Joint Segmentation and Classification of Time Series Using Class-Specific Features

Zhen Jane Wang and Peter Willett, *Fellow, IEEE*

***Abstract*—We present an approach for the *joint segmentation and classification* of a time series. The segmentation is on the basis of a menu of possible statistical models: each of these must be describable in terms of a sufficient statistic, but there is no need for these sufficient statistics to be the same, and these can be as complex (for example, cepstral features or autoregressive coefficients) as fits. All that is needed is the probability density function (PDF) of each sufficient statistic under its own assumed model—presumably this comes from training data, and it is particularly appealing that there is no need at all for a *joint* statistical characterization of all the statistics. There is similarly no need for an a-priori specification of the number of sections, as the approach uses an appropriate penalization of an over-zealous segmentation.**

The scheme has two stages. In stage one, rough segmentations are implemented sequentially using a piecewise generalized likelihood ratio (GLR); in the second stage, the results from the first stage (both forward and backward) are refined. The computational burden is remarkably small, approximately linear with the length of the time series, and the method is nicely accurate in terms both of discovered number of segments and of segmentation accuracy. A hybrid of the approach with one based on Gibbs sampling is also presented; this combination is somewhat slower but considerably more accurate.

***Index Terms*—Classification, class-specific, GLR, order selection, segmentation.**

I. Introduction

A. Motivation

SUPPOSE we have M classes characterized by different parameter vectors, and we observe a time series that consists of sections generated by iteratively selecting a class. How can we use the time series to recover the class index, model parameters, and the transition points between sections? In other words, we are interested in the joint segmentation and classification of such a time series.

The segmentation of time series into sections arising from different statistical models (classes) has received reasonably extensive research attention [5]. The problem has quite wide applications, from discrete Fourier transform (DFT) data partitioning [20], to ionic-channel detection [26], to speech treatment and understanding [24], to audio content analysis [34]. Of particular interest to us is the posing of the acoustic transient detection problem (e.g., [14], [17], [25]), as one of segmentation. Notionally, we are given a time series that may be given by, for example, an autoregressive (AR) model of low complexity if it is "ambient" and free of interesting artifacts; but it may contain a short-duration signal (or even, perhaps, more than one) whose presence indicates a situation of interest. One can assume that there has been some effort to model each sort of such transient signal, both in terms of selection of some sort of statistics that might be thought "sufficient" (mathematically, we shall assume that they are just so) and in terms of characterizing statistically the distribution of each such statistic based on some training data. At a more specific level, perhaps one transient type is best represented as a medium-order AR process, two others as autoregressive moving average (ARMA) process, a fourth in terms of specific short-time Fourier transform (STFT) coefficients, and several others in terms of their multi-resolution coefficients; and one can assume that each of these "sufficient" statistics has, for example, a vector Gaussian-mixture probability density function (PDF) that approximates it when the transient is truly present. How do we segment this to locate and characterize transient signals?

We have worked on a transient detection problem by assigning sections of a *white* time series to those parts with governing PDF's either f_0 or f_1, where the PDF's f_0 and f_1 represent the signal-absent and signal-present hypotheses respectively. This segmentation problem is hard even when the number of sections is known, since one needs, in principle, to investigate all breakpoint combinations. When the number of sections is unknown, the problem is harder still: given that more sections means a better match, how many sections should there be? But a computationally efficient solution even for that problem is available in [32], and has the additional feature that each f_1 segment (i.e., where transient exists) can have a different scale (variance) parameter.

In this paper, however, the problem is harder still: the segments are not necessarily white, and may not even be directly comparable: they can be described by different parameter vectors, such as white versus AR(3) versus (another) AR(3) versus AR(6). Let us assume, temporarily, that the number of segments is known. Then one can pose the problem of maximizing, over all possible feasible combinations of breakpoints and models, the joint likelihood of all statistics. One's popularity amongst users is unlikely to increase after one does this, however: not only is the maximization ridiculous from a computational viewpoint, but, except in very limited situations, one would never have such a joint likelihood[1].

We provide here a computationally efficient solution via modification on the techniques of [32], there applied to the simple

Manuscript received November 11, 2002; revised May 16, 2003. This work was supported by NUWC through Contract N66604-01-1-1125. The paper was recommended by Associate Editor I. Bloch.

Z. Wang is with the Electrical and Computer Department, University of Maryland, College Park, MD 20742 USA (e-mail: wangzhen@glue.umd.edu).

P. Willett is with the Electrical and Computer Department Department, University of Connecticut, Storrs, CT 06269 USA (e-mail: willett@engr.uconn.edu).

Digital Object Identifier 10.1109/TSMCB.2003.819486

[1] This really is a significant practical concern: perhaps one *can* estimate a PDF of coefficients from a fourth-order AR model based on an obtainable amount of training data. But if that is augmented by third-order ARMA coefficients, by cepstral values, and by a few selected STFT outputs, the resulting vector is unrealistically large ever to be faithfully represented by any estimated PDF.

variance-shift segmentation problem. There is no need for the above "temporary" assumption of a known number of segments: a likelihood penalty term provides the appropriate punishment of over-segmented descriptors. Further, and perhaps most important, we avoid joint estimation of a massive feature vector via Baggenstoss' new "class-specific" statistical descriptors for noncomparable hypotheses [3]. There are good segmenters for white data, there are techniques for segmentation of nonwhite data (e.g., [5]), and there are ways to segment based on data-derived statistics whose properties are well-modeled globally, for all segment types. But in many practical problems these properties must be estimated, and hence the class-specific approach holds great appeal since each statistic needs only to be estimated over its "own" hypothesis. We have seen little treatment[2] of segmentation based on nonglobal statistics: that is this paper's contribution.

B. Background

There have been numerous attempts to solve segmentation and classification problems. From the *optimization point of view*, there are optimal and sub-optimal approaches. The usual statistically *optimal* criterion is the maximum likelihood (ML), although some researchers do prefer least-squares to avoid specifying the distribution of the process [22]. To reduce the computational load of an exhaustive breakpoint evaluation, assuming independence among segments, an optimal search can be realized via dynamic programming (DP) [10], [27] or simulated annealing [21]. With the ML approach, the number of segments can be automatically chosen by the minimum description length (MDL) rule as in [20], [18]. However, though the computational burden is cleverly reduced vis-a-vis a direct search, the DP implementation is still computationally expensive and in practice formidable, especially when the number of data, segments and/or models is high. Therefore, in practice, *suboptimal* methods based on sequential estimation have been studied to lighten the computational complexity [9], [8], [6], [2]. There is no likelihood calculated as the criterion in most suboptimal approaches, due to both the modeling and computational difficulty. For example, a heuristic rule-based procedure is proposed to segment and classify audio signals in [34]; and simple thresholding segmentation with neural classification is applied to speech analysis in [24]. Reduced performance is observed in those implementations, and various tunable thresholds are required whose choice tends to be empirical.

From the *procedural point of view*, there can be joint segmentation/classification, or two-step (either implementing segmentation or classification first) approaches. Examples of the former are the joint segmentation and recognition of phonemes using the stochastic segment model (SSM) [12], and the DP recursion in [27]. In most cases, however, algorithms work sequentially in segmentation and classification steps: the signals are segmented first according to some statistic, such as power, that does not require a precise statistical description of hypotheses; then the resultant segments are classified [24], [26], [34]. For example, a classification plus segmentation procedure is presented in [23] in which signals are first divided into fixed size windows; then each window is classified; then consecutive windows of the same types are merged into a segment. Therefore, the procedure is really problem and criteria dependent – a perfectly reasonable engineering expedient, but not particularly optimal.

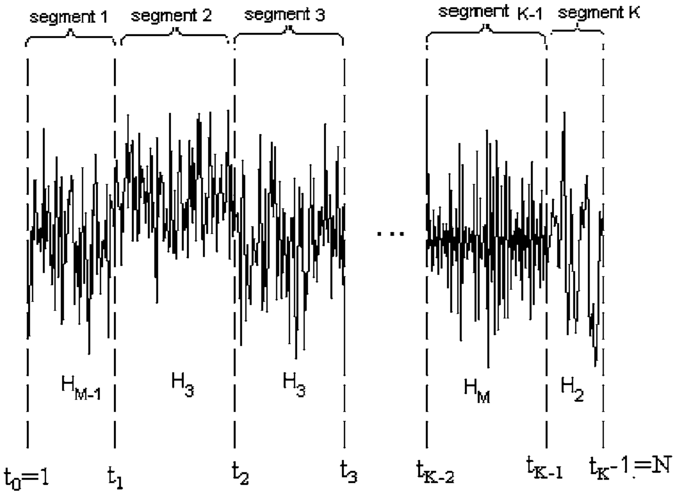

Fig. 1. Illustration of segmentation problem.

