Batched K-arm pulls for Stochastic Multi-Armed Bandits

Shashwat Shukla

Karan Taneja

Sucheta Ravikanti

Problem Setup

Stochastic multi-armed bandit setting with **N** arms

Arm **i** yields mean reward $\boldsymbol{\mu_i}$

K arms pulled at time **t**: $A_K(t)$, set of cardinality K

Define $R(A) = \sum_{i \in A} \mu_i \qquad R_K^* = \max_{|A|=K} R(A)$

Simple regret after **M** time-steps: $S_K(M) = R_K^* - R(A_K(M))$

Pseudo Cumulative regret for **M** time-steps: $C_K(M) = M \cdot R_K^* - \sum_{t=1}^{M} R(A_K(t))$

Goal: Minimize expected regret

Motivation for simple regret

Looking for oil drilling sites:

- N potential sites, can send out K teams at a time
- Canada, can only drill in summers, induces batch pulls
- Correlated rewards because oil distributes via diffusive processes
- Need to make decision in limited rounds (policy timeline)
- Simple regret because long term payoff matters

http://www.canadianoilstocks.ca/investing-in-drilling-companies/precision-drilling-map/

Motivation for cumulative regret

Advertisement targeting:

- N websites, can advertise on K websites at a time (marketing budget)
- Monthly ad contract induces batch pulls
- Correlated rewards: people visit related sites
- Correlated noise: common factors like recession
- Cumulative regret because selling ads is an ongoing process

https://monishap.me/online-advertisement/

Aggressive Elimination for Top-k Arm

- A recursive algorithm, decreasing the number of coins at each step.
- S_r: the number of coins at recursive call (or round) r.
- k: the number of coins to be selected.
- δ: the confidence parameter, algorithm works with probability $1-\delta$.
- Δ: a lower bound the on reward gap between the coin k and $k+1$.

- $\log^* n = 1 + \log^* \log(n)$ if $n > 1$, else 0
- $\log^*(100) = 3, \log^*(10^7) = 4, \log^*\left(10^{20}\right) = 4$

- $ilog^{(r)}(a) = \max\{\log(ilog^{(r-1)}(a)), 1\}, ilog^{(0)}(a) = a$
- $ilog^{(3)}(1000) = 3$

Aggressive Elimination for Top-k Arm

Algorithm 1 AGRESSIVE-ELIMINATION$(S_r, k, r, \delta, \Delta)$

Input: set $S_r \subseteq [n]$ of coins, number of desired top items k, number of rounds r, confidence parameter $\delta \in (0,1)$, and lower bound on gap parameter $\Delta \leq \Delta_k$
Let $m = m_r = |S_r|$ and $t_r := \frac{2}{\Delta^2} \cdot \left(\text{ilog}^{(r)}(m) + \log(8k/\delta)\right)$.
Toss each coin $i \in S_r$ for t_r times.
For each $i \in S_r$, define $\widehat{p}_i$ as the fraction of times coin i turns up heads.
Sort the coins in S_r in a decreasing order of $\widehat{p}$-values.
if $r = 1$ **then**
    Return: the set of k most biased coins (according to $\widehat{p}$-values).
else
    Let $m_{r-1} := k + \frac{m}{\text{ilog}^{(r-1)}(m)}$ and S_{r-1} be the set of m_{r-1} most biased coins according to $\widehat{p}$.
end if
if $m_{r-1} \leq 2k$ **then**
    Return: AGRESSIVE-ELIMINATION$(S_{r-1}, k, 1, \delta/2, \Delta)$.
else
    Return: AGRESSIVE-ELIMINATION$(S_{r-1}, k, r-1, \delta/2, \Delta)$.
end if

Agarwal, Arpit, et al. "Learning with limited rounds of adaptivity: Coin tossing, multi-armed bandits, and ranking from pairwise comparisons." Conference on Learning Theory. 2017.

Perturbed Aggressive Elimination for Top-k Arm

Algorithm 1 AGRESSIVE-ELIMINATION$(S_r, k, r, \delta, \Delta)$

Input: set $S_r \subseteq [n]$ of coins, number of desired top items k, number of rounds r, confidence parameter $\delta \in (0,1)$, and lower bound on gap parameter $\Delta \leq \Delta_k$

Let $m = m_r = |S_r|$ and $t_r := \frac{2}{\Delta^2} \cdot \left(\text{ilog}^{(r)}(m) + \log(8k/\delta)\right)$.

Toss each coin $i \in S_r$ for t_r times.

Think of it as a sequence of t_r *sub-rounds* where all n coins are sampled in each round.

For each $i \in S_r$, define $\widehat{p}_i$ as the fraction of times coin i turns up heads.

