Universal Prediction of Individual Sequences

Siva K Gorantla
IE598 Class Presentation
Outline

• Problem Setup
• Algorithm
• Algo for Gambling
• Proofs (converse)
• Related Work
• Future Directions?
Binary Output Sequence

\[ x_1, x_2, \cdots, x_t \]

\[ \hat{x}_{t+1} \]

At time t:
Binary Output Sequence

\[ x_1, x_2, \cdots, x_t, x_{t+1}, x_{t+2}, \ldots \ldots. \]

\[ \hat{x}_{t+1}, \hat{x}_{t+2}, \ldots \ldots. \]

At time \( t \):

At time \( t+1 \):
Binary Output Sequence

\[ x = x_1, x_2, \cdots, x_t, x_{t+1}, x_{t+2}, \ldots \ldots \ldots \text{ Infinite binary sequence} \]

\[ \hat{x}_{t+1}, \hat{x}_{t+2}, \ldots \ldots \ldots \]

At time t:

At time t+1:

**Objective:** Minimize the relative frequency of prediction errors.
Binary Output Sequence

\[ x = x_1, x_2, \cdots, x_t, x_{t+1}, x_{t+2}, \cdots \cdots \] Infinite binary sequence

\[ \hat{x}_{t+1}, \hat{x}_{t+2}, \cdots \cdots \]

At time t:
At time t+1:

Objective: Minimize the relative frequency of prediction errors.

• i.i.d., then Past \( \not\Rightarrow \) Future.

• Predictors helpful whenever Past helps in predicting the future(Patterns).
Finite State (FS) Predictor

\[ x = x_1, x_2, \cdots \]

Inefficient/Infeasible to remember the entire sequence \((x_1, \cdots, x_t)\) – Instead remember ‘state’ of the sequence \((s_t)\)
Finite State(FS) Predictor

\[ \mathbf{x} = x_1, x_2, \cdots \]

Inefficient/Infeasible to remember the entire sequence \((x_1, \cdots, x_t)\) –
Instead remember ‘state’ of the sequence \((s_t)\)

Predictor Rule:

\[ \hat{x}_{t+1} = f(s_t) \quad s_t \in S = \{1, 2, \cdots, S\} \]

Next State Rule:

\[ s_{t+1} = g(s_t, x_t) \]

Finite State Predictor:
Finite State (FS) Predictor

\( x = x_1, x_2, \cdots \)

Inefficient/Infeasible to remember the entire sequence \((x_1, \cdots, x_t)\) –
Instead remember ‘state’ of the sequence \((s_t)\)

Predictor Rule:

\[
\hat{x}_{t+1} = f(s_t) \quad \text{subject to} \quad s_t \in S = \{1, 2, \cdots, S\}
\]

Next State Rule:

\[
s_{t+1} = g(s_t, x_t)
\]

Finite State Predictor:

\[
\hat{x}_{t+1} \sim p(x_{t+1}|s_t)
\]

f can be stochastic
Literature

**Best fixed Predictor:** (single-state) => Not saving any patterns

- Suppose frequency of zeros and ones are known e.g: 0.7 and 0.3
  - Best strategy = **fixed strategy**: predict either “0” or “1” all the time.
  - error = 0.3
Best fixed Predictor: (single-state) => Not saving any patterns

- Suppose frequency of zeros and ones are known e.g: 0.7 and 0.3
  - Best strategy = fixed strategy: predict either “0” or “1” all the time.
  - error = 0.3

- Suppose no information is known about the sequence.

  “Behavior of sequential predictors of binary sequences” – Tom Cover

Universal Predictor with same performance as fixed strategy.

  error -> 0.3
Markov Predictor: \[ S_t = (x_{t-k}, \ldots, x_{t-1}) \]

- Suppose prior information is known – frequency of #(s,0) and #(s,1).
  - Best Markov Predictor.
  - error = \( \pi^{MP} \)
Markov Predictor: \[ s_t = (x_{t-k}, \ldots, x_{t-1}) \]

- Suppose prior information is known – frequency of #(s,0) and #(s,1).
  - Best Markov Predictor.
  - error = \( \pi^{MP} \)

- Suppose no information is known about the sequence.