C. Modeling

In this paper we are interested in a *joint* segmentation and classification procedure, and we apply the likelihood as the criterion. A segmentation and classification problem of interest here is depicted in Fig. 1. Let the time series $\{x_1, x_2, \ldots, x_N\}$ be composed of K segments with $K-1$ transition times $\tau = \{t_1, t_2, \ldots, t_{K-1}\}$. The data within segment i is assumed to be a realization of model m_i, where $1 \leq m_i \leq M$, and thus be characterized by the PDF $p(x_{t_{i-1}}, \ldots, x_{t_i-1}|H_{m_i})$. Write $\mu = \{m_1, m_2, \ldots, m_K\}$, indicating the class indices of all segments. Thus, determination of unknown K, τ and μ is formulated as a joint segmentation and classification problem. As is we hope reasonable, we know as a priori information that the segments satisfy

$$L_{\min} \leq t_i - t_{i-1} \leq L_{\max} \tag{1}$$

meaning that segments are of lengths restricted between $L_{\min}$ and $L_{\max}$. (This constraints can be trivialized out of existence with no theoretical price to be paid; but in terms of implementation, making $L_{\min}$ as large as reasonably possible is a good idea. One assumes that a "segment" consisting of, for example, three samples is not particularly interesting.)

It is further assumed that the K segments are statistically independent, and hence that the PDF of the data set $\{x\}$ is

$$p(x|\tau, \mu) = \Pi_{i=1}^{K} p(x[t_{i-1}, t_i - 1]|H_{m_i}) \tag{2}$$

where $t_0 \equiv 0$ and $t_K - 1 \equiv N$ by definition, and $x[t_{i-1}, t_i - 1]$ represents the data $\{x_{t_{i-1}}, x_{t_{i-1}+1}, \ldots, x_{t_i-1}\}$. It is helpful to recall that $p(x[t_{i-1}, t_i - 1]|H_{m_i})$ must be estimated from exogenous training data, since normally it is not exactly known. This difficulty often limits the segmentation problem to the single-model case, in which only parameters change as time. For example, the time series could be white and Gaussian (the model), and the variance can differ between segments; or in a more involved situation [2], a series of AR models each with order 16 was chosen for speech segmentation. Now, in the case that the models are fundamentally different from one another *and*

[2] However, the reader may find [7] of interest. That work treats the discovery of DNA segments in which these have only a local statistical description; and indeed the local statistical description is itself to be estimated. The problem here is related, but different.

contain unknown parameters (e.g., with various AR orders but unknown coefficients) the segmentation problem is made even more difficult by the fact that one must compare different model complexities. The class-specific (CS) method [3] allows such comparison, and further it enables one to represent $x[t_{i-1}, t_i-1]$ by a sufficient statistic (such as the estimated AR coefficients). We discuss this shortly in Section II, where each class H_j is fully represented by the CS features and their corresponding PDF.

In this paper, we assume that the following knowledge of the generating models/classes is available. First, each model H_j is sufficiently characterized by a set of features. Next, the PDF of the set of features under class j is known; presumably it has already been estimated based on training data, but that is not the concern of this paper[3]. Third—and this is more of a technical detail—we must assume that there is a "common" hypothesis H_0 in the sense to defined in Section II-A. Finally, for a given time series, our purpose is to divide it into appropriate segments and correctly classify each segment, and we shall examine the performance measures of our proposed scheme accordingly.

D. Plan of the Paper

Our purpose is to find efficient realizations for the joint segmentation and classification problem. We have provided two: the faster one is presented in some detail here, and another based on the iterated conditional mean (similar to Gibbs sampling) is given in detail in [33], but since we compare to it and hybridize it, it appears here in the Appendix. The remainder of the paper is as follows. In Section II we give background on several items vital to the approaches. Specifically, we begin with a short description of Baggenstoss' CS classification ideas in Section II-A, and then we show how the optimal DP approach can be formulated in Section II-B. In Section II-C the appropriate penalty term on the model complexity is given: although one might expect MDL, this turns out not to work particularly well, and we give an alternative approach that relies on CS for penalization of model complexity and an extra MDL-like term for the number of segments.

Section III is devoted to the presentation of the two-stage sequential generalized likelihood ratio (GLR) approach, as generalized from [32]. The first stage here is a simple and speedy GLR segmenter, similar in some respects to a Page detector. The second stage is a more ruminative "refinement" both of the solution from the previous stage and its twin that operates on time-reversed data. If haste is a great concern, one can use the first stage alone.

The proposed scheme is applied to two simulation examples in Section IV; in each case the models are reasonably complex, and PDF's are estimated from the data as they would be in practice. Several alternatives are explored, including the direct and refined "two-stage" approach, and a Gibbs sampling approach that is initialized according to the two-stage solution.

II. Some Useful Results

A. Classification Using Class-Specific (CS) Features

It is well-known [31] that the optimum Bayesian classifier (hypothesis tester) for M classes is

$$\arg\max_j p(H_j|x) = \arg\max_j \{p(x|H_j)p(H_j)\}. \tag{3}$$

Without loss of generality, in this paper we assume that $p(H_j)$ are identical and can be ignored. With an additional "dummy" class, H_0, used in the denominator, the M-ary classifier (3) can be realized by knowing only the likelihood *ratios* $p(x|H_j)/p(x|H_0)$, for $j = 1, \ldots, M$, and thus we have the equivalent form

$$\arg\max_j \left\{ \frac{p(x|H_j)}{p(x|H_0)} p(H_j) \right\}. \tag{4}$$

Then by using a well-known property that any likelihood ratio is invariant when written in terms of a statistic $z(x)$ that is *sufficient* for the test, meaning

$$\frac{p(x|H_j)}{p(x|H_0)} = \frac{p(z|H_j)}{p(z|H_0)}. \tag{5}$$

In [3] Baggenstoss introduced a novel approach to reformulate (3) into an equivalent "class-specific" classifier

$$\arg\max_j \left\{ \frac{p(\mathbf{z}_j|H_j)}{p(\mathbf{z}_j|H_0)} p(H_j) \right\} \tag{6}$$

where H_0 is a common "null" hypothesis. Care must be taken that this null hypothesis is such that each statistic $\{\mathbf{z}_j\}$ is indeed sufficient for testing H_j versus H_0; but in many cases a simple assumption such as that x be independent Gaussian with zero-mean and unit-variance is adequate. Also of interest is the PDF "projected" from the domain of the sufficient statistic to that of the original data [4], defined as

$$p(x|H_j) = p(x|H_0) \frac{p(\mathbf{z}_j|H_j)}{p(\mathbf{z}_j|H_0)}. \tag{7}$$

Note that:

- the $p(\mathbf{z}_j|H_0)$ are assumed known exactly due to the choice of a simple "null" model and analysis of the effect of the sufficient statistic;
- normally $p(\mathbf{z}_j|H_j)$ are not known, and need to be estimated from the training data;
- it is assumed that this training has already been done.

In our method the estimated PDF's $p(x|H_j)$ are obtained by projection [4]. It is also noted via (7) that the CS method allows the likelihood-based "competition" of different models based only on their CS features. Finally, let us note that Baggenstoss [3], [4] tends to assume that "sufficient" features $\{\mathbf{z}_j\}$ are first selected by an expert, and that each has its PDF estimated from its class' training data.

B. Maximum Likelihood Approach Using Dynamic Programming

According to (7) we can rewrite the PDF in (2) as

$$p(x|\tau, \mu) = p(x|H_0)\Pi_{i=1}^{K} \frac{p(\mathbf{z}_{m_i}[t_{i-1}, t_i-1]|H_{m_i})}{p(\mathbf{z}_{m_i}[t_{i-1}, t_i-1]|H_0)} \tag{8}$$

where $\mathbf{z}_{m_i}[t_{i-1}, t_i - 1]$ represents the sufficient statistics $\mathbf{z}_{m_i}$ calculated from the data $\{x_{t_{i-1}}, \ldots, x_{t_i-1}\}$. Let

$$\Lambda(x|\tau, \mu) = \sum_{i=1}^{K} \{\ln(p(\mathbf{z}_{m_i}[t_{i-1}, t_i-1]|H_{m_i})) - \ln(p(\mathbf{z}_{m_i}[t_{i-1}, t_i-1]|H_0))\}. \tag{9}$$

[3] The PDF $p(x[t_{i-1}, t_i-1]|H_j)$ can thus be obtained by projection through using the PDF of the features, as we shall discuss shortly.

A statistically optimal approach is the maximum likelihood segmenter, which chooses τ and μ to maximize (8). It is equivalent to maximize (9), since

$$\begin{aligned}\{\hat{\tau},\hat{\mu}\} &= \arg\max_{\tau,\mu}\{p(x|\tau,\mu)\} = \arg\max_{\tau,\mu}\{\Lambda(x|\tau,\mu)\}\\ &= \arg\max_{\tau}\{\max_{\mu}\{\Lambda(x|\tau,\mu)\}\}.\end{aligned} \tag{10}$$

Since it is assumed each segment is independent of other segments, meaning maximization over parameters for each segment can be performed in parallel, one may apply dynamic programming to avoid an exhaustive search [4] by defining (11) and (12), shown at the bottom of the page. The recursive DP implementation is

$$I_k(L) = \max_{t_{k-1}\in S_k}\{I_{k-1}(t_{k-1}-1) + \Delta_k[t_{k-1},L]\} \tag{13}$$

where S_k indicates the restriction

$$\begin{aligned}\max\{(k-1)L_{\min}+1, L-L_{\max}+1\} &\le t_{k-1}\\ &\le \min\{(k-1)L_{\max}+1, L-L_{\min}+1\}.\end{aligned} \tag{14}$$

It is clear that the computational complexity is reduced substantially as compared to a direct maximization, since only $\Delta_k[t_{k-1},L]$ needs to be computed at each step. The solution to the original problem occurs for $k=K$ and $L=N$.

