Learning, Sampling, and Memory, in the Small Data Regime

[Some] New Problems and Perspectives

Last Lecture, CS265/CME309

Gregory Valiant

3 Vignettes:

Learning Populations of Parameters:

Can a large number of data sources compensate for heterogeneity among sources?

(NIPS'17, w. Weihao Kong and Kevin Tian, ICML'19, w. Ramya Vinayak, Sham Kakade, Weihao Kong)

Sample Amplification:

Is *sampling* easier than learning?

(w. Brian Axelrod, Shivam Garg, Vatsal Sharan.)

Selective Prediction:

Can we make accurate predictions about the future without any assumptions that the future will be like the past?

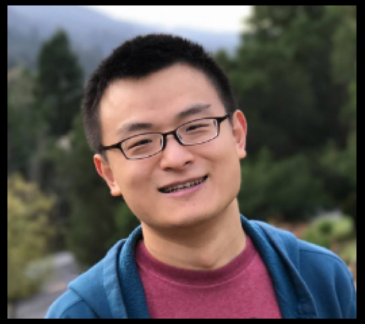
(COLT'19 w. Mingda Qiao)

Part I

Learning Populations of Parameters

- Tian, Kong, Valiant , *Learning Population of Parameters*, NeurIPS'17
- Vinayak, Kong, Valiant, Kakade, *Maximum Likelihood Estimation for Learning Populations of Parameters,* ICML'19,

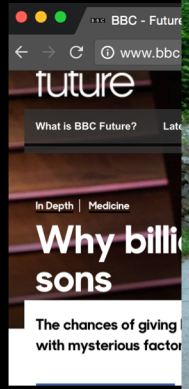
Kevin Tian



Weihao Kong

Ramya Vinayak

Sham Kakade



romosome in human spermatozoa

G Vosa - Nature, 1970 - Springer

TEMPTS have been made to distinguish between
a of mammals containing either an X or a Y
e by looking for differences in size and surface
f the two classes of cells (see refs. 1 and 2 for
t the results of such

5 Related articles More

A Basic Question: *Do some people have higher probabilities of having male vs female children?*

Why is it hard to study?

- Each person provides little data

Broader Motivation

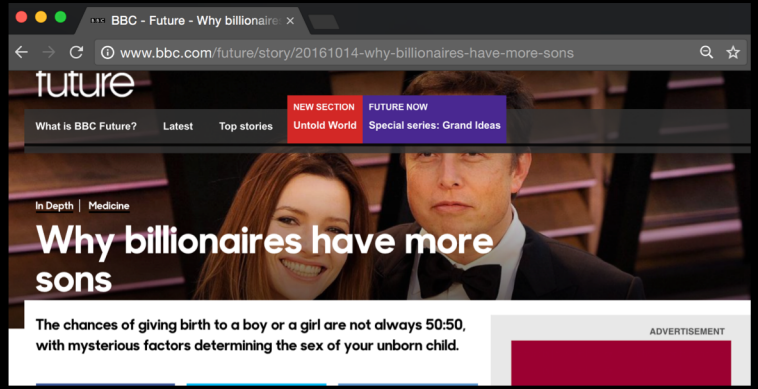
Many datasets consist of **large** number of heterogeneous users/sources, each contributing a modest amount of data.

Main Goals:

- Learn **population** of models/parameters
- Use knowledge of population to improve estimates of each user's model

"Federated Learning"

To what extent can the large number of sources compensate for the lack of data from each source?



The Y chromosome in human spermatozoa

P Barlow, CG Vosa - Nature, 1970 - Springer

Abstract ATTEMPTS have been made to distinguish between spermatozoa of mammals containing either an X or a Y chromosome by looking for differences in size and surface properties of the two classes of cells (see refs. 1 and 2 for reviews), but the results of such

Cited by 275 Related articles More

A Basic Question: *Do some people have higher probabilities of having male vs female children?*

Why is it hard to study?

- Each person provides little data

Model: *n* people, *i*th person has coin with bias p_i

Observe *t* i.i.d. tosses of each coin

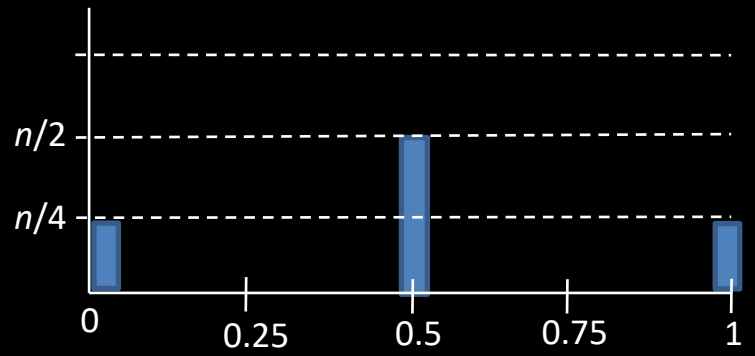
p_1	p_2	p_3	p_4	p_5	p_6	p_7	. . .	p_{n-1}	p_n
H	T	T	T	T	T	H	. . .	H	T
H	H	H	T	T	H	H	. . .	T	T

t

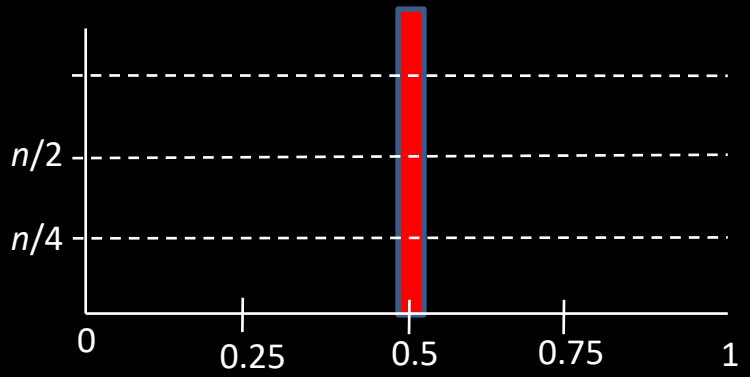
Empirical estimates of p_i's are unreliable: *t tosses*-->error 1/sqrt(t)

e.g. t=2, suppose we observe:

