belongs to any particular distribution and therefore do not require that the data fit any predefined distribution. They require less restrictive assumptions about the data compared to the parametric approaches, which assume the data comes from a type of probability distribution and makes inference about the parameters of the distribution.

In our approach, we define an outlier to be an uncommon segment of the data in time. We use a nonparametric statistical approach to represent the statistical behavior of the temporal data while monitoring the incoming streaming data at different temporal scales and looking for discrepancies between the local statistical behavior of the signal and its historical behavior. The approach learns the evolution of the normal behavior over time. The statistical representation of the data is optimized for providing the highest accuracy without imposing high computational demand. In addition, we show that the approach is distance-metric agnostic, which makes it applicable to all types of data and settings.

We have applied the proposed approach to a data set obtained from sensing various human activities using a set of multimodal sensors. In addition, we have used sensor data generated by motes measuring temperature, humidity, and light in the Intel Berkeley Research Laboratory [1], as well as time series data containing passenger interarrival times at a bus stop [2]. The results reported in this paper are focused on a single sensor data stream. The results of our experiments have shown that the proposed approach is capable of spotting the event-bearing time segments of the data in different resolutions depending on the time length of the events.

An overview of anomaly detection techniques

Anomaly or outlier detection refers to detecting patterns in a given data set that do not conform to a *normal* or expected behavior. The term *normal* refers to a baseline that may be known *a priori* or learned through time. The presence of outliers in a data set may be due to noise or unwanted system behavior. Noise may be caused by measurement error or communication error, but the nature of unwanted system behavior is application dependent. For example, in network or system performance monitoring, it may be link or server failures, and in security, it may be denial of service attacks or intrusion detection. In accounting and transaction monitoring, it may be due to fraud, whereas in surveillance applications, it may be due to abnormal activity.

The approaches used to perform anomaly detection depend on the application and the nature of the data. The broad categories of approaches are given as follows: the rule-based, pattern-matching, model-based, similarity-based, and statistical approaches.

Modeling approaches

Modeling approaches require prior knowledge of the application domain and are given as follows:

- *Rule-based* approaches use a database containing the rules governing the behavior of the faulty system or an abnormal behavior to determine whether an anomaly has occurred. The anomaly or fault is determined by monitoring a series of symptoms that are predefined by the rules. Rule-based methods rely heavily on human expertise and are not adaptive to new and evolving environments. Case-based reasoning is an extension of the rule-based approach, which uses the history of faults to make decisions and, because it can build new rules, is more adaptive to evolving environments. However, it relies heavily on having past information and is not efficient in computation time and complexity.
- *Pattern matching or profiling* uses online learning to build profiles or patterns for normal behavior, and deviations from them are considered anomalies. These methods do not scale well for evolving behaviors over time.
- *Model-based* approaches use different types of models to characterize the normal behavior of the monitored system. The most popular predictive model used is unsupervised support vector machines. The model-based approaches need training data in order to build the model.
- *Parametric statistical* approaches perform model fitting and assume a known underlying distribution [3] of the data or are based on statistical estimation of the distribution parameters [4]. These methods flag as outliers those observations that deviate from the model assumptions. However, these methods rely heavily on prior knowledge of the data distribution and are not suitable for arbitrary or new and unknown data sets.

Data-mining approaches

Data-mining approaches do not require any prior knowledge of the application domain and include the following:

- *Distance*- or *similarity*-based approaches [5] detect outliers by computing distances among observations or

points in a multidimensional space. These methods do not scale well and are not useful in very high dimensional data spaces since data points are sparse [6] and finding outliers is non-obvious. In addition, finding a good distance metric and a good threshold is not easy and obvious. Similarity-based approaches cannot be applied to stochastic data. *Depth*-based methods identify the neighborhood of an object based on spatial relationships and consider the proximity factor to decide whether an object is an outlier with respect to neighboring objects or to a cluster. Similarity-based approaches have better computational efficiency than depth-based approaches.

- *Nonparametric statistical* approaches [7, 8] do not assume prior knowledge of the data distributions. *Nonparametric density*-based outlier detection methods are popular and seem to be more promising than other approaches since they are efficient to compute in a streaming environment. They are suitable for unknown environments and can easily be combined when there are multiple dimensions.

Many of the proposed outlier detection approaches as referred to in [6] require prior knowledge about the application. They choose the distance metric heuristically by performing empirical testing and selecting the one that performs best for the specific application.

Proposed approach

Our approach is based on the observation that the statistical characteristics of the data do not have abrupt changes if there are no outliers or events of interest; that is, statistically, the data follows a constant *baseline* model or evolves slowly as time progresses. However, when an event or anomaly occurs, it manifests itself through perturbing such statistical behavior and causing it to shift characteristics from what was seen before. The issues then are how to capture and represent the statistical behavior of the stream at different times and at what temporal scale to look for such perturbations. We have validated this observation in the examination of various data samples.

Based on this observation, our algorithm to detect anomalies is a nonparametric distribution-based approach using a sliding time window for which the distribution of the observed data is determined. This distribution is compared to a *baseline*, which represents the expected or historic behavior. The baseline distribution is calculated using the data values seen in the past. There are two methods for determining the baseline. In the first method, the baseline uses a growing window starting at some time t_0. In this case, the start of the historic data is fixed and is set to the beginning of a new episode. For example, t_0 can be the beginning of a new season or the first day of the week. In the second method, the baseline uses a shifting window in the past, which means t_0 moves as time goes by. In this case, the shifting window used to calculate the baseline represents only recent observations of the environment.