C. Estimation of Number of Segments via the Minimum Segment Number (MSN)

In practice, information about the number of segments K is unavailable. Naturally the likelihood function from (8) increases monotonically with number of segments due to there being more parameters estimated. This implies that the segmentation procedure should include a criterion for selecting the number of segments (as in order selection applications). Various criteria for selecting the order are described in the literature. There are two such important criteria from the use of information-theoretic arguments: one being the information-theoretic criteria (AIC) pioneered by Akaike [1] and the other the minimum description length (MDL) criterion proposed by Rissanen [28], [29]. The resulting cost functions of AIC and MDL have the following form:

$$C(k) = -\ln p(x|\hat{\tau},\hat{\mu}) + P(k) \tag{15}$$

where the first term is the negative maximum log-likelihood function and $P(k)$ is the penalty function, and $P(k)=n_k$ for AIC and $P(k)=(n_k/2)\ln(N)$, with n_k being the total number of parameters requiring estimation for the k segments, and N being the length of observation. One minimizes $C(k)$ over k and selects the minimizing number of segments and the corresponding parameters as the most parsimonious fit of the data.

From (15), we see that the principal difference between AIC and MDL lies in their structure-dependent penalty terms $P(k)$. It suggests that we should include such a term, but one specifically suited for our problem. We thus propose a similar measure, the *Minimum Segment Number*, defined as

$$MSN(k;x) = \sum_{i=1}^{k}\ln\left(\frac{p(\mathbf{z}_{\hat{m}_i}[\hat{t}_{i-1},\hat{t}_i-1]|H_{\hat{m}_i})}{p(\mathbf{z}_{\hat{m}_i}[\hat{t}_{i-1},\hat{t}_i-1]|H_0)}\right) - P_{MSN}(k) \tag{16}$$

for $k_{\min}\le k\le k_{\max}$, where $\hat{t}_i$'s and $\hat{m}_i$'s are suitable estimates, and $P_{MSN}(.)$ is a problem-dependent penalty term. For example, we have found $P_{MSN}(k)=k\ln(N)$ useful for AR processes. At any rate, the number of segments K is determined as

$$\hat{K} = \arg\max_{k\in[k_{\min},k_{\max}]} MSN(k). \tag{17}$$

Note that the class-specific formalism allows us to separate in (16) the classification penalty (i.e., the denominator in each likelihood ratio) from the penalty on the number of segments. Now, for unknown K, the DP approach is implemented as

$$T_{dp}(k) = I_k(N) - P_{MSN}(k) \tag{18}$$

where $I_k(N)$ is defined as in (13).

III. Approach: Sequential GLR and Refinement

Here we present an alternative segmentation approach. It is based on the procedure of [32], but is modified to allow for more general (multiple!) hypotheses and class-specific decision-making. First, a rough segmentation, calculating GLR sequentially, is implemented. One can stop at this point, since the results are reasonable and the procedure is fast. But if a refined estimate is desired, then this "rough" segmentation should be repeated for the time-reversed time series. Then the results from both are combined, and further processed to obtain more reliable segmentation.

A. Simplified Sequential GLR Method

The signal within each segment is assumed to be homogeneous in the sense that it is characterized only by model m_i and the CS features $\mathbf{z}_{m_i}$. The GLR method detects changes in either models or parameters starting from the location of the previous boundary. Its implementation is described as follows:

1) Initialization: Set $t_0=1$, and let i, the index of the current segment, be 1. Since the length of segment is in the range of $[L_{\min}, L_{\max}]$, we regard $x[1,L_{\min}]$ as the first segment and set $t_i=L_{\min}+1$. The estimated model m_i is

$$m_i = \arg\max_j\left\{\frac{p(\mathbf{z}_j[t_{i-1},t_i-1]|H_j)}{p(\mathbf{z}_j[t_{i-1},t_i-1]|H_0)}\right\} \tag{19}$$

$$\Delta_i[t_{i-1},t_i-1] = \max_{m_i}\{\ln(p(\mathbf{z}_{m_i}[t_{i-1},t_i-1]|H_{m_i})) - \ln(p(\mathbf{z}_{m_i}[t_{i-1},t_i-1]|H_0))\} \tag{11}$$

$$I_k(L) = \max_{\{t_1,t_2,\ldots,t_{k-1}\},t_0=1,t_k=L+1}\sum_{i=1}^{k}\Delta_i[t_{i-1},t_i-1]. \tag{12}$$

in the usual class-specific way.

2) Change Detection: Since there is at most one change during the interval $[t_i, t_i + L_{\min} - 1]$, let y be the sequence of $x[t_{i-1}, t_i + L_{\min} - 1]$. Three situations are possible.

1) The hypothesis H_a is that all the observations in y belong to the same class m_i, characterized by $\mathbf{z}_{m_i}[t_{i-1}, t_i + L_{\min} - 1]$.
2) The first alternative H_{b1} is that there exists an unknown change time $r_0 \in S$ such that, before r_0, the observations in y belong to class m_i, and after the change the observations in y belong to class j, where $j \neq m_i$.
3) The second alternative H_{b2} is that there exists an unknown change time $r_0 \in S$ such that $x[t_{i-1}, r_0]$ belongs to class m_i characterized by $\mathbf{z}_{m_i}[t_{i-1}, r_0]$, and $x[r_0 + 1, t_i + L_{\min} - 1]$ also belonging to class m_i but characterized by $\mathbf{z}_{m_i}[r_0 + 1, t_i + L_{\min} - 1]$, where $\mathbf{z}_{m_i}[r_0 + 1, t_i + L_{\min} - 1]$ is distinguished from $\mathbf{z}_{m_i}[t_{i-1}, r_0]$

In the previous

$$S \equiv [t_i - 1, \min(t_{i-1} + L_{\max} - 1, t_i + L_{\min} - 1)] \quad (20)$$

due to the length constraint (1).

The second alternative refers to the case that segments come from different hypotheses that happen to share a common sufficient statistic – for example, both could refer to an elevation in variance for which the empirical second moment is sufficient, but these levels are different. At any rate, the change detection problem is to test between the hypothesis H_a and the composite hypothesis H_b, where $H_b = H_{b1} \cup H_{b2}$.

Now since the segments are assumed to be independent, it is convenient to introduce the following hypotheses:

$$\begin{aligned}\mathbf{H}_a &: p(y) = p(x[t_{i-1}, t_i + L_{\min} - 1]|H_{m_i})\\ \mathbf{H}_b &: \exists r_0 \in S, \text{such that}\\ &\quad p(y) = p(x[t_{i-1}, r_0]|H_{m_i}) p(x[r_0 + 1, t_i + L_{\min} - 1]|H_j).\end{aligned} \quad (21)$$

According to the GLRT formalism [31], to test, r_0 and j should be replaced by their ML estimates

$$\begin{aligned}(\hat{r}_0, \hat{j}) &= \arg \max_{(r_0 \in S, j \in [1,M])} \ln \left\{ \frac{(p(y|H_b)}{p(y|H_a))} \right\}\\ &= \arg \max_{(r_0 \in S)} \{\max_j \{\ln(p(x[t_{i-1}, r_0]|H_{m_i}))\\ &\quad + \ln(p(x[r_0 + 1, t_i + L_{\min} - 1]|H_j))\}\}\\ &= \arg \max_{(r_0 \in S)} \left\{ \ln \left(\frac{p(\mathbf{z}_{m_i}[t_{i-1}, r_0]|H_{m_i})}{p(\mathbf{z}_{m_i}[t_{i-1}, r_0]|H_0)} \right) \right.\\ &\quad \left. + \max_j \left\{ \ln \left(\frac{p(\mathbf{z}_j[r_0 + 1, t_i + L_{\min} - 1]|H_j)}{p(\mathbf{z}_j[r_0 + 1, t_i + L_{\min} - 1]|H_0)} \right) \right\} \right\}\\ &\approx \arg \max_{(r_0 \in S)} \left\{ \ln \left(\frac{p(\mathbf{z}_{m_i}[t_{i-1}, r_0]|H_{m_i})}{p(\mathbf{z}_{m_i}[t_{i-1}, r_0]|H_0)} \right) \right.\\ &\quad + \left[\frac{t_i + L_{\min} - 1 - r_0}{L_{\min}} \right]\\ &\quad \left. \cdot \max_j \left\{ \ln \left(\frac{p(\mathbf{z}_j[r_0 + 1, r_0 + L_{\min}]|H_j)}{p(\mathbf{z}_j[r_0 + 1, r_0 + L_{\min}]|H_0)} \right) \right\} \right\}. \end{aligned} \quad (22)$$

Since we assume the minimum length of each segment is $L_{\min}$, and since accuracy improves with quantity of data, it is more reliable for us to calculate $\mathbf{z}_j$ based on $x[r_0 + 1, r_0 + L_{\min}]$. In other words, as shown in Fig. 2(a), the window W_3 of with fixed length $L_{\min}$ is used to estimate the *PDF* for the data within the window W_2. Then applying the "projection" formula in (7) (see [4]) and considering the true length, we therefore have approximately the second term of (22).