"Compound Bayes predictors for sequences with apparent Markov Structure" – Tom Cover

Universal Predictor with same performance as Best Markov predictor.

error \( \rightarrow \pi^{MP} \)
Fixed, Markov $\rightarrow$ Finite State

Finite State Predictor: $s_t \in \{1, 2, \cdots, S\}$

- Suppose prior information is known – frequency of #(s,0) and #(s,1).
  - Best FS Predictor.
  - error = $\pi^{FS}$
  -
Fixed, Markov $\rightarrow$ Finite State

**Finite State Predictor:** $s_t \in \{1, 2, \cdots, S\}$

- Suppose prior information is known – frequency of $(s,0)$ and $(s,1)$.
  - Best FS Predictor.
  - $\text{error} = \pi^{FS}$
- Suppose no information is known about the sequence.

Does there exist an Universal Predictor with same performance as Best Finite State Predictor?

$\text{error} \rightarrow \pi^{FS}$?
Fixed, Markov $\rightarrow$ Finite State

**Finite State Predictor:** $s_t \in \{1, 2, \cdots, S\}$

- Suppose prior information is known – frequency of $(s,0)$ and $(s,1)$.
  - Best FS Predictor.
  - $\text{error} = \pi^{FS}$
- Suppose no information is known about the sequence.

Does there exist an Universal Predictor with same performance as Best Finite State Predictor?

$$\text{error} \rightarrow \pi^{FS} ?$$

1. $\exists$ Markov Predictor $\approx \pi^{FS}$
2. Markov Predictor + increasing $k \rightarrow \pi^{FS}$
3. Lempel-Ziv Parsing Algorithm: Markov Predictor with time varying order
Scope of the technique:

LZ algorithm for Data Compression

Data Compression

Gambling

Prediction

LZ algorithm for Gambling

LZ algorithm for Prediction

In general: Sequential Decision Problems
Predictability
- Min fraction of prediction errors possible

Fix a finite sequence: \( x_1, \ldots, x_n \)
Fix \( s_1, g : s_1, \ldots, s_n \)
Predictability

- Min fraction of prediction errors possible

Fix a finite sequence: \( x_1, \cdots, x_n \)
Fix \( s_1, g : s_1, \cdots, s_n \)

Compute:

<table>
<thead>
<tr>
<th>( N_n(s,0) )</th>
<th>( N_n(s,1) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_n(1,0) )</td>
<td>( N_n(1,1) )</td>
</tr>
<tr>
<td>( N_n(2,0) )</td>
<td>( N_n(2,1) )</td>
</tr>
<tr>
<td>( N_n(S,0) )</td>
<td>( N_n(S,1) )</td>
</tr>
</tbody>
</table>
Predictability
- Min fraction of prediction errors possible

Fix a finite sequence: \( x_1, \cdots, x_n \)
Fix \( s_1, g \):
\( s_1, \cdots, s_n \)
Compute:

<table>
<thead>
<tr>
<th>( N_n(s,0) )</th>
<th>( N_n(s,1) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_n(1,0) )</td>
<td>( N_n(1,1) )</td>
</tr>
<tr>
<td>( N_n(2,0) )</td>
<td>( N_n(2,1) )</td>
</tr>
<tr>
<td>( N_n(S,0) )</td>
<td>( N_n(S,1) )</td>
</tr>
</tbody>
</table>

• Best prediction rule:
\[
\hat{x}_{t+1} = f(s_t) = \begin{cases} 
0 & \text{if } N_n(s_t, 0) > N_n(s_t, 1) \\
1 & \text{otherwise}
\end{cases}
\]
Predictability
- Min fraction of prediction errors possible

\[ x_1, \ldots, x_n \]
\[ g(s_t, x_t) \]
\[ s_t \]
\[ f(s_t) \]
\[ \hat{x}_{t+1} \]

Fix a finite sequence: \( x_1, \ldots, x_n \)
Fix s_1,g : \( s_1, \ldots, s_n \)
Compute:

\[ N_n(s,0) \quad N_n(s,1) \]
\[ N_n(1,0) \quad N_n(1,1) \]
\[ N_n(2,0) \quad N_n(2,1) \]
\[ \ldots \]
\[ N_n(S,0) \quad N_n(S,1) \]

