Information-Based Objective Functions for Active Data Selection

By David J. C. Mackay

Presented by Aditya Sanghi and Grant Watson

Motivation

- Active learning – A learning algorithm which is able to interactively query for more data points for the training process, so as to better achieve some goal.
- Actively selecting data can be useful in two cases.
  - Slow/expensive measurements
  - Useful data subset selection
- Basic Idea: Come up with objective functions over the input space that quantify the information gained that will help in actively selecting new data

Statement of the Problem: The setup ……

- Already gathered input and output pairs

$$D_N = \{\mathbf{x}^{(m)}, \mathbf{t}^{(m)}\}$$

- Data are modeled with an interpolant $\mathbf{y}(\mathbf{x}; \mathbf{w}, \mathcal{A})$
- An interpolation model *H*
  - architecture *A*
  - Regularizer or prior on *w*
  - Cost function or noise model *N*

Statement of the Problem: The goal

- Roughly, our goal is to pick points to add to our dataset which are most informative in some sense.
- Depends on what we are interested in -
  - Selecting new data points to be maximally informative about the values that that model's parameters *w* should take
  - The above, but only concerning a region in the input space
  - Selecting data to give us maximal information to discriminate between two models.
- Possible Problem: Can our choice bias over inferences ?
  - No, our Bayesian inference will be conditioned on the data so in some sense we have marginalized out the sampling strategy

Choice of Information measure

- Measuring information gain either by calculating change in entropy or cross entropy when we select a data point
- Change in entropy:

$$\Delta S = S_N - S_{N+1}$$

- Where S_N is:

$$S_N = \int d^k \mathbf{w}\, P^N(\mathbf{w}) \log \frac{m(\mathbf{w})}{P^N(\mathbf{w})}$$

Measure on w

Probability of Parameters before you receive the datum

Choice of Information measure

- Cross entropy:

Probability of Parameters before you receive the datum

$$G = \int d^k \mathbf{w}\, P^{N+1}(\mathbf{w}) \log \frac{P^N(\mathbf{w})}{P^{N+1}(\mathbf{w})}$$

$$G' = -G$$

Probability of Parameters after you receive the datum

- G' is the KL Divergence
  - Measure of how much information we gain when we are informed the true distribution of w is $P^{N+1}(w)$ rather than $P^N(w)$

Comparing Information Measures

- Change in entropy
  - Shrinkage of high probability bubble region
  - Invariant under translation
- Cross entropy
  - Can also respond to translation
-

$$E(\Delta S) = E(G')$$

where expectation is over P(t) [N + 1 datum generating distribution]

- The above shows that E(ΔS) is independent of m(w) and it does not matter which form of information we use

Review of Mackay's Notation

$$P(D\,|\mathbf{w},\beta,\mathcal{A}) = \frac{\exp(-\beta E_D(D|\mathbf{w},\mathcal{A}))}{Z_D(\beta)}$$

Likelihood

$$\text{where } \beta = 1/\sigma_\nu^2,\ E_D = \textstyle\sum_m \frac{1}{2}(y(x_m) - t_m)^2, \text{ and } Z_D = (2\pi/\beta)^{N/2}$$

$$P(\mathbf{w}|\mathcal{A},\mathcal{R},\alpha) = \frac{\exp(-\alpha E_W(\mathbf{w}|\mathcal{A},\mathcal{R}))}{Z_W(\alpha)}$$

Regularizing Function

Prior

$$\text{where } Z_W = \int d^k\mathbf{w}\ \exp(-\alpha E_W)$$

Review of Mackay's Notation

$$P(\mathbf{w}|D,\alpha,\beta,\mathcal{A},\mathcal{R}) = \frac{P(D|\mathbf{w},\beta,\mathcal{A})P(\mathbf{w}|\alpha,\mathcal{A},\mathcal{R})}{P(D|\alpha,\beta,\mathcal{A},\mathcal{R})}$$

Posterior

$$M(\mathbf{w}) = \alpha E_W + \beta E_D,$$

$$P(\mathbf{w}|D,\alpha,\beta,\mathcal{A},\mathcal{R}) = \frac{\exp(-M(\mathbf{w}))}{Z_M(\alpha,\beta)}$$

$$\text{where } Z_M(\alpha,\beta) = \int d^k\mathbf{w}\ \exp(-M).$$

Task 1: Deriving the total information gain

$$M(\mathbf{w}) \simeq M^*(\mathbf{w}) = M(\mathbf{w}_{\mathrm{MP}}) + \frac{1}{2}\Delta\mathbf{w}^{\mathrm{T}}\mathbf{A}\Delta\mathbf{w}$$

$$S = \frac{k}{2}(1 + \log 2\pi) + \frac{1}{2}\log\left(m^2 \det \mathbf{A}^{-1}\right)$$ Where we used $P(\mathbf{w}) \propto e^{-M^*(\mathbf{w})}$

Expanding y around w_{mp}:

$$\mathbf{y}(\mathbf{x}) \simeq \mathbf{y}(\mathbf{x}; \mathbf{w}_{\mathrm{MP}}) + \mathbf{g}(\mathbf{x}) \cdot \Delta\mathbf{w} \qquad g_j = \partial y / \partial w_j$$

If the datum *t* falls in the region such that our quadratic approximation applies

$$\mathbf{A}_{N+1} \simeq \mathbf{A} + \beta\mathbf{g}\mathbf{g}^{\mathrm{T}} \qquad \nabla\nabla\tfrac{1}{2}[\mathbf{t} - \mathbf{y}(\mathbf{x}; \mathbf{w})]^2 \simeq \mathbf{g}\mathbf{g}^{\mathrm{T}}$$

It is independent of the value that the datum t actually takes, so we can evaluate $A_{\mathrm{N}+1}$ just by calculating g

Task 1: Deriving the total information gain

$$\text{Total information gain} = \frac{1}{2}\Delta \log\left(m^2 \det \mathbf{A}\right)$$

$$= \frac{1}{2}\log(1 + \beta \mathbf{g}^{\mathrm{T}} \mathbf{A}^{-1} \mathbf{g})$$

Using $\det\left[\mathbf{A} + \beta \mathbf{g}\mathbf{g}^{\mathrm{T}}\right] = (\det \mathbf{A})(1 + \beta \mathbf{g}^{\mathrm{T}} \mathbf{A}^{-1} \mathbf{g})$

Interpretation:

- More information if low intrinsic noise
- More information if higher interpolant variance
- Assuming constant noise, this measure will most likely encourage picking points at the edges of our current data set

Task 2: Information gain in a region of interest

- Motivation – The total information gain will encourage data selection at edges. Redefine the problem to look at local regions
- Problem Statement - We wish to gain maximal information about the value of the interpolant at a particular point $x_{(u)}$
- Again assuming quadratic approximation, the variance in interpolant is given by

$$\sigma_u^2 = g_{(u)}^T A^{-1} g_{(u)}$$

Task 2: Information gain in a region of interest

$$\text{Marginal information gain} = \frac{1}{2}\Delta \log \sigma_u^2$$

$$\left[\mathbf{A} + \beta \mathbf{g}\mathbf{g}^{\mathrm{T}}\right]^{-1} = \mathbf{A}^{-1} - \frac{\beta \mathbf{A}^{-1}\mathbf{g}\mathbf{g}^{\mathrm{T}}\mathbf{A}^{-1}}{1 + \beta \mathbf{g}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{g}} \longrightarrow = -\frac{1}{2}\log\left[1 - \frac{(\mathbf{g}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{g}_{(u)})^2}{\sigma_u^2(\sigma_\nu^2 + \sigma_{\mathbf{x}}^2)}\right]$$

Interpretation -

- Top term is maximal when you pick you align the input with the regional point.
- Example cases:
  - Constant intrinsic noise, interpolant variance → picking a point at the sample location will maximize correlation
  - Much stronger noise at $x_{(u)}$ → Denominator might overpower the numerator at $x_{(u)}$; best pick somewhere else.

Task 2: Information gain in a region of interest

- Want to construct objective function that defines information gain for multiple points that represents a region.
  - Define regional representatives with output variables $\{y^{(u)}\}$ and inputs $\{x^{(u)}\}$, where u = 1 …. V.
- Two candidates:
  - Joint information gain
    - We'll skip this. It ends up not being useful since using maximizing this gain can create arbitrary correlations in the representatives' sensitivities.
  - Mean Marginal information gain

Task 2: Information gain in a region of interest

Mean Marginal information Gain:

- Take a weighted average of the individual $y^{(u)}$ entropies

$$S^{M} = \sum_{u} P_u \, S[P(y^{(u)})] = \frac{1}{2} \sum_{u} P_u \, \log \sigma_u^2 + \text{const}$$

$$\text{Mean marginal information gain} = -\frac{1}{2} \sum_{u} P_u \, \log \left[1 - \frac{(\mathbf{g}^{T} \mathbf{A}^{-1} \mathbf{g}_{(u)})^2}{\sigma_u^2 (\sigma_\nu^2 + \sigma_{\mathbf{x}}^2)} \right]$$

Where $P_{(u)}$ is the probability that we will be asked to predict $y^{(u)}$

Task 2: Information gain in a region of interest

Mean Marginal information Gain:

- Two simple variations:
  - $\sigma_u^2 \rightarrow \sigma_v^2 + \sigma_u^2$: This may lead to different choices if $\sigma_u^2 < \sigma_v^2$
  - $E^M = \Sigma_u P_u \sigma_u^2$: More strongly penalizes large variance

Case of linear models

$$y = \Sigma w_h \phi_h(x)$$

- Hessian Matrix will be independent of {t}
- The sensitivities g only depend on the ϕ_h.
- Consequence: we can completely specify the information gains for a sequence of choices before even seeing the targets.

Task 3: Discriminating two models

- Again under quadratic approximation, two models will make slightly different gaussian predictions about the value of any datum

$$P(t \mid \mathcal{H}_i) = \text{Normal}(\mu_i, \sigma_i^2)$$
$$\mu_i = \mathbf{y}[\mathbf{x}; \mathbf{w}_{\text{MP}}(i)]$$
$$\sigma_i^2 = \mathbf{g}_i^{\text{T}} \mathbf{A}_i^{-1} \mathbf{g}_i + 1/\beta$$

- Intuition for choice of x:
  - More information when the means are well separated with respect to a scale defined by σ_1 and σ_2
  - Separated variances allows us to explore different Occam factors

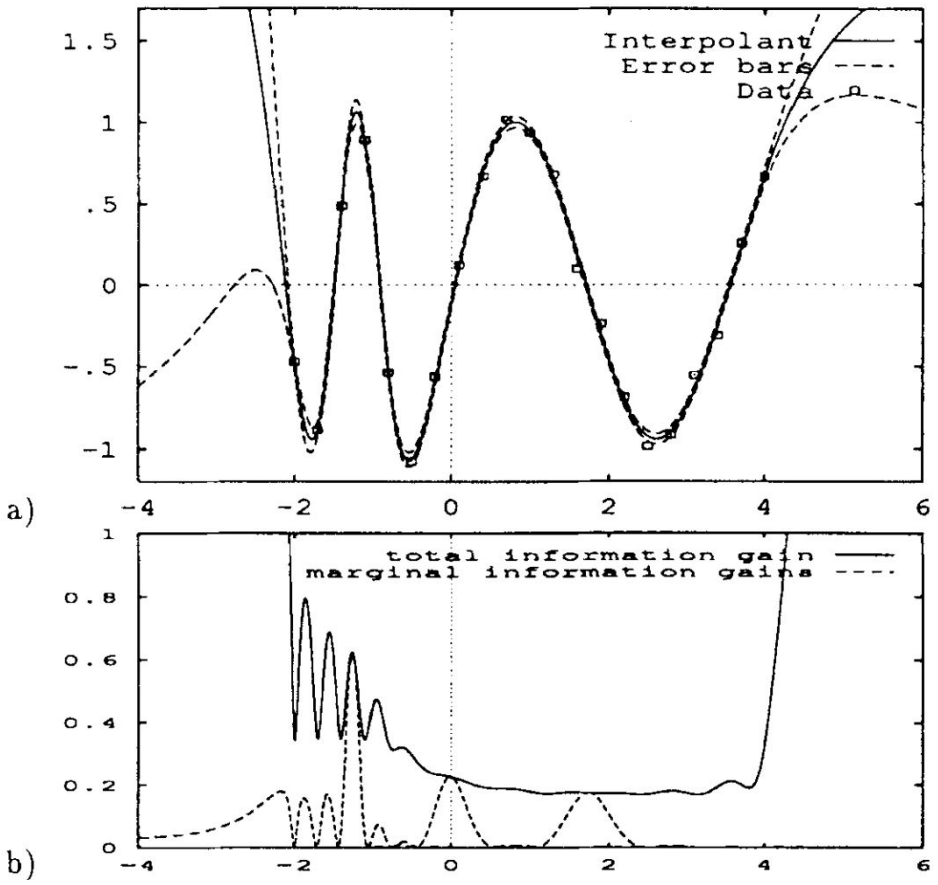
Task 3: Discriminating two models

$$S = -\sum_i P(\mathcal{H}_i) \log P(\mathcal{H}_i)$$

weak likelihood ratio: $P(t \mid \mathcal{H}_1)/P(t \mid \mathcal{H}_2)$ or $|\mu_1 - \mu_2| \ll \sigma_1, \sigma_2$

$$E(\Delta S) \simeq \frac{P(\mathcal{H}_1)P(\mathcal{H}_2)}{2}\left[\left(\frac{1}{\sigma_1^2}+\frac{1}{\sigma_2^2}\right)(\mu_1-\mu_2)^2+\left(\frac{\sigma_1^2-\sigma_2^2}{\sigma_1\sigma_2}\right)^2\right]$$

Demonstration and Discussion



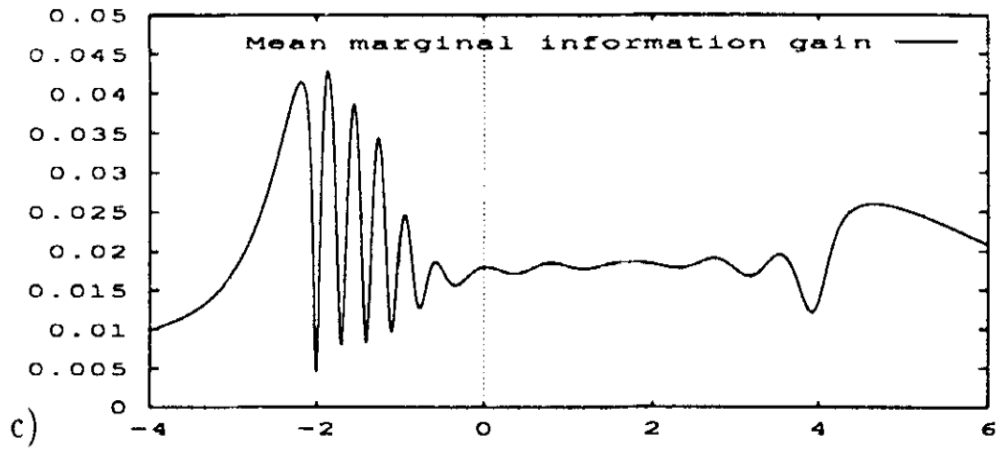


Figure 1: Demonstration of total and marginal information gain. (a) The data set, the interpolant, and error bars. (b) The expected total information gain and three marginal information gains. (c) The mean marginal information gain, with the region of interest defined by 300 equally spaced points on the interval $[-2.1, 4.1]$. The information gains are shown on a scale of nats (1 nat = $\log_2 e$ bits).

The Achilles' Heel

- We have been assuming the models have been correct.
- Incorrect models can result in blowup away from region of interest
- Example: Predicting accurately at $x^{(u)}$ with a linear model on quadratic data
  - Information gained encourages us to take points as far away as possible
  - This contradicts what will be most informative here which is sampling values close to $x^{(u)}$
- Further research: information gain in the context of approximate models

Complexity

- The task of computing for example the mean marginal likelihood is cheaper than computing + inverting the Hessian
  - O(Nk^2)+O(k^3) and O(Ck^2) + O(CVk), respectively. C = # of candidate points, V = # of region defining points.

Summary

- We defined 3 objective functions over x (total, marginal, mean marginal information gains) to address different contexts (maximizing information in total, at a point, and at a set of points).
- Requires validity of the quadratic/local Gaussian approximation of the cost function M(w).
- Weakness: assumption that models are correct.

