Distance-Based Outliers in Sequences

Girish Keshav Palshikar

Tata Research Development and Design Centre (TRDDC),
54B Hadapsar Industrial Estate Pune 411013, India
GK.Palshikar@tcs.com

Abstract. Automatically finding *interesting*, *novel* or *surprising* patterns in time series data is useful in several applications, such as fault diagnosis and fraud detection. In this paper, we extend the notion of distance-based outliers to time series data and propose two algorithms to detect both global and local outliers in time series data. We illustrate these algorithms on some real datasets.

Keywords: Novelty detection, Outlier detection, Time series, Sequence mining.

1 Introduction

Analyzing a sequence of values is an important task in many practical applications. For example, the sequence of observed values of the parameters of a chemical process is analyzed to understand output quality and for process diagnosis. Telemetry data sent by a system onboard a satellite is analyzed to evaluate the system's health. The trades performed by a trader in a stock exchange can be analyzed to understand his/her financial performance in the market.

In such applications, the sequence to be analyzed consists of an ordered list of records (points). If each record consists of a single field then the sequence is *univariate*; otherwise it is *multivariate*. The ordering of records within a sequence is often based on a timestamp, in which case the sequence can be considered as a time series. An important question during the analysis of the sequence is: how do we identify *interesting*, *novel* or *anomalous* subsequences in the sequence? Note that identifying such subsequences is different from identifying single outlier points. We now need to define the meaning of terms such as interesting or anomalous. In the simplest case, extreme (high or low) values occurring in the sequence can be found out using standard statistical techniques for outlier detection in a time-series. However, in practice, we are often interested in more complex kinds of interesting or anomalous regions in the sequence. For example, (1) contiguous subsequences; or (2) noncontiguous subsequence (list of points not necessarily contiguous) etc. In this paper, we focus on the problem of automatically identifying contiguous subsequences of a given sequence, which are interesting or anomalous in a well-defined sense.

2 Related Work

Basic statistical techniques for outlier detection, including in time series data, are discussed in [1]. The notion of distance-based outliers in (non time series) datasets was

G. Chakraborty (Ed.): ICDCIT 2005, LNCS 3816, pp. 547 – 552, 2005.

proposed in [4]. A related notion was proposed in [6]. This paper extends the approach in [4] to time series data. Several other techniques for *novelty detection* have been proposed [2], [7], [3], [5] for identifying *interesting* subsequences in a time series. See also H. Geirsson et al [http://hraun.vedur.is/ja/skyrslur/contgps/ node8.html].

3 Distance-Based Outliers Detection in Sequences

3.1 Outlier Subsequence

An *n-sequence* (or *a sequence of length n*) is an ordered finite sequence $s = \langle \mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{n-1} \rangle$ of $n \geq 1$ elements. Elements of a multivariate (or multidimensional) sequence are tuples (or vectors). An *m*-sequence $\langle \mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{m-1} \rangle$ is a (*contiguous*) *subsequence* of another sequence $s = \langle \mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{n-1} \rangle$ if $\mathbf{x}_0 = \mathbf{s}_i$, $\mathbf{x}_1 = \mathbf{s}_{i+1}$, ..., $\mathbf{x}_{m-1} = \mathbf{s}_{i+m-1}$, for some $0 \leq i \leq n - m$ i.e., a subsequence is a contiguous part of the original sequence; e.g., <2, 8, 5> is a subsequence of sequence <8,7,2,8,5,4,4>. We consider the problem of detection of interesting or anomalous subsequences in a given single sequence. For this, we adapt the notion of a distance-based outlier in a set of points, proposed in [4], to distance-based outlier subsequence of a given sequence.

Let $\mathbf{d}(\mathbf{x}_i, \mathbf{x}_j)$ denote the function to compute the distance between two elements $\mathbf{x}_i$ and $\mathbf{x}_j$ of a sequence; e.g., **d** could be Euclidean, Mahanttan or general Minkowski distance. There are several ways in which the distance $d(\alpha, \beta)$ between two *m*-sequences $\alpha = \langle \mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{m-1} \rangle$ and $\beta = \langle \mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{m-1} \rangle$ can be computed. For example, the *Minkowski distance* is defined as

$$d(\alpha, \beta) = \sqrt[p]{\mathbf{d}^p(\mathbf{x}_0, \mathbf{y}_0) + \mathbf{d}^p(\mathbf{x}_1, \mathbf{y}_1) + \ldots + \mathbf{d}^p(\mathbf{x}_{m-1}, \mathbf{y}_{m-1})}$$

For example, for $\alpha = \langle 7, 2, 3 \rangle$, $\beta = \langle 3, 0, 5 \rangle$, $\mathbf{d}(\mathbf{x}_2, \mathbf{y}_2) = \mathbf{d}(2, 0) = 2$, whereas $d(\alpha, \beta) = [(7 - 3)^2 + (2 - 0)^2 + (3 - 5)^2]^{1/2} = 4.9$. When each $\mathbf{x}_i$ and $\mathbf{y}_i$ is either 0 or 1, $p = 1$ and when $\mathbf{d}(\mathbf{x}, \mathbf{y}) = \text{XOR}(x, y)$, the above distance *d* reduces to usual Hamming distance between two Boolean *m*-sequences.

