# Decision Making for Robots and Autonomous Systems

Fall 2015

George Konidaris
gdk@cs.duke.edu

# Multi-Arm Bandits

Direct model of:

- Actions (but no "consequences")
- Stochastic outcomes
- Unknown utility

# n-Arm Bandit

An *n*-armed bandit is defined as follows:

- Set of *n* actions: $A = \{A_1, \ldots, A_n\}$
- Reward/utility distribution for each action: $P(r \mid A_i)$
- An agent lifetime, *k*.

At each time t:

- The agent selects action $A_t$
- The agent receives reward $r_t \sim P(r \mid A_t)$ ***(unknown)***
- The agent wishes to maximize reward over its lifetime:

$$\sum_{t=1}^{k} r_t$$

(in expectation)

# Critical Assumption

The generic *n*-arm bandit assumes that:

- $r_t \sim P(r \mid A_t)$

… specifically, reward is dependent on the arm, and independent of:

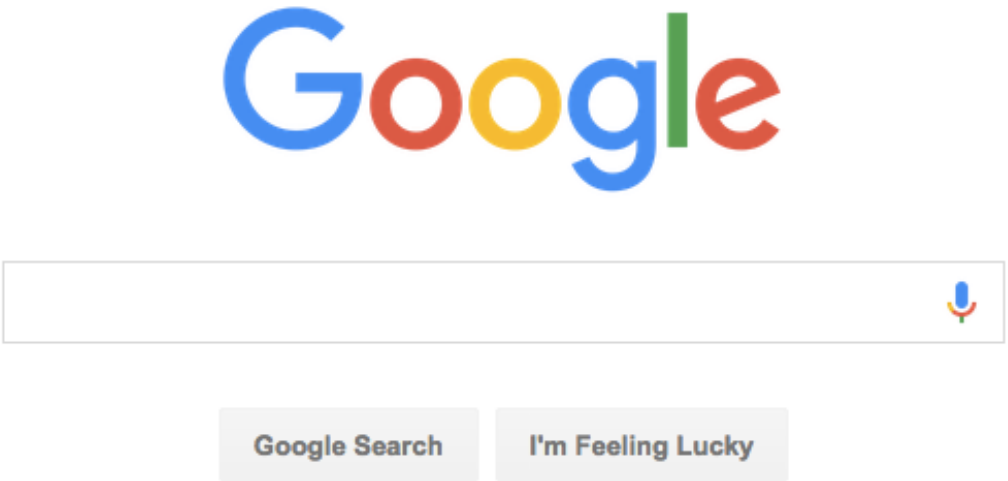
- Time
- Number of executions
- Any other variables

This is a very strong modeling assumption!

**Let's consider our previous scenarios.**

# Application Scenario 1

# Application Scenario II




amazon.com

# Contextual Bandit

A *contextual* bandit is defined as follows:

- Set of $n$ actions: $A = \{A_1, \ldots, A_n\}$
- Context space: $X \subseteq \mathbb{R}^d$
- Reward/utility distribution for each action: $P(r \mid A_i, X_i)$
- An agent lifetime, $k$.

At each time t:

- The agent observes context $X_i \in X$
- The agent selects action $A_t$
- The agent receives reward $r_t \sim P(r \mid A_t, X_i)$ ***(unknown)***
- The agent wishes to maximize reward over its lifetime.

# Policies and Values

A policy is now a mapping from *contexts* to *distributions over actions*:

$$\pi : X \rightarrow P(A_i)$$

and we can define:

$$Q(X, A_i) = \mathbb{E}\left[r_i | X\right]$$

Expected reward is now *a function of X*.

# Value Functions

Estimating Q is now a problem of *regression*.