Sort the coins in S_r in a decreasing order of $\widehat{p}$-values.

if $r = 1$ **then**

  Return: the set of k most biased coins (according to $\widehat{p}$-values).

else

  Let $m_{r-1} := k + \frac{m}{\text{ilog}^{(r-1)}(m)}$ and S_{r-1} be the set of m_{r-1} most biased coins according to $\widehat{p}$.

end if

if $m_{r-1} \leq 2k$ **then**

  Return: AGRESSIVE-ELIMINATION$(S_{r-1}, k, 1, \delta/2, \Delta)$.

else

  Return: AGRESSIVE-ELIMINATION$(S_{r-1}, k, r-1, \delta/2, \Delta)$.

end if

Perturbed Aggressive Elimination for Top-k Arm

Algorithm 1 AGRESSIVE-ELIMINATION$(S_r, k, r, \delta, \Delta)$

1: **Input:** set $S_r \subseteq [n]$ of coins, number of desired top items k, number of rounds r, confidence parameter $\delta \in (0,1)$, and lower bound on gap parameter $\Delta \leq \Delta_k$
2: Let $m = m_r = |S_r|$ and $t_r := \frac{2}{\Delta^2} \cdot \left(\text{ilog}^{(r)}(m) + \log(8k/\delta)\right)$.
3: Toss each coin $i \in S_r$ for t_r times.
4: For each $i \in S_r$, define $\widehat{p}_i$ as the fraction of times coin i turns up heads.
5: Sort the coins in S_r in a decreasing order of $\widehat{p}$-values.
6: **if** $r = 1$ **then**
7: **Return:** the set of k most biased coins (according to $\widehat{p}$-values).
8: **else**
9: Let $m_{r-1} := k + \frac{m}{\text{ilog}^{(r-1)}(m)}$ and S_{r-1} be the set of m_{r-1} most biased coins according to $\widehat{p}$.
10: **end if**
11: **if** $m_{r-1} \leq 2k$ **then**
12: **Return:** AGRESSIVE-ELIMINATION$(S_{r-1}, k, 1, \delta/2, \Delta)$.
13: **else**
14: **Return:** AGRESSIVE-ELIMINATION$(S_{r-1}, k, r-1, \delta/2, \Delta)$.
15: **end if**

In our perturbed algorithm, in a sequence of t_r *sub-rounds*, we sample the top-k coins only.

Perturbed Aggressive Elimination for Top-k Arm

- If $\mu_i(t)$ is the estimate of mean, t is the time, $T_i(t)$ is the number of times i^{th} coin has been sampled.
- The perturbed estimate of coin i is given by

$$\theta_i(t) = \mu_i(t) + \sqrt{\frac{(2+\epsilon)\log(t)}{1 \vee T_i(t)}}\, Z_{it}$$

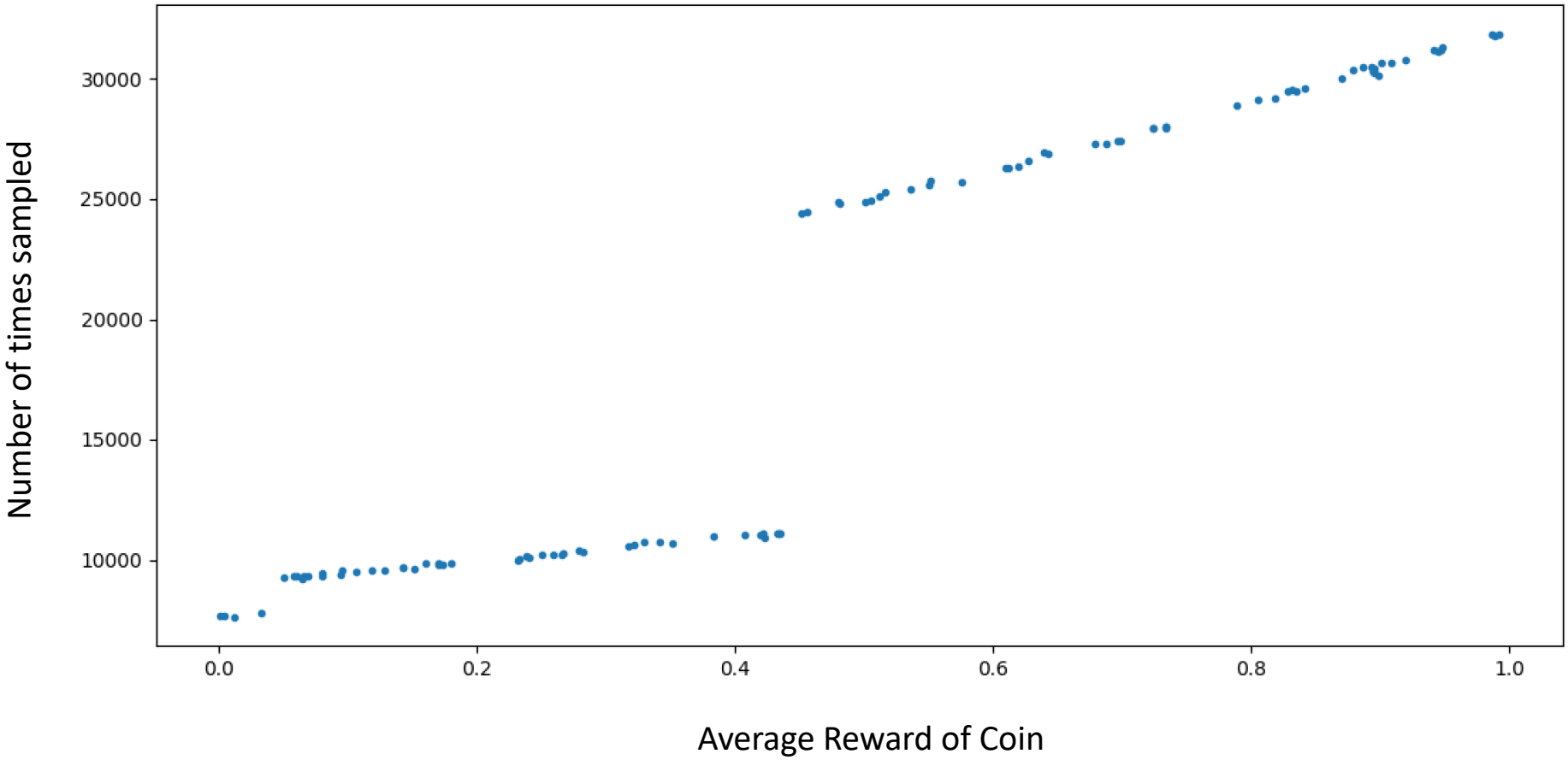
where Z_{it}'s are randomly sampled from $Z \in [-1, 1]$.

