R/D Optimal Linear Prediction

Paolo Prandoni, Martin Vetterli

Abstract

A common technique to extend linear prediction to nonstationary signals is time segmentation: the signal is split into small portions and the modelization is carried out locally. The accuracy of the analysis is however dependent on the window size and on the signal characteristics, so that the problem of finding a good segmentation is crucial to the entire modeling scheme. In this paper we will present an algorithm which determines the optimal segmentation with respect to a cost function relating prediction error to modeling cost. The proposed approach casts the problem in a rate/distortion framework, whereby the segmentation is implicitly computed while minimizing the modelization distortion for a given modelization cost. The algorithm is implemented by means of dynamic programming and takes the form of a trellis-based Lagrangian minimization. The optimal linear predictor, when applied to speech coding, dramatically reduces the amount of bitrate devoted to the modeling parameters in comparison to fixed-window schemes.

EDICS: 1-CODI

P. Prandoni is with the Laboratoire de Communication Audio Visuelle, École Polytechnique Fédérale de Lausanne, CH - 1015 Lausanne. E-mail: prandoni@de.epfl.ch, tel: +41 21 693 5629, fax: +41 21 693 4312.

M. Vetterli is with the Electrical Engineering and Computer Science Department, University of California at Berkeley, Berkeley CA 94720 and with the École Polytechnique Fédérale de Lausanne, CH - 1015 Lausanne. E-mail: vetterli@de.epfl.ch, tel: +41 21 693 5698, fax: +41 21 693 4312

I. Introduction

Arguably, linear prediction (LP) is amongst the most ubiquitous and successful signal processing tools, with applications ranging from channel equalization to data compression, and with a wealth of efficient and very robust algorithmic implementations. As a particular case of the more general least squares system identification problem, linear prediction relies on the special assumption that the unknown system is purely autoregressive; in signal processing parlance, this amounts to postulating that the experimental data is the output of an all-pole IIR linear filter $H(z) = 1/A(z)$. One can show that, under ideal conditions, linear prediction computes the exact inverse filter $A(z)$ together with the correct model order; inverse filtering the data by $A(z)$ recovers the original filter input.

Perhaps the greatest success of linear prediction techniques is to be found in speech coding: a whole class of coders, generally grouped under the label of *hybrid time-domain coders*, possesses a LP block at its core. This stems from the intuitively sound and physically relevant modelization of the human speech apparatus as a excitator-resonator pair: the oscillation of the vocal cords or a turbulent air flow are the input to the vocal tract in producing voiced and unvoiced sounds. Although the vocal tract is not exactly an all-pole IIR (there are zeros related to the nasal cavities), the accuracy of the achievable modelization is striking. After inverse filtering, the excitation (or *residual*) is recovered and different compression schemes are defined by the way the residual and the LP coefficients are encoded. At one extreme of simplicity, systems such as LPC-10 [1] and RELP [2] simply quantize the LP coefficients and a parametrized version of the residual in open-loop fashion. At the other extreme, codebook-based encoders such as CELP [3] utilize the LP direct filter to produce a speech waveform from a synthetic residual; the residual is constructed from codebook entries as to minimize (in some perceptual "norm") the error between original and resynthesized speech, with the encoder operating in closed-loop.

It is important to note at this point that although the closed-loop encoding process for a speech coder is usually optimized with respect to ad-hoc perceptual criteria, the inverse filter computed by the LP block is always obtained via a least squares minimization, where the goodness-of-fit is measured by the mean squared error (MSE). In this paper we are concerned precisely with the global optimization of the composite linear predictor

for an arbitrary signal with respect to its overall squared error and its composite order. Indeed, the fundamental difficulty with linear prediction of real signals lies with their inherent nonstationarity; this is usually addressed by means of fixed-length windowing in the expectation that, over small time intervals, the signal varies little. While this is true in general terms, it does still happen that abrupt signal transitions fall within a windowed segment, and the probability of such events increases with the window size. On the other hand, the window size cannot be reduced arbitrarily since, for locally stationary segments, the accuracy of the estimation increases with it. Clearly, one would like to obtain a flexible segmentation in which the boundaries coincide with the transitions in the signal and the length of the segments minimize the local squared error. Such a flexible segmentation implies asynchronous coding and buffering; while this is bad news for low-delay applications, it opens the way to unequal resource allocation for different portions of the signal. It is a well known fact, for instance, that unvoiced speech can be coded with a coarser LP description than voiced speech, since it has a flatter spectral structure. For a given bit budget, LP parameters should receive more bits where they are most needed.

In the following we will present an algorithm based on dynamic programming and resource allocation techniques which, for a given signal, jointly determines the optimal segmentation and the optimal sequence of predictors for the segments with respect to the global LP error; we will show that the problem can be effectively cast and solved in a generalized rate/distortion framework. Our approach differs from previous results attempting a local optimization of the LP encoding process [4], [5] in that we completely do away with frame-based encoding; the segmentation is completely flexible and the final cost (in bits) of the LP parameters is minimized from within the LP encoding process.

The outline of the paper is the following: in section 2 we will review the basic linear prediction problem and show some of the tradeoffs connected to local windowing. Section 3 will introduce the R/D framework and show how an efficient solution to the optimal LP problem can be obtained using standard optimization techniques. Finally, in section 4, we will develop a trellis-based algorithm for the efficient implementation of the optimization process and present some experimental results comparing standard LP to the optimal LP in the context of speech coding.