Automated Curriculum Learning for Neural Networks

A. Graves, M. Bellemare, J. Menick,
R. Munos, K. Kavukcuoglu

Presenters:

Davi Frossard Andrew Toulis

October 20, 2017

Summary

AUTOMATED CURRICULUM LEARNING

- Interest in curriculums resurfaced in 2009 (Bengio et al.)
  - Manually steering models to train on **gradually more difficult** tasks achieved faster convergence.
- Core idea for Automated Curriculum Learning:
  Given a dataset of input-output pairs $\{\mathbf{x}, \hat{\mathbf{y}}\}$ and a model that has trained on $\{\mathbf{x}_{[0..N]}, \hat{\mathbf{y}}_{[0..N]}\}$, **learn to choose** the next training example $\{\mathbf{x}_{N+1}, \hat{\mathbf{y}}_{N+1}\}$ that maximizes learning.

AUTOMATED CURRICULUM LEARNING

- Cast curriculum learning as a Multi-Armed Bandit:
  - Curriculum with N tasks as a N**-Armed Bandit**
  - No assumptions made about rewards ("adversarial").
  - An agent selects an arm and observes its payoff, while the other payoffs are not observed.
  - **Adaptive policy** seeks to maximize payoff from bandit.

THE EXP3 ALGORITHM FOR ADVERSARIAL BANDITS

- Goal: Minimize regret with respect to best arm.
- Chooses arm i according to policy π_t with **probability**:

$$\pi_t^{Exp3}(i) = \frac{e^{w_{t,i}}}{\sum_{j=1}^{N} e^{w_{t,j}}}$$

- where $w_{t,i}$ are weights calculated as a sum of historically-observed, importance-sampled rewards:

$$w_{t,i} = \eta \sum_{s<t} \tilde{r}_{s,i}$$

$$\tilde{r}_{s,i} = r_s 1_{[a_s=i]} / \pi_s^{Exp3}(i)$$

WEAKNESSES OF EXP3: SHIFTING REWARDS

- Exp3 closely matches the best **single arm strategy** over the whole trajectory.
- For curriculum learning, a good strategy often changes:
  - Easier cases in training data will provide high rewards during early training, but have diminishing returns.
  - Over time, more difficult cases will provide higher rewards.

THE EXP3.S ALGORITHM FOR SHIFTING REWARDS

- Addresses issues of Exp3 by encouraging **exploration** with probability ϵ and by **mixing weights** additively:

$$\pi_t^{Exp3.S}(i) = (1-\epsilon)\frac{e^{w_{t,i}}}{\sum_{j=1}^{N} e^{w_{t,j}}} + \frac{\epsilon}{N}$$

$$w_{t,i} = \log\left[(1-\alpha_t)\exp\left(w_{t-1,i} + \eta\tilde{r}_{t-1,i}\right) + \frac{\alpha_t}{N-1}\sum_{j\neq i}\exp\left(w_{t-1,j} + \eta\tilde{r}_{t-1,j}\right)\right]$$

- This effectively **decays** the importance of old rewards and allows the model to react faster to changing scenarios.

LEARNING A SYLLABUS OVER TASKS

- Given: separate tasks with **unknown difficulties**
- We want to maximize the **rate of learning**:

1. At each timestep t, we sample a task index k from π_t.
2. We then sample a data batch from this task: $\{\mathbf{x}^k_{[0..B]}, \hat{\mathbf{y}}^k_{[0..B]}\}$
3. A measure of *learning progress* ν and the effort τ (computation time, input size, etc.) are calculated.
4. The rate of learning is $r_t = \frac{\nu}{\tau}$ and is re-scaled to $[-1, 1]$.
5. Parameters w of the policy π are updated using Exp3.S

LEARNING PROGRESS MEASURES

- It is computationally expensive (or intractable) to measure the global impact of training on a particular sample.
- We desire proxies for progress that depend only on the current sample or a single extra sample.
- The paper proposes two types of progress measures:
  - **Loss-driven**: compares predictions before/after training.
  - **Complexity-driven**: information theoretic view of learning.

PREDICTION GAIN

- Prediction Gain is the change in sample loss before and after training on a sample batch **x**:

$$\nu_{PG} = L(x, \theta) - L(x, \theta_x)$$

- Moreover, when training using gradient descent:

$$\Delta\theta \propto -\nabla L(x, \theta)$$

- Hence, we have a Gradient Prediction Gain approximation:

$$\begin{aligned}\nu_{GPG} &= L(x, \theta) - L(x, \theta_x) \\ &\approx -\nabla L(x, \theta) \cdot \Delta\theta \\ &\propto ||\nabla L(x, \theta)||^2\end{aligned}$$

BIAS-VARIANCE TRADE-OFF

- Prediction Gain is a biased estimate of the **expected** change in loss due to training on a sample x:

$$E_{x' \sim Task_k}[L(x', \theta) - L(x', \theta_x)]$$

- In particular, it favors tasks that have high variance.
  - This is since sample loss decreases after training, even though loss for other samples from the task could increase.
- An unbiased estimate is the Self Prediction Gain:

$$\nu_{SPG} = L(x', \theta) - L(x', \theta_x), \quad x' \sim D_k$$

- ν_{SPG} has naturally higher variance due to sampling of x′

SHIFTING GEARS: COMPLEXITY IN STOCHASTIC VI

- Consider the objective in stochastic variational inference, where P_ϕ is a variational posterior over parameters θ and Q_ψ is a prior over θ:

$$L_{VI} = \underbrace{KLD(P_\phi||Q_\psi)}_{\text{Model Complexity}} + \overbrace{\sum_{x' \in D} E_{\theta \sim P_\phi}[L(x', \theta)]}^{\text{Data Compression under } P_\phi}$$

- Training **trades-off** better ability to compress data with higher model complexity. We expect that complexity increases the most from highly generalizable data points.

VARIATIONAL COMPLEXITY GAIN

- The Variational Complexity Gain after training on a sample batch **x** is the change in KL Divergence:

$$\nu_{VCG} = KLD(P_{\phi_x}||Q_{\psi_x}) - KLD(P_{\phi}||Q_{\psi})$$

- We can design P and Q to have a closed-form KLD. Example: both Diagonal Gaussian.
- In non-variational settings, when using L2 regularization (Gaussian Prior on weights), we can define the L2 Gain:

$$\nu_{L2G} = ||\theta_x||^2 - ||\theta||^2$$

GRADIENT VARIATIONAL COMPLEXITY GAIN

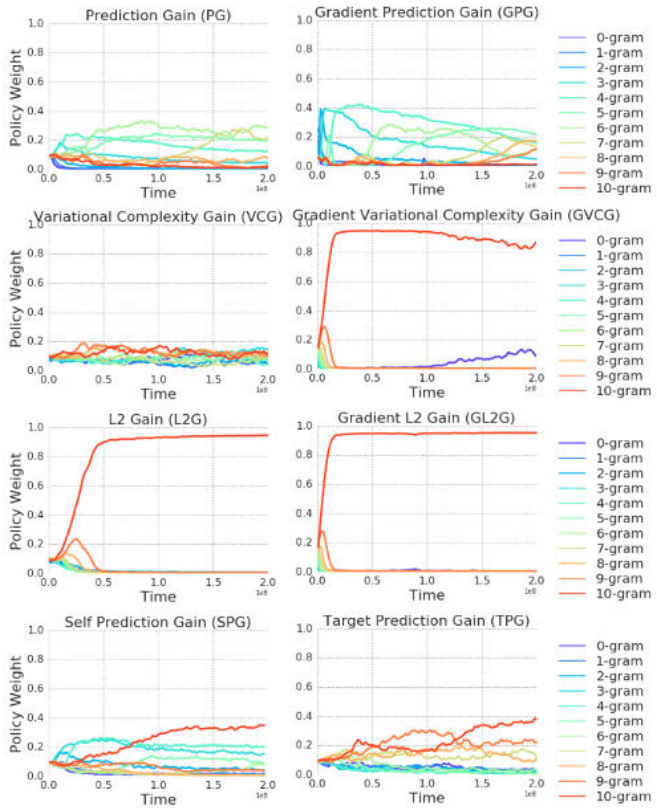
- The Gradient Variational Complexity Gain is the directional derivative of the KLD along the gradient descent direction of the data loss:

$$\nu_{GVCG} \propto \nabla_\phi KLD(P_\phi || Q_\psi) \cdot \nabla_\phi E_{\theta \sim P_\phi}[L(x, \theta)]$$

  - Other loss terms are not dependent on x.
- This gain worked well experimentally, perhaps since the curvature of model complexity is typically flatter than loss.

EXAMPLE EXPERIMENT: GENERATED TEXT

- 11 datasets were generated using increasingly complex language models. Policies gravitated towards complexity:



Credit: Automated Curriculum Learning for Neural Networks

EXPERIMENTAL HIGHLIGHTS

- Uniformly sampling across tasks, while inefficient, was a very strong benchmark. Perhaps learning is dominated by gradients from tasks that drive progress.
- For variational loss, GVCG yielded higher complexity and faster training than uniform sampling in one experiment.
- Strategies observed: a policy would focus on a task until completion. Loss would reduce on unseen (related) tasks!

SUMMARY OF IDEAS

- Discussed several progress measures that can be evaluated using training samples or one extra sample.
- By evaluating progress from each training example, a multi-armed bandit determines a stochastic policy, over which task to train from next, to maximize progress.
- The bandit needs to be non-stationary. Simpler tasks dominate early on (especially for Prediction Gain), while difficult tasks contain most of the complexity.

TAKEAWAYS

- Better learning efficiency can be achieved with the right measure of progress, but this involves experimentation.
- Final overall loss was better in one out of six experiments. A research direction is to find better local minimas.
- Most promising: Prediction Gain for MLE problems, and Gradient Variational Complexity Gain for VI.
- Variational Complexity Loss was noisy and performed worse than its gradient analogue. Determining why is an open question. It could be due to terms independent of x.

Finite-time Analysis of the Multiarmed Bandit Problem

Peter Auer, Nicolò Cesa-Bianchi, Faul Fischer

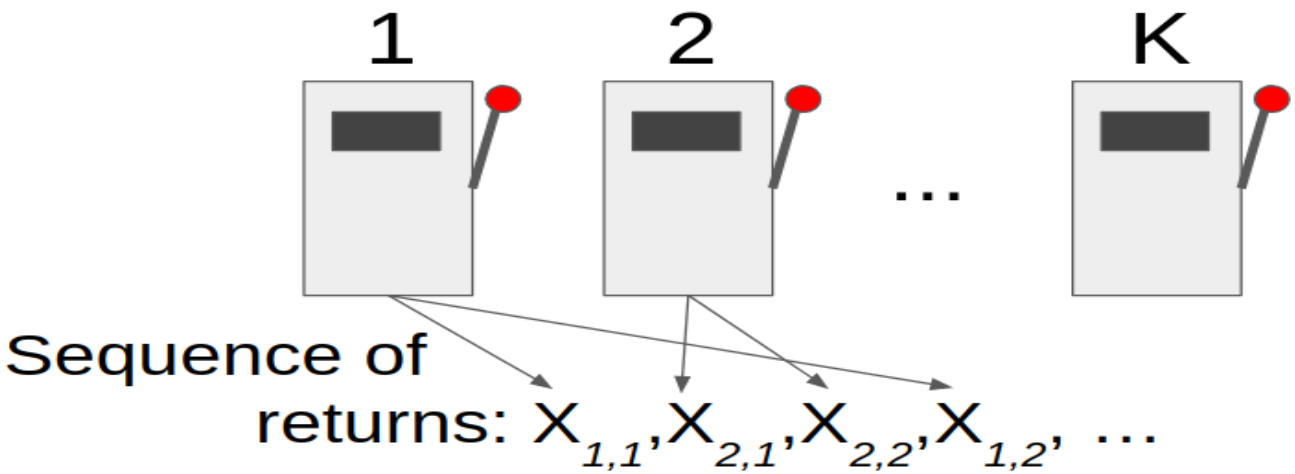
Presented by Eric Langlois

October 20, 2017

Exploration vs. Exploitation

- In reinforcement learning, must maximize long-term reward.
- Need to balance exploiting what we know already vs. exploring to discover better strategies.

Multi-Armed Bandit



- K slot machines, each with static reward distribution p_i.
- **Policy** selects machines to play given history.
- The n^{th} play of machine i ($\in 1 \ldots K$) is a random variable $X_{i,n}$ with mean μ_i.
- Goal: Maximize total reward.

Regret

How do we measure the quality of a policy?

- $T_i(n)$ - number of times machine i is played in first n plays.
- **Regret**: Expected under-performance compared to optimal play. The regret after n steps is

$$\text{Regret} = \mathbf{E}\left[\sum_{i=1}^{K} T_i(n)\Delta_i\right]$$

$$\Delta_i = \mu^* - \mu_i \qquad\qquad \mu^* = \max_{1\leq i\leq K} \mu_i$$

- Uniform random policy: linear regret
- ϵ-greedy policy: linear regret

Asymptotically Optimal Regret

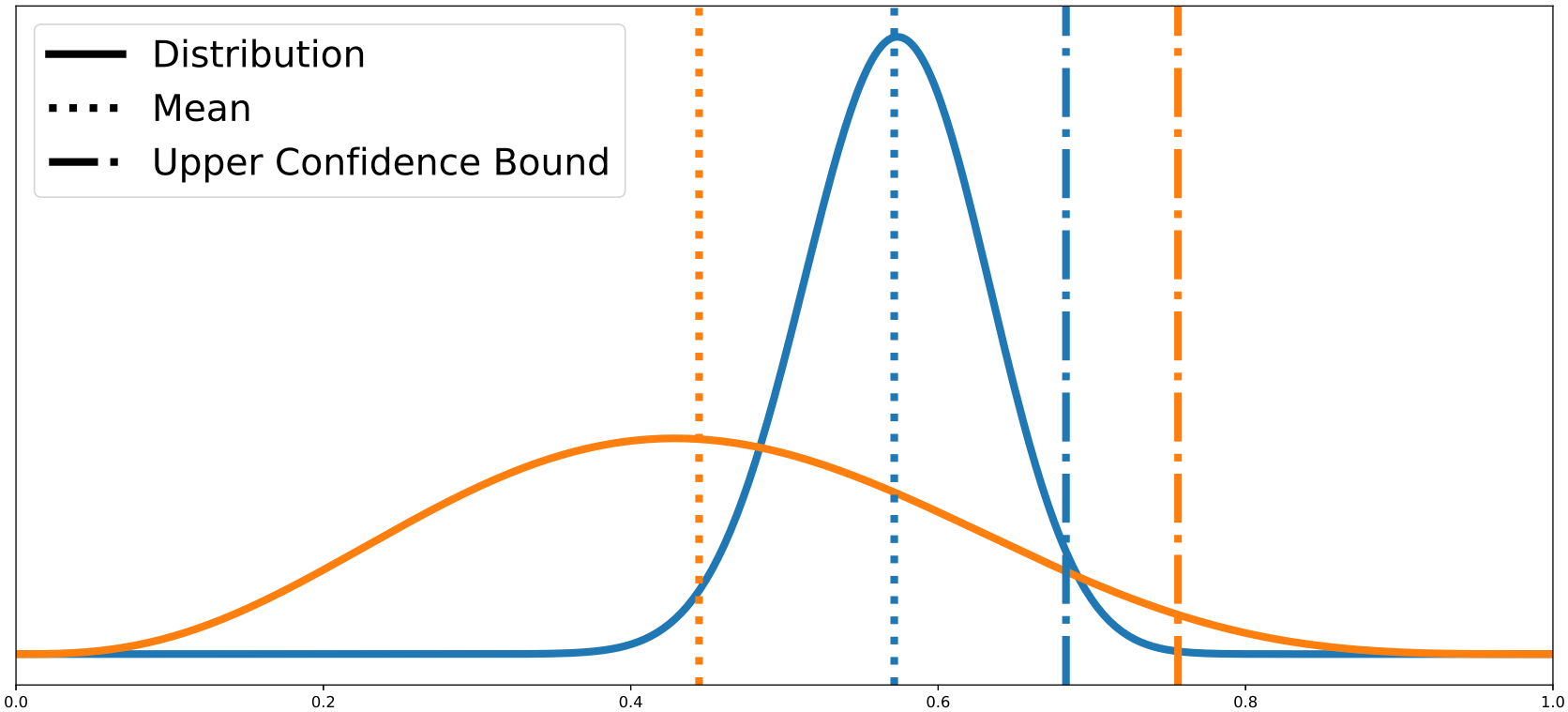
- Lai and Robbins (1985) proved there exist policies with

$$\mathbb{E}\left[T_i(n)\right] \leq \left(\frac{1}{D(p_i \parallel p^*)} + o(1)\right) \ln n$$

$$p_i = \text{probability distribution of machine } i$$

- Asymptotically achieves logarithmic regret.
- Proved that logarithmic regret is optimal.
- Agrawal (1995): Asymptotically optimal policies in terms of sample mean instead of KL divergence.