Kim, Baekjin, and Ambuj Tewari. "On the Optimality of Perturbations in Stochastic and Adversarial Multi-armed Bandit Problems." arXiv preprint arXiv:1902.00610 (2019).

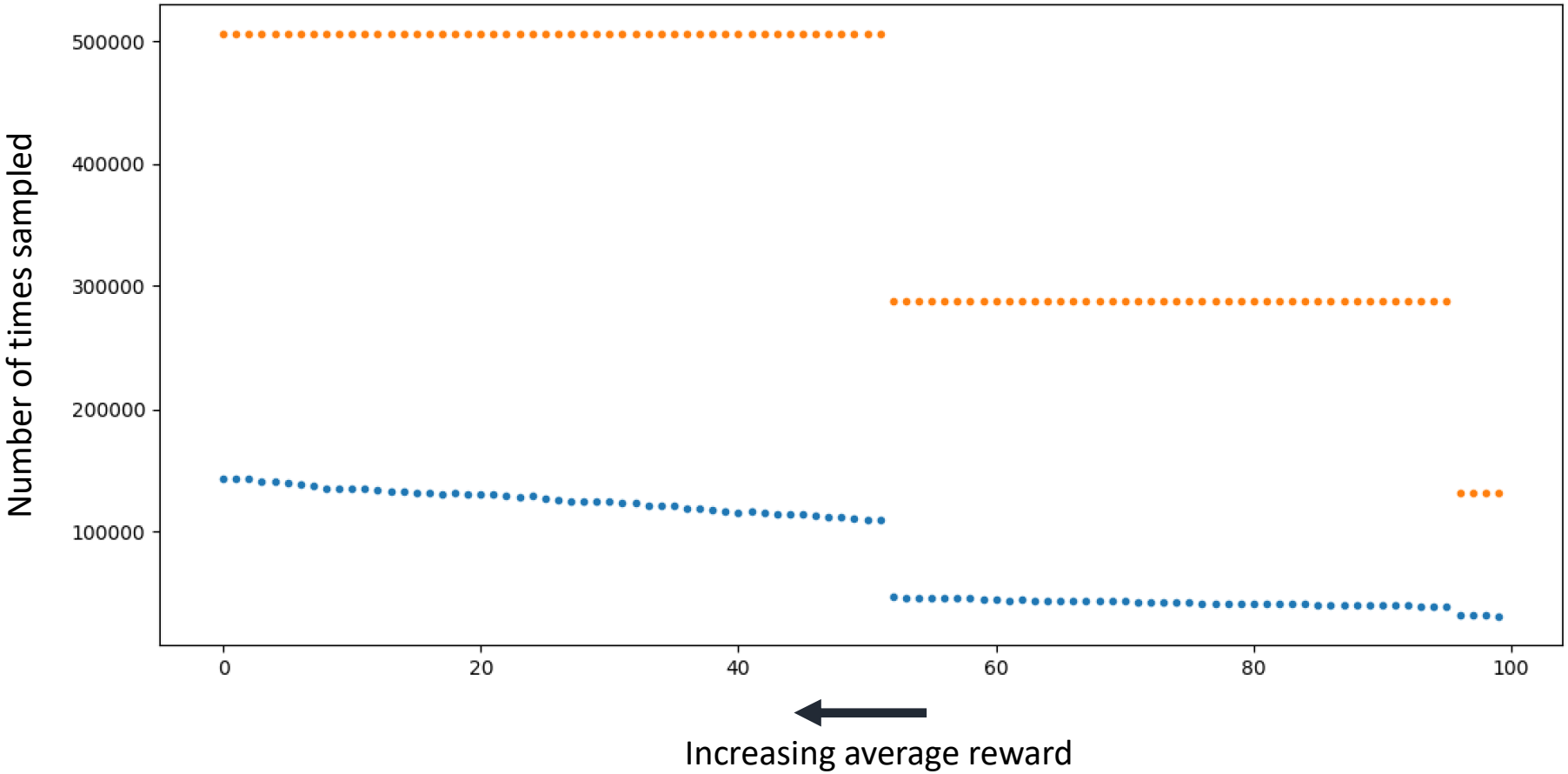
Perturbed Aggressive Elimination for Top-k Arm