II. Linear Prediction

In the classic stochastic formulation, given a discrete time stationary signal $x(n)$, an order-p linear predictor computes p coefficients $a_1, \ldots, a_p$ so that the linear combination

$$\hat{x}(n) = a_1 x(n-1) + a_2 x(n-2) + \ldots + a_p x(n-p) \tag{1}$$

minimizes the expected squared error:

$$d^2 = \mathrm{E}\{|x(n) - \hat{x}(n)|^2\}. \tag{2}$$

Minimization of (2) with respect to the a_i's yields a set of *normal equations* which are of the form

$$\sum_{i=1}^{p} a_i \rho(i,j) = \rho(0,j), \text{ for } j = 1, \ldots, p \tag{3}$$

where $\rho(i,j) = \mathrm{E}_n\{x(n-i)x(n-j)\}$, the signal's autocorrelation at lag $|i-j|$.

For a finite data set, the problem loses its probabilistic nature and becomes a particular case of least squares approximation. In fact, for a truly autoregressive signal with no added noise, p samples suffice to uniquely determine the order-p generating model using Padè's method [6]; these hypotheses are of course never met in practice so that many more data points must be taken into account to minimize the effects of noise and model mismatch. For a data set $\{x(m), x(m+1), \ldots, x(m+M-1)\}$, $M \gg p$, the goal of linear prediction becomes the minimization of the squared error

$$d^2 = \sum_{n=A}^{B} (x(n) - \hat{x}(n))^2; \tag{4}$$

the summation limits, A and B, reflect the implicit assumptions on the signal values outside of the known support and determine the influence of "border effects" on the LP computation. For $A = m + p$ and $B = m + M - 1$, the error is evaluated only within the interval, and no further assumptions are made; this LP strategy is called the *covariance method*. For $A = m$ and $B = m + M - 1 + p$, together with the assumption that the signal is identically zero outside of $[m, m + M - 1]$, we have the so-called *autocorrelation method*; this method often requires a tapering window since the discontinuity between the signal and the zero extension might have adverse effects on the parameter estimation, but

it is the most advantageous computationally. In both cases, the deterministic LP problem produces a set of normal equations formally identical to (3) in which

$$\rho(i,j) = \sum_{n=A}^{B} x(n-i)x(n-j) \tag{5}$$

The formulation in (5) is very close to the empiric autocorrelation formula; for an ergodic signal this would converge to the true autocorrelation as the sample size grows to infinity, and this formal analogy explains why it is customary to apply random processes terminology (such as "stationary") even to finite data sets.

The goodness-of-fit indicator for the deterministic linear predictor is the error in (4), usually normalized by M (the MSE); this measure is however blind to the specific causes of misadjustment and, for a fixed set of samples, it reacts similarly to phenomena such as additive noise, undermodeling, and signal transitions. However, if the data set is a windowed portion of a longer signal, the behavior of the MSE as a function of window length and window placement offers more insight. Figure 1 shows a portion of a typical speech signal; the three lower panels show the MSE as a function of the window length for the three points in the signal marked A, B, and C. The first panel is relative to the onset of a voiced sound, and the MSE decreases non monotonically until the drift in the periodic waveform make it increase again; the non monotonic behavior in the the curve is due to the data window straddling, at times, a non integer number of periods. The second panel shows a monotonically decreasing MSE corresponding to a slightly noisy, silent gap between utterances. The third panel shows the abrupt transition in the MSE occurring at sharp signal onsets such as at point C. These simple examples suffice to demonstrate the potential benefits of a dynamic window size.

As for the order of the linear predictor, it is easy to show that the MSE is a nondecreasing function of p [7]; under ideal conditions, for an autoregressive signal of order q, the curve obtained by plotting the MSE as a function of p remains constant for $p \geq q$. In general, for any dataset, the curve flattens out as p grows and a flatness test has indeed been suggested in the past as the method to determine the correct order for the predictor [8]. It is important at this point to introduce the notion of *cost* of the LP representation; we can measure this cost by the number of bits needed to encode (and transmit) the predictor's

parameters and, independently of the overall encoding strategy, this number is clearly a non decreasing function of the LP order. In modeling and compression application, the diminishing returns in terms of MSE with growing model order must be weighted against the corresponding increase in bit rate. An optimal tradeoff would deploy different orders to different portions of the signal so that each segment is coded with the same benefit.

If we now consider the composite LP analysis of a signal with respect to a given segmentation, the measure of accuracy of the modelization is given by the cumulative squared errors of the segmental predictors while the cost (in terms of encoded LP parameters) is an nondecreasing function of the sum of model orders and of the total number of segments. If we identify the squared error as the model's distortion and its descriptive cost as its rate, the optimal segmentation and the optimal order allocation can be determined jointly as the solution to a R/D minimization over the space of all possible segmentations and all possible sequences of orders. Please note that true optimality can not be based on a local "stationarity" test nor on a local flatness test but only on a global minimization where the segmentation and the orders play the role of free variables.

III. Rate/Distortion Optimality

A. Problem statement

For a dataset $\boldsymbol{x} = \{x(0), \ldots, x(N-1)\}$ define a *segmentation* $\boldsymbol{t}$ as a collection of $k+1$ time indices: $\boldsymbol{t} = \{t_0 = 0 < t_1 < t_2 < \ldots < t_{k-1} < t_k = N\}$. The number of segments defined by $\boldsymbol{t}$ is $\sigma(\boldsymbol{t}) = k$, $1 \leq k \leq N$, with the i-th segment being $\{x(t_i), \ldots, x(t_{i+1}-1)\}$; segments are strictly disjoint. Let $T_{[0,N]}$ be the set of all possible segmentations for $\boldsymbol{x}$, which we will simply write as T when the signal range is self-evident; it is clearly $|T_{[0,N]}| = 2^{N-1}$.

