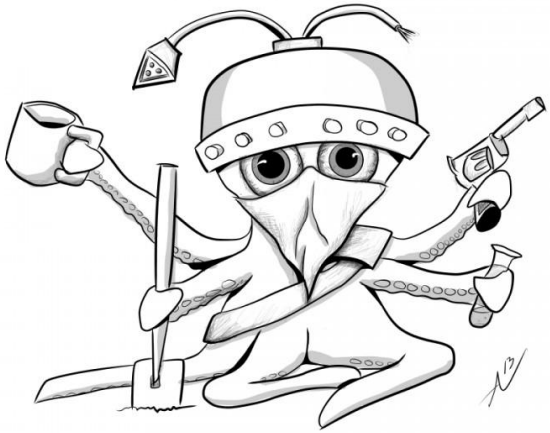
Multi-Armed Bandit Solutions as Neural Net Learning Algorithms

By:
River Ludington,
Tristan Caputo,
Paul Lenharth



Neural Network Learning

- Neural Nets can be monitored or unmonitored
- Unmonitored Nets Require learning algorithms
- The Bandit Problem is a good theoretical framework

Multi-armed Bandit Problem

- Multi-armed bandit is a model used for the exploitation/exploration problem
- Formulated with slot machines of different, unknown payouts
- Goal is to maximize money
- Requires quickly finding the “best” machine and playing only it
- Probability makes knowledge of best machine uncertain, must keep testing others
- Testing nets less money

Applications: Clinical Trials, Recommendations Alg.

- Identify the best treatment (best slot machine) and ensure that as many patients as possible can receive it in the clinical trial. (less regret)

- Identify the best recommendations (best slot machine) and ensure that the user has a good chance of clicking on it. (less regret)

Definitions

Overlapping Distributions: differences in averages are less than or close to standard deviation

Spread-out Distributions: differences in averages much bigger than standard deviation

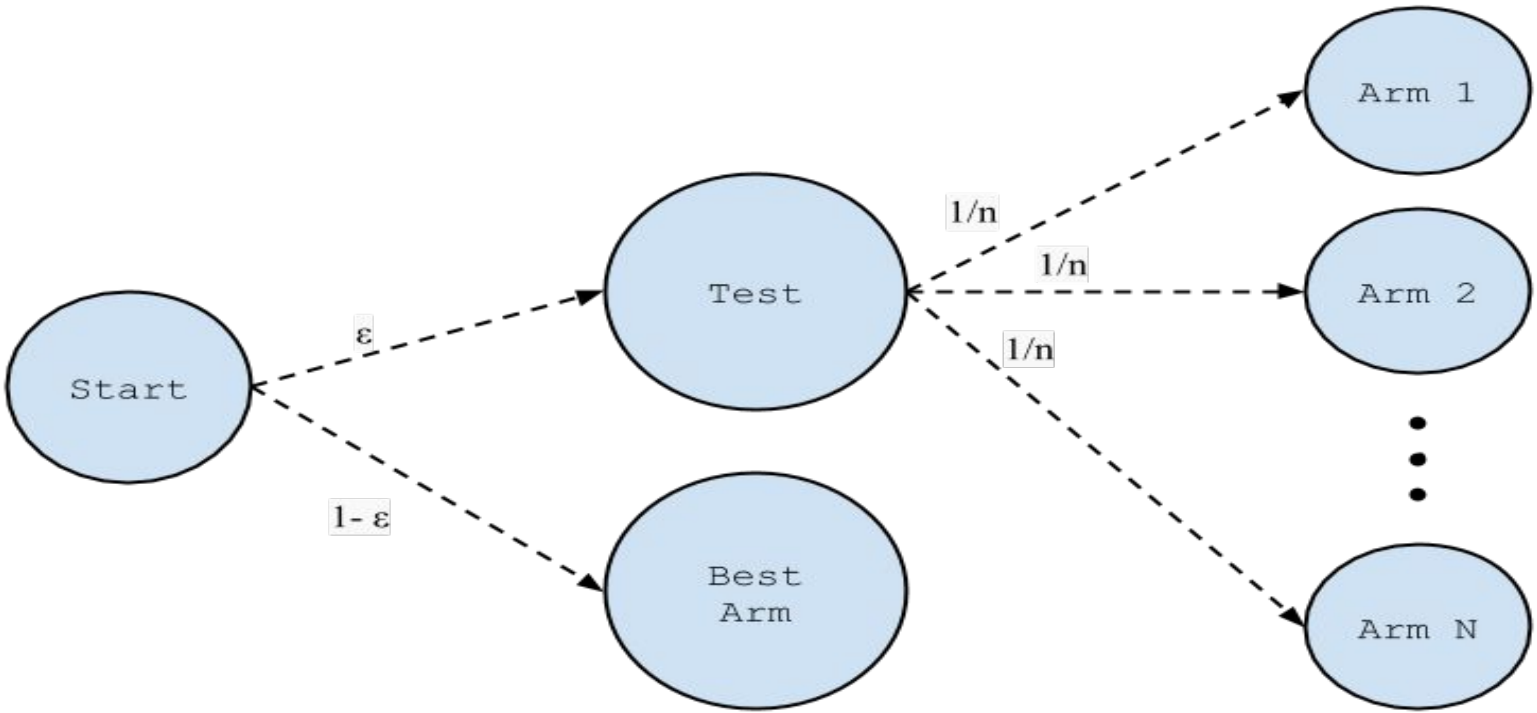
Normalized regret: cumulative money lost compared to playing the best machine, relative to the best machine. $R=1-M_{average}/mu_{best}$ where $M_{average}$ is average reward per turn and mu_{best} is the actual average of the best machine