- Number of rounds remain the same: $\log^*(n)$.
- Sample complexity in each round was $O\left(\frac{n\ log\frac{k}{\delta}}{\Delta_k^2}\right)$.
- With our algorithm, sample complexity is now $O\left(\frac{k\ log\frac{k}{\delta}}{\Delta_k^2}\right)$.

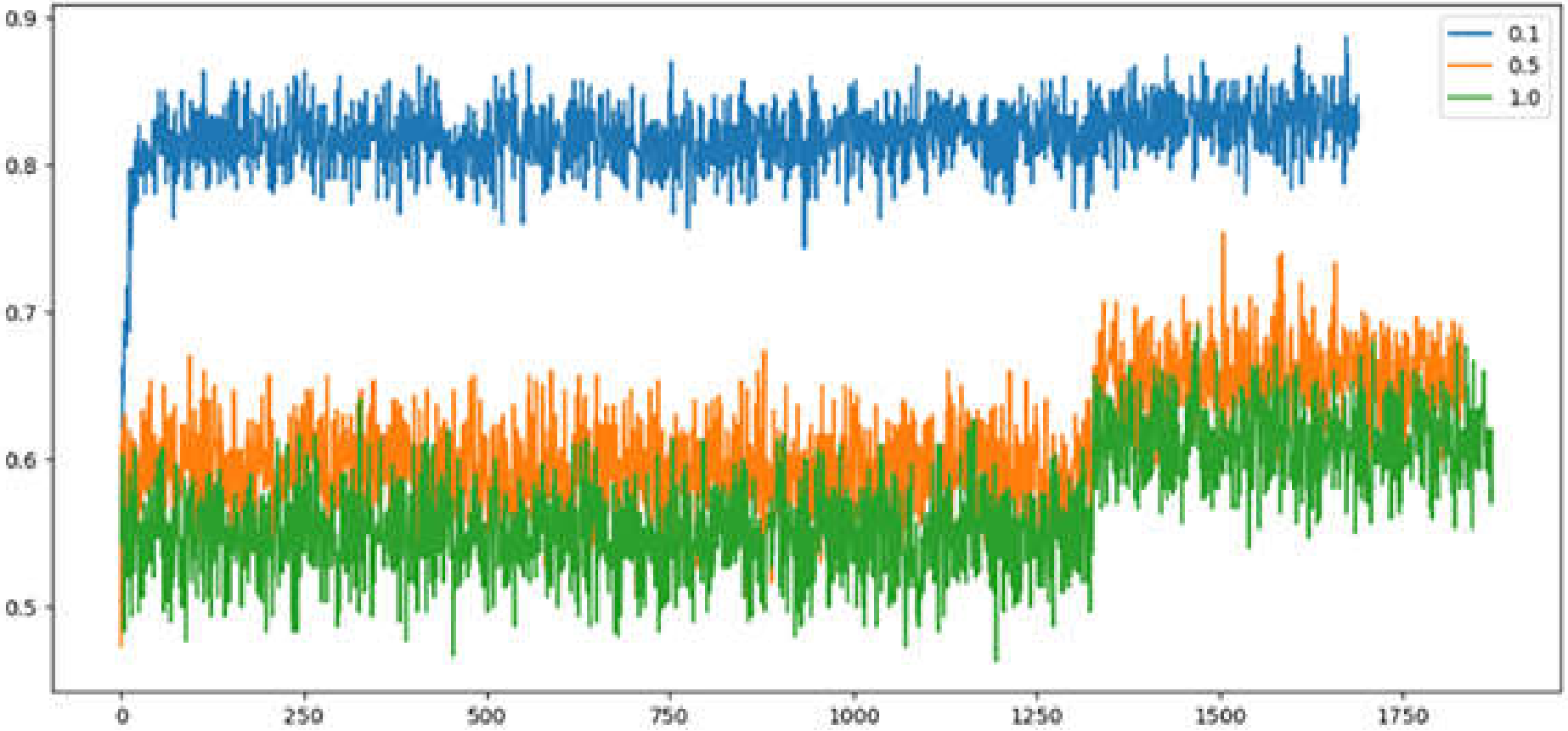
Coins with high rewards are sampled more



We sample much less



Perturbation Amplitude controls Exploration-Exploitation Tradeoff



Gaussian Processes

Gaussian Processes (GP) define distributions over functions.

