Bayes-Optimal Methods for Optimization via Simulation:

The Probabilistic Bisection Algorithm

Rolf Waeber, Peter I. Frazier, Shane G. Henderson

Operations Research & Information Engineering, Cornell University

Research supported by:
Air Force Office of Scientific Research (AFOSR)
National Science Foundation (NSF)

STOR-i Workshop
January 9-10 2014
Lancaster University

I am interested in Bayesian sequential experimental design

Many of the problems I'm interested in have this form:

- There is some unknown function $f : \mathbb{X} \mapsto \mathbb{Y}$.
- We have a Bayesian prior distribution on f.
- We want to learn something about f, such as:
  - Optimization: Find a point x such that $f(x)$ is large.
  - Level-set estimation: Find $\{x : f(x) > a\}$, for some constant a.
  - Sensitivity analysis: Find i such that $|\partial f(x)/\partial x_i|$ is large.
- We have the ability to evaluate $f(x)$, for some x's that we may choose in a sequential fashion.
- Evaluating f is really expensive, so we can't do it too many times.
- How should we choose our x's?

Example: Website optimization at Yelp

yelp

Find tacos, cheap dinner, Max's | Near Lancaster, UK

Home | About Me | Write a Review | Find Friends | Messages | Talk

Pubs Lancaster

Showing 1-10 of 43

Businesses > Nightlife > Bars > Pubs

Show Filters

1. The Borough Lancaster

8 reviews

£££ · Restaurants, Pubs

3-5 Dalton Square
Lancaster LA1 1PP
UK
+44 1524 64170

My friend chose this restaurant for an impromptu birthday meal, just because she had passed it and it looked nice we had no idea what was in store for us! Let me start with the service we...

2. Merchants

7 reviews

££ · Pubs

27 Castle Hill
Lancaster LA1 1YN
UK
+44 1524 66466

I really love the Merchants. The story I've been told is that someone was excavating a basement or something and suddenly realized they were standing above 3 huge wine cellars (dating back to...

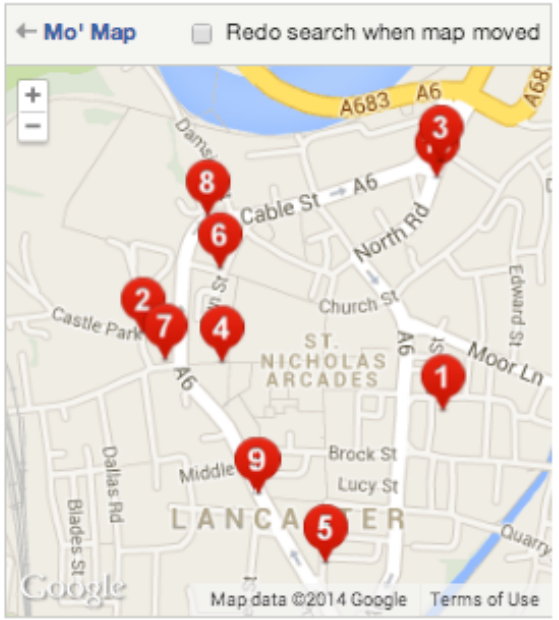
3. The Yorkshire House

4 reviews

££ · Pubs

2 Parliament Street
Lancaster LA1 1DB
UK
+44 1524 64679

As stated by others a superb alternative ale house. Dark and scary just the way the customers like it. Vey helpful bar staff to advise on the intruiging and ever changing ale range. Watch out...



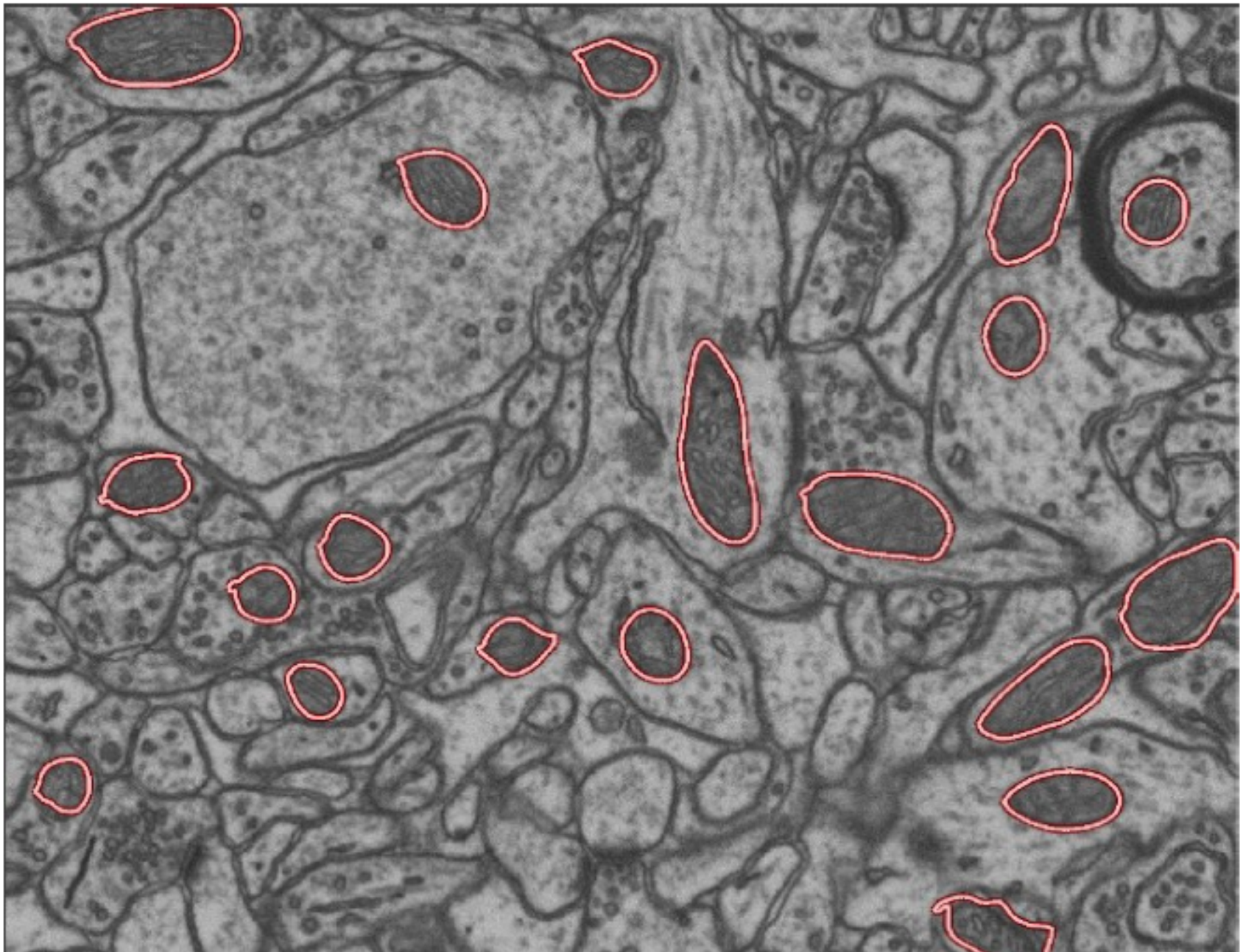
Ads by Google related to: **Pubs Lancaster**

50% Off **Lancaster** Hotels
www.laterooms.com/**Lancaster**-Hotels
32 4* hotels all within 1 mile of **Lancaster**. Book today
745,019 people follow LateRooms.com on Google+

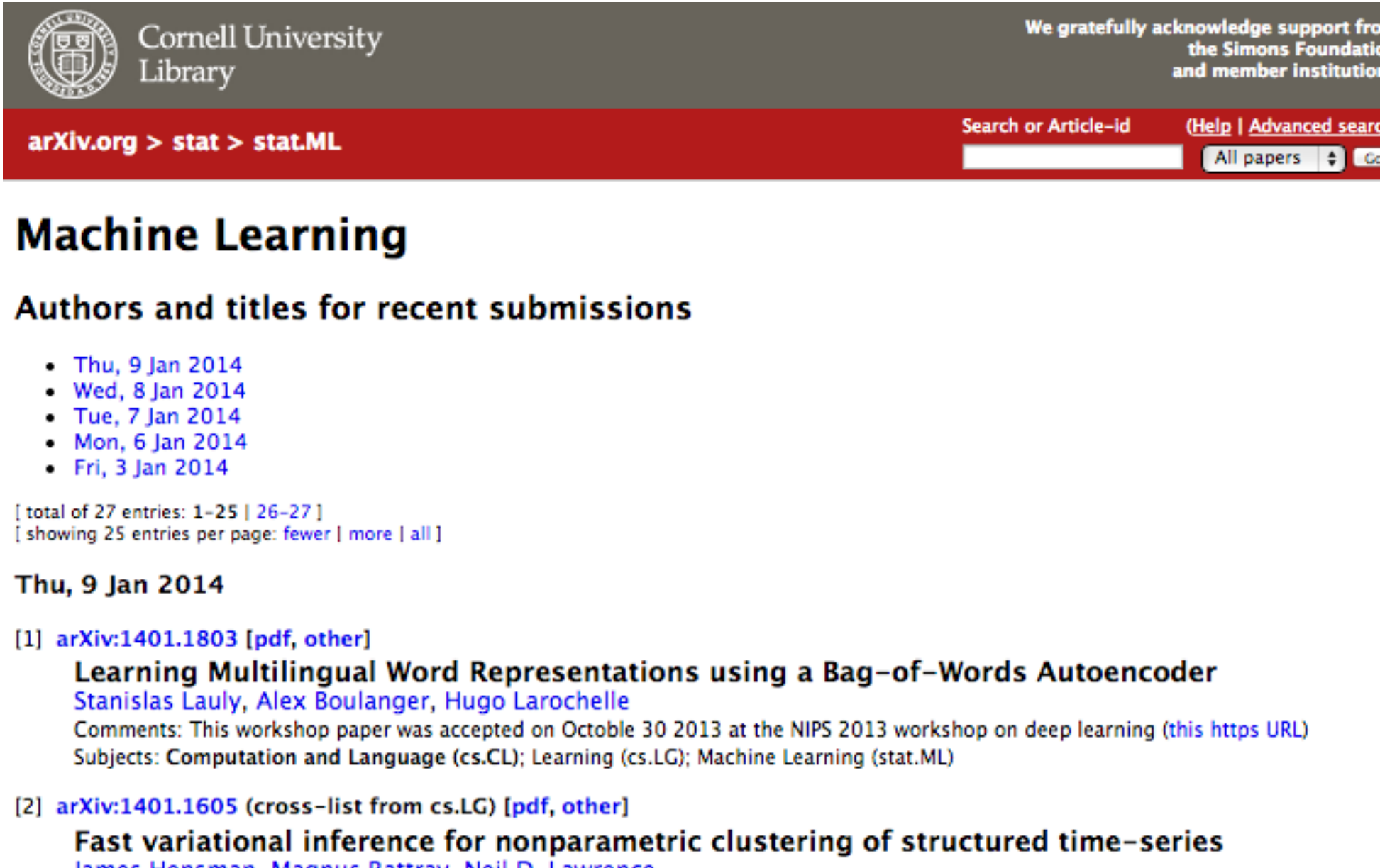
Luxury Hotels | Discount Hotels
Budget Hotels | Best Reviewed Hotels

Lancaster Pub Restaurants

Example: Detecting edges in computer vision



Example: Learning user preferences at arxiv.org



Cornell University Library

We gratefully acknowledge support fro
the Simons Foundatio
and member institutio

arXiv.org > stat > stat.ML

Search or Article-id (Help | Advanced searc

All papers

Machine Learning

Authors and titles for recent submissions

- Thu, 9 Jan 2014
- Wed, 8 Jan 2014
- Tue, 7 Jan 2014
- Mon, 6 Jan 2014
- Fri, 3 Jan 2014

[total of 27 entries: 1-25 | 26-27]
[showing 25 entries per page: fewer | more | all]

Thu, 9 Jan 2014

[1] arXiv:1401.1803 [pdf, other]

Learning Multilingual Word Representations using a Bag-of-Words Autoencoder

Stanislas Lauly, Alex Boulanger, Hugo Larochelle

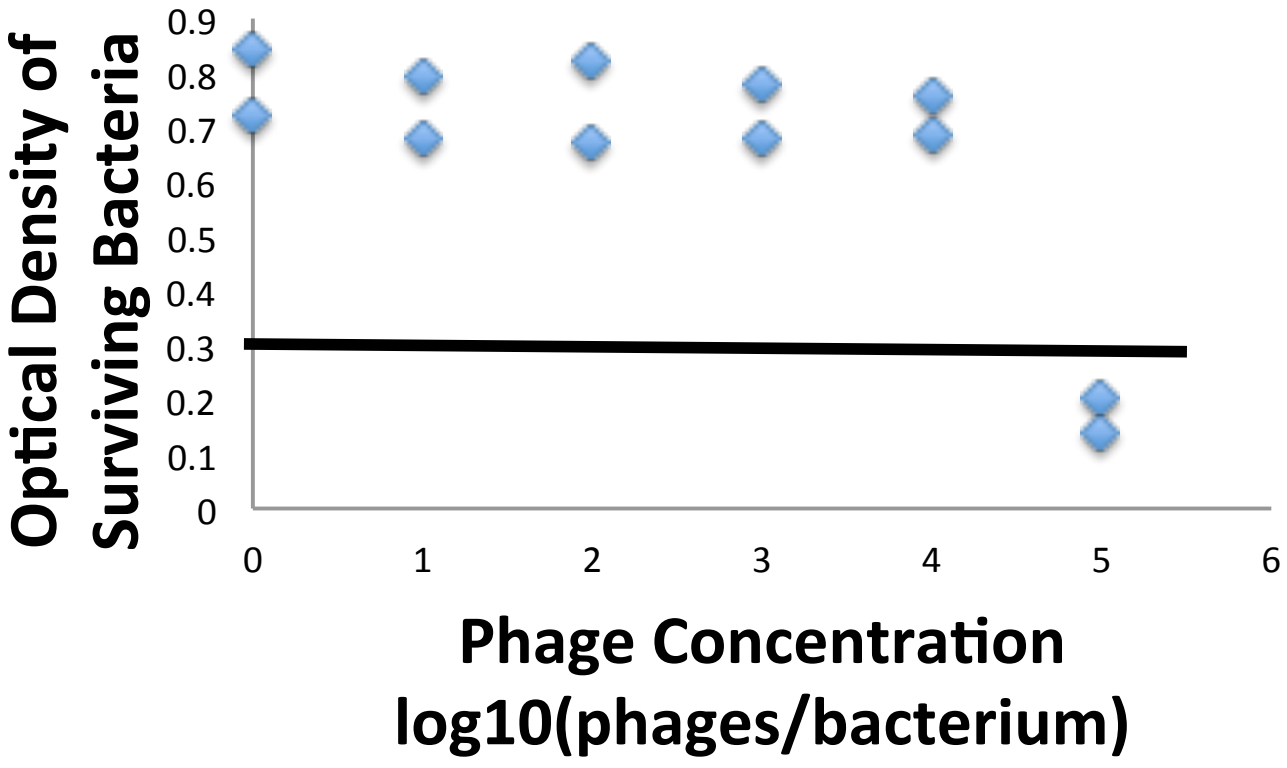
Comments: This workshop paper was accepted on Octoble 30 2013 at the NIPS 2013 workshop on deep learning (this https URL)

Subjects: **Computation and Language (cs.CL)**; Learning (cs.LG); Machine Learning (stat.ML)

[2] arXiv:1401.1605 (cross-list from cs.LG) [pdf, other]

Fast variational inference for nonparametric clustering of structured time-series

Example: Root-finding experiments in biology



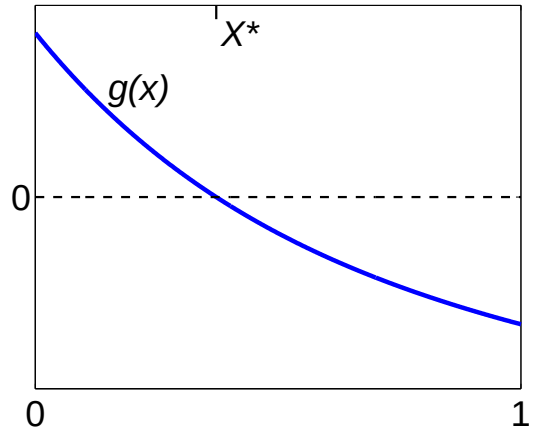
These problems are all partially observable Markov decision processes

In these problems:

- The value function gives the value-to-go of the optimal policy, as a function of the current posterior.
- If we knew the value function, we could figure out the optimal measurement to take next.
- In principal, the value function can be computed via dynamic programming.
- In practice, the curse of dimensionality usually prevents us from solving the dynamic program.