If $\hat{r}_0$ is obtained now, recalling that the length of each segment is no less than $L_{\min}$, we further note that there are either one or two segments within the observation $\{y_1\} = x[t_{i-1}, \hat{r}_0 + L_{\min}]$. Since different numbers of parameters are involved in the modeling of $\{y_1\}$, we employ the $MSN(l; y_1)$ defined in Section II-C

$$\hat{l} = \arg \max_{l \in [1,2]} MSN(l; y_1) \quad (23)$$

to determine the number of segments l, and the model indices.

The update is carried out as the following.

- *No new segment (H_a):* If both $\hat{l} = 1$, meaning H_a is preferred, then no new change is indicated. The current segment index k is unchanged. We update $t_i = t_i + L_{\min}$ and also update m_i yielding $MSN(1; y_1)$.
- *New segment (H_{b1} or H_{b2}):* Otherwise, if $\hat{l} = 2$, a new segment is indicated. We update $t_i = \hat{r} + 1$, $t_{i+1} = \hat{r}_0 + L_{\min} + 1$, and also update m_i and m_{i+1} correspondingly such that m_i, m_{i+1} and $\hat{r}$, together yields the $MSN(2; y_1)$. We then update $i = i + 1$.

3) Finish: If the end of the time-series has not yet been reached, go back to step 2. Continue this segmentation procedure until all data have been visited. We thus obtain the estimated number of segments $K = i$, and the estimates $\tau = \{t_1, \ldots, t_K\}$ and $\mu = \{m_1, \ldots, m_K\}$.

Note that points of the observation $\{x\}$ are visited sequentially, and hence the estimate of t_i (also m_i) is obtained continually but separately. No joint estimation of τ is needed (as it would be in the DP approach), and therefore this approach is quite fast.

Note also that both $p(\mathbf{z}_j|H_j)$ and $p(\mathbf{z}_j|H_0)$ are required. By choosing an appropriate H_0 (such as iid Gaussian) as our class-specific normalizing PDF, an accurate PDF of $p(\mathbf{z}_j|H_0)$ is obtained even in its far tails via the saddlepoint approximation [19]. The distribution $p(\mathbf{z}_j|H_j)$ is estimated from training data, often via a Gaussian mixture approximation [13].

B. Scheme With Refinement

As indicated earlier, the scheme operates in two stages, as shown in Fig. 2(b).

Stage 1 – Rough Segmentation and Classification: We apply the sequential GLR method from Section III-A to segment the original time-series $\{x_1, x_2, \ldots, x_N\}$, and we record the results as $\{K_g, \tau_g, \mu_g\}$. Then the time-reversed series $\{x_N, x_{N-1}, \ldots, x_1\}$ is also jointly segmented and classified similarly, with the results stored as $\{K_{gr}, \tau_{gr}, \mu_{gr}\}$ – ideally these "reversed" results should coincide with the original ones, but in practice there is always discrepancy.

Stage 2 – Refinement:[4] Both $\{K_g, \tau_g, \mu_g\}$ and $\{K_{gr}, \tau_{gr}, \mu_{gr}\}$ contain information about the correct segmentation/classification on $\{x\}$. These are blended and improved as follows.

[4] To conserve space, many details of this sub-procedure are suppressed; please refer to [32] for full information.

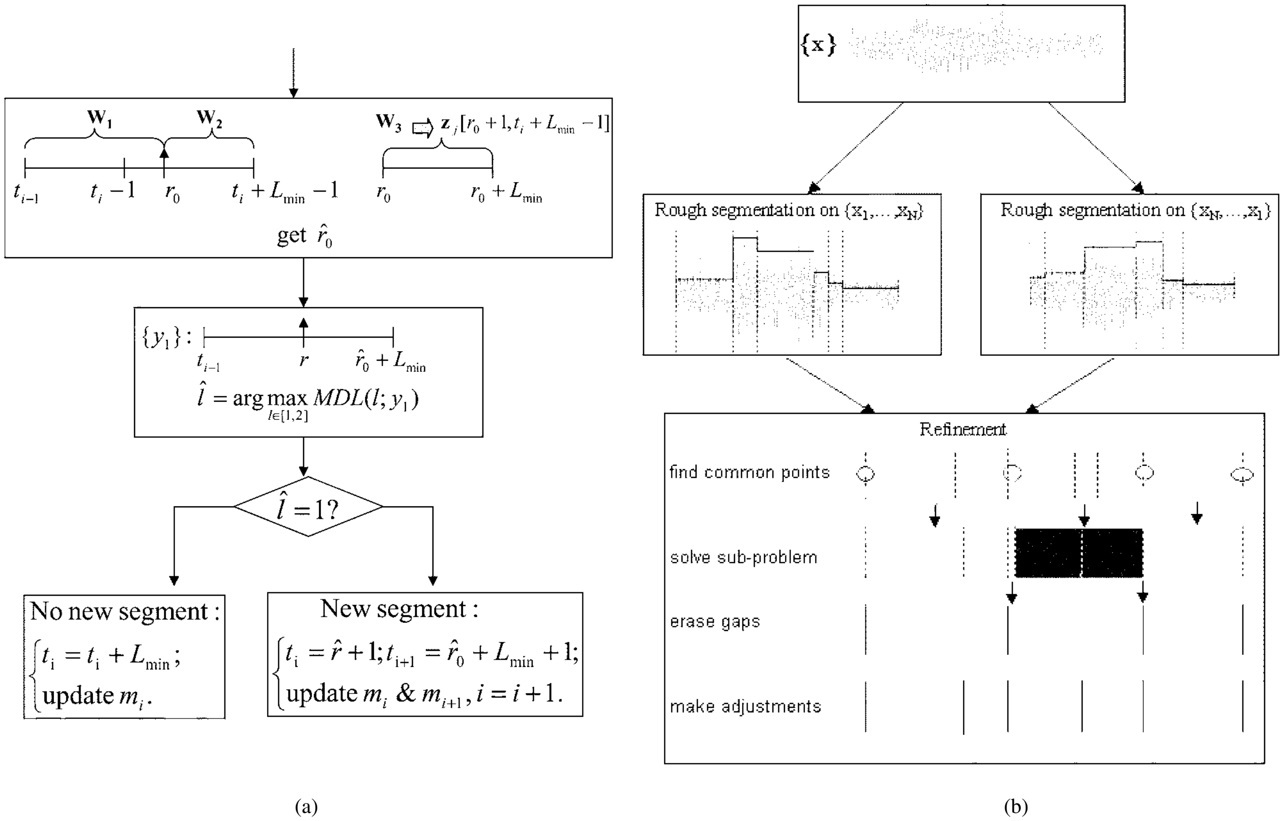
$\mathbf{W}_1$
$\mathbf{W}_2$
$\mathbf{W}_3 \Rightarrow \mathbf{z}_i[r_0+1, t_i+L_{\min}-1]$
t_{i-1}
$t_i - 1$
r_0
$t_i + L_{\min} - 1$
r_0
$r_0 + L_{\min}$
get $\hat{r}_0$
$\{y_1\}:$
t_{i-1}
r
$\hat{r}_0 + L_{\min}$
$\hat{l} = \arg\max_{l \in [1,2]} MDL(l; y_1)$
$\hat{l} = 1?$
No new segment:
$t_i = t_i + L_{\min}$; update m_i.
New segment:
$t_i = \hat{r} + 1; t_{i+1} = \hat{r}_0 + L_{\min} + 1$; update m_i & $m_{i+1}, i = i + 1$.
(a)
{x}
Rough segmentation on {x1,... xN}
Rough segmentation on {xN,...,x1}
Refinement
find common points
solve sub-problem
erase gaps
make adjustments
(b)

Fig. 2. Structure of the sequential GLR scheme. (a) Procedure of change detection in stage 1. (b) The overall procedure.

1) *Find common detected changes in τ_g and τ_{gr}:* A common change is indicated if $\exists i_1$ and i_2, such that $|\tau_g(i_1) - \tau_{gr}(i_2)| \leq L_{\min}/4$, where $L_{\min}/4$ appears to be a suitable threshold). It is reasonable to assume that a change truly occurs in the range between the minimum and maximum of the pair.

2) *Solve sub-problems:* Based on common changes, we divide the time-series x into k_c data records and segment each separately. We save the results for the i^{th} sub-problem as $\{\kappa_i, \tau_i, \mu_i\}$.

3) *Erase gaps:* We assume the existence of a change instant within the data $y = x[\tau_i(\kappa_i - 1), \tau_{i+1}(1) - 1]$ (since a "common" change is observed from τ_g and τ_{gr}). Therefore, $r(i)$ is estimated via its ML estimate, and the model index is also updated correspondingly.