Next, assume there are Q different LP models which could be applied to a segment; these are essentially predictors of different orders, but could include predictors whose parameters are then quantized and coded in different ways. Given a segmentation $\boldsymbol{t}$, define an *allocation* $\boldsymbol{p}(\boldsymbol{t})$ as a collection of $\sigma(\boldsymbol{t})$ model indices p_i, $1 \leq p_i \leq Q$; let $P(\boldsymbol{t})$ be the set of all possible allocations for $\boldsymbol{t}$, with $|P(\boldsymbol{t})| = Q^{\sigma(\boldsymbol{t})}$. Again, when the dependence on the underlying segmentation is clear, we will simply write $\boldsymbol{p}$ instead of $\boldsymbol{p}(\boldsymbol{t})$.

For a given segmentation $\boldsymbol{t}$ and a related allocation $\boldsymbol{p}$, define $R(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{p})$ as the cost,

in bits, associated to the sequence of LP models and define $D(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{p})$ as the cumulative squared error of the segmental LP computation. Since there is no overlap between segments and the LP models are applied independently, we can write

$$D(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{p}) = \sum_{k=1}^{\sigma(\boldsymbol{t})} d_x^2(t_k, t_{k+1}; p_k) \tag{6}$$

where $d_x^2(t_k, t_{k+1}; p_k)$ is the squared error as in (4) for a linear predictor of order p_k applied to $\{x(t_i), \ldots, x(t_{i+1} - 1)\}$. Similarly, we will assume that the LP coefficients are coded independently and that their cost in bits is a function of the predictor's order $b(p)$. An important remark at this point is that, by allowing for a data-dependent segmentation, *information about the segmentation and the allocation themselves must be provided along with the LP coefficients.* This takes the form of side information, which uses up part of the global bit budget of the R/D optimization and must therefore be included in the expression for the overall rate. We will then write

$$R(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{p}) = \sum_{k=1}^{\sigma(\boldsymbol{t})} (c + b(p_k)) = \sum_{k=1}^{\sigma(\boldsymbol{t})} r(p_k) \tag{7}$$

where c is the side information associated to a new segment specifying its length and the relative LP order[1].

Our goal is to arrive at a minimization of the global squared error with respect to the local LP orders and to the data segmentation using the global rate as a parameter controlling the number of segments and the distribution of LP resources amongst the segments. Formally, this amounts to solving the following constrained problem:

$$\begin{cases} \min\limits_{\boldsymbol{t} \in T} \min\limits_{\boldsymbol{p} \in P(\boldsymbol{t})} \{D(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{p})\} \\ \\ R(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{p}) \leq R_C \end{cases} \tag{8}$$

While at first the task of minimizing (8) seems daunting, requiring $O(Q^N)$ explicit comparisons, we will show how it can be solved in polynomial time for almost all rates using standard optimization techniques.

[1]Here, the cost of side information is assumed constant for simplicity. However, no major changes in the subsequent derivation are needed if this cost depends on the segment's parameters.

B. Efficient Solution

The problem of optimal resource allocation has been studied extensively in the context of quantization and coding for discrete datasets [9], [10] and has been successfully applied to the context of signal compression and analysis [11], [12], [13]. In the following we will rely extensively on the results in [14], to which the reader is referred for details and proofs.

For the time being assume that a segmentation $\boldsymbol{t}_0$ is given (a fixed-window segmentation, for instance) and that the only problem is to find the optimal allocation of LP orders; each allocation defines an operational point in the R/D plane as in Figure 2-(a) and the inner minimization in (8) requires us to find the allocation yielding the minimum distortion amongst all the allocations with the same given rate. However, if we restrict our search to the convex hull of the entire set of R/D points, the minimization can be reformulated using Lagrange multipliers: define a functional $J(\lambda) = D(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{p}) + \lambda R(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{p})$; if, for a given λ,

$$\boldsymbol{p}^* = \arg \min_{\boldsymbol{p} \in P(\boldsymbol{t}_0)} \{J(\lambda)\} \tag{9}$$

then $\boldsymbol{p}^*$ (star superscripts denote optimality) defines a point on the convex hull which solves the problem:

$$\begin{cases} \min\limits_{\boldsymbol{p} \in P(\boldsymbol{t}_0)} \{D(\boldsymbol{x}, \boldsymbol{t}_0, \boldsymbol{p})\} \\ \\ R(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{p}) \leq R(\boldsymbol{x}, \boldsymbol{t}_0, \boldsymbol{p}^*) \end{cases} \tag{10}$$

It may help the intuition to show why the set U of (R, D) pairs solutions to (9) indeed defines a convex line on the R/D plane. Given any two solutions (R_i, D_i) and (R_j, D_j), $R_i > R_j$, the line in the R/D plane connecting them has an (absolute) slope $\gamma = (D_j - D_i)/(R_i - R_j)$. Convexity requires that all solutions (R, D) such that $R_i \leq R \leq R_j$ lie below this line. Suppose this was not the case for a solution $(R, D) \in U$ in terms of slopes connecting (R, D) to (R_i, D_i) and to (R_j, D_j) this implies

$$\frac{D_j - D}{R - R_j} < \gamma < \frac{D - D_i}{R_i - R}. \tag{11}$$

The (R, D) pair is by hypothesis a solution to (9) for a given λ; however, if $\lambda < \gamma$, (11) implies $D_i + \lambda R_i < D + \lambda R$ and we have a contradiction; otherwise, if $\lambda \geq \gamma$, we have

again $D_j + \lambda R_j \leq D + \lambda R$; therefore $(R, D) \notin U$. This holds for all elements of U and, for a dense solution set, it provides an intuitive meaning to the value of λ in (9) as the derivative of the convex hull at the relative solution point.

If we now let the segmentation vary, we simply obtain a larger population of operational R/D points which are indexed by segmentation-allocation pairs as in Figure 2-(b). Again, if we choose to restrict the minimization to the convex hull of the composite set of points, we can solve the associated Lagrangian problem as a double minimization:

$$J^*(\lambda) = \min_{\boldsymbol{t} \in T} \min_{\boldsymbol{p} \in P(\boldsymbol{t})} \{J(\lambda)\}; \tag{12}$$

It should be noted that the restriction to the convex hull is of little practical limitation when the set of R/D points sufficiently dense; this is indeed the case given the cardinalities of T and P.