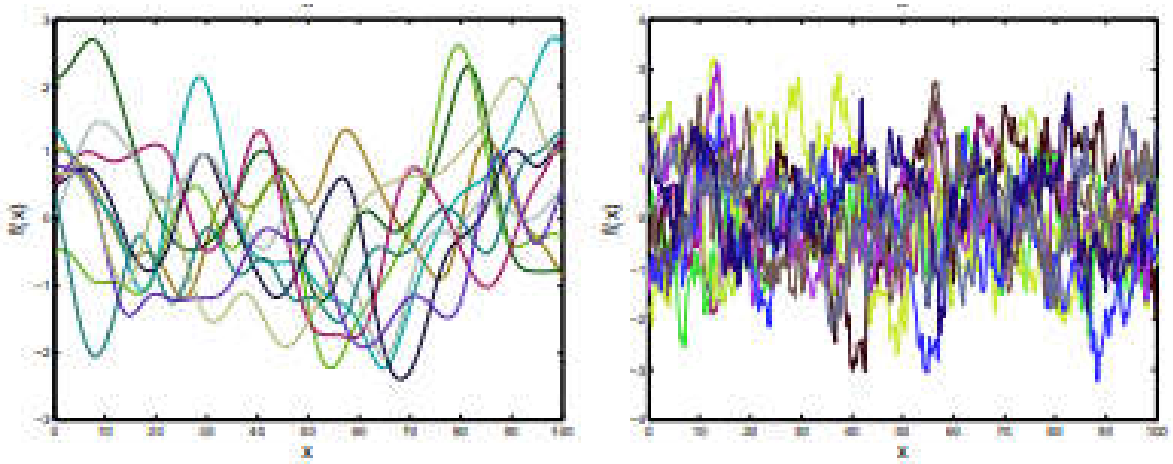
Mean of function distribution: $\mu(x)$

Covariance kernel: $k(x, x')$

$$p(f(x), f(x')) = \mathsf{N}(\mu, \Sigma)$$

$$\mu = \begin{bmatrix} \mu(x) \\ \mu(x') \end{bmatrix} \quad \Sigma = \begin{bmatrix} K(x,x) & K(x,x') \\ K(x',x) & K(x',x') \end{bmatrix}$$

Smoothness assumption encoded in kernel: $K(x_i, x_j) = v_0 \exp\left\{ -\left(\frac{|x_i - x_j|}{r} \right)^{\alpha} \right\}$



Samples for a smooth and rough kernel

Gaussian Processes Regression

Data samples $\mathcal{D}_t = \{\mathbf{X}_t, \mathbf{y}_t\}$, new inputs $\mathbf{X}_\star$

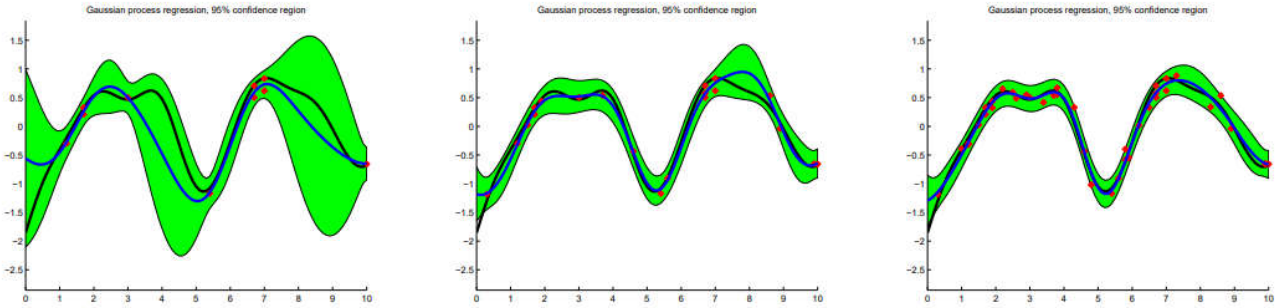
Joint distribution:

$$\begin{bmatrix} \mathbf{y}_t \\ \mathbf{f}_\star \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{X}_t, \mathbf{X}_t) + \sigma_\epsilon^2 \mathbf{I} & K(\mathbf{X}_t, \mathbf{X}_\star) \\ K(\mathbf{X}_\star, \mathbf{X}_t) & K(\mathbf{X}_\star, \mathbf{X}_\star) \end{bmatrix}\right)$$

$$K(\mathbf{X}_\star, \mathbf{X}_\star) = \begin{bmatrix} k(\mathbf{x}_1^\star, \mathbf{x}_1^\star) & k(\mathbf{x}_1^\star, \mathbf{x}_2^\star) & \dots & k(\mathbf{x}_1^\star, \mathbf{x}_n^\star) \\ k(\mathbf{x}_2^\star, \mathbf{x}_1^\star) & k(\mathbf{x}_2^\star, \mathbf{x}_2^\star) & \dots & k(\mathbf{x}_2^\star, \mathbf{x}_n^\star) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_n^\star, \mathbf{x}_1^\star) & k(\mathbf{x}_n^\star, \mathbf{x}_2^\star) & \dots & k(\mathbf{x}_n^\star, \mathbf{x}_n^\star) \end{bmatrix}$$