4) *Make adjustments:* Re-iterate the MSN tests now for adjacent segments to adjust their total number. Since our purpose is to reach the maximum of the whole likelihood of $\{x\}$ as nearly as possible, we may also be able to adjust the results to increase the overall likelihood of $\{x\}$ with no change of the number of segments. Repeat this process until there is no further decrease in the number of total segments. We then record the final estimates as τ and μ.

There are several advantages for this scheme: first, the computational burden is much lighter than the optimal ML approach using DP. The computation load grows approximately linearly with the length of the time-series; second, the sequential GLR method in Stage 1 itself could be used as an on-line segmenter; third, there are no hard thresholds or other "tuning" parameters, since only the MSN penalty term is involved; and fourth, although we have presented the scheme from a class-specific segmentation point of view, it could easily be extended to use other criteria.

IV. Simulations

In what follows, we give results in the application of the scheme to two segmentation problems: one problem in which the segments are AR Gaussian time series of different orders, and another in which the model types are mixed (AR versus mean-shifted Gaussian versus variance-shifted Gaussian). In each case the segmentation is according to the respective sufficient statistics: if it were possible to evaluate, say, the statistics of eighth-order AR coefficients when the process was truly fourth-order AR, then the problem would be solved in a manner similar to [32]. Since the models' sufficient statistics are not directly comparable, the class-specific approach of this paper is vital. The PDF of each statistic under its corresponding hypothesis is estimated using a Gaussian mixture; there is

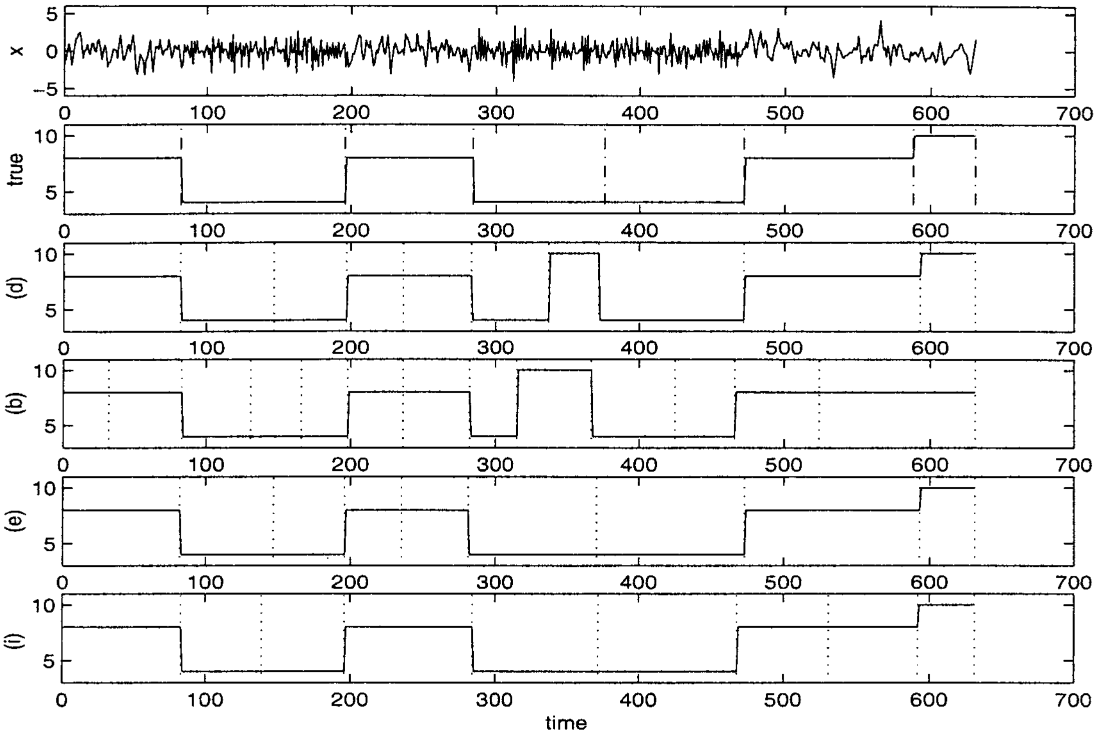

Fig. 3. Example results of our schemes on the segmentation and classification of AR processes. Top: time-series. Next: true change instants and AR model order, out of the menu 4, 8, and 10. (d): Results for the time-series $\{x_1, x_2, \ldots, x_N\}$ via the DP method. (b): Using the simplified sequential method of Section III-A. (e): Using the refined scheme of Section III-B. (i): Using the iterative approach of Appendix .

no attempt made to estimate statistics of noncorresponding models.

We make comparison to a scheme based on iterating the conditional mean—this amounts to Gibbs sampling. The scheme itself is based on the corresponding scheme in [32] for white Gaussian segmentation, and is related to the segmentation approach in [11]. Since there is much overlap, we consign a sketch of the approach in the Appendix, and refer the reader to [33] for more details.

A. Application to AR Processes

We now consider a time-series consisting of AR processes—AR models play an important role in analyzing speech signals and underwater signals. In this example, we are interested in three AR models

$$\begin{aligned} \mathbf{H}_1: &\quad AR(4) \text{ model,} \\ \mathbf{H}_2: &\quad AR(8) \text{ model} \\ \mathbf{H}_3: &\quad AR(10) \text{ model.} \end{aligned} \tag{24}$$

We select the H_0 hypothesis as the iid Gaussian noise with zero-mean and unit-variance. To differentiate an AR(p) model from H_0, a set of autocorrelation function (ACF) samples is approximately sufficient. Thus we can define the CS features $\mathbf{z}$ for AR(p) model as the first $p+1$ ACF lags $\{r_0, r_1, \ldots, r_p\}$. Since $p(\mathbf{z}|H_j)$ will be estimated via the Gaussian mixture approach, it is often desired to work with an alternative feature set by invertible transformation of $\mathbf{z}$. Therefore, finally we choose the CS features $\mathbf{z}_1 = \{\log(r_0), \kappa^4\}$, $\mathbf{z}_2 = \{\log(r_0), \kappa^8\}$ and $\mathbf{z}_3 = \{\log(r_0), \kappa^{10}\}$, where $\kappa^p = \{\kappa_1, \kappa_2, \ldots, \kappa_p\}$ and

$$\kappa_i = \log\left(\frac{(1-K_i)}{(1+K_i)}\right) \tag{25}$$

where $\{K_i\}$ are the reflection coefficients.

We evaluate our schemes in simulation. Data was generated under each class hypothesis using random parameter values. Based on 1000 training series from each class, with lengths distributed uniformly in the interval $[L_{\min}, L_{\max}] = [32, 120]$, the PDF's $p(\mathbf{z}_j|H_j)$ for $j = 1$, 2, and 3 were estimated using Gaussian mixture approximation. A clever means to calculate accurate CS null-hypothesis densities $p(\mathbf{z}_j|H_0)$ via the saddle-point approximation was described as in [19]. In our implementation, for the observation $\{x\}$, we define the penalty term in the MSN test (16) as

$$P_{MSN}(k) = k \times \ln(N) \tag{26}$$

where k is the number of segments within $\{x\}$ and N is the length of $\{x\}$; this coincides with the earlier suggestion.

An example is shown in Fig. 3, where $K = 7$, $L_{\min} = 32$, $L_{\max} = 120$. The observation $\{x\}$ itself, the true change-instants τ and the orders of the AR processes μ are indicated in the top plots. We test the DP approach (termed d method, for DP), the simplified sequential GLR method (termed b, for *b*asic), the final procedure (termed e, for *e*nhanced) that includes refinement, and the iterative conditional-mean approach (termed i). The results are shown in Fig. 3. In this example we find that the e and i methods have gratifyingly well matched the true AR orders, though additional segments (of the same AR order) were decided by either approach. There is a class-mismatch around time sample 350 in the DP (d) method. There are several mismatches in (b): most of these are resolved by the refinement stage (e). The iterative scheme (i) is quite accurate.