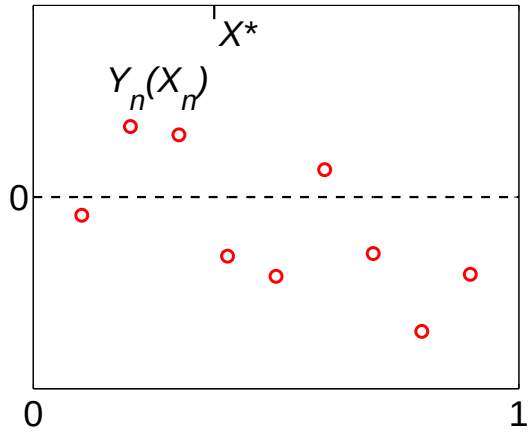
In this talk, we will consider one problem in which we can circumvent the curse of dimensionality, and efficiently compute the optimal policy.

We consider the stochastic root-finding problem



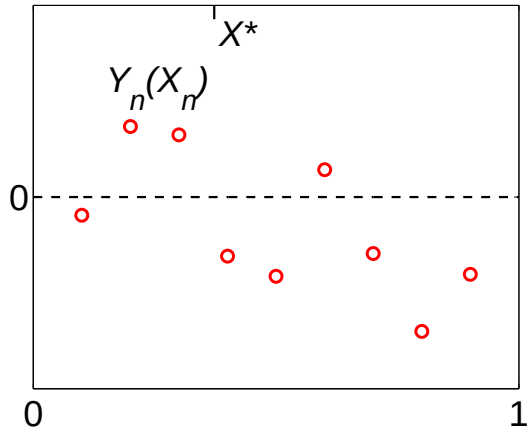
- Consider a function $g : [0,1] \to \mathbb{R}$.
- Suppose there is a unique $X^* \in [0,1]$ such that
  - $g(x) > 0$ for $x < X^*$,
  - $g(x) < 0$ for $x > X^*$.
- Our **goal** is to find $X^* \in [0,1]$.

We consider the stochastic root-finding problem



- Consider a function $g : [0,1] \to \mathbb{R}$.
- Suppose there is a unique $X^* \in [0,1]$ such that
  - $g(x) > 0$ for $x < X^*$,
  - $g(x) < 0$ for $x > X^*$.
- Our **goal** is to find $X^* \in [0,1]$.
- We can only observe $Y_n(X_n) = g(X_n) + \varepsilon_n(X_n)$, where $\varepsilon_n(X_n)$ is conditionally independent noise with zero mean (median).

We consider the stochastic root-finding problem

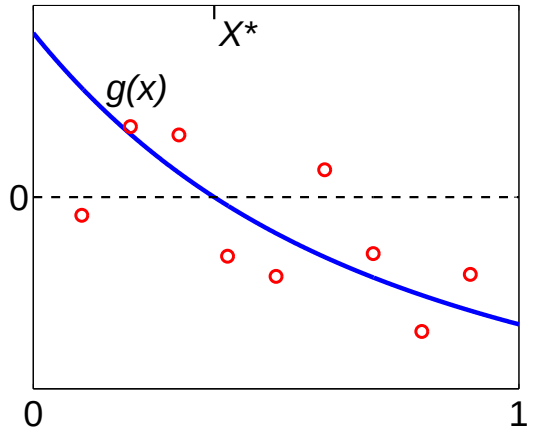


- Consider a function $g : [0,1] \to \mathbb{R}$.
- Suppose there is a unique $X^* \in [0,1]$ such that
  - $g(x) > 0$ for $x < X^*$,
  - $g(x) < 0$ for $x > X^*$.
- Our **goal** is to find $X^* \in [0,1]$.
- We can only observe $Y_n(X_n) = g(X_n) + \varepsilon_n(X_n)$, where $\varepsilon_n(X_n)$ is conditionally independent noise with zero mean (median).
- We must **decide:**
  - Where to place samples X_n for $n = 0, 1, 2, \ldots$ to best support estimation of X^* after n iterations.

This problem, or a multivariate version of it, arises in many applications

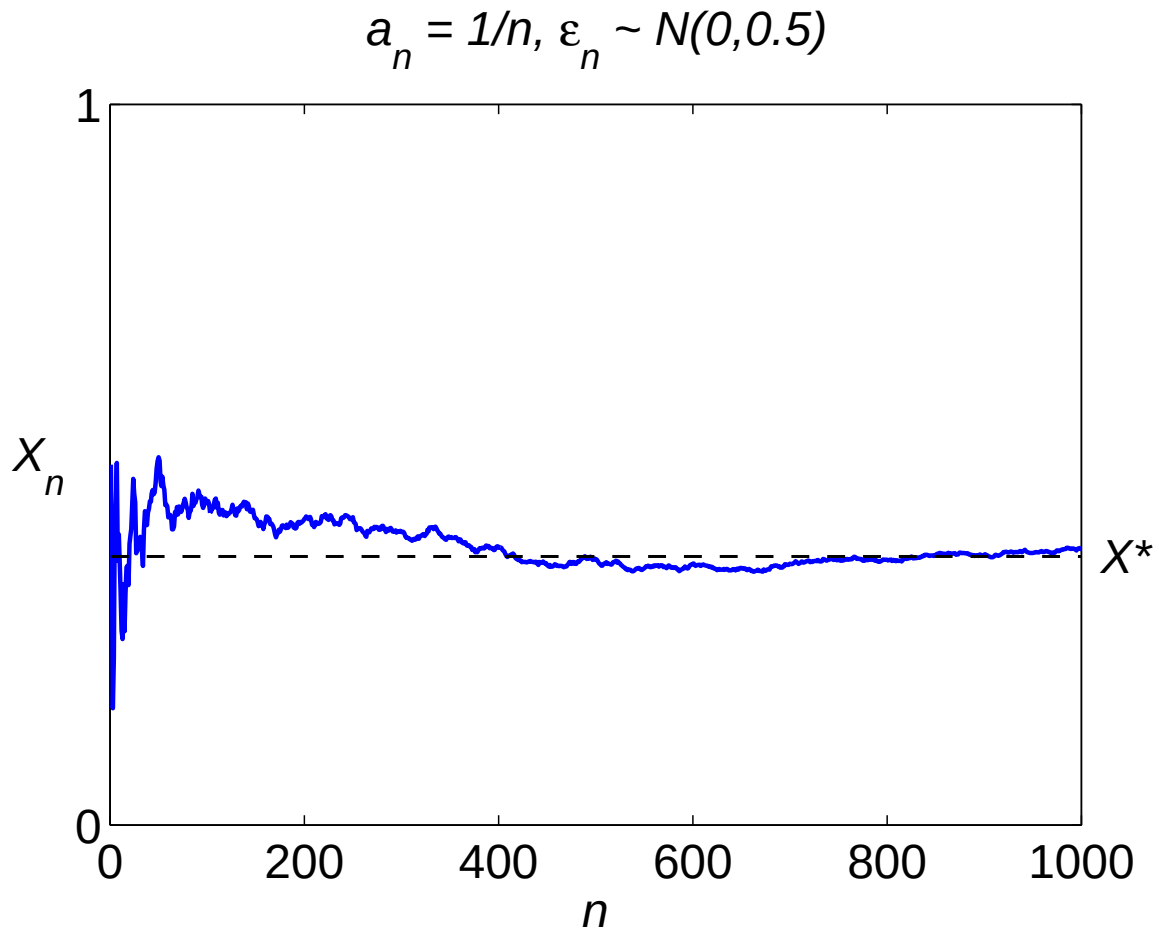
- Optimization via simulation
  - Maximizing a concave function via noisy observations of its gradient.
- Medicine, Biology, Chemistry:
  - Estimating the median effective dose (ED50), the median lethal dose (LD50), the half maximal inhibitory concentration (IC50), ...
- Machine Learning:
  - Regression with big datasets
  - Other kinds of parameter estimation problems with big datasets
- Computer vision:
  - Edge detection
  - Object detection and tracking
  - Scanning electron microscopy

Stochastic approximation is a standard method for solving this problem, due to [Robbins and Monro, 1951]



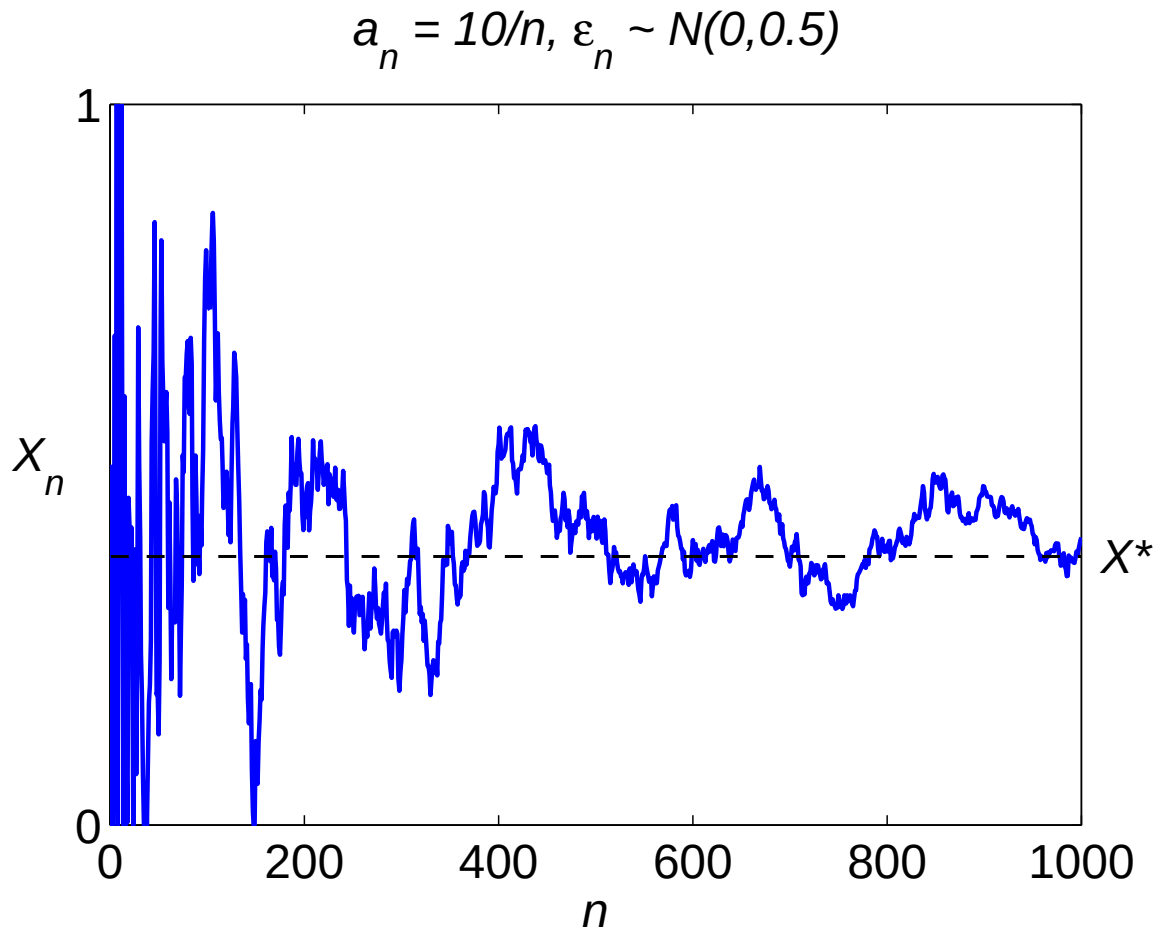
1. Choose an initial estimate $X_0 \in [0,1]$;
2. Select a tuning sequence $(a_n)_n \geq 0$, $\sum_{n=0}^{\infty} a_n^2 < \infty$, and $\sum_{n=0}^{\infty} a_n = \infty$. (Example: $a_n = d/n$ for $d > 0$.)
3. $X_{n+1} = \Pi_{[0,1]}(X_n + a_n Y_n(X_n))$, where $\Pi_{[0,1]}$ is the projection to $[0,1]$.

Stochastic approximation works well when the tuning sequence is well-chosen



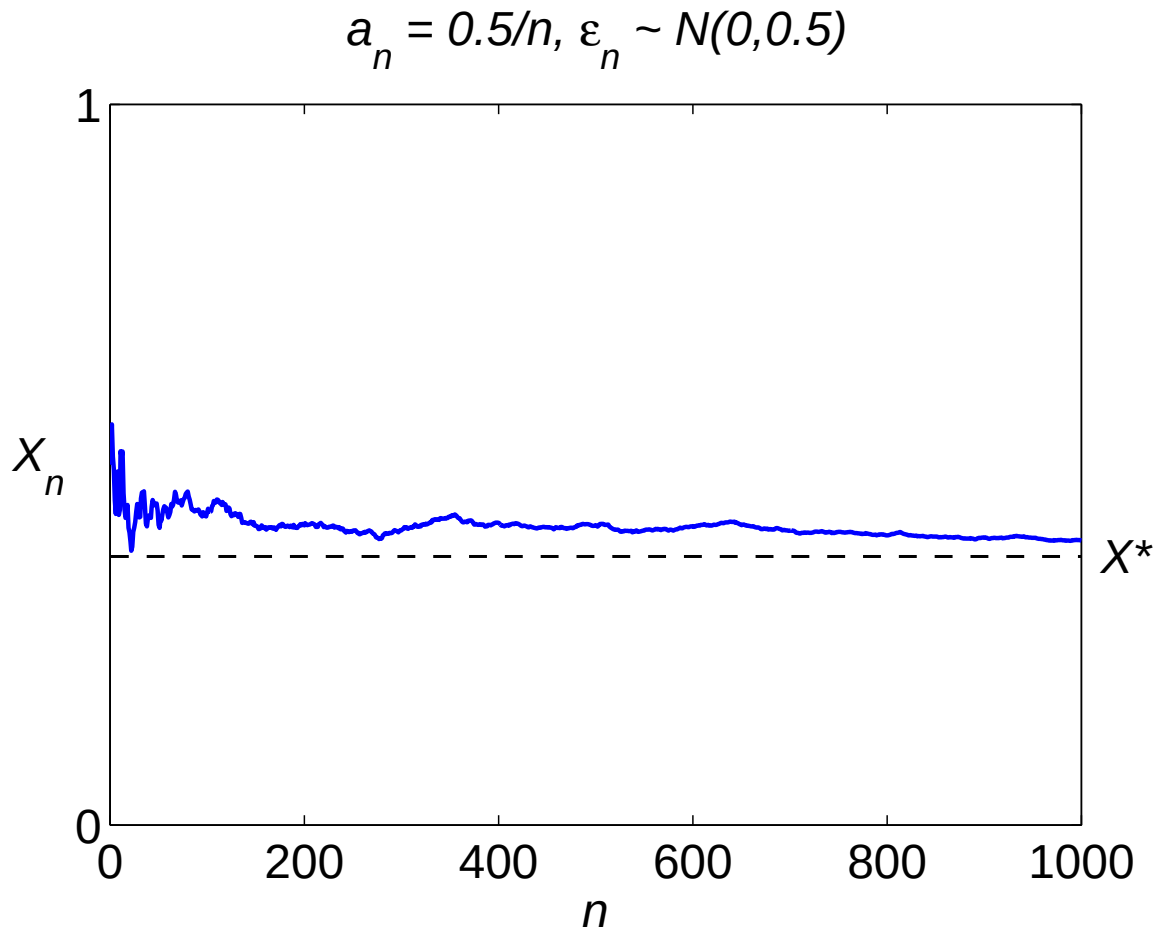
This shows stochastic approximation when the tuning sequence is well-chosen.