Upper Confidence Bound Algorithms



- Core idea: optimism in the face of uncertainty.
- Select arm with highest upper confidence bound.
- Assumption: distribution has support in $[0, 1]$.

UCB1

Initialization: Play each machine once.

Loop: Play the machine i maximizing

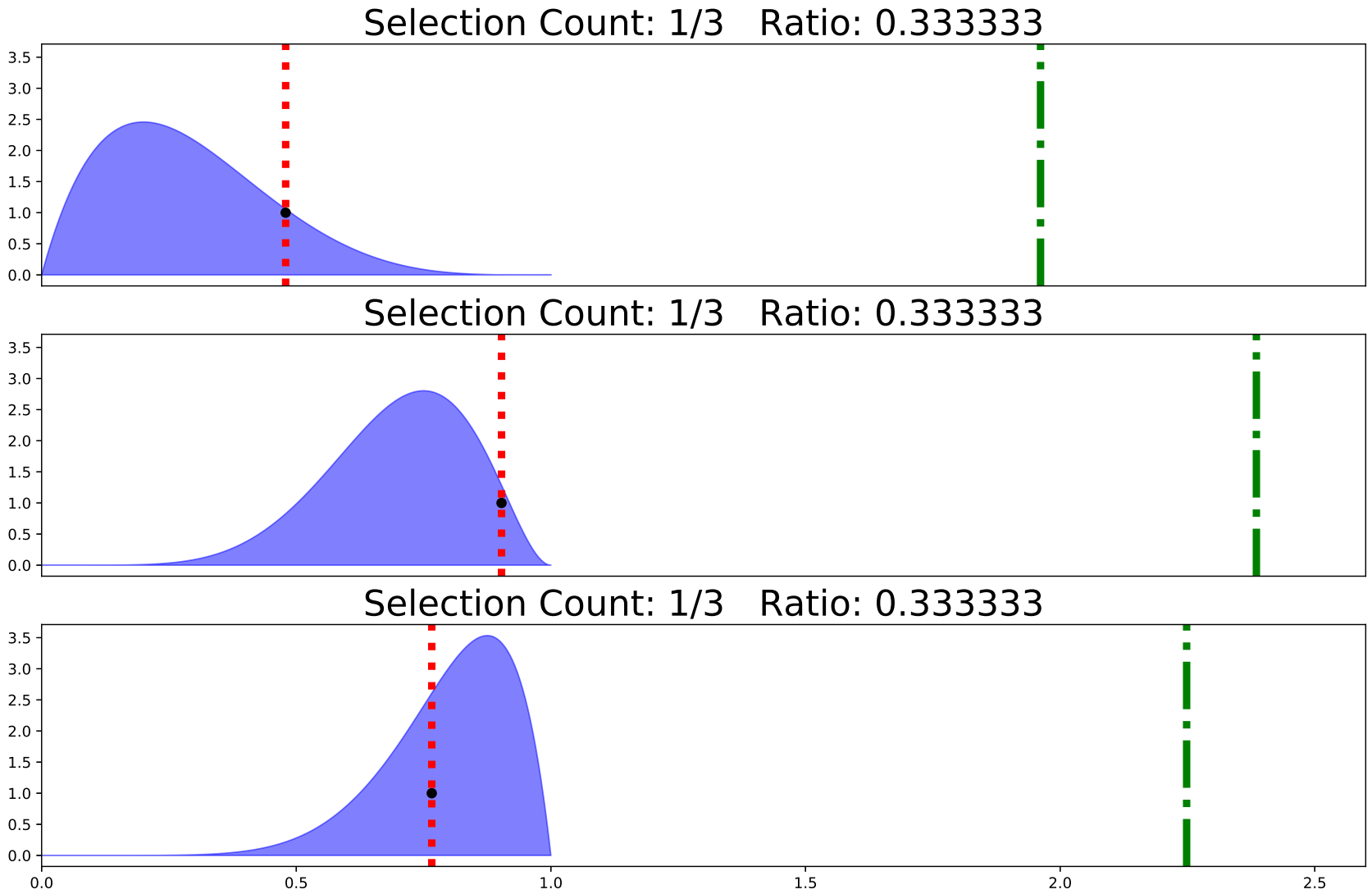
$$\bar{x}_i + \sqrt{\frac{2 \ln n}{n_i}}$$

$\bar{x}_i$ - Mean observed reward from machine i.

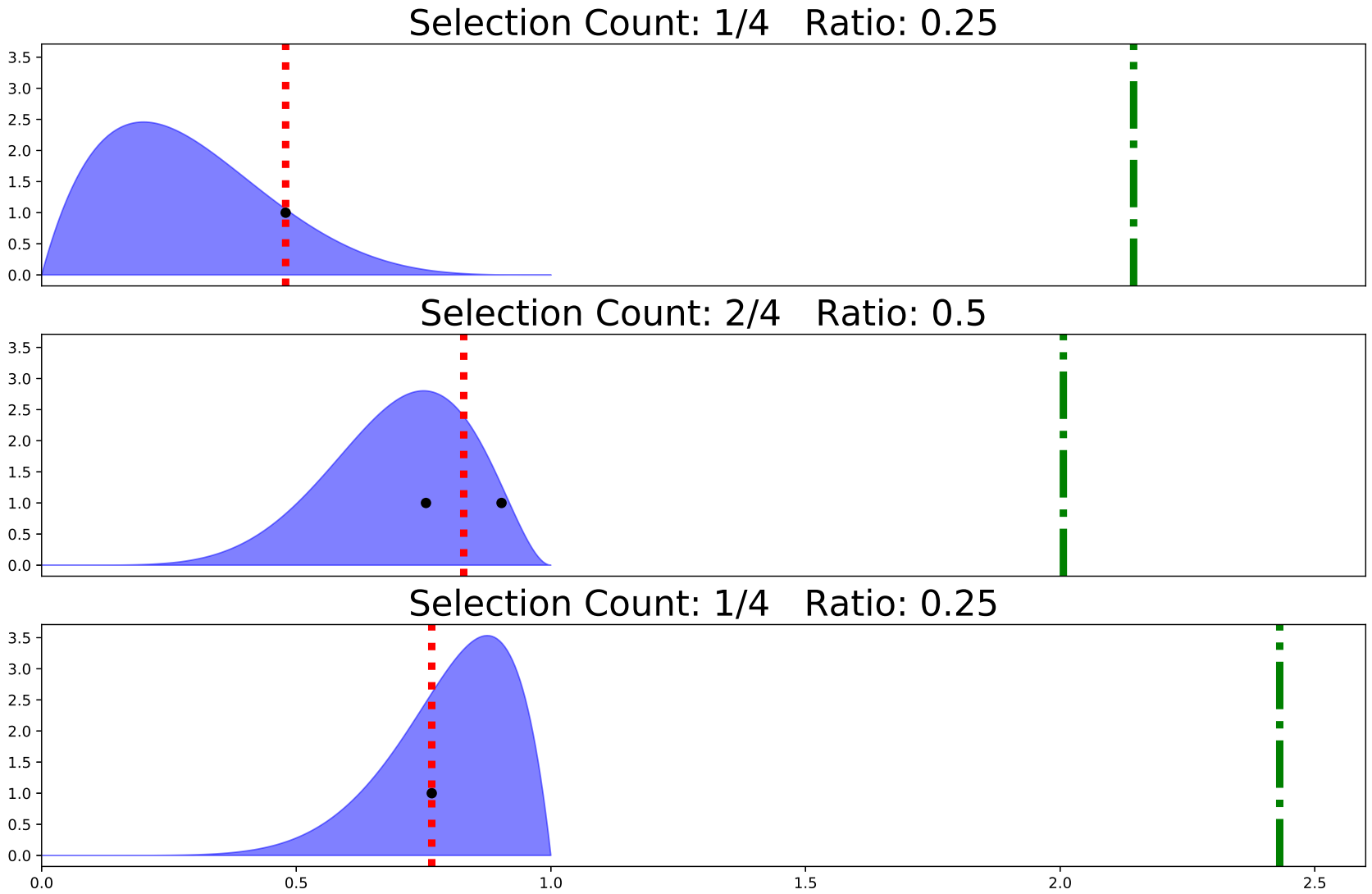
n_i - Number of times machine i has been played so far

n - Total number of plays done so far.

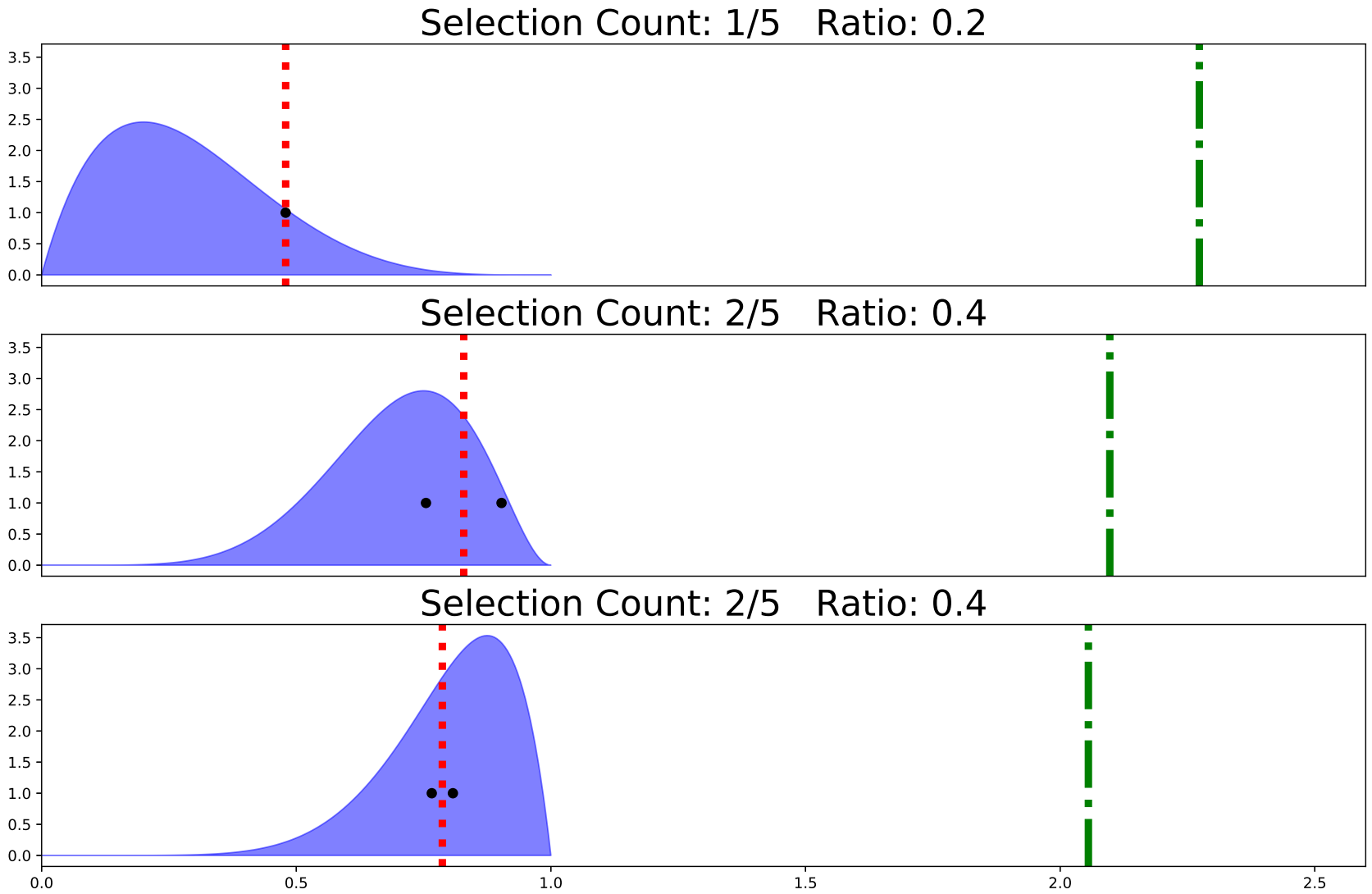
UCB1 Demo


Selection Count: 1/3 Ratio: 0.333333
Selection Count: 1/3 Ratio: 0.333333
Selection Count: 1/3 Ratio: 0.333333
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

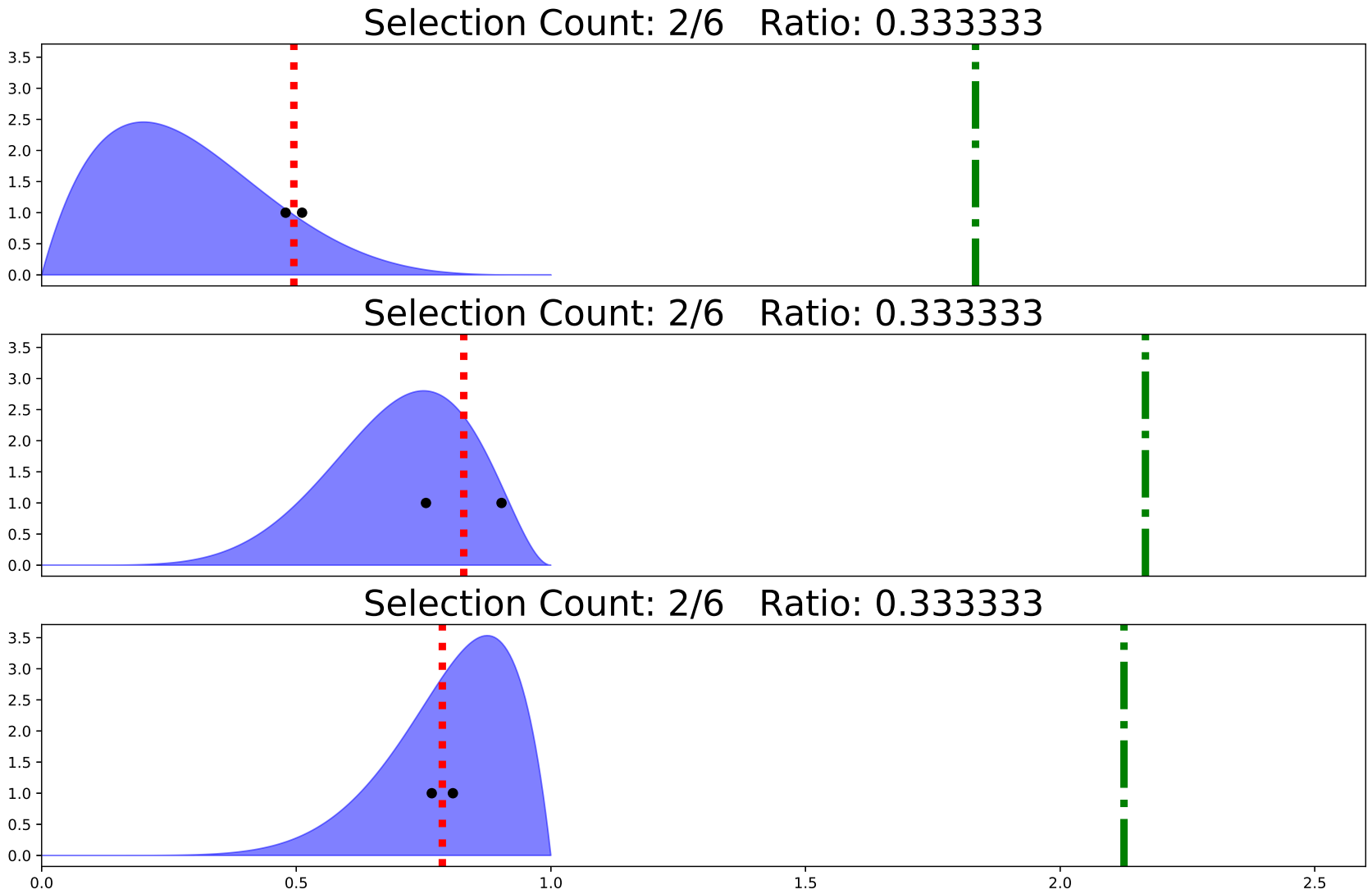
UCB1 Demo


Selection Count: 1/4 Ratio: 0.25
Selection Count: 2/4 Ratio: 0.5
Selection Count: 1/4 Ratio: 0.25
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