We use histograms to approximate the underlying statistical characteristics of the data. Histograms have an advantage of being simple and fast to compute as opposed to an alternative such as the kernel density estimator. In addition, histograms can be easily updated in streaming data environments as data is being read over time.

Different types of anomalies unravel at different temporal scales because of the nature of the anomaly. Hence, we simultaneously analyze the data at multiple temporal scales. This means that we use multiple window sizes for the sliding window. This is an alternative method to using wavelets [9] or scale space filters [10] for analyzing the data at multiple scales and using fixed-sized windows. In a previous paper [11], we showed that, if the streaming data is analyzed at multiple scales, different outliers may be detected depending on the scale at which they happen.

As the current statistical behavior of the data is compared with the baseline, a distance vector is generated, consisting of one distance value every time the moving window is shifted. The distance vector is then passed through a maxima detector that determines the outlier points for the particular time scale. To detect the outlier segments, we find the maxima points (i.e., outliers) within each calculated distance vector. We have used three different methods for detecting the maxima points. Maximum point detection is performed at each temporal scale.

1. *Constant threshold*—The first method uses a constant threshold T. Each value in the distance vector having a value greater than T is marked as an outlier.
2. *Top percentage*—The second method selects the largest $N\%$ of the values for a window of time. This method is useful for domains where an approximate percentage of anomalies is expected.
3. *Maximum neighbor*—The third method for detecting the maxima points compares each point in the distance vector to its neighbors on both sides. We let V be the number of neighbors on each side. If the value is larger than all the $2V$ neighbors, then the point is considered a maxima point, hence detecting the local maxima points.

The appropriate maxima detection method may be chosen at deployment time in order to better tune the system. Note that, as mentioned earlier, the proposed approach may be used to reduce the data overload and to extract the segments of data that may need further analysis using other machine-learning methods. Therefore, the choice of the maxima detection method may also be affected by the amount of processing power one has and the rate of the incoming data.

The input data may be the raw data or derived features from the raw data stream. The features are decided at design time and are based on the particular environments and data streams being monitored. For audio signals, the most common types of low-level features are the mel-frequency cepstral coefficients [12], which represent the spectral envelope of the audio signal. The features may be elementary or at higher levels such as concepts, the number of human voices detected [13], etc. They may be the extracted semantic concepts, which may be the type of objects detected, being car, airplane, or human. The features may also be aggregated results such as the sum, maximum, minimum, or algebraic functions for computing coarser granularity such as average, standard deviation, and variance. Keeping only the high-level and the aggregated features helps overcome the information overload and the storage required for buffering the streaming data.

Detection accuracy

In this section, we describe how to maximize the detection accuracy by minimizing the error in generating the distribution of the data when there is no prior knowledge of the distribution type. One fundamental problem in generating a faithful representation (i.e., distribution) of a given data set is choosing the right smoothing parameter h or, in the case of histograms, the bin width. If the bin width is too large, it averages out the details of the distribution. On the other hand, if the bin width is too small, the distribution just tells what the observation values are, which does not give a good representation of the data distribution. Choosing h to be very large will result in small variance but large bias, which is referred to as oversmoothing. On the other hand, choosing small values for h will result in small bias but a large variance. This is referred to as undersmoothing. The bias and variance may be controlled simultaneously by choosing an intermediate value of the bin width and allowing the bin width to slowly decrease as the sample size increases. The method in calculating the optimum bin width must be computationally simple in order to maintain the advantage of simplicity of using histograms.

The asymptotic mean integrated squared error $(AMISE)$ represents the error in estimating a density function by a histogram. It uses the L^2 distance for computing the distance between the actual function and its estimate and is given by [14]

$$AMISE(h) = \frac{1}{nh} + \frac{1}{12}h^2 R(f'), \tag{1}$$

where n is the number of data samples; h is the smoothing parameter (i.e., bin width); and $R(f')$ is the statistical roughness of the first derivative of density function f. The only assumption made on the density function is that it has an absolutely continuous derivative and a square-integrable first derivative.

The optimum bin width that minimizes the $AMISE$ error is [14]

$$h^* = \left[\frac{6}{R(f')}\right]^{1/3} n^{-1/3}. \tag{2}$$

Equation (2) shows that the optimum value for h depends on the number of data points n, as well as the roughness of the density function, which is data dependent. The value for n is known, but the roughness of the density function depends on the shape of the function and is not known *a priori*. By definition, the statistical roughness is given by

$$R(\phi) \equiv \int \phi(x)^2 dx,$$

which can be computed using a biased estimator given by the following [15]:

$$\hat{R}(f') = \frac{1}{n^2 h^3}\sum_k (v_{k+1} - v_k)^2 - \frac{2}{nh^3}, \tag{3}$$

where v_k is the number of data points in the kth bin.

Substituting (3) into (1) gives the biased cross-validation (BCV) [16] estimate of the $AMISE(h)$, which is

$$BCV(h) = \frac{5}{6nh} + \frac{1}{12n^2 h}\sum_k (v_{k+1} - v_k)^2. \tag{4}$$

We now make a simple assumption that the range of measurement values and the quantization level of the measurements is known.