Stochastic approximation works poorly when the tuning sequence is poorly chosen



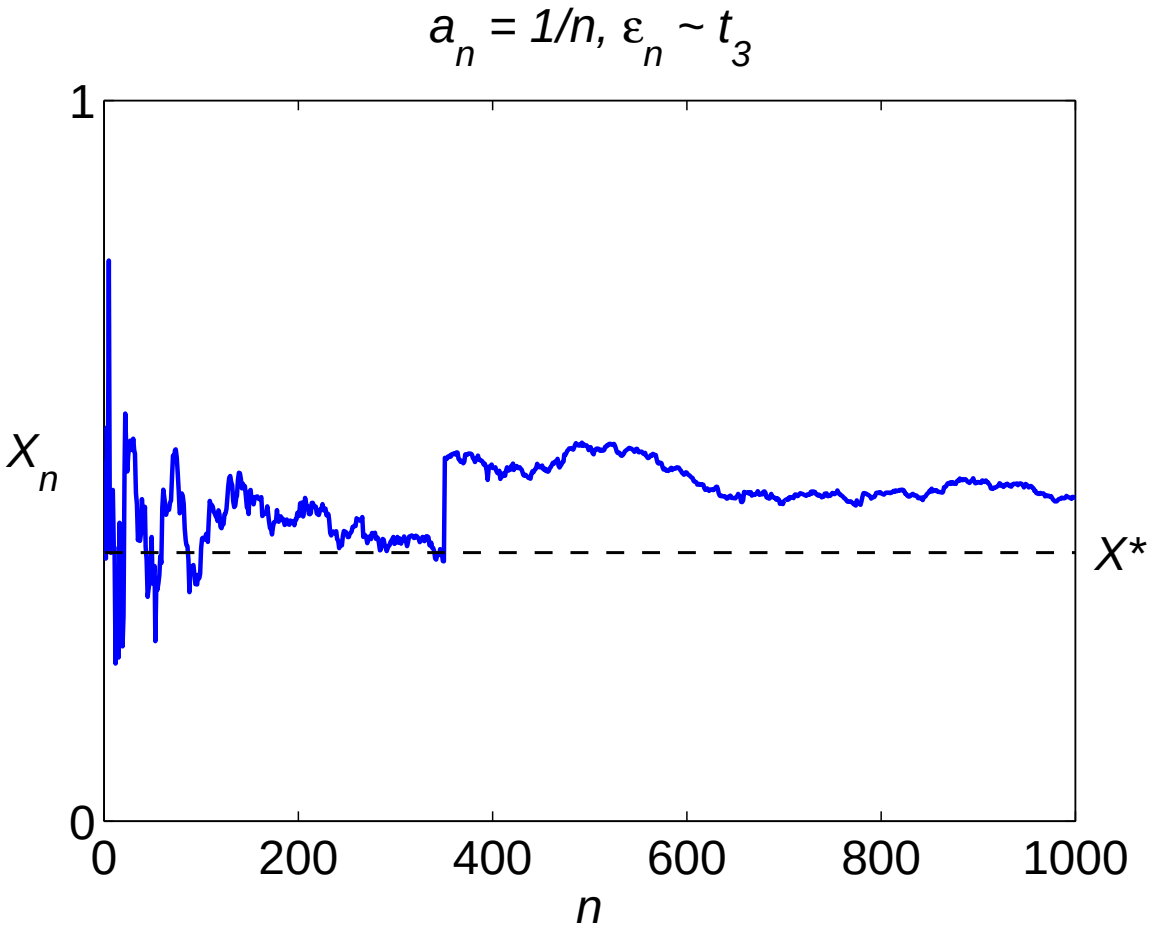
This shows stochastic approximation when the tuning sequence is too large.

Stochastic approximation works poorly when the tuning sequence is poorly chosen



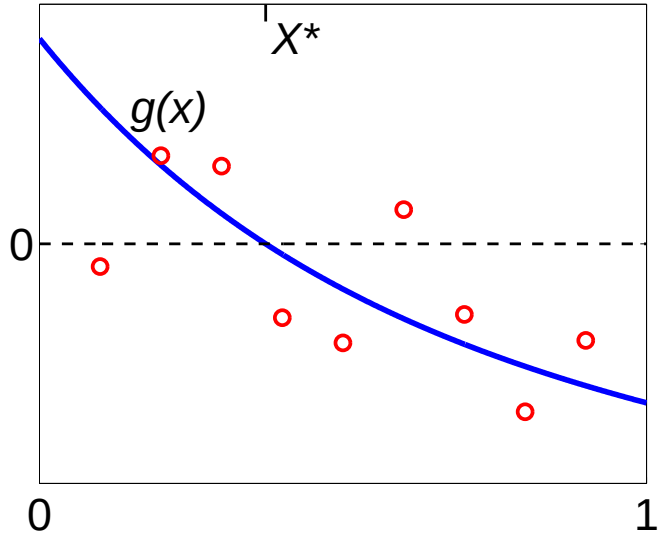
This shows stochastic approximation when the tuning sequence is too small.

Stochastic approximation can work poorly when the noisy has heavy tails



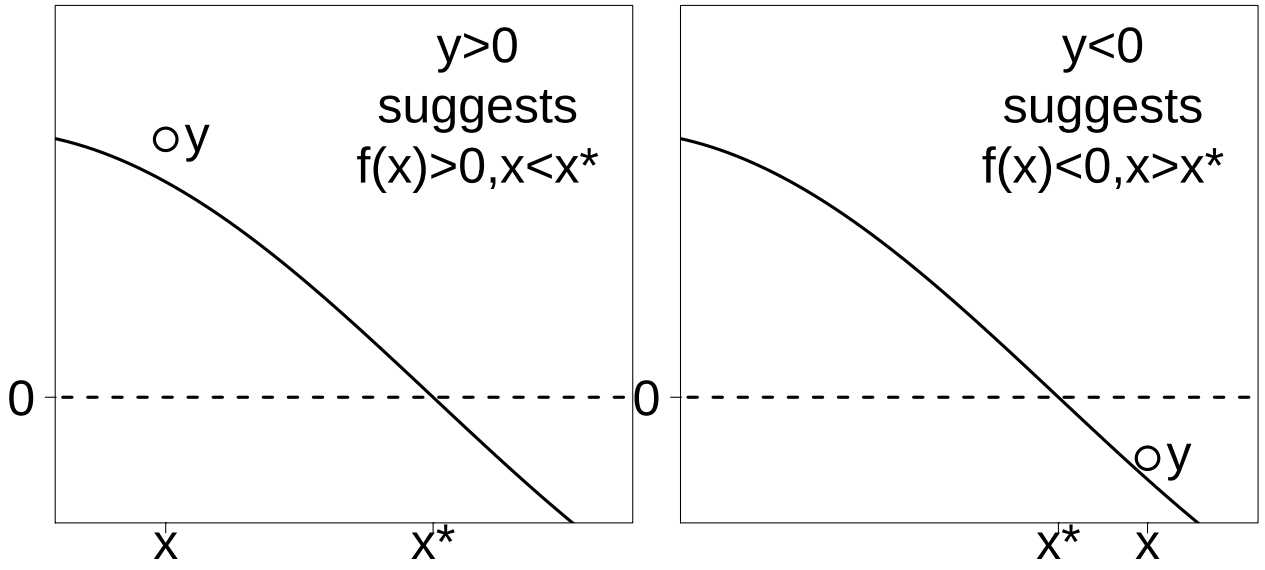
We consider a different approach

What about a bisection algorithm?



- Noise will cause the deterministic bisection algorithm to fail.
- We develop a noise-tolerant version of the bisection algorithm, using a dynamic programming analysis of a stylized version of the problem.
- This new method will be better than stochastic approximation in some situations, and worse in others.

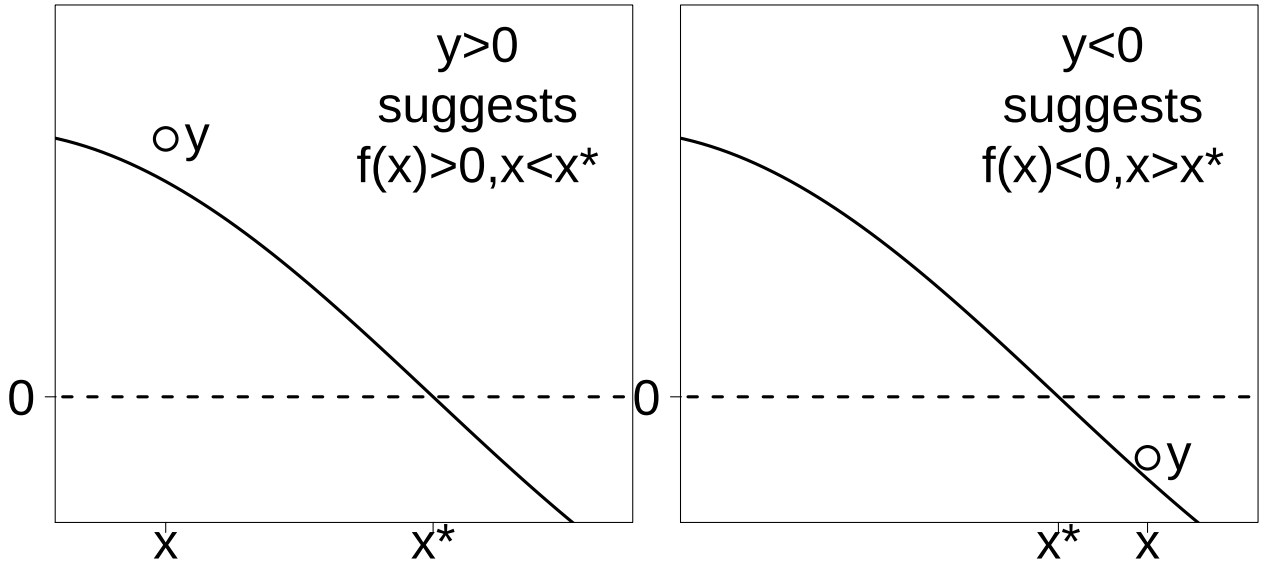
To do bisection, we will use "left/right" observations



By observing one or more samples $Y_n(x)$, at a fixed x, we can obtain (noisy) information about:

- the sign of $g(x)$; and
- whether X^* is to the left or right of x.

We can get noisy "left/right" observations



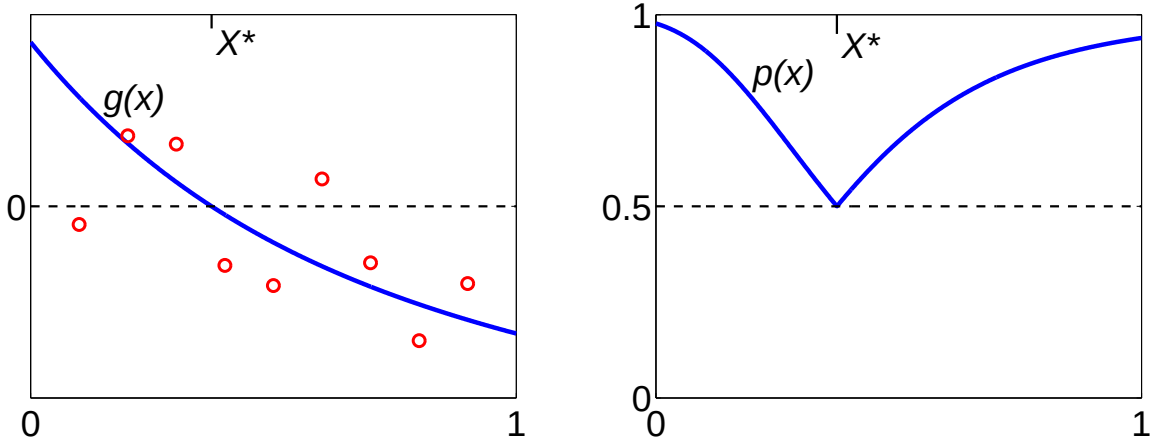
Let $Z_n(X_n)$ be such a noisy left/right observation, with the following properties:

$$Z_n(X_n) = \begin{cases} \text{sign}\,(g(X_n)) & \text{with probability } p(X_n), \\ -\text{sign}\,(g(X_n)) & \text{with probability } 1 - p(X_n), \end{cases}$$

for some function $p(\cdot)$, which may, or may not, be known.

Here is the simplest way to construct a noisy left/right observation

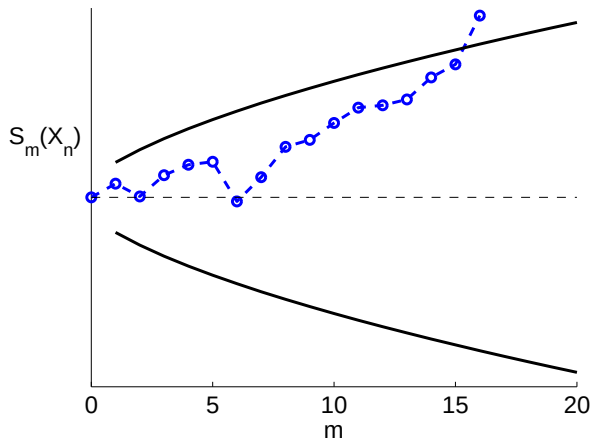
- We could take $Z_n(X_n) = \text{sign}(Y_n(X_n))$.
- This construction requires only a single observation to construct Z_n.
- Under this construction, with homoscedastic normal noise and $g(\cdot)$ as given below left, $p(\cdot)$ is given below right:



- When constructed in this way, $p(\cdot)$ is generally unknown, and $p(x)$ may be arbitrarily close to $1/2$.

There are other ways to construct a noisy left/right observation

- We can also construct $Z_n(X_n)$ by sampling repeatedly at a single point X_n.
- In particular, we can sample sequentially using an α-level test of power 1 [Siegmund, 1985] for the mean of $Y_n(x)$.



- Under a parametric assumption (additive normal noise, or Bernoulli observations), this ensures $p(x) \geq p_c$ for all $x \neq X^*$.
- Here, $p_c > 1/2$ is a known constant that we choose when we design the α-level test.

Here is the program for the rest of the talk

First, we consider a stylized setting where $p(\cdot)$ is constant and known.

- We perform a Bayesian analysis, and find the optimal policy using dynamic programming.
- This Bayes-optimal policy was first proposed in 1964 by Horstein, though it was not known to be Bayes-optimal.

Second, we create an algorithm that can be used in practice.

- This algorithm allows $p(\cdot)$ to be unknown, but requires it to be bounded below by a known constant, $p_c > 1/2$.
- This assumption is met using sequential sampling, as previously noted.
- We show consistency, how to construct a confidence interval, and find frequentist rates of convergence.

Outline

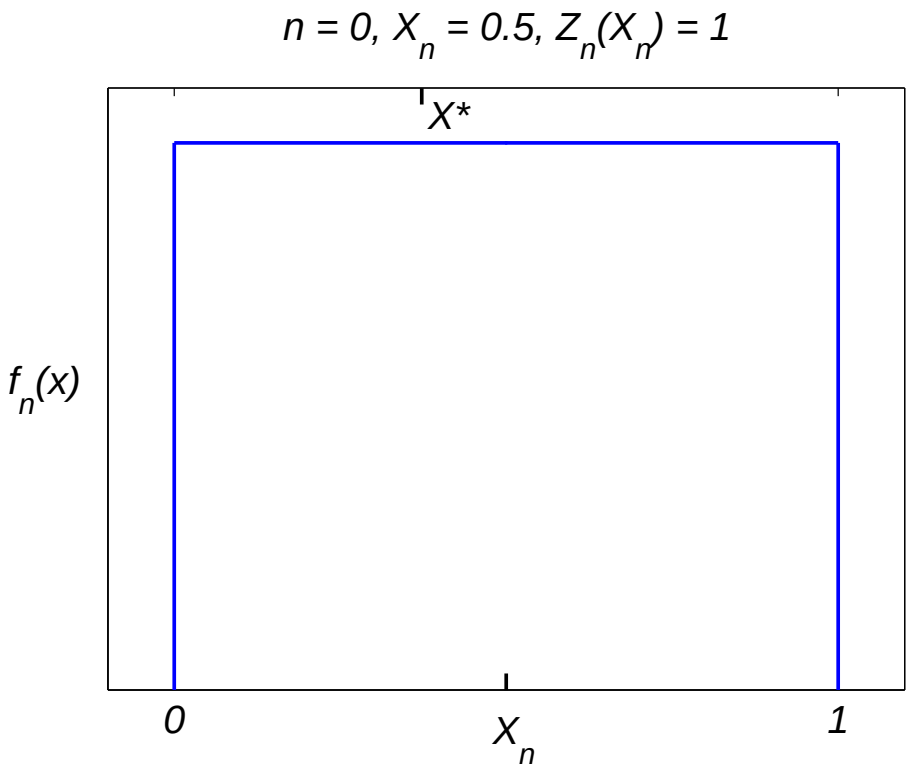
1. Stylized Setting: $p(\cdot)$ is known and constant

