Universal Prediction of Individual Sequences

Siva Kumar Gorantla
IE 598 Project Report

Unlike standard statistical approaches to forecasting, prediction of individual sequences does not impose any probabilistic assumption on the data-generating mechanism. Yet, prediction algorithms can be constructed that work well for all possible sequences, in the sense that their performance is always nearly as good as the best forecasting strategy in a given reference class. In this report, the problem of predicting the next outcome of an individual binary sequence using finite memory [1], is considered.

The problem of sequential prediction, deprived of any probabilistic assumption, is deeply connected with the information-theoretic problem of compressing an individual data sequence. A pioneering research in this field was carried out by Ziv [2] and Lempel and Ziv [3], who solved the problem of compressing an individual data sequence almost as well as the best finite-state automaton. As shown by Feder, Merhav, and Gutman [1], the LempelZiv compressor can be used as a randomized forecaster [4, Chap 4] (for the absolute loss [4, Chap 8]) with a vanishing per-round regret against the class of all finite-state experts, a surprising result considering the rich structure of this class. In addition, Feder, Merhav, and Gutman devise, for the same expert class, a forecaster with a convergence rate better than the rate provable for the LempelZiv forecaster (see also [5] for further results along these lines). Another important contribution of [1] is introducing the Markov experts as a class of predictors.

We now make comparisons of the problem with other prediction problems in fields of gambling, compression and complexity.

I. Literature

Gambling,Prediction,Compression and Complexity:

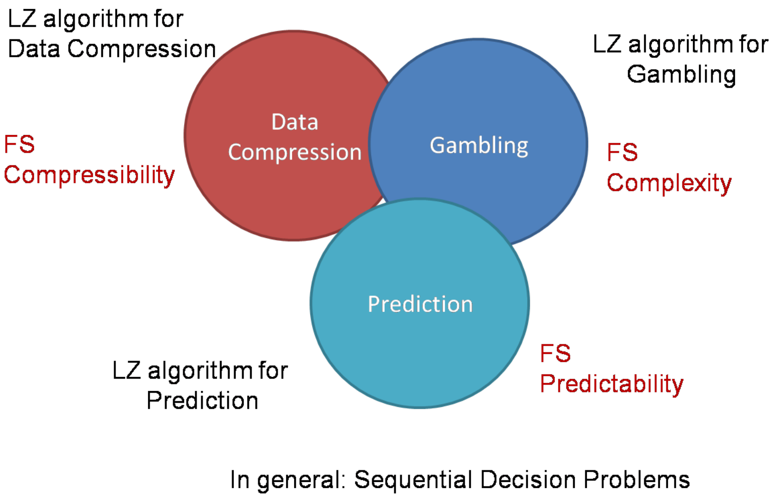

Fig. 1. Gambling Prediction and Compression

Models of prediction of individual sequences arose in disparate areas motivated by problems as different as playing repeated games(game theory), compressing data(information theory), gambling(sequential investment, mathematical finance) or information content(learning). In general, these can be generalized as a *sequential compound decision problem* rigorously studied since 1950s.

Cover, Ziv and others gave the information-theoretic foundations of sequential prediction, first motivated by applications for data compression and “universal” coding, and later extended to models of sequential gambling and investment. This theory mostly concentrates on a particular loss function, the so-called logarithmic or selfinformation loss, as it has a natural interpretation in the framework of sequential probability assignment. Sequential prediction aimed at minimizing logarithmic loss is intimately related to maximizing benefits by repeated investment in the stock market.

Connections between prediction with expert advice and information content of an individual sequence have been explored by Vovk and Watkins [6], who introduced the notion of predictive complexity of a data sequence, a quantity that, for the logarithmic loss, is related to the Kolmogorov complexity of the sequence

In the context of Feder, Merhav and Gutman’s work [1], prediction schemes are proposed based on Lempel-Ziv parsing algorithm and is analogous to the schemes in gambling [7] and data compression [3] in the context of Finite-State experts. Analogous to the Finite State (FS) compressibility defined in [3], or the FS complexity defined in [7], the FS predictability is of an infinite individual sequence is defined as the minimum asymptotic fraction of errors that can be made by any FS predictor (See Figure 1).

II. The problem setup

Imagine an observer receiving sequentially an arbitrary deterministic binary sequence $x_1, x_2, \cdots$ and wishing to predict at time t the next bit x_{t+1} based on the past $x_1, x_2, \cdots, x_t$. While only a limited amount of information from the past can be memorized by the observer, it is desired to keep the relative frequency of prediction errors as small as possible in the long run.

It might seem surprising, at first glance, that the past can be useful in predicting the future because when a sequence is arbitrary, the future is not necessarily related to the past. Nonetheless, it turns out that sequential (randomized) prediction schemes exist that utilize the past, whenever helpful in predicting the future, as well as any finite-state (FS) predictor.