Even in the form of (12) the double minimization would still require an exhaustive search over all R/D points, in addition to a search for the optimal λ. By taking the structure of rate and error into account, we can however rewrite (12) as:

$$J^*(\lambda) = \min_{\boldsymbol{t} \in T} \min_{\boldsymbol{p} \in P(\boldsymbol{t})} \{\sum_{k=1}^{\sigma(\boldsymbol{t})} (d_x^2(t_k, t_{k+1}; p_k) + \lambda r(p_k))\} \tag{13}$$

Since all quantities are nonnegative and the segments are non overlapping, the inner minimization over $P(\boldsymbol{t})$ can be carried out independently term-by-term, reducing the number of comparisons to $Q\sigma(\boldsymbol{t})$ per segmentation. Now the key observation is that, whatever the segmentation, all segments are coded with the same rate/distortion tradeoff as determined by λ; therefore, for a given λ, we can determine the optimal $\boldsymbol{t}$ (in the sense of (8)) using dynamic programming [15]. Indeed, suppose a breakpoint t belongs to $\boldsymbol{t}^*$, the optimal segmentation; then it is easy to see that

$$J^*_{[0,N]}(\lambda) = J^*_{[0,t]}(\lambda) + \min_{\boldsymbol{t} \in T_{[t,N]}} \min_{\boldsymbol{p} \in P(\boldsymbol{t})} \{J(\lambda)\} \tag{14}$$

(where subscripts indicate the signal range for the minimization). In other words, if t is an optimal breakpoint, the optimal cost functional for $\{x(0), \ldots, x(t-1)\}$ is independent of subsequent data. This defines an incremental way to jointly determine the optimal segmentation and allocation as a recursive optimality hypothesis for all data points: for

$1 \le t \le N$,

$$J^*_{[0,t]}(\lambda) = \min_{0 \le \tau \le t-1} \{ J^*_{[0,\tau]}(\lambda) + \min_{1 \le p \le Q} \{ d_x^2(\tau, t; p) + \lambda r(p) \} \} \tag{15}$$

(where $J^*_{[0,0]}(\lambda) = 0$). At each step t, only the new $J^*_{[0,t]}(\lambda)$ and the minimizing τ need be stored. The total number of comparisons for the double minimization is therefore $O(N^2)$. The latter must be iterated over λ until the rate constraint in (8) is met; luckily, the overall rate is a monotonically nonincreasing function of λ (for a proof, see again [14]) so that the optimal value can be found with a fast bisection search [11].

IV. Implementation

In the following we will present an algorithmic implementation of the R/D optimal linear prediction based on a trellis search. While direct implementation of (15) requires only linear storage, the method is not efficient when iterating over λ since all intermediate quantities would have to be recomputed. The trellis, on the other hand, requires quadratic storage but dramatically improves the speed of the search for the correct λ.

The LP "engine" will be Durbin's algorithm [16], an autocorrelation method. Although no tapering data window will be employed on the segments, the adaptive segmentation is sufficiently flexible to minimize boundary effects and the optimal solutions does not significantly differ from what would be obtained using the covariance method; the autocorrelation method simply puts a slightly higher bias on longer segment lengths. The following derivation is primarily illustrative of the key algorithmic issues, while still computationally efficient. More elaborate implementations can be obtained tailoring the algorithm to more sophisticated LP schemes such as [17].

A. Preliminaries

While the segmentation algorithm allows for an extremely fine resolution, with minimum segment size of one data point, in most application it is more appropriate to allow for a coarser granularity, with segment sizes multiple of a C-point interval. Call this minimal span a *cell* and define $\boldsymbol{y} = \{y(0), \ldots, y(L-1)\}$ the collection of cells corresponding to $\boldsymbol{x}$ (assume $N = CL$, possibly by zero-padding). The optimal segmentation/allocation

algorithm can therefore be applied to $\boldsymbol{y}$ with the substitution

$$d^2_{\boldsymbol{y}}(t_k, t_{k+1}; p_k) \leftarrow d^2_{\boldsymbol{x}}(Ct_k, Ct_{k+1}; p_k) \tag{16}$$

Assume we allow for Q possible linear predictor orders for the segments: $p_1 < p_2 < \ldots < p_Q$. Since the Durbin's recursion computes the prediction error for all lower order problems as well, we will solve an order-p_Q LP problem for all segments. This involves the computation of the sum-products (5) which, for the autocorrelation method, depend only on the lag $l = |i - j|$. With respect to $\boldsymbol{y}$, for cell k we have

$$\rho_k(l) = \sum_{n=Ck}^{C(k+1)-1-l} x(n)x(x+l) \tag{17}$$

and, for an interval $\{y(k), \ldots, y(h)\}$, $\rho_{[k,h]}(l)$ satisfies the following chain relation:

$$\rho_{[k,h]}(l) = \rho_{[k,h-1]}(l) + \rho_h(l) + \delta_h(l) \tag{18}$$

with $= \delta_h(l) = \sum_{n=Ch-l}^{Ch-1} x(n)x(n+l)$.

B. Trellis Algorithm

Step 1: Initialization

Compute for all cells $y(t), 0 \leq t \leq L-1$ both $\rho_t(l)$ and $\delta_t(l)$ for $0 \leq l \leq p_Q + 1$ ($\delta_0(l) = 0$).

