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

$$x_1, x_2, \cdots, x_t$$

$$\hat{x}_{t+1}$$

$\uparrow$

At time t:

Binary Output Sequence

$$x_1, x_2, \cdots, x_t \; x_{t+1} \;\; x_{t+2} \;\; \ldots\ldots\ldots$$

$$\hat{x}_{t+1} \;\; \hat{x}_{t+2} \;\; \ldots\ldots\ldots$$

At time t:

At time t+1:

Binary Output Sequence

$$\mathbf{x} = \quad x_1, x_2, \cdots, x_t \; x_{t+1} \;\; x_{t+2} \;\; \ldots\ldots\ldots$$

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

$\mathbf{x} = \quad x_1, x_2, \cdots, x_t \; x_{t+1} \;\; x_{t+2} \;\; \ldots\ldots\ldots$ 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.

- i.i.d., then Past $\not\Rightarrow$ Future.
- Predictors helpful whenever Past helps in predicting the future(Patterns).

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)

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:

$$\hat{x}_{t+1} = f(s_t) \qquad s_t \in \mathcal{S} = \{1, 2, \cdots, S\}$$

Next State Rule:

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

Finite State Predictor:

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)

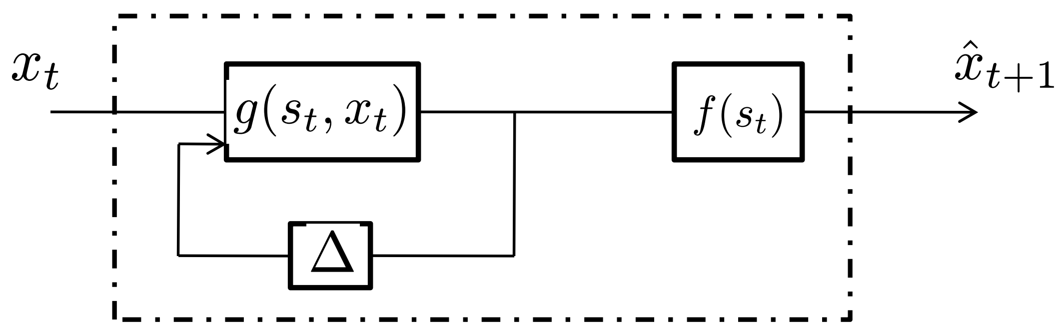
Predictor Rule:

$$\hat{x}_{t+1} = f(s_t) \qquad\qquad s_t \in \mathcal{S} = \{1, 2, \cdots, S\}$$

Next State Rule:

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

Finite State Predictor:



f can be stochastic

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

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

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

- 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

Literature

Markov Predictor: $s_t = (x_{t-k}, \cdots, x_{t-1})$

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

Literature

Markov Predictor: $s_t = (x_{t-k}, \cdots, x_{t-1})$

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

Fixed, Markov → 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 = π^{FS}
-

Fixed, Markov → 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 = π^{FS}
- Suppose no information is known about the sequence.

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

error -> π^{FS} ?

Fixed, Markov → 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 = π^{FS}
- Suppose no information is known about the sequence.

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

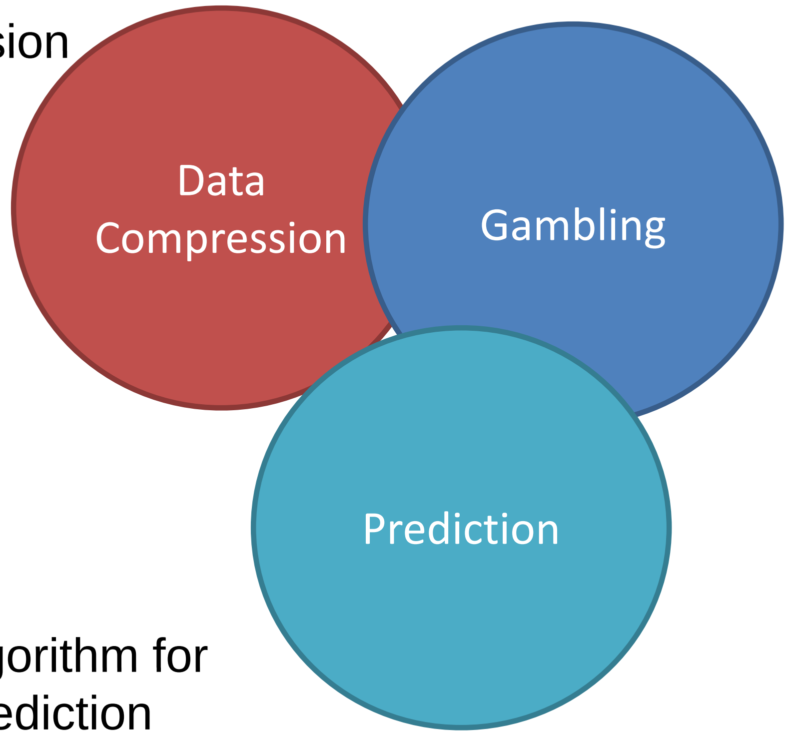
error -> π^{FS} ?