A. Finite State Predictor

The prediction rule $f(.)$ of an FS predictor is defined by

$$\hat{x}_{t+1} = f(s_t)$$

where s_t is the current state which takes on values in a finite set $\mathcal{S} = \{1, 2, \cdots, S\}$. The state s_t is updated according to the next-state rule $g(\cdot, \cdot)$.

$$s_{t+1} = g(x_t, s_t)$$

Thus the finite-state machine (FSM) can be represented by

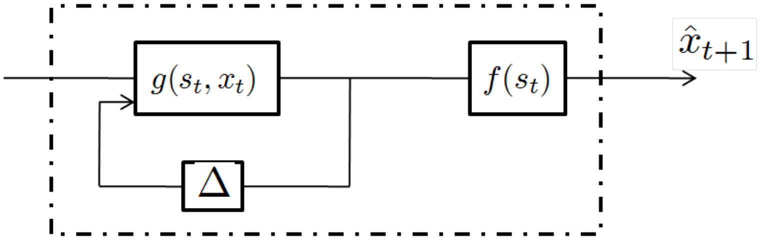

Fig. 2. Finite State Machine

Suppose a finite sequence $x_1^n = x_1, \cdots, x_n$ is known along with the initial state s_1, and the next-state function g (and hence the state sequence) are provided. In this case, as discussed in [7], the best prediction rule for the sequence x_1^n is deterministic and given by

$$\hat{x}_{t+1} = f(s_t) = \begin{cases} 0 \text{ if } N_n(s_t, 0) > N_n(s_t, 1) \\ 1 \text{ otherwise} \end{cases} \tag{1}$$

where $N_n(s, x)$ is the joint count of $s_t = s$ and $x_{t+1} = x$ along the sequence x_1^n. Note that this optimal rule depends on the entire sequence and hence cannot be determined sequentially.

The minimum fraction of prediction errors is

$$\pi(g; x_1^n) = \frac{1}{n} \sum_{s=1}^{S} \min\{N_n(s, 0), N_n(s, 1)\}$$

The minimum fraction of prediction errors with respect to all FSM's with S states is called the *S-state predictability*

$$\pi_S(x_1^n) = \min_{g \in G_s} \pi(g; x_1^n)$$

The FS predictability is defined as

$$\pi(\mathbf{x}) = \lim_{S \to \infty} \limsup_{n \to \infty} \pi_S(x_1^n)$$

The finite-state predictability of an infinite sequence is defined as the minimum fraction of prediction errors that can be made by any finite-state (FS) predictor.

III. LZ incremental parsing algorithm

In this section, the scheme based on Limpel-Ziv parsing algorithm is shown which achieves the FS predictability of a sequence.The underlying idea is that the incremental parsing algorithm induces another technique for gradually changing the Markov order with time at an appropriate rate (increasing the size of the state space $\mathcal{S}$ with time).The LZ forecaster has the following key properties:

- Markov: Remembers the last few entries
- Incremental: State size increases with n of the order of $\log n$.

First, parse a sequence into distinct phrases s.t each phrase is the shortest string which is not a previously parsed phrase.

```
                                  A, B,  C, D,  E
00101010100.....  ------------>  {X,0,01,010,1,0100,......}
```

Grow a tree s.t. each new phrase is represented by a leaf in the tree as shown in fig 3. Let K_j denotes the number of leaves in the j-th step (Here $K_j = 7$) and assign a weight $\frac{1}{K_j}$ to each leaf. This can be thought of as assigning a uniform probability mass function to the leaves. The weight of each internal node is the sum of weights of its two offsprings. The conditional probability $\hat{p}_t^{LZ}(x_{t+1}|x_1^t)$ of a symbol x_{t+1} given its past as the ratio between the weight of the node corresponding to x_{t+1} that follows the current node x_t, and the weight of the node associated with x_t.

The estimator for the sequential prediction is given as

$$\hat{x}_{t+1} = \begin{cases} \text{“}0'', \text{ with probability } \phi_t(\hat{p}_t^{LZ}(0|x_1^t)), \\ \text{“}1'', \text{ with probability } \phi_t(\hat{p}_t^{LZ}(1|x_1^t)), \end{cases} \tag{2}$$

where $\phi_t(.)$ is defined as in (3).

$$\phi(\alpha) = \begin{cases} 0, 0 \leq \alpha < \frac{1}{2} - \epsilon, \\ \frac{1}{2\epsilon}[\alpha - \frac{1}{2}] + \frac{1}{2}, \frac{1}{2} - \epsilon \leq \alpha \leq \frac{1}{2} + \epsilon, \\ 1, \frac{1}{2} + \epsilon \leq \alpha < 1, \end{cases} \tag{3}$$