Can conclude most p_i's *are* ≈ 0.5



Histogram/Distribution of empirical $\hat{p}_i$



Recovered Histogram/Distribution

Why? *If > 10% p_i's NOT in (0.48, 0.52), then expect to see different statistics.*

Model: *n* people, *i*th person has coin with bias p_i

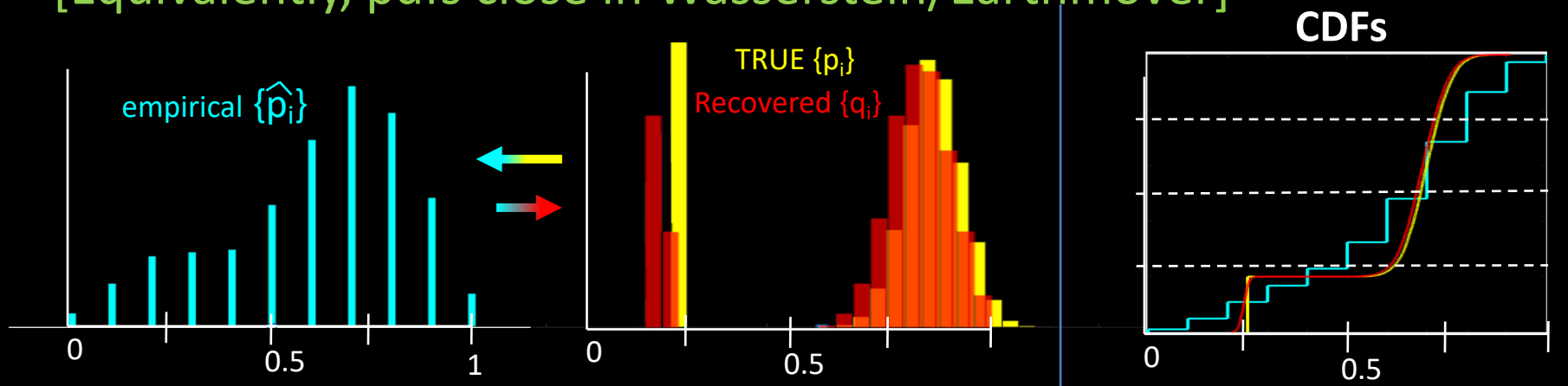
Observe *t* i.i.d. tosses of each coin

p_1	p_2	p_3	p_4	p_5	p_6	p_7	. . .	p_{n-1}	p_n
H	T	T	T	T	T	H	. . .	H	T
H	H	H	T	T	H	H	. . .	T	T

t

Empirical estimates of p_i's are unreliable: *t tosses*-->error 1/sqrt(t)

Goal: Return approximation {*q*} of the set/distribution {p_i} s.t. cumulative density functions (cdfs) are close in L1 distance. [Equivalently, pdfs close in Wasserstein/Earthmover]



Model: *n* people, *i*th person has coin with bias p_i

Observe *t* i.i.d. tosses of each coin

p_1	p_2	p_3	p_4	p_5	p_6	p_7	. . .	p_{n-1}	p_n
H	T	T	T	T	T	H	. . .	H	T
H	H	H	T	T	H	H	. . .	T	T

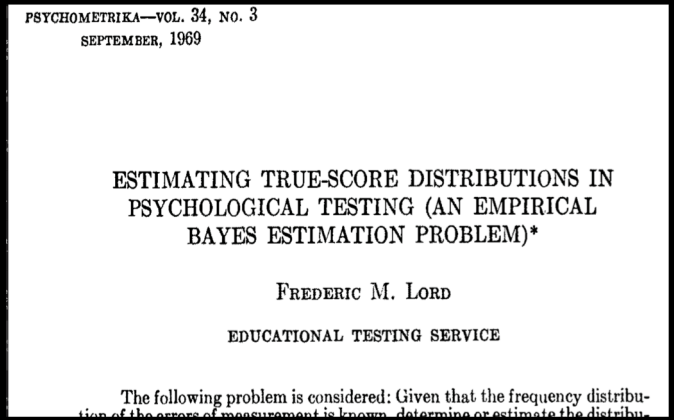
t

Empirical estimates of p_i's are unreliable: *k tosses*-->error 1/sqrt(t)

Goal: Return approximation {*q*} of the set/distribution {p_i} s.t. cumulative density functions (cdfs) are close in L1 distance.

Frederic M. Lord first considered this problem in the psychological testing setting [Lord 65, 69] while working for ETS.

[Lord's research shaped SAT, GRE, GMAT, LSAT and the TOEFL test, and was called "Father of Modern Testing"]



PSYCHOMETRIKA—VOL. 34, NO. 3
SEPTEMBER, 1969

ESTIMATING TRUE-SCORE DISTRIBUTIONS IN PSYCHOLOGICAL TESTING (AN EMPIRICAL BAYES ESTIMATION PROBLEM)*

FREDERIC M. LORD

EDUCATIONAL TESTING SERVICE

The following problem is considered: Given that the frequency distribu-

Model: *n* people, *i*th person has coin with bias p_i

Observe *t* i.i.d. tosses of each coin

p_1	p_2	p_3	p_4	p_5	p_6	p_7	. . .	p_{n-1}	p_n
H	T	T	T	T	T	H	. . .	H	T
H	H	H	T	T	H	H	. . .	T	T

t

Empirical estimates of p_i's are unreliable: *t tosses*-->error 1/sqrt(t)

Goal: Return approximation {*q*} of the set/distribution {p_i} s.t. cumulative density functions (cdfs) are close in L1 distance.

Thm: Can learn set {p_i} to [cdf L1] error **O(1/*t*)** using *t* samples from each individual, for $n>2^t$. (And this is optimal.)

Also extends naturally to more complex distributions, e.g. each person has (p_i, q_i, w_i, …), and goal is to learn joint distribution over parameters.

Algorithm/Proof

Accurately recover low-order moments of P, i.e. $\int x^j P(x)dx$ for j=1,2,…,t

+

Solve moment inverse problem (given approx. moments, return distribution/set)

Observation: If each person contributes $\geq t$ observations, can recover each of the first *t moments to error O* ($2^t/sqrt(N)$)

High-order moment estimation unreliable for $N<2^t$!!

