Making sequential decisions under uncertainty

You have N choices

In every round $t = 1, 2, \ldots, T$

- Choose one out of N, using only past observations
- observe (uncertain) reward, feedback

Maximize total reward, other objective

Examples

- Movie Recommendation, online advertising
  - Observe clicks, likes
- Portfolio optimization
  - Observe money made (reward), how stocks behaved, market index changes etc.
- Pricing and Revenue management
  - Observe demand, revenue from sales
- Game playing
  - Observe improvement in the player position, other game state
- Robot navigation and control
  - Observe performance and accuracy

Common elements of sequential decision making models

- Can use only past feedback
    - Feedback from time steps before this
- Past feedback has some relation to future reward

- Learn from past to predict future and optimize

Distinctions

How past is related to future?

- Stochastic process
  - IID
  - Markovian
- Adversarial
  - An arbitrary sequence of feedback, but restricted in certain specific ways

Distinctions

- Full information models
  - *Reward* for your decision but *feedback* on instantaneous performance of all possible choices, **Can answer what-if**
  - E.g. buying stocks, pay-per-impression advertising, bidding/offer model of selling goods
  - Online packing, online matching, online convex programming
- Limited feedback models
  - E.g., Feedback only on performance of your decision
  - E.g. movie recommendations, pay-per-click advertising, posted price model of selling goods, game playing
  - Multi-armed bandits, Reinforcement learning

Managing exploitation-exploitation tradeoff

The **multi-armed bandit problem**
(Thompson 1933; Robbins 1952)

Multiple rigged slot machines in a casino.
Which one to put money on?

- Try each one out

WHEN TO STOP TRYING (EXPLORATION) AND START PLAYING (EXPLOITATION)?

Multi-armed bandit model

- N arms (choices). Pulling an arm generates a reward
- In each round t , pull one arm I_t of the N arms.
- Observe stochastic reward r_t
- Maximize $\sum_t r_t$

- You do not know what you would get by pulling another arr
- Limited feedback or "bandit feedback"

Key properties of the model

- You observe the feedback for only the decision you make
  - WHAT IF had pulled another arm?
  - Need to explore
- Natural instinct is to take best choice according to current data
  - Exploitation
- Algorithms to manage the Exploration-Exploitation tradeoff
  - Adapt from learnings so far, to waste less time on exploring bad choices