- Best prediction rule:
\[ \hat{x}_{t+1} = f(s_t) = \begin{cases} 0 & \text{if } N_n(s_t, 0) > N_n(s_t, 1) \\ 1 & \text{otherwise} \end{cases} \]

- Minimum Fraction of Prediction errors:
\[ \pi(g; x_1^n) = \frac{1}{n} \sum_{i=1}^{S} \min\{N_n(s, 0), N_n(s, 1)\} \in [0, \frac{1}{2}] \]
Predicatability - 2

$$\pi(g; x_1^n) \rightarrow \text{Fix } x_1^n, S, g$$
Predicatability - 2

\[ \pi(g; x^n_1) \rightarrow \text{Fix } x^n_1, S, g \]

- S-state predictability of \( x^n_1 \)

\[ \pi_S(x^n_1) = \min_{g \in G_s} \pi(g; x^n_1) \rightarrow \text{Fix } x^n_1, S \]
Predicatability - 2

\[ \pi(g; x_1^n) \rightarrow \text{Fix } x_1^n, S, g \]

- S-state predictability of \( x_1^n \)

\[ \pi_S(x_1^n) = \min_{g \in G_s} \pi(g; x_1^n) \rightarrow \text{Fix } x_1^n, S \]

- asymptotic S-state predictability

\[ \pi_S(x) = \limsup_{n \to \infty} \pi_S(x_1^n) \rightarrow \text{Fix } x, S \]
Predicatability - 2

\[ \pi(g; x^n_1) \quad \longrightarrow \quad \text{Fix } x^n_1, S, g \]

- S-state predictability of \( x^n_1 \)

\[ \pi_S(x^n_1) = \min_{g \in G_s} \pi(g; x^n_1) \quad \longrightarrow \quad \text{Fix } x^n_1, S \]

- asymptotic S-state predictability

\[ \pi_S(x) = \limsup_{n \to \infty} \pi_S(x^n_1) \quad \longrightarrow \quad \text{Fix } x, S \]

- FS predictability

\[ \pi(x) = \lim_{S \to \infty} \pi_S(x) \quad \longrightarrow \quad \text{Fix } x \]
Predicatability - 2

\[ \pi(g; x^n_1) \rightarrow \text{Fix } x^n_1, S, g \]

- S-state predictability of \( x^n_1 \)
  \[ \pi_S(x^n_1) = \min_{g \in G_s} \pi(g; x^n_1) \rightarrow \text{Fix } x^n_1, S \]

- Asymptotic S-state predictability
  \[ \pi_S(x) = \limsup_{n \to \infty} \pi_S(x^n_1) \rightarrow \text{Fix } x, S \]

- FS predictability
  \[ \pi(x) = \lim_{S \to \infty} \pi_S(x) \rightarrow \text{Fix } x \]

Note: Attained by FSM that depend on particular sequence \( x \)

We want sequential prediction scheme which work independent of \( x \)
and yet achieve \( \pi(x) \)
Propose a scheme

\[ \pi(g; x^n_1) \leftarrow \hat{\pi}(g; x^n_1) \]

\[ \pi_S(x^n_1) \leftarrow \hat{\pi}_S(x^n_1) \]

\[ \pi_S(x) \leftarrow \hat{\pi}_S(x) \]

\[ \pi(x) \leftarrow \hat{\pi}(x) \]
LZ incremental parsing algo

• Parse a sequence into distinct phrases s.t each phrase is the shortest string which is not a previously parsed phrase.

\[001010101000\ldots\] \[\rightarrow\] \{X,0,01,010,1,0100,\ldots\}
LZ incremental parsing algo

- 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,……}

- Growing a tree s.t. each new phrase is represented by a leaf in the tree.
LZ incremental parsing algo

• 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,……\}

• Growing a tree s.t. each new phrase is represented by a leaf in the tree.
LZ incremental parsing algo

• Parse a sequence into distinct phrases s.t each phrase is the shortest string which is not a previously parsed phrase.