3.2 Algorithm 1

We now adapt Knorr's notion of distance-based outliers in a set of points to distance-based outlier *m*-subsequences of a given sequence. Let $s = \langle \mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{n-1} \rangle$ be a given *n*-sequence. Let $m \geq 1$ be a given integer. Let $\Omega(s, m)$ denote the set of all possible *m*-subsequences of *s*; e.g., $\Omega(\langle 8,7,2,8,5,4,4 \rangle, 4) = \{\langle 8,7,2,8 \rangle, \langle 7,2,8,5 \rangle, \langle 2,8,5,4 \rangle, \langle 8,5,4,4 \rangle\}$. Clearly, $\Omega(s, m) = n - m + 1$. Knorr [4] proposed a distance-based definition of an outlier in a given set *S* of points: a point $x \in S$ is an outlier if at least *p*% points in *S* are at a distance > *D* from *x*, where *p* and *D* are user specified positive real numbers. We propose a simple generalization of this definition to adapt it for outlier subsequences of a given sequence.

Definition 1. Let $s = \langle \mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{n-1} \rangle$ be a given *n*-sequence. Let *m* be a given integer such that $0 \leq m \leq n\text{-}1$. Let $0 \leq p \leq 1$ and $D \geq 0$ be two given real numbers. An

m-subsequence $a = \langle \mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{m-1} \rangle$ of s is a (p, m, D)-*outlier* in s if at least p% of the m-subsequences in $\Omega(s, m)$ are at a distance > D from a.

Consider a 19-sequence s = <2, 5, 6, 2, 3, 1, 2, 9, 9, 9, 1, 2, 2, 1, 3, 1, 0, 2, 1>. For m = 3, $\Omega(s, m)$ contains 19 – 3 + 1 = 17 3-subsequences. Suppose D = 10.0 and p = 60%. For the 3-subsequence <2, 3, 1> starting at 4th position, there is only 1 subsequence in $\Omega(s, m)$ at a distance > 10.0 (using Euclidean distance); thus the fraction of 3-subsequences at a distance > 10.0 from this subsequence is 1/17 = 5.9%. Since 5.9 < 60.0, this 3-subsequence is not an outlier. For the subsequence <9, 9, 9>, there are 11 subsequences (i.e., 11/17 = 64.7%) which are at a distance > 10.0 from it. Thus this 3-subsequence is an outlier, for the given values of p and D.

Knorr [4] contains an algorithm to find a set of distance-based outliers from a given set of points. We present below a simple generalization of the core of Knorr's algorithm to detect outlier m-subsequences of a given sequence.

```
// Modified Knorr's algorithm for distance-based outlier m-
// subsequences; m ≥ 1. 0 ≤ p ≤ 1 = fraction of m-subsequences
// at distance > D from an outlier; D = a distance value
algorithm knorr_seq
input  sequence s of n elements;
input  m, p, D;
M := n – m + 1; // no. of m-subsequences of s
for (i = 0; i <= (n - m); ) {
    for (j = 0, count = 0; j <= n - m; j++) {
        d := d(<s_i, s_{i+1}, ..., s_{i+m-1}>, <s_j, s_{j+1}, ..., s_{j+m-1}>);
        if ( d > D ) then count++; end if;
    } // end for
    if ( count/total > p ) then {
        printf("Outlier sub-sequence from %d to %d\n",i,i+m-1);
        i = i + m;
    } else i++; end if;
} // end for
```

Essentially, the algorithm compares every candidate m-subsequence $a = \langle \mathbf{s}_i, \mathbf{s}_{i+1}, \ldots, \mathbf{s}_{i+m-1} \rangle$ with every other m-subsequence $b = \langle \mathbf{s}_j, \mathbf{s}_{j+1}, \ldots, \mathbf{s}_{j+m-1} \rangle$, incrementing *count* if $d(a, b) > D$. Thus, for every candidate m-subsequence of the given sequence, the algorithm counts the number of m-subsequence that are at a distance > D from it. If this number exceeds the specified limit, that m-subsequence is declared as an outlier. The user has to provide values for the parameters p, D and m. Our implementation offers a choice of various distance measures to the user (e.g., Manhattan, Euclidean, etc.). Clearly, the complexity of the algorithm is $O(n^2)$ where n = size of the given sequence. For correctness, we state the following without proof:

Proposition 2. Every m-subsequence declared as an outlier by the algorithm knorr_seq satisfies Definition 1. Conversely, every m-subsequence that satisfies Definition 1 is declared as an outlier by the algorithm, provided no subsequence overlapping with it has already been declared an outlier.