Organize the data on a trellis as follows (see Figure 3-(a)): at each step t, $0 \leq t \leq L-1$, create a set S_t containing $t+1$ new states $s_{t,v}$, $0 \leq v \leq t$. Then, for each $s_{t,v} \in S_t$:
A) associate to the state a set of autocorrelation estimates $\varphi_{t,v}(l) = \varphi_{t-1,v-1}(l) + \rho_t(l) + \delta_t(l)$ (with $\varphi_{\cdot,-1}(\cdot) = 0$)
B) solve Durbin's recursion from order p_1 to p_Q using $\varphi_{t,v}$ and associate to $s_{t,v}$ the relative squared errors $d^2_y(t-v, t+1; p_i)$, $1 \leq i \leq Q$.
Finally, create an extra state set $S_L = \{s_{L,0}\}$.

Step 2: Trellis path population

For a given value for λ, let $j^*_0 = 0$ and, for $0 \leq t \leq L-1$, determine the minimum cumulative Lagrangian cost $j^*_{t+1} = \min_v \min_i \{j^*_{t-v} + d^2_y(t-v, t+1; p_i) + \lambda r(p_i)\}$, $0 \leq v \leq t$,

$1 \leq i \leq Q$, using the values stored in the trellis. Assume the minimum is for v^* and $p_i = p^*$: connect s_{t,v^*} to $s_{t+1,0}$ and associate (j^*_{t+1}, v^*, p^*) to the path. Also, connect $s_{t,v}$ to $s_{t-1,v-1}$ for $v > 0$ (see Figure 3-(b)).

Step 3: Backtracking

From $j^*_L = d^*(\lambda) + \lambda r^*(\lambda)$ obtain the optimal rate and distortion for the given λ. Then follow the path backward in the trellis from $s_{L,0}$ to $s_{0,0}$ collecting, where applicable, the values for v^* and p^*. Assume k such values are collected, numbered from k to 1 as we proceed backward: the optimal segmentation is then $\boldsymbol{t}^* = \{t_0, \ldots, t_k, t_{k+1} = L\}$ with $t_i = t_{i+1} - v^*_i$, and the optimal allocation is $\boldsymbol{p}^* = \{p^*_1, \ldots, p^*_k\}$ (Figure 3-(b)).

Step 4: Iteration over λ

In some cases, especially when the rate constraint is specified to within a tolerance interval, an educated guess for λ can avoid the need for an explicit search. In most cases, however, the value for λ must be determined iteratively until the rate constraint R_C is met as closely as possible. The search algorithm exploits the convexity of the solution set and proceeds as follows [11]: first determine $\lambda_{\min}$ and $\lambda_{\max}$ so that $r^*(\lambda_{\max}) \leq R_C \leq r^*(\lambda_{\min})$ (see below); choose a starting value for λ between $\lambda_{\min}$ and $\lambda_{\max}$.

Then, run the path population and backtracking sections (Steps 2 and 3 above); if $R_C < r^*(\lambda)$ then replace $\lambda_{\max}$ by λ, else replace $\lambda_{\min}$ by λ; determine the new value as $\lambda = (d^*(\lambda_{\min}) - d^*(\lambda_{\max}))/(r^*(\lambda_{\max}) - r^*(\lambda_{\min}))$ and repeat until the rate constraint is met.

Initial values

Given the monotonic relationship between λ and $r^*(\lambda)$, an obvious choice for the initial minimum value is $\lambda_{\min} = 0$, for which the minimum allowable distortion is achieved. A good starting value for $\lambda_{\max}$ can be inferred from the following argument. For a rate of zero bits, since no prediction is made, the resulting distortion is equal to the energy of the entire signal, $D_0 = \sum_n x^2(n)$. As we now sweep λ from $+\infty$ to 0, consider the first value λ_1 for which $r^*(\lambda_1) \geq 1$ bit. Because of (12), for all (R, D) pairs it is $D + \lambda_1 R \geq d^*(\lambda_1) + \lambda_1 r^*(\lambda_1)$

and, in particular, for $(0, D_0)$ we can write

$$\lambda_1 \leq \frac{D_0 - d^*(\lambda_1)}{r^*(\lambda_1)} \leq D_0 \tag{19}$$

We can therefore set $\lambda_{\max} = \sum_n x^2(n)$.

C. Commentary

The computational requirements for the algorithm can be broken up as follows. The initialization step requires $O(N)$ adds and multiplies for the computation of the autocorrelation estimates. Then, for each state in the trellis, we need to sum up the cumulative estimates ($O(p_Q)$ adds and multiplies) and solve a LP problem with Durbin's algorithm ($O(p_Q^2)$ adds and multiplies plus $O(p_Q)$ divisions). The number of states is $L(L+1)/2$, $L = N/C$, so in total the initialization step requires $O((N/C)^2)$ operations. Each iteration over λ requires a maximum of p_Q comparisons (plus two adds and one multiply) per state, with again a total of $O((N/C)^2)$ operations. Storage is proportional to the number of states.

For large datasets, as in the case of speech, the algorithm can easily be applied to data segments separated by clearly identifiable features such as silence or gaps, for which an a-priori allocation decision can be made with little risk of suboptimality. Alternatively, since backtracking from time index $t_0 < L$ yields the optimal segmentation for $[0, t_0]$, a splitting point can also be determined heuristically by executing the backtracking step for all t's: once a point t_0 of stable path convergence is found, in the sense that at least a minimum number of successive backtracked paths converge in t_0, the trellis can be restarted at that point and the allocation for $t < t_0$ determined independently of successive data. Finally, the algorithm can alternatively be run over a sliding window of sufficient length $W \gg C$ [18].