2. Practical Setting: $p(\cdot)$ is unknown, and bounded below by $p_c > 1/2$.

We first consider a stylized setting

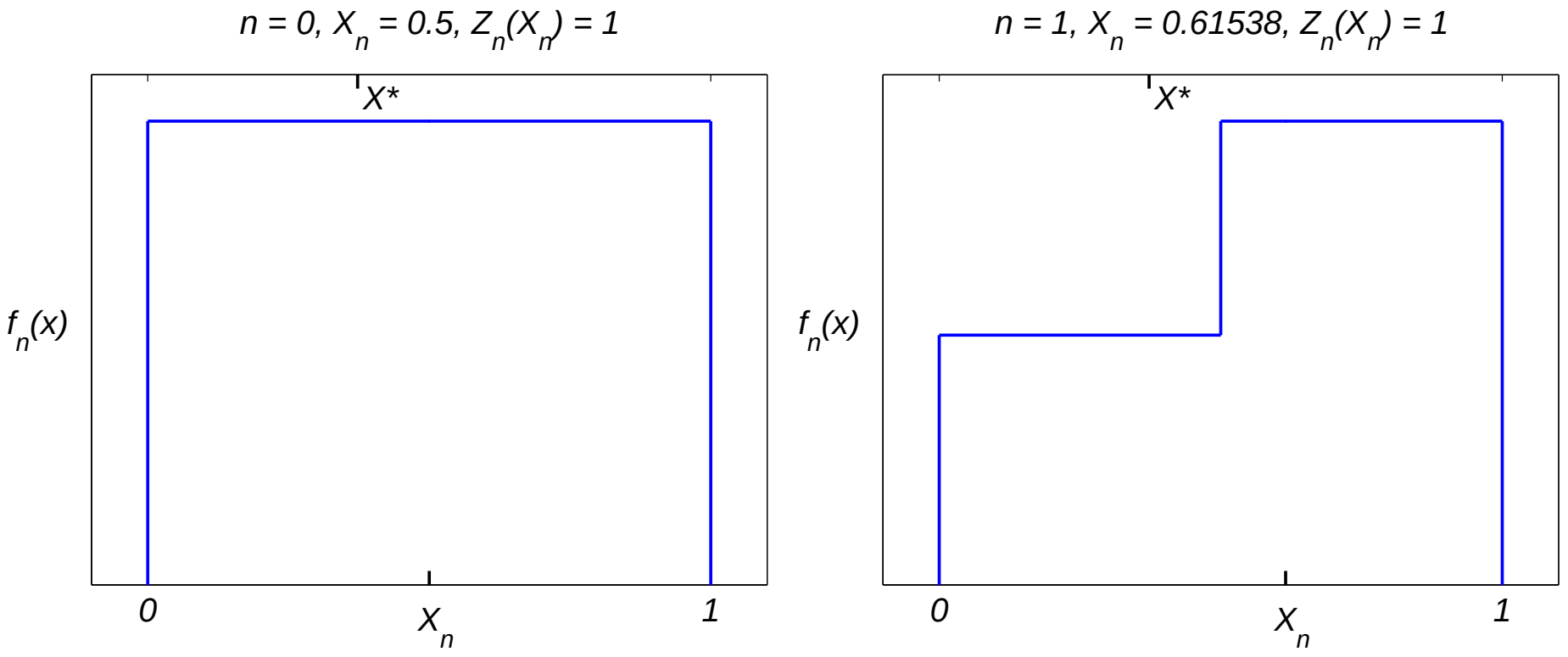
- Recall that $p(X_n)$ is the probability that our noisy left/right observation $Z_n(X_n)$ is correct.
- For the moment, assume that $p(\cdot)$ is constant, known, and strictly greater than $1/2$.
- This assumption is not generally met in practice.

We perform a Bayesian analysis



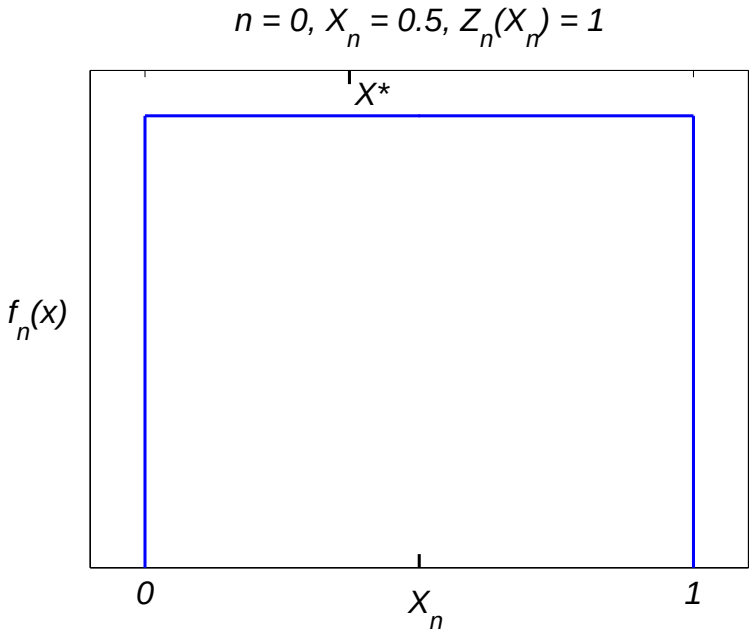
- We place a Bayesian prior density f_0 on the root X^*, e.g., Uniform($[0,1]$).

We perform a Bayesian analysis

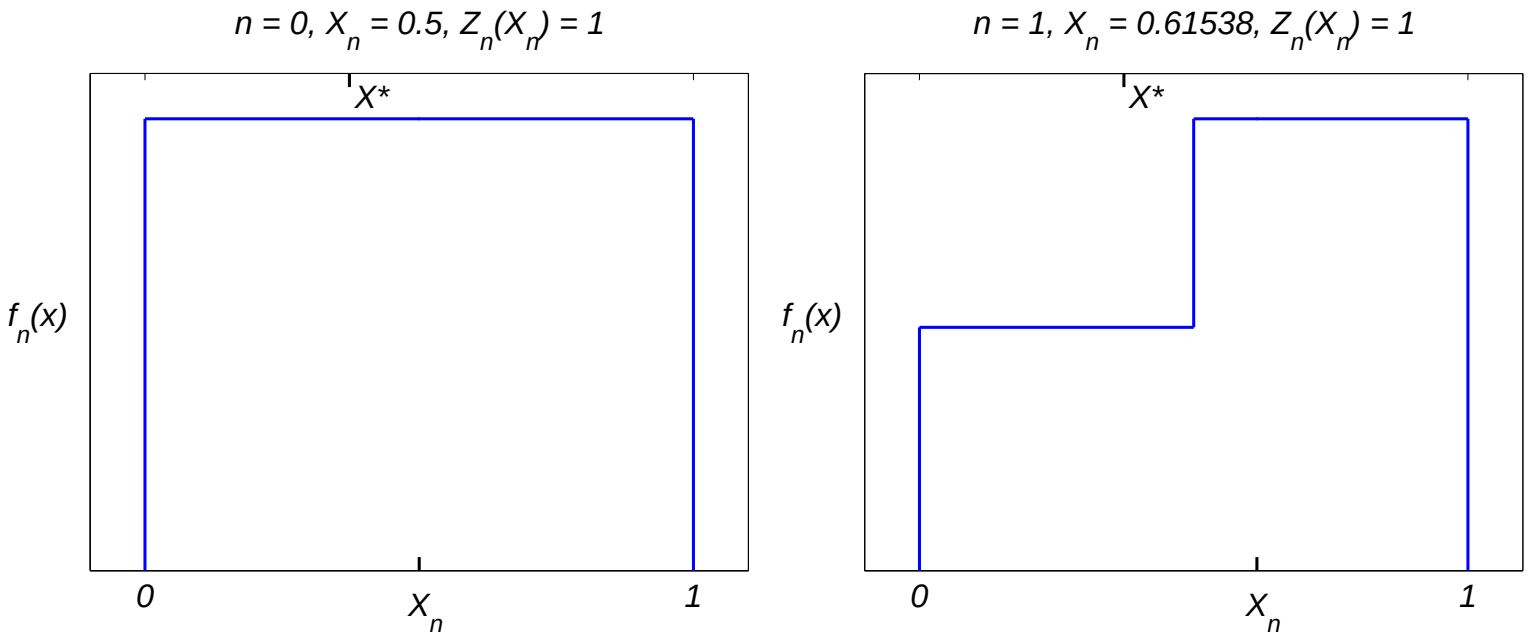


- We place a Bayesian prior density f_0 on the root X^*, e.g., Uniform$([0,1])$.
- Then each observation $Z_n(X_n)$ produces a new posterior density f_n on x_*, $f_n(x) = \mathbb{P}\{X^* \in dx \mid X_{1:n}, Z_{1:n}\}$

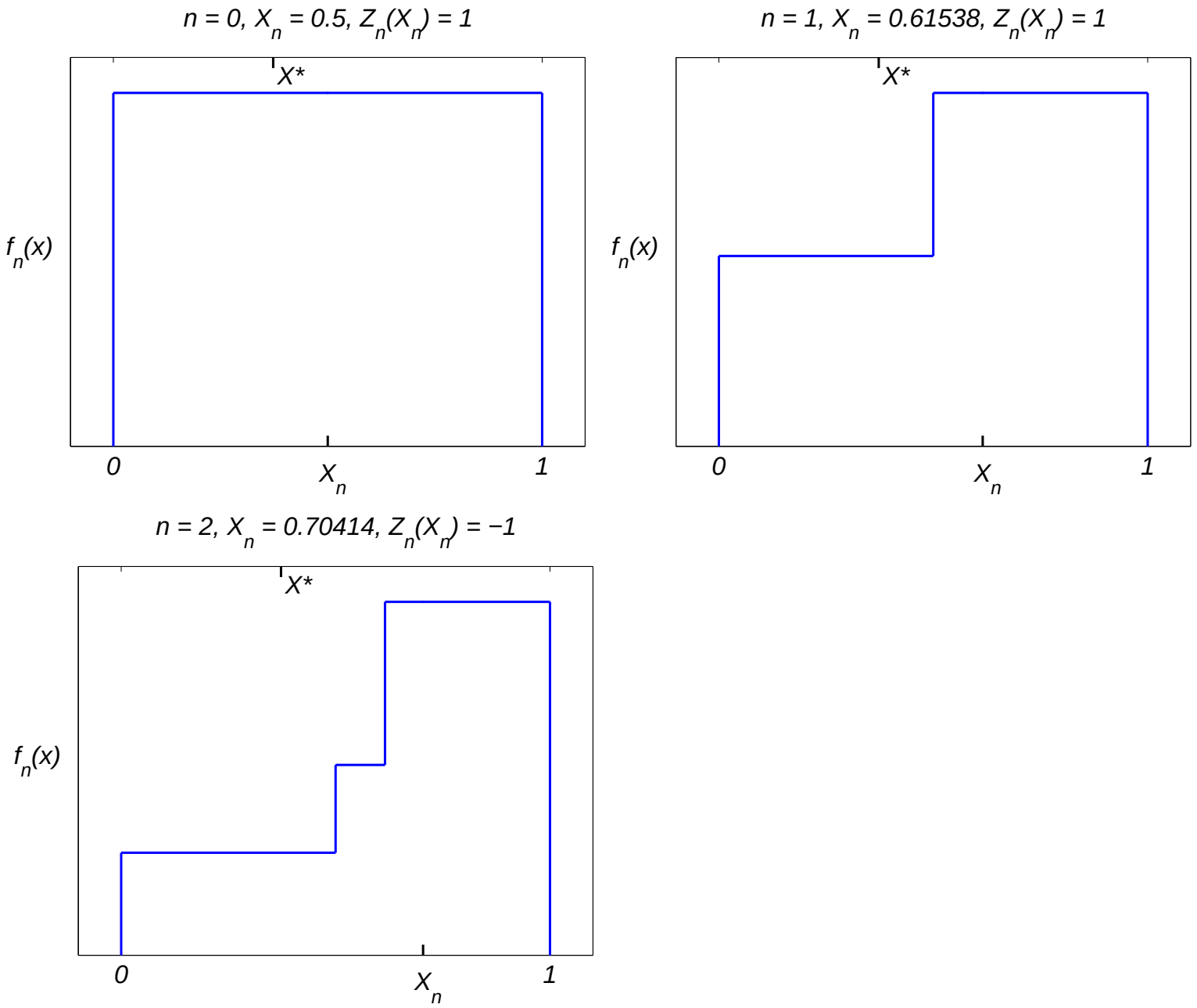
Here is an animation of the sequence of posterior densities



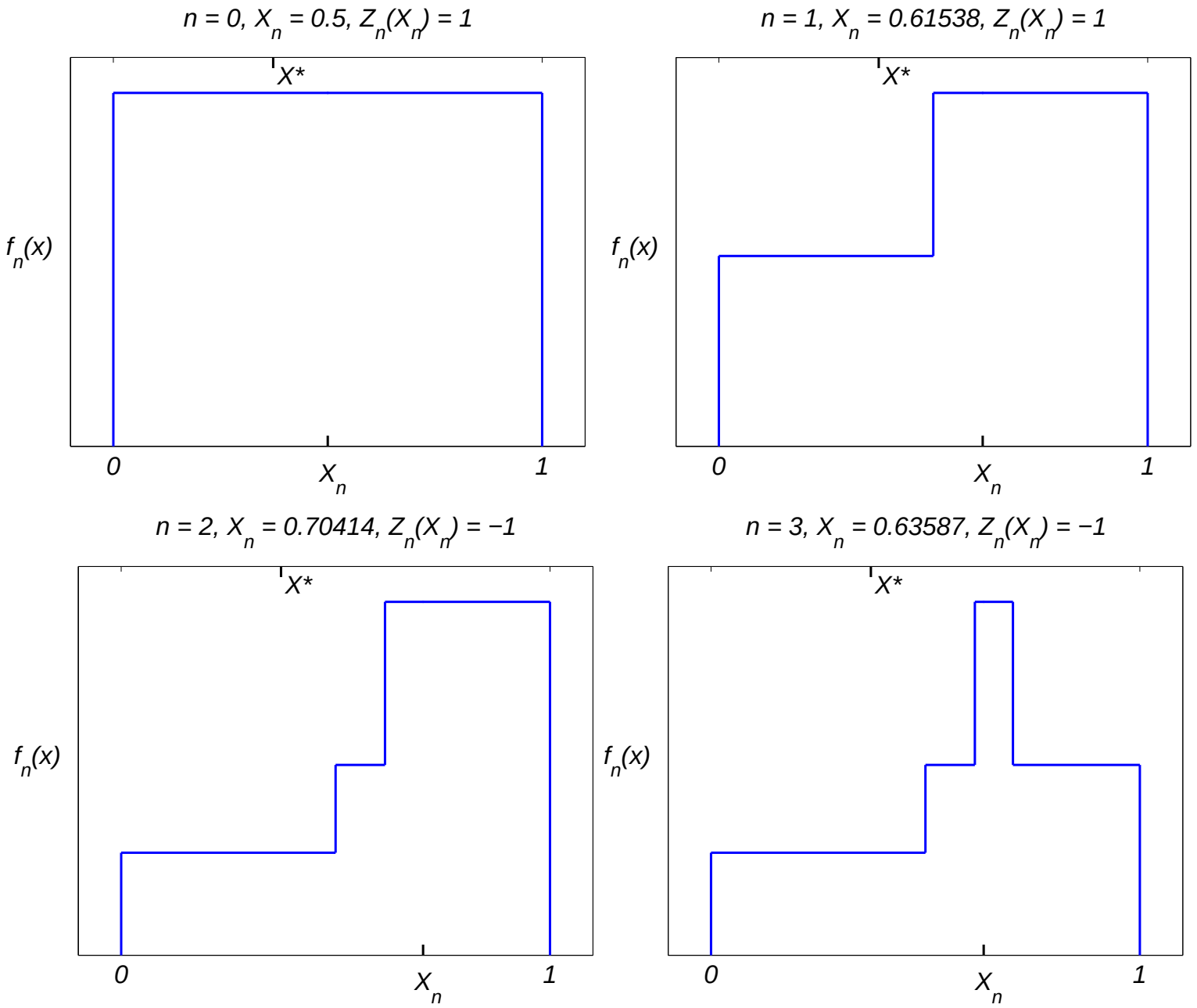
Here is an animation of the sequence of posterior densities



Here is an animation of the sequence of posterior densities



Here is an animation of the sequence of posterior densities



We measure performance by the entropy of the posterior

- We have a finite budget of N measurements.
- After this budget is exhausted, we measure our remaining uncertainty by the **entropy** of the posterior distribution on the root's location,

$$H(f_N) = -\int f_N(x) \log f_N(x)\, dx.$$

- We wish to find an adaptive method for choosing X_n so as to minimize the expected entropy of the posterior.

Our problem is a dynamic programming problem

- We wish to find an adaptive method for choosing x_n so as to minimize the expected entropy of the posterior.

$$\inf_{\pi} \mathbb{E}^{\pi} \left[H(f_N) \right],$$

- This can be formulated as a dynamic programming problem.
- The value function is

$$V_n(f_n) = \inf_{\pi} \mathbb{E}^{\pi} \left[H(f_N) \mid f_n \right].$$

- The value function satisfies the dynamic programming recursion: $V_n(f_n) = \inf_{x_n \in [0,1]} \mathbb{E}\left[V_{n+1}(f_{n+1}) \mid f_n \right]$.

In principle we could solve the dynamic program via brute force...

