Energy Consumption Forecasting via Order Preserving Pattern Matching

N. Denizcan Vanli*, Muhammed O. Sayin*, Hikmet Yildiz*, Tolga Göze† and Suleyman S. Kozat*
*Department of Electrical and Electronics Engineering
Bilkent University, Ankara, Turkey 06800
Email: {vanli@ee, sayin@ee, hyildiz@ug, kozat@ee}.bilkent.edu.tr
†Alcatel-Lucent, Istanbul, Turkey
Email: tolga.goze@alcatel-lucent.com

***Abstract*—We study sequential prediction of energy consumption of actual users under a generic loss/utility function. Particularly, we try to determine whether the energy usage of the consumer will increase or decrease in the future, which can be subsequently used to optimize energy consumption. To this end, we use the energy consumption history of the users and define finite state (FS) predictors according to the relative ordering patterns of these past observations. In order to alleviate the overfitting problems, we generate equivalence classes by tying several states in a nested manner. Using the resulting equivalence classes, we obtain a doubly exponential number of different FS predictors, one among which achieves the smallest accumulated loss, hence is optimal for the prediction task. We then introduce an algorithm to achieve the performance of this FS predictor among all doubly exponential number of FS predictors with a significantly reduced computational complexity. Our approach is generic in the sense that different tying configurations and loss functions can be incorporated into our framework in a straightforward manner. We illustrate the merits of the proposed algorithm using the real life energy usage data.**

***Index Terms*—Order preserving pattern matching, sequential prediction, online learning.**

I. Introduction

Due to rapid climate changes and increasing awareness of global warming, the demand for a low carbon future is steadily growing. A prevalent method to reduce carbon emissions is renewable and efficient energy production. For a successful realization of this goal, the energy profile (particularly, the energy usage patterns) of the consumers should be carefully analyzed. To accomplish this, we study the sequential prediction of energy consumption trend and introduce an algorithm to predict the future relative energy consumption of customers according to their past energy usage patterns. Specifically, observing past energy usage samples, we predict the trend of the samples, i.e., determine whether an increase or a decrease in the energy usage will happen in the future.

Since we are interested in the relative value of the future consumption, we use the relative ordering pattern of the energy consumption history to construct our decisions, as explained later in the paper. In this sense, the relative ordering of the data in the past corresponds to the state, context, or side information in our algorithms. To motivate this choice of states, one can argue that an uphill trend or a downhill trend in energy usage (or pricing) may continue in the future since decisions or actions of people usually depend on their past experiences and their future actions may be inferred from their previous behavior patterns [1]. In our experiments, we demonstrate that we can accurately predict the relative electric consumption of actual customers using their past consumption patterns.

State dependent (or pattern matching) prediction is extensively studied both in signal processing and computational learning theory literatures since this structure naturally arises in different real life applications, e.g., [2]–[5]. In these studies [2]–[5], the states (or equivalence classes) usually correspond to different partitions of the regressor space and independent predictors are assigned to each state. However, in this paper, we are interested in the trend of the energy consumption rather than its actual value. In this sense, both the state definitions and the prediction framework are substantially different in this paper with respect to [2]–[5].

We emphasize that since we seek to predict an increase/decrease yielding a binary prediction problem, this problem is more inline with the relevant studies in the information theory literature such as [6] (and references therein). The universal binary prediction algorithm in [6] is proven to achieve the performance of any batch FS predictor in the long run. Hence, for any choice of states, one can use the algorithm of [6] to achieve the performance of any state dependent predictor. However, such algorithms require a substantial amount of past information in order to provide satisfactory performance, which is not available even for decent energy consumption pattern lengths. Hence, these asymptotical results may not be acceptable over finite data lengths, therefore, one should also learn the definition of the best state among with the optimal FS predictor that minimizes the accumulated loss for that state. In this sense, although such asymptotical results apply in the long run, they are not applicable over finite length data sequences and for nonstationary data.