\[\text{A, B, C, D, E} \]
\[00101010100\ldots\] \[\longrightarrow\] \[\{X, 0, 01, 010, 1, 0100, \ldots\}\]

• Growing a tree s.t. each new phrase is represented by a leaf in the tree.
LZ incremental parsing algo

- Parse a sequence into distinct phrases s.t. each phrase is the shortest string which is not a previously parsed phrase.

\[ \text{A, B, C, D, E} \]
\[ \text{00101010100….} \quad \longrightarrow \quad \{\text{X,0,01,010,1,0100,…..}\} \]

- Growing a tree s.t. each new phrase is represented by a leaf in the tree.
LZ incremental parsing algo

• 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,……}

• Growing a tree s.t. each new phrase is represented by a leaf in the tree.
LZ incremental parsing algo

• 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,......\}

• Growing a tree s.t. each new phrase is represented by a leaf in the tree.
LZ incremental parsing algo

- Parse a sequence into distinct phrases s.t. each phrase is the shortest string which is not a previously parsed phrase.

\[
\text{A, B, C, D, E} \quad \text{00101010100…..} \quad \rightarrow \quad \{X,0,01,010,1,0100,\ldots\}
\]

- Growing a tree s.t. each new phrase is represented by a leaf in the tree.
LZ incremental parsing algo

• Parse a sequence into distinct phrases s.t each phrase is the shortest string which is not a previously parsed phrase.

\[00101010100\ldots\quad \longrightarrow \quad \{X,0,01,010,1,0100,\ldots\}\]

• Growing a tree s.t. each new phrase is represented by a leaf in the tree.

What are these probabilities?
LZ incremental parsing algo

• Parse a sequence into distinct phrases s.t each phrase is the shortest string which is not a previously parsed phrase.

\[00101010100 \ldots \rightarrow \{X, 0, 01, 010, 1, 0100, \ldots\}\]

• Growing a tree s.t. each new phrase is represented by a leaf in the tree.

What are these probabilities?
Conditional probabilities of \(x_{t+1}\)

\[\hat{p}^{LZ}(x_{t+1}|x_t)\]
LZ incremental parsing algo-2

Let $c = c(x_1^n)$ be the number of parsed strings in $x_1^n$.

Let $N^j_t(x), j = 1, \cdots, c$ be the number of symbols equal to $x$ in the $j$th bin at time $t$.

The probability estimate of the next bit being $x$ entering $j$-th bin is

$$\hat{p}_x = \frac{N^j_t(x)+1}{N^j_t+2}$$
LZ incremental parsing algo-3

• Compute $\hat{p}_0, \hat{p}_1$ say 3/5, 2/5.

• Choose the one which is $>1/2$. Here $\hat{p}_0$

• If in addition, $\hat{p}_x \geq \frac{1}{2} + \epsilon$, declare $\hat{x}_{t+1} = x$.

   If $\hat{p}_x \leq \frac{1}{2} + \epsilon$, pick 0 or 1 randomly.

\[
\hat{x}_{t+1} = \begin{cases} 
0, & \text{with probability } \phi(\hat{p}_t(0)) \\
1, & \text{with probability } \phi(\hat{p}_t(1)) = 1 - \phi(\hat{p}_t(0))
\end{cases}
\]

\[
\phi(\alpha) = \begin{cases} 
0 & 0 \leq \alpha \leq \frac{1}{2} - \epsilon \\
\frac{1}{2\epsilon} \left[ \alpha - \frac{1}{2} \right] + \frac{1}{2} & \frac{1}{2} - \epsilon \leq \alpha \leq \frac{1}{2} + \epsilon \\
1 & \frac{1}{2} + \epsilon \leq \alpha \leq 1
\end{cases}
\]
LZ incremental parsing algo-3

- Compute $\hat{p}_0$, $\hat{p}_1$ say 3/5, 2/5.
- Choose the one which is >1/2. Here $\hat{p}_0$
- If in addition, $\hat{p}_x \geq \frac{1}{2} + \epsilon$, declare $\hat{x}_{t+1} = x$.
  If $\hat{p}_x \leq \frac{1}{2} + \epsilon$, pick 0 or 1 randomly.