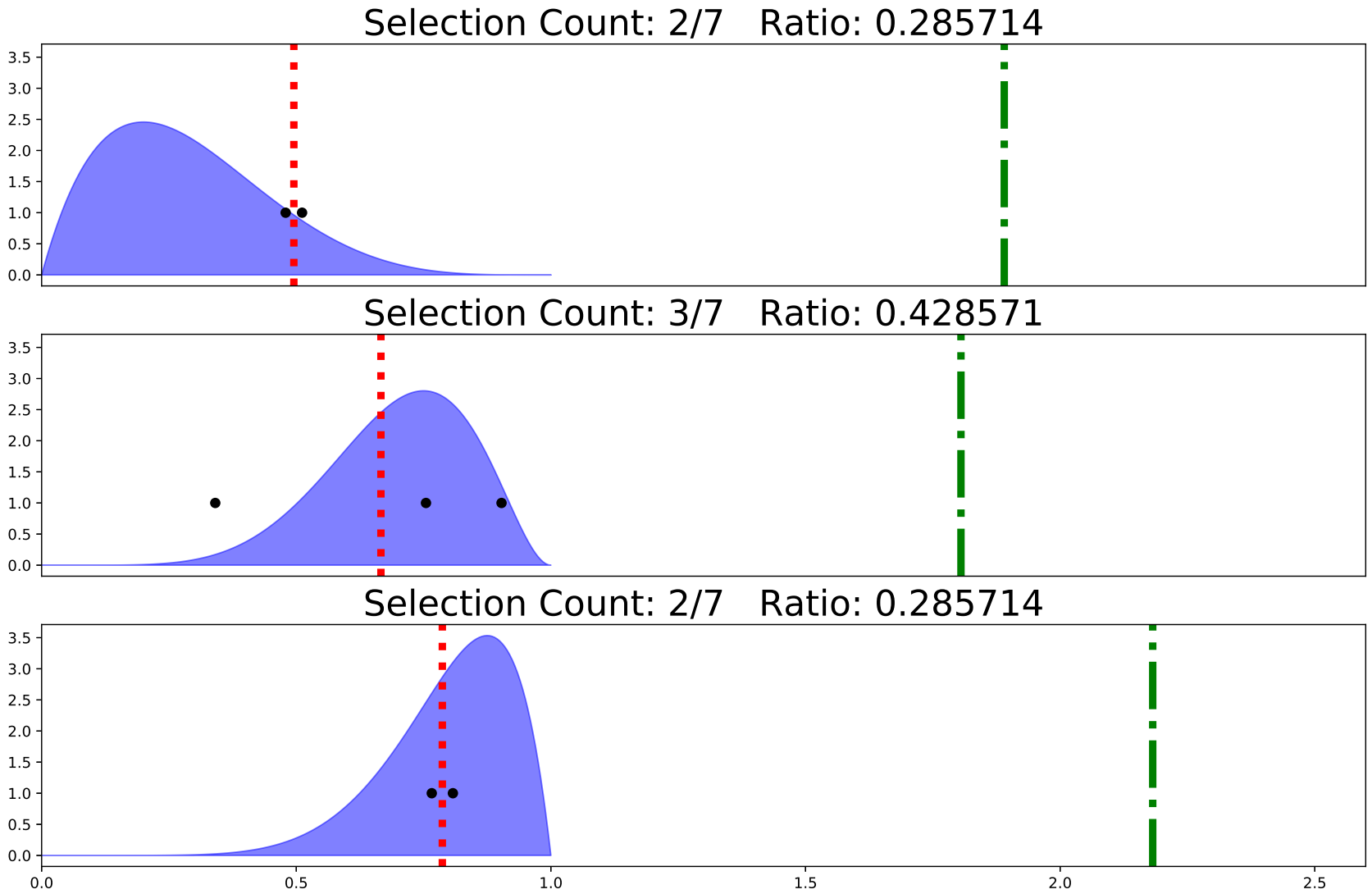
UCB1 Demo


Selection Count: 1/5 Ratio: 0.2
Selection Count: 2/5 Ratio: 0.4
Selection Count: 2/5 Ratio: 0.4
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

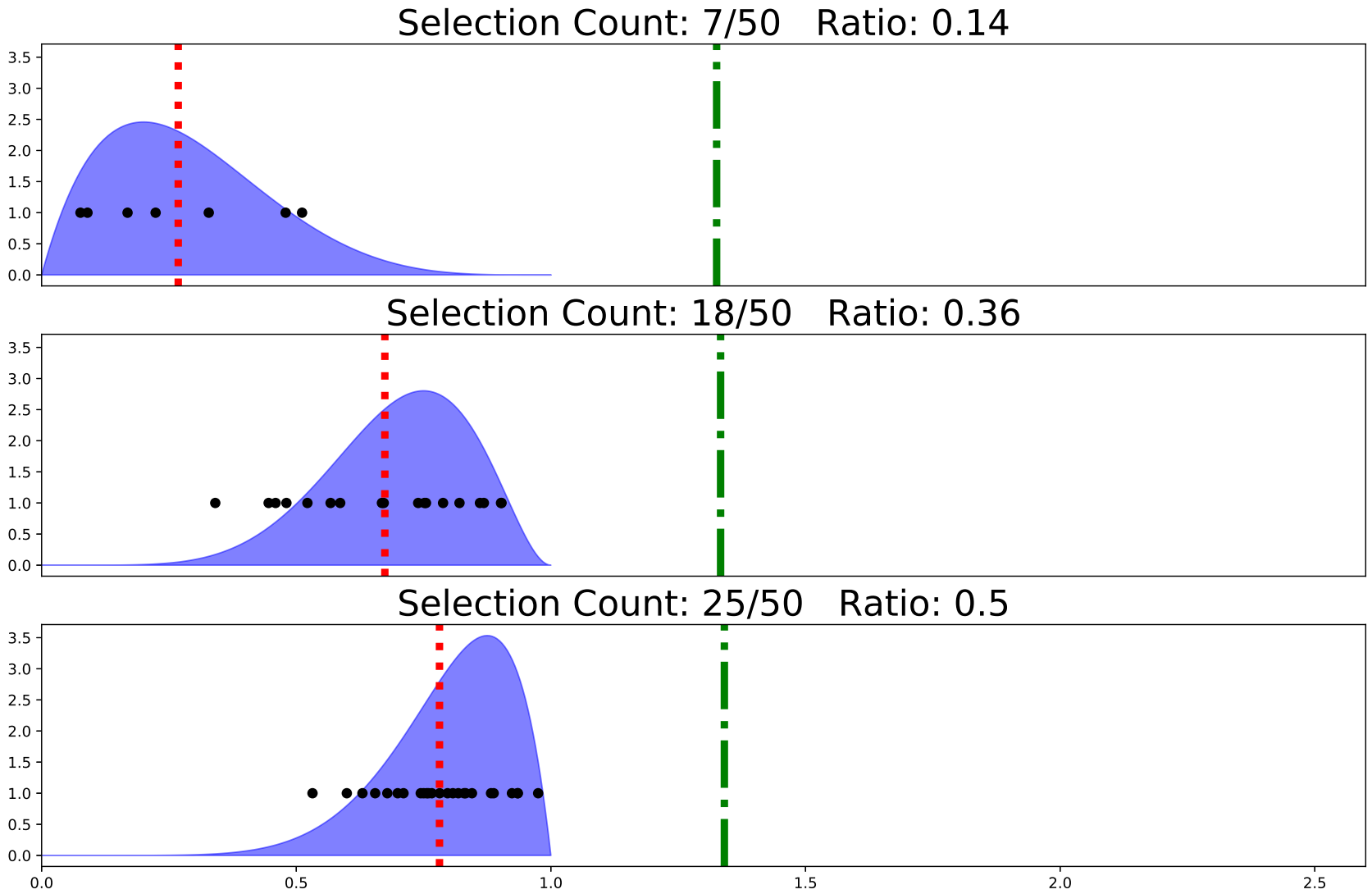
UCB1 Demo


Selection Count: 2/6 Ratio: 0.333333
Selection Count: 2/6 Ratio: 0.333333
Selection Count: 2/6 Ratio: 0.333333
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

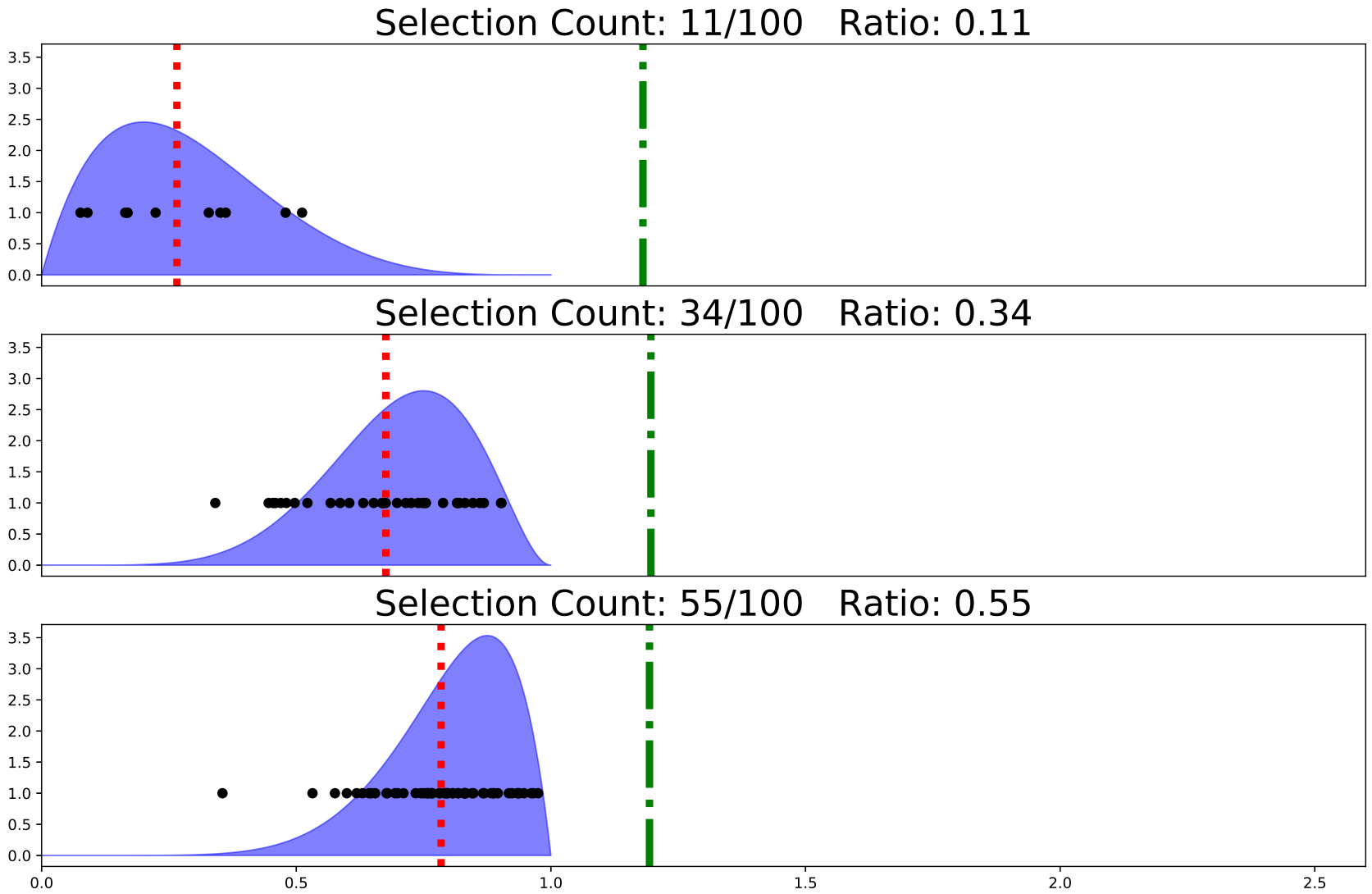
UCB1 Demo


Selection Count: 2/7 Ratio: 0.285714
Selection Count: 3/7 Ratio: 0.428571
Selection Count: 2/7 Ratio: 0.285714
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

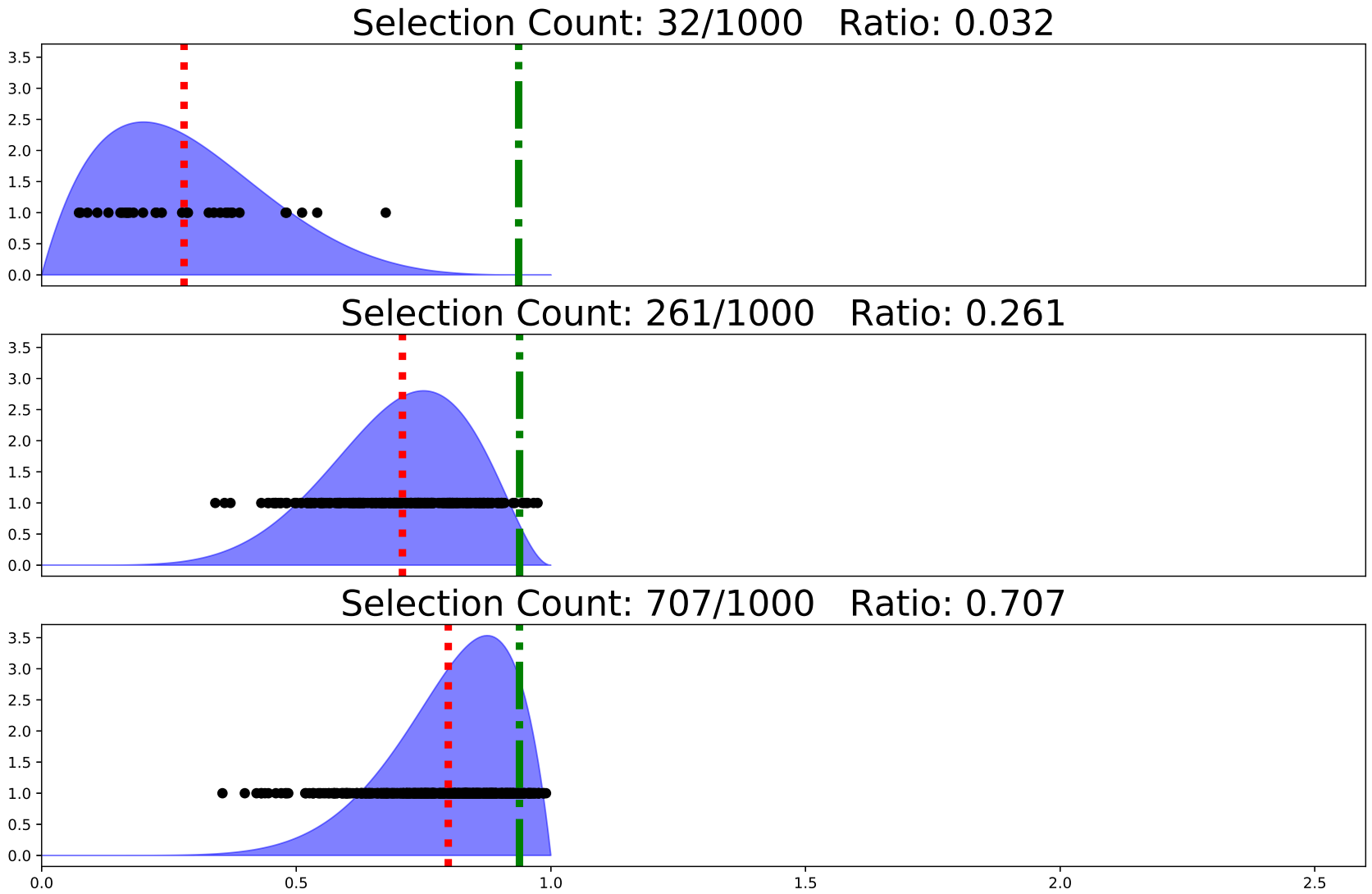
UCB1 Demo


Selection Count: 7/50 Ratio: 0.14
Selection Count: 18/50 Ratio: 0.36
Selection Count: 25/50 Ratio: 0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