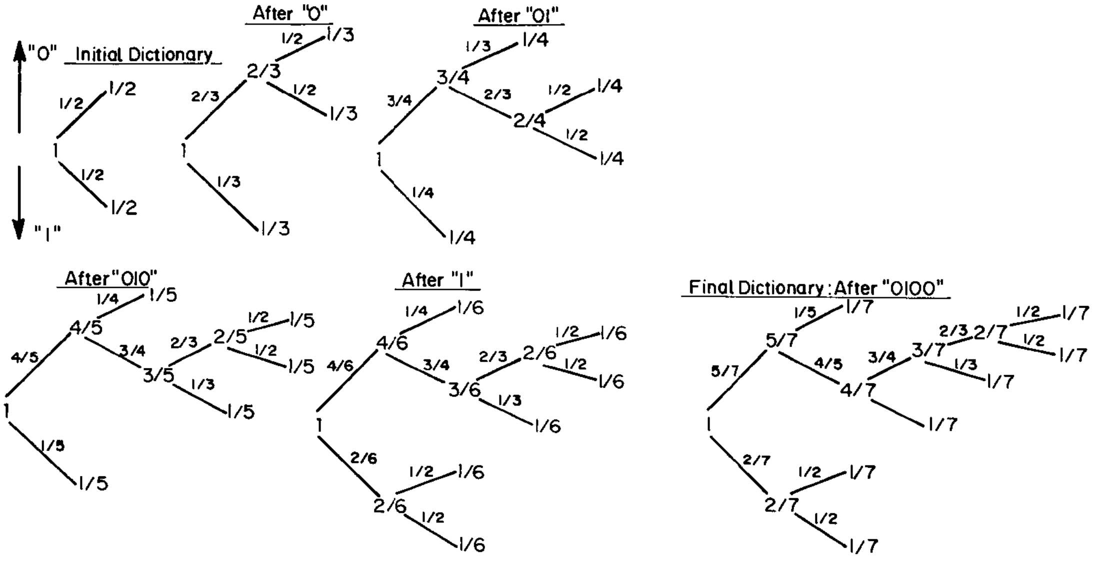

Fig. 3. Dictionary trees and probability estimate induced by the LZ scheme

Thus the probability of making an error is $1 - \phi(\hat{p}_t(x_{t+1}))$ and fraction of errors for the sequence x_1^n is:

$$\begin{aligned} \hat{\pi}(x_1^n) &= \frac{1}{n}\sum_{t=1}^{n}(1 - \phi(\hat{p}_t(x_{t+1}))) & (4) \\ &\to \pi(\mathbf{x}) & (5) \end{aligned}$$

The last step that the fraction of errors converge to the FS predictability is achieved when $\epsilon_t = \epsilon = \frac{1}{2\sqrt{t+2}}$ is varying with time.

A. LZ algorithm for gambling

The problem is similar - At each time step, either Horse 0 or Horse 1 wins. You get either double or nothing. *How do you invest on the horses taking into consideration revious winning patterns?*. The LZ incremental parsing algorithm can be applied for this problem and it is proven in [7] that it achieves the FS compressibility of the winning sequence. The crucial difference between LZ algorithm for gambling and prediction is given as follows:

- For prediction: $\hat{p}_t^{LZ}(x_{t+1}|x_1^t)$ is used to predict x_{t+1} according to (2).
- For Gambling: The conditional probabilities are used as follows: Invest $\hat{p}_t^{LZ}(0)$ on Horse 0 and invest $\hat{p}_t^{LZ}(1)$ on Horse 1.

References

[1] M. Feder, “Universal Prediction of Individual Sequences,” *IEEE Transactions on Information Theory*, vol. 38, no. 4, pp. 1258–1270, 1992.

[2] J. Ziv, “Coding theorems for individual sequences,” *IEEE Transactions on Information Theory*, vol. 24, pp. 405–412, 1978.

[3] J. Ziv and A. Lempel, “A universal algorithm for sequential data-compression,” *IEEE Transactions on Information Theory*, vol. 23, pp. 337–343, 1977.

[4] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning and games*, 1st ed. Cambridge, 2006.

[5] M. Feder and N. Merhav, “Universal schemes for sequential decision from individual data sequences,” *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1280–1292, 1993.

[6] V. Vovk and C. Watkins, “Universal portfolio selection,” *ACM Computaitional Learning Theory*, pp. 12–23, 1998.

[7] M. Feder, “Gambling using a finite-state machine,” *IEEE Transactions on Information Theory*, vol. 37, pp. 1459–1465, 1991.