Fact: Given 2 bounded distributions over reals, *P*, *Q* if first *t* moments match, then Wasserstein distance :

$W_1(P, Q) < 1/t$

(robust version if moments similar)

What about when N << exp(t) ???

Number of people: *N*	Minimax error rate	Algorithm
$N > 2^t$	$O(\frac{1}{t})$	Moment-Matching, MLE
$2^t > N > t^5$	$O(\frac{1}{\sqrt{t \log N}})$	MLE [VKVK'19]
$\text{const} < N < t^5$	??? $O(\frac{1}{\sqrt{t \log N}})$???	??? Probably MLE ???
$N = constant$	$O(\frac{1}{\sqrt{t}})$	Empirical

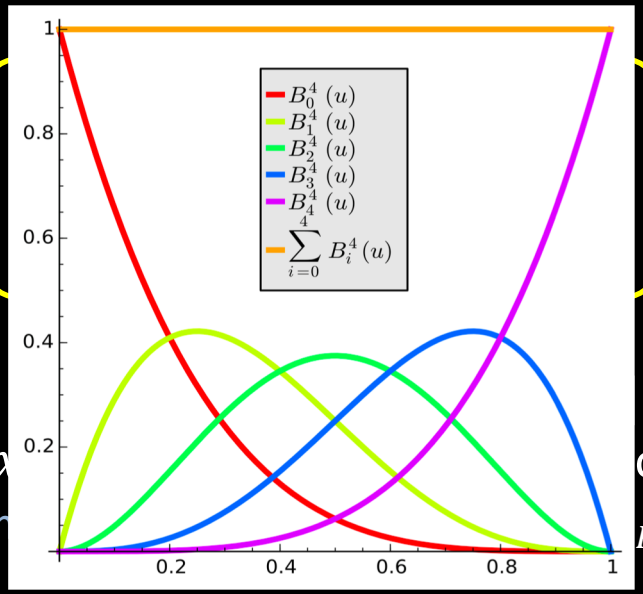
MLE: find distribution P s.t. if p_i drawn i.i.d. from P, maximizes likelihood of observed set of statistics. (Computationally tractable!!)

Proof Idea

P and P_{mle} have similar expected summary statistics.

+

Similar expected summary statistics imply small Wasserstein distance.



$B_s^t(x$

Com

omial

$_P[x^j]$

Given 2 distributions over [0,1], *P*, *Q*, if expected "fingerprints" differ by $O(1/\sqrt{N})$, then Wasserstein distance :

$$W_1(P, Q) < \max\left(\frac{1}{t}, \frac{1}{\sqrt{t \log N}}\right)$$

What about when N << exp(t) ???

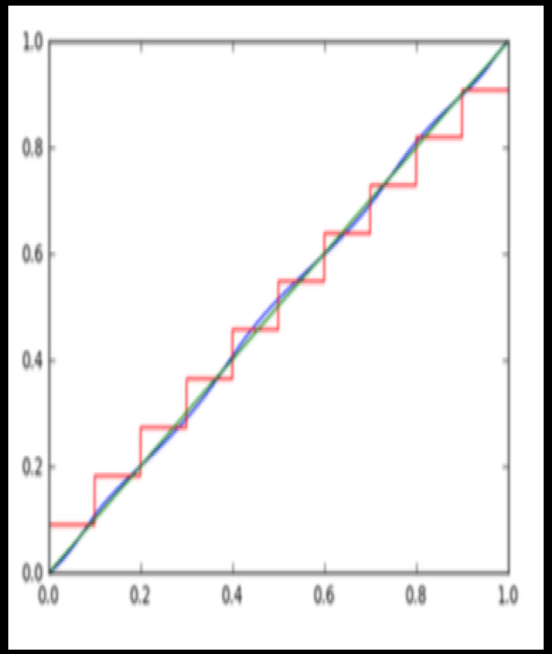
Number of people: *N*	Minimax error rate	Algorithm
$N > 2^t$	$O(\frac{1}{t})$	Moment-Mathing MLE
$2^t > N > t^5$	$O(\frac{1}{\sqrt{t \log N}})$	MLE [VKVK'19]
$\text{const} < N < t^5$	??? $O(\frac{1}{\sqrt{t \log N}})$???	??? Probably MLE ???
$N = constant$	$O(\frac{1}{\sqrt{t}})$	Empirical

MLE Algorithm: find unordered set $\{p_i\}$ that maximizes likelihood of observed set of statistics. (Computationally tractable!!)

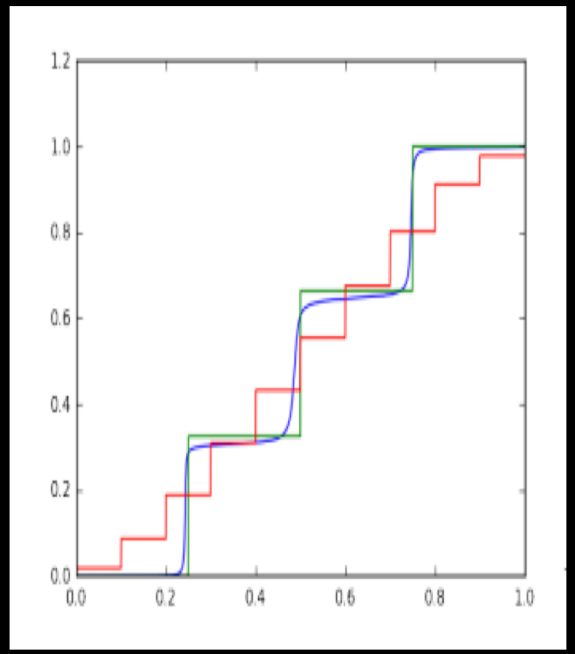
Empirical Results [validation]