Optimal Convergence: the ability of an algorithm to have no regret asymptotically

Algorithms Used

1. Epsilon-Greedy (many variations)
2. Softmax (Also known as Boltzmann)
3. Thompson Sampling
4. Vary-Greedy

Epsilon Greedy



Vary-Greedy

Variant of Epsilon greedy

Only difference is Vary-Greedy will ‘drop’ arms whose payouts are 1 or more standard deviation less than mean payout of the experimental best machine while doing exploration.

Softmax (Boltzmann)

Probability function for arm n, where mu is the sample mean

$$P(n) = \frac{\mu_n^2 e^{\mu_n^2}}{\sum_{n=1}^{N} \mu_n^2 e^{\mu_n^2}}$$

Probability of selecting arm n

Probability Normalization

Thompson Sampling

Algorithm:

1. Assume every arm is a Beta(α,β) distribution
2. Initialize α,β = 1 (uniform distribution)
3. Every turn, the algorithm chooses the arm with the highest mean
    a. The mean is $\alpha/(\alpha+\beta)$
4. For every successful arm pull, it's distribution is updated to Beta(α+1,β) or failure is updated to Beta(α,β+1)

Measure of Success

1. Make the most money possible
2. Reduce “Regret”

The normalized regret each round is the difference between the mean reward of the optimal arm, and the reward of the arm chosen [1] divided by the mean reward of the optimal arm.

$$R = \frac{\mu^* - \mu_{j(t)}}{\mu^*}$$

Where R is the regret, μ^* is the expected reward from the best arm, $\mu_{j(t)}$ is the reward from the arm j chosen at round t

Citation: Our Final Paper

Room for Growth

According to our Research:

> “simple heuristics such as ε-greedy and Boltzmann exploration outperform theoretically sound algorithms on most settings by a significant margin.”
>
> - Volodymyr Kuleshov, Doina Precup

Note: The higher end algorithms are pursuit, reinforcement comparison, UCB1, UCB1-Tuned

Motivation

1. Explore long-term vs short-term performance of algorithms
2. Inform algorithm choice based on empirical data

Methods

1. We coded the Multi-arm bandit problem to accommodate:
   - Bernoulli or Gaussian Distributions
   - Different numbers of arms
   - Different means and variances

   Coded common learning algorithms:
   - Thompson Sampling
   - Softmax (Boltzmann)
   - Epsilon-Greedy

   Coded a custom algorithm:
   - Vary-Greedy
2. Initialization of averages was performed by playing each machine once
3. At every turn, when the algorithm chose an arm to play, the regret was measured.
4. Parameters for the Multi-Arm Bandit were varied and each algorithm was tested
5. The average was taken over 100 runs (of a 1000 turns each)

Methods (‘Semi’-Optimizing Algorithms)