1. $\exists$ Markov Predictor $\approx \pi^{FS}$
2. Markov Predictor + increasing k → π^{FS}
3. Limpel-Ziv Parsing Algorithm: Markov Predictor with time varying order

Scope of the technique:

LZ algorithm for
Data Compression

LZ algorithm for
Gambling

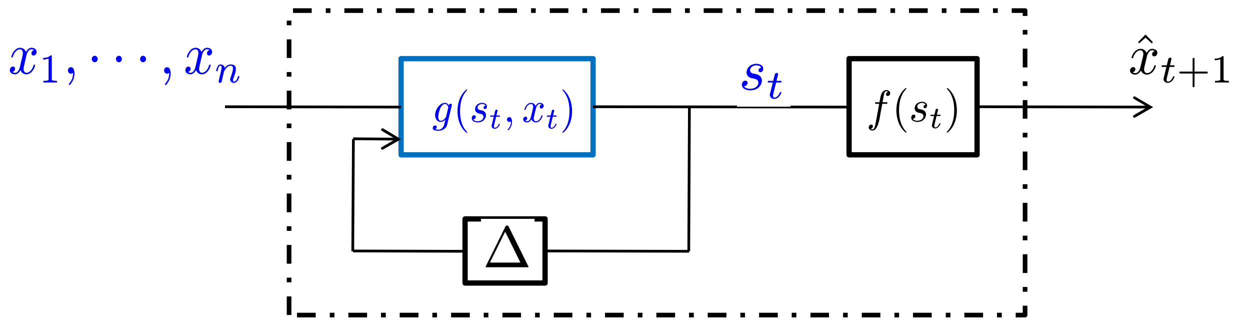


LZ algorithm for
Prediction

In general: Sequential Decision Problems

Predictablility

- Min fraction of prediction errors possible

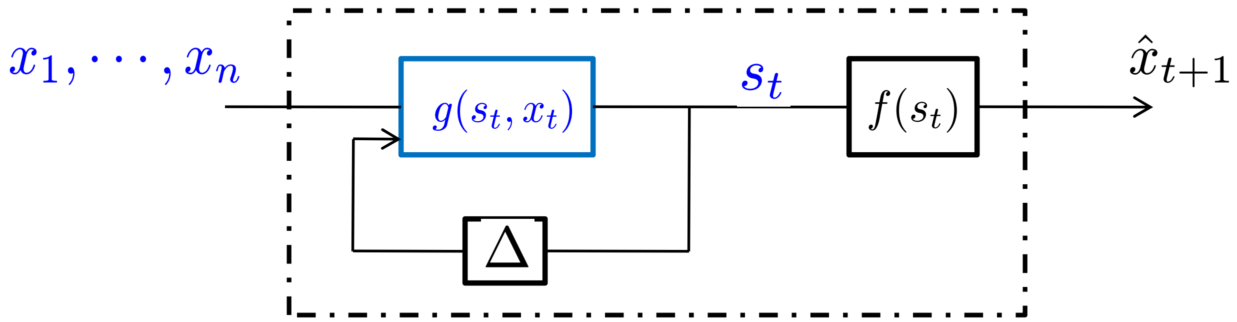


Fix a finite sequence: $x_1, \cdots, x_n$

Fix s_1,g : $s_1, \cdots, s_n$

Predictablility

- Min fraction of prediction errors possible



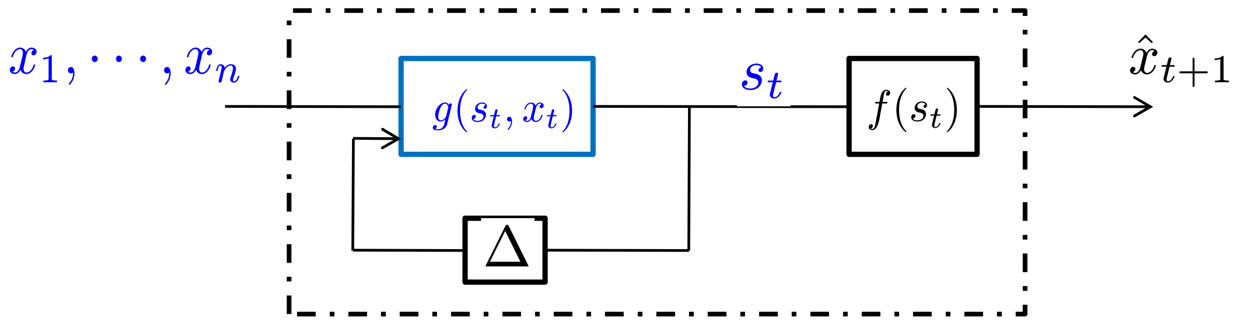
Fix a finite sequence: $x_1, \cdots, x_n$
Fix s_1,g : $s_1, \cdots, s_n$

Compute:

$N_n(s,0)$	$N_n(s,1)$
$N_n(1,0)$	$N_n(1,1)$
$N_n(2,0)$	$N_n(2,1)$
$N_n(S,0)$	$N_n(S,1)$

Predictablility

- 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:

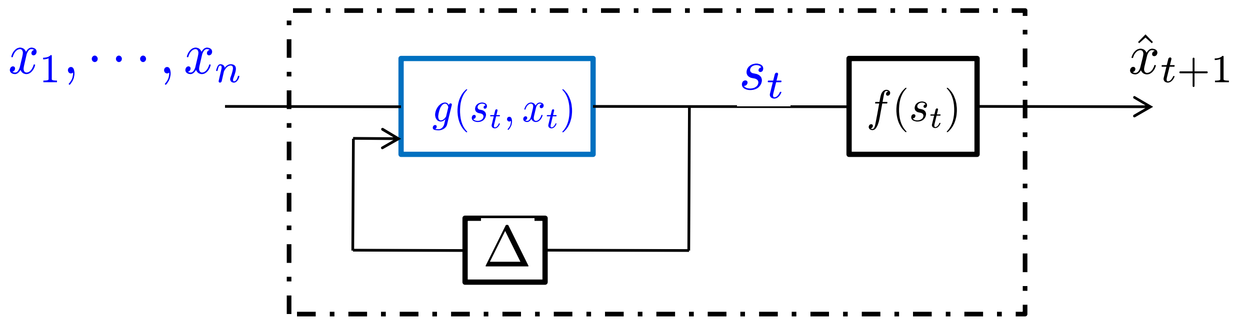
$N_n(s,0)$	$N_n(s,1)$
$N_n(1,0)$	$N_n(1,1)$
$N_n(2,0)$	$N_n(2,1)$
$N_n(S,0)$	$N_n(S,1)$