As we saw in (2), the optimum bin width is inversely proportional to the statistical roughness $R(f')$. Since the density function f can be arbitrarily rough, there is no lower bound on bin width h. However, if the sampled data values are bounded between two known values $a_{\min}$ and $a_{\max}$, or in other words, the density function is zero outside of $(a_{\min}, a_{\max})$, there is a lower bound on the roughness $R(f')$ that results in an upper bound on bin width h [15]

$$h^* < \frac{a_{\max} - a_{\min}}{\sqrt[3]{2n}}. \tag{5}$$

There is also a lower bound q on the bin width, which is the quantization level of the measurements, since the bin width must be at least equal to or greater than the quantization step size, as measured by the sensor acquisition process. Therefore, we have the following lower and upper bounds on bin width h:

$$q < h^* < \frac{a_{\max} - a_{\min}}{\sqrt[3]{2n}}. \tag{6}$$

Solving for n in inequality (6), we get the inequality (7), which gives the upper bound for n. This is the maximum

allowable size for the baseline window, which does not allow it to grow arbitrarily large

$$n \leq \frac{1}{2}\left(\frac{a_{\max} - a_{\min}}{q}\right)^3. \tag{7}$$

Distance metrics such as the Kullback–Leibler (*KL*) divergence are well suited for measuring the divergence of a given distribution from a reference or expected distribution. However, most distance metrics are binwise, which requires the number of bins of the two histograms to be the same. Given that both distributions are zero outside the range $(a_{\min}, a_{\max})$, the bin widths for the two distributions must be made the same to perform binwise comparison. This condition imposes a restriction on the range of window sizes (i.e., the value for n) in order to keep the *AMISE* below a desired value. Let n_c and n_b be the number of data points within the current and the baseline windows, respectively. We define three optimization problems:

1. To determine the common optimum bin width for two histograms with different number of data points.
2. To determine the bounds on the baseline window size.
3. To determine the bounds on the current window size.

Optimization problem 1—Given n_c and n_b, find h for optimum (i.e., minimum) *AMISE* for both histograms. In this optimization, we assume that the window sizes are given (based on other criteria such as prior knowledge about the events being monitored), and we find bin width h for which the *AMISE* is minimized for both histograms as follows:

$$h^* = \arg\ \min_h(AMISE(n_c, h) + AMISE(n_b, h)) \tag{8}$$

However, this is a nonlinear optimization and requires a search over h. We can determine the optimum bin width for two window sizes n_c and n_b by averaging the values of the corresponding h_c and h_b using (1). The following is an approximation for (8):

$$\left.\begin{array}{ll}\text{Given}: n_c & \text{Calculate}: h_c^* \\ \text{Given}: n_b & \text{Calculate}: h_c^*\end{array}\right\} \Rightarrow h^* = \frac{h_2^* + h_b^*}{2}. \tag{9}$$

Optimization problem 2—Given n_c, find the range for n_b such that $|AMISE - AMISE_{opt}| \leq \tau$. In this optimization, we assume that the shifting current window size (i.e., resolution) is given, and we find the range for the baseline window for which the $|AMISE - AMISE_{opt}|$ is always kept below a given value. The $AMISE_{opt}$ can be computed using (4) for the current window size n_c.

Optimization problem 3—Given n_b, find the range for n_c such that $|AMISE - AMISE_{opt}| \leq \tau$. In this optimization, we assume that the window for the baseline is given, and we find the range for the current shifting window for which the $|AMISE - AMISE_{opt}|$ is always kept below a given value. This gives us the range of time scales (i.e., resolutions) we are allowed to use. The method used to solve this optimization is the same as optimization problem 2.

Optimization techniques 2 and 3 may be chosen at deployment time, depending on prior knowledge of the length of the history (i.e., baseline) or the range of the temporal span of the events of interest. The number of temporal scales may be chosen depending on the processing power available at deployment time.

Computational complexity

One of the most important aspects of stream processing is the computational complexity and the speed of the data processing. In this section, we analyze the order of complexity and, later in the paper, show the empirical results.

The histogram generation algorithm uses a binary search tree by performing a recursive search with a complexity of $O(\log_2 N)$, where N is the number of histogram bins. The computational complexity of calculating the *KL* distance between two histograms is $O(N)$. The total computation complexity for generating two histograms and calculating their divergence using the *KL* distance is

$$\begin{aligned}\textit{Order of complexity} &= O_{\text{Hist}} + O_{KL} \\ &= O(\log_2 N) + O(N).\end{aligned} \tag{10}$$

We expect the computational time in generating two histograms and computing their distance to increase exponentially as the number of bins increases or as the bin width decreases. We provide the empirical results in the experiments section.

Sensitivity to the distance metric

The density-based change detection approaches use the well-known distance metrics or variations to measure the divergence of the true data distribution from a model distribution. Prior art suggests that the performance of these change detection techniques rely heavily on the distance metric being used. In prior work [17], it has been pointed out that different distance metrics have different sensitivities to changes. For example, the commonly used L^1 distance metric is too sensitive. On the other extreme, L^p norms (for $p > 1$) are far too insensitive in detecting changes.

The choice of the distance metrics depends mainly on the application, and in data mining, it is chosen heuristically. With the number of new kinds of data increasing rapidly, it is increasingly difficult to choose a distance metric that performs well for each data set. This choice is usually made by performing empirical testing and consensus over the accuracy of the results for each application. Aggarwal [18] proposed a user-centric method for modeling a distance metric, which performs better than a pure learning mechanism but is still sensitive to the size of the training data set.

In the previous section, we saw that the choice of the bin width affects the error in generating a histogram. When the bin width is chosen to be smaller than the optimum, the variance of the histogram is high, and therefore, the distance between two histograms having high variance will have a high variance as well. However, as the bin width increases, the histogram gets coarser, and we expect the effect of using different distance metrics to get minimized, but as we saw earlier, this would be at the cost of degraded accuracy.

