Deformable Markov Model Templates for Time-Series Pattern Matching

Xianping Ge
Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425
xge@ics.uci.edu

Padhraic Smyth
Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425
smyth@ics.uci.edu

ABSTRACT

This paper addresses the problem of automatically detecting specific patterns or shapes in time-series data. A novel and flexible approach is proposed based on segmental semi-Markov models. Unlike dynamic time-warping or template-matching, the proposed framework provides a systematic and coherent framework for leveraging both prior knowledge and training data. The pattern of interest is modeled as a K-state segmental hidden Markov model where each state is responsible for the generation of a component of the overall shape using a state-based regression function. The distance (in time) between segments is modeled as a semi-Markov process, allowing flexible deformation of time. The model can be constructed from a single training example. Recognition of a pattern in a new time series is achieved by a recursive Viterbi-like algorithm which scales linearly in the length of the sequence. The method is successfully demonstrated on real data sets, including an application to end-point detection in semiconductor manufacturing.

Categories and Subject Descriptors

I.5.4 [**Pattern recognition**]: Applications—*Waveform analysis*; G.3 [**Mathematics of computing**]: Probability and statistics—*Time series analysis*

General Terms

Time series, pattern matching, deformable templates, segmental Markov models, hidden Markov models

1. INTRODUCTION

A fundamental problem in pattern recognition and data mining is the problem of automatically recognizing specific waveforms in time-series based on their *shapes*. Applications in the context of time-series data mining include exploratory data analysis of time-series, monitoring and diagnosis of critical systems, classification of time-series, and unsupervised discovery of recurrent patterns. We propose a novel Markov-based representation for waveform shapes and couple this to an efficient and optimal Viterbi-like algorithm for online waveform detection. The detection algorithm is optimal in the maximum likelihood sense of detecting the time-series segment which is most likely to have been generated by the waveform model.

Much work on this problem in the data mining literature has emphasized the issue of scalability in this context [1, 7, 2, 5], i.e., being able to scale one's representation method and matching algorithm to massive time-series archives. In this paper we explicitly focus on fundamental *signal representation* and *matching* aspects of the problem, rather than scalability. We believe that the representation and matching problems are still not adequately solved (we explain why in Section 3) and, thus, our philosophy is that these issues need to be addressed first before scalability is considered. Having said all of this, we will discuss in section 5 how our approach can be scaled up in an efficient fashion. However, the main focus is on representation and matching.

In the following sections of the paper, we begin by providing a general definition of the problem (Section 2). In Section 3 we then discuss related prior work on this problem. In Section 4 we propose our new segmental semi-Markov model framework for pattern representation. A specific pattern-detection algorithm is described in Section 5, followed by results and evaluation in Section 6.

2. STATEMENT OF PROBLEM

Consider that we have a waveform pattern Q of interest and we wish to detect any occurrences of this pattern Q in a (potentially much longer) time-series R. We will assume that Q (for "query" pattern) is a univariate waveform, uniformly sampled in time, with a distinct "shape." Generalizations to multivariate and non-uniformly-sampled waveforms are relatively straightforward and not discussed in this paper due to space limitations.

Figure 1(a) provides a simple example of such a pattern Q and its shape. This particular pattern signals the end of what is known as the *plasma etch process* in semiconductor manufacturing. The end of the process is referred to as the *end-point*. A skilled semiconductor engineer can manually interpret and detect this pattern *offline* after the entire process is complete. However, in a real operational

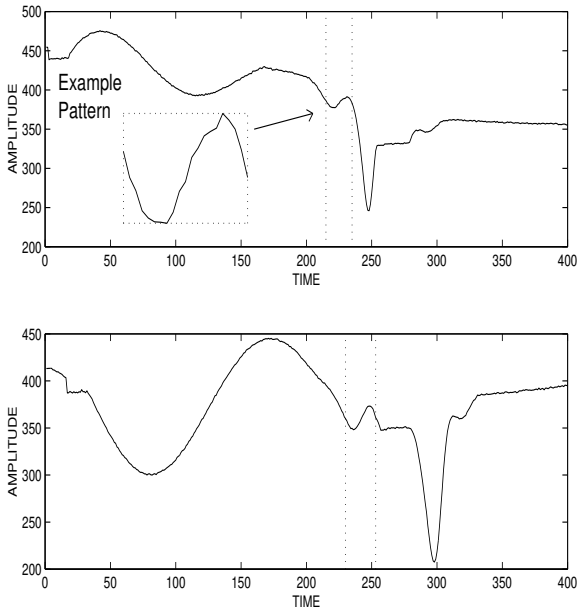

Figure 1: An example of an interferometry sensor from a semiconductor manufacturing process: (a) (top) a waveform pattern indicating the end of the plasma etch process is indicated with dotted line, (b) (bottom) another run of the same process where we wish to detect a similar pattern (indicated by dotted lines).

manufacturing environment the pattern must be detected *online* in real-time, as sensor data is measured (this automated detection enables automated control of the process). Current detection technology using existing sensors relies on threshold-based techniques for endpoint detection. This approach typically only works well if the end-point pattern consists of a simple increase or decrease in the level of the sensor. However, more sophisticated sensors can generate relatively complex endpoint patterns, such as the waveform in Figure 1(a), and thresholding is no longer applicable.

An example of a new run of the process is provided in Figure 1(b), where the estimated location of the pattern Q has been determined (manually) in the indicated window. Accurate online detection of the pattern Q is critical for automatic control of the etch step in the process: if the etch-step ends too early or too late the semiconductor wafer will not be etched properly, resulting in significant financial loss. We will return to this particular semiconductor manufacturing problem in more detail in Section 6 where we discuss experimental results.