D. Results

As an example, we present here a comparison between a fixed-window coding strategy such as that employed in the LPC-10 speech coder [1] and the R/D optimal LP coding. The LP block in the LPC-10 coder splits the data into fixed 22.5 ms frames (180 data points for 8 KHz sampled speech) and computes an order-10 predictor for each frame. In

our coder, we set $Q = 6$, allowing the order to span the range from $p_1 = 5$ to $p_6 = 10$, and we set the cell size to $C = 60$ samples. Quantization follows the LPC-10 specifications, with 5 bits for the first four coefficients, 4 bits for the next four, 3 and 2 bits for the last two; the LP parameters are in the form of reflection coefficients and therefore the effects of quantization can be easily integrated in Durbin's recursion. The cost of side information, which specifies the length of the current segment, is set to 10 bits per segment, with three bits for the predictor's order and seven bits for the segment's length. The algorithms are applied to a 2.2 seconds speech signal from a standard test corpus with DC bias removed and normalized to unit amplitude.

Figure 4 shows the set of solutions obtained by iterating the algorithm between $\lambda_{\min}$ and $\lambda_{\max}$; the distortion is the cumulative squared error normalized by the signal's energy, while the rate is normalized by the length of the speech sample. The circle shows the operating (R, D) point for the fixed-window algorithm at its constant bitrate of 1822 bit/sec; the normalized error power is -9.02 dB. At an equivalent rate, the R/D optimal algorithm gains 0.74 dB; more significantly, an equivalent error power of -9.4 dB is achieved at a bitrate of 277 bit/sec, which is approximately six times lower (point C in Figure 4). Figure 5 show the segmentations and allocations relative to points A, B and C in the R/D curve alongside with the speech signal. The width of a block represent a segment's length, and its height the corresponding LP order. At first it might seem surprising that a coarse segmentation such as C has the same MSE characteristics as the fine windowing of the LPC-10 scheme; yet, while in both cases most of the signal's energy is still in the residual, its time distribution is very different. The LPC-10 22.5 ms interval is appropriate as a baseline grid to resolve fast transients but offers no gain with respect to longer range prediction; under the same distortion constraint, the dynamic segmentation algorithm still resolves the main transients, but avoids isolating small speech portions, which cannot be modeled by an order-10 predictor anyway, in favor of a more accurate estimation of the quasi-stationary parts.

V. Conclusions

The problem of determining a good segmentation for the analysis of nonstationary signals is necessarily related to a cost function capturing the performance of the modeling

tool. In the case of linear prediction, the goal is to minimize the global squared error while keeping the description cost for the composite LP model under a certain threshold (parsimonious modeling). In this context, the optimal segmentation can automatically be determined together with the best sequence of local predictors using a trellis-based Lagrange minimization algorithm. This is a totally flexible rate/distortion approach in which model parameters, cost and accuracy requirements, and granularity of the analysis can be varied at will. While not directly amenable to real-time implementations (which is the inescapable price of global optimality), practical applications this LP scheme include speech analysis and coding for storage purposes in which, as shown, dramatic bandwidth savings can easily be achieved.

References

[1] T. E. Tremain, "The government standard linear predictive coding algorithm: Lpc-10," *Speech Technology*, pp. 40–49, April 1982.

[2] B. Fette, "High quality secure voice communication," *Speech Technology*, pp. 40–48, October 1989.

[3] M. Schroeder and B. Atal, "Code excited linear prediction: High quality speech at low bit rates," in *Proc. ICASSP*, 1985, pp. 937–940.

[4] B. Fette and C. Jaskie, "A 600 bps LPC voice coder," in *Proc. MILCOM '91*, 1991, pp. 1215–1219.

[5] E. Bryan George, A. V. McCree, and V. R. Viswanathan, "Variable frame rate parameter encoding via adaptive frame selection using dynamic programming," in *Proc. ICASSP*. IEEE, 1996, vol. 1, pp. 271–274.

[6] J. S. Lim and A.V. Oppenheim, Eds., *Advanced Topics in Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1988.

[7] A. M. Kondoz, *Digital Speech*, Wiley, Chichester, England, 1994.

[8] J. Makhoul, "Linear prediction: A tutorial review," *Proc. IEEE*, vol. 63, pp. 561–580, April 1975.

[9] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*, Kluver, 1992.

[10] Eve Riskin, "Optimal bit allocation via the G-BFOS algorithm," *IEEE Trans. IT*, vol. 37, no. 2, pp. 400–402, March 1991.

[11] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Tran. on IP*, vol. 2, no. 2, pp. 160–175, April 1993.

[12] Z. Xiong, K. Ramchandran, C. Herley, and M. T. Orchard, "Flexible time segmentations for time-varying wavelet packets," *IEEE Proc. Intl. Symp. on Time–Frequency and Time–Scale Analysis*, pp. 9–12, Oct. 1994.

[13] P. Prandoni, M. Goodwin, and M. Vetterli, "Optimal time segmentation for signal modeling and compression," in *Proc. ICASSP*, Munich, April 1997, vol. 3, pp. 2029–2032.

[14] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, and Signal Proc.*, vol. 36, no. 9, pp. 1445–1453, September 1988.

[15] R. Bellman, *Dynamic Programming*, Princeton University Press, 1957.

[16] P. Clarkson, *Optimal and Adaptive Signal Processing*, CRC Press, 1993.

[17] M. Morf, B. Dickinson, T. Kailath, and A. Vieira, "Efficient solution of covariance equations for linear prediction," *IEEE Trans. ASSP*, vol. ASSP-25, no. 5, pp. 429–433, October 1977.

[18] Prandoni P. and Vetterli M., "Optimal bit allocation with side information," in *Proc. ICASSP*, Phoenix, USA, March 1999.