A "good" segmenter and classifier should find a number of segments that is close to truth; it should locate the beginnings

TABLE I

MONTE CARLO COMPARISON FOR MMULTIPLE-ORDER AR PROCESSES. HERE THE SUBSCRIPT d REPRESENTS THE DP METHOD; b: THE FORWARD SIMPLIFIED GLR METHOD; e: THE 2-STAGE SEQUENTIAL GLR SCHEME; i: THE ITERATIVE APPROACH BASED ON CONDITIONAL MEANS; ei: THE HYBRID METHOD, e FOLLOWED BY i. $\tilde{K}$ REFERS TO THE AVERAGE, THE AVERAGE ABSOLUTE ERROR, AND THE STANDARD DEVIATION OF THE ESTIMATED NUMBER OF SEGMENTS, *PM* DENOTES THE AVERAGE PERFORMANCE MEASURE FROM (27), AND t SHOWS THE AVERAGE CPU TIME REQUIRED. FOR EACH SIMULATION RUN THE DATA LENGTH IS DRAWN RANDOMLY FROM THE RANGE $[k_{\min}, k_{\max}]$

No. of segments	3	5	7	8	10
$\tilde{K}_d$	3.74(0.82,0.83)	5.83(0.91,0.94)	7.98(1.08, 0.96)	9.16 (1.30,1.25)	11.2(1.42,1.352)
$\tilde{K}_b$	3.93(1.03,0.98)	6.72(1.80,1.46)	9.82(2.84,1.59)	11.26(3.28,1.61)	14.48(4.5,1.95)
$\tilde{K}_e$	3.67(0.81,0.89)	6.01(1.07,1.08)	8.41(1.53,1.22)	9.58(1.68,1.42)	11.98(2.06,1.55)
$\tilde{K}_i$	3.37(0.57,0.89)	5.27(0.63,0.97)	7.35(0.83,1.09)	8.23(0.97,1.24)	10.24(1.02,1.40)
PM_d	0.1620	0.1543	0.1478	0.1576	0.1636
PM_b	0.1955	0.1818	0.1826	0.1834	0.2026
PM_e	0.1487	0.1601	0.1499	0.1633	0.1783
PM_i	0.1354	0.1530	0.1227	0.1538	0.1652
PM_{ei}	0.1419	0.1461	0.1415	0.1597	0.1697
$t_d(min.)$	3.570	14.780	34.504	46.070	70.240
$t_b(min.)$	0.072	0.147	0.214	0.246	0.311
$t_e(min.)$	0.262	0.536	0.799	0.956	1.248
$t_i(min.)$	0.642	1.995	4.126	5.249	8.486
$t_{ei}(min.)$	0.426	0.953	1.502	1.975	2.778

and ends of these segments reasonably accurately; it should classify each segment highly accurately; and it should be expeditious. We intend to report both the first and the last of the above: the number of segments found versus the true number, and the CPU time needed for calculation. In terms of the estimated number of segments $\hat{K}$, we consider the simple average of $\hat{K}$, its average absolute error, and its standard deviation to indicate the difference of schemes. The deviation can help to tell whether the averages are actually significantly different, since the empirical mean could obscure large fluctuation in the estimated K. Evaluating the performance regarding the second objective is not straightforward and there is no universal rule. As such, we propose the criteria concerning the second objective as

$$PM = \frac{1}{N} \sum_{i=1}^{N} I_{c(i)=\hat{c}(i)} \tag{27}$$

where N is the length of the observation $\{x\}$, c and $\hat{c}$ are correspondingly the true and estimated class index vectors, and

$$c(i) = \text{the true class index for point } i \text{ of the series } \{x\} \tag{28}$$

and $\hat{c}(i)$ is similarly defined. Based on this definition, the smaller the *PM* is, the better performance a classifier provides.

A Monte Carlo comparison is shown in Table I, where each datum is based on 100 simulation runs, and the figure $\tilde{K}$ gives the average, the average absolute error, and the standard deviation of the estimated number of segments $\hat{K}$. An interesting picture emerges. First, all methods tend to overestimate the number of segments, and the number recognized by the iterative method using a uniform initialization appears to be more accurate than the DP scheme.

Turning to the computational cost, it is clear that the new sequential GLR approach is remarkably good—the new schemes have a computational load that is approximately linear in the length of the time series. As expected, though DP is intelligent in reducing the computational burden itself, it is still expensive and this expense grows quickly with problem size. The computational loads of new schemes are orders of magnitude less.

Considering the performance measure *PM*, which places greater stress on sample-by-sample model accuracy, it is noted that the overall performances of the proposed methods are startlingly good as compared to the optimal approach using DP (even better *PM* is observed in many cases for method i), and it is clear that there is some room for performance improvement of b and it benefits considerably from the refinement stage in the e method, with the price of around twice more the computation.

Is there some way to be more efficient? The iterative approach i spends much of its time coarsely "hunting;" if this part of its

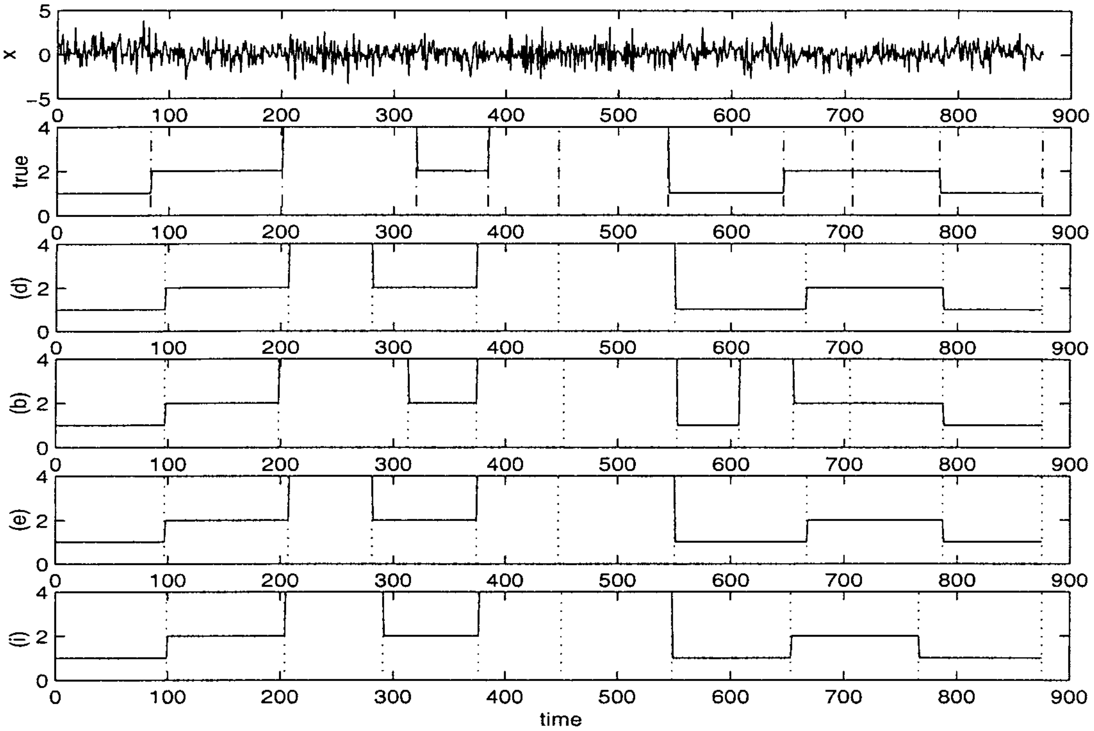

Fig. 4. Results of our scheme on the segmentation and classification of multiple structures, where $K = 10$, $L_{\min} = 32$, $L_{\max} = 120$. Top: time-series. Next: true model, out of the menu 1, 2 and 4 (see (29)). (d): Results for the time-series $\{x_1, x_2, \ldots, x_N\}$ via the DP method. (b): Using the simplified GLR method of Section III-A. (e): Using the refined scheme of Section III-B. (i): Using the iterative approach of Appendix A.

duty is taken over by the quick scheme e, and the i approach reserved for a final super-refinement, then we can. We report on the hybrid scheme ei, in which i method begins respectively from the solutions of the refined sequential GLR methods – the estimated number of segments $\hat{K}$ is clamped to the values passed from e. The performance is promising: *PM* is lower than e method, with the price of almost double the computation.

Concerning K and *PM*, we note that the iterative approach provides the best performance: it yields smaller errors both in estimating the number of segments and in classifying each segment. The sequential GLR scheme yields somewhat worse performance, but it requires the lightest computational load. Therefore, our conclusion is that: the sequential GLR scheme is the best choice when CPU time is the strictest limitation; the basic iterative approach i is best if we want better accuracy; and we choose the hybrid ei approach when jointly considering the performance and computational load. The DP approach should be out of picture in this application due to its prohibitive computation.

B. Application to Multiple Structures

To explore our scheme to applications having competing models with different structures, we consider

$$\begin{aligned} \mathbf{H}_1: &\quad y_i \sim N(\alpha, 1), \text{ for } i \in [1, n] \\ \mathbf{H}_2: &\quad y_i \sim N(0, \beta^2), \text{ for } i \in [1, n] \\ \mathbf{H}_4: &\quad y \text{ follows } AR(4) \text{ model} \end{aligned} \tag{29}$$

where $N(\mu, \sigma^2)$ represents the Normal distribution with μ-mean and σ^2-variance, and α and β^2 are random variables whose distributions are not known. In other words, the observations in the H_1 model are iid Gaussian with unit-variance but unknown (different) means. Again, we select the H_0 hypothesis as the iid Gaussian noise with zero-mean and unit-variance.