Here, we analyze how the size of the bin width affects the distance vector obtained while using different distance metrics. We introduce two different metrics to compare the computed distance vector and discuss the results in the experiments section to follow. We call this metric $d(h)$.

The first method for computing metric $d(h)$ measures the difference in the shapes of two discrete vectors (i.e., distance vectors). We use the gradient function to first determine the rate and direction of change of the values in each distance vector and then count the number of times the directions (i.e., slopes) of the two vectors have opposite values.

Let

$$\vec{v}_1(h) = D_1(P, Q, h)$$
$$\vec{v}_2(h) = D_2(P, Q, h)$$
$$K : \text{length of distance vector.}$$

D_1 and D_2 are any two distance metrics used to compute distance vectors v_1 and v_2, respectively, by comparing distributions P and Q, with h being the common bin width used to compute P and Q.

Method 1 (compares only the *shapes* of the distance vectors):

Let: $h_{\min} = q$ be the quantization level of the sensor measurements
and $h_{\max} = (a_{\max} - a_{\min})/\sqrt[3]{2n}$ be the upper bound on h as shown previously in (6)
for ($h = h_{\min}$ to $h_{\max}$) (11)
  $d(h) = 0$
  for ($i = 1$ to K)
    if $((\nabla \vec{v}_1(h))_i/(\nabla \vec{v}_2(h))_i < 0)$
      $d(h) + +$;
  end
end

The second method measures the L^1 distance between the two distance vectors and therefore measures the absolute distance between the values and not just the shape of the vectors.

Method 2:

$$d(h) = \sum_{i=1}^{K} \left| \vec{v}_1^i(h) - \vec{v}_2^i(h) \right|. \quad (12)$$

Experiments and results

We have used three different data sets, which we describe here.

Infrared sensor data set

The sensor data set used in our experiments was obtained by monitoring people walking in a hallway. The objective is to detect activities that are out of the norm. We used a wideband passive infrared (PIR) sensor sampling at a rate of 256 samples per second for a duration of 108 minutes. The PIR measures the temperature difference between the target object and the background. Therefore, its output value is affected by the type of the object (i.e., human body versus nonliving objects), the speed and size of the objects passing by, and the distance of the object to the sensor. In our experiments, the objects detected are people walking by (in a limited range of speed) and in a limited range of distance from the sensor as the hallway has a limited width. Therefore, the PIR output value is directly proportional to the number of people in the field of view of the sensor. The PIR sensor produces integer values that range from 0 to 65,535 with a quantization level $q = 1$.

Calculating the statistical roughness

As mentioned, the optimum h for generating a histogram depends on the number of data points n and the roughness of the distribution given by (3). The roughness is affected by the contents of the underlying data. Using the infrared data, we calculate the sensitivity of optimum h to the roughness only (i.e., keeping n constant). For a given window size n, we find the value of h that corresponds to the smallest $BCV(h)$ given by (4) for different segments of the data by using a sliding window of size n (shifting with 50% overlap with the previous position), and we calculate the standard deviation for optimum h computed from multiple data segments of size n. We repeat this for different values of n.

From this experiment, we have observed that when the number of data points n grows larger than a value (in this case, 60,000 data points or 3% of the total data points), the optimum h is not affected by the contents of the data and only depends on the number of data points n. This shows that optimum h can be estimated by only knowing n, which is the dominant factor in determining the optimum bin width.

Detection accuracy

Figure 1 shows the receiver operating characteristic (ROC) curves using different distance metrics. We have compared using optimum h, undersmoothed h, and oversmoothed h for three different time scales. The ROC curves plotted show the true positive rate (i.e., sensitivity) versus the false positive rate (i.e., 1-specificity), with $sensitivity = TP/(TP + FN)$ and $specificity = TN/(FP + TN)$, where

TP	true positive;
FP	false positive;
TN	true negative;
FN	false negative.