$$\hat{x}_{t+1} = \begin{cases} 0, & \text{with probability } \phi(\hat{p}_t(0)) \\ 1, & \text{with probability } \phi(\hat{p}_t(1)) \end{cases}$$

$$\phi(\alpha) = \begin{cases} 0 & 0 \leq \alpha \leq \frac{1}{2} - \epsilon \\ \frac{1}{2\epsilon} \left[ \alpha - \frac{1}{2} \right] + \frac{1}{2} & \frac{1}{2} - \epsilon \leq \alpha \leq \frac{1}{2} + \epsilon \\ 1 & \frac{1}{2} + \epsilon \leq \alpha \leq 1 \end{cases}$$
LZ incremental parsing algo-3

\[ \hat{x}_{t+1} = \begin{cases} 
0, & \text{with probability } \phi(\hat{p}_t(0)) \\
1, & \text{with probability } \phi(\hat{p}_t(1)) 
\end{cases} \]

Probability of making an error: \( 1 - \phi(\hat{p}_t(x_{t+1})) \)

\[ \hat{\pi}(x_1^n) = \frac{1}{n} \sum_{i=0}^{n} (1 - \phi(\hat{p}_t(x_{t+1}))) \]
LZ incremental parsing algo-3

\[
\hat{x}_{t+1} = \begin{cases} 
0, \text{ with probability } \phi(\hat{p}_t(0)) \\
1, \text{ with probability } \phi(\hat{p}_t(1)) 
\end{cases}
\]

Probability of making an error: \[1 - \phi(\hat{p}_t(x_{t+1}))\]

\[
\hat{\pi}(x^n_1) = \frac{1}{n} \sum_{i=0}^{n} (1 - \phi(\hat{p}_t(x_{t+1})))
\]

\[
\hat{\pi}(x^n_1) \to \pi(x)
\]
LZ incremental parsing algo-3

\[ \hat{x}_{t+1} = \begin{cases} 
0, \text{ with probability } \phi(\hat{p}_t(0)) \\
1, \text{ with probability } \phi(\hat{p}_t(1)) 
\end{cases} \]

Probability of making an error: \[ 1 - \phi(\hat{p}_t(x_{t+1})) \]

\[ \hat{\pi}(x^n_1) = \frac{1}{n} \sum_{i=0}^{n} (1 - \phi(\hat{p}_t(x_{t+1}))) \]

\[ \hat{\pi}(x^n_1) \rightarrow \pi(x) \]

A, B, C, D, E

0010101010100..... \rightarrow \{X,0,01,010,1,0100,,......\}

As the number ‘n’ increases, the number of states ‘S’ increases.

**LZ incremental parsing algorithm.**

- Markov: Remembers last few entries.
- Incremental: States increase with n.
LZ algorithm for Gambling

At each step, either Horse 0 or Horse 1 wins.
You get double or nothing.
How do you invest taking into consideration previous winning patterns?

What are these probabilities?

Conditional probabilities of $x_{t+1}$

$\hat{p}_{LZ}^{LZ}(x_{t+1}|x^t_1)$
LZ algorithm for Gambling

What are these probabilities?
Conditional probabilities of $x_{t+1}$

\[ \hat{p}^{LZ}(x_{t+1}|x_t^t) \]

- At each step, either Horse 0 or Horse 1 wins.
- You get double or nothing.
- How do you invest taking into consideration previous winning patterns?

\[ \hat{p}^{LZ}(x_{t+1}|x_t^t) \rightarrow (\text{Prediction}) \text{ Use to predict } \hat{x}_{t+1} \]
\[ \rightarrow (\text{Gambling}) \text{ Invest } \hat{p}_0 \text{ on Horse 0 and } \hat{p}_1 \text{ on Horse 1.} \]
Gambling Using a Finite State Machine

- Finite State Complexity:

\[ S_n = S_0 2^n (1 - H^{FS}(x^n_1)) \]