For each arm $A_i$, given:

$D = \{(X_1, r_1), (X_2, r_2), \ldots, (X_n, r_n)\}$

and parametric form $f(X, \theta)$

estimate:

$$Q(X, A_i) = \mathbb{E}\left[r_i | X\right]$$

# Linear Q Approximation

Very simple form of function approximator:

$$Q_i(X) = \theta \cdot X = \sum_{p=1}^{d} \theta_p X[p]$$

from which we can derive least-squares estimator

$$\theta = (\bar{X}\bar{X}^T)^{-1}\bar{X}^T r$$

where:

$\bar{X}$ is a *nxd* matrix of data points

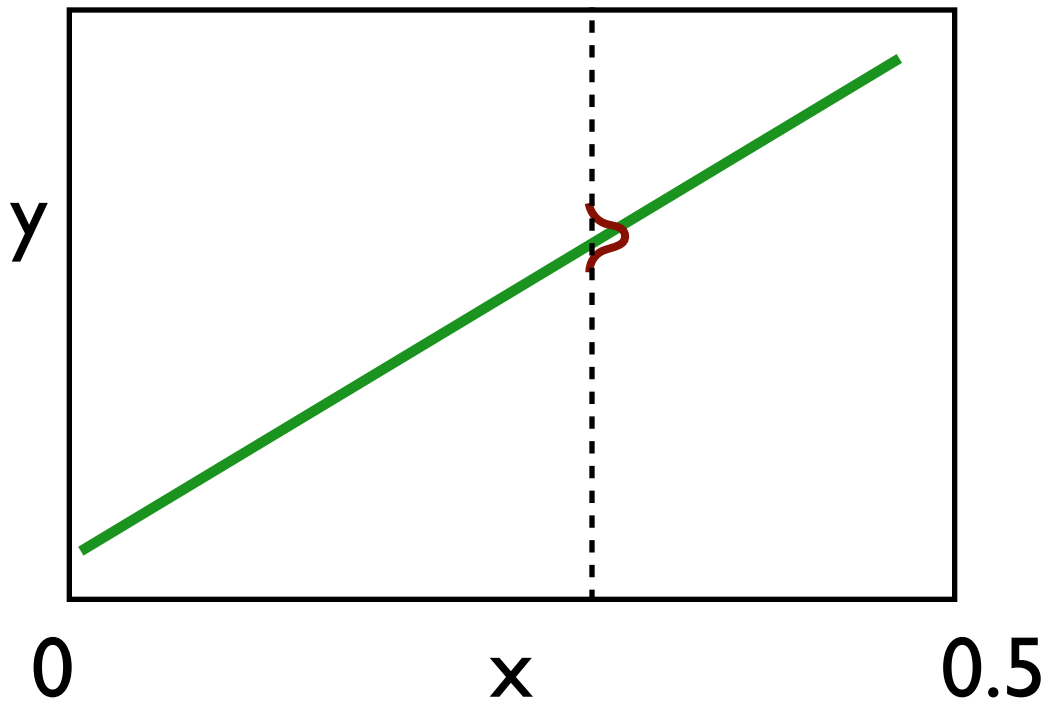
*r* is an *nx1* vector of rewards

# Exploration Strategy

OK, that's how we fit Q. Now how might be select actions?

- e-greedy
- Can we do something like UCB?

Linear regression gives you an estimated variance, too!



# LinUCB

Pick the action that maximizes:

$$\arg\max_i \left( X_t \cdot \theta_i + \alpha \sqrt{X_t^T (\bar{X}^T \bar{X})^{-1} X_t} \right)$$

where $\alpha = 1 + \sqrt{\ln(2/\delta)/2}$
and $\delta$ is a probability tolerance.

variance

Assumptions:

- Linearity
- Constant variance

# Generalizing

More general linear Q approximation:

$$Q_i(X) = \theta \cdot \Phi(X)$$

where $\Phi$ is a set of ***basis functions*** or ***features*** defined over *X*.

Representation is linear in parameters $\theta$

- Easy to fit
- Error surface bowl-shaped
- One minimum

But $\Phi$ can be arbitrarily complex:

- Expansions (e.g., Taylor series, Fourier)
- Can represent complex functions

# Example Application

Li, Chu, Langford, Shapire. *A Contextual-Bandit Approach to Personalized News Article Recommendation*. WWW 2010.

**Figure 1: A snapshot of the "Featured" tab in the Today Module on Yahoo! Front Page. By default, the article at F1 position is highlighted at the story position.**

# News Recommendations

*Offline* evaluation required.