This same type of waveform recognition problem (also sometimes referred to as "subsequence matching") occurs in a variety of data mining contexts. In interactive data exploration of time-series archives (for example in finance or marketing) a data analyst may wish to know if this week's pattern has ever occurred before, or if the weekly pattern of customer-visits over time at one particular store is common to other stores. In diagnosis and fault detection an engineer may wish to query an archival database in real-time to determine what past situations (contexts) are most similar to the current sensor pattern Q (e.g., see [12] for an example from space shuttle sensor monitoring). A third data mining task is unsupervised discovery of patterns in time-series; any data mining algorithm that tries to discover recurring (previously unknown) patterns in a data set will need to be able to solve this "waveform matching" problem as a primitive operation to support such unsupervised discovery (e.g., [6]). Thus, the problem of detecting waveforms in time-series has broad applicability and relevance to data mining of time-series data.

The inherent difficulty of this type of recognition problem typically stems from the inherent variability in both the waveform Q to be detected and the "background" variability in the time-series R_i. For example, based on the physics of the associated plasma etching process, it is known that the specific shape of the "end-point pattern" Q in Figure 1(a) can vary substantially from run to run. Nonetheless, each realization of the pattern still possesses the same inherent general shape characteristics that allows a human observer to detect it in a relatively straightforward, almost gestalt, manner.

A key point here is the notion of *shape variability*. If we can characterize systematically the manner in which a shape can vary in "shape-space" then in principle we have a sound footing from which to engineer a detection system since we can characterize which types of variations are expected and which are not. This point has not escaped researchers in the computer vision and image analysis community, who have pioneered the notion of shape-space variability for the potentially more difficult problem of *two-dimensional pattern detection* in recent years. Work such as [3, 15] is seminal in this context, using the notion of identifying landmark points to represent the shape of an outline in 2d, and then characterizing shape variability among a population of such outlines in terms of a shape-space distributions on the vector of landmarks. Our work here is inspired by this general shape-space framework, but applied to 1-dimensional waveform shapes. By leveraging the natural constraints imposed by a 1-dimensional time axis (rather than 2d images), we are able to use a richer underlying representation for our shapes than simply landmark-based models (namely generative state-based Markov models), in turn leading to efficient and accurate detection algorithms.

3. BACKGROUND AND MOTIVATION

3.1 Related Work

The simplest (and weakest) approach to this problem is to use direct template-matching (aka sequential scanning), i.e., compute the amplitude distance, point by point, between the pattern waveform Q and subwindows within R, and calculate the mean-squared error. If the error is below some threshold a detection is declared. Figure 2 illustrates the limitations of this approach, using our plasma etch data as an example. There are several minima of the root mean square error (RMSE) for different subwindows in R, several of which are below the RMSE value at the "obvious" location based on human judgment at about time point 240, where this time point was determined by an engineer familiar with the process. A number of variations on this theme

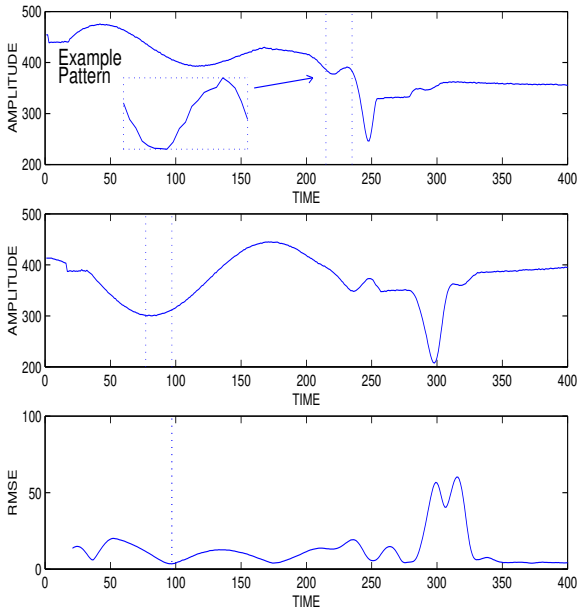

Figure 2: Results of applying a simple template matching technique to the data in Figure 1.

allow for some slack in the matching process. However, an underlying problem is that there is no notion of variability in the template Q. Thus, even small variations in a new occurrence of the waveform may lead to large RMSE distances and large variations in new waveforms may produce small RMSE distances (e.g., the local minima around times 40, 95,175 and 275 in the bottom portion of Figure 2).

Dynamic time-warping (DTW) generalizes the template-matching approach to explicitly allow for some slack in the matching of the time-axes of Q and R. How much slack is tolerated is encoded by the choice of distance function in the DTW matching algorithm. [4] introduced the concept to time-series data mining. The main problem with DTW in a data mining context is that construction of an effective distance measure can be highly non-trivial and very problem-dependent. A second general problem is that DTW focuses only on one specific type of pattern variability, namely elasticity in time, whereas in practice other deformations may also be present.

There has bee substantial interest in this problem in the data mining literature (e.g., [1, 7, 2, 17, 20, 5, 10, 13]). Much of this work can be characterized procedurally in the following general manner: (1) find an approximate and robust representation for the time-series (e.g., Fourier coefficients, piecewise linear models, etc.), (2) define a flexible matching function which can handle various pattern deformations (scalings, transformations, don't cares, etc), and (3) provide an efficient scalable algorithm, using this representation and this matching function, for massive time-series data sets. While these approaches in general provide a wealth of useful heuristics for waveform matching, they do not explicitly account for uncertainty in the matching process, i.e., there are no probabilistic semantics associated with the matching process, Consequently, one cannot quantify in general the inherent uncertainty associated with any detection decision. A second (related) limitation is that they do not provide a coherent quantitative mechanism for *adaptation*, i.e., either adapting the detection algorithm on the basis of training data and/or via prior knowledge. As we will see in later sections, the probabilistic probabilistic framework in this paper can handle these issues in a systematic and straightforward manner.