- Using LZ algorithm for Gambling:

\[ S_n = S_0 2^n (1 - \hat{H}^{LZ}(x^n_1)) \]

\[ \hat{H}^{LZ} \rightarrow H^{FS} \quad \text{[Meir Feder '91]} \]
Scope of the technique:

- LZ algorithm for Data Compression
- LZ algorithm for Gambling
- FS Compressibility
- FS Complexity
- LZ algorithm for Prediction
- FS Predictability

In general: Sequential Decision Problems
Proofs

• **S = 1**  Single-State Machine

Fix a finite sequence: \( x_1, \ldots, x_n \)

If \( N_n(1, 0), N_n(1, 1) \) are known, optimal solution:

\[
\hat{x}_{t+1} = \begin{cases} 
0, & \text{if } N_n(1, 0) > N_n(1, 1) \\
1, & \text{otherwise}
\end{cases}
\]

\[
\pi_1(x_1^n) = \frac{1}{n} \min\{N_n(1, 0), N_n(1, 1)\}
\]

Non-Sequential
Proof – Step 1

- **S = 1  Single-State Machine**

  Fix a finite sequence:  \( x_1, \ldots, x_n \)

  If \( N_n(1, 0), N_n(1, 1) \) are not known:

  At each \( t \), update \( N_t(1,0) \) and \( N_t(1,1) \), compute

  \[
  \hat{p}_x = \frac{N_t(s,x) + 1}{t+2}
  \]

  \[
  \hat{x}_{t+1} = \begin{cases} 
  0, & \text{with probability } \phi(\hat{p}_t(0)) \\
  1, & \text{with probability } \phi(\hat{p}_t(1)) 
  \end{cases}
  \]

  \[
  \hat{\pi}_1(x^n_1) \to \pi_1(x^n_1), \quad \forall x^n_1
  \]
Proof – Step 1

• $S = 1$ Single-State Machine

Assume $N_n(1,0) > N_n(1,1)$ WLOG

$$\pi(x^n_1) = \frac{1}{n}N_n(1,1)$$

Predicts “0” every time.

$$\hat{\pi}(x^n_1) \leq \hat{\pi}(\tilde{x}^n_1)$$

Worst sequence

$$\hat{\pi}(\tilde{x}^n_1) = E[\text{fraction of errors}]$$

----- as a function of $\epsilon$

$$\hat{\pi}(\tilde{x}^n_1) = \frac{1}{n} \sum_{i=0}^{n} (1 - \phi(\hat{p}_t(x_{t+1})))$$

$$\leq \frac{N_n(1,1)}{n} + \frac{\epsilon}{1-2\epsilon} + O\left(\frac{\log n}{n}\right)$$

$\epsilon$ fixed

$$\leq \frac{N_n(1,1)}{n} + O\left(\frac{1}{\sqrt{n}}\right)$$

$\epsilon_t = \epsilon = \frac{1}{2\sqrt{t+2}}$
Proofs

\[ \hat{\pi}(x_1^n) \leq \frac{N_n(1,1)}{n} + O\left(\frac{1}{\sqrt{n}}\right) \]

Proposed a Scheme
Compute worst Case performance

Propose a scheme

\[ \pi(g; x_1^n) \quad \hat{\pi}(g; x_1^n) \]

\[ \pi_S(x_1^n) \quad \hat{\pi}_S(x_1^n) \]

\[ \pi_S(x) \quad \hat{\pi}_S(x) \]

\[ \pi(x) \quad \hat{\pi}(x) \]
Proof-Step 2

• S known, g known

Fix a finite sequence: \(x_1, \ldots, x_n\)

\[
\hat{p}_t(x|s) = \frac{N_t(s,x)+1}{N_t(s)+2}, \quad x = 0, 1
\]

\[
\hat{x}_{t+1} = f(s_t) = \begin{cases} 
0, \text{ with probability } \phi(\hat{p}_t(0|s_t)) \\
1, \text{ with probability } \phi(\hat{p}_t(1|s_t))
\end{cases}
\]