Posterior = Conditional distribution: $p(\mathbf{f}_\star | \mathbf{X}_t, \mathbf{y}_t, \mathbf{X}_\star)$

$$m_t(\mathbf{x}) = K(\mathbf{x}, \mathbf{X}_t)\left[K(\mathbf{X}_t, \mathbf{X}_t) + \sigma_\epsilon^2 \mathbf{I}\right]^{-1} \mathbf{y}_t \qquad k_t(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - K(\mathbf{x}, \mathbf{X}_t)\left[K(\mathbf{X}_t, \mathbf{X}_t) + \sigma_\epsilon^2 \mathbf{I}\right]^{-1} K(\mathbf{X}_t, \mathbf{x}')$$



(a) (b) (c)

http://biorxiv.org/content/biorxiv/early/2017/10/10/095190.full.pdf

GP-UCB for cumulative regret

Algorithm 1 The GP-UCB algorithm.

Input: Input space D; GP Prior $\mu_0 = 0,\ \sigma_0,\ k$

for $t = 1, 2, \ldots$ **do**

Choose $\boldsymbol{x}_t = \underset{\boldsymbol{x} \in D}{\operatorname{argmax}}\, \mu_{t-1}(\boldsymbol{x}) + \sqrt{\beta_t}\sigma_{t-1}(\boldsymbol{x})$

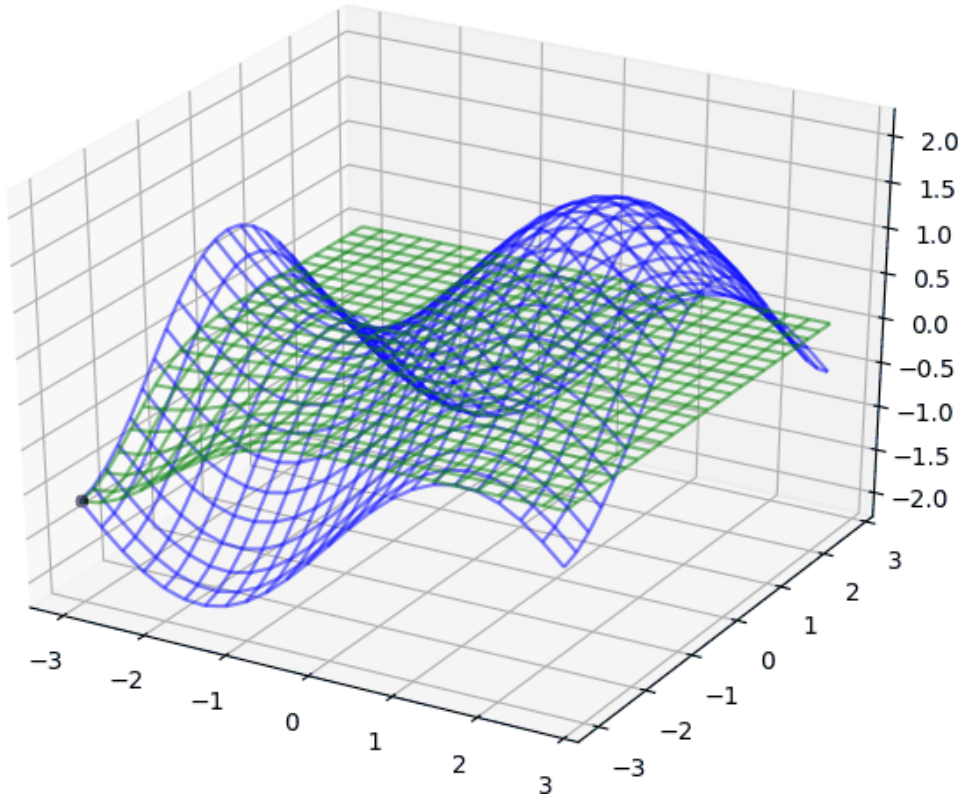
Sample $y_t = f(\boldsymbol{x}_t) + \epsilon_t$

Perform Bayesian update to obtain μ_t and σ_t

end for

$$\mu_T(\mathbf{x}) = \mathbf{k}_T(\mathbf{x})^\top (\mathbf{K}_T + \sigma^2 \mathbf{I}) \mathbf{y}_T$$

$$k_T(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{k}_T(\mathbf{x})^\top (\mathbf{K}_T + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_T(\mathbf{x}')$$



https://arxiv.org/pdf/0912.3995.pdf

Naive Batch GP-UCB (BGPUCB)

Pick K arms from UCB hull with maximum sum.

Over-optimistic: sums up confidence bounds of correlated variables

No combinatorial blow-up.

Batch Thompson Sampling (BGPTS)

Draw sample function from current posterior.

Pick K arms from this sample function with maximum sum.