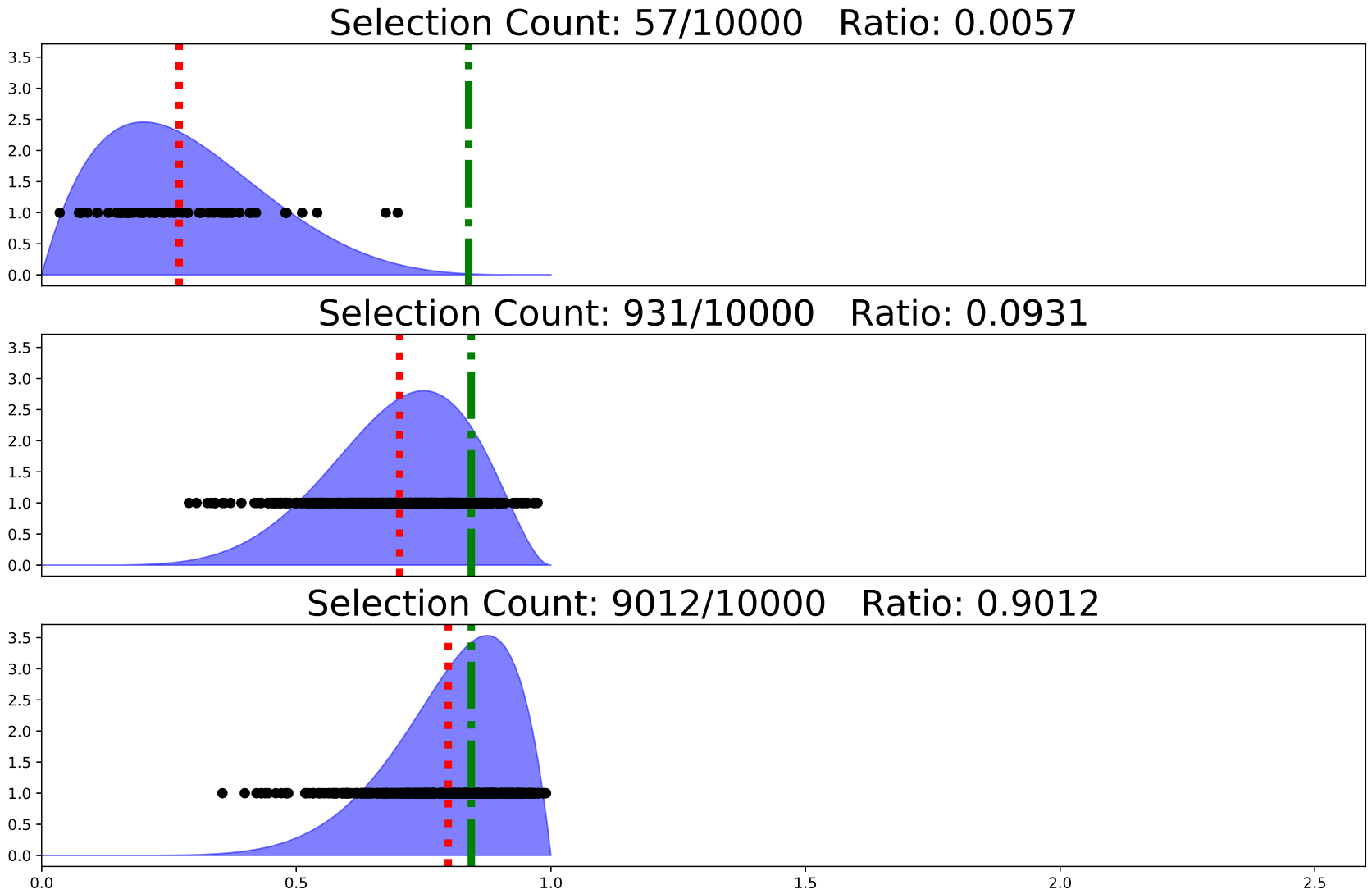
UCB1 Demo


Selection Count: 11/100 Ratio: 0.11
Selection Count: 34/100 Ratio: 0.34
Selection Count: 55/100 Ratio: 0.55
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

UCB1 Demo


Selection Count: 32/1000 Ratio: 0.032
Selection Count: 261/1000 Ratio: 0.261
Selection Count: 707/1000 Ratio: 0.707
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

UCB1 Demo


Selection Count: 57/10000 Ratio: 0.0057
Selection Count: 931/10000 Ratio: 0.0931
Selection Count: 9012/10000 Ratio: 0.9012
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

UCB1: Regret Bound (Theorem 1)

For all $K > 1$, if policy UCB1 is run on K machines having arbitrary reward distributions $P_1, \ldots, P_K$ with support in $[0, 1]$, then its expected regret after any number n of plays is at most

$$\left[8 \sum_{i:\mu_i<\mu^*} \left(\frac{\ln n}{\Delta_i}\right)\right] + \left(1 + \frac{\pi^2}{3}\right) \left(\sum_{i=1}^{K} \Delta_i\right)$$

UCB1: Definitions for Proof of Bound

- I_t - Indicator RV equal to the machine played at time t.
- $\bar{X}_{i,n}$ - RV of observed mean reward from n plays of machine i.

$$\bar{X}_{i,n} = \sum_{t=1}^{n} X_{i,t}$$

- An asterisk superscript refers to the (first) optimal machine. e.g. $T^*(n)$ and $\bar{X}^*_n$.
- Braces denote the indicator function of their contents.
- The number of plays of machine i after time n under UCB1 is therefore

$$T_i(n) = 1 + \sum_{t=K+1}^{n} \{I_t = i\}$$

UCB1: Proof of Regret Bound

$$
\begin{aligned}
T_i(n) &= 1 + \sum_{t=K+1}^{n} \{I_t = i\} \\
&\leq \ell + \sum_{\substack{t=K+1 \\ T_i(t-1) \geq 1}}^{n} \{I_t = i\}
\end{aligned}
$$

- Strategy: For every sub-optimal arm i, need to establish bound on total number of plays as a function of n.
- Assume we have seen ℓ plays of machine i so far and consider number of remaining plays.

UCB1: Proof of Regret Bound

$$T_i(n) \leq \ell + \sum_{\substack{t=K+1 \\ T_i(t-1) \geq \ell}}^{n} \{I_t = i\}$$

$$\leq \ell + \sum_{\substack{t=K+1 \\ T_i(t-1) \geq \ell}}^{n} \{\bar{X}^*_{T^*(t-1)} + c_{t-1,T^*(t-1)} \leq \bar{X}_{i,T_i(t-1)} + c_{t-1,T_i(t-1)}\}$$

- Let $c_{t,s} = \sqrt{\frac{2 \ln t}{s}}$ be the UCB offset term.
- Machine i is selected if its UCB $= \bar{X}_{i,T_i(t-1)} + c_{t-1,T_i(t-1)}$ is largest of all machines.
- In particular, must be larger than the UCB of the optimal machine.

UCB1: Proof of Regret Bound

$$T_i(n) \leq \ell + \sum_{\substack{t=K+1 \\ T_i(t-1) \geq \ell}}^{n} \{\bar{X}^*_{T^*(t-1)} + c_{t-1,T^*(t-1)} \leq \bar{X}_{i,T_i(t-1)} + c_{t-1,T_i(t-1)}\}$$

$$\leq \ell + \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \sum_{s_i=\ell}^{t-1} \{\bar{X}^*_s + c_{t,s} \leq \bar{X}_{i,s_i} + c_{t,s_i}\}$$

- Do not care about the particular number of times machine i and machine $*$ have been seen.
- Probability is upper bounded by summing over *all* possible assignments of $T^*(t-1) = s$ and $T_i(t-1) = s_i$.
- Relax the bounds on t as well.

UCB1: Proof of Regret Bound

$$T_i(n) \leq \ell + \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \sum_{s_i=\ell}^{t-1} \{\bar{X}_s^* + c_{t,s} \leq \bar{X}_{i,s_i} + c_{t,s_i}\}$$

The event $\bar{X}_s^* + c_{t,s} \leq \bar{X}_{i,s_i} + c_{t,s_i}$ implies at least one of the following:

$$\bar{X}_s^* \leq \mu^* - c_{t,s} \tag{1}$$

$$\bar{X}_{i,s_i} \geq \mu_i + c_{t,s_i} \tag{2}$$

$$\mu < \mu_i + 2c_{t,s_i} \tag{3}$$

Chernoff-Hoeffding Bound

Let $Z_1, \ldots Z_n$ be i.i.d random variables with mean μ and domain $[0, 1]$. Let $\bar{Z}_n = \frac{1}{n}(Z_1 + \cdots + Z_n)$. Then for all $a \geq 0$,

$$\mathsf{P}\left[\bar{Z}_n \geq \mu + \alpha\right] \leq e^{-2na^2} \qquad \mathsf{P}\left[\bar{Z}_n \leq \mu - \alpha\right] \leq e^{-2na^2}$$

Applied to inequalities (1) and (2), these give the bounds

$$\mathsf{P}\left[\bar{X}_s^* \leq \mu^* - c_{t,s}\right] \leq \exp\left(-2s\left(\frac{2\ln t}{s}\right)\right) = t^{-4}$$

$$\mathsf{P}\left[\bar{X}_{i,s_i} \geq \mu_i + c_{t,s_i}\right] \leq t^{-4}$$

UCB1: Proof of Regret Bound

The final inequality, $\mu^* < \mu_i + 2c_{t,s_i}$ is based on the width of the confidence interval. For $t < n$, it is false when s_i is large enough:

$$\Delta_i = \mu^* - \mu_i \leq 2\sqrt{\frac{2\ln t}{s_i}}$$

$$\Rightarrow \frac{\Delta_i^2}{4} \leq \frac{2\ln t}{s_i}$$

$$\Rightarrow s_i < \frac{8\ln t}{\Delta_i^2}$$

- In the regret bound summation, $s_i \geq \ell$ so we set $\ell = \frac{8\ln t}{\Delta_i^2} + 1$
- Inequality (3) then contributes nothing to the bound.

UCB1: Proof of Regret Bound

With $\ell = \frac{8 \ln t}{\Delta_i^2} + 1$ we have the bound on $\mathbb{E}[T_i(n)]$:

$$\begin{aligned}
\mathbb{E}[T_i(n)] &\leq \ell + \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \sum_{s_i=\ell}^{t-1} \left(\mathsf{P}\left[\bar{X}_s^* \leq \mu^* - c_{t,s} \right] + \mathsf{P}\left[\bar{X}_{i,s_i} \geq \mu_i + c_{t,s_i} \right] \right) \\
&\leq \ell + \sum_{t=1}^{\infty} \sum_{s=1}^{t} \sum_{s_i=1}^{t} 2t^{-4} \\
&\leq \frac{8 \ln n}{\Delta_i^2} + 1 + \frac{\pi^2}{3}
\end{aligned}$$

Substituted into the regret formula, this gives our bound.

UCB1-Tuned

- UCB1: $\mathbb{E}[T_i(n)] \leq \frac{8 \ln n}{\Delta_i^2} + 1 + \frac{\pi^2}{3}$
- Constant factor $\frac{8}{\Delta_i^2}$ is sub-optimal. Optimal: $\frac{1}{2\Delta_i^2}$.
- In practice the performance of UCB1 can be improved further by using the confidence bound

$$\bar{X}_{i,s} + \sqrt{\frac{\ln n}{n_i} \min\left\{\frac{1}{4}, V_i(n_i)\right\}}$$

where

$$V_i(s) = \left(\frac{1}{s}\sum_{\tau=1}^{s} X_{i,\tau}^2\right) - \bar{X}_{i,s}^2 + \sqrt{\frac{2 \ln t}{s}}$$

- No proof of regret bound.

Other Policies

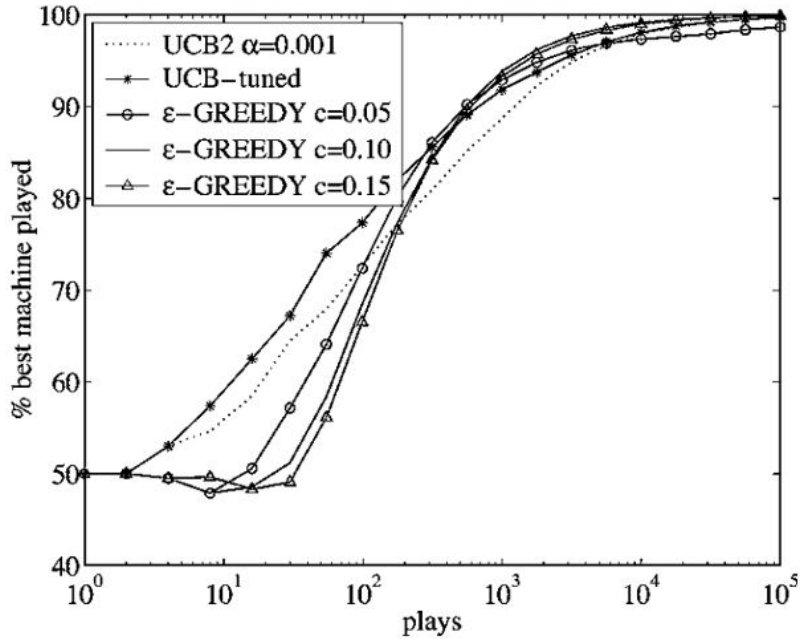
- UCB2: More complicated; gets arbitrarily close to optimal constant factor on regret.
- UCB1-NORMAL: UCB1 adapted for normally distributed rewards.
- ϵ_n-GREEDY: ϵ-greedy policy with decaying ϵ.

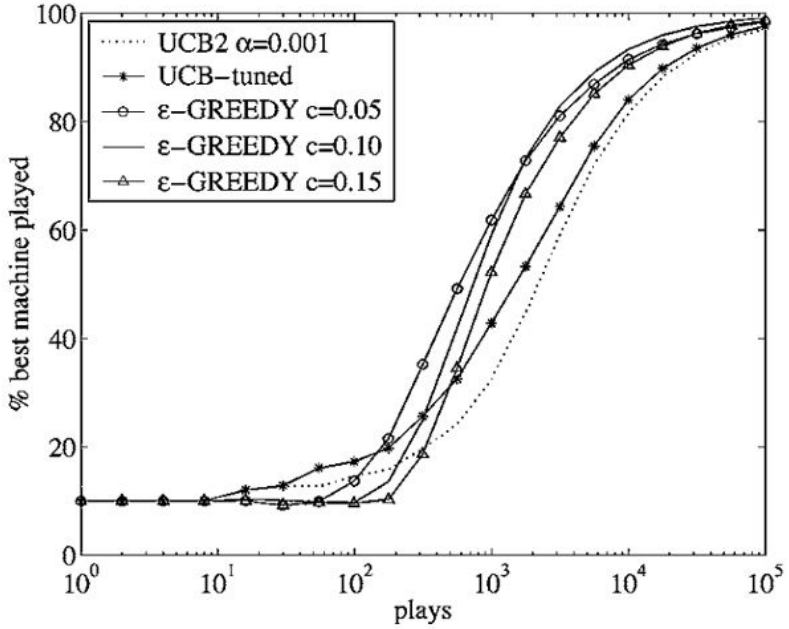
$$\epsilon_n = \min\left\{1, \frac{cK}{d^2 n}\right\}$$

where

$$c > 0 \qquad\qquad 0 < d \leq \min_{i:\mu_i<\mu^*} \Delta_i$$

EXPERIMENTS





Two machines:

Bernoulli 0.9 and 0.8

10 machines:

Bernoulli 0.9, 0.8, . . . , 0.8

Auer, Peter, Nicolo Cesa-Bianchi, and Paul Fischer. "Finite-time analysis of the multiarmed bandit problem." Machine learning 47.2-3 (2002): 235-256.

Comparisons

- UCB1-Tuned nearly always far outperforms UCB1
- ϵ_n-GREEDY performs very well if tuned correctly, poorly otherwise. Also poorly if there are many suboptimal machines.
- UCB1-Tuned is nearly as good as the best ϵ_n-GREEDY without any tuning required.
- UCB2 is similar to UCB1-Tuned but slightly worse.