In this paper, we first introduce a sequential prediction algorithm where the state information is fixed, i.e., the relative ordering of the past data of length h is used as the state. We then introduce a hierarchical model that also sequentially learns the best state information from the data in order to minimize the prediction loss. Particularly, for all doubly exponential number ($\sim 2^{h^h}$) of FS predictors defined by the hierarchical model, we introduce a sequential algorithm that *i)*

978-1-4799-7088-9/14/$31.00 ©2014 IEEE

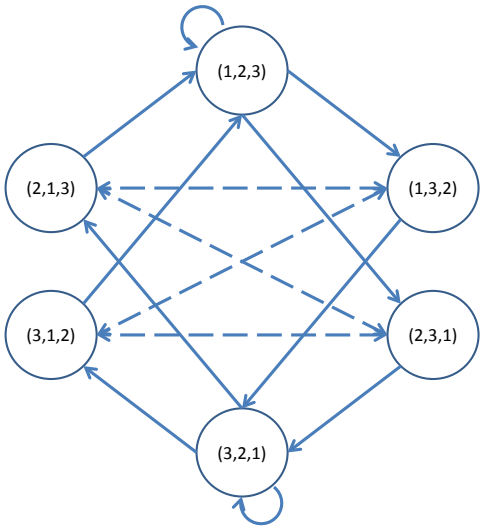

Fig. 1: Relative ordering patterns for $h = 3$, where solid lines represent one directional transitions and dashed lines represent bi-directional transitions.

achieves the performance of the optimal FS predictor among all FS predictors, *ii)* operates with a computational complexity linear in the pattern length, i.e., $O(h)$, *iii)* can incorporate any convex loss function as well as nested tying configuration in a straightforward manner.

II. FS Prediction Using Order Preserving Patterns

We sequentially observe a real valued sequence (i.e., the energy consumption data) $x_1, x_2, \ldots$ and produce an output $\hat{d}_t$ based on $x_1, \ldots, x_t$ at each time t, $x_t \in \mathbb{R}$. Then, the true d_t is revealed yielding a loss (or gain, according to the definition of the utility function) $l(d_t, \hat{d}_t)$ for some predetermined loss function $l(\cdot, \cdot)$. For any n, the accumulated loss is given by $\sum_{t=1}^{n} l(d_t, \hat{d}_t)$. We use a finite state (FS) predictor to produce the output $\hat{d}_t$, where the relative ordering patterns are selected as the states as shown in Fig. 1. In its most generic form, a FS predictor has a prediction function $\hat{d}_t = f_t(s_t)$, where s_t is the current state taking values from a finite set $s_t \in \mathcal{S}$, $\mathcal{S} = \{1, \ldots, S\}$, e.g., the set of relative ordering patterns. The states are traversed according to the next state function $s_{t+1} = g(s_t, x_{t+1}, x_t, \ldots, x_{t-h+2})$.

In this paper, we use the relative ordering pattern of the past sequence as our states. In particular, at each time t, we use the last h samples of the sequence history $x_{t-h+1}, \ldots, x_t$ to define equivalence classes or states. A length h sequence can have $h!$ different ordering patterns. As an example, for $h = 3$, we can have 6 different possible patterns as shown in Fig. 1, where "3" represents the location of the largest value and "1" represents the location of the smallest value, e.g., the sequence $\{x_{t-2}, x_{t-1}, x_t\} = \{5, -2, 3\}$ corresponds to the pattern or ordering $(3, 1, 2)$. Given h and this set of ordering patterns, one can arbitrarily assign each pattern to a state so that $s_t \in \{(1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), (3,2,1)\}$ for each t. After fixing the state assignments, s_{t+1} is known after observing x_{t+1}.

For such a state definition, one can easily construct a sequential algorithm asymptotically achieving the performance of the optimal batch FS predictor such as [6]

$$f_t(s_t) = \frac{\sum_{z=1}^{t-1} I_z^{\{s_t\}} d_z}{\sum_{z=1}^{t-1} I_z^{\{s_t\}}}, \tag{1}$$

where $I_z^{\{s_t\}}$ is the indicator function representing whether the length-h sequence corresponds to state s_t.

Although (1) sequentially learns the optimal batch FS predictor for each state based on the past occurrences of these states, it can only provide satisfactory results if there are enough occurrences of each state pattern in the past $x_1, \ldots, x_t$. However, for even moderate h that define meaningful patterns in real life applications [7], say for $h = 10$, the number of patterns grows as $h! \approx h^h = 10^{10}$. In this sense, to train (1) using ordering patterns, we require a substantial amount of past observations, which is not available in most real life applications even for stationary data. As described in the next section, one can mitigate this problem by defining "super set" equivalence classes or tying certain states together as widely used in speech recognition applications when there are not enough data to adequately train all the phoneme states [7].

