PERGAMON

Computers & Industrial Engineering 43 (2002) 703–720

computers & industrial engineering

www.elsevier.com/locate/dsw

Multiscale approximation methods (MAME) to locate embedded consecutive subsequences—its applications in statistical data mining and spatial statistics

Xiaoming Huo

School of Industrial and System Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, USA

Abstract

In statistical data mining and spatial statistics, many problems (such as detection and clustering) can be formulated as optimization problems whose objective functions are functions of consecutive subsequences. Some examples are (1) searching for a high activity region in a Bernoulli sequence, (2) estimating an underlying boxcar function in a time series, and (3) locating a high concentration area in a point process. A comprehensive search algorithm always ends up with a high order of computational complexity. For example, if a length-n sequence is considered, the total number of all possible consecutive subsequences is $\binom{n+1}{2} \approx n^2/2$. A comprehensive search algorithm requires at least $O(n^2)$ numerical operations.

We present a multiscale-approximation-based approach. It is shown that most of the time, this method finds the exact same solution as a comprehensive search algorithm does. The derived multiscale approximation methods (MAMEs) have low complexity: for a length-n sequence, the computational complexity of an MAME can be as low as $O(n)$. Numerical simulations verify these improvements.

The MAME approach is particularly suitable for problems having large size data. One known drawback is that this method does not guarantee the exact optimal solution in every single run. However, simulations show that as long as the underlying subjects possess statistical significance, a MAME finds the optimal solution with probability almost equal to one. © 2002 Published by Elsevier Science Ltd.

Keywords: Data mining; Maximum likelihood estimate; Multiscale approximation

1. Introduction

1.1. Data mining and statistics

Companioning the fast growth of internet and the advances in data acquisition, huge amounts of data are now available. A new field of science—data mining—has emerged to compensate the needs of extracting useful information from massive and complex data. Many novel methods have been developed. It is exciting to know that statistics is playing a more and more important role. This is mainly

0360-8352/02/$ - see front matter © 2002 Published by Elsevier Science Ltd.

PII: S0360-8352(02)00134-1

due to the fact that there is always a substantial amount of randomness present in the data. It becomes a recent challenge to develop methods that are both statistically sound and computationally efficient.

In this paper, three statistical problems are analyzed. These problems are motivated by cases in data mining and spatial statistics. Existing ways to solve these problems usually require a comprehensive search, and are too costly. A multiscale approximation method (MAME) can have low computational complexity and can approximate the ultimate optimal solution with high precision.

1.2. Low complexity methods

To illustrate the importance of having low complexity algorithms, three problems are studied. They are:

1. In statistical data mining, a region that has high activity (Amaratunga & Cabrera, 2001) needs to be identified.
2. In a detection problem, an interval that has a different statistical average is of interest.
3. In spatial statistics, particularly in clustering, in order to choose clusters, a region having high concentration of data points needs to be calculated.

A common feature of these problems is that interest areas are either intervals (in continuous cases) or consecutive subsequences (in discrete cases).

A statistically sensible approach to solve these problems is to calculate their *maximum likelihood estimate* (MLE). Note that in these problems, the start and end positions of a subsequence are variables of a likelihood function. The computational complexity of optimizing such a function can be computed: for a length-n sequence of observations, the total number of consecutive subsequences is equal to $\binom{n+1}{2} \approx n^2/2$. Hence an algorithm that is based on a comprehensive search requires at least $O(n^2)$ operations.

One possibility of finding a low complexity method is to take advantage of local properties at boundaries of interested regions, e.g. considering derivatives of Gaussian in detecting a boxcar function (Sadler & Swami, 1999). However, this approach does not adopt a global optimality criterion, therefore is not expected to possess a global optimality. On the other hand, the $O(n^2)$ complexity of a comprehensive search method makes it too expensive to be practical. In computational mathematics and signal processing, an algorithm whose computational complexity is $O(n)$ or $O(n\log(n))$ is generally considered acceptable. This will be our goal in designing a new approach.

1.3. Multiscale approximation

The main approach in this paper is to develop MAMEs. Recently, multiscale methods have been proven to be effective in applied mathematics (Daubechies, 1992), signal processing (Mallat, 1998), and statistics (Vidakovic, 1999). To illustrate the idea of a MAME, we study the problem of approximating a consecutive subsequence in a length-n sequence.

1. *Dyadic subsequence*. At scale k, where k is a non-negative integer, only consecutive subsequences that start from index $i2^k + 1$ and end with index $j2^k$ ($0 \le i < j$, i and j are integers) are considered. Each of these subsequences is called a dyadic subsequence. A subsequence is called a *basic dyadic*

subsequence if its starting and ending indices are $i2^k + 1$ and $(i+1)2^k$ $(i = 0, 1, 2, \ldots)$, hence the length of a basic dyadic subsequence is equal to 2^k. Apparently, a dyadic subsequence is a superposition of consecutive basic dyadic subsequences at the same scale. From now on, a subsequence is presumed to be consecutive unless it is specified differently.

2. *Number of dyadic subsequences at a fixed scale*. When the value of the scale indicator k increases, the number of dyadic subsequences at this scale decreases. For example, if the length of the entire sequence is $N = 2^J$, then at scale k, the total number of dyadic subsequences is

$$\begin{pmatrix} 2^{J-k} + 1 \\ 2 \end{pmatrix} = 2^{J-k}(2^{J-k} + 1)/2.$$

When k is large, there are a small number of subsequences to be considered. When $k = 0$, all possible subsequences are considered.