Validation on 3 known distributions, *t=10*, *n = 10k*

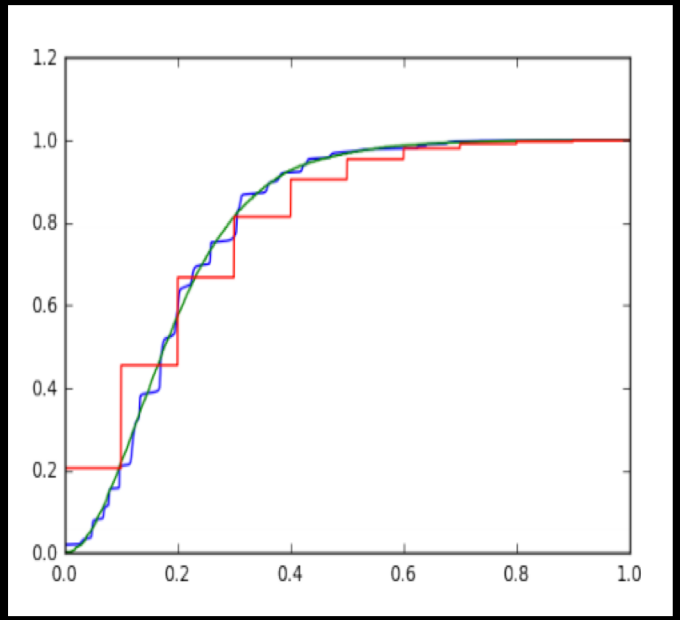
Green = ground truth, Red = empirical, Blue = recovered



Uniform on [0,1]



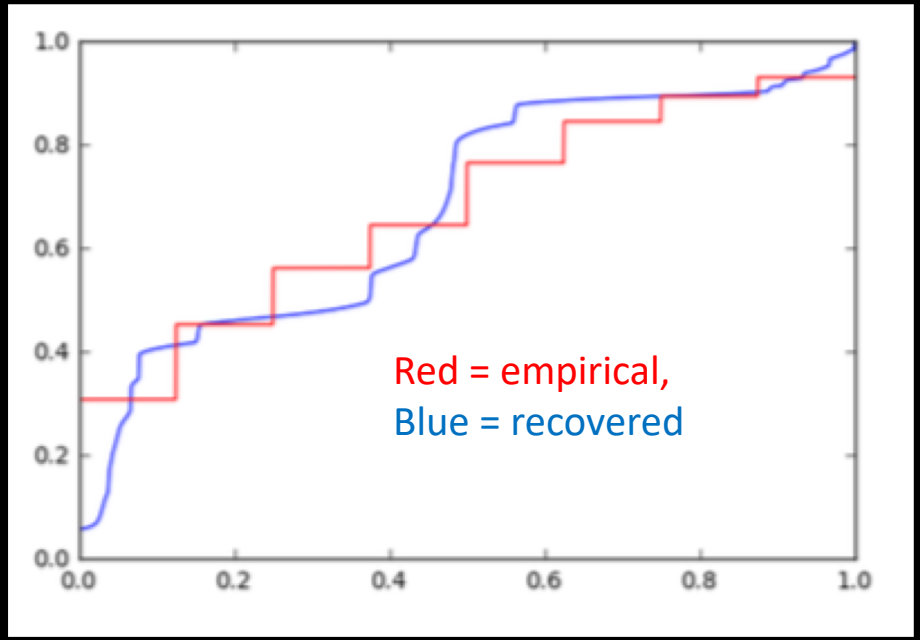
3-spike



p_i = probability flight *i* is significantly delayed

Empirical Results [election biases]

p_i = probability *i*th county votes democrat vs republican in each presidential election.
[t=8, elections 1976-2004 , *n* = 3116 counties]

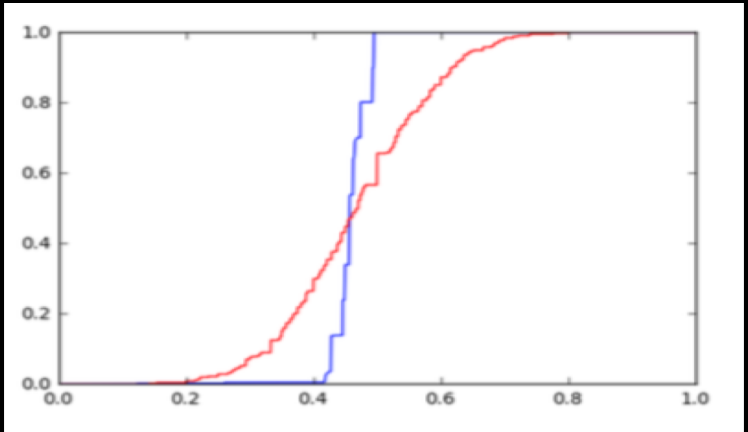


Recovered CDF (quite robust, similar results if only use first 6 or 7 moments)

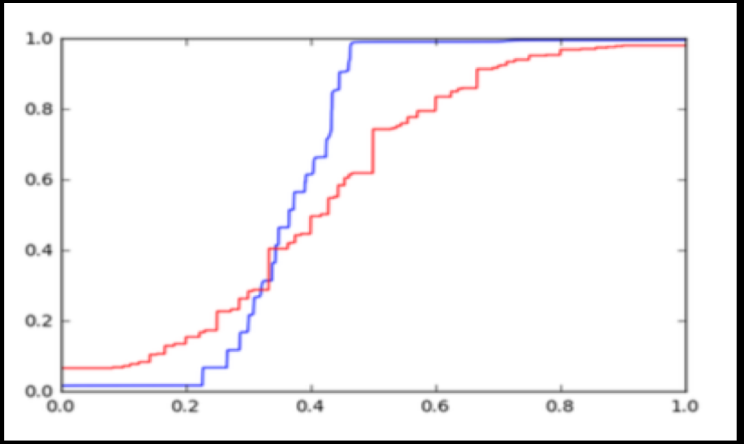
Empirical Results [basketball stats]

p_i = probability of scoring a 3-point attempt in *i*th basketball game.
[t=8, $n \approx 500$ games in which ≥ 8 shots attempted]