III. A Hierarchical Order Preserving Predictor

Although a sequence of length h, $(x_{t-h+1}, \ldots, x_t)$, can have $h!$ different ordering patterns, most of these patterns share similar characteristics that can be exploited to group (or tie) them together to form different states each representing a collection of these patterns. In this paper, we use the appearance time of the elements as the main characteristics in order to group the patterns in a nested manner. As example in Fig. 2, for $h = 3$, we show how we hierarchically divide all possible patterns into different groups or equivalence classes. While we have the complete states at level $i = h - 1$, at each level $i < h - 1$, we group each $h - i$ ordering patterns from level $i+1$ into one of the $P(h, i) = h!/(h-i)!$ different equivalence classes starting from the oldest sample to the most recent one. As an example, at level-1, we combine the states $c_{2,5} = (1, 2, 3)$ and $c_{2,6} = (2, 1, 3)$ into the equivalence class $c_{1,3} = (\cdot, \cdot, 3)$ since the most recent element of both $c_{2,5}$ and $c_{2,6}$ are the largest one among the pattern.

With this definition of new equivalence classes, we have a smaller set of states and corresponding state predictors to train, which can be carried out by using much less observations of the data. Hence, at the beginning of the learning process, one can use this super set as a coarser representation that can be efficiently learned and then gradually switch to the original whole model with better modelling power as the data length increases. However, such super set definitions or switching between state sets can significantly effect the performance and their optimal selection are highly data dependent [3]. Furthermore, the effectiveness of the super sets or original sets may change over time, i.e., if the underlying data is highly nonstationary, then the whole model with all ordering patterns may never have enough data to adequately train predictors even if the data length increases to infinity. To this end, we introduce a sequential algorithm that elegantly and effectively performs such decisions by intrinsically implementing and

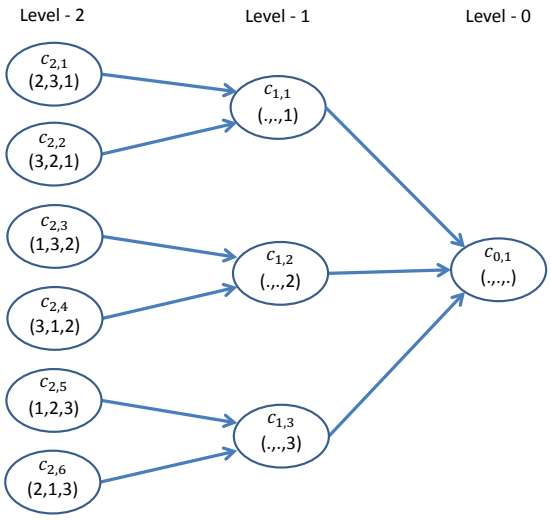

Fig. 2: The tying configuration for the relative ordering patterns with $h = 3$, where the equivalence classes at lower levels are formed by combining the equivalence classes at higher levels.

combining a huge number of ordering pattern based FS predictors.

A. A Universal Approach

We observe that various collections of the nodes in Fig. 2 completely covers all the original ordering patterns. As an example, the equivalence classes $\{c_{1,1}, c_{1,2}, c_{1,3}\}$ and $\{c_{1,1}, c_{2,3}, c_{2,4}, c_{1,3}\}$, completely covers the original set of the patterns $\{c_{2,1}, c_{2,2}, c_{2,3}, c_{2,4}, c_{2,5}, c_{2,6}\}$. Hence, each of these tying configurations can be used to construct a FS predictor by using equivalence classes as states and using the sequential method in (1) to produce prediction functions and the final output. For the introduced super set equivalence class definition with a history of length h, there are $K_h \approx 2^{h!} \approx 2^{h^h}$ different tying configurations (since $K_{h+1} = K_h^{h+1} + 1$), each of which completely covers the entire pattern set.