This algorithm will not generate overlapping outlier subsequences, due to the jump in the value of i (statement $i = i + m$) after an outlier sub-sequence is found. Fig. 1

shows the daily quantity of a commodity traded on a stock exchange for 52 days. The above algorithm, called with $m = 4$, $p = 0.40$ (40%), $D = 150000.0$ and using Euclidean distance, reports the following two 4-subsequences as outliers: 43 … 46 and 47 … 50. This is reasonable, since the volume is drastically different in these periods compared to the other days.

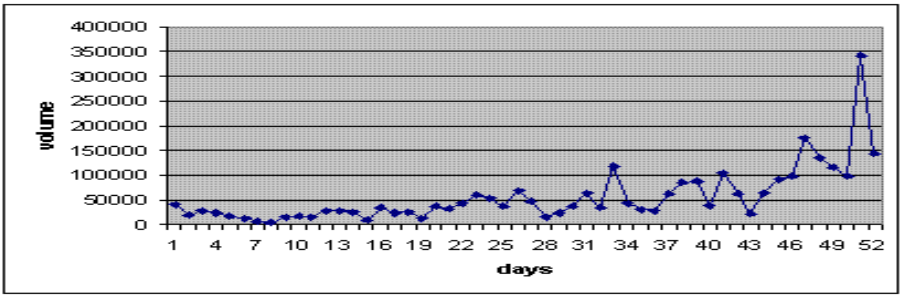

Fig. 1. Daily trading volume for a period of 52 days

3.2 Algorithm 2

Consider the time series in Fig. 2. The subsequence from 100 to 124, consisting of two cycles that are much shorter than their neighbours, is naturally an *interesting*. However, it is difficult to find it as an outlier using the above algorithm, since the values in this region occur as part of many other cycles. This is an example of a *local outlier*, which is an outlier only in relation to a few of its immediate (left and right) neighbouring subsequences. In contrast, Definition 1 considered the entire sequence and hence the resulting outliers can be called *global outliers*.

Definition 3. Let s be a given sequence. Let $\alpha = <\mathbf{s}_i, \mathbf{s}_{i+1}, \ldots, \mathbf{s}_j>$ be a given subsequence of s. Let $0 \leq m \leq n\text{-}1$, $k \geq 1$ be given integers. The set $\Psi_L(m, k, \alpha)$ of k *left neighbours* of α contains the following k m-subsequences $\{<s_{i-m-k+1},\ldots,s_{i-1}>,\ldots,<s_{i-m}, s_{i-k}>\}$. The set $\Psi_R(m, k, \alpha)$ of k *right neighbours* of α contains the following k m-subsequences $\{<s_{j+1},\ldots,s_{j+m}>,\ldots,<s_{j+k}, s_{j+k+m}>\}$. We define the *set of neighbours* of α as $\Psi(m, k, \alpha) = \Psi_L(m, k, \alpha) \cup \Psi_R(m, k, \alpha)$.

For s = <3,5,4,6,8,9,5,5,4,6,3,5,6,2,5>, α = <5,5,4>, $m = 3$, $k = 4$, the set of 4 left neighbours of α is $\Psi_L(3, 4, \alpha)$ = {<6,8,9>, <4,6,8>, <5,4,6>, <3,5,4>}; the set of 4 right neighbours of α is $\Psi_R(3, 4, \alpha)$ = {<6,3,5>, <3,5,6>, <5,6,2>, <6,2,5>}.

Definition 4. Let s be a given sequence. Let $0 \leq m \leq n\text{-}1$, $k \geq 1$ be given integers. Let $0 \leq p \leq 1$ and $D \geq 0$ be two given real numbers. An m-subsequence a of s is a (p, m, D, k)-*left-local-outlier* (or, simply *left outlier*) in s if at least p% of the m-subsequences in $\Psi_L(m, k, a)$ are at a distance $> D$ from a. *Right outlier* and *local outlier* are defined similarly using $\Psi_R(m, k, a)$ and $\Psi(m, k, a)$.