3. *Multiscale approximation*. From now on, a subsequence is called an *optimal subsequence*, if it optimizes an objective function among all possible subsequences. Our objective is to find such an optimal subsequence. The key idea of a MAME is that we start with dyadic subsequences at a coarse scale (when k is large); hopefully, dyadic subsequences that overlap with a large proportion of the optimal subsequence should have optimal values relative to other dyadic subsequences at the same scale. Based on this, an algorithm can be derived. Let us assume that we want to maximize a likelihood function. We start from a coarse scale, at which only dyadic subsequences whose likelihood are large are kept. These dyadic subsequences served as 'seeds'. Moving into a scale that is one scale finer (when k is reduced by 1), only dyadic subsequences that can be obtained by *slightly modifying* the seeds from the previous scale are considered. Here slightly modifying means addition or deletion of a few basic dyadic subsequences at the ends of a seed subsequence. The updated seeds are chosen by considering the value of a likelihood function on these dyadic subsequences. This process is repeated until the finest scale is reached. This approach ensures that at each scale, only a small number of subsequences need to be considered. Considering the number of scales is a logarithm of the length of the sequence, this approach is going to introduce a low complexity algorithm. If at every scale, a dyadic subsequence that can be grown into the optimal subsequence is kept, this search method will find the optimal solution.

1.4. Main results

In this paper, algorithms that are designed based on the above idea are presented. The key advantage is that because of the embedded tree structure that will be described later, the computational complexity of the derived methods can be as low as $O(n)$. This is dramatically lower than the one that is associated with a comprehensive search. Also, at the same time the new method finds the optimal solution with a probability that is close to one. (Note a comprehensive search can always find the optimal solution. Unfortunately, our new method does not guarantee that.) Simulations are performed to estimate the chances that a MAME finds the ultimate optimal answer. The empirical percentages from these simulations are very high. We argue that in the cases that MAME does not find the optimal solution, a statistically meaningful estimate may not exist.

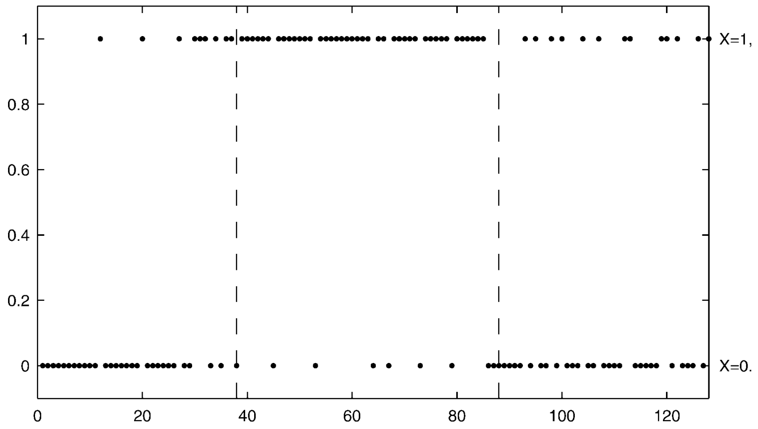


Fig. 1. A realization of high activity region model. In this case, $n = 128$, $i_0 = 38$, $j_0 = 88$, $p_0 = 0.3$, and $p_1 = 0.8$. The true underlying region is marked by two vertical dashed lines. Note that the boundary of the high activity region is vague.

1.5. Organization

This paper is organized as follows. In Section 2, formulation of these three problems are presented. In Section 3, the idea of multiscale approximation (MA), and an embedded binary tree is described in detail. In Section 4, the MAME algorithm is described, together with some remarks. In Section 5, some simulation results are used to show the effectiveness of the MAME approach. In Section 6, a comparison between the computing time of MAME and comprehensive searches is given. In Section 7, numerical experiments to study the effectiveness of the MAME are given. In Section 8, there is a brief description on the software that is used. In Section 9, some discussions are presented. Finally, we conclude in Section 10.

2. Formulation

We describe three statistical problems. The first and third problems are related to data mining and spatial statistics. The second one is from signal processing. For the third problem, some 2-d examples are studied by Allard and Fraley (1997). However, they use a different set of terminology.

For each problem, we start with its statistical model. Based on this model, an estimate is derived. The idea of MLE is adopted. Numerically finding these estimator requires solving an optimization problem. We would like to reiterate that all the objective functions of these optimization problems are functions of consecutive subsequences. These problems are ubiquitous in applications.

2.1. High activity regions

Amaratunga and Cabrera (2001) studied a problem of how to locate a region that the success probabilities are higher inside than the outside in a situation when the responses are Bernoulli. Such a region is called a *high activity region*.

We consider the following statistical model: we have a sequence of random variables $X_i, i = 1, 2, \ldots;$ each of them satisfies a Bernoulli distribution and for given $1 \leq i_0 \leq j_0 \leq n$, and $0 \leq p_0 < p_1 \leq 1$, we have

$$X_i \sim \begin{cases} B(p_0), & \text{if } i < i_0, \text{ or } i > j_0, \text{ and} \\ B(p_1), & \text{if } i_0 \le i \le j_0 \end{cases}.$$

In this model, the parameters are i_0, j_0, p_0, and p_1. To illustrate what the data may look like, we give a realization of this model in Fig. 1.

We now consider how to estimate p_0 and p_1. For a sequence of observed values, $x_1, x_2, \ldots, x_n$, its loglikelihood function is

$$\log L(i_0, j_0, p_0, p_1; x_1, x_2, \ldots, x_n) = \log(p_1) \sum_{i=i_0}^{j_0} x_i$$

$$+ \log(1 - p_1) \sum_{i=i_0}^{j_0} (1 - x_i) + \log(p_0) \sum_{i<i_0 \text{ or } i>j_0} x_i + \log(1 - p_0) \sum_{i<i_0 \text{ or } i>j_0} (1 - x_i).$$

For fixed i_0 and j_0, the following values of p_0 and p_1 maximize the loglikelihood function,

$$\hat{p}_1 = \sum_{i=i_0}^{j_0} x_i/(j_0 - i_0 + 1),$$

$$\hat{p}_0 = \sum_{i<i_0 \text{ or } i>j_0} x_i/(n - (j_0 - i_0 + 1)).$$

The loglikelihood function becomes

$$\log L(i_0, j_0, \hat{p}_0, \hat{p}_1; x_1, x_2, \ldots, x_n) = (j_0 - i_0 + 1)H(\hat{p}_1) + [n - (j_0 - i_0 + 1)]H(\hat{p}_0), \tag{2.1}$$

where $H(p) = p\log(p) + (1 - p)\log(1 - p)$. Note this loglikelihood is a function of i_0 and j_0. So searching for an MLE is equivalent to finding the optimizer for the function in Eq. (2.1). Note that our objective is to find the optimal solution without carrying out a comprehensive search.