Suppose we construct all the FS predictors $\hat{d}_{t,k}$, $k = 1, \ldots, K_h$ and run them in parallel and predict d_t. We then combine the outputs of these FS predictors to produce a final weighted output

$$\hat{d}_t = \sum_{k=1}^{K} \mu_{t,k}\, \hat{d}_{t,k}, \tag{2}$$

where the combination weights measure the relative performance of each FS predictor on the past observations, i.e.,

$$\mu_{t,k} = \frac{\exp\left(-a \sum_{z=1}^{t-1} l(d_z, \hat{d}_{z,k})\right)}{\sum_{r=1}^{K_h} \exp\left(-a \sum_{z=1}^{t-1} l(d_z, \hat{d}_{z,r})\right)}, \tag{3}$$

and a is a positive constant controlling the learning rate by normalizing the total sum.

It can be shown that the weighted mixture algorithm (2) sequentially achieves the performance of the best algorithm in the mixture, i.e., when applied to any $x_1, x_2, \ldots$ and $d_1, d_2, \ldots$, yields the performance

$$\sum_{t=1}^{n} l(d_t, \hat{d}_t) \leq \min_{k=1,\ldots,K_h} \sum_{t=1}^{n} l(d_t, \hat{d}_{t,k}) + O(\log K_h), \tag{4}$$

for various loss functions [8], [9] such as the squared error loss $(d_t - \hat{d}_t)^2$, for any n without knowing the optimal $\hat{d}_{t,k}$ or the data length n. Hence, this sequential algorithm is as good as the any of the FS predictors that can be defined in Fig. 2.

However, in this form this algorithm cannot be implemented since even for a decent length pattern such as $h = 4$, we need to run $K_h = 6562$ FS predictors in parallel and monitor their performances to construct (2), which is clearly not plausible. In the next section, we introduce a method that implements (2) with complexity only linear in the pattern length h.

B. Low Complexity Implementation of (2)

For an efficient implementation of (2), we first assign a prediction function $\hat{d}_t^{(c_{i,j})}$ to each equivalence class $c_{i,j}$. Each equivalence class predictor is sequential and constructs its output based on its past such as in (1). We then note that although there are K_h FS predictors, a data sequence can only be included in only one equivalence class in each level. As an example, for the sequence $(5, -2, 3)$, only the equivalence classes $(3, 1, 2)$, $(\cdot, \cdot, 2)$, and $(\cdot, \cdot, \cdot)$ includes this pattern. Hence, although there are K_h different FS predictors, their outputs, at any time t, can only be the output of h different equivalence class predictors.

In order to use this observation, we first define a loss function for each equivalence class predictor $\hat{d}_t^{(c_{i,j})}$ as follows

$$L_t^{(c_{i,j})} \triangleq \exp\left(-a \sum_{z=1}^{t-1} l(d_z, \hat{d}_z^{(c_{i,j})})\, I_z^{(c_{i,j})}\right). \tag{5}$$

We also define a loss for the FS predictors, say for the k-th one, as follows

$$L_{t,k} \triangleq \exp\left(-a \sum_{z=1}^{t-1} l(d_z, \hat{d}_{z,k})\right). \tag{6}$$

Then using the observation, we conclude

$$L_{t,k} = \prod_{c_{i,j} \in \mathcal{C}_k} L_t^{(c_{i,j})}, \tag{7}$$

where $\mathcal{C}_k$ represents the set of all equivalence classes in the k-th FS predictor.

According to these definitions, the remaining question is to find an efficient scheme to calculate $\sum_{k=1}^{K_h} L_{t,k}$ and $\sum_{k=1}^{K_h} L_{t,k} \hat{d}_{t,k}$. To this end, for each equivalence class $c_{i,j}$, we define another recursion parameter

$$R_t^{(c_{i,j})} \triangleq L_t^{(c_{i,j})} + \prod_{c_{i+1,j'} \in \mathcal{D}^{(c_{i,j})}} R_t^{(c_{i+1,j'})}, \tag{8}$$

where $\mathcal{D}^{(c_{i,j})}$ represents the descendants of the equivalence class $c_{i,j}$. As an example, for the equivalence class $c_{0,1}$, we have descendant equivalence classes $\mathcal{D}^{(c_{0,1})} = \{c_{1,1}, c_{1,2}, c_{1,3}\}$. As can be shown after some algebra, if we expand the recursive formulation for $R_t^{(c_{0,1})}$, we get $R_t^{(c_{0,1})} = \sum_{k=1}^{K_h} L_{t,k}$, which is equal to the denominator of (3).