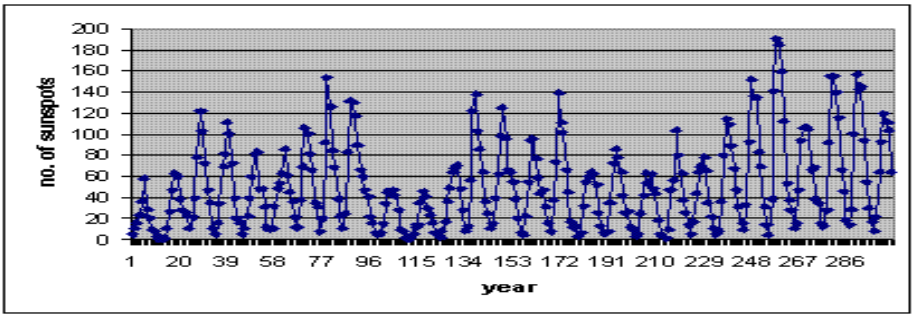

Fig. 2. Average number of sunspots per year

We now modify the algorithm to detect left local outliers in a given sequence; algorithms to detect right outliers and local outliers are similar. The algorithm counts how many of the k left neighbours of a particular candidate m-subsequence a are at a distance $> D$ from it. If this number is $> M$, where M is given by the user, then it declares a as a left outlier. Our implementation offers a choice of various distance measures to the user (e.g., Manhattan, Euclidean etc.). The complexity of the algorithm is $O(k*n)$ where n = size of the given sequence and k = the no. of neighbours to be checked on the left side. For correctness, we state the following without proof:

Proposition 5. Every m-subsequence declared as a left outlier by the algorithm knorr_seq2 satisfies Definition 4. Conversely, every m-subsequence that satisfies Definition 4 is declared as a left outlier by the algorithm, provided no subsequence overlapping with it has already been declared a left outlier.

```
algorithm knorr_seq2
input  sequence s of n elements;
input  m, k, M, D;
for (i = 0; i <= (ps->N - m); ) {
   for (j=i-m-k+1,count=0; j >= 0 && j+m-1 < i; j++) {
        d := d(<s_i,s_{i+1},...,s_{i+m-1}>, <s_j,s_{j+1},...,s_{j+m-1}>);
        if ( d > D ) then count++; endif;
   } // end for
   if ( count > M ) then {
       printf("Left outlier: start=%d end=%d\n",i,i+m-1);
       i = i + m;
   } else
       i++;
} // end for
```

We have also extended the approach to detect inliers, such as those in Fig. 2.

4 Conclusions and Further Work

We proposed an extension of the distance-based outlier detection approach of [4] to detect interesting subsequences of a given sequence. The essential idea is that

interesting subsequences can be modeled as *outliers* in the distance-based framework. We presented two algorithms to detect both global and local outliers in a given time-series data. An implementation provides a choice of several variants of these algorithms, along with different types of distance (or similarity) measures. We demonstrated the use of these algorithms to detect some interesting subsequences in some example datasets. The first limitation of this approach is that the user has to provide values for 3-4 parameters, which requires some experimentation. We are looking at the use of machine-learning algorithms for automatically learning values for these parameters, from a given set of already known interesting subsequences. Also, the quadratic complexity makes the algorithms too slow for large time series datasets. We are looking at the use of some well known index structures to improve the efficiency. Though, in principle, our techniques should work well even with multidimensional time series, we need to validate this on real-life time series. We are conducting several experiments to compare our results with those reported by other well-known algorithms for novelty detection in time series.

Acknowledgements

I would like to thank Prof. Mathai Joseph for his support and colleagues in TRDDC for useful discussions and help. Sincere thanks to Dr. Manasee Palshikar for providing the foundation for all my research work.

References

1. V. Barnett, T. Lewis, *Outliers in Statistical Data*, John Wiley and Sons, 1994.
2. D. Dasgupta, S. Forrest, "Novelty Detection in Time Series Data using Ideas from Immunology", Proc. 5th Conf. Intelligent Systems, 1996.
3. E. Keogh, S. Lonardi, B. Chiu, "Finding Surprising Patterns in a Time Series Database in Linear Time and Space", Proc. 8th ACM Int. Conf. Knowledge Discovery and Data Mining, ACM Press, pp. 550 – 556, 2002.
4. E. M. Knorr, R. T. Ng, "Algorithms for Mining Distance-based Outliers in Large Datasets", Proc. VLDB Conf., 1998, pp. 392 – 403.
5. J. Ma, S. Perkins, "Online Novelty Detection on Temporal Sequences", Proc. Int. Conf. Know. Discovery Data Mining, Springer-Verlag, pp. 275 – 295, 2003.
6. S. Ramaswamy, R. Rastogi, K. Shim, "Efficient Algorithms for Mining Outliers from Large Datasets", Proc. SIGMOD2000, ACM Press, pp. 162-172, 2000.
7. C. Shahabi, X. Tian, W. Zhao, "TSA-Tree: A Wavelet based Approach to Improve the Efficiency of Multilevel Surprise and Trend Queries", Proc. 12th Int. Conf. Scientific Statistical Database Management, pp. 55 – 68, 2000.