Red = empirical, Blue = recovered



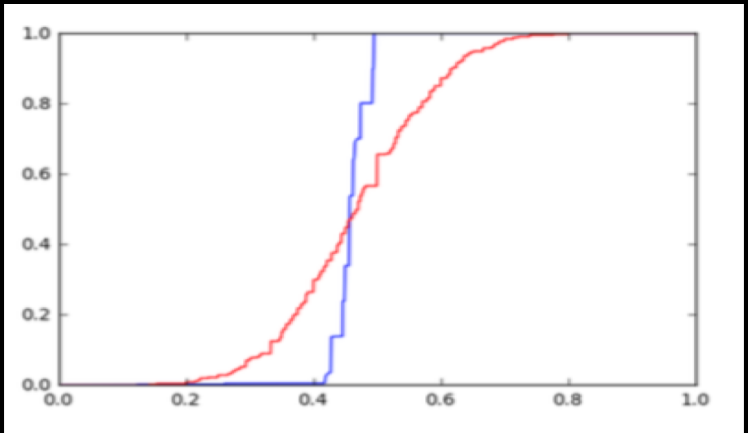
Stephen Curry
(known for consistency)



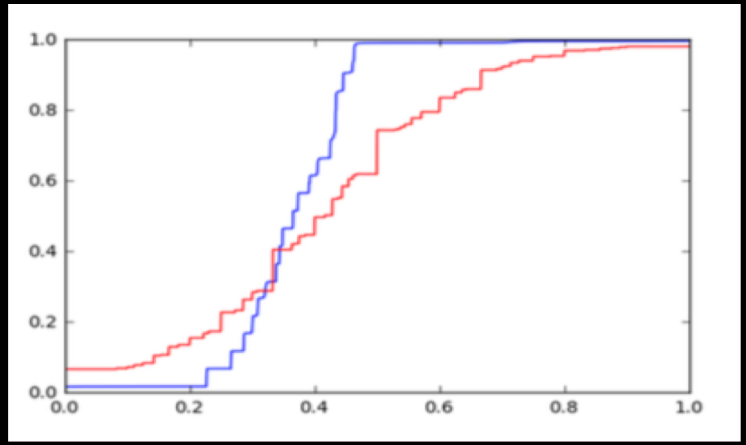
Danny Green
a.k.a. "Icy Hot"

Empirical Results [basketball stats]

Can use recovered $\{p_i\}$ as prior: e.g. given that Green has made 0 of 4 attempts so far, what is our estimate of the probability of making next shot? [i.e. should he be allowed to continue shooting?]



Stephen Curry
(known for consistency)



Danny Green
a.k.a. "Icy Hot"

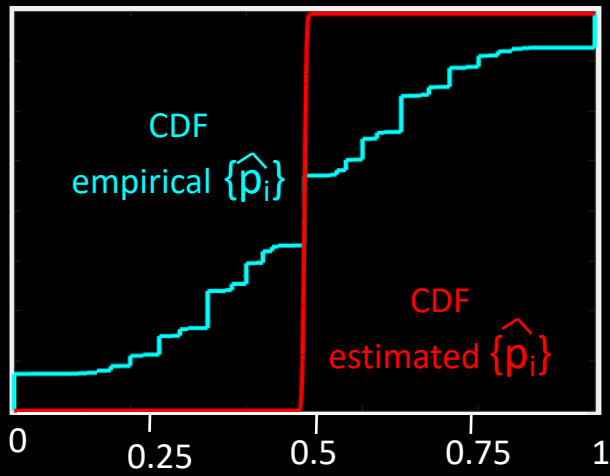
Empirical Results [Offspring Sex Ratios]

p_i = probability next offspring is Male vs Female

- Understanding $\{p_i\}$ is hard in humans…lack of independence, etc.
- Much easier in dogs: each litter typically has 4-8 puppies => can recover 4-8 moments!

[If prob(Male) significantly affected by relative time of sex/ovulation, should observe it in data!]

- Data from Norwegian Kennel Club: >20,000



Data consistent with all p_i's exactly ½.

Would need larger dataset (>1M litters) to resolve variance of 1%.... (still hoping to get this data)

Part II

Sample Amplification

Axelrod, Garg, Sharan, Valiant, Sample Amplification: Increasing Dataset Size even when Learning is Impossible (arxiv: 1904.12053)

Brian Axelrod

Shivam Garg

Vatsal Sharan

What does it mean that a GAN made this image?

(Does it mean that GANS "know" the distribution of renaissance portraits?)

Is it possible to generate samples from a distribution, without knowing the distribution?

How hard it is to tell whether a dataset is “genuine”, i.e. a set of i.i.d. draws from D?

Sample Amplification Setting

Amplifier

Input: *n* i.i.d. samples from D

Output: m > n "samples"