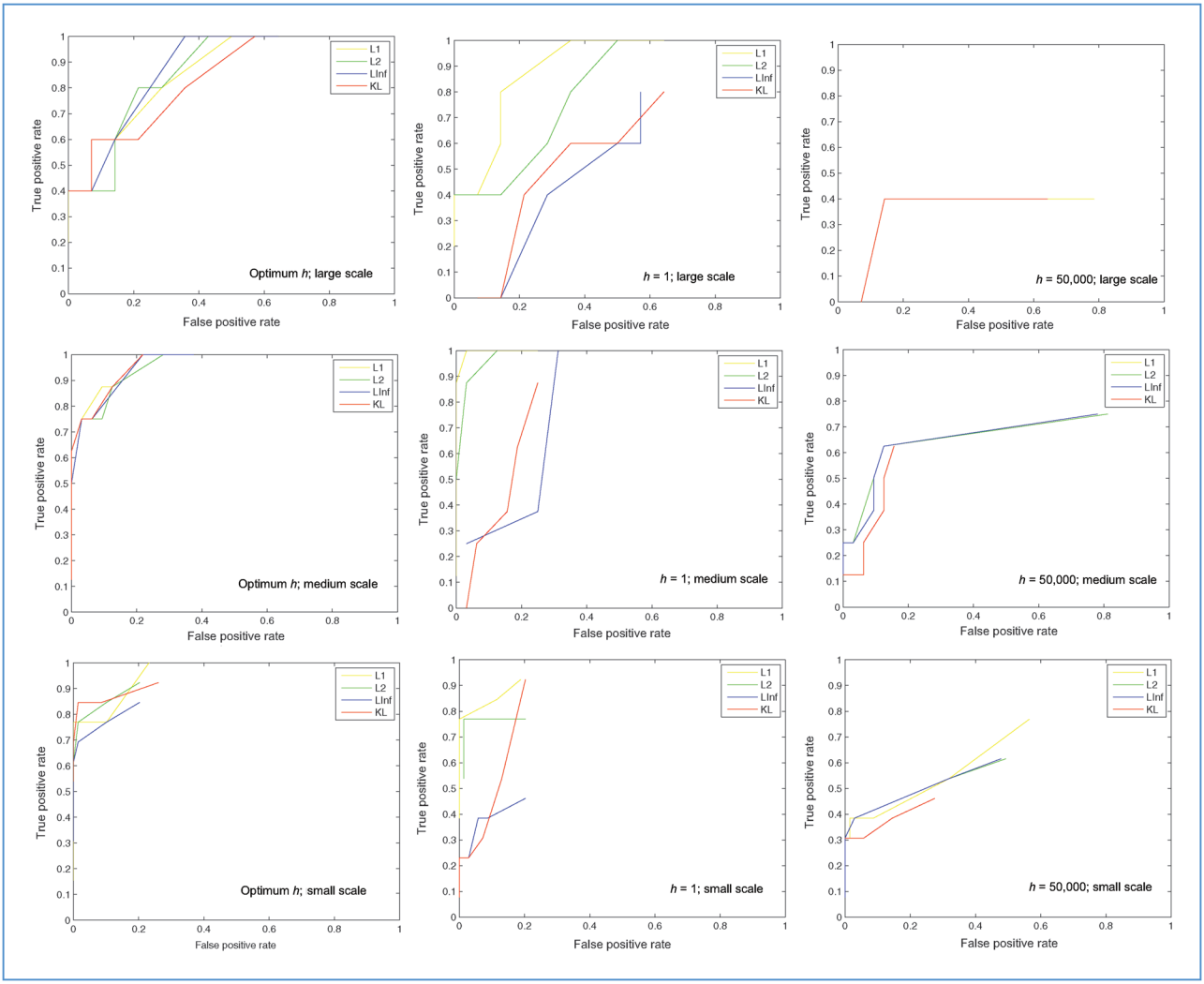

Figure 1

Comparing the ROC curves using different distance metrics (L^1, L^2, L^∞, and KL) for optimum h, undersmoothed h, and oversmoothed h. Three different time scales using the PIR (passive infrared detector) data are shown. The large scale uses a time window of 312 seconds, the medium scale uses a time window of 156 seconds, and the small scale uses a time window of 78 seconds.

In this figure, we have compared various distance metrics such as L^1, L^2, L^∞, and the KL.

The L^1 norm distance metric is given by

$$L^1(P,Q) = \sum_i |p_i - q_i|,$$

where P is the true or the observed distribution, Q is the model or baseline distribution, and p and q are their densities, respectively.

The L^2 norm distance metric is given by

$$L^2(P,Q) = \sum_i (p_i - q_i)^2.$$

The L^∞ norm (i.e., Chebyshev distance) is defined by

$$L^\infty(P,Q) = \max_i(|p_i - q_i|).$$

The KL divergence metric is given by

$$KL(P,Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}.$$

The figure shows that using optimum h results in a much higher accuracy in detecting changes. This figure also shows that, when using the optimum bin width, all the different distance metrics perform well (as shown for three time scales).

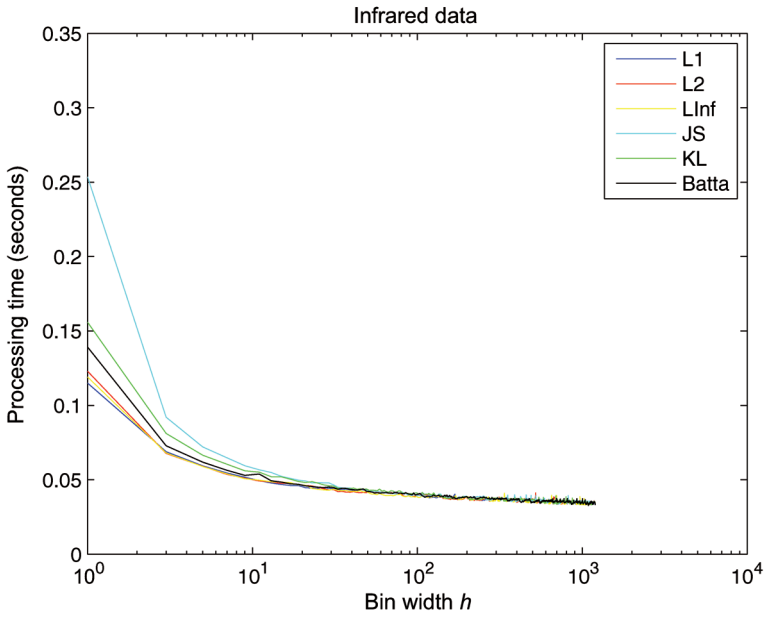

Figure 2

Computation speed as a function of bin width for different distance metrics using the infrared data. The distance metrics used are L^1, L^2, L^∞, *JS*, *KL*, and Bhatta (Bhattacharyya). ($n_c = 40{,}000; n_b = 10^6; h_{opt} = 450$.)

Computational complexity

Figure 2 shows the empirical results on the effect of the bin width on computation time. The measured computation time is for segmenting the data stream, generating the histograms, and detecting the outlier segments. In this experiment, the number of data points in the shifting window (i.e., time scale) is $n_c = 40{,}000$. The number of data points in the baseline window is $n_b = 10^6$. For this scale, the optimum bin width is calculated to be roughly 450. The plot shows that the processing time drops exponentially as the bin width increases, which supports our analysis given in (10). We can see that, using the proposed optimum bin width (as opposed to an arbitrary value), we achieve higher accuracy and a low processing time.