A Tutorial on Thompson Sampling

Daniel J.Russo, Benjamin Van Roy, Abbas Kazerouni, Ian, Osband, and Zheng Wen

Presenters

Mingjie Mai Feng Chi

October 20, 2017

Outline

- Example problems
- Algorithms and applications to example problems
- Approximations for complex model
- Practical modeling considerations
- Limitations
- Further example: Reinforcement learning in Markov Decision Problems

Exploitation vs Exploration

- Restaurant Selection
- Online Banner Advertisements
- Oil Drilling
- Game Playing
  - Multi-armed bandit problem

Formal Bandit problems

Bandit problems can be seen as a generalization of supervised learning, where we:

- Actions $x_t \in \mathcal{X}$
- Unknown probability distribution over rewards: $(p_1, \ldots, p_K)$
- Each step, pick one x_t
- observe response y_t
- receive the instantaneous reward $r_t = r(y_t)$
- the goal is to maximize mean cumulative reward $\mathbb{E} \sum_t r_t$

Regret

- The optimal action is $x_t^* = \max_{x_t \in \mathcal{X}} \mathbb{E}[r|x_t]$
- The *regret* is the opportunity loss for one step:
  $\mathbb{E}[\mathbb{E}[r|x_t^*] - \mathbb{E}[r|x_t]]$
- The *total regret* is the total opportunity loss :
  $\mathbb{E}[\sum_{\tau=1}^{t}(\mathbb{E}[r|x_\tau^*] - \mathbb{E}[r|x_\tau])]$
- Maximize cumulative reward $\equiv$ minimize total regret

Bernoulli Bandit

- Action: $x_t \in \{1, 2, ..., K\}$
- Success probabilities: $(\theta_1, ..., \theta_K)$, where $\theta_k \in [0, 1]$
- Observation:

$$y_t = \begin{cases} 1 & \text{w.p. } \theta_k \\ 0 & \text{otherwise} \end{cases}$$

- Reward: $r_t(y_t) = y_t$
- Prior belief: $\theta_k \sim \beta(\alpha_k, \beta_k)$

Algorithms

The data observed up to time t: $\mathbb{H}_t = \{(x_1, y_1), ..., (x_{t-1}, y_{t-1})\}$

- Greedy
  - $\hat{\theta} = \mathbb{E}[\theta|\mathbb{H}_{t-1}]$
  - $x_t = \text{argmax}_k \, \hat{\theta}_k$
- ϵ-Greedy
  - $\hat{\theta} = \mathbb{E}[\theta|\mathbb{H}_{t-1}]$
  - $x_t = \begin{cases} \text{argmax}_k \, \hat{\theta}_k & \text{w.p. } 1 - \epsilon \\ unif(\{1, \ldots, K\}) & \text{otherwise} \end{cases}$
- Thompson Sampling
  - $\hat{\theta}$ is sampled from $\mathsf{P}(\theta_k|\mathbb{H}_{t-1})$
  - $x_t = \text{argmax}_k \, \hat{\theta}_k$

COMPUTING POSTERIORS WITH BERNOULLI BANDIT

- Prior belief: $\theta_k \sim \beta(\alpha_k, \beta_k)$
- At each time period, apply action x_t, reward $r_t \in \{0, 1\}$ is generated with success probability $\mathbb{P}(r_t = 1 | x_t, \theta) = \theta_{x_t}$
- Update distribution according to Baye's rule.
- due to conjugacy property of beta distribution we have:

$$(\alpha_k, \beta_k) \leftarrow \begin{cases} (\alpha_k, \beta_k) & \text{if } x_t \neq k \\ (\alpha_k, \beta_k) + (r_t, 1 - r_t) & \text{if } x_t = k. \end{cases}$$

Side By Side Comparison

Algorithm 1 BernGreedy(K, α, β)

for $t = 1, 2, \ldots$ **do**
  #estimate model:
  for $k = 1, \ldots, K$ **do**
    $\hat{\theta}_k \leftarrow \alpha_k / (\alpha_k + \beta_k)$
  end for

  #select and apply action:
  $x_t \leftarrow \text{argmax}_k \, \hat{\theta}_k$
  Apply x_t and observe r_t

  #update distribution:
  $(\alpha_{x_t}, \beta_{x_t}) \leftarrow (\alpha_{x_t}, \beta_{x_t}) + (r_t, 1 - r_t)$
end for

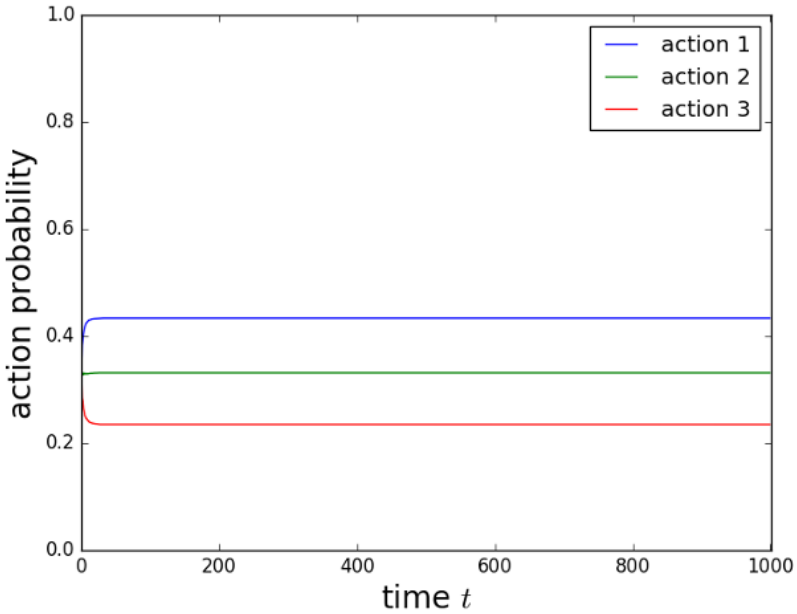
Algorithm 2 BernThompson(K, α, β)

for $t = 1, 2, \ldots$ **do**
  #sample model:
  for $k = 1, \ldots, K$ **do**
    Sample $\hat{\theta}_k \sim \text{beta}(\alpha_k, \beta_k)$
  end for

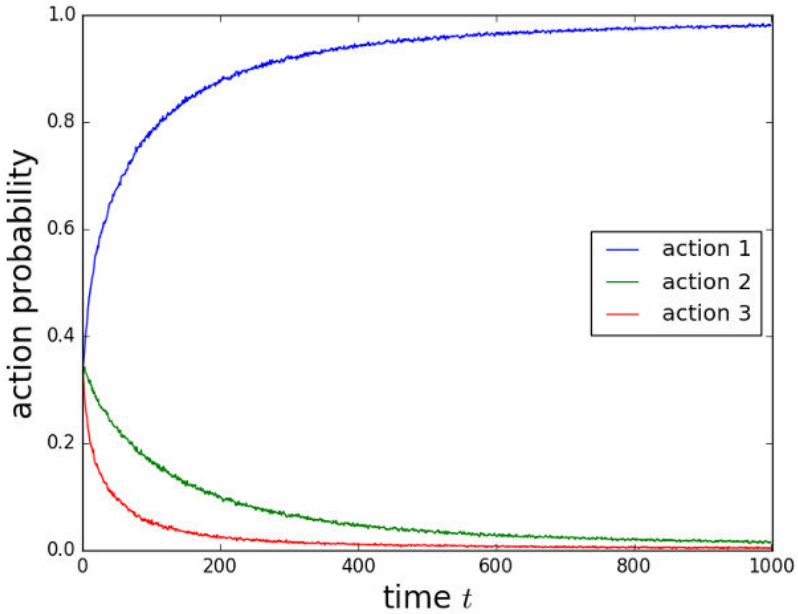
  #select and apply action:
  $x_t \leftarrow \text{argmax}_k \, \hat{\theta}_k$
  Apply x_t and observe r_t

  #update distribution:
  $(\alpha_{x_t}, \beta_{x_t}) \leftarrow (\alpha_{x_t}, \beta_{x_t}) + (r_t, 1 - r_t)$
end for

PERFORMANCE COMPARISON



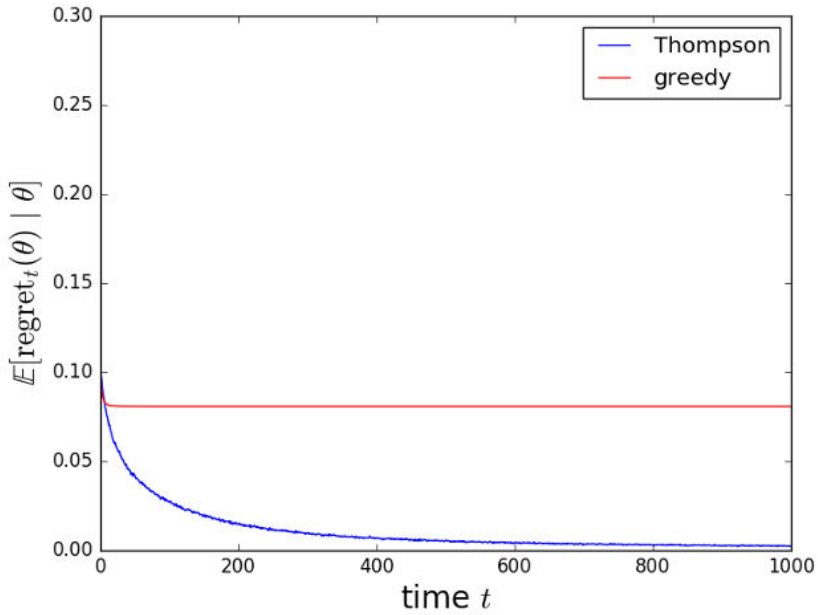
(a) greedy algorithm



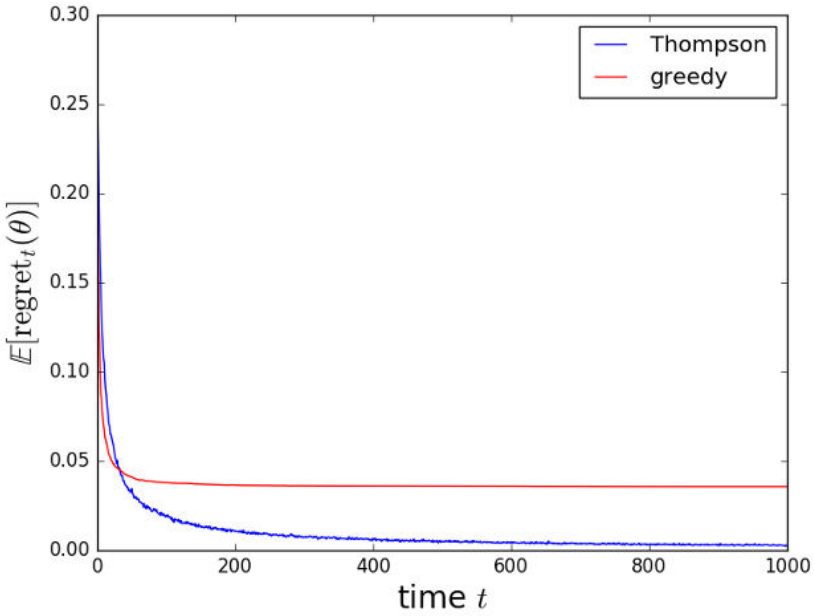
(b) Thompson sampling

Figure: Probability that the greedy algorithm and Thompson sampling selects an action. $\theta_1 = 0.9, \theta_2 = 0.8, \theta_3 = 0.7$

PERFORMANCE COMPARISON



(a) $\theta = (0.9, 0.8, 0.7)$



(b) average over random θ

Figure: Regret from applying greedy and Thompson sampling algorithms to the three-armed Bernoulli bandit.

ONLINE SHORTEST PATH

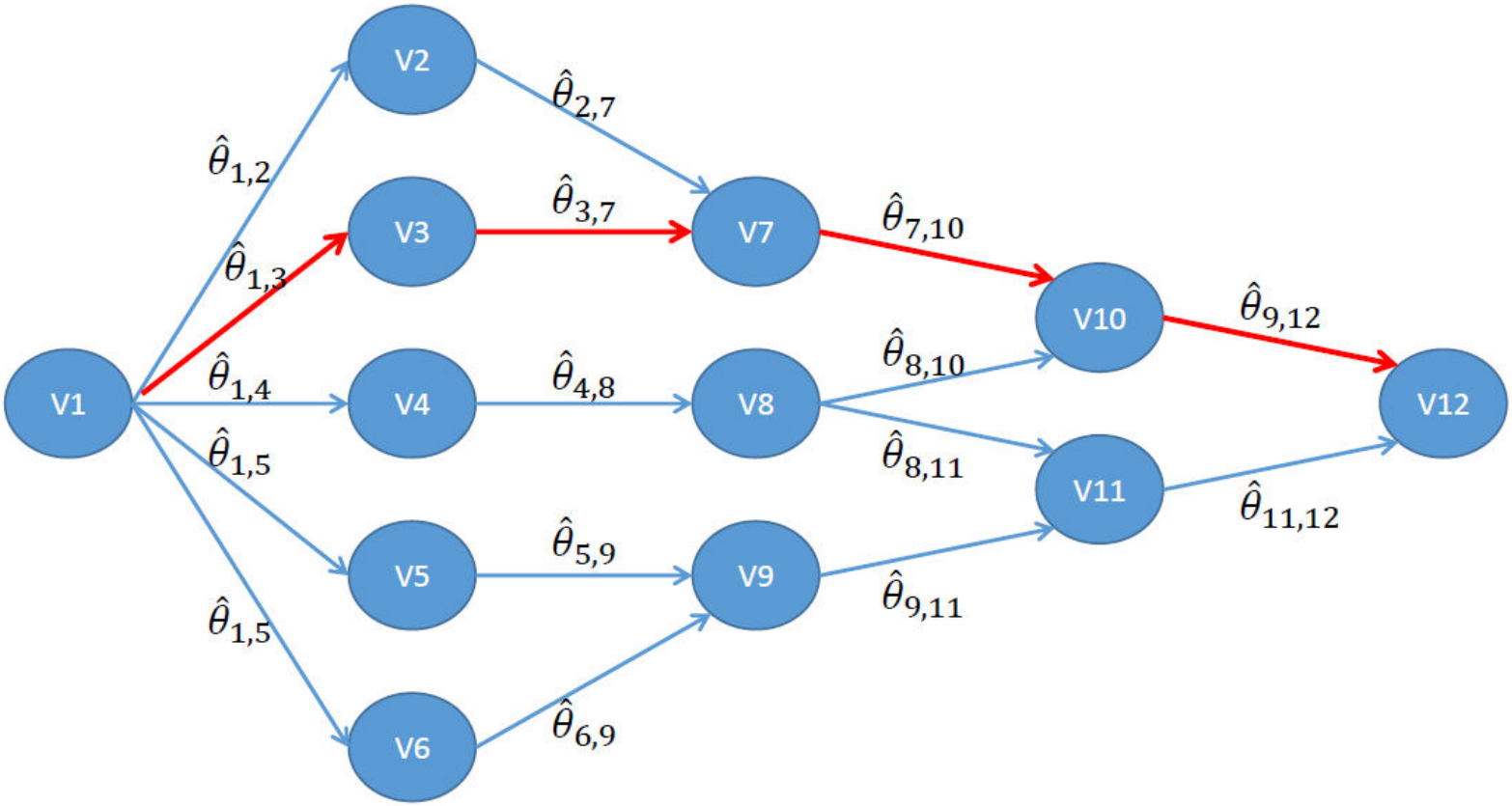


Figure: Shortest Path Problem.

Online Shortest Path - Independent travel time

Given a graph $G = (V, E, v_s, v_d)$, where $v_s, v_d \in V$, we have that

- Mean travel time: θ_e for $e \in E$,
- Action: $x_t = (e_1, ..., e_M)$, a path from v_s to v_d
- Observation: $(y_{t,e_1}|\theta_{e_1}, ..., y_{t,e_M}|\theta_{e_M})$ are independent, where $\ln(y_{t,e}|\theta_e) \sim N(\ln \theta_e - \frac{\tilde{\sigma}^2}{2}, \tilde{\sigma}^2)$, so that $\mathbb{E}[y_{t,e}|\theta_e] = \theta_e$
- Reward: $r_t = -\sum_{e \in x_t} y_{t,e}$
- Prior belief: $\ln(\theta_e) \sim N(\mu_e, \sigma_e^2)$ also independent.