Probabilistic matching property still holds.

No combinatorial blow-up.

Improved Batch GP-UCB (IBGPUCB)

Enumerate all C(N,K) sets.

Each set has arms X_1, X_2, X_3,...,X_K.

Mean of this super-arm: $\sum_{i=1}^{K} \mu_i$

Variance of super-arm: $\sum_{i=1}^{K} \sum_{j=1}^{K} \rho_{ij} \sigma_j \sigma_j$

Confidence bounds computed by summing up K-slices of the covariance matrix.

Pick super-arm with highest improved UCB.

Combinatorial blow-up.

Look-ahead Batch GP-UCB (LBGPUCB)

Smooth prior: (correlated means and) correlated confidence bounds.

Might lead to less exploration.

Sampling one arm enough to update nearby values as well.

Hence, IGPUCB may be picking too many arms close to each other.

$$\mu_T(\mathbf{x}) = \mathbf{k}_T(\mathbf{x})^\top (\mathbf{K}_T + \sigma^2 \mathbf{I}) \mathbf{y}_T$$
$$k_T(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{k}_T(\mathbf{x})^\top (\mathbf{K}_T + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_T(\mathbf{x}')$$

Idea: Covariance updates only depend on picked *locations* (not values)

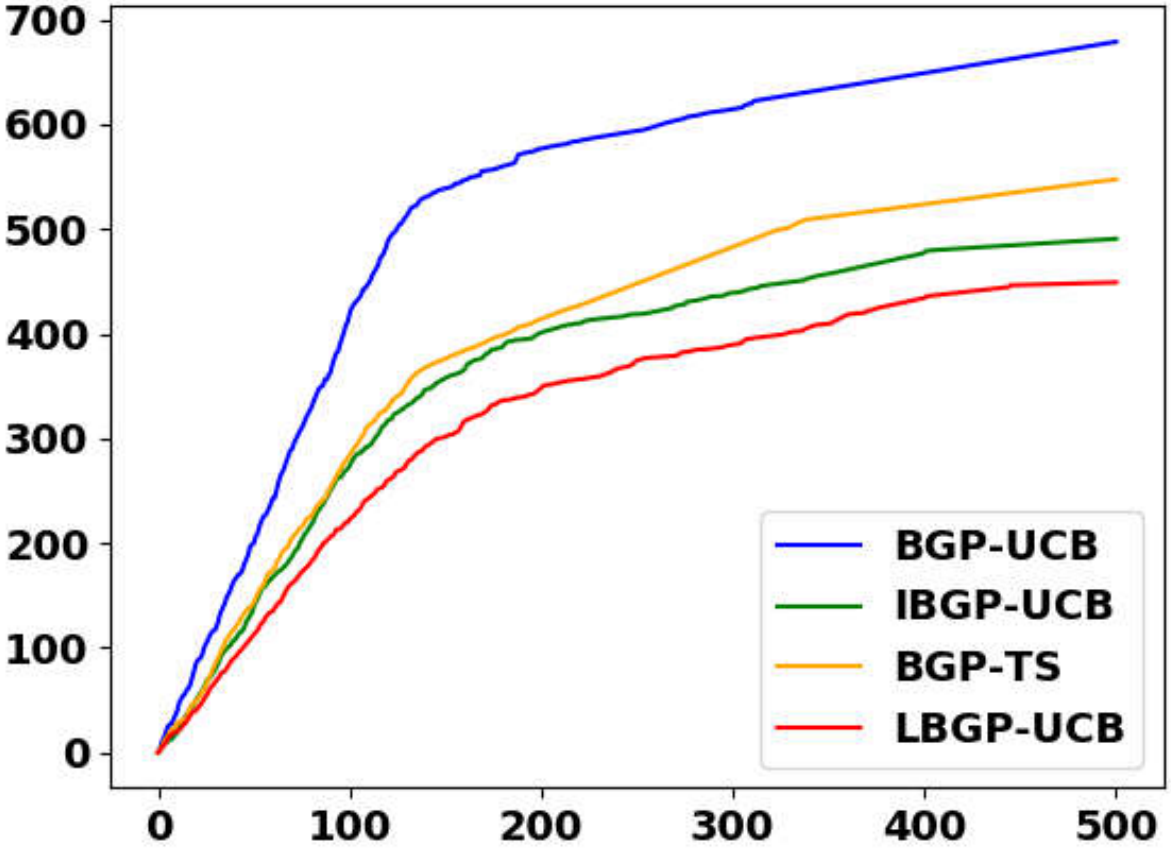
In a batch, pick the K arms on at a time using GP-UCB.

Update mean after each batch (virtual sampling).

Update covariance at each step.

No combinatorial blowup.

Simulations for Cumulative Regret for K=3



Noise variance = 1.0; Averaged over 10 functions; 5 runs per function

Other considerations

- Correlated noise can be incorporated (only covariance matrix changes).
- Learn the hyperparameters of the GP kernel as well.
- Faster approximate GP inference.
- Improved Thompson sampling by taking only positive perturbations.
- Compare IBGP-UCB and LBGP-UCB for simple regret.
- Alloting some arms to pure exploration may improve performance.