3.2 Motivation for our Approach

All of the work above can be characterized as being *distance-based* (where we use the word "distance" in the loose sense of computing some scalar dissimilarity measure between two waveforms, rather than a formal distance metric). In other words, the focus of these approaches is largely based on defining flexible distance measures for matching two waveforms Q and Q', given some suitable underlying representation for the waveforms. A significant practical problem here is that a distance measure that is optimized for one type of pattern and application domain (e.g., detecting arrythmia patterns in cardiac monitoring) may be entirely inappropriate (and ineffective) when applied to a different domain (e.g., financial data analysis). Thus, we question whether it is really feasible to find "general-purpose" universal distance metrics which are broadly useful. The question remains of course how then should one invent a new distance metric for every new application?

One answer is provided by the general approach of *probabilistic generative modeling*, which is fundamentally different to the distance based concept. In simple terms, this means that we construct a *model* for Q (call it M_Q, typically it defines a probability distribution on waveforms). From this model we can generate or simulate sample waveforms (i.e., other realizations of waveforms Q from an assumed data generating process M_Q). Typically this model consists of a mean shape and a distribution function which describes variation about this mean shape.

From a matching perspective an important point is that we can measure the "similarity" of any new pattern Q' to our model M_Q simply by computing $p(Q'|M_Q)$, the *likelihood* that Q' came from M_Q. (Usually we take negative log-likelihood to be a distance function, $-\log p(Q'|M_Q)$, which will be smaller the closer Q' is to the model M_Q). The generality and simplicity of the likelihood definition underlies both the elegance and power of the approach. The distance measure (between Q' and Q) is implicitly specified by $-\log p(Q'|M_Q)$ once the model M_Q is defined. In other words there is no need to construct an ad hoc distance measure between patterns, since it is automatically defined by the likelihood of the model.

For example, if M_Q generates patterns of fixed length with a specific mean shape and additive Gaussian noise, our likelihood measure (more specifically the negative of the log of the Gaussian likelihood, $\log p(Q'|M_Q)$) simply reduces to the Euclidean distance between Q and Q'. The power of the approach lies in how we can generalize this concept to much more expressive models M_Q. Indeed this general idea is not new and implicitly permeates much work in statistical pattern recognition. The hidden Markov model (HMM) ap-

proach to speech recognition uses precisely this framework where Q' is an observed acoustic waveform and we have a set of M_Q's for different words.

In a data mining context, [12] proposed a version of probabilistic generative models, but where the probabilistic model was defined on the observed deformations between two waveforms, i.e., the method as proposed still required a definition of a distance measure. The flexible Markov generative model we propose in this paper is a more direct approach, as well as providing a more flexible modeling framework.

4. SEMI-MARKOV AND SEGMENTAL MARKOV MODELS FOR WAVEFORM REPRESENTATION

An influential idea in pattern recognition is recognition by parts, i.e., decomposing an object into a model composed of (a) individual components, and (b) the relations (temporal or spatial) between these components. Recognition then becomes a matter of detecting individual components and then "parsing" their likely configurations relative to each other. Following this line of thought, we choose to model a waveform as K distinct segments with constraints on how the segments are "linked."

4.1 Segmental Observation Models

We begin our discussion with a standard discrete-time finite-state Markov model where each segment in the data corresponds to a state of the Markov model. Let the number of states be K. The parameters of the model include π, the initial state distribution (typically we will constrain the waveform to always begin with the same segment), and A, the $K \times K$ state transition matrix (again, this transition matrix can be constrained to be a "left-to-right" model which enforces a strict ordering in time of the segments). Let $\mathbf{y} = y_1 y_2 \ldots y_t \ldots y_T$ be the observed waveform measurements. The corresponding states are defined as $\mathbf{s} = s_1 s_2 \ldots s_t \ldots s_T$ (i.e., segment labels) and are hidden (not observed directly). In this *hidden* Markov model, the joint distribution of the observed data sequence $\mathbf{y}$ and a state sequence $\mathbf{s}$, can be factored as:

$$p(\mathbf{y}, \mathbf{s}) = \left(\prod_{t=2}^{T} p(y_t|s_t) p(s_t|s_{t-1}) \right) p(y_1|s_1)\pi(s_1), \qquad (1)$$

We have not yet described the functional form of the conditional densities $p(y_t|s_t)$ which relate the observed data to the hidden states. In the standard HMM framework (e.g., in speech recognition) the real-valued y_t's are often modeled as Gaussians or mixtures of Gaussians. For Gaussians, this implies a piecewise constant process with one mean μ_i per state i with additive Gaussian noise. Mixtures allow switching between multiple means per state, but still imply a constant "shape" process within each state as a function of time.

For waveform modeling this assumption of a constant shape plus noise is often inappropriate, e.g., the waveform pattern in Figure 1 could not be modeled parsimoniously using a piecewise constant model. A natural generalization of the constant model is to allow each state (or segment) to generate data in the form of a regression curve, i.e.,

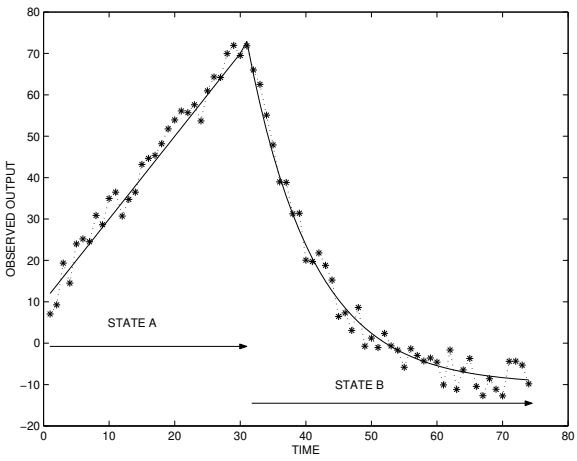

Figure 3: A simple illustration of the output of a simulated segmental Markov model. The solid lines show the underlying deterministic components of the regression models within each state, and the dotted lines show the actual noisy observations.

$$y_t = f_i(t|\theta_i) + e_t \qquad (2)$$

where $f_i(t|\theta_i)$ is a state-dependent regression function with parameters θ_i and e_t is additive independent noise (often assumed Gaussian, but not necessarily so). For Gaussian noise we have that $p(y_t|s_t = i)$ is Gaussian with a mean $f_i(t)$ which is a function of time and with variance σ^2. Note that conditioned on the regression parameters θ_i, the y_t's only depend on the current state s_t, as in the standard regression framework (i.e., observations are conditionally independent of everything else given the current state and state regression parameters). Thus, the likelihood of the data can be still be expressed in the simple product form of Equation 1, but now the $p(y_t|s_t)$ terms are dependent on the time t (as in Equation 2) rather than being constant. An important point is that this product form for the likelihood makes both parameter estimation (model learning) and inference (pattern detection) relatively straightforward—we will take advantage of this fact when we apply this technique to waveform detection later in this paper.