Examples: Clinical trials

On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples

William R. Thompson
Biometrika
Vol. 25, No. 3/4 (Dec., 1933), pp. 285-294

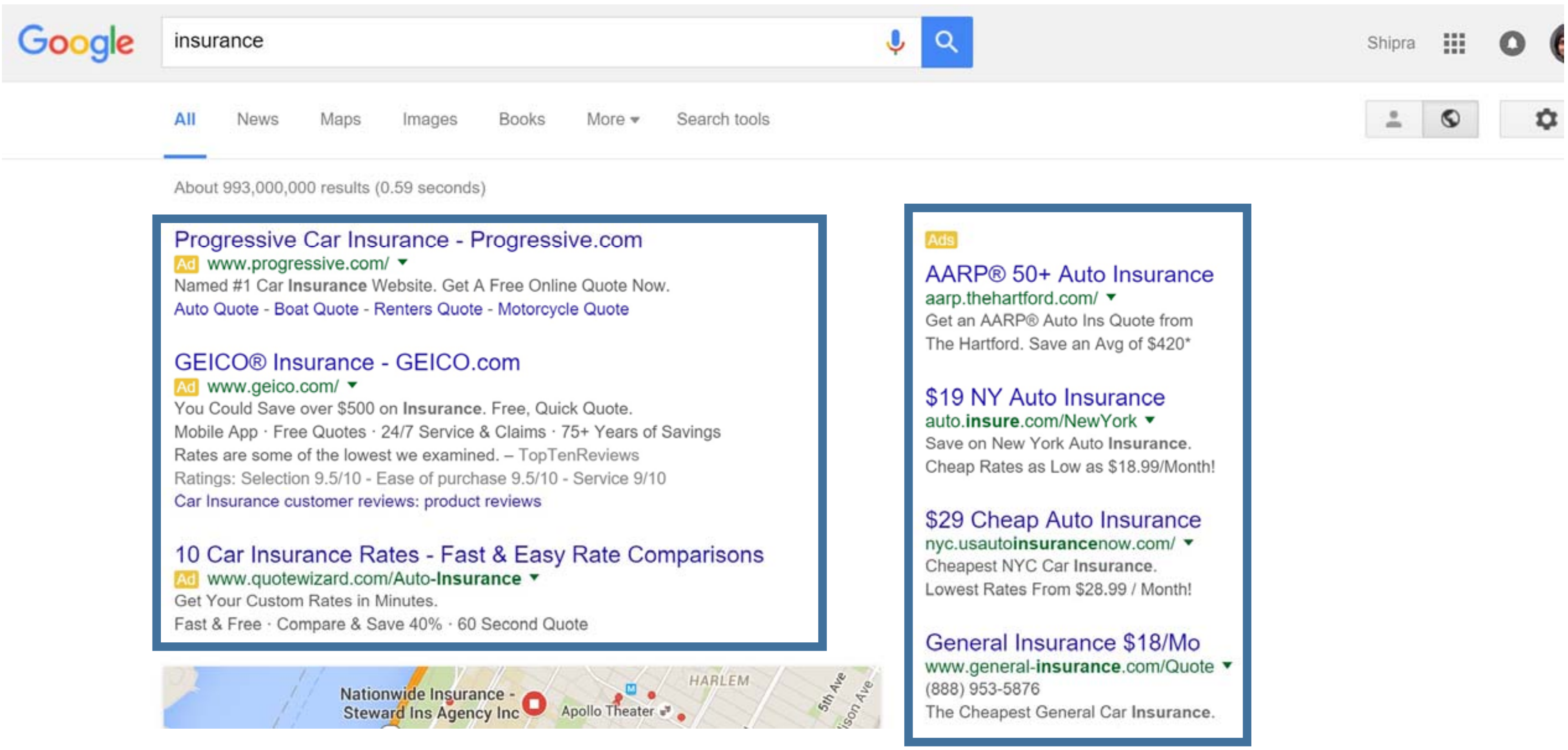
- Patients arrive sequentially
- Pick one out of N treatments
- Cure as many patients as possible
- You can observe the performance of a treatment by administration

Administer the currently best performing treatment, Or

Try a less understood treatment?

Response depend on patient features, one patient informs about others: Contextual bandits

Internet advertising: pick a few from N ads



Chances to click can depends on the search query : Contextual bandits

Chances to click can depend on other ads: Bandits with assortments (MNL-bandit)

Examples: Dynamic Pricing

- A Seller with goods to price
- N possible discrete prices
- Observes sales (or no sale) only for the offered price
- Explore different prices or pick the best performing price so far?

Other considerations:

Continuous space of prices: Continuum armed bandits

Often involves inventory constraints: Bandits with global constraints

Reinforcement learning

- Limited feedback model with uncertainty generated by Markovian stochastic process
- Reward is at time t is determined by the “action or arm” and “state” of the system
- At time t
  - Observe the current “state” of the system s_t
  - Take action a_t
  - Observe reward r_t, from fixed **unknown reward distribution**
  - System transitions to next state s_{t+1} with **unknown probability** $\mathrm{P}(s_t, a_t, s_{t+1})$
- Maximize total reward or discounted reward

Reinforcement learning

- Trial-and-error
  - Explore-exploit
- Explore different actions, and observe reward and state transitions to learn
  - which actions have high reward in a given state
  - Which actions take you to good states
- Adapt exploration to past observations – learn from past mistakes
  - Limit exploration of bad states and bad actions

Game playing

- A computer algorithm playing game like Atari Breakout
- Maximize the score
- Need to make moves sequentially
- State: what you see on screen
- Limited feedback: Can observe only the outcome of move made
- Solve it by reinforcement learning – use only feedback, trial and error