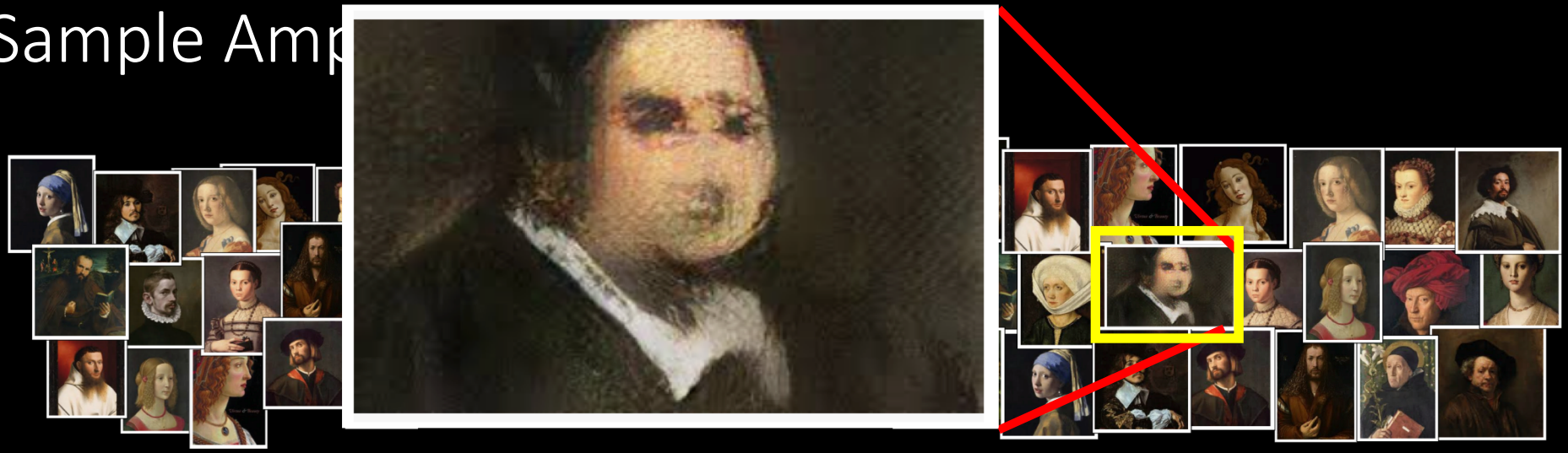
Input: *m* datapoints, distribution *D*

Verifier

Output: ACCEPT or REJECT

Promise: If input is *m* i.i.d. draws from D, then w. prob > ¾, must ACCEPT.

Sample Amp



Input: *n* i.i.d. samples from D

Output: m > n "samples"

Input: *m* datapoints, distribution *D*

Verifier

Output: ACCEPT or REJECT

Promise: If input is *m* i.i.d. draws from D, then w. prob > ¾, must ACCEPT.

Sample Amplification

Definition: A class of distributions *C* admits (*n*,*m*)-amplification, if there is an *(n,m)* Amplifer s.t. **for all** D ∈ C, any Verifier will ACCEPT with prob > 2/3.

Input: *n* i.i.d. samples from D

Output: m > n "samples"

Input: *m* datapoints, distribution *D*

Verifier

Output: ACCEPT or REJECT

Promise: If input is *m* i.i.d. draws from D, then w. prob > ¾, must ACCEPT.

Sample Amplification

Definition: A class of distributions *C* admits (*n*,*m*)-amplification, if there is an *(n,m)* Amplifer s.t. **for all** D ∈ C, any Verifier will ACCEPT with prob > 2/3.

- 2-player setup similar to pseudo-randomness setting, though here, Verifier is computationally unbounded (and knows D)
- Every class C admits (n,n)-amplification (how/why?)
- Verifier does not see Amplifier's *n* input samples. (Otherwise equivalent to *learning*)
- Up to constant factors, equivalent to asking whether Amplifier can output *m* samples, whose T.V. distance to *m* i.i.d. samples from D is small.

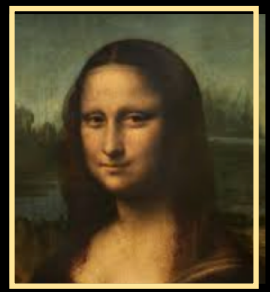
Thm 1: Let *C* be class of Gaussians in *d* dimensions, with fixed covariance (e.g. "isotropic"), and **unknown** mean:

- Nontrivial amplification possible starting at *n = sqrt(d).*
- [Learning to constant TV distance requires n = d]

(*n, n* + n/sqrt(d))-amplification is possible (and optimal, to constant factors)

Thm 2: If output ⊃ input samples, require n > d/ log d for nontrivial amp.

Intuitively, issue is new "samples" would be too correlated with originals:



New Algorithm:

1) Draw $x_{n+1}…x_m$ from noisy version of empirical mean u* of input samples
2) For each input sample x_i "decorrelate" it from u*
3) Return list $x_{n+1}…x_m$ and "decorrelated" original samples.

Thm 3: Let C be class of discrete distributions supported on $\leq k$ elements.
(n, n + n/sqrt(k))-amplification is possible (and optimal, to constant factors)

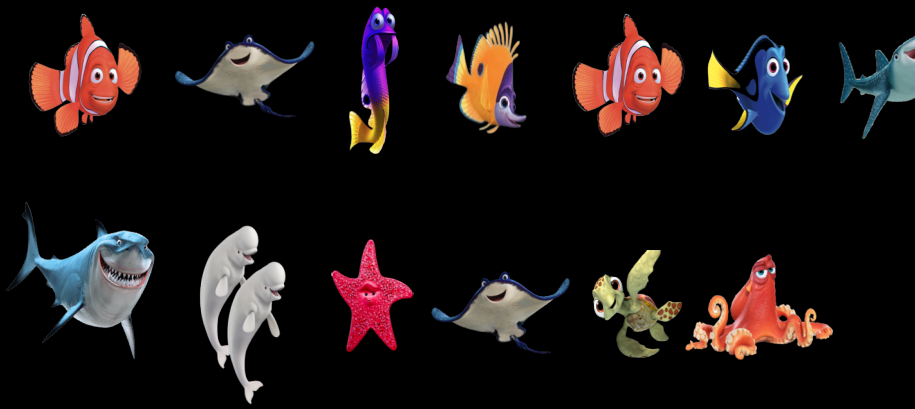
Algorithm*:

1) Draw $x_{n+1}...x_m$ from empirical distribution of n input samples
2) Return random permutation of set of n input samples, and $x_{n+1}...x_m$

*morally…but need some extra stuff to enable the proof…

Intuition: Consider D = Uniform(unknown k items), n = c * sqrt(k)

Sanity Check: Birthday paradox
"Expect $\approx c^2 \pm$ c duplicates….whp above algorithm will add an extra copy of c elements.



What property of a class of distributions determines threshold at which non-trivial amplification is possible?

MORE powerful Verifier?

How much does Verifier need to know about n input samples to preclude amplification without learning? [How much do we need to know about a GAN's input, to evaluate its output?]

LESS powerful Verifier?

What if Verifier doesn't know D*, only gets sample access?*

Part III
Selective Prediction
Accurately predicting the future, with NO assumptions

- Mingda Qiao, Gregory Valiant, *Selective Prediction,* COLT'19.

Mingda Qiao

Non-robust learning/estimation:

Strong distributional assumptions on data
(e.g. data drawn i.i.d., exchangeable, etc.)