This segmental Markov model [9] is a natural one for modeling waveform shapes. It decomposes the waveform Q into local segments, each of which consists of a parametric functional "shape" with additive noise, and the segments are "linked" in a Markov manner. The problem of finding the best-fitting parameters for this model, given observed data in the form of a particular waveform Q (or set of waveforms), is quite straightforward as long as the parameters θ_i appear linearly in the shape functions $f_i(t|\theta_i)$. Figure 3 shows a simple example of the output of a simulated segmental Markov model. The process begins in state A which produces observations according to a noisy linear regression model. After 30 time steps or so it transitions to state B and produces observations according to a noisy exponential decay model.

4.2 Duration Modeling with Semi-Markov Processes

In the standard Markov framework the distribution of the durations of the system in state i is given by

$$p_i(t_d = d) = a_{ii}^{d-1}(1 - a_{ii}) \quad (3)$$

where a_{ii} is the self-loop transition probability of state i and d is the number of time-steps spent in state i. In other words, the Markov assumption constrains the state-duration distributions to be geometric in form. In reality we will want a more flexible way to model duration distributions to reflect the fact that each segment of the waveform being modeled has a typical duration length (mean time) and some variability around that mean time.

The problem of modifying the standard Markov model to allow for arbitrary state-durations can be addressed by the use of *semi-Markov* models (e.g., [8]). A semi-Markov model has the following generative description:

- On entering state i a duration time t_d is drawn from a state-duration distribution $p_i(t_d)$.
- The process remains in state i for time t_d.
- At time t_d the process transitions to another state according to a transition matrix A, and the process repeats.

The state-duration distributions, $p_i(t_d), 1 \leq i \leq M$, can be modeled using parametric distributions (such as log-normal, Gamma, etc) or non-parametrically by mixtures, kernel densities, etc. If t_d is constrained to take only integer values we get a discrete-time semi-Markov model. For waveform modeling, by including the state-duration distributions in the model, we can encode a prior on how long we expect the process to remain in each state.

We can combine this semi-Markov approach with the segmental hidden Markov model described in the last section. This gives us a flexible framework for defining distributions $p(Q)$ over waveforms Q. This waveform model allows us to specify shape of the waveform within each segment, the mean and variance of the duration length for each segment, and the ordering of the segments. Having a probabilistic model for our waveforms has certain distinct advantages that are worth mentioning explicitly:

- we can estimate the parameters of our model from data,
- we can combine these estimates with prior knowledge using Bayesian priors, and
- we can quantify the likelihood that the waveform Q occurs in any arbitrary position in a time-series R.

We elaborate further on these issues in the discussion in the next section below.

5. WAVEFORM PATTERN MATCHING

Given the general framework presented above we now formulate a specific algorithm for detecting a waveform pattern Q in a time-series R. Specifically, given an example of the waveform Q, we show how to construct a semi-Markov segmental hidden Markov model to fit this waveform, and then give a computationally efficient solution for detecting any instances of this waveform which are embedded in a time-series R.

5.1 Constructing a Waveform Model from Data

The construction of the model begins with a piecewise linear representation of the example waveform pattern using a standard piecewise linear segmentation algorithm [11, 21]. For illustration, we show in Figure 4 the piecewise linear representation of the example waveform pattern Figure 1(a), where we arbitrarily set the error tolerance $\epsilon = 1.0$ for the segmentation algorithm so that the approximating error at each point will not exceed ϵ in amplitude on this data. We could also automatically set the error tolerance by calculating the noise scale in the data, e.g., by filtering, or Fourier transformations. We are assuming each segment to be linear, although polynomials, splines, etc., could also be used.

Let the number of segments in the pattern be K. We construct a K-state segmental HMM each state of which corresponds to one segment in the piecewise linear representation. Because one can only start from segment 1, then go to segment 2, etc., the transition matrix A will be left-to-right, i.e., $A_{i,i+1} = 1$, $A_{i,j} = 0$ if $j \neq i + 1$ and $A_{i,j}$ is the probability of going to state j given that the process is in state i. The initial state distribution will be $\pi = [1, 0, \ldots, 0]$. The output probability distribution of state i will be of the form

$$p(y_{m+1}y_{m+2}\ldots y_{m+d_i}|s_i) = p(d_i|s_i)p(\theta_i|s_i)\prod_{t=m+1}^{m+d_i} p(y_t|f_i(\theta_i, t)) \quad (4)$$

where

- $p(d_i)$ is the probability of the duration d_i (i.e., the length of the segment in time),
- θ_i is state i's parameter of the regression function which has functional form $f_i(\theta_i, t)$,
- $p(y_t|f_i(\theta_i, t))$ is assumed to be a Gaussian distribution with mean $f_i(\theta_i, t)$ and variance σ_y^2.

Since we are using piecewise linear representation of the example waveform pattern here, the regression function $f_i(\theta_i, t)$ will be a linear function $f_i(\theta_i, t) = b_i t + c_i$. Of the two parameters, the intercept c_i is ignored in the model and allowed to be freely fit in the detection process to allow shifting in time. Thus, θ_i includes only b_i which is set to be the slope of the segment in the example waveform pattern. In the absence of prior knowledge about how this slope will change, for simplicity, we assume θ_i to be fixed (non-random) in Equation 4.