The sufficient statistics $\mathbf{z}_j$, $j = 1, 2$, for testing H_j against H_0 are easy; and $\mathbf{z}_4$ is as in the previous section. Therefore, we have the following CS features:

$$\begin{aligned} \mathbf{z}_1 &= \frac{1}{n} \sum_{i=1}^{n} y_i \\ \mathbf{z}_2 &= \frac{1}{n} \sum_{i=1}^{n} y_i^2 \\ \mathbf{z}_4 &= \{\log(r_0), \kappa^4\} \end{aligned} \tag{30}$$

where $\kappa_i = \log((1-K_i)/(1+K_i))$ and $\{K_i\}$ are the reflection coefficients. Clearly, $p(\mathbf{z}_1|H_0)$ and $p(\mathbf{z}_2|H_0)$ can be easily and exactly obtained, and as in Section IV-A an accurate $p(\mathbf{z}_4|H_0)$ was obtained via the saddlepoint approximation.

Data was generated under each class hypothesis using random parameter values. Based on 1000 training samples from each class, with length distributed uniformly in the interval $[L_{\min}, L_{\max}] = [32, 120]$, the PDF's $p(\mathbf{z}_j|H_j)$ were estimated using a Gaussian mixture approximation. As in Section IV-A, the penalty term is chosen as

$$P_{MSN}(k) = k \times \ln(N). \tag{31}$$

An example is shown in Fig. 4, and in this case it appears that all schemes except b match well.

Monte Carlo comparisons based on 100 simulation runs are given in Table II. All methods except b tend to under-segment, but the e scheme on average gives the best estimation of K. It is also noted that despite requiring much lighter computational load, both the e and the hybrid ei methods provide better *PM*

TABLE II
MONTE CARLO COMPARISON FOR MULTIPLE STRUCTURE MODELS. HERE THE SUBSCRIPT d REPRESENTS THE DP METHOD; b: THE FORWARD SIMPLIFIED GLR METHOD; e: THE FINAL (REFINED) SEQUENTIAL GLR SCHEME; i: THE ITERATIVE APPROACH BASED ON CONDITIONAL MEANS; ei: THE HYBRID METHOD. $\tilde{K}$ REFERS TO THE AVERAGE, THE AVERAGE ABSOLUTE ERROR, AND THE STANDARD DEVIATION OF THE ESTIMATED NUMBER OF SEGMENTS, *PM* DENOTES THE AVERAGE PERFORMANCE MEASURE FROM (27), AND t SHOWS THE AVERAGE CPU TIME REQUIRED. FOR EACH SIMULATION RUN THE DATA LENGTH IS DRAWN RANDOMLY FROM THE RANGE $[k_{\min}, k_{\max}]$

No. of segments	3	5	7	8	10
$\tilde{K}_d$	2.77(0.31,0.51)	4.30(0.74,0.67)	6.28(0.74,0.68)	6.81(1.23,0.85)	8.40(1.62,1.02)
$\tilde{K}_b$	3.25(0.55,0.74)	5.28(0.84,1.12)	7.70(0.96,1.07)	8.53(0.99,1.23)	10.7(1.24,1.47)
$\tilde{K}_e$	2.90(0.28,0.52)	4.76(0.62,0.81)	6.82(0.58,0.85)	7.43(0.81,0.92)	9.40(0.9,1.06)
$\tilde{K}_i$	2.66(0.40,0.54)	4.08(0.98,0.76)	6.07(0.97,0.81)	6.64(1.38,0.88)	8.20(1.82,1.07)
PM_d	0.1354	0.1597	0.1267	0.1703	0.1826
PM_b	0.1795	0.1992	0.1460	0.1833	0.1793
PM_e	0.1348	0.1622	0.1270	0.1601	0.1722
PM_i	0.1489	0.1838	0.1355	0.1745	0.1854
PM_{ei}	0.1254	0.1577	0.1158	0.1571	0.1546
$t_d(min.)$	1.670	6.040	14.184	17.946	28.6502
$t_b(min.)$	0.054	0.104	0.160	0.186	0.232
$t_e(min.)$	0.150	0.304	0.479	0.538	0.704
$t_i(min.)$	0.277	0.740	1.405	1.652	2.399
$t_{ei}(min.)$	0.2133	0.501	0.817	0.907	1.342

than even the d method[5]. Our conclusion for this application comes as: overall the sequential GLR (e) scheme is the best choice; and, with the price of 70% more computation, the ei approach should be chosen for improved accuracy. In the previous example the i scheme was appealing in that it was the most frugal with sections: in this example its parsimony remains, but it appears misplaced.

V. CONCLUSION

The need to segment a time series into intervals that are locally statistically homogeneous arises in a number of applications, the two most accessible being in speech processing (the sections are phonemes) and transient detection (the sections are signals whose presence may be of interest). Previous approaches to segmentation have tended to be either special-purpose, in need of the number of segments, or slow. However, two techniques with none of these weaknesses were developed in a previous paper for the *special case* that segmentation was on the basis of a scale change in white data [32]. These two techniques are both fast and accurate. However, they were tailored to the specific application given.

In this paper the faster of the two is extended to far more general case that: there is a list of possible statistical models; each of these is describable in terms of a sufficient statistic (but these need not be the same); and there is in hand an estimate for the PDF of each sufficient statistic under its own assumed model. There is no need for a joint probabilistic model (whose availability would be unrealistic) of all sufficient statistics, this via the new class-specific multihypothesis testing breakthrough. And there is no need to know the number of sections, since an appropriate penalty term, similar to MDL, is given.

The approach has two stages. In stage one, rough segmentations are implemented sequentially using a piecewise GLR; in the second stage, the results from the first stage (both forward and backward) are refined. The computational burden is remarkably small, approximately linear with the length of the time series. The method is quite satisfactory in terms both of segmentation complexity and accuracy. The approach can be run either as one stage or as two: in the former case it is remarkably fast, and while it is reasonably accurate, it can be improved. The full (two-stage) implementation is somewhat more intensive, but is

[5] While this may seem a cause for concern, recall that DP maximizes likelihood, which is not directly measured by *PM*; and that DP's likelihood-maximizing properties are adulterated by the penalty term.

very accurate. If both accuracy and speed are concerns, a hybrid between the approach presented here that initializes a refinement using Gibbs sampling is preferable.

Appendix
Markov Chain Monte Carlo Conditional Mean Approach

The key to model (9) is to estimate τ, the segment transition times. Here we present an iterative method to estimate τ by revisiting the data for each transition time t_i. We treat τ as composed of random variables, and assume a noninformative (uniform) prior. Now, for an arbitrary transition time t_i, treat other transition times as known and obtain the posterior conditional distribution $p(t_i|x, t_1, \ldots, t_{i-1}, t_{i+1}, .., t_K)$ from that; then this PDF serves as updated information to estimate t_i.

Suppose K is known. The basic form of the proposed iterative approach is as follows:

1) Choose a noninformative prior for τ, to have a minimal impact on the posterior distribution **[15]**, according to

$$p(\tau) = \frac{1}{M} I_{\{(1) \text{ is true}\}} \qquad (32)$$

where $I_{\{.\}}$ is 1 if the length constraints **(1)** are satisfied by τ and 0 otherwise. The total number of possible choice of τ is M, a parameter whose actual value needs not be computed since it is the same for all estimates.

2) Initialize the iteration counter $n = 1$, and choose the initial estimated values τ to divide the series $\{x\}$ into K similar-sized segments.

3) Classify each segment. That is, using the current segment boundaries $\tau = \{t_0, t_1, \ldots, t_K\}$ and the class-specific formalism in **(6)**

$$\hat{m}_{i+1} = \arg\max_j \left\{ \frac{p(\mathbf{z}_j[t_i, t_{i+1}-1]|H_j)}{p(\mathbf{z}_j[t_i, t_{i+1}-1]|H_0)} p(H_j) \right\} \qquad (33)$$

in which $\mathbf{z}_j[t_i, t_{i+1} - 1]$ is the statistic sufficient for discrimination of hypothesis H_j from H_0, and in which H_0 is the common "null" hypothesis (such as white unit-normality) that has been chosen.

4) Update the estimate of t_i, for $i = 1, \ldots, K - 1$. The following conditional posterior distributions $p(t_i|t_0, \ldots, t_{i-1}, t_{i+1}, \ldots, t_K, x)$ ($\equiv p(t_i|\bar{t}_i, x)$) for $i = 1, \ldots, K-1$ are required for updating the estimate of t_i, where

$$p(t_i|\bar{t}_i, x) \propto \frac{p(\mathbf{z}_{\hat{m}_i}[t_{i-1}, t_i - 1]|H_{\hat{m}_i})}{p(\mathbf{z}_{\hat{m}_i}[t_{i-1}, t_i - 1]|H_0)} \frac{p(\mathbf{z}_{\hat{m}_{i+1}}[t_i, t_{i+1} - 1]|H_{\hat{m}_{i+1}})}{p(\mathbf{z}_{\hat{m}_{i+1}}[t_i, t_{i+1} - 1]|H_0)} I_{\{(1) \text{ is true}\}} \qquad (34)$$

for $i = 1, \ldots, K - 1$, where $\bar{t}_i = \{t_0, \ldots, t_{i-1}, t_{i+1}, \ldots, t_K\}$, and $\hat{m}_i$ is the ML estimate corresponding to $[t_{i-1}, t_i]$, as derived in **[32]**, and more specifically for this application in **[33]**. Note that $t_0 = 1$ and $t_K - 1 = N$ by definition. Now, for $i = 1, \ldots, K - 1$, update t_i by its conditional mean

$$t_i = \hat{t}_i = E(t_i|\bar{t}_i, x) = \sum_{t_i \in S} p(t_i|\bar{t}_i, x) t_i \qquad (35)$$

where S indicates the set satisfying the constraints **(1)** and in which $p(t_i|\bar{t}_i, x)$ is from **(34)**. Since t_i is a transition time, it should be integral, and thus rounding is appropriate.