- Best prediction rule:

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

Predictablility

- 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:

$N_n(s,0)$	$N_n(s,1)$
$N_n(1,0)$	$N_n(1,1)$
$N_n(2,0)$	$N_n(2,1)$
$N_n(S,0)$	$N_n(S,1)$

- 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:

$$\pi(g; x_1^n) = \frac{1}{n} \sum_{i=1}^{S} \min\{N_n(s,0), N_n(s,1)\} \in [0, \tfrac{1}{2}]$$

Predicatability - 2

$\pi(g; x_1^n) \longrightarrow$ Fix x_1^n , S, g

Predicatability - 2

$\pi(g; x_1^n) \longrightarrow$ Fix x_1^n, S, g

- S-state predictability of x_1^n

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

Predicatability - 2

$\pi(g; x_1^n) \longrightarrow$ Fix x_1^n , S, g

- S-state predictability of x_1^n

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

- asymptotic S-state predictability

$$\pi_S(\mathbf{x}) = \limsup_{n \to \infty} \pi_S(x_1^n) \longrightarrow \text{Fix } \mathbf{x} \text{ , S}$$

Predicatability - 2

$\pi(g; x_1^n) \longrightarrow$ Fix x_1^n, S, g

- S-state predictability of x_1^n

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

- asymptotic S-state predictability

$$\pi_S(\mathbf{x}) = \limsup_{n \to \infty} \pi_S(x_1^n) \longrightarrow \text{Fix } \mathbf{x}\text{, S}$$

- FS predictability

$$\pi(\mathbf{x}) = \lim_{S \to \infty} \pi_S(\mathbf{x}) \longrightarrow \text{Fix } \mathbf{x}$$

Predicatability - 2

$\pi(g; x_1^n)$ $\longrightarrow$ Fix x_1^n , S, g

- S-state predictability of x_1^n

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

- asymptotic S-state predictability

$$\pi_S(\mathbf{x}) = \limsup_{n \to \infty} \pi_S(x_1^n) \longrightarrow \text{Fix } \mathbf{x} \text{, S}$$

- FS predictability

$$\pi(\mathbf{x}) = \lim_{S \to \infty} \pi_S(\mathbf{x}) \longrightarrow \text{Fix } \mathbf{x}$$

Note: Attained by FSM that depend on particular sequence $\mathbf{x}$

We want sequential prediction scheme which work independent of x

and yet achieve $\pi(\mathbf{x})$

Predicatbility-2

$$\pi(g; x_1^n) \xleftarrow{\text{Propose a scheme}} \hat{\pi}(g; x_1^n)$$

$$\pi_S(x_1^n) \longleftarrow \hat{\pi}_S(x_1^n)$$

$$\pi_S(\mathbf{x}) \longleftarrow \hat{\pi}_S(\mathbf{x})$$

$$\pi(\mathbf{x}) \longleftarrow \hat{\pi}(\mathbf{x})$$

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.

A, B, C, D, E

00101010100….. ------------> {X,0,01,010,1,0100,……}

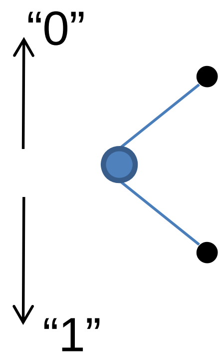
LZ incremental parsing algo

• 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.



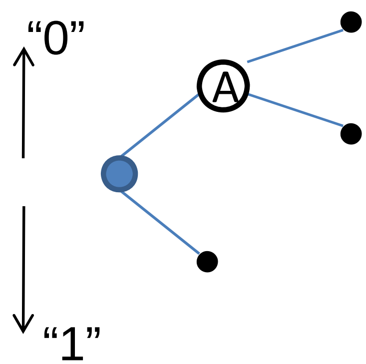
LZ incremental parsing algo

• 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.



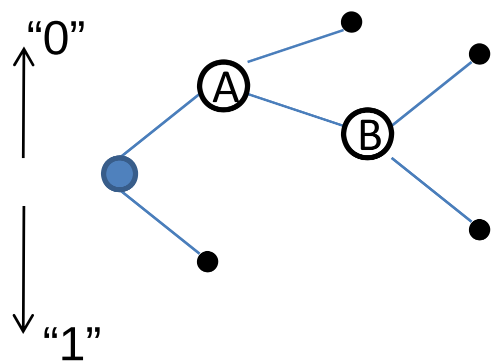
LZ incremental parsing algo

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

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.



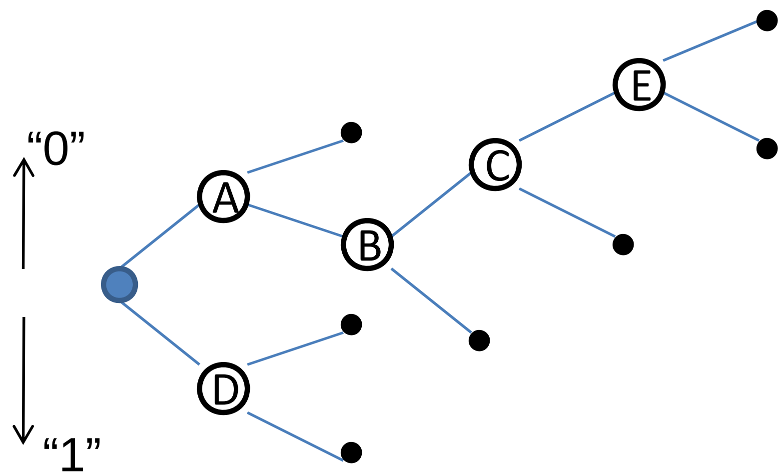
LZ incremental parsing algo

• 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.