The state duration distribution $p(d_i|s_i)$ for state i is set to be a left-truncated Gaussian distribution with mean being l_i,

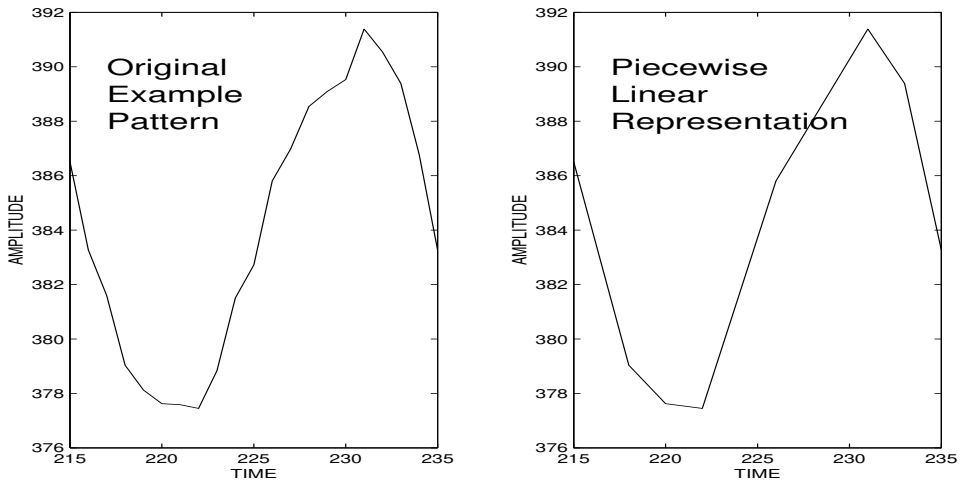

Figure 4: The example waveform pattern Figure 1(a), and its piecewise linear representation.

the length (in time) of the corresponding segment in the example pattern, and standard deviation being $l_i \times k\%$ (where k was set to 20 for the results reported below).

The variance of the additive noise, σ_y^2, is set to be the mean squared error of the piecewise linear representation (as fitted to the original data).

As described here the training procedure essentially amounts to setting the means and variances in an appropriate data-dependent manner based on a single waveform observation. The generalization to training a model from multiple waveforms is straightforward, if all waveforms are represented by the same number of linear segments. A completely unsupervised approach (using multiple waveforms) is to directly train a semi-Markov segmental hidden Markov model using the EM algorithm. Here the number of segments (states) is fixed a priori and EM determines the mean lengths and shapes for each segment by maximizing the overall likelihood of the observed waveform data.

5.2 A Viterbi-like Algorithm to Compute the Most Likely State Sequence

Once we have a semi-Markov segmental hidden Markov model M_Q for our waveform(s) Q, a basic task is to find the most likely state (i.e., segment label) sequence $\hat{\mathbf{s}} = s_1 s_2 \ldots s_t \ldots$ for a data sequence $\mathbf{y} = y_1 y_2 \ldots y_t \ldots$.

Here we give a recursive Viterbi-like algorithm based on dynamic programming. At each time t, this algorithm calculates the quantity $\hat{p}_i^{(t)}$ for each for each state i, $1 \leq i \leq K$, where $\hat{p}_i^{(t)}$ is defined as

$$\hat{p}_i^{(t)} = \max_{\mathbf{s}} \{p(\mathbf{s}|y_1 y_2 \ldots y_t)|\mathbf{s} = s_1 s_2 \ldots s_t,\ s_t = i\}. \quad (5)$$

In other words, $\hat{p}_i^{(t)}$ is the likelihood of the most likely state sequence that ends with state i (i.e., y_t is the last point of segment i).

The recursive function for calculating $\hat{p}_i^{(t)}$ is

$$\hat{p}_i^{(t)} = \max_{d_i} \left(\max_j \hat{p}_j^{(t-d_i)} A_{ji} \right) p(d_i) p(y_{t-d_i+1} \ldots y_t|\theta_i) \quad (6)$$

In the above equation, the outer maximization ($\max_{d_i}$) is over all possible values of the duration d_i of state i. Recall that y_t is now fixed to be the last point of segment i, the last point of the previous segment will be $t - d_i$. For a given d_i, the inner maximization ($\max_j$) is over all possible previous states j that transitions to state i at time $t - d_i$. The state j and the time $t - d_i$ for the maximum value $\hat{p}_i^{(t)}$ are recorded in $PREV(i, t)$.

Obviously, the overall most likely state sequence for the data sequence $y_1 y_2 \ldots y_t$ will be the state sequence with the likelihood $\max_i \hat{p}_i^{(t)}$, and can be found by tracing back using $PREV(i, t)$ (see the pseudocode "MLSS" in Figure 5). Note that this detection procedure is optimal in a maximum likelihood sense, i.e., it finds the state-sequence which is most likely to account for the observed data.

5.3 Online Detection of the Waveform

To detect a waveform inside a (much longer) time series $y_1 y_2 \ldots y_t \ldots$, an obvious approach would be to match the model against every subwindow $y_i y_{i+1} \ldots y_j$, find the most likely state sequence $s_i s_{i+1} \ldots s_j$, and declare "found" if the likelihood is above a certain threshold. The problems with this approach are (1) how to set the likelihood threshold and (2) the redundant computation from the fact that the computation for every subwindow is carried out from scratch, even if a subwindow overlaps with another subwindow.