2.2. An underlying boxcar function

In signal processing, sometimes we need to detect a boxcar function embedded in a noisy signal. We consider the following statistical model. The sequence, $X_1, X_2, \ldots X_n$, is a sequence of Gaussian random variables such that for fixed i_0 and j_0, $i_0 \le j_0$, we have

$$X_i \sim \begin{cases} N(0, 1), & \text{when } i < i_0 \text{ or } i > j_0, \text{ and} \\ N(\mu, 1), & \text{when } i_0 \le i \le j_0 \end{cases},$$

where $\mu > 0$ is an unknown but fixed parameter.

Now we consider its MLE. Given a sequence of observations, $x_1, x_2, \ldots, x_n$, for the loglikelihood

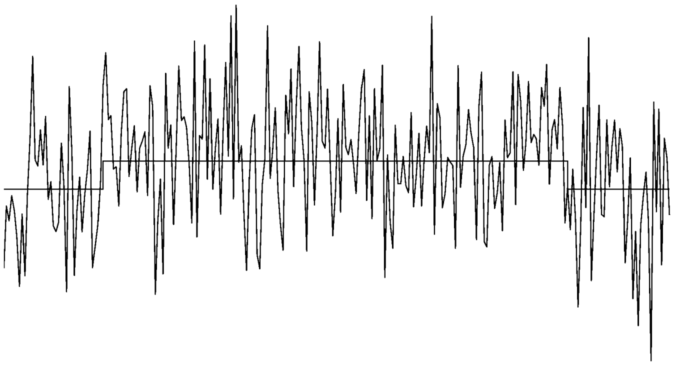


Fig. 2. A realization of our detection model. In this case, $n = 256$, $i_0 = 39$, $j_0 = 217$, and $\mu = 2\sqrt{2\log(n)/(j_0 - i_0 + 1)} \approx 0.50$. The piecewise horizontal curve is the underlying boxcar function. Note that relative to the noise level, the amplitude of the boxcar function is small.

function L, we have

$$-\log L(\mu, i_0, j_0; x_1, x_2, \ldots, x_n) \propto \sum_{i=i_0}^{j_0} \tfrac{1}{2}(x_i - \mu)2 + \sum_{i<i_0 \ or \ i>j_0} \tfrac{1}{2}x_i^2$$

$$= \tfrac{1}{2}\sum_{i=1}^{n} x_i^2 - \mu \sum_{i=i_0}^{j_0} x_i + \tfrac{1}{2}(j_0 - i_0 + 1)\mu^2.$$

Note that for fixed i_0 and j_0, the following value of μ maximizes the loglikelihood function,

$$\hat{\mu} = \sum_{i=i_0}^{j_0} x_i/(j_0 - i_0 + 1).$$

Substituting the μ by $\hat{\mu}$, the loglikelihood function becomes

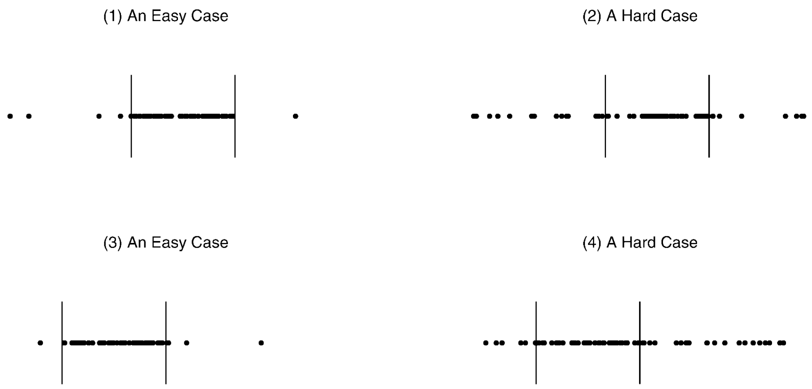
$$-2\log L(\hat{\mu}, i_0, j_0; x_1, x_2, \ldots, x_n) \propto \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=i_0}^{j_0} x_i\right)^2 /(j_0 - i_0 + 1).$$

Note that the above function is a function of i_0 and j_0. In this case, finding the MLE is equivalent to finding a pair of $(\hat{i}_0, \hat{j}_0)$ such that

$$(\hat{i}_0, \hat{j}_0) = \underset{i_0, j_0}{\operatorname{argmax}} \frac{\sum_{i=i_0}^{j_0} x_i}{\sqrt{j_0 - i_0 + 1}}. \tag{2.2}$$

Again, this becomes an optimization problem over all subsequences.

Note that this formulation enables us to detect a boxcar function in a very noisy situation. Recall that


(1) An Easy Case

(2) A Hard Case

(3) An Easy Case

(4) A Hard Case

Fig. 3. Four cases to illustrate the concept of high concentration area (HCA). The HCAs are marked by vertical lines. Each case has 128 data points.

in general, a random variable

$$R_{a,b} = \sum_{i=a}^{b} X_i/\sqrt{b-a+1}, \qquad \text{for } 1 \leq a \leq b \leq n,$$

is normally distributed with variance 1 and mean determined by how much this subsequence overlaps with the underlying subsequence and the value of μ. Assuming $\mu = 0$, we know that the maximum value of all $R_{a,b}$ is roughly equal to $\sqrt{2\log(n)}$ (Shao, 1995; Steinebach, 1998). When $\mu > 0$, $a = i_0$, and $b = j_0$, we have $R_{i_0,j_0} \sim N(\mu\sqrt{j_0 - i_0 + 1}, 1)$. In our model, as long as $\mu\sqrt{j_0 - i_0 + 1} \gg \sqrt{2\log(n)}$, we can reliably declare that the underlying signal is going to be detected. Note that this means the value of μ can be very small. It can be so small that if we only look at the X_i's on the boundary of the underlying subsequence, there is no hope to detect it. This demonstrates the advantage of this formulation over some local information based methods. To illustrate, a noisy sequence is given in Fig. 2.