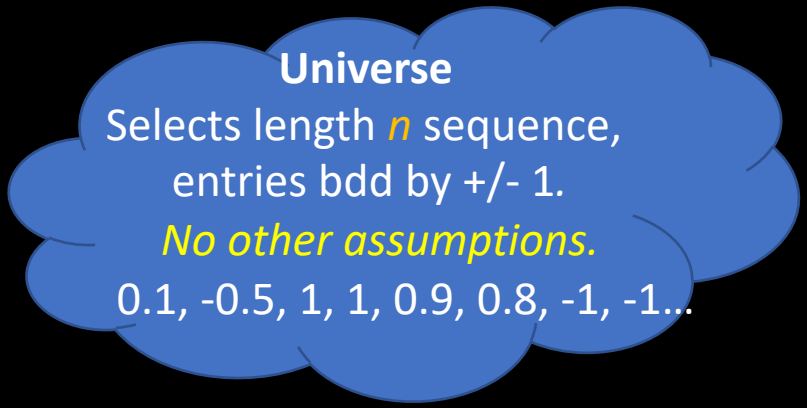
Robust learning/estimation:

a portion of the

Strong distributional assumptions on data
(e.g. data drawn i.i.d., exchangeable, etc.)

In what settings can we perform accurate prediction/estimation/learning with worst-case data (i.e. no assumptions)?

Accurately Predicting the Future, without Assumptions



You

Start observing sequence

0.1, -0.5, 1, 1, …

At *random* time $t \leftarrow [1,2,..,n]$ must make prediction about future:

"Average of next w=37 values will be 0.78"

After prediction is made, game ends, Universe reveals sequence, and measures prediction error.

- Time at which prediction is made is uniformly random
- We choose time horizon of prediction, *w*

Thm [Drucker'13, Qiao,V.'19]: Exists strategy s.t. expected squared prediction error is O(1/log n) which is optimal (for worst-case Universe).

Comparison with Online Learning

Online Learning Formulation

- Predictor has access to a group of experts.
- Each day, make a prediction based on experts' advice.
- In the end, compare performance to the best expert.

- Well-known result: as $T \to +\infty$, among the first T days,
  $$[\text{Avg. Loss}] \leq [\text{Avg. Loss of Best Expert}] + o(1)$$

Our Setting:

- Time at which prediction is made chosen at random
- Power to choose time horizon, *w* of prediction
- Compare loss wrt ground truth (not expert benchmark)

Algorithm and Proof of Proposition

Claim: If $n=2^k$ and whole sequence has average value v then Alg achieves expected squared error

$$L(k,v) \leq 4\, v(1-v)/k$$

Algorithm:

- Draw $c \leftarrow [0,1,2,\ldots,\log n - 1]$
- Set $w = 2^c$
- Draw $t \leftarrow \{w, 3w, 5w, 7w,\ldots,n-w\}$
- Predict $u = \text{average}(x_{t-w+1},\ldots,x_t)$

Proof by induction on k:

Base case k=1: $L(1,v) = \sup_{x_1,x_2 \text{ in } [0,1] \text{ s.t. } (x_1+x_2)/2=v} (x_1-x_2)^2 = \min(4v^2, 4(1-v)^2) \leq 4v(1-v)$

Induction step: let x_1,x_2 be averages of first and second halves of sequence.

$$L(k,v) \leq \sup_{x_1,x_2 \text{ in } [0,1] \text{ s.t. } (x_1+x_2)/2=v} \left[(1/k)(x_1-x_2)^2 + (k-1)/k \quad (1/2)(L(k-1,x_1)+L(k-1,x_2)) \right]$$

Prob. of picking m=n/2

Apply induction hypothesis to this!!!

Lower bound proof idea

Thm: There exists a distribution over length *n* binary sequences s.t. no algorithm can choose *t, w,* and *u* which estimates *average*($x_{t+1},\ldots,x_{t+w}$) with expected squared error better than O(1/log n).

Intuition: Design sequence that is simultaneously anticoncentrated at all timescales, *w.*

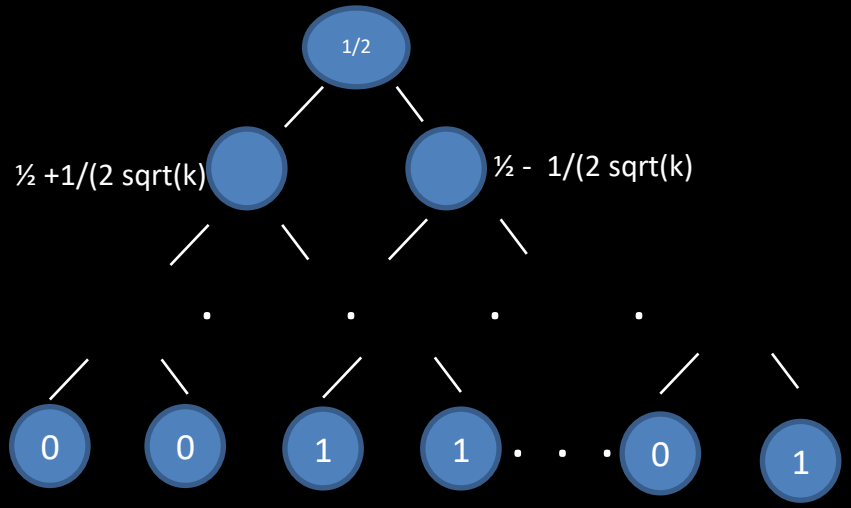
Lower bound proof idea

Thm: There exists a distribution over length *n* binary sequences s.t. no algorithm can choose *t, w,* and *u* which estimates *average*($x_{t+1},\ldots,x_{t+w}$) with expected squared error better than O(1/log n).