To deal with these problems, we augment the model with two extra "background" states: a *pre-pattern* background state (state 0) to model the data before the pattern, and a *post-pattern* background state (state $K + 1$) for the data after the pattern. This augmented model may be seen as a "global" model that can be matched directly against the whole time series $y_i y_{i+1} \ldots y_j$ (instead of the subwindows). These background states are also called garbage states in

```
function s_1 s_2 ... s_t = MLSS(y_1 y_2 ... y_t)
1.  for each state i
2.      Compute p̂_i^(t), PREV(i,t);
3.  end for
4.  j = argmax_i p̂_i^(t);
5.  [j', t'] = PREV(j,t);
6.  for k = t' + 1 to t
7.      s_k = j;
8.  end for
9.  if (t' > 0)
10.     [j, t] = [j', t'];
11.     goto 3;
12. else
13.     return;
14. end if
```

```
procedure DETECT(y_1 y_2 ... y_t ...)
1.  t = 1;
2.  s_1 s_2 ... s_t = MLSS(y_1 y_2 ... y_t);
3.  if (s_t == K)
4.      declare 'found';
5.      stop;
6.  else
7.      t = t + 1;
8.      goto 2;
9.  end if
```

Figure 5: Pseudo-code for MLSS (finding most likely state sequence $s_1 s_2 \ldots s_t$ for data sequence $y_1 y_2 \ldots y_t$), and DETECT (online detection of waveform).

the speech recognition literature [19]. The parameters of the background states (e.g., the state duration distribution, the noise variance, parameter θ of the regression function, etc.) are estimated in a similar way to other (ordinary) states in the model. They could instead be set according to prior knowledge. For example, the probability distribution on the state duration of the pre-pattern background state (i.e., its length in time) can be set to reflect a prior probability distribution of the starting time of the pattern.

With this augmented model, we run the MLSS algorithm in Figure 5 online as new data points $y_1 y_2 \ldots y_t \ldots$ are coming in. If, at time t, $s_t = K$ in the most likely state sequence where K is the last segment in the waveform, we declare that the waveform is detected with end-time y_t. See the pseudocode DETECT in Figure 5.

5.4 Complexity of the Method

The complexity of the above detection algorithm will be $O(|R| \times (K+2) \times C)$ where $|R|$ is the size of the new time series R, $K+2$ is the number of states (K pattern-states and 2 background states), and C is the complexity of evaluating Equation 6. Generally speaking, C will be $O(v_i \times (K+1))$ where v_i is the number of possible values for d_i (the duration of state i), $K+1$ is the maximum number of predecessor states for any state. In the current setting where the transition matrix A is left-to-right, each state i has only one predecessor state, so $C = O(v_i)$. Letting $V = \max_i v_i$, the overall complexity of the algorithm is $O(|R| \times K \times V)$. Thus, this method is essentially linear in $|R|$.

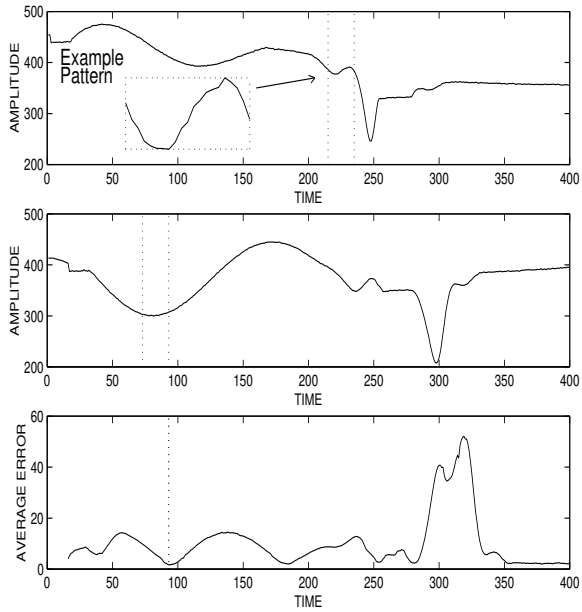

Figure 6: Results of applying dynamic time-warping to the data in Figure 1.

6. EXPERIMENTAL RESULTS

6.1 Results on Plasma Etch Process Data

Plasma etch [14, 18] is a critical process in semiconductor manufacturing. A semiconductor wafer is bombarded with a gas containing various chemical components within a plasma gas chamber. Different gas combinations are used in sequence to remove different layers from the wafer. Since there is no direct way of measuring when a layer has been etched through, control of the plasma process (i.e., when to halt gas flow so as to stop etching) is achieved by inferring the nature of the material being etched indirectly from the composition of the gas within the chamber. For example, from the interferometry data in Figure 1(a), an engineer can manually detect the endpoint by seeing the pattern (enclosed in the dotted rectangle). Here is where the pattern matching algorithm fits in. We would like to detect the same pattern in the interferometry data of future runs, e.g., Figure 1(b).

For comparison, we ran both the template-matching (Figure 2) and dynamic time warping (Figure 6) techniques. The global minima in the two resulting error curves do not correspond to the best match, i.e., both minima correspond to false alarms around time $t = 90$ seconds.

We built a semi-Markov segmental model for the pattern in Figure 1(a), and ran the pattern matching algorithm (the DETECT procedure in Figure 5) on the new time series in Figure 1(b). At time $t = 252$, the DETECT procedure correctly detected the end of the pattern, i.e., the maximum of the score function agrees precisely with the engineer's subjective judgement on the location of the end-point.