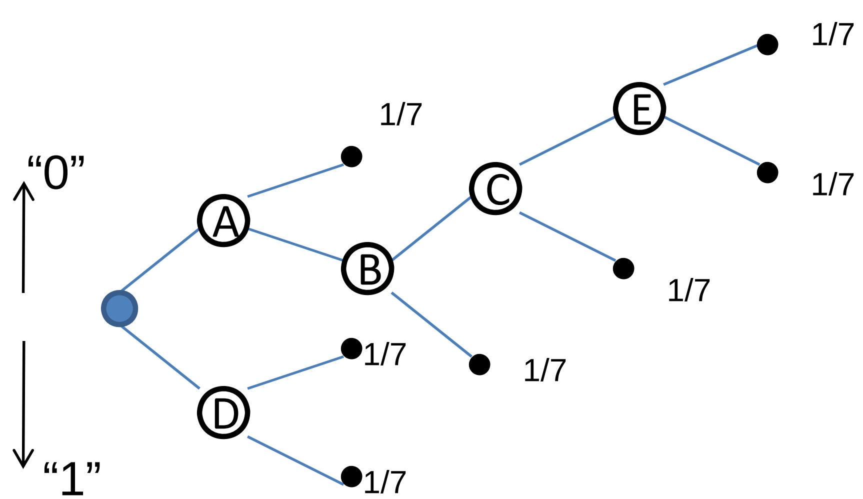
LZ incremental parsing algo

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

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.



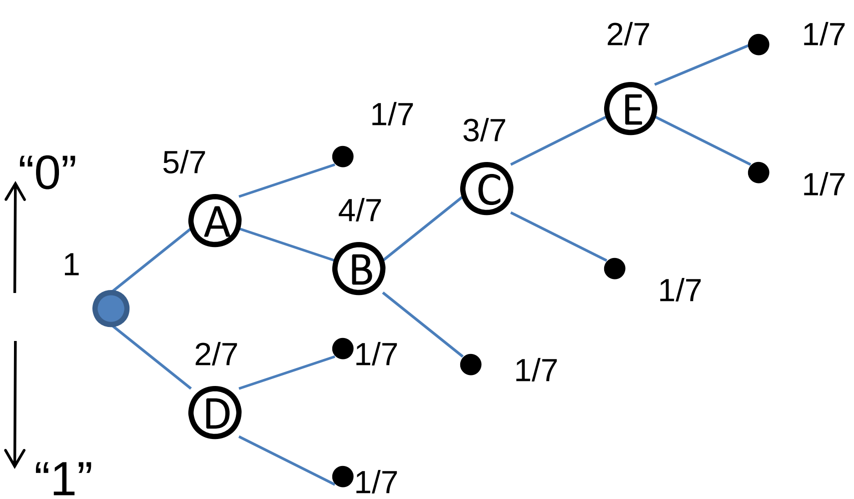
LZ incremental parsing algo

- 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.



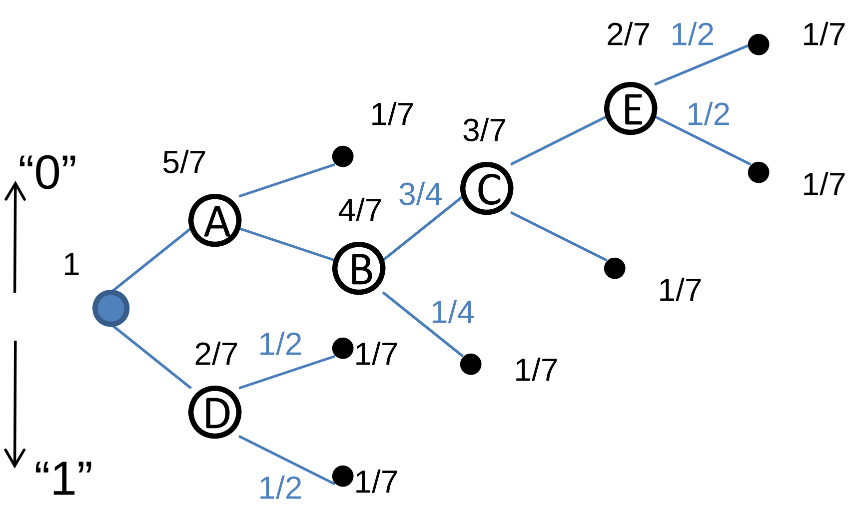
LZ incremental parsing algo

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

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.



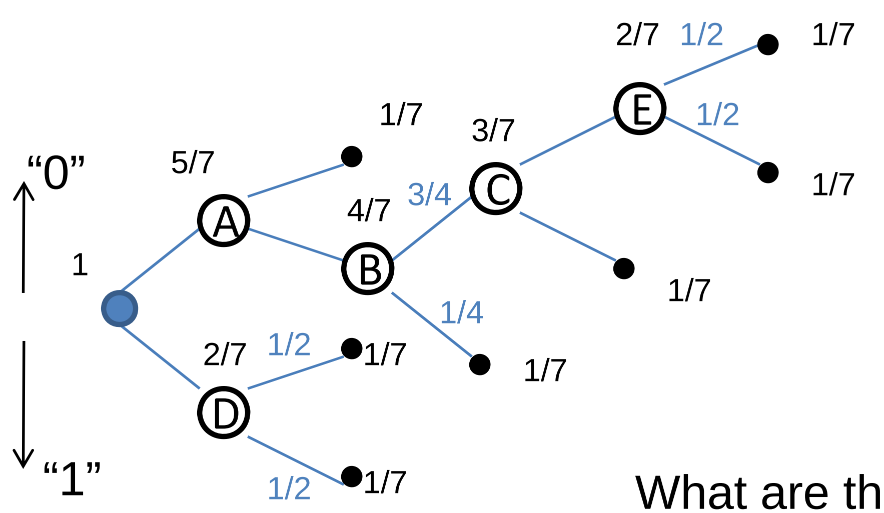
LZ incremental parsing algo

• 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.