- We can parameterize the posterior f_n in terms of the points previously measured, $X_1, \ldots, X_n$, and the responses $Z_1, \ldots, Z_n$. (holding f_0 fixed).
- Thus, we can identify f_n with a point in $\mathbb{S}_n = [0,1]^n \times \{0,1\}^n$.
- If we had enough memory on our computer to store $V(f_n)$ for a dense grid of points in $\mathbb{S}_n$, we could solve the problem using dynamic programming:
  - Compute $V_n(f_n)$ (with some small error) using the dynamic programming (DP) recursion:

$$V_n(f_n) = \inf_{x_n \in [0,1]} \mathbb{E}\left[V_{n+1}(f_{n+1}) \mid f_n\right].$$

  - Then compute a near-optimal x_n for any given f_n by finding the value attaining the infimum in the DP recursion.

In practice we do not have the computing power to solve the dynamic program via brute force.

- Our state space at time n is $\mathbb{S}_n = [0,1]^n \times \{0,1\}^n$.
- If we discretize each dimension of $[0,1]^n$ into 100 pieces, our dense grid in $\mathbb{S}_n$ has $100^n \times 2^n$ points.
- If we store each $V_n(f_n)$ in double-precision floating point (64 bits), this would require:
  - 2.4 TB of storage for $n = 5$.
  - 465 TB for $n = 6$.
  - ...
  - 7.5×10^{11} TB for $n = 10$.
- This exponential scaling in the storage requirement is called the **curse of dimensionality**.

Instead, we solve the dynamic program via trickery

Theorem

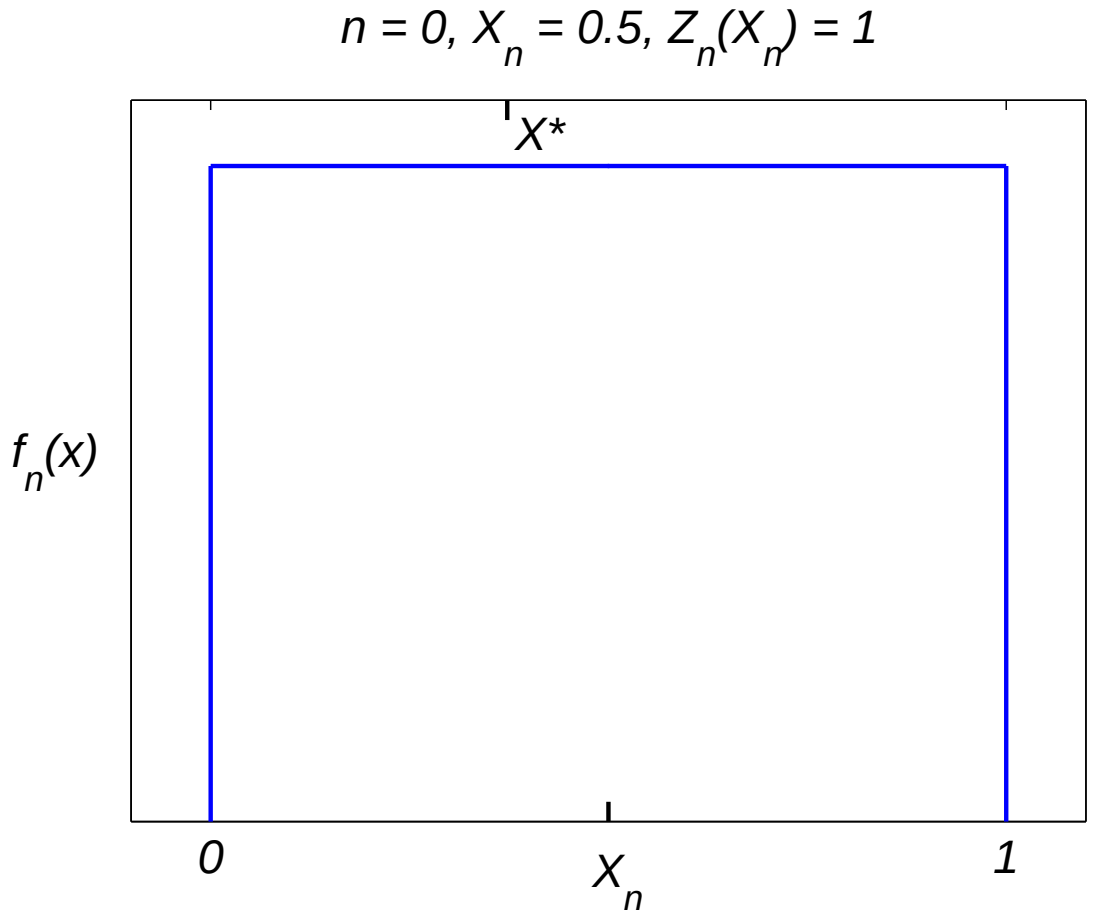
Suppose $p(\cdot) = p$ is constant, known, and bounded away from 1/2, and we use the entropy loss function. Then, the value function can be written explicitly as

$$V(f_n) = H(f_n) - (N-n)\left[-p\log_2(p) - (1-p)\log_2(1-p)\right],$$

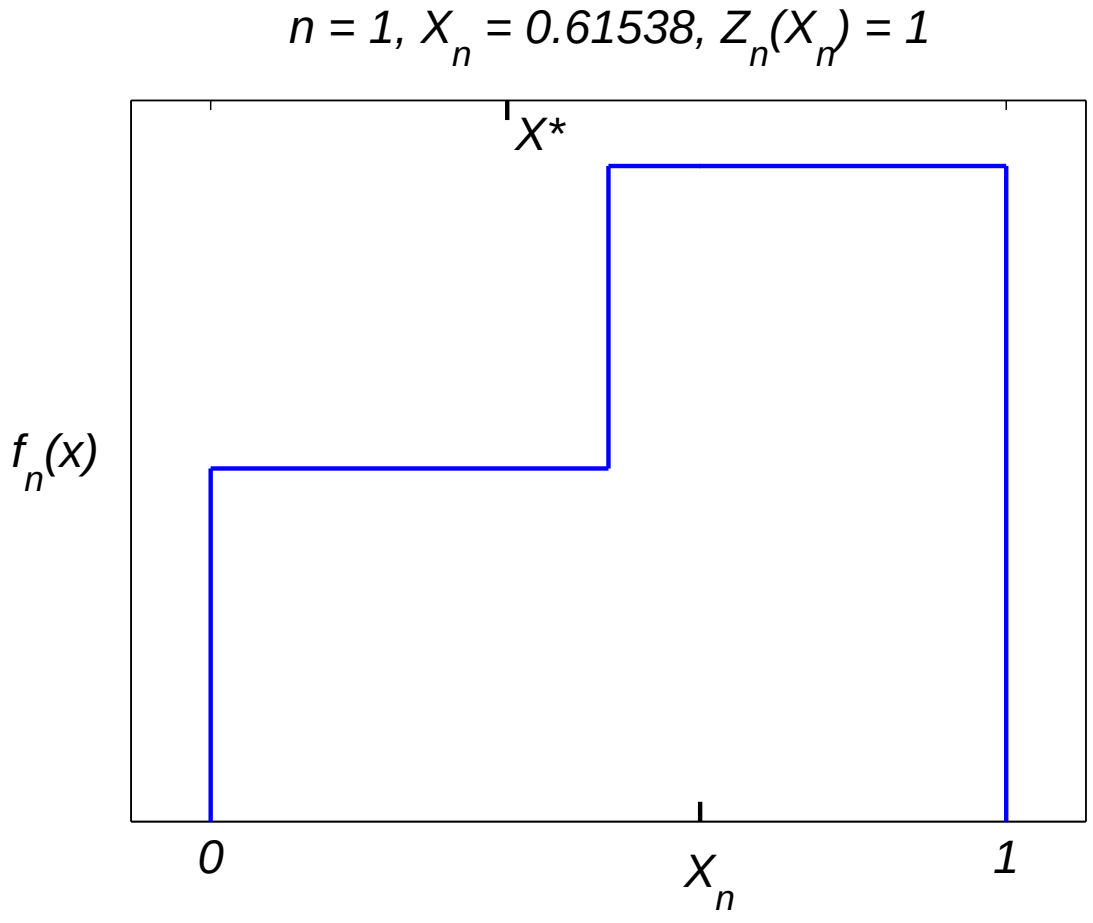
and **the policy that chooses X_n at the median of f_n is optimal**.

This result appears in [Jedynak, Frazier, Sznitman 2011], and is also shown in a more elementary way in [Waeber, Frazier, Henderson 2013].

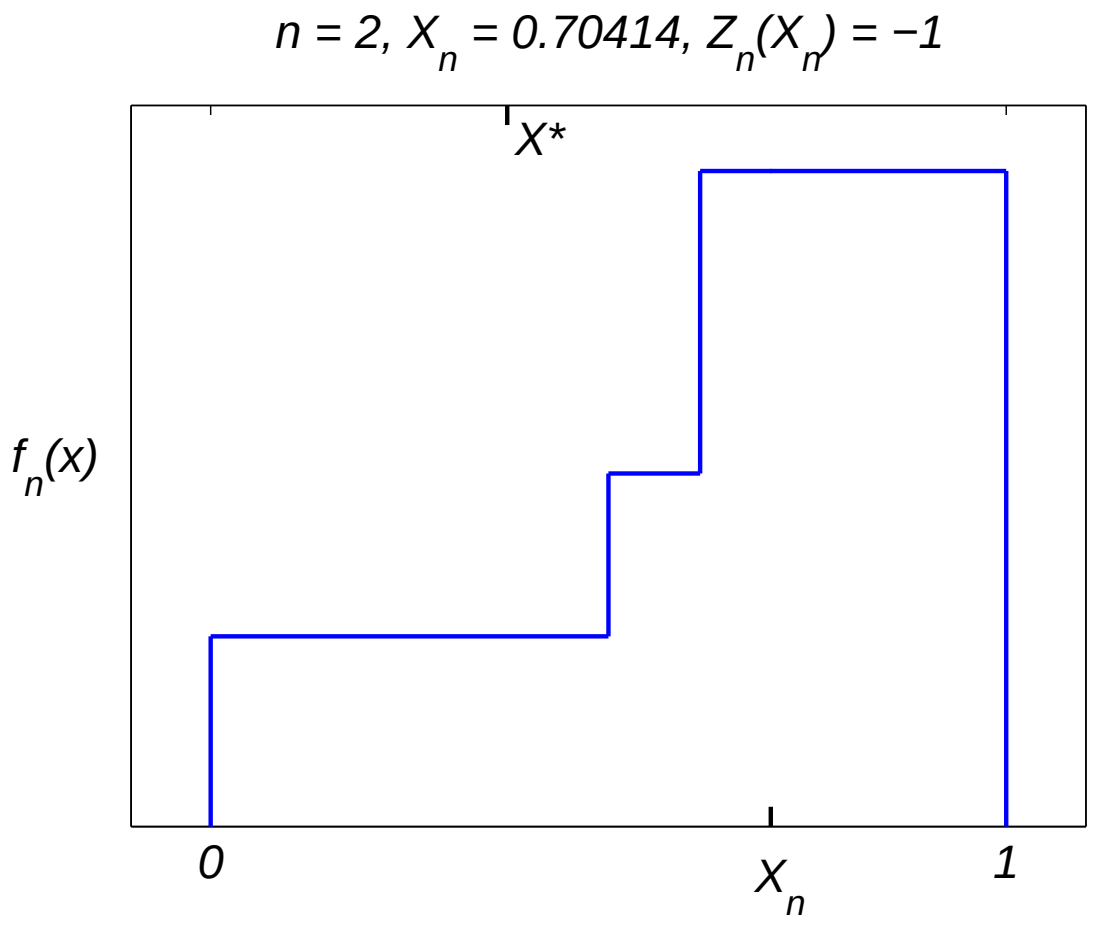
Animation showing the posterior under the optimal policy



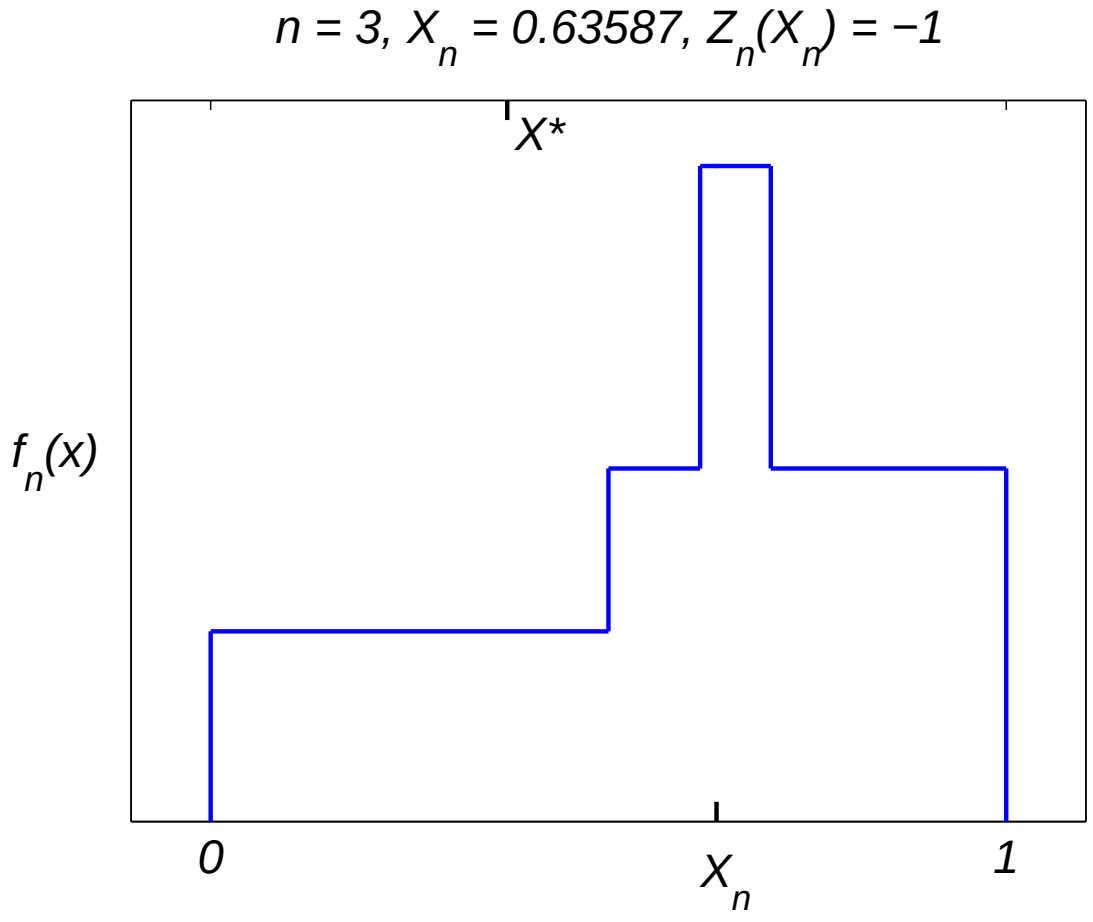
Animation showing the posterior under the optimal policy



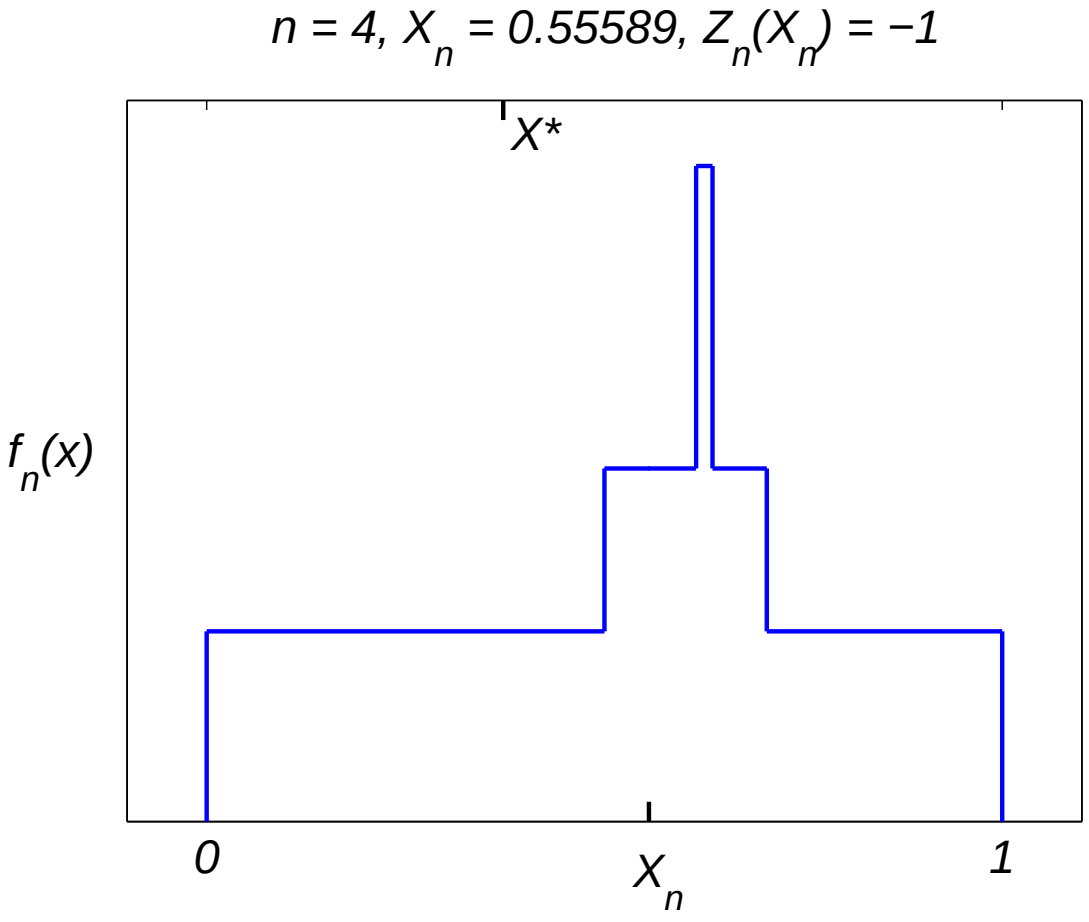
Animation showing the posterior under the optimal policy