Decompose \(x_1^n\) into S subsequences \(x^n(S)\) of length \(N_n(s)\)

\[
\hat{\pi}(g; x_1^n) \leq \frac{1}{n} \sum_{i=1}^{S} \left[ \min\{N_n(s,0), N_n(s,1)\} + N_n(s)\delta_1(N_n(s)) \right] \\
\leq \pi(g; x_1^n) + O(\sqrt{S/n})
\]
Proofs

\[ \hat{\pi}(x^n_1) \leq \frac{N_{n(1,1)}}{n} + O\left(\frac{1}{\sqrt{n}}\right) \]

Proposed a Scheme

Compute worst Case performance

\[
O(\sqrt{S/n}) \quad \begin{array}{c}
\pi(g; x^n_1) \\
\pi_S(x^n_1) \\
\pi_S(x) \\
\pi(x)
\end{array}
\rightarrow
\begin{array}{c}
\hat{\pi}(g; x^n_1) \\
\hat{\pi}_S(x^n_1) \\
\hat{\pi}_S(x) \\
\hat{\pi}(x)
\end{array}
\]
Refinement of an FS machine

\[ g \rightarrow \tilde{g} \quad \text{s.t.} \quad s_t = h(\tilde{s}_t) \]

A refinement can do better than the original.

\[ \pi(g; x^n_1) \geq \pi(\tilde{g}; x^n_1) \]

For a given S, over all \( g \in G_S \)

\[ |G| = S^{2S} \]

\[ \tilde{s}_t = (s^1_t, s^2_t, \ldots, s^M_t) \]

\[ \pi(\tilde{g}; x^n_1) \leq \pi(g; x^n_1) \quad \forall g \in G_S \]

\[ \pi(\tilde{g}; x^n_1) \leq \min_{g \in G_S} \pi(g; x^n_1) = \pi_S(x^n_1) \leq O(\sqrt{S^{2S}/n}) \]
Proofs

\[ \hat{\pi}(x_1^n) \leq \frac{N_n(1,1)}{n} + O\left(\frac{1}{\sqrt{n}}\right) \]

Proposed a Scheme

Compute worst Case performance

\[ O(\sqrt{S/n}) \]
\[ \pi(g; x_1^n) \quad \leftrightarrow \quad \hat{\pi}(g; x_1^n) \]

\[ O(\sqrt{S^2S/n}) \]
\[ \pi_S(x_1^n) \quad \leftrightarrow \quad \hat{\pi}_S(x_1^n) \quad \text{S-State predictability} \]

Define a new refined state

\[ \pi_S(x) \quad \leftrightarrow \quad \hat{\pi}_S(x) \]

\[ \pi(x) \quad \leftrightarrow \quad \hat{\pi}(x) \]
Markov Predictors

\[ s_t = (x_{t-k}, \ldots, x_{t-1}) \]

Let \( \mu_k(x) \) be the k-th order Markov predictability

Refinement: \( k^* > k \Rightarrow \mu_k(x) > \mu_{k^*}(x) \)

Scheme:

\[ \hat{x}_{t+1} = f(s_t) = \begin{cases} 
0, \text{ with probability } \phi(\hat{p}_t(0|(x_{t-k}, \ldots, x_{t-1}))) \\
1, \text{ with probability } \phi(\hat{p}_t(1|(x_{t-k}, \ldots, x_{t-1}))) 
\end{cases} \]

\[ \hat{p}_x = \frac{N_t(x_{t-k+1} \ldots x_{t-1} \cdot x_0) + 1}{N_t(x_{t-k+1} \ldots x_{t}) + 2} \]

\[ \hat{\mu}_k(x_1^n) \leq \mu_k(x_1^n) + O(\sqrt{\frac{2^k}{n}}) \]
Markov Predictors

\[ \hat{\mu}_k(x^n_1) \leq \mu_k(x^n_1) + O(\sqrt{2^k/n}) \]

Refinement: \( k^* > k \Rightarrow \mu_k(x) > \mu_{k^*}(x) \)

\[ \lim_{k \to \infty} \mu_k(x) = \mu(x) \quad \text{Markov Predictability} \]
Markov Predictors

\[ \hat{\mu}_k(x^n_1) \leq \mu_k(x^n_1) + O(\sqrt{2^k/n}) \]