It is also interesting to look at the normalized likelihood $[\hat{p}_K^{(t)}]^{\frac{1}{t}}$ for state K as a function of time t. (Recall that $\hat{p}_K^{(t)}$ is the likelihood of the most likely state sequence ending

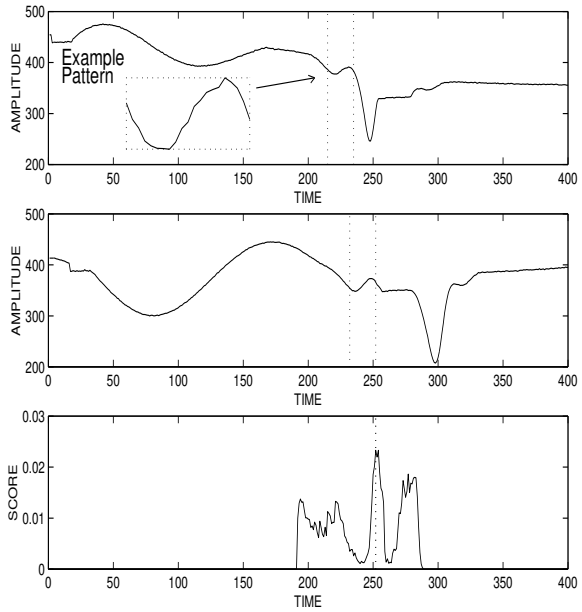

Figure 7: Results of applying the semi-Markov segmental model to the data in Figure 1. The SCORE is $[\hat{p}_K^{(t)}]^{\frac{1}{t}}$, the normalized likelihood for state K (the last segment of the pattern).

with state K at time t, and K is the last segment of the pattern. If $\hat{p}_K^{(t)}$ is the highest among all states, then the pattern is detected.) We plot $[\hat{p}_K^{(t)}]^{\frac{1}{t}}$ as the SCORE in Figure 7. It can be seen that the peak of the SCORE curve is at $t = 252$ which is when the online DETECT procedure finds the pattern.

6.2 Results on Financial Data

A second dataset is the New York Stock Exchange (NYSE) daily index closes for 1999 (available online at `http://www.nyse.com/marketinfo/stats/Nya99.prn`.) This dataset contains, among others, the NYSE Composite Index, and the Utility Index, as shown in Figure 8. These two indices, like the two semiconductor manufacturing sensor runs, share some common patterns, but are still quite different from each other (e.g., the amplitude of the Composite Index is between 560 and 680, while the Utility Index is between 450 and 600.) In Figure 8, we show the results of selecting a pattern in the NYSE Composite Index (top), building the semi-Markov segmental model, then searching for this pattern in the NYSE Utility Index (bottom). The algorithm correctly finds a similar pattern.

6.3 Results on ECG Time Series

The third data set (Figure 9) is an ECG time series (`http://www.ms.washington.edu/~s530/data.html`) from [16] . A heart-beat pattern was selected from an early part of the time-series (Figure 10, top) and used to build a semi-Markov segmental model using the default method described in Section 5.1. The online detection algorithm of Section 5.3 was then used to detect later occurrences of the pattern as in Figure 10 (bottom). Note that the general method described in

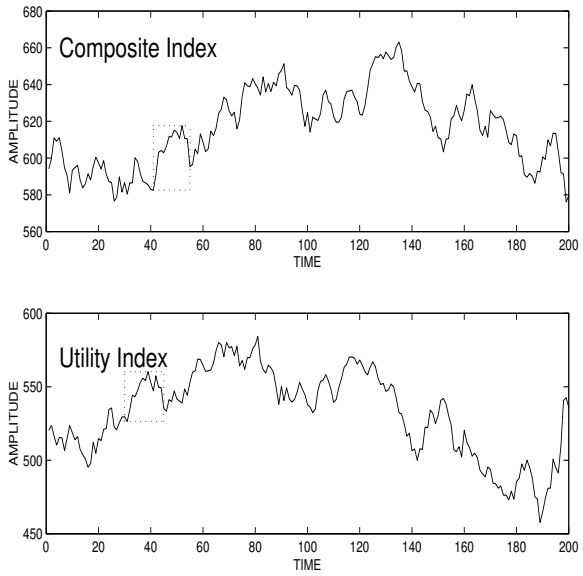

Figure 8: Results of applying the semi-Markov segmental model to financial time-series. A pattern is picked from NYSE Composite Index (top) to build the model, then we search for this pattern in the NYSE Utility Index (bottom).

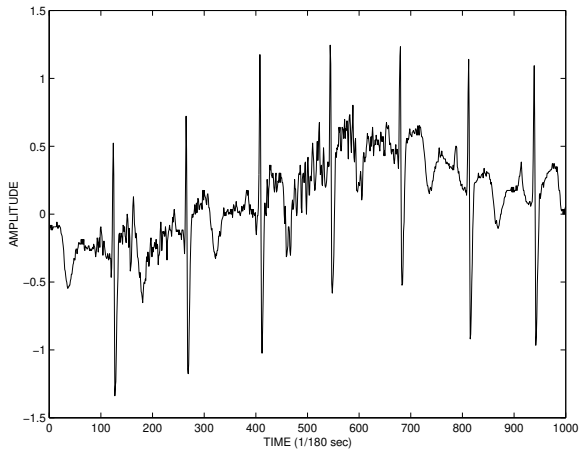

Figure 9: ECG time series.

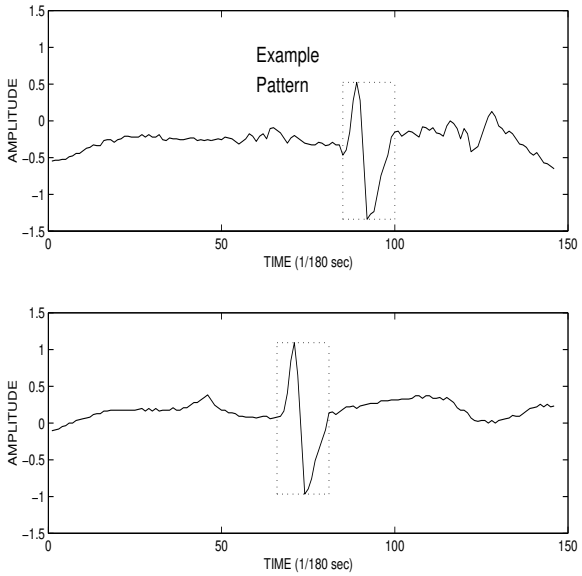

Figure 10: Results of applying the semi-Markov segmental model to recognize the "heart beat" pattern in the ECG time series. Top: an example pattern. Bottom: a similar pattern found by the pattern matching algorithm.