Online Shortest Path - Independent travel time

- At each tth iteration with posterior parameters (μ_e, σ_e) for each $e \in E$.
  - greedy algorithm: $\hat{\theta}_e = \mathbb{E}_p[\theta_e] = e^{\mu_e + \sigma_e^2/2}$
  - Thompson sampling: draw $\hat{\theta}_e \sim logNormal(\mu_e, \sigma_e^2)$
- pick an action x to maximize $\mathbb{E}_{q_{\hat{\theta}}}[r(y_t)|x_t = x] = -\sum_{e \in x_t} \hat{\theta}_e$
  - can be solved via Dijkstra's algorithm
- observe $y_{t,e}$, and update parameters

$$(\mu_e, \sigma_e^2) \leftarrow \left(\frac{\frac{1}{\sigma_e^2}\mu_e + \frac{1}{\tilde{\sigma}^2}\left(\ln y_{t,e} + \frac{\tilde{\sigma}^2}{2}\right)}{\frac{1}{\sigma_e^2} + \frac{1}{\tilde{\sigma}^2}}, \frac{1}{\frac{1}{\sigma_e^2} + \frac{1}{\tilde{\sigma}^2}} \right)$$

Binomial bridge

- apply above algorithm to a Binomial bridge with six stages with 184,756 paths.
- $\mu_e = -\frac{1}{2}, \sigma_e^2 = 1$ so that $\mathbb{E}[\theta_e] = 1$, for each $e \in E$, and $\tilde{\sigma}^2 = 1$

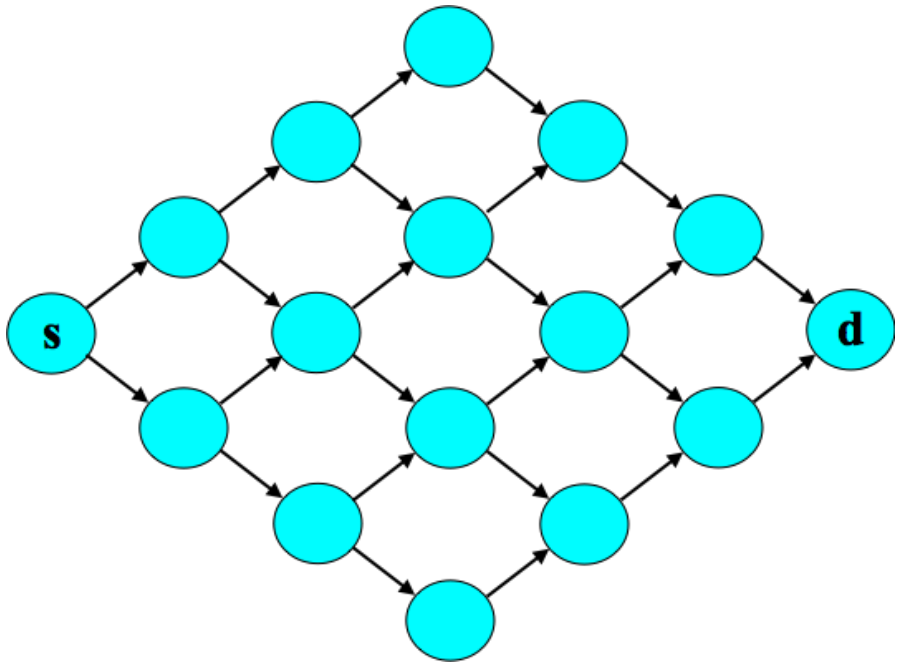
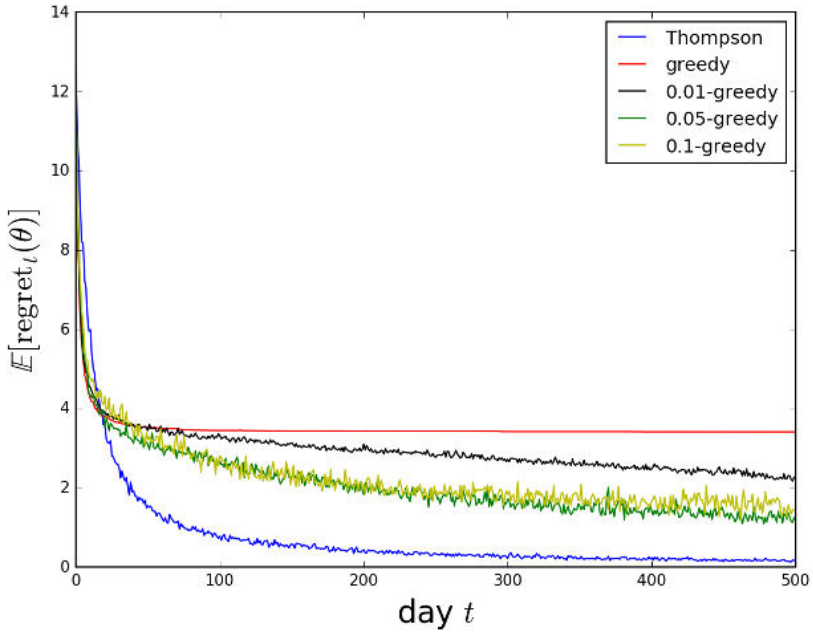
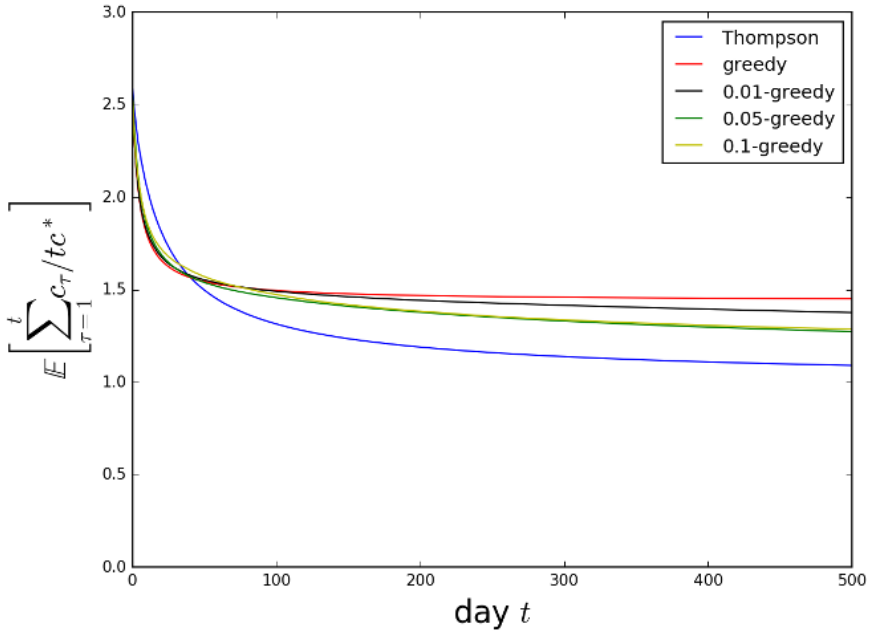


Figure: A binomial bridge with six stages.



(a) regret



(b) total travel time/optimal

Figure: Performance of Thompson sampling and ϵ-greedy algorithms in the shortest path problem.

ONLINE SHORTEST PATH - CORRELATED TRAVEL TIME

- independent $\theta_e \sim logNormal(\mu_e, \sigma_e^2)$
- $y_{t,e} = \zeta_{t,e}\eta_t\nu_{t,\ell(e)}\theta_e$
  - $\zeta_{t,e}$ is an idiosyncratic factor associated with edge e (road construction/closure, accident, etc)
  - η_t a common factor to all edges (weather, etc).
  - $\ell(e)$ indicates whether edge e resides in the lower half of the bridge
  - $\nu_{t,0}, \nu_{t,1}$ are factors bear a common influence on edges in the upper or lower halves (signal problems)

ONLINE SHORTEST PATH - CORRELATED TRAVEL TIME

- Prior setup:
  - take $\zeta_{t,e}$, η_t, $\nu_{t,1}$, $\nu_{t,0}$ to be independent $logNormal(\tilde{\sigma}^2/6, \tilde{\sigma}^2/3)$.
  - only need to estimate θ_e, and marginal $y_{t,e}|\theta$ is the same as independent case, but the joint distribution over $y_t|\theta$ differs.
- Correlated observations induce dependencies in posterior, although mean travel times are independent.

ONLINE SHORTEST PATH - CORRELATED TRAVEL TIME

- Let $\phi, z_t \in \mathbb{R}^N$ be defined by

$$\phi_e = \ln \theta_e \qquad \text{and} \qquad z_{t,e} = \begin{cases} \ln y_{t,e} & \text{if } e \in x_t \\ 0 & \text{otherwise.} \end{cases}$$

- Define a $|x_t| \times |x_t|$ covariance matrix $\tilde{\Sigma}$ with elements

$$\tilde{\Sigma}_{e,e'} = \begin{cases} \tilde{\sigma}^2 & \text{for } e = e' \\ 2\tilde{\sigma}^2/3 & \text{for } e \neq e', \ell(e) = \ell(e') \\ \tilde{\sigma}^2/3 & \text{otherwise,} \end{cases}$$

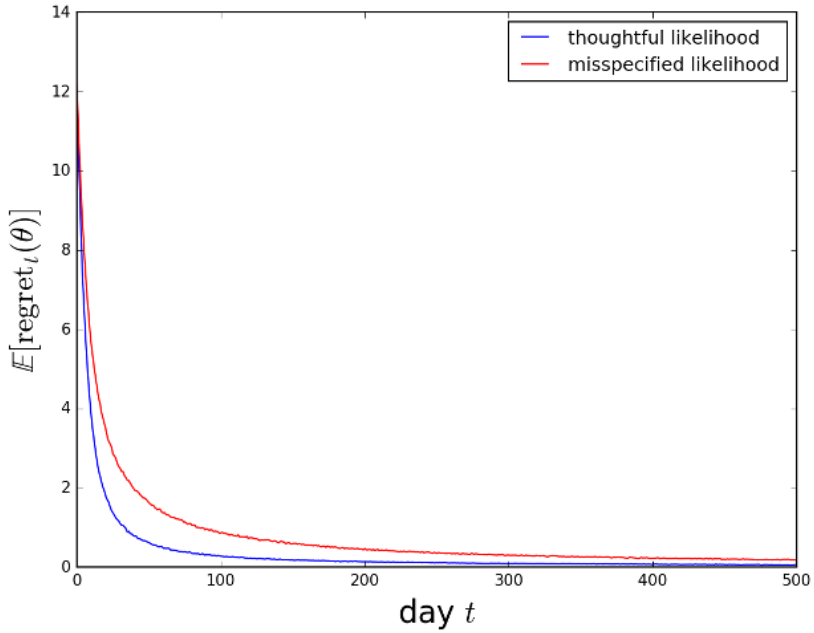
- for $e, e' \in x_t$, and a $N \times N$ concentration matrix

$$\tilde{C}_{e,e'} = \begin{cases} \tilde{\Sigma}^{-1}_{e,e'} & \text{if } e, e' \in x_t \\ 0 & \text{otherwise,} \end{cases}$$

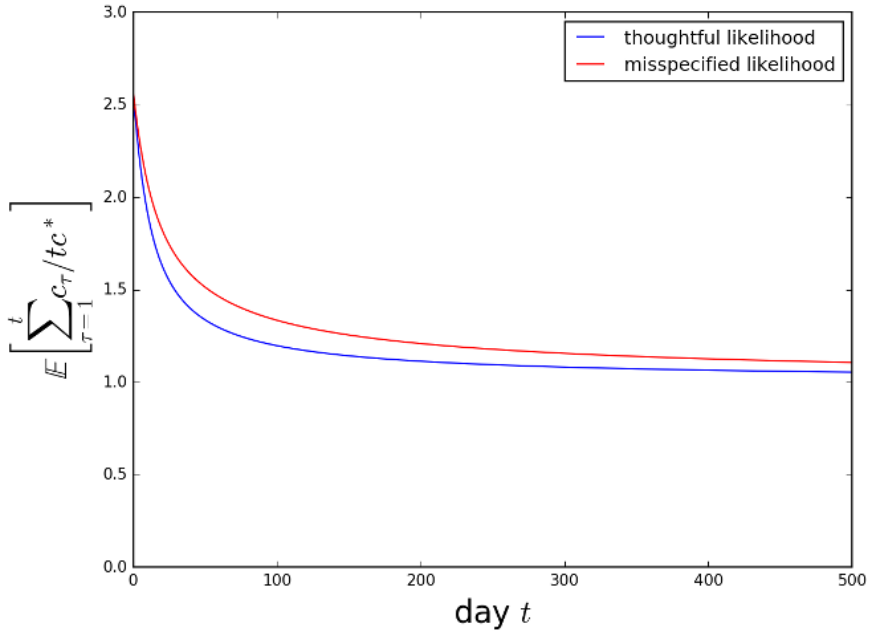
ONLINE SHORTEST PATH - CORRELATED TRAVEL TIME

- Apply Thompson sampling
  - Each tth iteration, sample a vector $\hat{\phi}$ from $N(\mu, \Sigma)$, then setting $\hat{\theta}_e = \hat{\phi}_e$ for each $e \in E$.
  - An action x is selected to maximize $\mathbb{E}_{q_{\hat{\theta}}}[r(y_t)|x_t = x] = -\sum_{e \in x_t} \hat{\theta}_e$, using Djikstra's algorithm or an alternative.
  - for $e, e' \in E$. Then, the posterior distribution of ϕ is normal with a mean vector μ and covariance matrix Σ that can be updated according to

$$(\mu, \Sigma) \leftarrow \left(\left(\Sigma^{-1} + \tilde{C} \right)^{-1} \left(\Sigma^{-1} \mu + \tilde{C} z_t \right), \left(\Sigma^{-1} + \tilde{C} \right)^{-1} \right).$$



(a) regret



(b) total travel time/optimal

Figure: Performance of two versions of Thompson sampling in the shortest path problem with correlated travel time.

Approximations of posterior sampling for complex model

- Gibbs Sampling
- Langevin Monte Carlo
- Sampling from a Laplace Approximation
- Bootstrapping

Gibbs Sampling

- History: $\mathbb{H}_{t-1} = ((x_1, y_1), \ldots, (x_{t-1}, y_{t-1}))$
- Starts with an initial guess θ^0
- For each n^{th} iteration, sample each k^{th} component according to $\hat{\theta}_k^n \sim f_{t-1}^{n,k}(\theta_k)$

$$f_{t-1}^{n,k}(\theta_k) \propto f_{t-1}((\hat{\theta}_1^n, \ldots, \hat{\theta}_{k-1}^n, \theta_k, \hat{\theta}_{k+1}^{n-1}, \ldots, \hat{\theta}_K^{n-1}))$$

- After N iterations, $\hat{\theta}^N$ is taken to be the approximate posterior sample

Langevin Monte Carlo

- Let g be the posterior distribution
- Euler method for stimulating Langevin daynmics:

$$\theta_{n+1} = \theta_n + \epsilon \nabla \ln g(\theta_n) + \sqrt{\epsilon} W_n \qquad n \in \mathbb{N}$$

- $W_1, W_2, \cdots$ are i.i.d. standard normal random variables and $\epsilon > 0$ is a small step size
- Stochastic gradient Langevin Monte Carlo: use sampled minibatches of data to compute approximate

Sampling from a Laplace Approximation

- Assume posterior g is unimodal and its log density $\ln g(\theta)$ is strictly concave around its mode $\overline{\theta}$
- A second-order Taylor approximation to the log-density gives

$$\ln g(\theta) \approx \ln g(\overline{\theta}) - \frac{1}{2}(\theta - \overline{\theta})^\top C(\theta - \overline{\theta}),$$

where

$$C = -\nabla^2 \ln g(\overline{\theta}).$$

- Approximation to the density g using a Gaussian distribution with mean $\overline{\theta}$ and covariance C^{-1}

$$\tilde{g}(\theta) = \sqrt{|C/2\pi|} e^{-\frac{1}{2}(\theta-\overline{\theta})^\top C(\theta-\overline{\theta})}$$

Bootstrapping

- History: $\mathbb{H}_{t-1} = ((x_1, y_1), \ldots, (x_{t-1}, y_{t-1}))$
- Uniformly sample with replacement from $\mathbb{H}_{t-1}$
- Hypothetical history $\hat{\mathbb{H}}_{t-1} = ((\hat{x}_1, \hat{y}_1), \ldots, (\hat{x}_{t-1}, \hat{y}_{t-1}))$
- Maximize the likelihood of θ under $\hat{\mathbb{H}}_{t-1}$

Bernoulli bandit

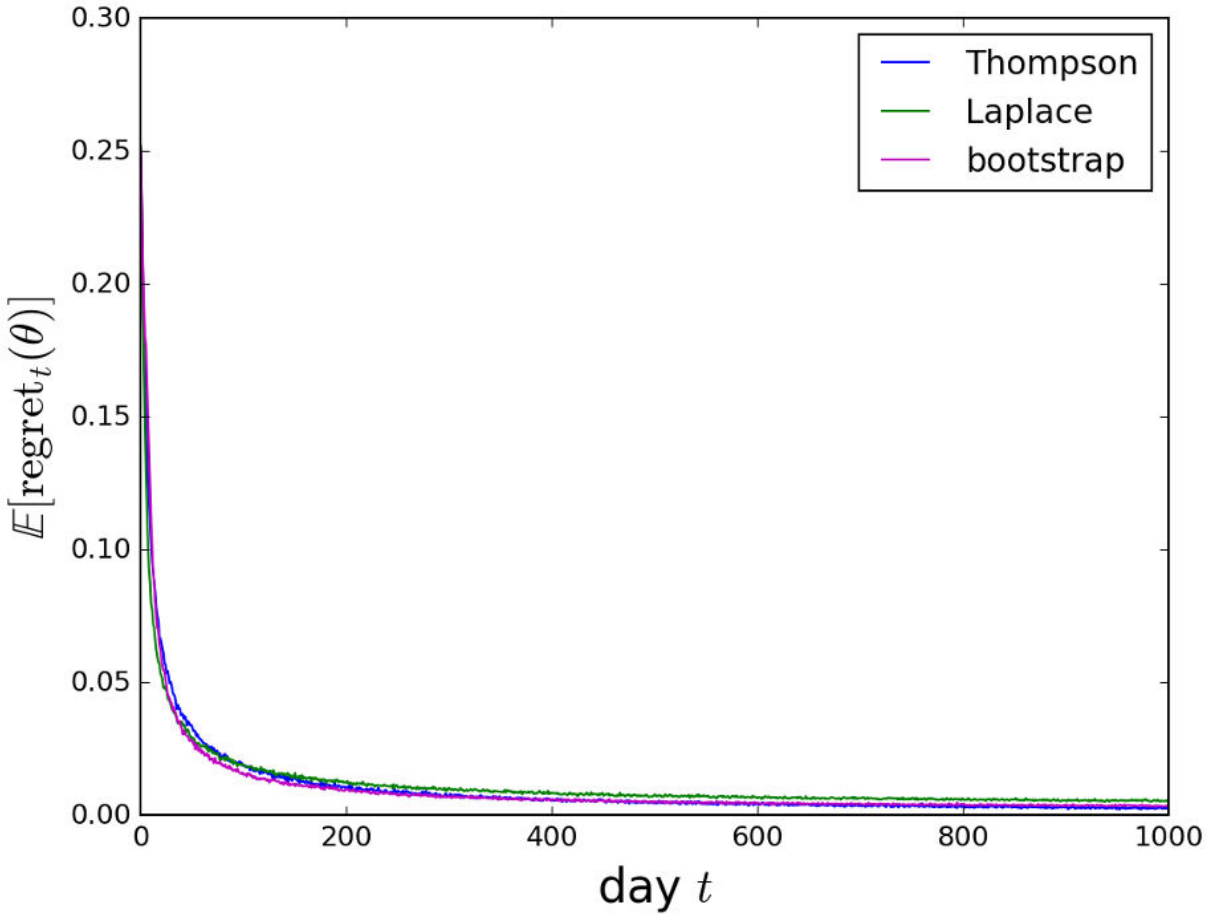


Figure: Regret of approximation methods versus exact Thompson sampling (Bernolli bandit)