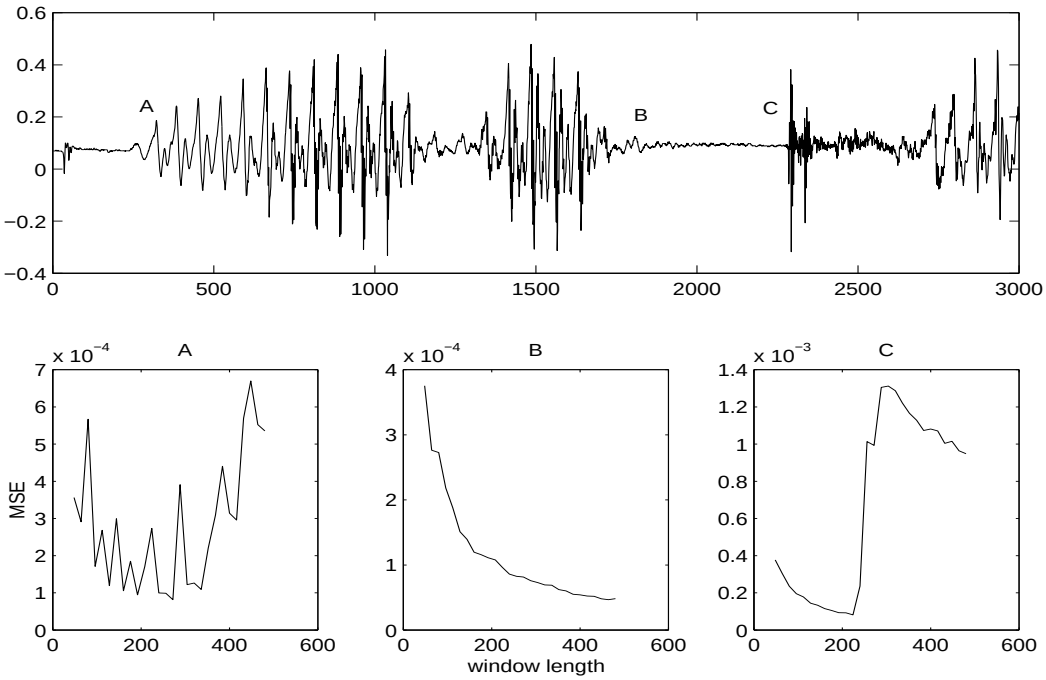

Fig. 1. Speech segment (upper panel) and linear prediction MSE as a function of window length for data starting at points A, B, and C (three lower panels respectively).

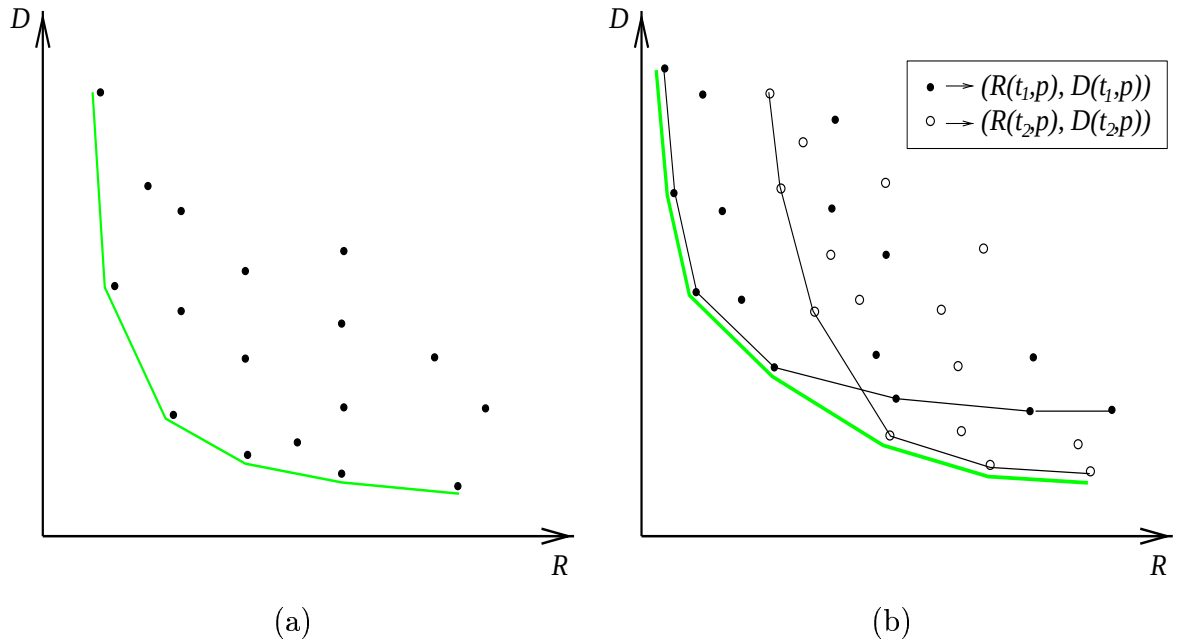

Fig. 2. R/D convex hulls: (a) convex hull for a single segmentation; (b) composite convex hull for two segmentations: black and white dots are (R, D) pairs relative to segmentations $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$; thin lines show the distinct hulls for each segmentation.

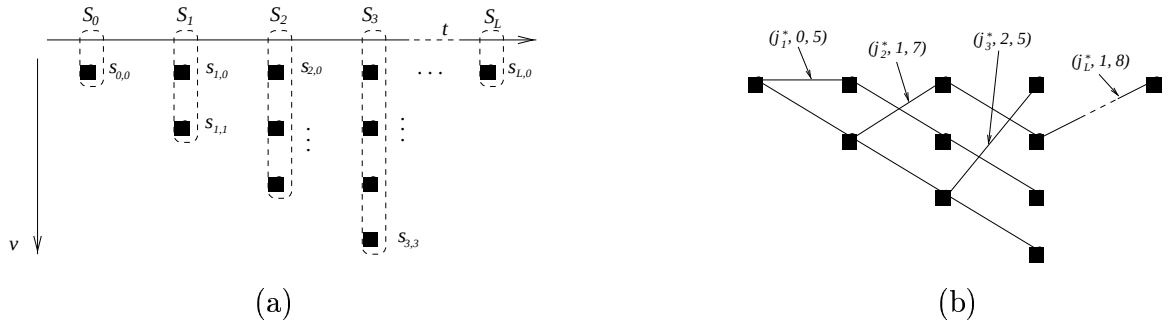

Fig. 3. (a) Initialization step for the trellis; (b) path population. Backtracking from $s_{L,0}$ would yield $\boldsymbol{t} = \{0, 2, L\}$, $\boldsymbol{p} = \{7, 8\}$.