2.3. High concentration areas

In some cases in practice, we see that there are regions in which data are more concentrated than the rest of the space. Some artificial examples are given in Fig. 3.

One difficulty with this problem is how to quantitatively define a *high concentration area* (HCA). Similar problems have been studied in Allard and Fraley (1997) and Chernoff and Rubin (1954). We adopt the following statistical model: for a sequence of independently and identically distributed random variables X_i, $i = 1, 2, \ldots, n$, each of them has the following probability density function,

$$\text{for } 0 \leq a < b \leq 1,\ \mu_1 > \mu_2 > 0, \qquad f(x) = \begin{cases} \mu_1, & \text{if } x \in [a,b], \\ \mu_2, & \text{if } x \notin [a,b] \end{cases}.$$

To make $f(\cdot)$ a probability distribution, we must have $\int_0^1 f(x)\mathrm{d}x = 1$, hence

$$\mu_1(b-a) + \mu_2[1 - (b-a)] = 1. \tag{2.3}$$

Apparently in the above definition, we assume that the common domain of X_i's is [0,1]. The probability density function is piecewise constant in intervals $[0,a)$, $[a,b]$, and $(b,1]$. Because $\mu_1 > \mu_2$, interval

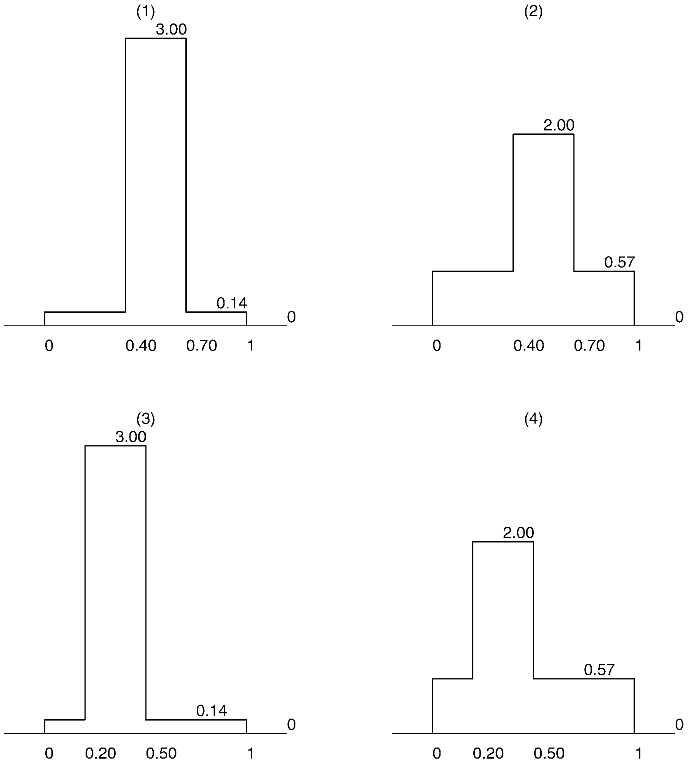


Fig. 4. Four probability density functions of our models for high concentration areas. Their parameters are in the graphs. For example, in (1), we have $\mu_1 = 3.00$, $a = 0.40$, and $b = 0.70$. They are the distributions by which the data points in Fig. 3 are generated.

$[a, b]$ is an area that has *high concentration*; we call it a HCA. Fig. 4 gives some illustrations of these probability density functions.

Now we derive its MLE. For fixed a and b, the loglikelihood is

$$\log L(a, b, \mu_1, \mu_2; x_1, x_2, \ldots, x_n) = \tau_1 \log(\mu_1) + \tau_2 \log(\mu_2),$$

where τ_1 and τ_2 are numbers of observations inside and outside the interval $[a, b]$, respectively. Taking into account of condition (2.3), the loglikelihood is maximized when

$$\mu_1 = \hat{\mu}_1 = \frac{\tau_1/n}{(b-a)},$$

and

$$\mu_2 = \hat{\mu}_2 = \frac{1 - \tau_1/n}{1-(b-a)}.$$

Hence the loglikelihood function becomes

$$\frac{1}{n}\log L(a, b, \hat{\mu}_1, \hat{\mu}_2; x_1, x_2, \ldots, x_n) = \frac{\tau_1}{n}\log\left(\frac{\tau_1/n}{b-a}\right) + \left(1 - \frac{\tau_1}{n}\right)\log\left(\frac{1-\tau_1/n}{1-(b-a)}\right). \tag{2.4}$$

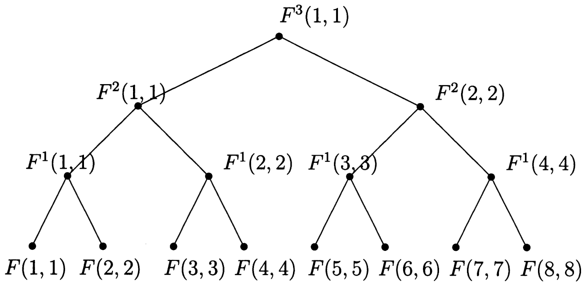

$F^3(1,1)$
$F^2(1,1)$
$F^2(2,2)$
$F^1(1,1)$
$F^1(2,2)$
$F^1(3,3)$
$F^1(4,4)$
$F(1,1)$ $F(2,2)$ $F(3,3)$ $F(4,4)$ $F(5,5)$ $F(6,6)$ $F(7,7)$ $F(8,8)$

Fig. 5. A binary tree when $n = 8$. The support of a parental node is a combination of supports of its two children.

Note that the above is a function of a and b. As a matter of fact, it is the same as the relative information between Bernoulli (τ_1/n) and Bernoulli $(b - a)$, (Cover & Thomas, 1991). It is also known as the Kullback Liebler distance.

Our next objective is to use the idea of MA to efficiently estimate the values of a and b that maximize the loglikelihood function in Eq. (2.4). Note that it is not trivial to optimize the function in Eq. (2.4). We may restrict ourselves to a condition that values of a and b coincide with values taken by the random sequence X_i's. However, it still seemingly requires a comprehensive search, which is not desirable. We next illustrate how MA can help to find an efficient solution.