In this figure, we have compared various distance metrics such as L^1, L^2, L^∞, Jensen-Shannon (*JS*), *KL*, and the *Bhattacharyya* distance.

The *JS* distance metric is given by

$$JS(P,Q) = \frac{1}{2}KL(P,M) + \frac{1}{2}KL(Q,M),$$

where

$$M = \frac{1}{2}(P+Q).$$

The Bhattacharyya distance metric is given by $Bhatta(P,Q) = -\ln(BC(p,q))$, where

$$BC(p,q) = \sum_{x \in X} \sqrt{p(x)q(x)}.$$

Distance metric analysis

Figure 3 shows the shape comparison between the distance vectors using different distance metrics as a function of h using the metric specified by (11). In the figure, we show the shape difference using six different types of distance metrics such as L^1, L^2, L^∞, *KL*, *JS*, and *Bhattacharyya*. In this experiment, the optimum h is around 80. We can see that, once the optimum h is reached, the effect of using the type of the distance metric is low.

Figure 4 shows a similar graph but uses (12) to measure the difference between the distance vectors using different distance metrics. This metric not only uses the shapes of the distance vectors but also measures the absolute distance between the values in the vectors. The results are similar to Figure 3.

As can be seen in these figures, as the bin width increases, the computation time and the sensitivity of the detection to the choice of the distance metric both drop at the same rate. We see that choosing the optimum bin width gives the

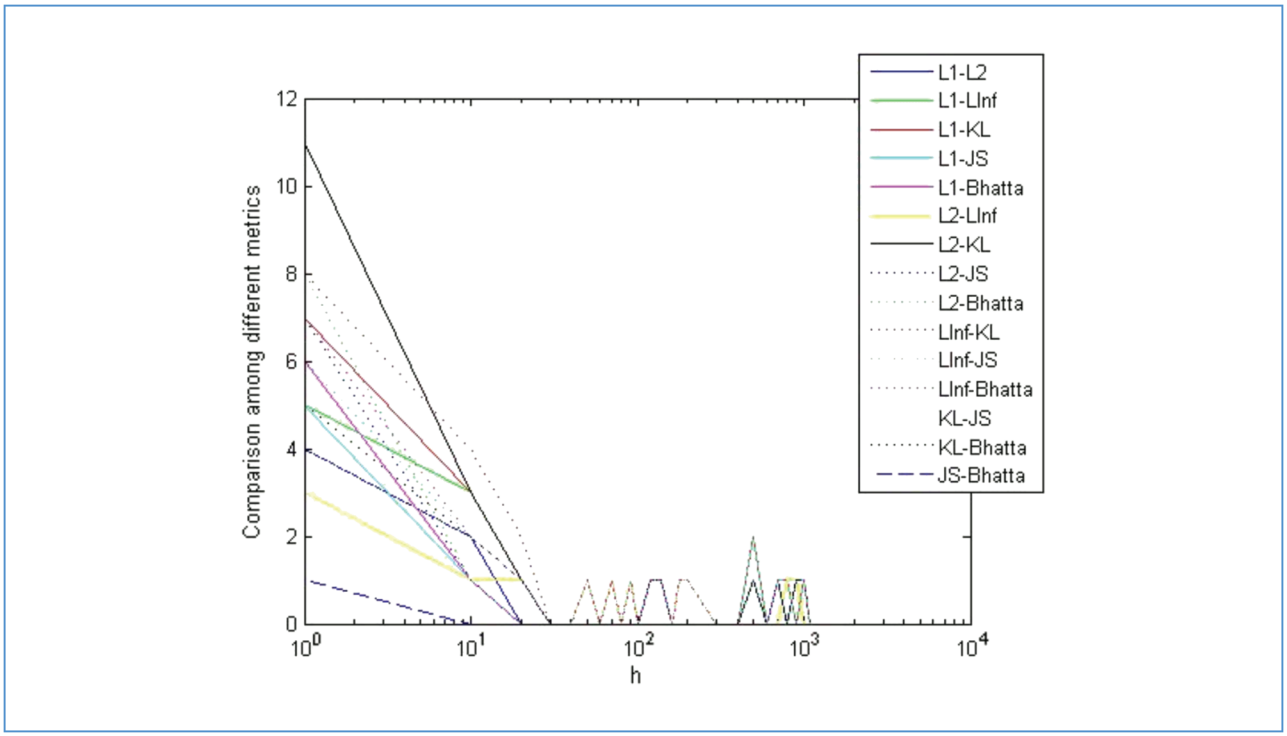

Figure 3

Comparison among distance different metrics $d(h)$ using (11) versus h using the infrared data. The current window is $t_c - t_b = 312$ seconds, and *average* $h_{opt} = 80$.

best accuracy (i.e., Figure 1) while having low computation time (i.e., Figure 2). It is also performs the same no matter what distance metric is being used (i.e., Figures 3 and 4).

Motes data set

For the second data set, we have used sensor data generated by motes measuring temperature, humidity, and light in the Intel Berkeley Research Laboratory [1]. We have used the temperature and light measurements by mote 1 between February 28 and April 5, 2004. The sampling rate for the measurements for both temperature and light is once every 31 seconds. The temperature measurements are float numbers ranging between 0 and 123, and the light measurements are floats ranging between 0 and 714.