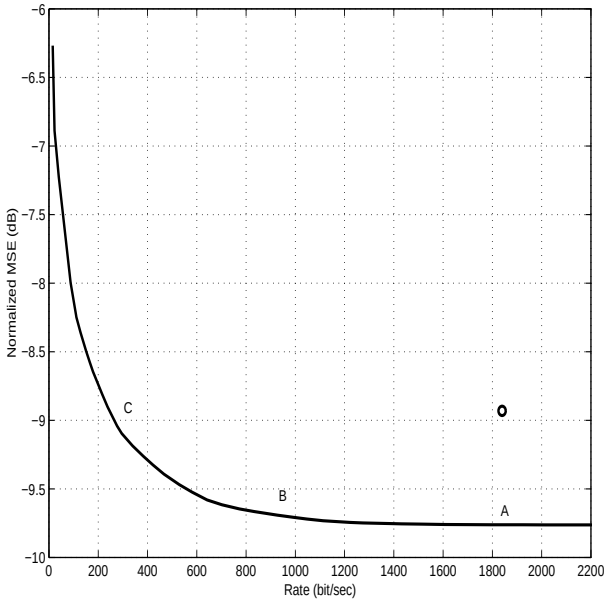

Fig. 4. Operational Rate/Distortion curve for a speech segment; the dot indicates the operational R/D point of the fixed-window, LPC-10 predictor.

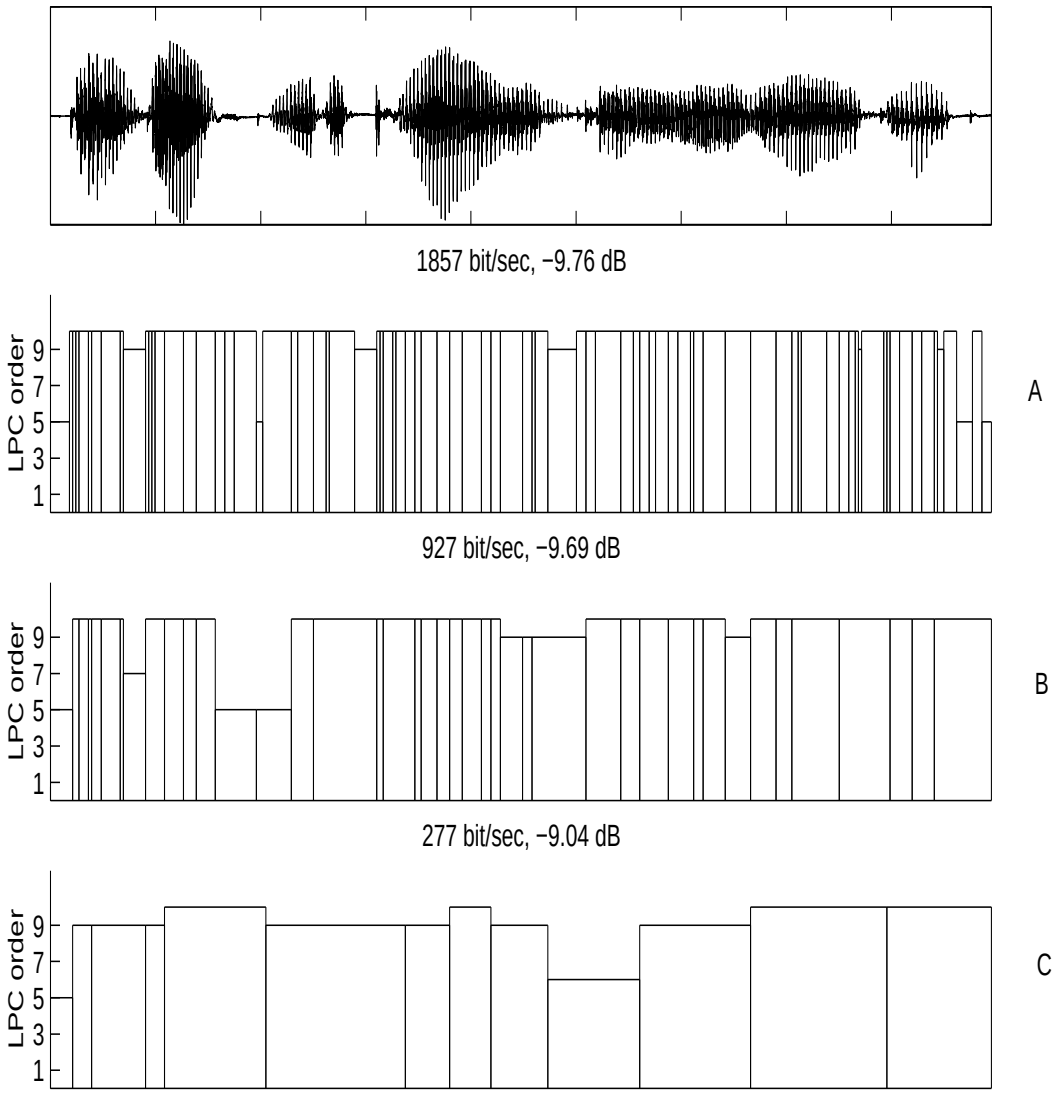

Fig. 5. Segmentations and allocations for the speech signal in the upper panel at the (R, D) operational points marked A, B, and C in Figure 4.