3. Multiscale approximation and a binary tree

From Eqs. (2.1), (2.2) and (2.4), it is evident that in these optimization problems, all objective functions are functions of subsequences. Hence only the starting and ending positions of a subsequence matter. The following notation is introduced. Let $F(i,j)$, $1 \le i \le j \le n$, denote the value of an objective function when the subsequence starts at i and ends at j. The *support* of this function is a consecutive integer sequence $\{i, i+1, \ldots, j\}$. Furthermore, for $1 \le k \le \log_2(n)$, we define

$$F^0(i,j) = F(i,j),$$

and

$$F^k(i,j) = F^{k-1}(2i-1, 2j), \qquad 1 \le i \le j \le n/2^k.$$

We can easily find out that the support of $F^k(i,j)$ is $\{(i-1)2^k + 1, (i-1)2^k + 2, \ldots, j2^k\}$, which is a dyadic subsequence at scale k. To illustrate, a binary tree structure is introduced. It is a tree made by all items like $F^k(i,i), 0 < k \le \log_2(n), 1 \le i \le n/2^k$. The support of these items are basic dyadic subsequences. An illustration of such a tree in the case when $n = 8$ is given in Fig. 5.

Based on the binary tree structure, the idea of MA can be described as follows. One level is chosen as a starting point, for example level $k_0, 0 \le k_0 \le \log_2(n)$. We consider all $F^{k_0}(i,j), 1 \le i \le j \le n/2^{k_0}$ at this level. There are $[(n/2^{k_0} + 1)n/2^{k_0}]/2$ of them. When k_0 is large, the number of $F^{k_0}(i,j)$ has low order of complexity. For example, if $k_0 \ge 1/2 \log_2(n)$, there are less than n terms to consider. The function values of all of them are computed, and are sorted. When going down one level, firstly, only those F's who have relatively more optimal values at the upper level are kept; secondly, at the current level, an F is considered only if it can be *grown* out of an F inherited from the upper level. Here one F can be grown out of another F if the former has a support that is generated by deletion or addition a basic dyadic subsequence from the support of the latter. For example, if $F^k(i,j)$ is selected at the upper level, at the current level, any F that is one of $F^{k-1}(2i-2, 2j-1)$, $F^{k-1}(2i-2, 2j)$, $F^{k-1}(2i-2, 2j+1)$, $F^{k-1}(2i-1, 2j-1)$, $F^{k-1}(2i-1, 2j)$, $F^{k-1}(2i-1, 2j+1)$, $F^{k-1}(2i, 2j-1)$, $F^{k-1}(2i, 2j)$, or $F^{k-1}(2i, 2j+1)$ is considered an F-term that grows out of $F^k(i,j)$. At the lower level, we have at most nine times the number of F's selected in the upper level. The reason we can restrict ourselves to these F's is because based on the MA, only these F's whose supports overlap with the underlying optimal subsequence have optimal (minimal or maximal, depending on the optimization problem) values. These F's are sorted and the best few are selected. Then we go down one level. This process is repeated until the lowest level is reached. If the number of F's that are kept at each level is carefully chosen, then it is possible that a derived algorithm has low complexity. The computational complexity can be as low as $O(n)$. In Section 4, an algorithm is formally described.

To facilitate further analysis, the conditions of the success of a MAME are summarized as follows.

1. In a high level, the F's whose support overlap with the underlying optimal subsequence must have optimal values.
2. The growing (inheritance) procedure introduced above must keep the promising F's.

Moreover, a subsequence (associated with $F^{k_1}(i_1,j_1)$) is called a *root subsequence* of another subsequence (associated with $F^{k_2}(i_2,j_2)$, if and only if $F^{k_2}(i_2,j_2)$ can be grown from $F^{k_1}(i_1,j_1)$ after following several steps of adding and deleting basic dyadic subsequences. Recall that a subsequence is an optimal subsequence, if it optimizes the value of function $F(i,j)$. So the key for the success of an MAME is that at each step, a root subsequence(s) of the optimal subsequence is (are) preserved.

4. A tree-based algorithm

We now give a formal description of our algorithm. Note that this is a simple extension of previous discussion. Without loss of generality, in the following description, we assume that the value of $F(i,j)$ needs to be maximized.

4.1. A multiscale approximation method

1. Start with $k_0 = \lceil 1/2 \log_2(n) \rceil$, where $\lceil \cdot \rceil$ is the minimum integer that is not less than the input variable. All possible $F^{k_0}(i,j)$'s are computed, and the largest $\lfloor n/\log_2(n) \rfloor$ are kept, where $\lfloor \cdot \rfloor$ is the maximum integer that is no larger than the input variable. Then move down one level.
2. For the $\lfloor n/\log_2(n) \rfloor$ F-terms that are inherited from the above level, we consider F-terms at the current level that can be grown out of the inherited ones. As we described earlier, we shall have at most $9\lfloor n/\log_2(n) \rfloor$ of them. We compute their values, retain the ones associated with the largest $\lfloor n/\log_2(n) \rfloor$ values, and move on to the next lower level. This process is repeated until the bottom level is reached.
3. The subsequence that gives the maximum value at the last step is our estimate.

The value k_0 is chosen so that the total number of subsequences is not more than n. The value $\lfloor n/\log_2(n) \rfloor$ is chosen so that the complexity of sorting them is not more than $O(n)$.

Assuming that computing the value of an F function takes a constant number of operations, i.e. $O(1)$,

Table 1
The numbers of mismatches between an MAME and a comprehensive search method. In each case, 1000 simulations are carried out. The number in each cell is the number of mismatches

HAR	(p_0,p_1)	(i_0,j_0)					
		(39, 217)	(97, 217)	(144, 217)	(38, 160)	(90, 160)	(39, 90)
	(0.1, 0.8)	9	4	14	7	8	16
	(0.2, 0.7)	47	34	36	25	32	48
	(0.3, 0.6)	114	77	113	88	93	156
	(0.4, 0.6)	281	239	222	238	272	290
	(0.4, 0.5)	364	376	374	372	384	383
Boxcar	τ	(39, 217)	(97, 217)	(144, 217)	(38, 160)	(90, 160)	(39, 90)
	2	92	70	69	86	60	60
	1.5	135	100	99	131	108	121
	1	268	268	225	241	227	262
HCA	μ_1	(a,b)					
		(0.4, 0.7)	(0.2, 0.5)				
	3.0	0	0				
	2.5	8	3				
	1.7	43	43				
	1.3	779	757				

the above algorithm has at most $O(n)$ complexity. We list a result here, and leave the proof as a simple exercise for our readers.