Section 5.3 for online detection of a single pattern can easily be generalized to the problem of finding multiple repeating patterns (e.g., multiple repeating heart-beats as in this data set) by allowing transitions in the Markov model from the end state back to the start state, i.e., allowing repeated transitions through the state sequence.

7. DISCUSSION AND CONCLUSIONS

Overall we have found the proposed technique to be quite robust. The main parameter that is manually chosen is the standard deviation in the state duration models, which was set to 20% of the mean duration in the results described above. Generally speaking this should be a function of prior knowledge, i.e., how much variation in time do we expect in waveforms? On the data sets above we have found empirically that the method finds the ideal detection over a broad range of values both above and below 20%, indicating that the method is not overly sensitive to this parameter. The degree of sensitivity will of course in general be dependent on the nature of the both the waveform Q and the background time series R. Of course if one is training on multiple waveforms this variance term can be estimated directly.

There are a number of natural extensions of the proposed framework which we do not discuss in detail here due to space limitations. For example, in the semiconductor manufacturing process there are multiple runs (and associated endpoints) over time. One can extend the framework proposed here to include an online adaptive Bayesian component, where the model M_Q is adapted over time as new time series are added to the archive. An interesting issue here is how to incorporate both subjective human judgement (supervised labeling of waveforms) with the type of unsupervised detection demonstrated here. Another research direction is to embed this framework within a hierarchical Bayesian model with random variation in the segment coefficients rather than keeping them fixed. The underlying semi-Markov segmental hidden Markov model for waveforms can also be generalized, allowing for further modeling flexibility such as stochastic grammars, hierarchical hidden Markov models, and so forth.

In summary, we have proposed a general model-based framework for waveform matching, using generalized Markov models to encode the shape of a waveform. We believe that this can provide a more flexible and practical methodology than competing distance-based methods. For example, we illustrate how the model can be learned from data and how we can perform detection in a relatively automatic fashion using basic likelihood concepts. Based on these results we conclude that the proposed semi-Markov segmental HMM appears to be quite a useful, flexible, and accurate framework for time-series waveform pattern detection.

Acknowledgements

We would like to thank Wenli Collison, Tom Ni, and David Hemker of LAM Research for providing the plasma etch data and for discussions on change-point detection in plasma etch processes. The research described in this paper was supported by NSF CAREER award IRI-9703120 and by the NIST Advanced Technology Program and KLA-Tencor.

8. REFERENCES

[1] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. In *Proc. of the 4th International Conference on Foundations of Data Organization and Algorithms*, pages 69–84, Oct 1993.

[2] R. Agrawal, K.-I. Lin, H. S. Sawhney, and K. Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *VLDB*, pages 490–501, Sep 1995.

[3] Y. Amit, U. Grenander, and M. Piccioni. Structural image restoration through deformable templates. *Journal of the American Statistical Association*, 86:376–387, 1991.

[4] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD-94: AAAI Workshop on Knowledge Discovery in Databases*, pages 359–370, July 1994.

[5] K.-P. Chan and A. W.-C. Fu. Efficient time series matching by wavelets. In *Proceedings 15th International Conference on Data Engineering*, pages 126–133, Mar 1999.

[6] G. Das, K. Lin, H. Mannila, G. Rengenathan, and P. Smyth. Rule discovery from time series. In *Proceedings of the 1998 Conference on Knowledge Discovery and Data Mining*, pages 16–22. AAAI Press, 1998.

[7] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *SIGMOD–Proceedings of Annual Conference*, pages 419–429, May 1994.

[8] J. D. Ferguson. Variable duration models for speech. In *Proc. Symposium on the Application of Hidden Markov Models to Text and Speech*, pages 143–179, Oct 1980.

[9] W. J. Holmes and M. J. Russell. Probabilistic-trajectory segmental HMMs. *Computer Speech and Language*, 13:3–37, 1999.

[10] Y.-W. Huang and P. S. Yu. Adaptive query processing for time-series data. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 282–286, 1999.

[11] H. Imai and M. Iri. An optimal algorithm for approximating a piecewise linear function. *Journal of Information Processing*, 9(3):59–62, 1986.

[12] E. Keogh and P. Smyth. A probabilistic approach to fast pattern matching in time series databases. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining - KDD 97*, pages 24–30, Aug 1997.

[13] E. J. Keogh and M. J. Pazzani. An indexing scheme for fast similarity search in large time series databases. In *Proc. Eleventh International Conference on Scientific and Statistical Database Management*, pages 56–67, Jul 1999.

[14] D. M. Manos and D. L. Flamm, editors. *Plasma Etching, An Introduction*. Academic Press, Inc., San Diego, 1989.

[15] K. V. Mardia and I. L. Dryden. *Statistical Shape Analysis*. John Wiley & Sons, Ltd, 1998.

[16] D. B. Percival and A. T. Walden. *Wavelet Methods for Time Series Analysis*. Cambridge University Press, 2000.

[17] H. Shatkay and S. B. Zdonik. Approximate queries and representations for large data sequences. In *Proceedings of the Twelfth International Conference on Data Engineering*, pages 536–545, Feb 1996.

[18] P. F. Williams, editor. *Plasma Processing of Semiconductors*. Kuwer Academic Publishers, 1997.

[19] J. G. Wilpon, L. R. Rabiner, C.-H. Lee, and E. R. Goldman. Automatic recognition of keywords in unconstrained speech using hidden Markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing,*, 38(11):1870–1878, Nov 1990.

[20] B.-K. Yi, H. V. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. In *Proceedings. 14th International Conference on Data Engineering*, pages 201–208, Feb 1998.

[21] Y. Zhu and L. D. Seneviratne. Optimal polygonal approximation of digitized curves. *IEE proceedings. Vision, image, and signal processing*, 144(1):8–14, Feb 1997.