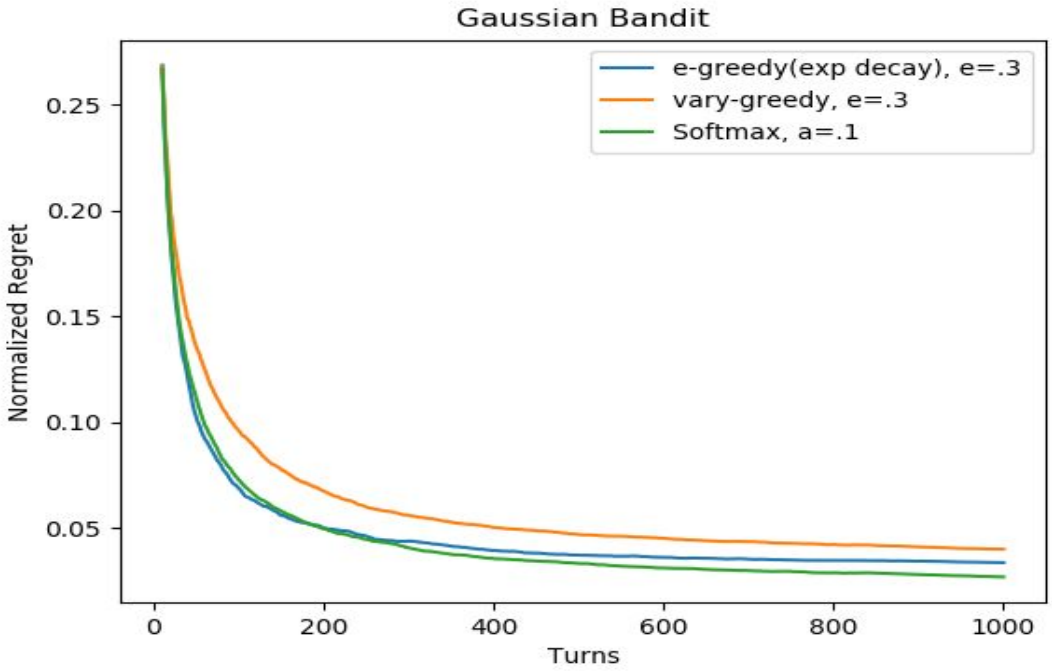
Epsilon-Greedy/VG: We found that setting decay to an exponential function ($ep=e^{-t}$ where t is the number of turns after initialization)

Softmax: We found that setting temperature $a<1$ is optimal for most cases

Thompson: No parameters were used due to time constraints

Sample Regret Plot

Plot showing regret level of EG,VG and Softmax in Overlapping distributions

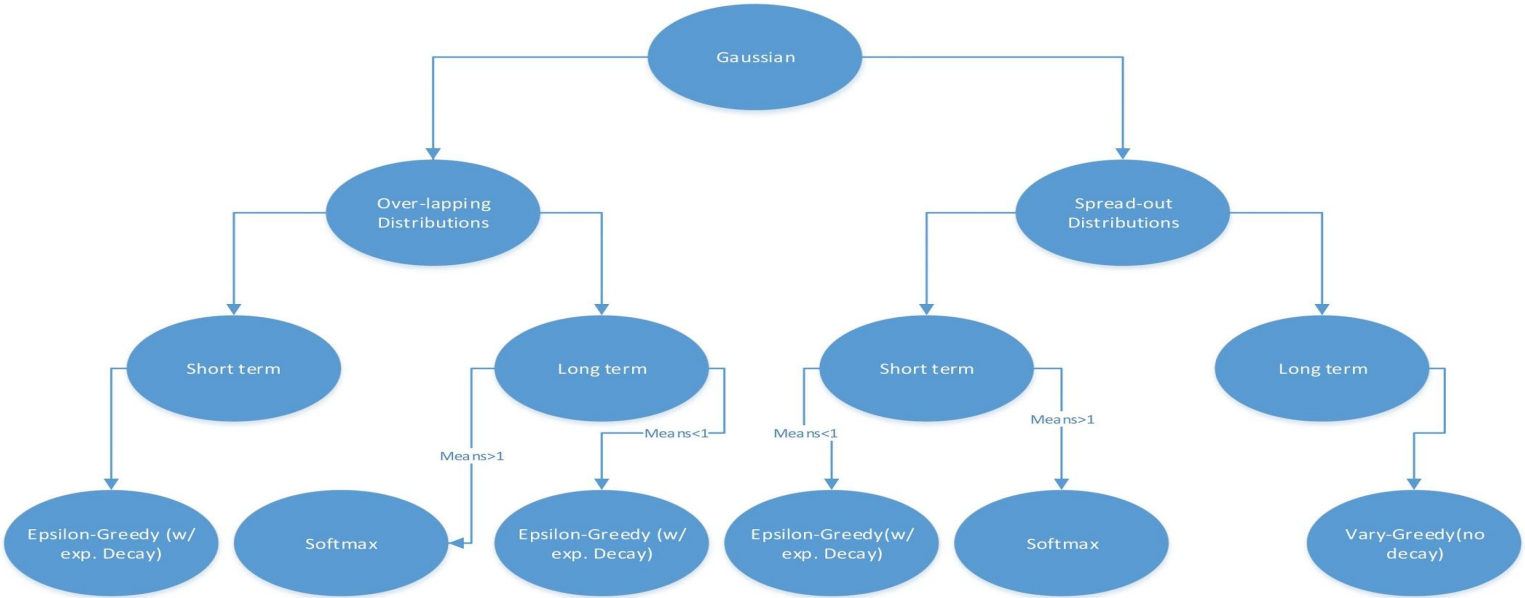


Epsilon-Greedy is optimal in short term. (0-200 runs).