Theorem 4.1. *If on average it takes O(1) operations to compute each F, then a MAME has O(n) computational complexity.*

Note that in our case, for the three optimization problems, this condition is satisfied: the value of their objective function can be calculated by a constant number of steps. Hence an MAME results on an $O(n \log(n)$ algorithm.

5. Simulations

Simulations are designed to evaluate the effectiveness of an MAME in finding the exact global optimizer. For each model, several groups of parameters are chosen. In each case, 1000 simulations are carried out. In each simulation, a sequence is generated by following the corresponding statistical model. For each generated sequence, first a comprehensive search is applied to find the optimal subsequence, then an MAME is used to see if it finds the same subsequence. The numbers of mismatches are reported.

Let us describe more on the choice of parameters. Recall that in the model for high activity regions (HAR), there are five parameters: n, i_0, j_0, p_0, and p_1. In the model for detecting an underlying boxcar function (Boxcar), there are four parameters n, i_0, j_0, and μ. In the model for HCA, there are five parameters n, a, b, μ_1, and μ_2. Some of the above parameters can be omitted. In Boxcar, taking into

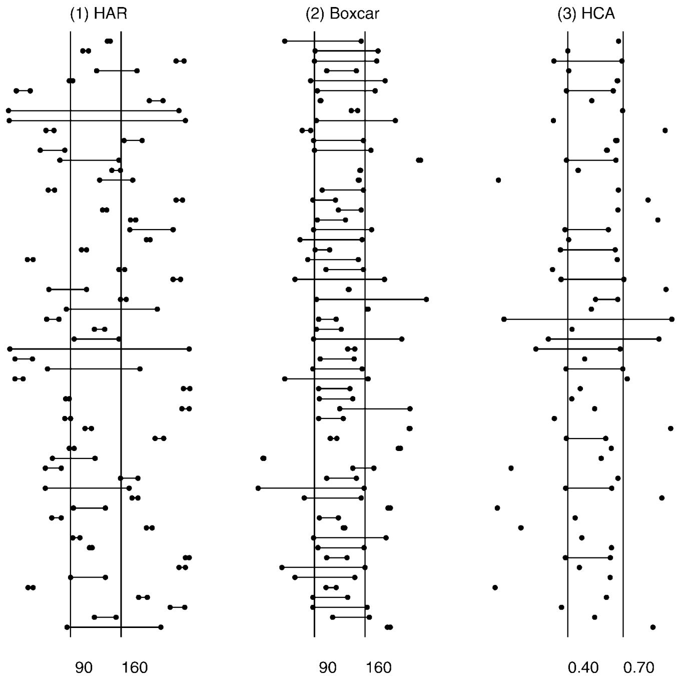


Fig. 6. The performance of the comprehensive search method in some cases that underlying objects are hard to detect. See value of parameters in the context.

account of the detectability of an underlying function, we can take $\mu = \tau\sqrt{2\log(n)/(j_0 - i_0 + 1)}$. Hence only the values of n, i_0, j_0, and τ need to be specified. In HCA, because of Eq. (2.3), value of μ_2 is determined by other parameters. The following simulations are designed. In HAR, the pair of (i_0, j_0) can be one of

$$(39, 217),\quad (97, 217),\quad (144, 217),\quad (38, 160),\quad (90, 160),\ \text{or}\ (39, 90).$$

The pair of (p_0, p_1) can be one of

$$(0.1, 0.8),\quad (0.2, 0.7),\quad (0.3, 0.6),\quad (0.4, 0.6),\ \text{or}\ (0.4, 0.5).$$

All together, there are 30 cases. In Boxcar, the pair (i_0, j_0) take the same values as in HAR. The value of τ can be one of

$$2.0,\quad 1.5,\ \text{or}\ 1.0.$$

So there are 18 cases for Boxcar. In HCA, the pair (a, b) can be one of

$$(0.4, 0.7),\ \text{or}\ (0.2, 0.5).$$

The value of μ_1 can be one of

$$3.0,\quad 2.5,\quad 1.7,\ \text{or}\ 1.3.$$

So there are eight cases for HCA. In all simulations, the length of the sequence is set to be $n = 256$.

The number of mismatches are reported in Table 1. In Table 1, the following can be observed. In each

Table 2
Average execution times of MAMEs and comprehensive search algorithms. The second row gives the lengths of the signals. The problem is Boxcar. The times are in seconds. Apparently when the data size is large, the MAME is significantly faster

	n			
	256	1024	4094	8192
Comprehensive search (T_1)	3.36	54.69	1283.61	7903.19
MAME (T_2)	0.38	1.42	4.73	8.32
Ratio (T_1/T_2)	9.91	44.61	275.22	946.31

model (e.g. HAR, Boxcar, or HCA), from the top to the bottom, the underlying subject becomes more and more difficult to detect. In the case when an underlying function is easy to detect, the results of an MAME and a comprehensive search coincide in most cases. However, when an underlying subject is difficult to detect, there are significant number of (several hundred) mismatches.

Fortunately, the abundant mismatches in the hardly detectable cases do not hurt the usability of MAMEs. In these cases, the results of a comprehensive search method are not necessarily close to the true underlying object either. This is illustrated in Fig. 6. The three cases that are considered are:

1. in HAR, $i_0 = 90$, $j_0 = 160$, $p_0 = 0.4$, and $p_1 = 0.5$;
2. in Boxcar, $i_0 = 90$, $j_0 = 160$, and $\tau = 1$; and
3. in HCA, $a = 0.4$, $b = 0.7$, and $\mu_1 = 1.3$.