The system detects anomalies such as changes in the lights' intensity (e.g., going on and off), as well as changes in the temperature in various locations of the laboratory. Using this data set, we again see the importance of analyzing the data at multiple temporal scales and using the optimum bin width.

Passenger arrival data set

For the third data set, we used a time series data set from [2], which captures the passenger arrival rates at a bus terminal in Santiago de Chile. The number of passengers arriving at the terminal is recorded every 15 minutes between 6:30 A.M. and 10:00 P.M. each day for a total of approximately 650 days. We have multiplied the data set several times to make the length of the data larger.

There are periods of low activity on weekends, when the passenger arrival rate is much lower than it is on weekdays. The system detects weekends as outliers, which is repeated over time, but as time goes on, the intensity of the outlier (i.e., distance value) is reduced as the system learns about the events as a normal behavior and perhaps not an anomaly. We have used different distance metrics such the *KL*, *JS*, and *Euclidean*, and observed similar results. Another observation is that when analyzed at a very small scale (e.g., 4 hours), the anomalies are not detected, but when the scale is roughly 50 hours (i.e., 2 days), the weekend is detected as an anomaly. Saturdays and Sundays are also detected as anomalies at the scale of roughly 24 hours (i.e., 1 day). We see an exponential decrease in the processing time as the bin width is increased, but the most gain is achieved when using the optimum bin width. Increasing the bin width to much larger values than the optimum does not gain much in speed and reduces the detection accuracy.

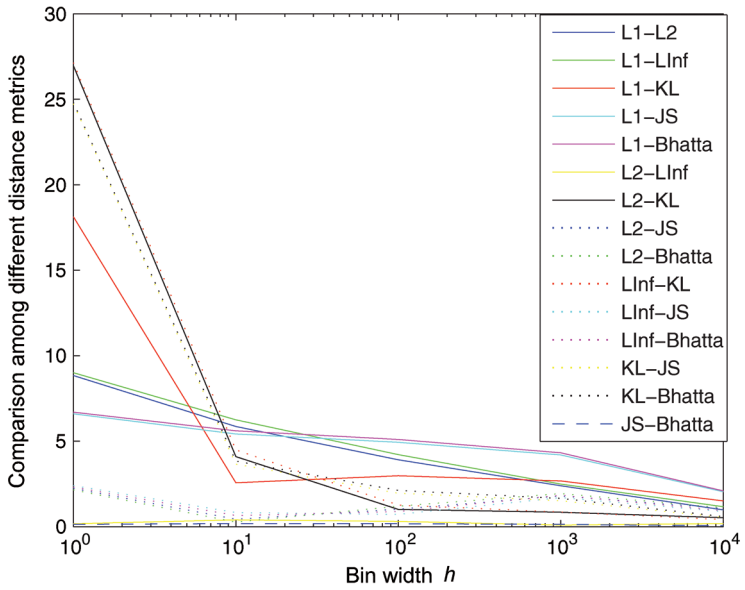

Figure 4

Comparison among distance different metrics $d(h)$ using (12) versus h using the infrared data. The current window is $t_c - t_b = 312$ seconds, and *average* $h_{opt} = 80$.

Conclusion and future work

In this paper, we have presented an approach for detecting anomalies that does not require any predefined rules, models, or domain-specific knowledge base. Our approach uses anomaly detection at multiple temporal scales using streams of sensor data. The approach is nonparametric and does not make any assumptions regarding the distribution of the data. We believe it is one of the first nonparametric self-optimizing algorithms of this nature in the technical community. Our approach optimizes operational constants such as optimal bin width on its own, rather than requiring human input to specify such constants. We have shown the importance of using the optimal bin width in generating histograms for representing the data as opposed to choosing an arbitrary value, in terms of accuracy and computational speed, as well as its effect on the choice of the distance metric.

The algorithms for selecting optimum bin width or, in a more general term, *quantization level* can be applied to other domains and classifiers. We are studying whether the optimal bin width (i.e., the number of bins) algorithm can be applied to calculating an optimal value for the dictionary size in the Bag of Words (BoW) model [19]. The BoW model is used in document and image classification. In this model, a text document or an image is represented as an unordered collection of words or code words, respectively. The BoW is then used to determine the similarity between two documents or images. We will study whether an optimum size for the BoW (e.g., dictionary size) can be calculated rather than using heuristic and domain specific values.

Acknowledgments

Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. government, the U.K. Ministry of Defence, or the U.K. government.

References

1. *Intel Lab Data*. [Online]. Available: http://db.csail.mit.edu/labdata/labdata.html
2. *Time Series Data Library*. [Online]. Available: http://www-personal.buseco.monash.edu.au/~hyndman/TSDL/
3. V. Barnett and T. Lewis, *Outliers in Statistical Data*. Hoboken, NJ: Wiley, 1994.
4. A. S. Hadi, "A modification of a method for the detection of outliers in multivariate samples," *J. Roy. Stat. Soc. Ser. B*, vol. 56, no. 2, pp. 393–396, 1994.