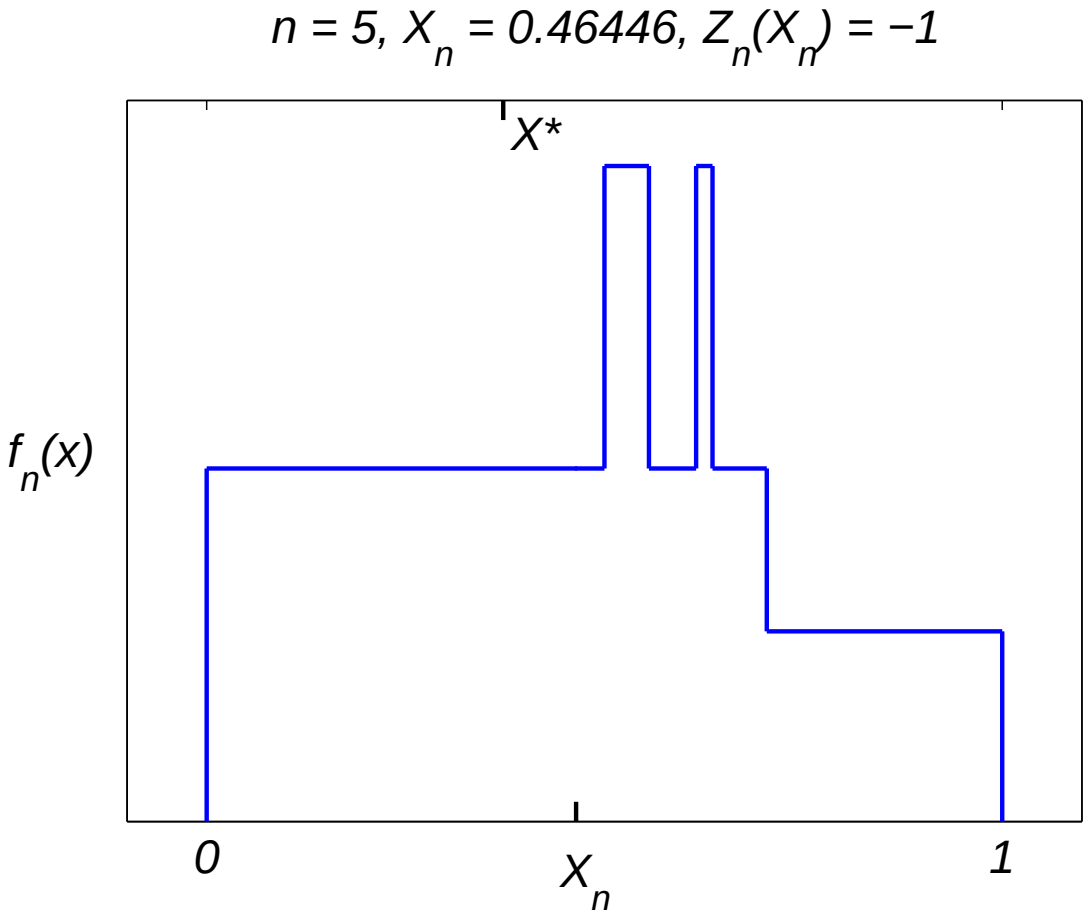
Animation showing the posterior under the optimal policy



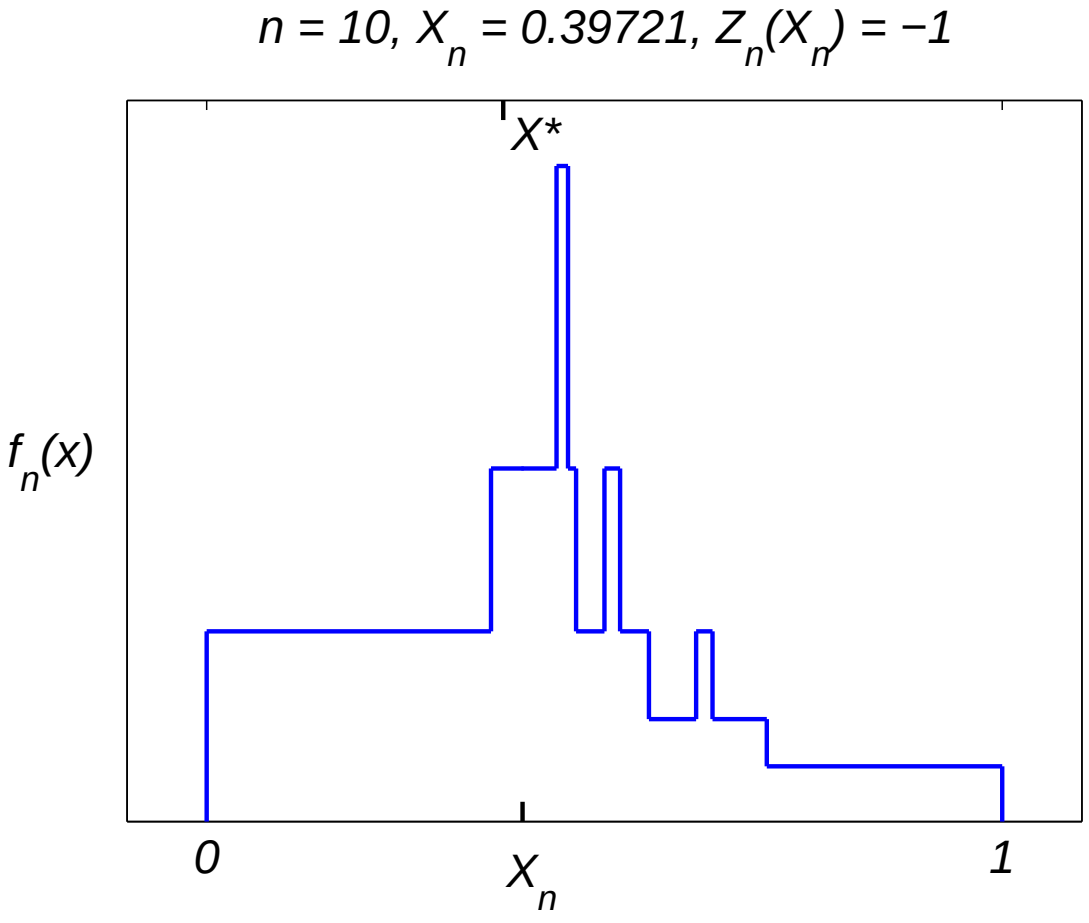
Animation showing the posterior under the optimal policy



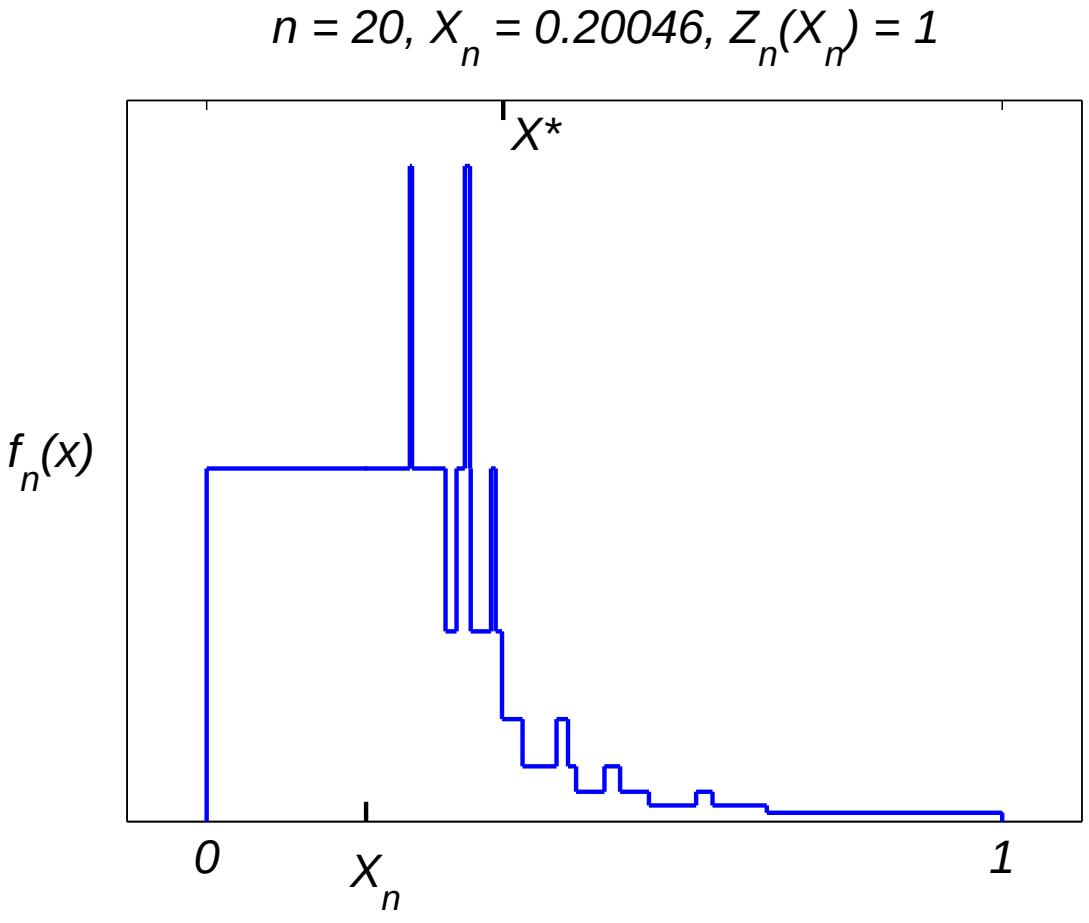
Animation showing the posterior under the optimal policy



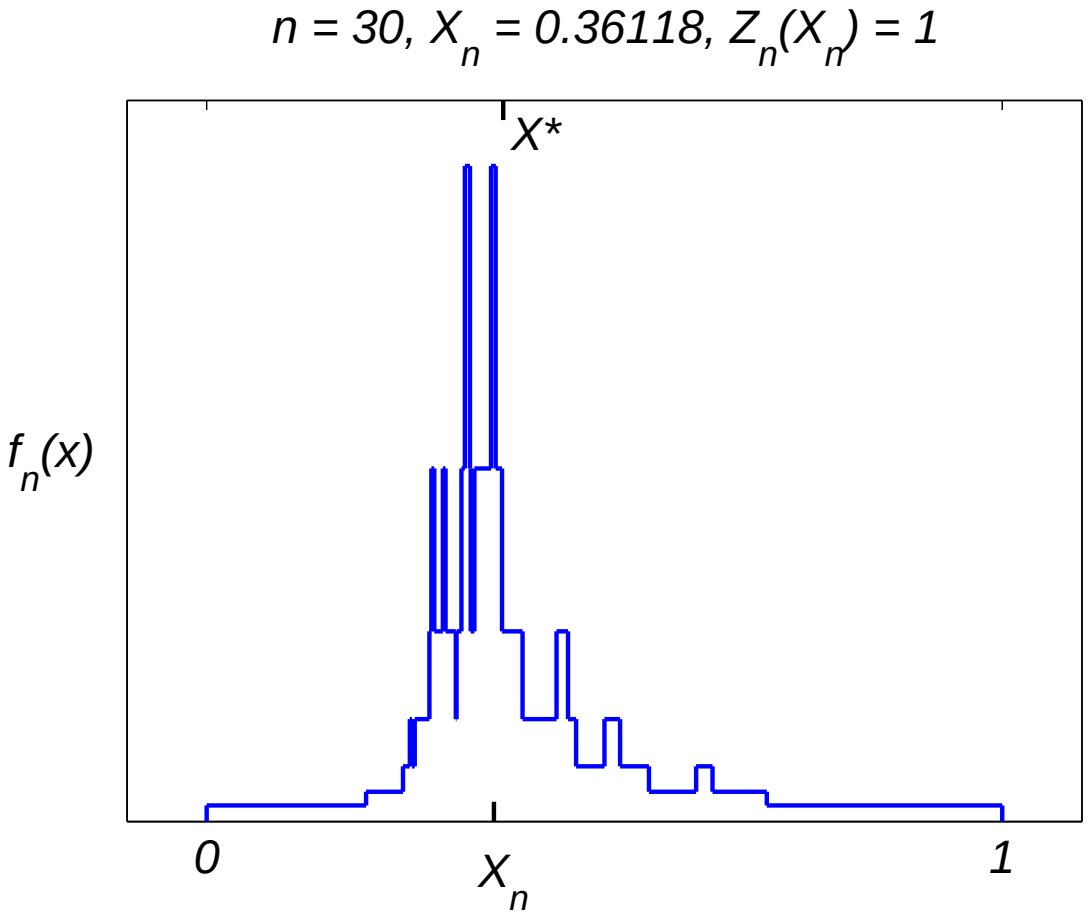
Animation showing the posterior under the optimal policy



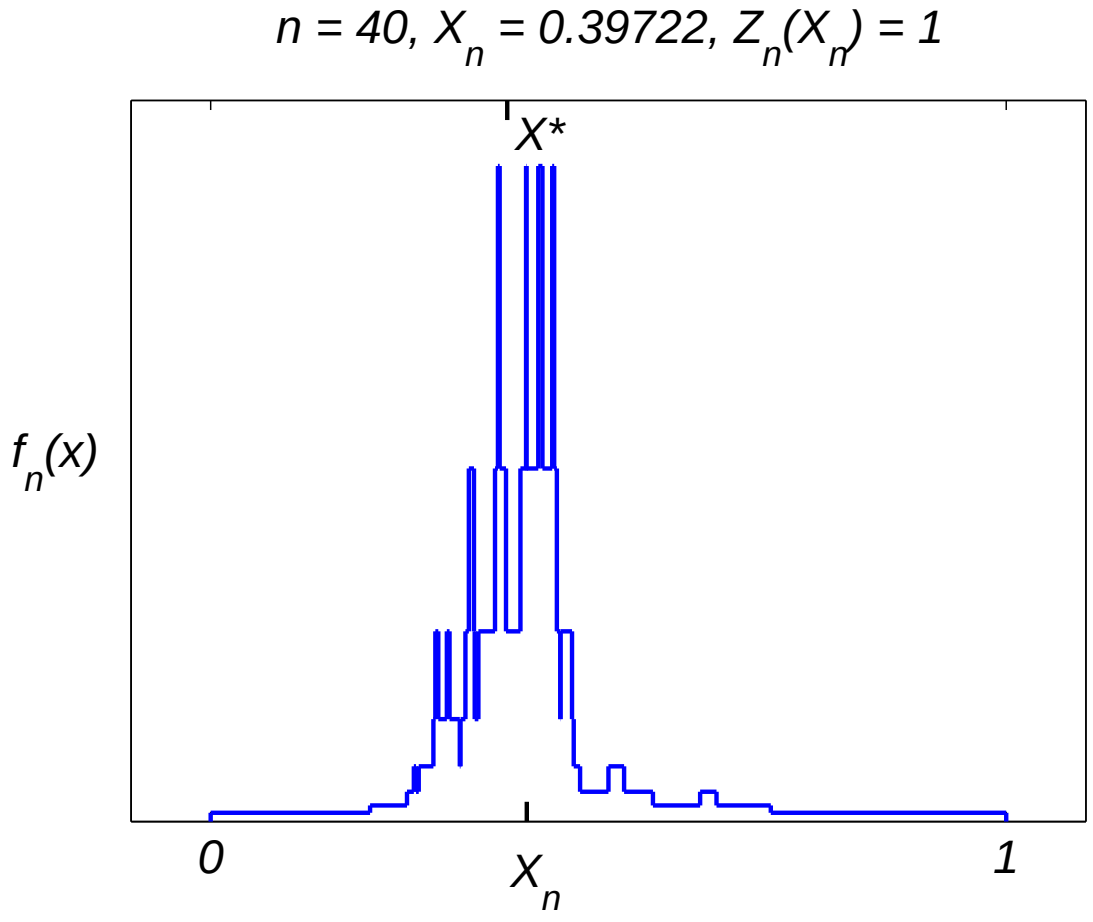
Animation showing the posterior under the optimal policy