In each case, 60 random sequences are generated, and their MLE (calculated by a comprehensive search algorithm) are plotted. Each horizontal line segment represents an MLE—its x-coordinate starts from $\hat{a}$ and ends at $\hat{b}$. Note that some horizontal line segments are so short that they appear as points in the figure. The true underlying objects are marked by two red vertical lines. It is easy to observe that there is significant difference between the true underlying objects and their MLE. In these cases, the MLEs are not optimal estimates. So there is no need to find a fast approximation method. In the cases when MLE is a good estimate, an MAME does a good approximation.

6. Computing time

In Table 2, the computing times of an MAME and a comprehensive search algorithm for detecting a boxcar function are compared. Both methods are implemented in MATLAB. Each value is an average computing time of 10 simulations.

Note that it is also possible to count the exact number of operations that are required in each algorithm. However, we choose to present their average execution time, because this presentation seems more intuitive.

7. When does a multiscale approximation method work?

Now we study when the idea of MA works. Evidently, if an MAME works, then at least one root subsequence of the *optimal* subsequence should be retained at each step of an MAME. Here ‘one step’

Table 3
The number of cases (out of 1000 simulations) that the ‘best order’ statistics are larger than the threshold $\lceil n/\log_2(n)\rceil$. When $n = 256$, there are four levels: $k = 1,2,3$, and 4. In each cell, the four numbers (number 1/number 2/number 3/number 4) are the numbers of cases that the best order statistics (at the corresponding levels) are above the threshold

HAR	(p_0,p_1)	(i_0,j_0)					
		(39, 217)	(97, 217)	(144, 217)	(38, 160)	(90, 160)	(39, 90)
	(0.1, 0.8)	0/0/0/0	0/0/0/0	0/0/0/0	0/0/0/0	0/0/0/0	0/0/0/0
	(0.2, 0.7)	0/0/0/0	0/0/0/0	0/0/0/0	0/0/0/0	0/0/0/0	0/0/0/0
	(0.3, 0.6)	2/9/16/12	0/4/6/4	2/9/4/12	0/5/11/5	0/10/26/14	1/21/31/28
	(0.4, 0.6)	4/52/69/71	3/44/80/80	1/42/83/80	6/50/70/71	1/43/73/75	4/55/90/95
	(0.4, 0.5)	11/82/151/172	4/63/169/184	3/76/139/168	8/64/177/192	1/62/151/169	3/55/146/167
Boxcar	τ	(39,217)	(97,217)	(144,217)	(38,160)	(90,160)	(39,90)
	2	2/1/0/0	0/0/0/0	2/0/0/0	2/1/0/0	0/0/0/0	0/0/0/0
	1.5	6/11/9/11	15/19/27/19	9/9/7/4	5/10/14/7	8/11/9/8	2/13/8/5
	1	45/91/126/119	42/75/110/113	46/82/108/69	45/84/106/97	43/84/104/84	42/61/77/58
HCA	μ_1	(a,b)					
		(0.4, 0.7)	(0.2, 0.5)				
	3.0	0/0/0/0	0/0/0/0				
	2.5	0/0/0/0	0/0/0/0				
	1.7	3/9/9/7	7/15/15/14				
	1.3	319/684/696/624	300/671/702/624				

stands for the process that is carried out at one level. As a matter of fact, the following result can be proved easily. We state it without a proof.

Theorem 7.1. *Let subsequence a be the optimal subsequence, and assume that the optimal subsequence is unique. The MAME finds the optimal subsequence, if and only if at each step of an MAME, at least one root subsequence of subsequence a is retained.*

The above result is very restrictive. It depends on the procedure of an MAME. One can examine the MA in a broader sense. For example, at one level, among all root subsequences, the best order statistic can be considered. Here the best order is the rank of the more optimal value given by a root subsequence. For example, if a root subsequence gives the most optimal value at one level, then the best order statistic at this level is equal to 1. The ‘best order’ statistics at all levels are good measures to evaluate when an MAME can work, in the sense that when the best order is large, it is likely that no MAMEs can work, because no method can find a root of the optimal subsequences at a coarser scale.

Simulations are designed to calculate the best order statistics. For simplicity, the setup of these simulations is the same as in the previous simulation that is described in Section 5.

The following tables summarize the results. In Table 3, the number of cases (out of 1000 simulations) in which the best order statistics are larger than a MAME threshold $\lfloor n/\log_2(n)\rfloor$ are given. In Tables 4 and 5, the 90-percentile and the 99-percentile are reported, respectively.

The following can be observed. For each model, when an underlying object becomes more difficult to be detected, the numbers of incidences that the best order statistics exceeding the prefixed threshold

Table 4
The 90-percentile of the best order statistics. There are four levels: $k = 1, 2, 3$, and 4. In each cell, the four numbers (number 1/number 2/number 3/number 4) are the 90-percentile at the corresponding levels

HAR	(p_0,p_1)	(i_0,j_0)					
		(39, 217)	(97, 217)	(144, 217)	(38, 160)	(90, 160)	(39, 90)
	(0.1, 0.8)	2/1/1/1	2/1/1/1	2/1/1/1	2/1/1/1	2/1/1/1	2/1/1/1
	(0.2, 0.7)	3/2/2/1	3/2/1/1	3/2/1/1	3/2/1/1	3/2/1/1	3/3/2/1
	(0.3, 0.6)	4/4/4/3	4/4/3/3	4/4/3/3	5/4/4/3	4/4/4/4	4/5/5/6
	(0.4, 0.6)	5/14/19/21	5/10/21/26	5/12/24/27	5/12/17/23	4/12/22/26	5/14/28/33
	(0.4, 0.5)	5/23/61/54	4/21/76/58	5/24/48/52	5/19/71/56	4/21/57/55	4/17/57/52
Boxcar	τ	(39,217)	(97,217)	(144,217)	(38,160)	(90,160)	(39,90)
	2	4/4/3/3	4/4/3/2	4/3/2/2	4/4/3/2	3/3/2/2	3/3/2/2
	1.5	5/6/4/4	5/5/5/3	4/4/3/3	5/5/4/3	5/4/3/3	3/4/3/3
	1	7/22/66/49	8/19/45/39	7/19/36/20	7/19/39/31	9/20/39/28	6/13/16/15
HCA	μ_1	(a,b)					
		(0.4, 0.7)	(0.2, 0.5)				
	3.0	1/1/1/1	1/1/1/1				
	2.5	1/1/1/1	1/1/1/1				
	1.7	3/3/2/1	3/3/2/1				
	1.3	3612/1615/463/121	2832/1595/449/120				

increase, and so do the percentiles. Hence an MA based method becomes less likely to be effective. In fact, when the best order statistic is large, one can say that it is just hopeless to approximate the optimal subsequence from a coarse level. This is consistent with the results in the table of mismatches (Table 1).