What are these probabilities?

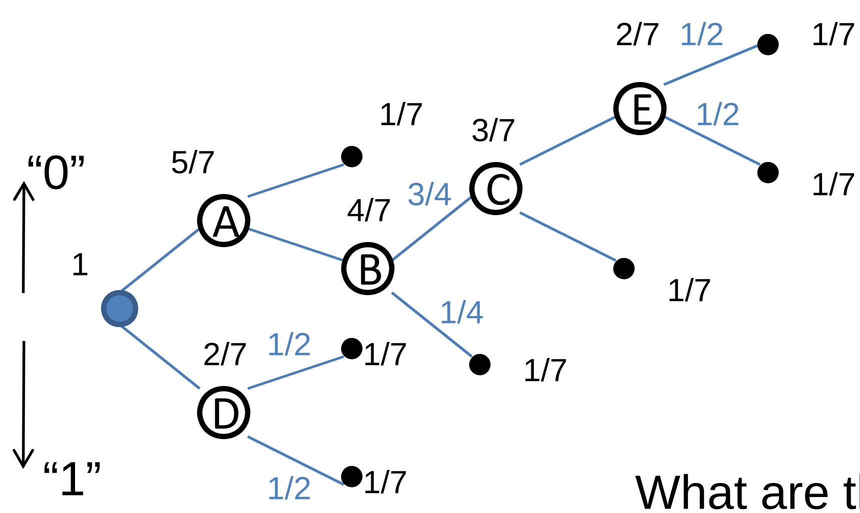
LZ incremental parsing algo

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

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.

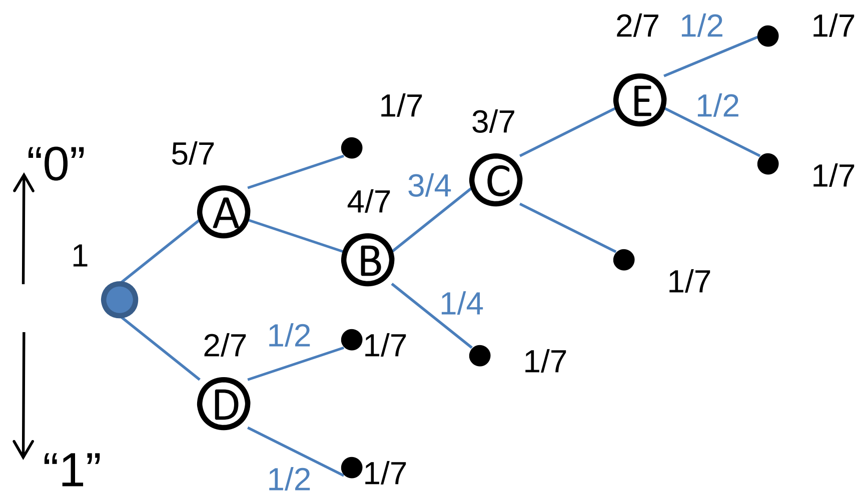


What are these probabilities?

Conditional probabilities of x_{t+1}

$$\hat{p}^{LZ}(x_{t+1}|x_1^t)$$

LZ incremental parsing algo-2



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

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

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

$$\hat{p}_x = \frac{N_t^j(x)+1}{N_t^j+2}$$

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.

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

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

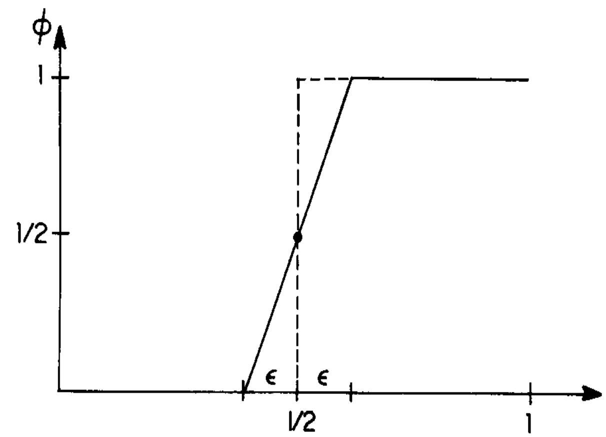
LZ incremental parsing algo-3

- 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: $1 - \phi(\hat{p}_t(x_{t+1}))$

$$\hat{\pi}(x_1^n) = \frac{1}{n} \sum_{i=0}^{n} (1 - \phi(\hat{p}_t(x_{t+1})))$$

$$\hat{\pi}(x_1^n) \rightarrow \pi(\mathbf{x})$$

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: $1 - \phi(\hat{p}_t(x_{t+1}))$