Algorithm 1 Universal Order Preserving Forecasting

% Initialization: $L_0^{\{c_{i,j}\}} \Leftarrow 1$, calculate $R_0^{\{c_{i,j}\}}$, $\forall c_{i,j}$.
for $t = 1$ **to** n **do**
  % Find the current state s_t.
  % Find the set of equivalence classes $\mathcal{E}_t$ containing s_t.
  % Calculate $\widetilde{R}_t^{(c_{0,1})}$, $\forall c_{i,j} \in \mathcal{E}_t$
  % Output $\hat{d}_t \Leftarrow \widetilde{R}_t^{(c_{0,1})} / R_t^{(c_{0,1})}$.
  % Observe d_t and update $\hat{d}_t^{\{c_{i,j}\}}$ as in (1), $\forall c_{i,j} \in \mathcal{E}_t$.
  $L_{t+1}^{(c_{i,j})} \Leftarrow L_t^{(c_{i,j})} \exp\left(-a\, l(d_t, \hat{d}_t^{(c_{i,j})})\right)$, $\forall c_{i,j} \in \mathcal{E}_t$
  % Update $R_t^{(c_{i,j})}$ as in (8), $\forall c_{i,j} \in \mathcal{E}_t$.
end for

The numerator of (2), i.e., $\sum_{k=1}^{K_h} L_{t,k}\hat{d}_{t,k}$, can be obtained using the recursion parameter (8) to define a new intermediate parameter

$$\widetilde{R}_t^{(c_{i,j})} \triangleq L_t^{(c_{i,j})}\hat{d}_t^{(c_{i,j})} + \widetilde{R}_t^{(c_{i+1,m})} \prod_{\substack{c_{i+1,j'} \in \mathcal{D}^{(c_{i,j})} \\ c_{i+1,j'} \neq c_{i+1,m}}} R_t^{(c_{i+1,j'})},$$

where $c_{i+1,m}$ represents the descendant of the equivalence class $c_{i,j}$ containing the current pattern. Similar to (8), if we expand the recursive formulation for $\widetilde{R}_t^{(c_{0,1})}$, we get $\widetilde{R}_t^{(c_{0,1})} = \sum_{k=1}^{K_h} L_{t,k}\hat{d}_{t,k}$, which is equal to the numerator of (2). Hence, we can calculate the final output in (2) by simply $\hat{d}_t = R_t^{(c_{0,1})} / \widetilde{R}_t^{(c_{0,1})}$, where the detailed description of the algorithm can be found in Algorithm 1.

IV. Real Life Experiments

In this section, we illustrate the merits of the proposed algorithm with real life examples under the squared error loss. To this end, we consider the prediction of the energy profiles of the actual consumers. Particularly, we forecast the energy consumption of actual consumers using their past consumption patterns, where the aim is to predict the consumption trend such that $d_t = 1$ if $x_{t+1} \geq x_t$ and $d_t = -1$, otherwise, i.e., we try to forecast an increasing or decreasing trend in the energy consumption patterns. In order to capture the convergence behavior of the algorithms perfectly, we choose $h = 4$ for this real life experiment.

In Figure 3, the accumulated squared error performances (normalized with time) of the proposed algorithms are compared, where "Univ" represents the universal predictor introduced in this paper, "Fin" represents the finest predictor for $h = 4$, i.e., the predictor using all equivalence classes at level-3 (e.g., see level-2 in Fig. 2 for $h = 3$) as its states, "Coar" represents the coarsest predictor, i.e., the predictor with only one state, i.e., the one at level-0 (e.g., see level-0 in Fig. 2).

Owing to its universal formulation, the performance of the "Univ" algorithm is comparable with the "Coar" algorithm when there is not sufficient amount of data to train finer energy consumption patterns (equivalence classes). However, as the data length increases, the performance of the "Coar" algorithm deteriorates with respect to the algorithms considering finer equivalence classes such as the "Fin" algorithm. On the other hand, the performance of the "Univ" algorithm is still as well as the "Fin" algorithm even after a significant amount of observations.