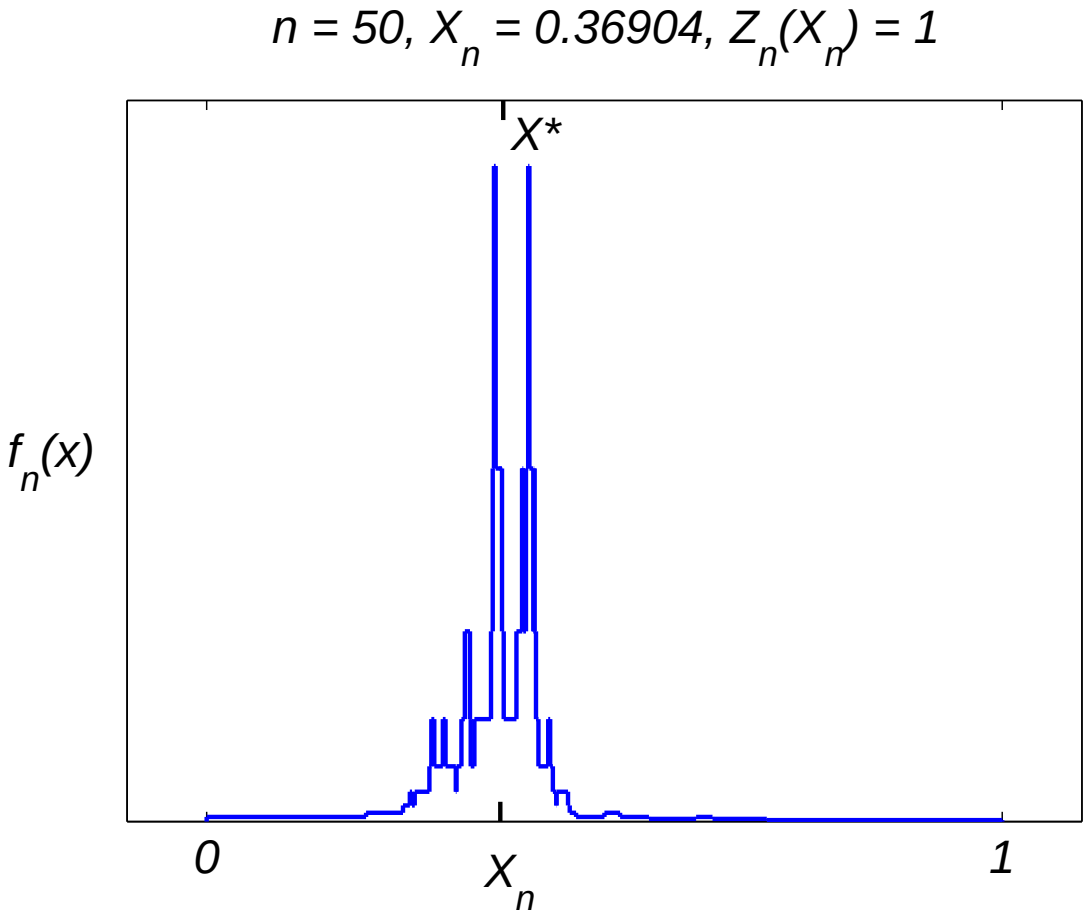
Animation showing the posterior under the optimal policy



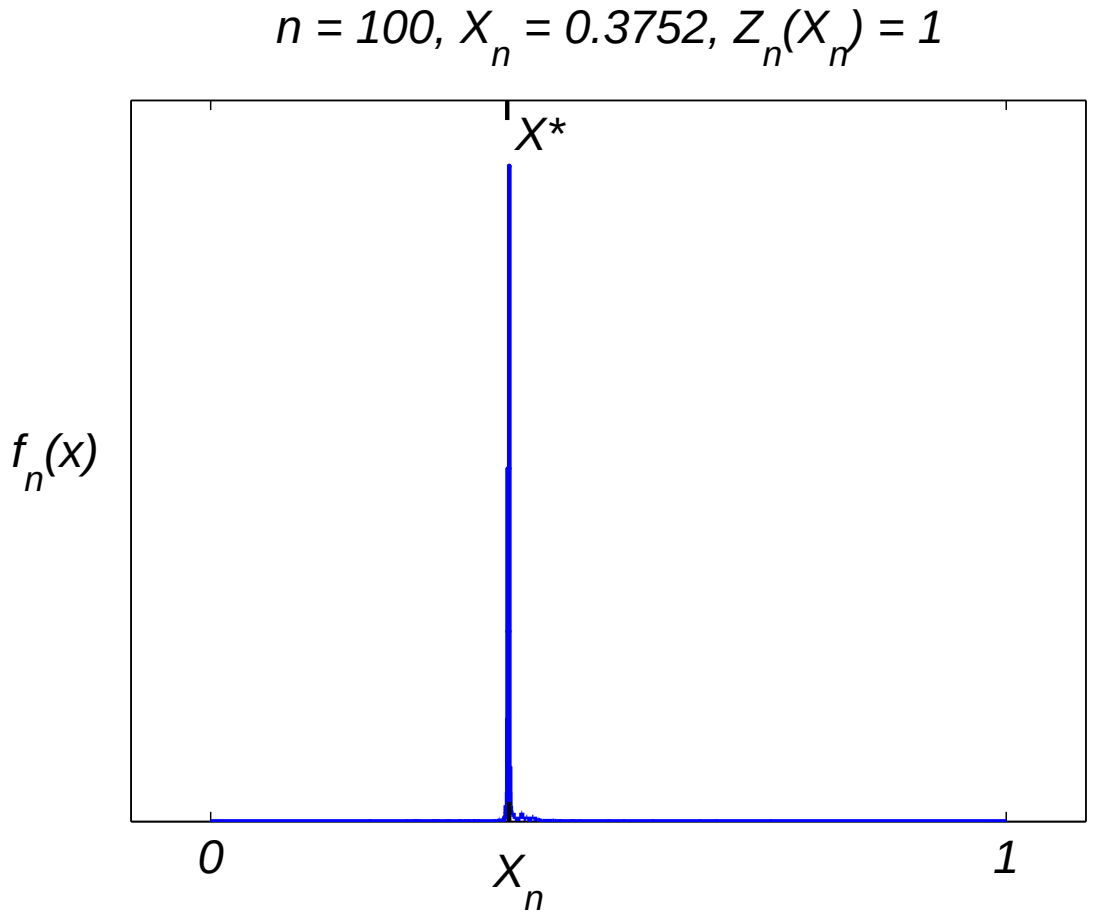
Animation showing the posterior under the optimal policy



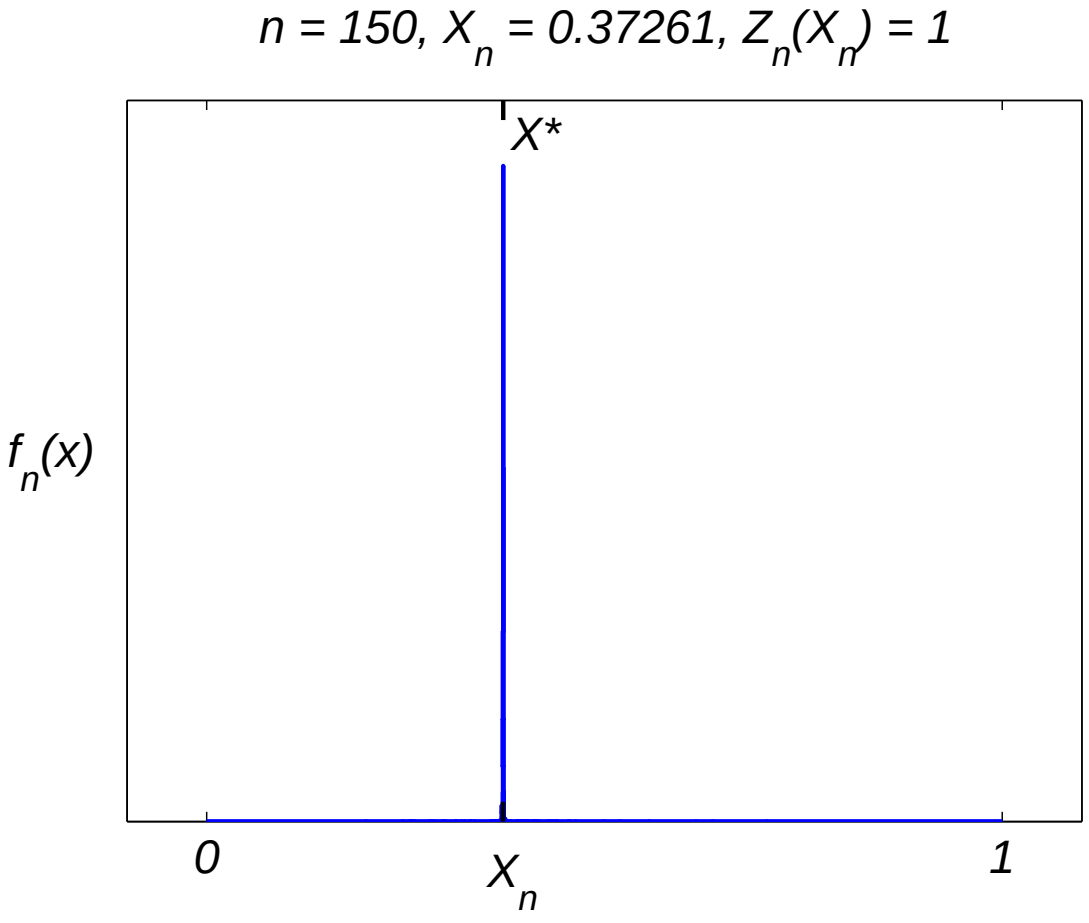
Animation showing the posterior under the optimal policy



Animation showing the posterior under the optimal policy



Animation showing the posterior under the optimal policy



This algorithm converges exponentially fast

- We used entropy as our measure of residual uncertainty because it makes the dynamic program tractable.
- In practice, we are interested in other measures, such as expected absolute loss.

Theorem

Under the policy that chooses X_n at the median of f_n, when $p(\cdot)$ is constant, known, and bounded away from $1/2$,

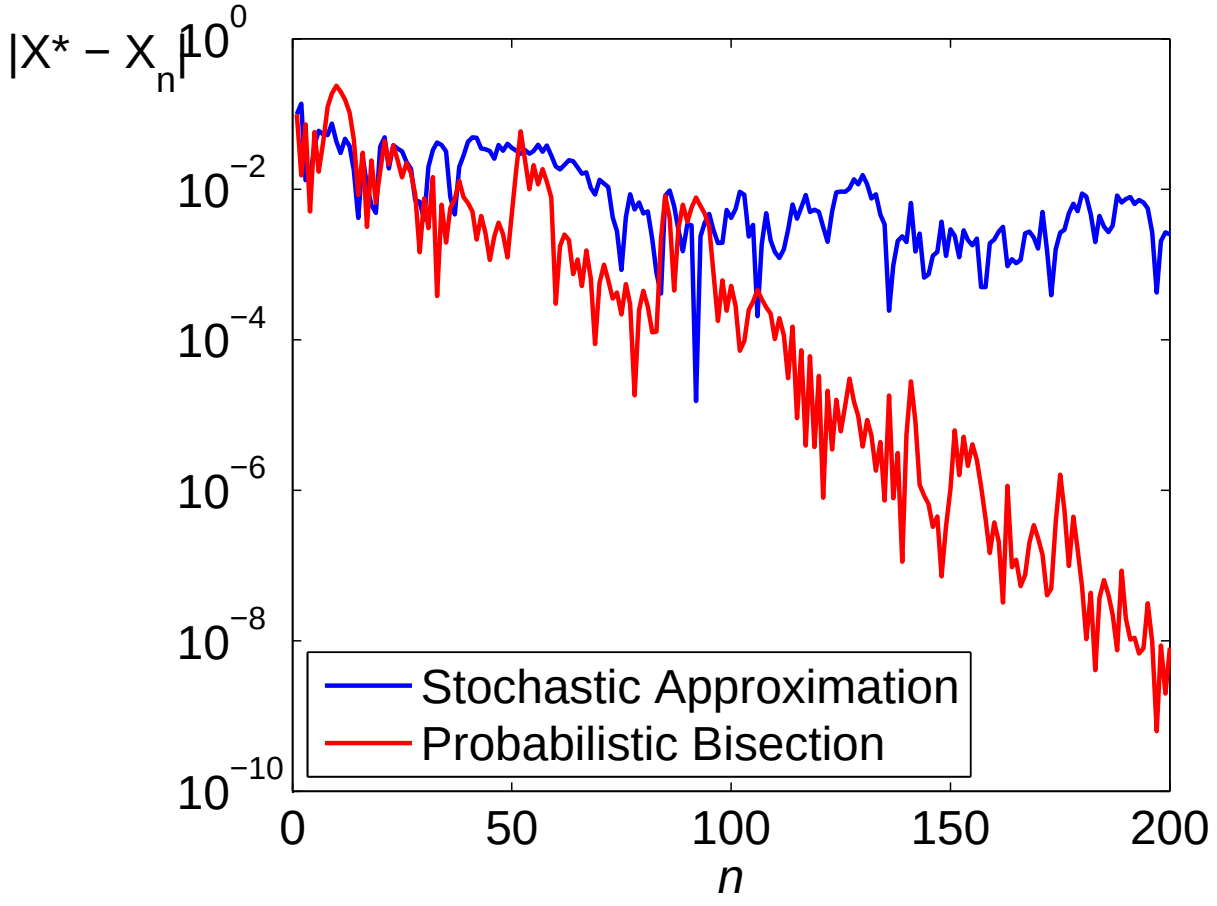
$$E|X_n - X^*| = O(e^{-rn}) \text{ for some } r > 0$$

[Waeber, Frazier, Henderson 2013].

The value of r depends on $p(\cdot)$.

In contrast, under stochastic approximation, $E|X_n - X^*| = O(n^{-1/2})$ (assuming a well-chosen tuning sequence, and that certain technical conditions are met).

This algorithm outperforms stochastic approximation when $p(\cdot)$ is constant and known



This algorithm is not new

- This algorithm was introduced in [Horstein, 1963], and is called the **Probabilistic Bisection Algorithm**, though it was not known to be Bayes-optimal.
- A discretized version was introduced in [Burnashev and Zigangirov, 1974].
- Analyses of the discretized version, or other related problems: [Feige et al., 1994], [Karp and Kleinberg, 2007], [Ben-Or and Hassidim, 2008], [Nowak, 2008], [Nowak, 2009], ...
- Survey paper: [Castro and Nowak, 2008]

This algorithm is not new

- This algorithm was introduced in [Horstein, 1963], and is called the **Probabilistic Bisection Algorithm**, though it was not known to be Bayes-optimal.
- A discretized version was introduced in [Burnashev and Zigangirov, 1974].
- Analyses of the discretized version, or other related problems: [Feige et al., 1994], [Karp and Kleinberg, 2007], [Ben-Or and Hassidim, 2008], [Nowak, 2008], [Nowak, 2009], ...
- Survey paper: [Castro and Nowak, 2008]

  > *"The [probabilistic bisection] algorithm seems to work extremely well in practice, but it is hard to analyze and there are few theoretical guarantees for it, especially pertaining error rates of convergence."*

Outline

1. Stylized Setting: $p(\cdot)$ is known and constant

2. Practical Setting: $p(\cdot)$ is unknown, and bounded below by $p_c > 1/2$.

Although our analysis was idealized, we can use probabilistic bisection in practice

- The previous analysis assumed $p(\cdot)$ was known and constant.
  - This assumption is not met in practice.
- We now use the sequential sampling method to construct our left/right observations $Z_n(X_n)$. These satisfy $p(\cdot) \geq p_c$.
- We still use the probabilistic bisection algorithm, but update f_n using p_c instead of the unknown $p(X_n)$.
  - f_n is no longer a true Bayesian posterior density.

We will analyze probabilistic bisection in this practical setting

We will give theorems related to consistency, confidence intervals, and rates of convergence for probabilistic bisection in this practical setting.