Refinement: \( k^* > k \Rightarrow \mu_k(x) > \mu_{k^*}(x) \)

\[ \lim_{k \to \infty} \mu_k(x) = \mu(x) \quad \text{Markov Predictability} \]

To attain \( \mu(x) \), the order \( k \) must grow as more data is available
Markov Predictors

\[ \hat{\mu}_k(x^n) \leq \mu_k(x^n) + O(\sqrt{2^k/n}) \]

Refinement: \( k^* > k \) \( \Rightarrow \) \( \mu_k(x) > \mu_{k^*}(x) \)

\[ \lim_{k \to \infty} \mu_k(x) = \mu(x) \quad \text{Markov Predictability} \]

To attain \( \mu(x) \), the order \( k \) must grow as more data is available

Increase **rapidly** to achieve higher-order Markov Predictability

Increase **slowly** to ensure reliable estimate of \( \hat{p}_t(0|(x_{t-k}, \cdots, x_{t-1})) \)

Order \( k \) should not grow faster than \( O(\log t) \) to satisfy both requirements
Markov Predictors

\[ \hat{\mu}_k(x_1^n) \leq \mu_k(x_1^n) + O(\sqrt{2^k/n}) \]
\[ \rightarrow \mu(x) \]
Markov Predictors

\[ \hat{\mu}_k(x_1^n) \leq \mu_k(x_1^n) + O(\sqrt{2^k/n}) \]

\[ \rightarrow \mu(x) \]

\[ \rightarrow \pi(x)? \]
Markov Predictors

\[ \hat{\mu}_k(x^n_1) \leq \mu_k(x^n_1) + O(\sqrt{2^k/n}) \]

\[ \rightarrow \mu(x) \]

\[ \rightarrow \pi(x) \]

\[ \mu(x) \geq \pi(x) \]

\[ \mu_k(x^n_1) \leq \pi(g; x^n_1) + \sqrt{\frac{\ln S}{2(k+1)}} \] for any k,S
Markov Predictors

\[ \hat{\mu}_k(x^n_1) \leq \mu_k(x^n_1) + O(\sqrt{2^k/n}) \]

\[ \rightarrow \mu(x) \]

\[ \rightarrow \pi(x)? \]

\[ \mu(x) \geq \pi(x) \]

\[ \mu_k(x^n_1) \leq \pi(g; x^n_1) + \sqrt{\frac{\ln S}{2(k+1)}} \]

\[ \leq \pi_S(x^n_1) + \sqrt{\frac{\ln S}{2(k+1)}} \]

\[ \mu(x) = \pi(x) \] for any k,S
Proofs

$$\hat{\mu}_k(x) \to \lim_{n \to \infty} \mu(x^n_1) = \mu(x) = \pi(x)$$

Bottom line: Markov Predictor + Increasing Order achieves FS predictability

LZ algorithm does the job
Other work

Universal prediction of individual binary sequences in the presence of noise - T. Weissman and N. Merhav ’99.

- Predict the next outcome of an individual binary sequence, based on noisy observations of the past.
- Predictor competes with “set of experts”, performs “almost” as well as best of the experts.

On context-tree prediction of individual sequences - Jacob Ziv, Neri Merhav.

- the prediction is based on a ``context'' (or a state) that consists of the k most recent past outcomes $x_{t-k},...,x_{t-1}$, where the choice of $k$ may depend on the contents of a possibly longer, though limited, portion of the observed past, $x_{t-k_{\text{max}}},...,x_{t-1}$
Other work

**Finite-Memory Universal Prediction of Individual Sequences** - Eado Meron and Meir Feder ‘04.

- FS predictor can be deterministic or stochastic.
- g can be stochastic.

**SEQUENTIAL PREDICTION OF INDIVIDUAL SEQUENCES UNDER GENERAL LOSS FUNCTIONS** - D Haussler – 1998

**Universal Prediction of Individual Binary Sequences in the Presence of Arbitrarily Varying, Memoryless Additive Noise** – T Weissman 00
Future Work?

In general: Sequential Decision Problems

- Data Compression
- Gambling
- Prediction

Directed information??
Causal but not just 1 time-step