5) If for ∀ i, t_i has converged, then stop. Otherwise, let $n = n + 1$ and return to step 3. The convergence criterion is that the maximum mismatch of τ between the n^{th} iteration and $(n-1)^{st}$ be no greater than 1.

6) Record the converged τ as the final estimates $\hat{\tau}$ and record the corresponding $\hat{\mu}$. In the above, a joint estimate of τ is avoided via the repeated update of individual t_i's, and therefore a huge computational savings is achieved. Usually, very few iterations are needed.

For the case that the number of segments K is unknown, this methods relies on the MSN in the same way as does the paper's main approach: the same procedure is executed for $K = 1, K = 2$, etc., until (16) begins to drop—the maximum value is chosen. It appears that using conditional means as estimates helps to avoid local minima. In this regard, and in fact as justification for its convergence, note that the approach shares the features of Gibbs Sampling, an implementation of Markov Chain Monte Carlo (MCMC) [16], [30]. Under Gibbs one would generate a random variable from $p(t_i|x, t_1, \ldots, t_{i-1}, t_{i+1}, .., t_K)$; here, for speed, we use that density's mean.

References

[1] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automat. Contr.*, vol. AC-19, pp. 716–723, Dec. 1974.

[2] R. Andre-Obrecht, "A new statistical approach for the automatic segmentation of continuous speech signals," *IEEE Trans. Acoustic, Speech Signal Process.*, vol. 36, pp. 29–40, Jan. 1988.

[3] P. Baggenstoss, "Class-specific feature sets in classification," *IEEE Trans. Signal Processing*, vol. 47, pp. 3428–3433, Dec. 1999.

[4] ——, "A theoretically optimal probabilistic classifier using class-specific features," in *Proc. ICPR'00 Conf.*, 2000.

[5] M. Basseville and I. Nikiforov, *Detection of Abrupt Changes: Theory and Application*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[6] M. Basseville and A. Benveniste, "Sequential detection of abrupt changes in spectral characteristics of digital signals," *IEEE Trans. Inform. Theory*, vol. 29, pp. 709–724, Sept. 1983.

[7] T. Bailey and C. Elkan, "Unsupervised learning of multiple motifs in biopolymers using expectation maximization," *Mach. Learning*, vol. 21, pp. 51–83, 1995.

[8] J. Besag, "On the statistical analysis of dirty pictures," *J. Royal Stat. Soc. B*, vol. 48, pp. 259–302, 1986.

[9] A. Brandt, "Detecting and estimating parameter jumps using ladder algorithms and likelihood ratio test," in *Proc. ICASSP'83 Conf.*, Boston, MA, 1983, pp. 1017–1020.

[10] N. Burobin, V. Mottl, and I. Muchnik, "An algorithm for detection of multiple change of properties of random process based on the dynamic programming method," in *Detection of Change in Random Processes*. New York: Optimization Software, Inc., 1986.

[11] O. Cappe, "A Bayesian approach for simultaneous segmentation and classification of count data," *IEEE Trans. Signal Processing*, vol. 50, pp. 400–410, Feb. 2002.

[12] V. Digalakis, M. Ostendorf, and J. Rohlicek, "Fast algorithms for phone classification and recognition using segment-based models," *IEEE Trans. Signal Processing*, vol. 40, pp. 2885–2896, Dec. 1992.

[13] B. Everitt and D. Hand, *Finite Mixture Distributions – (Monographs on Applied Probability and Statistics)*. London, U.K.: Chapman & Hall, 1981.

[14] B. Friedlander and B. Porat, "Performance analysis of transient detectors based on a class of linear data transforms," *IEEE Trans. Inform. Theory*, vol. 38, pp. 665–673, Mar. 1992.

[15] A. Gelman, J. Garlin, H. Stern, and D. Rubin, *Bayesian Data Analysis*. London, U.K.: Chapman & Hall, 1995.

[16] W. Gilks, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*. London, U.K.: Chapman & Hall, 1996.

[17] T. Hemminger and Y.-H. Pao, "Detection and classification of underwater acoustic transients using neural networks," *IEEE Trans. Neural Networks*, vol. 5, pp. 712–718, Sept. 1994.

[18] S. Kay, "Optimal segmentation of time series based on dynamic programming," *Private Communication*.

[19] S. Kay, A. Nuttall, and P. Baggenstoss, "Multidimensional probability density function approximations for detection, classification, and model order selection," *IEEE Trans. Signal Processing*, vol. 49, pp. 2240–52, Oct. 2001.

[20] R. Kenefic, "An algorithm to partition DFT data into sections of constant variance," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, pp. 789–95, Mar. 1998.

[21] M. Lavielle, "Optimal segmentation of random processes," *IEEE Trans. Signal Processing*, vol. 46, pp. 1365–73, May 1988.

[22] M. Lavielle and E. Moulines, "Least-squares estimation of an unknown number of shifts in a time series," *J. Time Series Anal.*, vol. 21, no. 1, pp. 33–59, 2000.

[23] G. Lu and T. Hankinson, "An investigation of automatic audio classification and segmentation," in *Proc. ICSP'00 Conf.*, 2000, pp. 776–81.

[24] O. Maeran, V. Piuri, and G. Gajani, "Speech recognition through phoneme segmentation and neural classification," in *Proc. IEEE Instrum. Meas. Technol. Conf.*, Ottawa, ON, Canada, May 19–21, 1997, pp. 1215–20.

[25] S. Marco and J. Weiss, "Improved transient signal detection using a wavepacket-based detector with an extended translation-invariant wavelet transform," *IEEE Trans. Signal Processing*, vol. 45, pp. 841–850, Apr. 1997.

[26] A. Moghaddamjoo, "Automatic segmentation and classification of ionic-channel signals," *IEEE Trans. Biomed. Eng.*, vol. 38, pp. 149–155, Feb. 1991.

[27] M. Ostendorf and S. Roucos, "A stochastic segment model for phoneme-based continuous speech recognition," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 37, pp. 1857–69, Dec. 1989.

[28] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.

[29] ——, "Stochastic complexity in statistical inquiry," in *World Scientific Series in Computer Science*. New York: World Scientific, 1989, vol. 15.

[30] M. Tanner, *Tools for Statistics Inference*. New York: Springer-Verlag, 1991.

[31] H. Van Trees, *Detection, Estimation and Modulation Theory: Part 1*. New York: Wiley, 1968.

[32] Z. Wang and P. Willett, "Two algorithms to segment white Gaussian data with piecewise constant variances," IEEE Trans. Signal Processing, vol. 51, pp. 373–385, Feb. 2002, to be published.

[33] Z. Wang, P. Willett, and P. Baggenstoss, "Class-specific segmentation of time series," *Proc. IEEE Aerosp. Conf.*, Mar. 2003.

[34] T. Zhang and C.-C. Kuo, "Audio content analysis for online audiovisual data segmentation and classification," *IEEE Trans. Speech Audio Processing*, vol. 9, pp. 441–457, May 2001.

Zhen Jane Wang received the B.Sc. degree (with highest honors) from Tsinghua University, China, in 1996, and the M.Sc. and Ph.D. degrees from the University of Connecticut, Storrs, in 2000 and 2002, respectively, all in electrical engineering.

She is currently a Research Associate with the Electrical and Computer Engineering Department and Institute for Systems Research, University of Maryland, College Park. Her research interests are in the broad areas of statistical signal processing, information security, and wireless communications.

Dr. Wang received the Outstanding Engineering Doctoral Student Award from the University of Connecticut.

Peter Willett (S'83–M'86–SM'97–F'03) received the B.Sc. degree from the University of Toronto, Toronto, ON, Canada, in 1982 and the Ph.D. degree from Princeton University, Princeton, NJ, in 1986.

He is a Professor of electrical and computer engineering at the University of Connecticut, Storrs. He has written, among other topics, about the processing of signals from volumetric arrays, decentralized detection, information theory, CDMA, learning from data, target tracking, and transient detection.

Dr. Willett is a member of the IEEE Signal Processing Society's Sensor Array and Multichannel Technical Committee. He is an Associate Editor for the IEEE Transactions on Aerospace and Electronic Systems and the IEEE Transactions on Systems, Man, and Cybernetics. He is a Track Organizer for Remote Sensing at the IEEE Aerospace Conference (2001-2003), and is co-chair of the Diagnostics, Prognosis, and System Health Management SPIE Conference in Orlando. He was Technical Program Co-Chair for the 1999 ISIF Fusion conference in Sunnyvale CA, and is presently Technical Program Co-Chair for the 2003 IEEE Systems, Man & Cybernetics Conference, Washington DC.