Data set gathering:

- Arms selected at random
- Evaluation: throw away data not according to $\pi$
- Unbiased estimator of value of $\pi$

Data elements:

- Article show in F1
- Article user data
- Whether or not user clicks
- May 1st 2009: 4.7 million views for training
- May 3rd-9th: 36 million views for evaluation

# News Recommendations

Features (X):

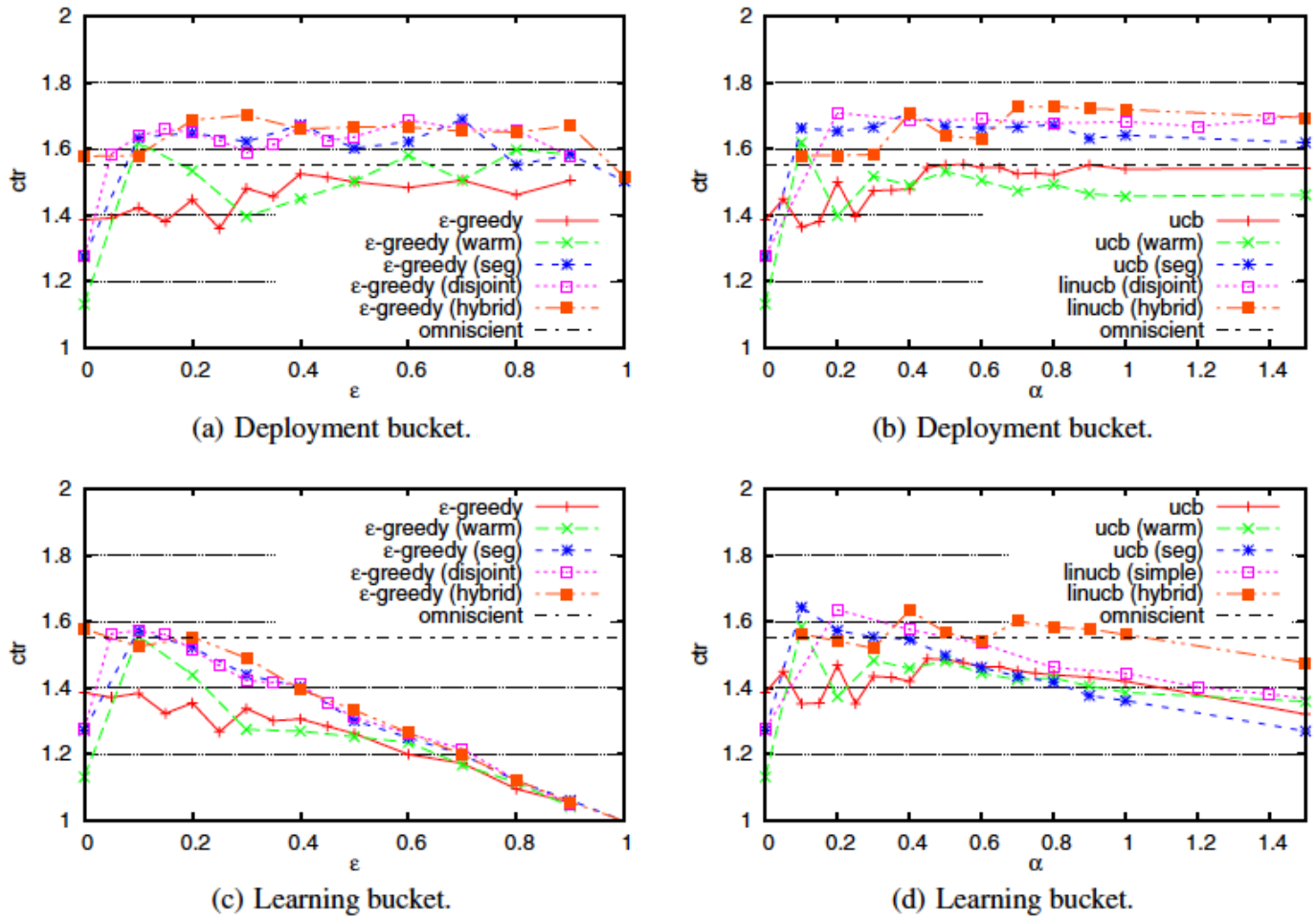
User

- Gender (2 classes)
  Age discretized to 10 segments
- 200 geographic features (metropolitan area, etc.)
- 1000 binary behavioral categories
- consumption history

Story

- URL categories (10s of classes from Yahoo!, set by URL)
- 10s of topics tagged by human editors

# Results




(a) Deployment bucket.