Online shortest path

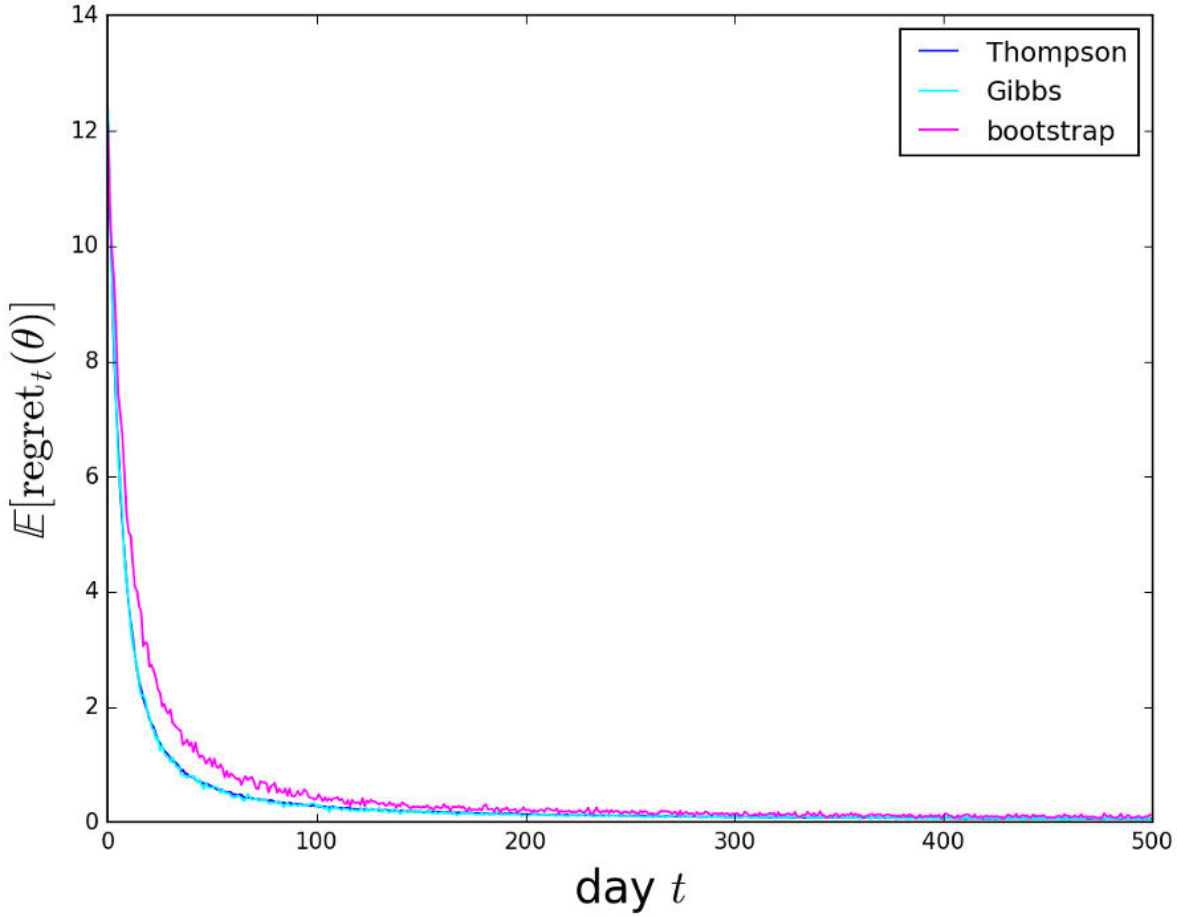


Figure: Regret of approximation methods versus exact Thompson sampling (online shortest path)

Practical modeling considerations

- Prior distribution specification
- Constraints and context
- Nonstationary systems

Prior distribution specification

- Prior: a distribution over plausible values
- Misspecified prior vs informative prior
- Thoughtful choice of prior based on past experience can improve learning performance

Constraints, context and caution

- Time-varying constraints
  - e.g. road closure in online shortest path problem
  - Use a sequence of action sets $\mathcal{X}_t$ that constraint action x_t and modify the optimization problem
- Contextual online decision problems
  - e.g. Agent observe weather report z_t before selecting a path x_t
  - Augment the action space and introduce time-varying constraint sets
- Caution against poor performance
  - e.g. $\mathcal{X}_t = \{x \in \mathcal{X} : \mathbb{E}[r_t | x_t = x] \geq \underline{r}\}$

Nonstationary system

- Model parameters θ that are not constant over time
- Ignore historical observations made beyond some number τ of the time periods in the past
- Model evolution of a belief distribution
  - In the context of Bernoulli bandit,

$$(\alpha_k, \beta_k) \leftarrow \begin{cases} ((1-\gamma)\alpha_k + \gamma\overline{\alpha}, (1-\gamma)\beta_k + \gamma\overline{\beta}) & \text{if } x_t \neq k \\ ((1-\gamma)\alpha_k + \gamma\overline{\alpha} + r_t, (1-\gamma)\beta_k + \gamma\overline{\beta} + 1 - r_t) & \text{if } x_t = k. \end{cases}$$

NONSTATIONARY SYSTEM

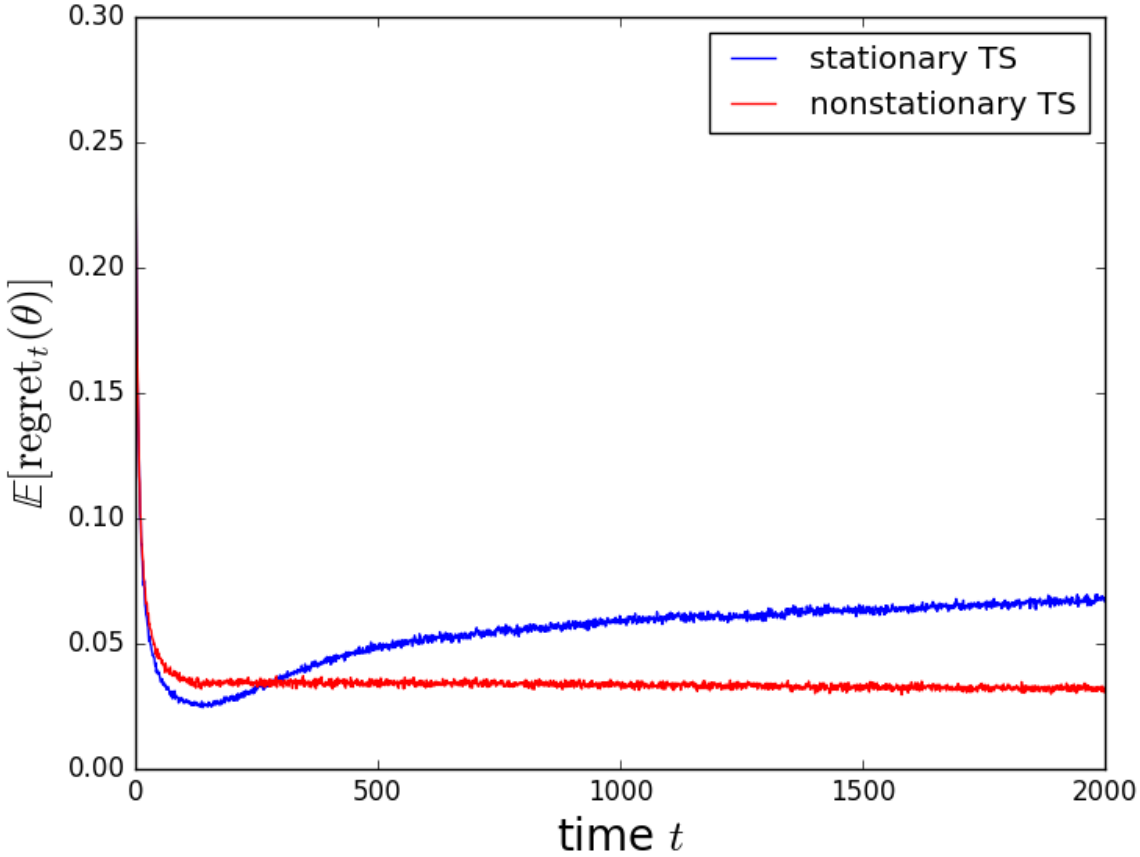


Figure: Comparison of TS vs nonstationary TS with a nonstationary Bernoulli bandit problem

LIMITATIONS

- Time-sensitive learning problems
- Nonstationary learning problems
- Problems requiring careful assessment of information gain
  - Suppose there are $k+1$ actions $\{0, 1, ..., k\}$, and θ is an unknown parameter drawn uniformly at random from $\Theta = \{1, .., k\}$. Rewards are deterministic conditioned on θ, and when played action $i \in \{1, ..., k\}$ always yields reward 1 if $\theta = i$ and 0 otherwise. Action 0 is a special "revealing" action that yields reward $1/2\theta$ when played.

Reinforcement learning in Markov Decision Problems

- Action: $x_t \in A$
- State of the system at time t: $s_t \in S$
- A response y_t is observed which is dependent on x_t and s_t
- An instantaneous reward is received at time t: $r_t = r(y_t, s_t)$
- The next state s_{t+1} is dependent on x_t and s_t

Reinforcement learning in Markov Decision Problems

- Objective: maximize the cumulative rewards in each distinct episode with H timesteps: $\sum_{k=1}^{K} \sum_{h=0}^{H-1} r(s_{kh}, a_{kh})$

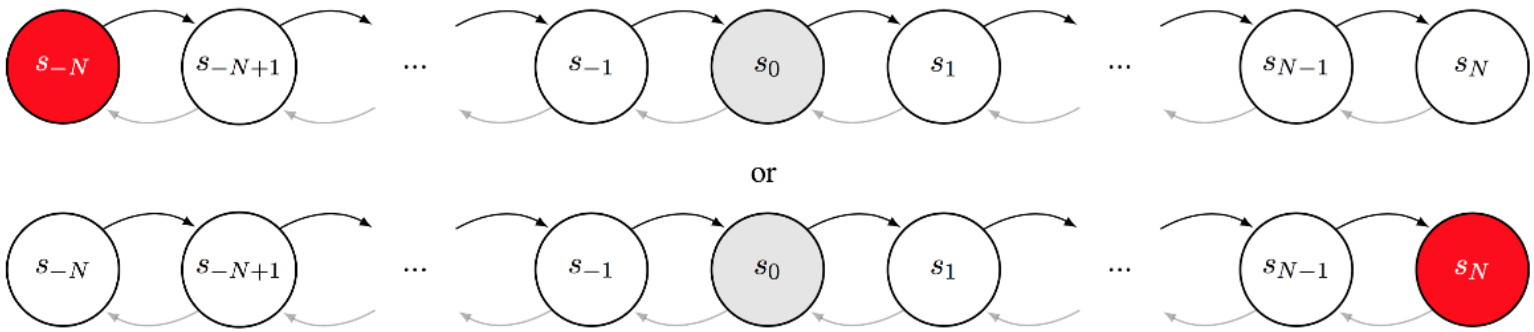


Figure: MDPs where TP every timestep leads to ineffcient exploration

Reinforcement learning in Markov Decision Problems

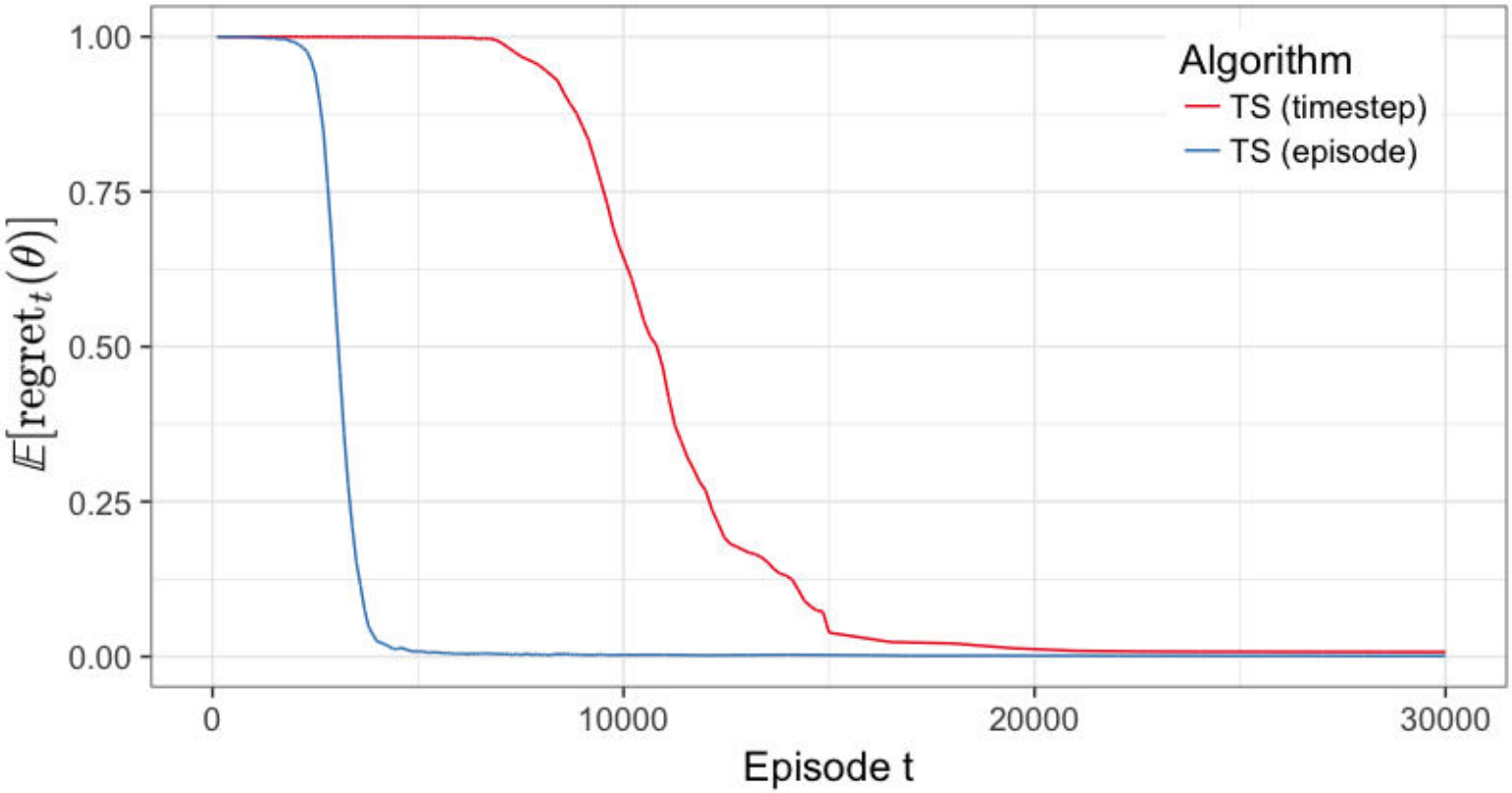


Figure: Comparing TS by episode or by timestep in a simple 24-state MDP