Idea: Make binary tree with $n=2^k$ leaves, which will be $x_1,\ldots,x_n$

Nodes at level *j* have values
1/2 +/- sqrt(j)/(2sqrt(k))

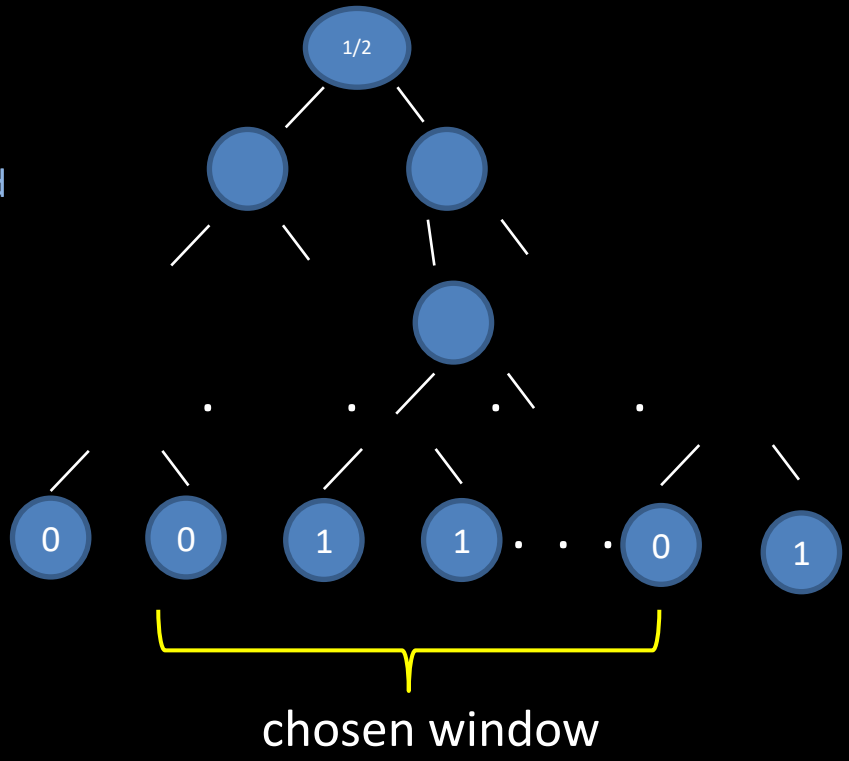
Each node chooses value s.t.
expected value = value-of-parent



Lower bound proof idea

Nodes at level *j* have values
1/2 +/- sqrt(j)/(2sqrt(k))

Each node chooses value s.t. expected value = value-of-parent



Lower bound proof idea

Nodes at level *j* have values
1/2 +/- sqrt(j)/(2sqrt(k))

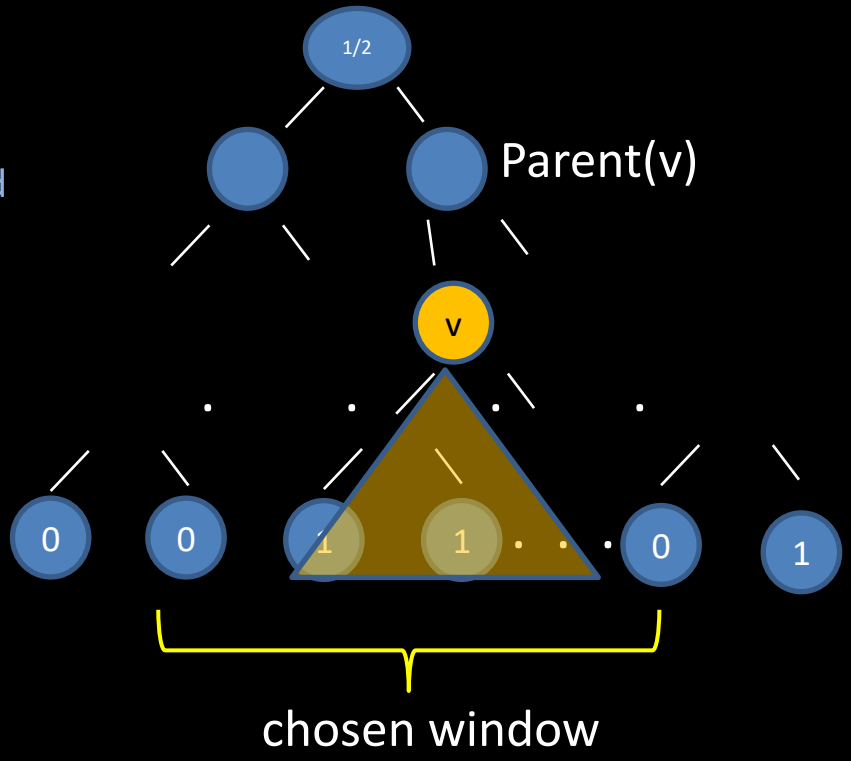
Each node chooses value s.t. expected value = value-of-parent

Find node whose children are:

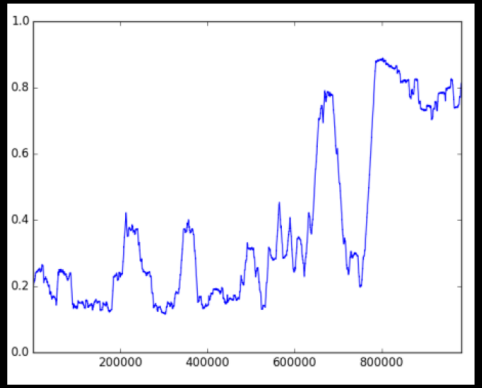
- Subset of chosen window
- Constant fraction of window

Claim: Variance due to *v* and *v's* children, even after conditioning on ALL other nodes in the tree, is O(1/k)=O(1/log n).

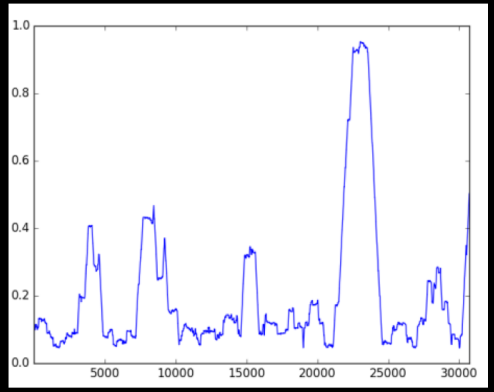
Proof Idea: Var[v|parent(v)]=O(1/k)…



Lower bound proof idea



w=2^15



w=2^10

e.g. construction with k=20,
Plots show moving average over windows of length w=2^10, and m=2^15….
Anti-concentrated simultaneously at both timescales!!

Nodes at level *j* have values
1/2 +/- sqrt(j)/(2sqrt(k))

Each node chooses value s.t.
expected value = value-of-parent