(b) Deployment bucket.

(c) Learning bucket.

(d) Learning bucket.

**Figure 2: Parameter tuning: CTRs of various algorithms on the one-day tuning dataset.**

# Results

| algorithm | size = 100% | | size = 30% | | size = 20% | | size = 10% | | size = 5% | | size = 1% | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | deploy | learn | deploy | learn | deploy | learn | deploy | learn | deploy | learn | deploy | learn |
| $\epsilon$-greedy | 1.596<br>0% | 1.326<br>0% | 1.541<br>0% | 1.326<br>0% | 1.549<br>0% | 1.273<br>0% | 1.465<br>0% | 1.326<br>0% | 1.409<br>0% | 1.292<br>0% | 1.234<br>0% | 1.139<br>0% |
| ucb | 1.594<br>0% | 1.569<br>18.3% | 1.582<br>2.7% | 1.535<br>15.8% | 1.569<br>1.3% | 1.488<br>16.9% | 1.541<br>5.2% | 1.446<br>9% | 1.541<br>9.4% | 1.465<br>13.4% | 1.354<br>9.7% | 1.22<br>7.1% |
| $\epsilon$-greedy (seg) | 1.742<br>9.1% | 1.446<br>9% | 1.652<br>7.2% | 1.46<br>10.1% | 1.585<br>2.3% | 1.119<br>−12% | 1.474<br>0.6% | 1.284<br>−3.1% | 1.407<br>0% | 1.281<br>−0.8% | 1.245<br>0.9% | 1.072<br>−5.8% |
| ucb (seg) | 1.781<br>11.6% | 1.677<br>26.5% | 1.742<br>13% | 1.555<br>17.3% | 1.689<br>9% | 1.446<br>13.6% | 1.636<br>11.7% | 1.529<br>15.3% | 1.532<br>8.7% | 1.32<br>2.2% | 1.398<br>13.3% | 1.25<br>9.7% |
| $\epsilon$-greedy (disjoint) | 1.769<br>10.8% | 1.309<br>−1.2% | 1.686<br>9.4% | 1.337<br>0.8% | 1.624<br>4.8% | 1.529<br>20.1% | 1.529<br>4.4% | 1.451<br>9.4% | 1.432<br>1.6% | 1.345<br>4.1% | 1.262<br>2.3% | 1.183<br>3.9% |
| linucb (disjoint) | 1.795<br>12.5% | 1.647<br>24.2% | 1.719<br>11.6% | 1.507<br>13.7% | 1.714<br>10.7% | 1.384<br>8.7% | 1.655<br>13% | 1.387<br>4.6% | 1.574<br>11.7% | 1.245<br>−3.5% | 1.382<br>12% | 1.197<br>5.1% |
| $\epsilon$-greedy (hybrid) | 1.739<br>9% | 1.521<br>14.7% | 1.68<br>9% | 1.345<br>1.4% | 1.636<br>5.6% | 1.449<br>13.8% | 1.58<br>7.8% | 1.348<br>1.7% | 1.465<br>4% | 1.415<br>9.5% | 1.342<br>8.8% | 1.2<br>5.4% |
| linucb (hybrid) | 1.73<br>8.4% | 1.663<br>25.4% | 1.691<br>9.7% | 1.591<br>20% | 1.708<br>10.3% | 1.619<br>27.2% | 1.675<br>14.3% | 1.535<br>15.8% | 1.588<br>12.7% | 1.507<br>16.6% | 1.482<br>20.1% | 1.446<br>27% |

**Table 1: Performance evaluation: CTRs of all algorithms on the one-week evaluation dataset in the deployment and learning buckets (denoted by "deploy" and "learn" in the table, respectively). The numbers with a percentage is the CTR lift compared to $\epsilon$-greedy.**

# Contextual Bandits

When is this model appropriate?
When is it not?

# Next

Thursday: what if actions have consequences (beyond reward)?