Thank You

References

1. Top k arms with batch pulls; fixed confidence.
2. Best arm; fixed confidence; unstructured MAB; uses a UCB variant, lil'UCB.
3. Pure Exploration in Finitely–Armed and Continuous–Armed Bandits.
4. Optimal perturbation algorithms for Stochastic Multi-Armed Bandits.
5. Thompson sampling for GP-MAB converges exponentially in probability to global optima.
6. Simple regret bounds for no-noise fixed horizon GP-MAB optimal strategy.
7. Top k arms with batch pulls; fixed budget; unstructured MAB.
8. Top k arms single pulls; fixed confidence; unstructured MAB.
9. Cumulative regret bounds for GP-UCB.
10. Tighter cumulative regret bounds for GP-UCB assuming kernel prior.
11. Top k-arms; fixed budget and fixed confidence.
12. Cumulative regret bounds for Combinatorial MAB.
13. Cumulative regret bounds for GP Combinatorial MAB.
14. Cumulative regret bounds for batch arm pulls; unstructured MAB
15. GP-UCB fit to human trials.

References

1. Agarwal, Arpit, et al. "Learning with limited rounds of adaptivity: Coin tossing, multi-armed bandits, and ranking from pairwise comparisons." Conference on Learning Theory. 2017.
2. Jamieson, Kevin, et al. "lil'ucb: An optimal exploration algorithm for multi-armed bandits." Conference on Learning Theory. 2014.
3. Bubeck, Sébastien, Rémi Munos, and Gilles Stoltz. "Pure exploration in finitely-armed and continuous-armed bandits." Theoretical Computer Science 412.19 (2011): 1832-1852.
4. Kim, Baekjin, and Ambuj Tewari. "On the Optimality of Perturbations in Stochastic and Adversarial Multi-armed Bandit Problems." arXiv preprint arXiv:1902.00610 (2019).
5. Basu, Kinjal, and Souvik Ghosh. "Analysis of Thompson sampling for Gaussian process optimization in the bandit setting." arXiv preprint arXiv:1705.06808 (2017).
6. Grünewälder, Steffen, et al. "Regret bounds for Gaussian process bandit problems." Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. 2010.
7. Jun, Kwang-Sung, et al. "Top Arm Identification in Multi-Armed Bandits with Batch Arm Pulls." AISTATS. 2016.
8. Kalyanakrishnan, Shivaram, et al. "PAC Subset Selection in Stochastic Multi-armed Bandits." ICML. Vol. 12. 2012.
9. Srinivas, Niranjan, et al. "Gaussian process optimization in the bandit setting: No regret and experimental design." arXiv preprint arXiv:0912.3995 (2009).
10. Chowdhury, Sayak Ray, and Aditya Gopalan. "On kernelized multi-armed bandits." Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017.
11. Kaufmann, Emilie, Olivier Cappé, and Aurélien Garivier. "On the complexity of best-arm identification in multi-armed bandit models." The Journal of Machine Learning Research 17.1 (2016): 1-42.
12. Chen, Wei, Yajun Wang, and Yang Yuan. "Combinatorial multi-armed bandit: General framework and applications." International Conference on Machine Learning. 2013.
13. Accabi, G., et al. "When Gaussian Processes Meet Combinatorial Bandits: GCB." (2018).
14. Perchet, Vianney, et al. "Batched bandit problems." The Annals of Statistics 44.2 (2016): 660-681.
15. Wu, Charley M., et al. "Mapping the unknown: The spatially correlated multi-armed bandit." bioRxiv (2017): 106286.

More references

https://see.stanford.edu/materials/aimlcs229/cs229-gp.pdf

https://www.csie.ntu.edu.tw/~cjlin/mlgroup/tutorials/gpr.pdf

https://www.biorxiv.org/content/biorxiv/early/2017/10/10/095190.full.pdf

https://www.cs.toronto.edu/~hinton/csc2515/notes/gp_slides_fall08.pdf

http://cs229.stanford.edu/section/gaussians.pdf

https://mlss2011.comp.nus.edu.sg/uploads/Site/lect1gp.pdf

https://banditalgs.com/2016/09/18/the-upper-confidence-bound-algorithm/

http://learning.eng.cam.ac.uk/pub/Public/Turner/Presentations/gp-approximation.pdf

https://arxiv.org/pdf/1301.2609.pdf

https://link.springer.com/content/pdf/10.1007%2F978-3-540-75225-7.pdf

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/thompson.pdf

https://arxiv.org/pdf/1507.01160.pdf

http://www.princeton.edu/~naomi/theses/ReverdyPhDThesis.pdf

https://ieeexplore.ieee.org/ielx7/6725831/6736491/06736565.pdf

http://www.cs.cmu.edu/~spandey/publications/dependent-bandit.pdf

https://arxiv.org/pdf/1902.02953.pdf