$$\hat{\pi}(x_1^n) = \frac{1}{n}\sum_{i=0}^{n}(1 - \phi(\hat{p}_t(x_{t+1})))$$

$$\hat{\pi}(x_1^n) \rightarrow \pi(\mathbf{x})$$

A, B, C, D, E

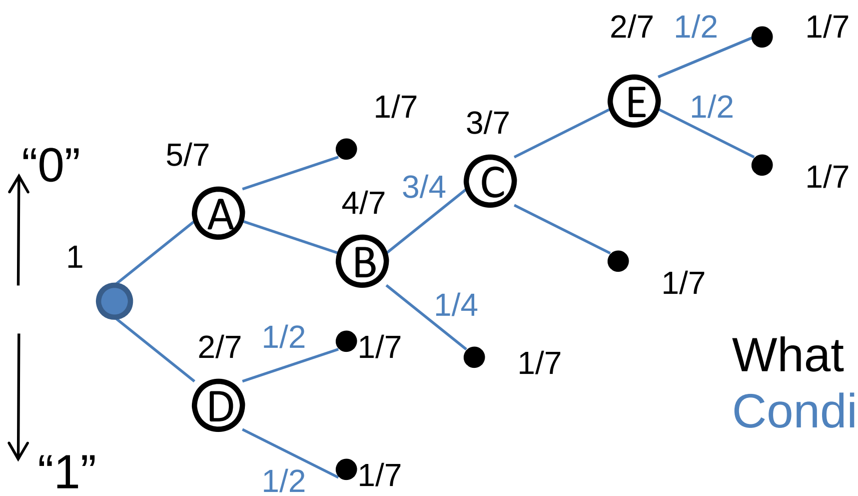
00101010100….. ------------> {X,0,01,010,1,0100,……}

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



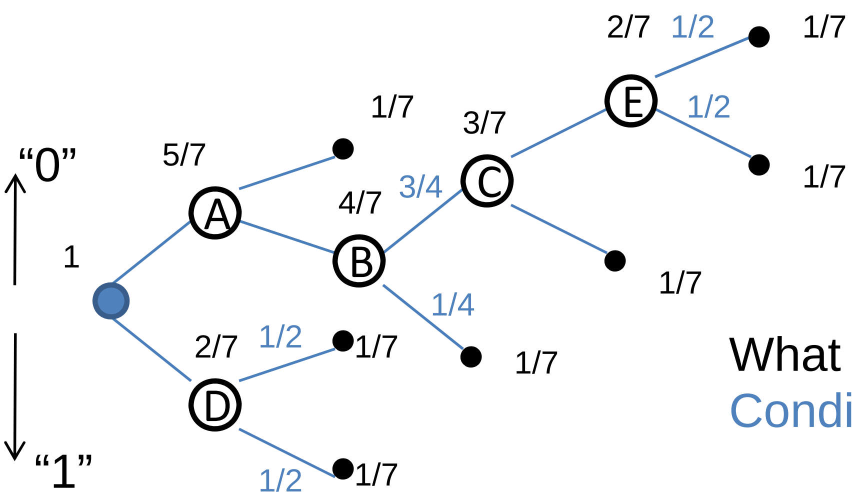
What are these probabilities?

Conditional probabilities of x_{t+1}

$$\hat{p}^{LZ}(x_{t+1}|x_1^t)$$

- 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?

LZ algorithm for Gambling



What are these probabilities?

Conditional probabilities of x_{t+1}

$$\hat{p}^{LZ}(x_{t+1}|x_1^t)$$

- 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?

$\hat{p}^{LZ}(x_{t+1}|x_1^t) \longrightarrow$ (Prediction) Use to predict $\hat{x}_{t+1}$

$\longrightarrow$ (Gambling) Invest $\hat{p}_0$ on Horse 0 and $\hat{p}_1$ on Horse 1.

Gambling Using a Finite State Machine

- Finite State Complexity:

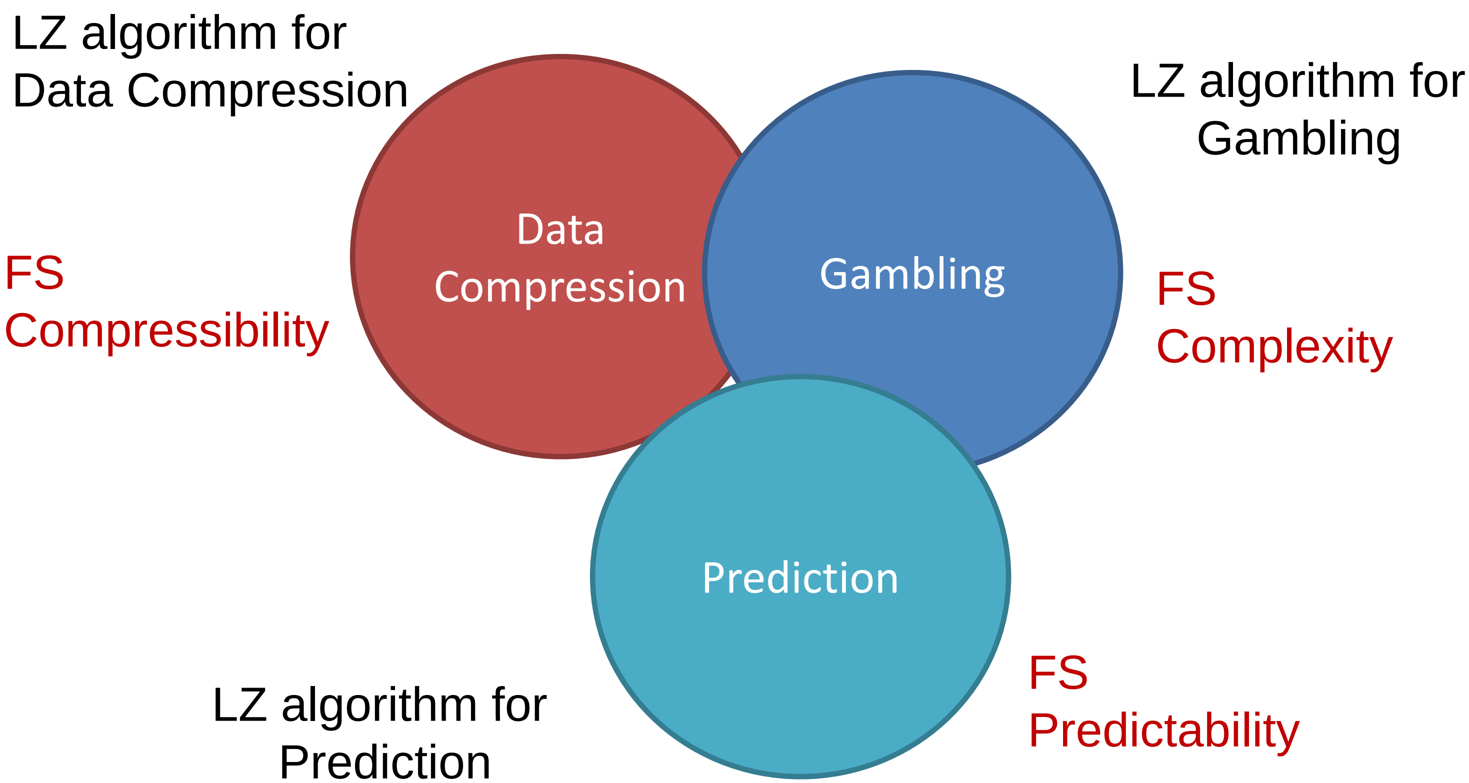
$$S_n = S_0 2^{n(1-H^{FS}(x_1^n))}$$

- Using LZ algorithm for Gambling:

$$S_n = S_0 2^{n(1-\hat{H}^{LZ}(x_1^n))}$$

$$\hat{H}^{LZ} \to H^{FS}$$ [Meir Feder ‘91]

Scope of the technique:



In general: Sequential Decision Problems

Proofs

- S = 1 Single-State Machine

Fix a finite sequence: $x_1, \cdots, x_n$

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

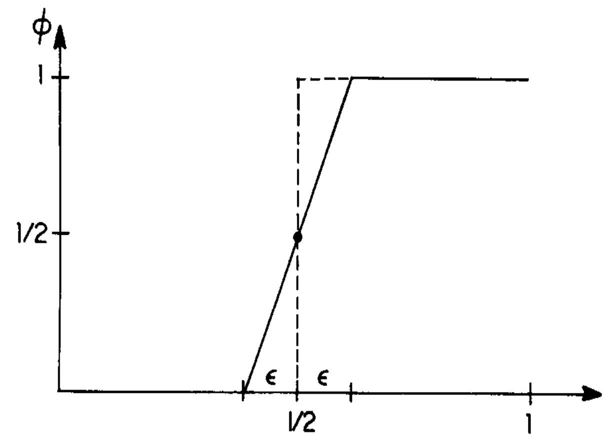
- S = 1   Single-State Machine

Fix a finite sequence: $x_1, \cdots, x_n$

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

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



$$\hat{\pi}_1(x_1^n) \to \pi_1(x_1^n), \quad \forall x_1^n$$

Proof – Step 1

- S = 1   Single-State Machine

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

$\pi(x_1^n) = \frac{1}{n} N_n(1,1)$ $\longrightarrow$ Predicts "0" every time.

$\hat{\pi}(x_1^n) \leq \hat{\pi}(\tilde{x}_1^n)$ $\longrightarrow$ Worst sequence

$\underleftrightarrow{0101010101\ldots.\ 01}$ $\underleftrightarrow{0000000000}$

$\hat{\pi}(\tilde{x}_1^n)$ = E[fraction of errors] ----- as a function of ϵ

$$\hat{\pi}(\tilde{x}_1^n) = \frac{1}{n} \sum_{i=0}^{n} (1 - \phi(\hat{p}_t(x_{t+1})))$$

$$\leq \frac{N_n(1,1)}{n} + \frac{\epsilon}{1-2\epsilon} + O(\frac{\log n}{n}) \qquad \epsilon \text{ fixed}$$

$$\leq \frac{N_n(1,1)}{n} + O(\frac{1}{\sqrt{n}}) \qquad \epsilon_t = \epsilon = \frac{1}{2\sqrt{t+2}}$$

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

$$\pi(g; x_1^n) \xleftarrow{\text{Propose a scheme}} \hat{\pi}(g; x_1^n)$$

$$\pi_S(x_1^n) \longleftarrow \hat{\pi}_S(x_1^n)$$

$$\pi_S(\mathbf{x}) \longleftarrow \hat{\pi}_S(\mathbf{x})$$

$$\pi(\mathbf{x}) \longleftarrow \hat{\pi}(\mathbf{x})$$

Proof-Step 2

- S known, g known

Fix a finite sequence: $x_1, \cdots, x_n$

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

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

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

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

$$\leq \pi(g; x_1^n) + O(\sqrt{S/n})$$

Proofs

$$\hat{\pi}(x_1^n) \leq \frac{N_n(1,1)}{n} + O(\frac{1}{\sqrt{n}})$$

Proposed a Scheme
Compute worst Case performance

$$O(\sqrt{S/n}) \qquad \pi(g; x_1^n) \longleftarrow \hat{\pi}(g; x_1^n)$$

$$\pi_S(x_1^n) \longleftarrow \hat{\pi}_S(x_1^n)$$

$$\pi_S(\mathbf{x}) \longleftarrow \hat{\pi}_S(\mathbf{x})$$

$$\pi(\mathbf{x}) \longleftarrow \hat{\pi}(\mathbf{x})$$

Refinement of an FS machine

$$g \to \tilde{g} \qquad\qquad s.t. \quad s_t = h(\tilde{s}_t)$$

A refinement can do better than the original.