8. Software

The software that is used to carry out the simulations in this paper is available from the author. They are some MATLAB functions. The following three are the key functions that have been used. For a free copy, please send an email request to the author (xiaoming@isye.gatech.edu).

- MAME_HAR—a MAME to search for an underlying subsequence, For the model of high activity region.
- MAME_boxcar—a MAME to search for an underlying interval, For the model of boxcar.
- MAME_HCA—a MAME to search for an underlying subsequence, for the model of HCA.

9. Discussion

9.1. Theoretical analysis

A MLE has many well-known nice statistical properties, for example, efficiency, etc. It is also known

Table 5
The 99-percentile of the best order statistics. There are four levels: $k = 1,2,3$, and 4. In each cell, the four numbers (number 1/number 2/number 3/number 4) are the 99-percentile at the corresponding levels

HAR	(p_0, p_1)	(i_0, j_0)					
		(39, 217)	(97, 217)	(144, 217)	(38, 160)	(90, 160)	(39, 90)
	(0.1, 0.8)	5/3/2/1	5/3/2/1	5/3/1/1	5/2/2/1	4/2/2/1	3/3/2/2
	(0.2, 0.7)	8/6/3/3	7/4/3/2	8/6/3/2	8/5/3/2	8/5/3/2	9/6/3/4
	(0.3, 0.6)	17/30/48/48	13/16/30/20	12/28/24/37	14/17/40/21	14/38/64/38	16/75/66/53
	(0.4, 0.6)	24/110/171/81	19/99/157/84	23/132/155/77	29/107/210/79	22/89/176/87	22/116/210/86
	(0.4, 0.5)	35/203/256/115	22/142/263/105	18/156/244/103	22/154/248/112	21/147/252/118	18/116/301/112
Boxcar	τ	(39, 217)	(97, 217)	(144, 217)	(38, 160)	(90, 160)	(39, 90)
	2	22/16/12/7	21/11/8/6	13/12/6/5	13/17/10/5	12/7/6/5	8/9/6/5
	1.5	25/45/45/44	79/158/93/55	31/31/21/16	23/36/52/30	30/38/28/31	17/47/25/25
	1	931/823/363/118	1,034/859/325/109	650/669/324/90	650/844/333/118	614/706/325/102	439/565/278/100
HCA	μ_1	(a,b)					
		(0.4, 0.7)	(0.2, 0.5)				
	3.0	2/1/1/1	2/1/1/1				
	2.5	3/2/1/1	3/2/1/1				
	1.7	7/25/13/7	15/1081/283/62				
	1.3	6979/2016/522/134	6505/2010/521/134				

that when the underlying subject has no statistical significance, then no MLE type of estimate can work. Our simulations indicate that when in a statistical sense, the underlying subject is detectable, then an MAME can work as well as an MLE method does. Apparently more theoretical analysis can be done in this direction. We leave it as our future research.

9.2. Impact in statistical modeling

In applied statistics and signal processing often it is more appropriate to form models that take subsequences or intervals as underlying objects. Unfortunately, people have tried not to do so, probably due to the fear of the high complexity of a comprehensive search algorithm. Introducing a low complexity approximation method will help to promote this kind of statistical models. This will enforce our arsenal of modeling tools.

9.3. Impact in data mining

A key idea in developing this method is to find a low complex substitute of an otherwise too expensive method. Nowadays, because of the advances in data acquisition, the size of the datasets that people encounter increase dramatically. In a case with a large dataset, an $O(n^2)$ method can be completely infeasible. An $O(n \log(n)$ approximation method is a better option. Our method can provide solutions to difficult problems in data mining.

9.4. Generalization

So far, only 1D cases are considered. The same principle can be applied to 2D cases, or cases with even higher dimensions. An example is the minefield detection problem that is described in Allard and Fraley (1997), which is basically a 2D version of an HCA problem. Introducing a 2D MAME will be our future research.

10. Conclusion

A MAME to estimate an underlying subsequence is proposed. Computationally, this new method is significantly faster. It has much lower order of complexity. Simulations show that when the underlying object is statistical detectable, the new method can find the same optimal solution as a MLE does. The new method has good application in the mining of interest regions in *large* datasets.

References

Allard, D., & Fraley, C. (1997). Nonparametric maximum likelihood estimation of features in spatial point processes using voronoï tessellation. *Journal of American Statistical Association*, 1485–1493.

Amaratunga, D., Cabrera, J (2001). *Mining data to find subsets of high activity*. Technical report, Rutgers University.

Chernoff, H., & Rubin, H. (1954). *The estimation of the location of a discontinuity in density*. *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, CA: University of California Press, pp. 19–37.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York: Wiley.

Daubechies, I. (1992). *Ten lectures on wavelets*. Philadelphia, PA: SIAM.

Mallat, S. (1998). *A wavelet tour of signal processing*. San Diego: Academic Press.

Sadler, B., & Swami, A. (1999). Analysis of multiscale products for step detection and estimation. *IEEE Transactions on Information Theory*, *45*(3), 1043–1051.

Shao, Q.-M. (1995). *On a conjecture of révész* (*123*). *Proceedings of American Mathematical Society*, pp. 575–582.

Steinebach, J. (1998). *On a conjecture of révész and its analogue for renewal processes*. *Asymptotic methods in probability and statistics*, New York: Elsevier, pp. 311–322.

Vidakovic, B. (1999). *Statistical modeling by wavelets*. New York: Wiley.