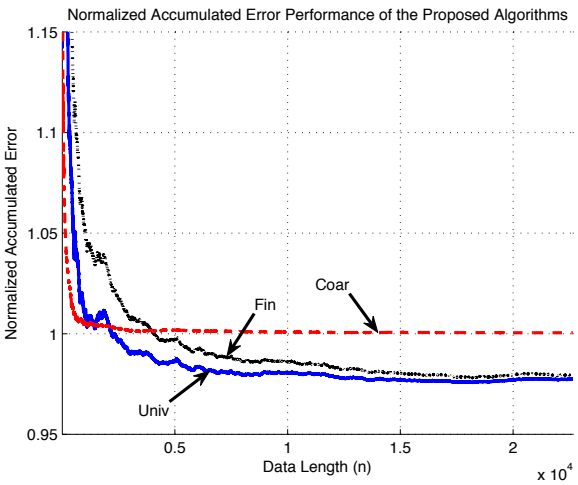

Fig. 3: Normalized accumulated squared error performance of the proposed algorithms.

We emphasize that as the pattern order h increases or when the underlying data is highly nonstationary, the convergence performance of the "Univ" algorithm will significantly outperform the performance of the "Fin" algorithm since the "Fin" algorithm may not be able to observe enough training sequences to achieve a satisfactory performance. This result is also apparent in Figure 3, where over short data sequences the performance of the "Fin" algorithm is worse compared to the "Univ" and "Coar" algorithms. Hence, the universal algorithm outperforms the constituent FS predictors by exploiting the time-dependent nature of the best choice among constituent FS predictors that are defined on the hierarchical structure.

V. Concluding Remarks

In this paper, we study sequential prediction of energy usage data of consumers, where we use the relative ordering patterns of the energy consumption history to construct states. Instead of directly using the relative ordering patterns of the energy consumption history, which can result a undesirably large number of states for even moderate length patterns, we define hierarchical equivalence classes by recursively tying certain patterns to avoid over training problems. With this equivalence class definitions, we construct a huge number of FS predictors, one of which is optimal for the underlying task. By introducing such a low complexity universal algorithm, we show that we can sequentially achieve the performance of the best sequential FS predictor out of 2^{h^h} possible FS predictors defined by this hierarchical structure, with computational complexity only linear in the length of the pattern h. Our results are generic such that they can be directly used for a wide range of hierarchical equivalence class definitions and hold for a wide range of loss functions [10]. Furthermore, we analyze the performance of our algorithm using a real life energy consumption data of the actual consumers and illustrate that the introduced algorithm can be efficiently used in the forecasting (or prediction) of energy profiling, modelling, and price management scenarios.

REFERENCES

[1] S. S. Kozat and A. C. Singer, "Universal semiconstant rebalanced portfolios," *Mathematical Finance*, vol. 21, no. 2, pp. 293–311, October 2010.

[2] N. D. Vanli and S. S. Kozat, "A comprehensive approach to universal piecewise nonlinear regression based on trees," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5471–5486, Oct 2014.

[3] D. P. Helmbold and R. E. Schapire, "Predicting nearly as well as the best pruning of a decision tree," *Machine Learning*, vol. 27, no. 1, pp. 51–68, 1997.

[4] E. Takimoto and M. K. Warmuth, "Predicting nearly as well as the best pruning of a planar decision graph," *Theoretical Computer Science*, vol. 288, no. 2, pp. 217 – 235, 2002.

[5] S. S. Kozat, A. C. Singer, and G. C. Zeitler, "Universal piecewise linear prediction via context trees," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3730–3745, 2007.

[6] M. Feder, N. Merhav, and M. Gutman, "Universal prediction of individual sequences," *IEEE Transactions on Information Theory*, vol. 38, pp. 1258–1270, 1992.

[7] L. Rabiner and R. Schafer, *Digital Processing of Speech Signals*. Englewood Cliffs, NJ: Prentice-Hall, 1978.

[8] V. Vovk, "Aggregating strategies," in *Proceedings of COLT*, 1990, pp. 371–383.

[9] ——, "Competitive on-line statistics," *International Statistical Review*, vol. 69, pp. 213–248, 2001.

[10] D. Haussler, J. Kivinen, and M. K. Warmuth, "Sequential prediction of individual sequences under general loss functions," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 1906–1925, 1998.