These theorems will assume:

- $X^* \in [0,1]$ fixed and unknown.
- $X_n \neq X^*$ for any finite $n \in \mathbb{N}$.
- $p(X_n) \geq p_c$ for all $n \in \mathbb{N}$.

We use the notation $q_c = 1 - p_c$.

Probalistic bisection is consistent

Theorem

$X_n \to X^*$ almost surely as $n \to \infty$.

We can create a confidence interval for X^*

- Notation: $q_c = 1 - p_c$; $\mu = p_c \ln 2p_c + q_c \ln 2q_c$.
- For $\alpha \in (0,1)$, define $b_n = n\mu - n^{1/2}(-0.5 \ln \alpha)^{1/2}(\ln 2p_c - \ln 2q_c)$.
- Our confidence interval will be:

$$J_n = conv(x \in [0,1] : f_n(x) \geq e^{b_n}).$$

Theorem

For $\alpha \in (0,1)$,

$$\mathbb{P}(X^* \in J_n) \geq 1 - \alpha,$$

for all $n \in \mathbb{N}$.

In contrast, we unaware of a method for a creating a true confidence interval using stochastic approximation.

This confidence interval shrinks at an exponential rate

Theorem

Choose $p_c \geq 0.85$, $\alpha \in (0,1)$. *For* $0 < r < \mu - q_c \ln 2p_c$ *there exists a* $N(p_c, r, \alpha) \in \mathbb{N}$, *such that*

$$\mathbb{P}(|J_n| \leq e^{-rn}, X^* \in J_n) \geq 1 - \alpha,$$

for all $n \geq N(p_c, r, \alpha)$.

Our estimates of the root converge at an exponential rate

Theorem

Define $\hat{X}_n$ to be any point in J_n, then there exists $r > 0$ such that

$$\mathbb{E}[|X^* - \hat{X}_n|] = O(e^{-rn}).$$

- This is extremely fast compared to stochastic approximation:

$$O(e^{-rn}) \text{ vs. } O(n^{-1/2}).$$

- And we have true confidence intervals for X^*.
- But n is the number of measurement points, what about total wall-clock time?

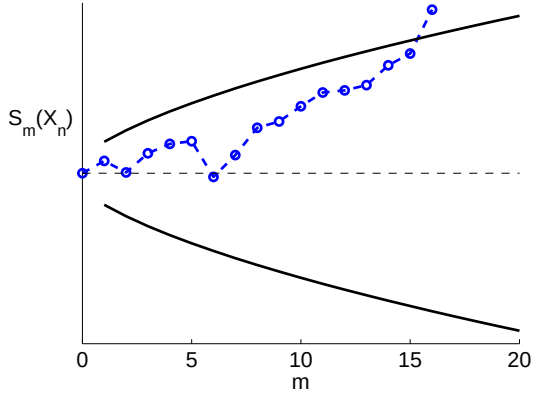
We have considered iteration count, but wall-clock time is more important

At each iteration of this bisection algorithm we:

- Sample sequentially at X_n, observing $S_m(X_n) = \sum_{i=1}^{m} Y_{n,i}(X_n)$, until

$$N_n = \inf\left\{ m : |S_m| \geq [(m+1)(\log(m+1) + 2\log(1/\alpha))]^{1/2} \right\},$$

- We should measure performance by wall-clock time: $T_n = \sum_{i=1}^{n} N_n$.



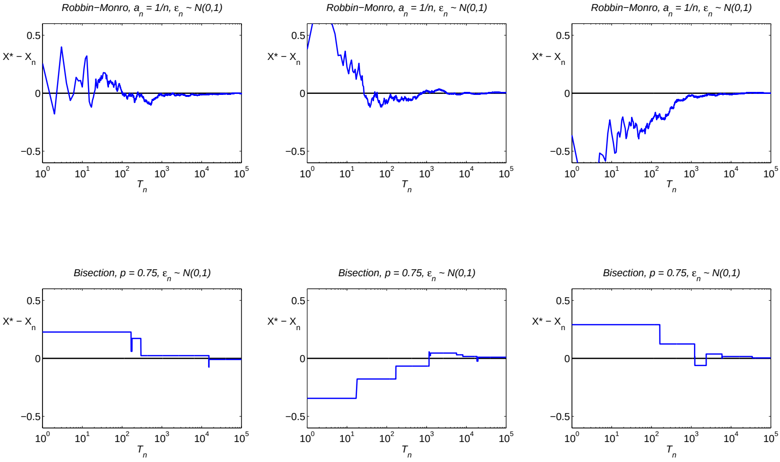
Considering wall-clock time is especially important if $g(x) \to 0$ as $x \to X^*$

Suppose we have iid normal noise.

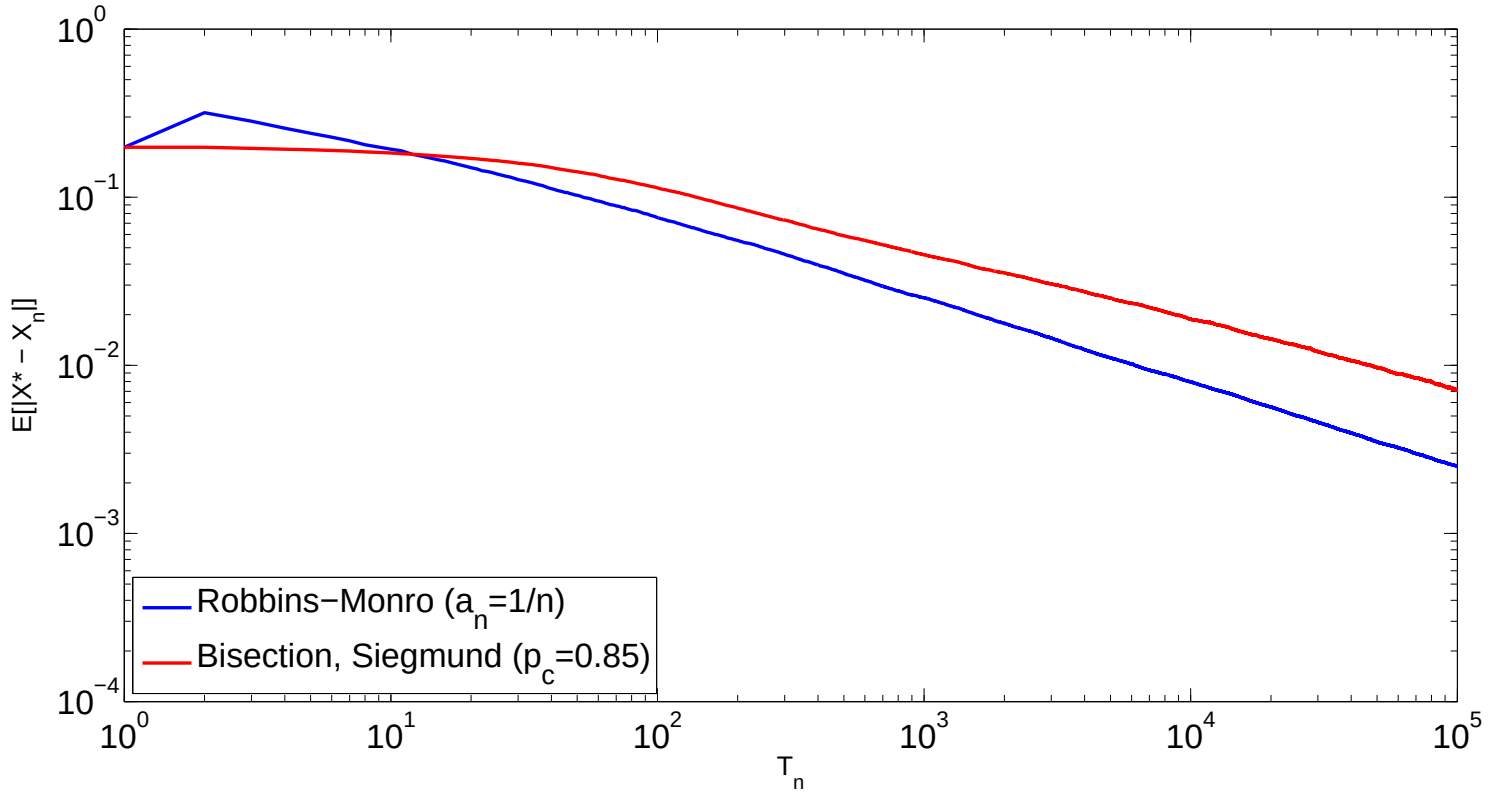
- If $g(\cdot)$ jumps across 0 at X^*, then the expected number of samples required per iteration of the bisection algorithm remains bounded as x approaches X^*.
- If $g(x) \to 0$ as $x \to X^*$, then the expected number of samples per iteration grows to infinity.

Under stochastic approximation, one iteration always requires one sample.

Stochastic approximation and probabilistic bisection have very different sample paths



Numerical comparisons show stochastic approximation is faster than probabilistic bisection in wall-clock time



Theory shows stochastic approximation is faster than probabilistic bisection in wall-clock time

- We use this result from [Farrell, 1964]:

$$\mathbb{E}_{g(x)}[N] \sim (1/g(x))^2 \log\log(1/|g(x)|) \text{ as } g(x) \to 0,$$

and for all tests of power one, if $\mathbb{P}_0(N = \infty) > 0$, then

$$\lim_{g(x)\to 0} g(x)^2 \mathbb{E}_{g(x)}[N] = \infty.$$

Theorem

If $g(x) \to 0$ as $x \to X^*$, and we have homoscedastic normal noise, then $(|X^* - X_n|(T_n)^{1/2})_n$ is not tight.

- If

$$g(x) \to 0 \text{ as } x \to X^*,$$

and if we use X_n as the best estimate of X^* then the Probabilistic Bisection Algorithm with power one tests is **asymptotically slower** than Stochastic Approximation.

We conjecture that probabilistic bisection is almost as fast as stochastic approximation in wall-clock time

- X_n might not be the best estimate for X^* when we use power one tests.
- Intuitively, observations where we spend more time should also be closer to X^*, hence an estimator of the form

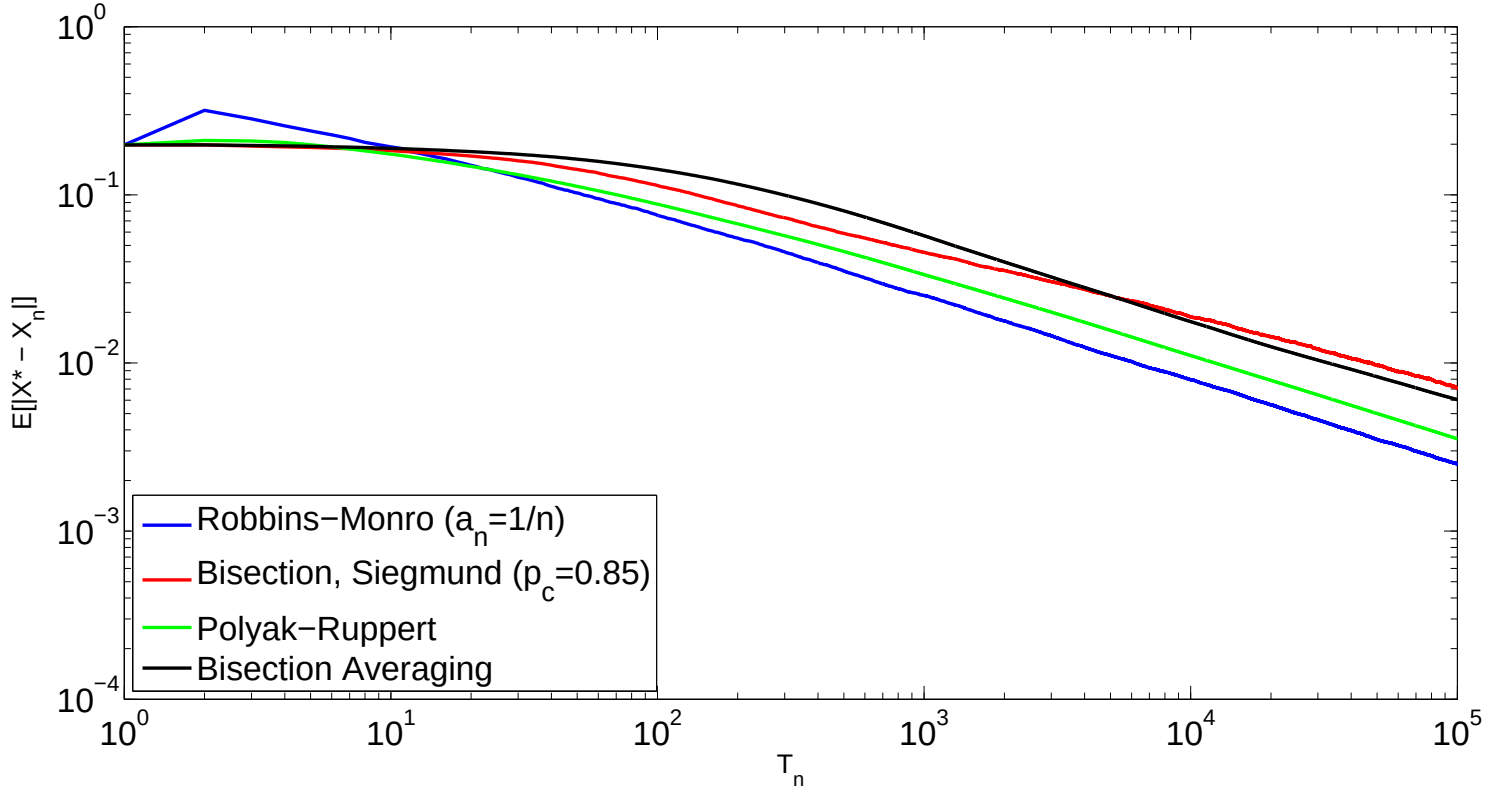
$$\tilde{X}_n = \frac{1}{T_n} \sum_{i=1}^{n} N_i X_i$$

  should perform better.
- **Conjecture:** For any $\varepsilon > 0$ it holds that

$$\mathbb{E}[|\tilde{X}_n - X^*|] = O(T_n^{-\frac{1}{2}+\varepsilon}),$$

  (if g satisfies some growth conditions).

Numerical Comparison Cont.



Conclusions

Advantages:

- Provides true confidence interval of the root X^*.
- Works extremely well if there is a jump at $g(X^*)$ (geometric rate of convergence).
- Only one tuning parameter.
- Better use of prior information.
- Better ability to track progress.
- Robust finite-time performance

Drawbacks:

- Asymptotically slower than stochastic approximation (but not by much).
- Higher computational cost per iteration.

Future Research:

- Use parallel computing (very little switching of $(X_n)_n$).
- Extension to higher dimensions.

Thank You

Any questions?

References I


Ben-Or, M. and Hassidim, A. (2008).
The Bayesian learner is optimal for noisy binary search (and pretty good for quantum as well).
In *49th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 221–230. IEEE.


Burnashev, M. V. and Zigangirov, K. S. (1974).
An interval estimation problem for controlled observations.
Problemy Peredachi Informatsii, 10(3):51–61.


Castro, R. M. and Nowak, R. D. (2008).
Active learning and sampling.
In Hero, A. O., Castañón, D. A., Cochran, D., and Kastella, K., editors, *Foundations and Applications of Sensor Management*, pages 177–200. Springer.


Farrell, R. H. (1964).
Asymptotic behavior of expected sample size in certain one sided tests.
Ann. Math. Statist., 35(1):36–72.


Feige, U., Raghavan, P., Peleg, D., and Upfal, E. (1994).
Computing with noisy information.
SIAM J. Comput., 23(5):1001–1018.

References II


Horstein, M. (1963).
Sequential transmission using noiseless feedback.
IEEE Trans. Inform. Theory, 9(3):136–143.


Jedynak, B., Frazier, P. I., and Sznitman, R. (2012).
Twenty questions with noise: Bayes optimal policies for entropy loss.
J. Appl. Prob., 49(1):114–136.


Karp, R. M. and Kleinberg, R. (2007).
Noisy binary search and its applications.
In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 881–890. SIAM.


Nowak, R. D. (2008).
Generalized binary search.
In *46th Annual Allerton Conference on Communication, Control, and Computing*, pages 568–574.


Nowak, R. D. (2009).
Noisy generalized binary search.
In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A., editors, *Adv. Neural Inf. Process. Syst. 22*, pages 1366–1374.

References III


Robbins, H. and Monro, S. (1951).
A stochastic approximation method.
Ann. Math. Statist., 22(3):400–407.


Siegmund, D. (1985).
Sequential Analysis: tests and confidence intervals.
Springer.


Waeber, R., Frazier, P. I., and Henderson, S. G. (2013).
Bisection search with noisy responses.
SIAM J. Control Optim.