$$\pi(g; x_1^n) \geq \pi(\tilde{g}; x_1^n)$$

For a given S, over all g $\in$ G_S

|G| = S^{2S}

$$\tilde{s}_t = (s_t^1, s_t^2, \cdots, s_t^M)$$

$$\pi(\tilde{g}; x_1^n) \leq \pi(g; x_1^n) \qquad \forall g \in G_S$$

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

$$O(\sqrt{S^{2S}/n})$$

Proofs

$\hat{\pi}(x_1^n) \leq \frac{N_n(1,1)}{n} + O(\frac{1}{\sqrt{n}})$ Proposed a Scheme
Compute worst Case performance

$O(\sqrt{S/n})$ $\pi(g; x_1^n) \longleftarrow \hat{\pi}(g; x_1^n)$

$O(\sqrt{S^{2S}/n})$ $\pi_S(x_1^n) \longleftarrow \hat{\pi}_S(x_1^n)$ S-State predictability

Define a new refined state

$\pi_S(\mathbf{x}) \longleftarrow \hat{\pi}_S(\mathbf{x})$

$\pi(\mathbf{x}) \longleftarrow \hat{\pi}(\mathbf{x})$

Markov Predictors

$$s_t = (x_{t-k}, \cdots, x_{t-1})$$

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

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

Scheme:

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

$$\hat{p}_x = \frac{N_t(x_{t-k+1} \cdots x_t 0) + 1}{N_t(x_{t-k+1} \cdots x_t) + 2}$$

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

Markov Predictors

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

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

$\lim_{k\to\infty} \mu_k(x) = \mu(x)$ Markov Predictability

Markov Predictors

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

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

$\lim_{k\to\infty} \mu_k(x) = \mu(x)$ Markov Predictability

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

Markov Predictors

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

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

$\lim_{k\to\infty} \mu_k(x) = \mu(x)$ Markov Predictability

To attain μ(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

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

$$\to \mu(x)$$

Markov Predictors

$$\hat{\mu}_k(x_1^n) \le \mu_k(x_1^n) + O(\sqrt{2^k/n})$$

$$\to \mu(x)$$

$$\to \pi(x)?$$

Markov Predictors

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

$$\to \mu(x)$$

$$\to \pi(x)?$$

$$\mu(x) \geq \pi(x)$$

$$\mu_k(x_1^n) \leq \pi(g; x_1^n) + \sqrt{\frac{\ln S}{2(k+1)}}$$ for any k,S

Markov Predictors

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

$$\to \mu(x)$$

$$\to \pi(x)?$$

$$\mu(x) \geq \pi(x)$$

$$\mu_k(x_1^n) \leq \pi(g; x_1^n) + \sqrt{\tfrac{\ln S}{2(k+1)}} \qquad \text{for any k,S}$$

$$\leq \pi_S(x_1^n) + \sqrt{\tfrac{\ln S}{2(k+1)}}$$

$$\mu(x) = \pi(x)$$

Proofs

$$\hat{\mu}_k(x) \to \lim_{n\to\infty} \mu(x_1^n) = \mu(x) = \pi(x)$$

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 $x_{t-k},\ldots,x_{t-1}$, where the choice of k may depend on the contents of a possibly longer, though limited, portion of the observed past, $x_{t-k\_max},\ldots,x_{t-1}$

Other work

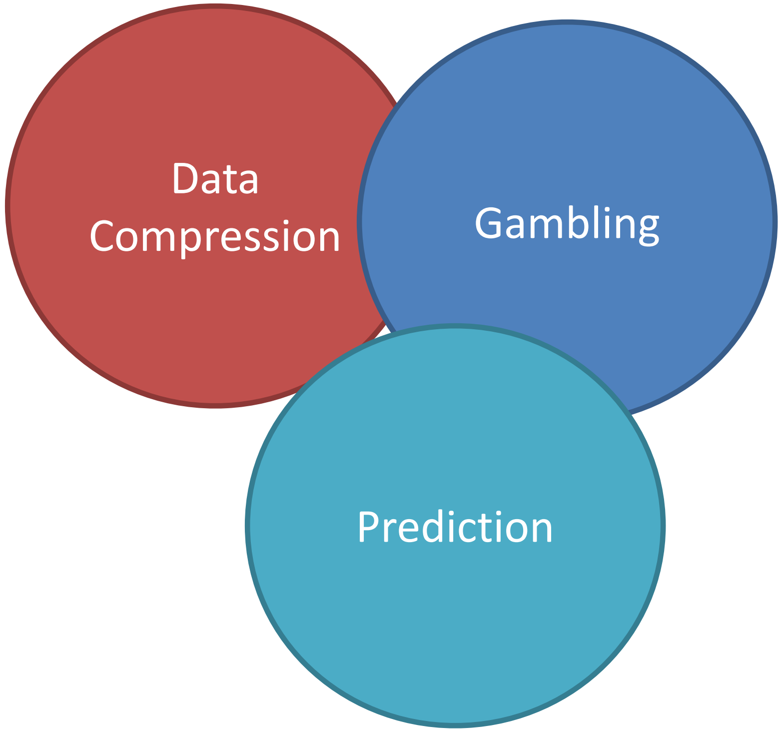
Finite-Memory Universal Prediction of Individual Sequences - Eado Meron and Meir Feder ‘04.

- 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?



Directed information??

Causal but not just 1 time-step

In general: Sequential Decision Problems