5. E. Knorr, R. Ng, and V. Tucakov, "Distance-based outliers: Algorithms and applications," *VLDB J.*, vol. 8, no. 3/4, pp. 237–253, Feb. 2000.
6. C. Aggarwal and P. Yu, "Outlier detection for high dimensional data," in *Proc. ACM SIGMOD*, T. Sellis, Ed., 2001, pp. 37–46, DOI: 10.1145/375663.375668. [Online]. Available: http://doi.acm.org/10.1145/375663.375668
7. K.-S. Goh, K. Miyahara, R. Radhakrishan, A. Xiong, and A. Divakaran, "Audio-visual event detection based on mining of semantic audio-visual labels," in *Proc. SPIE Conf. Storage Retrieval Multimedia Databases*, 2004, vol. 5307, pp. 292–299.
8. S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, "Online outlier detection in sensor data using nonparametric models," in *Proc. 32nd Int. Conf. VLDB*, 2006, pp. 187–198.
9. A. Haar, "Zur Theorie der orthogonalen Funktionensysteme," *Math. Ann.*, vol. 71, no. 1, pp. 38–53, Mar. 1911.
10. A. P. Witkin, "Scale-space filtering," in *Proc. Int. Joint Conf. Artif. Intell.*, 1983, pp. 1019–1021.
11. M. Beigi, S.-F. Chang, S. Ebadollahi, and D. Verma, "Multi-scale temporal segmentation and outlier detection in sensor networks," in *Proc. IEEE ICME*, 2009, pp. 306–309.
12. P. Mermelstein, "Distance measures for speech recognition, psychological and instrumental," in *Pattern Recognition and Artificial Intelligence*, C. H. Chen, Ed. New York: Academic, 1976, pp. 374–388.
13. C. Jutten and J. Herault, "Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture," *Signal Process.*, vol. 24, no. 1, pp. 1–10, 1991.
14. D. Scott, *Multivariate Density Estimation: Theory and Practice and Visualization*. Hoboken, NJ: Wiley, 1992.
15. N. Hjort, "On frequency polygons and average shifted histograms in higher dimensions," Stanford Univ., Stanford, CA, Tech. Rep. 22, 1986.
16. D. W. Scott and G. R. Terrell, "Biased and unbiased cross-validation in density estimation," *J. Amer. Stat. Assoc.*, vol. 82, no. 400, pp. 1131–1146, Dec. 1987.
17. D. Kifer, S.-B. David, and J. Gehrke, "Detecting change in data streams," in *Proc. 30th Int. Conf. VLDB*, 2004, vol. 30, pp. 180–191.
18. C. Aggarwal, "A framework for diagnosing changes in evolving data streams," in *Proc. ACM SIGMOD*, 2003, pp. 575–586, DOI: 10.1145/872757.872826. [Online]. Available: http://doi.acm.org/10.1145/872757.872826
19. D. D. Lewis, "Naïve (Bayes) at forty: The independence assumption in information retrieval," in *Proc. 10th ECML*, 1998, pp. 4–15.

Received February 15, 2011; accepted for publication March 21, 2011

Mandis S. Beigi *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (mandis@us.ibm.com)*. Ms. Beigi is a Senior Software Engineer in the Next Generation Computing department at the IBM T. J. Watson Research Center, Yorktown Heights, NY. She received a B.S. degree in electrical engineering from the State University of New York at Stony Brook, Stony Brook, in 1993, and M.S. and M.Phil. degrees in electrical engineering from Columbia University, New York, NY, in 1995 and 2009, respectively. She joined IBM in 1994 and has been working on networking and system management. In 2010, she received an IBM Outstanding Innovation Award for her work on System Software Agent for z/OS* communication server.

Shih-Fu Chang *Columbia University, New York, NY USA (sfchang@ee.columbia.edu)*. Professor Chang is Professor of Electrical Engineering and Director of Digital Video and Multimedia Lab at Columbia University, New York, NY. He has made significant contributions to multimedia search, visual communication, media forensics, and international standards for multimedia. He has been recognized with several award, including IEEE Kiyo Tomiyasu Award, Navy Office of Naval Research Young Investigator Award, IBM Faculty Award, ACM Recognition of Service Award, and National Science Foundation CAREER Award. He and his students have received many Best Paper and Best Student Paper Award from IEEE, ACM, and SPIE. He has worked in different advising and consulting capacities for IBM, Microsoft, Kodak, PictureTel, and several other institutions. He was elected to IEEE Fellow in 2004, served as Editor-in-Chief for *IEEE Signal Processing Magazine* (2006–2008), and Chair of Columbia Electrical Engineering Department (2007–2010).

Shahram Ebadollahi *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (ebad@us.ibm.com)*. Dr. Ebadollahi manages the Healthcare Transformation Research Group and is the Founding co-chair of the Medical Informatics community at the IBM T. J. Watson Research Center, Yorktown Heights, NY. In these capacities, he manages a group of scientists and physicians conducting research in the broad area of health informatics, especially applications of data mining, machine learning, advanced visualization, simulation, and comparative effectiveness in healthcare applications. In addition, he has been conducting active research in the domain of multimedia content analysis, event recognition, and computer vision. He received his Ph.D. and M.S. degrees in electrical engineering from Columbia University, New York, NY, and joined IBM Research n 2005 as a Research Staff Member. He is also an adjunct Assistant Professor with the department of electrical engineering at Columbia University.

Dinesh C. Verma *IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (dverma@us.ibm.com)*. Dr. Verma is a Research Staff Member and Department Group Manager at the IBM T. J. Watson Research Center, Yorktown Heights, NY. He manages the IT and Wireless Convergence team at the center. He is the program manager for the International Technology Alliance. He received his undergraduate degree from the Indian Institute of Technology, Kharagpur, India, in 1987 and his doctorate degree in 1991 from the University of California, Berkeley, in the Tenet Networking Group headed by Prof. Domenico Ferrari. He joined IBM Research in 1992 and worked on network control protocols and algorithms. He is a Fellow of the IEEE and has authored five books on the topic of networking.