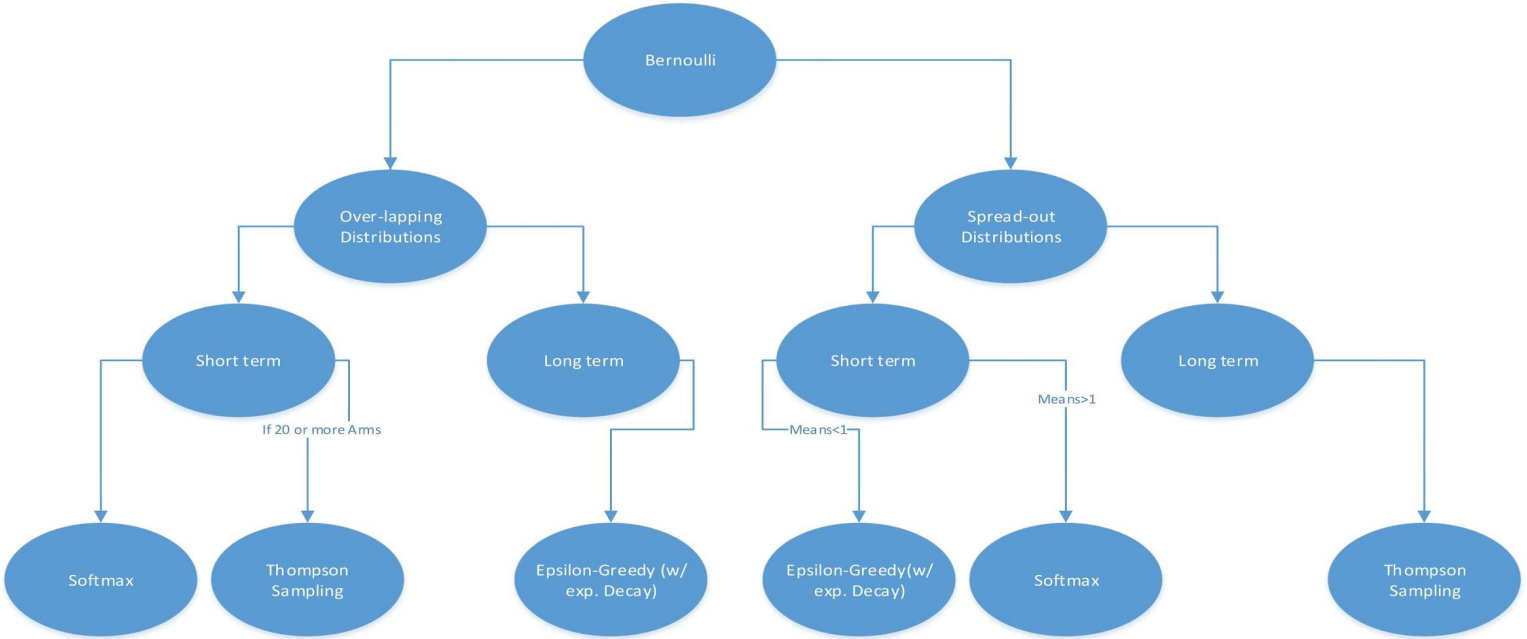
Softmax is optimal in long term (200-1000 runs).

Vary-Greedy is underperforming due to the distributions overlapping, VG cannot identify ‘bad’ arm thus will keep playing ‘bad’ arms.

Winning Algorithms for Gaussian



Winning Algorithms for Bernoulli



References

1. Vermorel, Joannes, and Mehryar Mohri. "Multi-armed bandit algorithms and empirical evaluation." *European conference on machine learning*. Springer, Berlin, Heidelberg, 2005.
2. Kuleshov, Volodymyr, and Doina Precup. "Algorithms for multi-armed bandit problems." *arXiv preprint arXiv:1402.6028* (2014).
3. Raja, Sudeep. “Multi Armed Bandits and Exploration Strategies.” *Multi Armed Bandits and Exploration Strategies – Sudeep Raja – MS/Phd Student at UMass Amherst*, 28 Aug. 2016, sudeepraja.github.io/Bandits/.
4. “Multi-Armed Bandits.” *The Data Incubator MultiArmed Bandits Comments*, blog.thedataincubator.com/2016/07/multi-armed-bandits-2/.