CS 774A Optimization Techniques

Combinatorial MAB Problem

Anurag Sharma[1] Shivam Khandelwal[1]

[1]Department of Mathematics and Statistics
IIT Kanpur

November 16, 2016

Outline

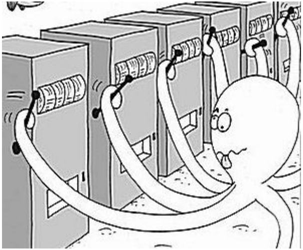
1. Introduction
   - Motivation
   - Exploration - Exploitation Dilemma
2. Combinatorial Multi Arm Bandit
   - CMAB Setting
   - Stochastic CMAB: Prior work
3. Mid term plan of action
4. Two main algorithms studied
   - CUCB algorithm
   - Application specifically for CUCB
   - PMC Bandit
   - Social Influence maximization bandit
   - ESCB Algorithm

Outline

1. Introduction
   - Motivation
   - Exploration - Exploitation Dilemma
2. Combinatorial Multi Arm Bandit
   - CMAB Setting
   - Stochastic CMAB: Prior work
3. Mid term plan of action
4. Two main algorithms studied
   - CUCB algorithm
   - Application specifically for CUCB
   - PMC Bandit
   - Social Influence maximization bandit
   - ESCB Algorithm

- K arms (or actions)
- Each time t, each arm i pays of a bounded real valued reward $x_i(t)$ say in [0,1].
- Each time t, the learner chooses a single arm $i_t \in \{1, \ldots K\}$ and receives reward $x_{i_t}(t)$. The goal is to maximize the return.

Figure: Multi Arm Slot machine



Outline

1. Introduction
   - Motivation
   - Exploration - Exploitation Dilemma
2. Combinatorial Multi Arm Bandit
   - CMAB Setting
   - Stochastic CMAB: Prior work
3. Mid term plan of action
4. Two main algorithms studied
   - CUCB algorithm
   - Application specifically for CUCB
   - PMC Bandit
   - Social Influence maximization bandit
   - ESCB Algorithm

Exploration - Exploitation Dilemma

- Online decision-making involves a fundamental choice:
  Exploitation Make the best decision given current information
  Exploration Gather more information

Exploration - Exploitation Dilemma

- Online decision-making involves a fundamental choice:
  Exploitation Make the best decision given current information
  Exploration Gather more information
- The best long-term strategy may involve short-term sacrifices
- Gather enough information to make the best overall decisions

Exploration - Exploitation Dilemma

- Online decision-making involves a fundamental choice:
  Exploitation Make the best decision given current information
  Exploration Gather more information
- The best long-term strategy may involve short-term sacrifices
- Gather enough information to make the best overall decisions

Examples:

1. Restaurant Selection
   Exploitation Go to your favourite restaurant
   Exploration Try a new restaurant

Outline

1 Introduction
- Motivation
- Exploration - Exploitation Dilemma

2 Combinatorial Multi Arm Bandit
- CMAB Setting
- Stochastic CMAB: Prior work

3 Mid term plan of action

4 Two main algorithms studied
- CUCB algorithm
- Application specifically for CUCB
- PMC Bandit
- Social Influence maximization bandit
- ESCB Algorithm

- At each round an arm M is selected from finite set $\mathbb{M} \subset \{0,1\}^d$
- Reward recieved is $M^T X(n) = \sum_{i=1}^{i=d} M_i X_i(n)$
- Reward vector is unknown and $\|M\|_1 = m \; \forall M \in \mathbb{M}$
- feedback framework:
    - Semibandit: $X_i(n)$ is revealed $\forall i$ (only if $M_i = 1$)
    - Bandit: Only reward $M^T X(n)$ is revealed

Aim

Based on the feedback received upto round $n-1$, select an arm at round n such that:

- Cumulative reward over a given time horizon consisting of T rounds is maximized
- Regret $R(T)$ is minimized

$$R(T) = \max_{M \in \mathbb{M}} \mathbb{E}[\sum_{n=1}^{T} M^T X(n)] - \mathbb{E}[\sum_{n=1}^{T} M(n)^T X(n)]$$

Challenge: Very large number of arms, i.e., in its combinatorial structure: the size of $\mathbb{M}$ grows as d^m

Outline

1. Introduction
   - Motivation
   - Exploration - Exploitation Dilemma
2. Combinatorial Multi Arm Bandit
   - CMAB Setting
   - Stochastic CMAB: Prior work
3. Mid term plan of action
4. Two main algorithms studied
   - CUCB algorithm
   - Application specifically for CUCB
   - PMC Bandit
   - Social Influence maximization bandit
   - ESCB Algorithm

Prior Work

Some quantification measures:

- Dimension $d = \dim \mathbb{M}$

Prior Work

Some quantification measures:

- Dimension $d = \dim \mathbb{M}$
- Decision size $= m$; because we have to choose m arms at a time.

Prior Work

Some quantification measures:

- Dimension $d = \dim \mathbb{M}$
- Decision size $= m$; because we have to choose m arms at a time.
- Optimality gap $\Delta = \min_{M \neq M^*}(\mu - \mu^*(M))$

Prior Work

Some quantification measures:

- Dimension $d = \dim \mathbb{M}$
- Decision size $= m$; because we have to choose m arms at a time.
- Optimality gap $\Delta = \min_{M \neq M^*}(\mu - \mu^*(M))$
- Regret Upper Bounds
  - LLR (Gai 2012) : $\mathcal{O}(\frac{m^3 d \Delta_{max}}{\Delta_{min}^2} log(T))$
  - CUCB (Chen 2013) : $\mathcal{O}(\frac{m^2 d}{\Delta_{min}} log(T))$
  - ESCB (Combes 2015) : $\mathcal{O}(\frac{\sqrt{m} d \Delta_{max}}{\Delta_{min}^2} log(T))$

Prior Work

Some quantification measures:

- Dimension $d = \dim \mathbb{M}$
- Decision size $= m$; because we have to choose m arms at a time.
- Optimality gap $\Delta = \min_{M \neq M^*}(\mu - \mu^*(M))$
- Regret Upper Bounds
    - LLR (Gai 2012) : $\mathcal{O}(\frac{m^3 d \Delta_{max}}{\Delta_{min}^2} log(T))$
    - CUCB (Chen 2013) : $\mathcal{O}(\frac{m^2 d}{\Delta_{min}} log(T))$
    - ESCB (Combes 2015) : $\mathcal{O}(\frac{\sqrt{m} d \Delta_{max}}{\Delta_{min}^2} log(T))$

Mid term plan of action

- Study Comb - MAB in stochastic and specific combinatorial setting
- Identifying constraints for $\sqrt{m}$ improvement in bounds in ESCB
- Estimate current bounds under relaxed conditions

Outline

1. Introduction
   - Motivation
   - Exploration - Exploitation Dilemma
2. Combinatorial Multi Arm Bandit
   - CMAB Setting
   - Stochastic CMAB: Prior work
3. Mid term plan of action
4. Two main algorithms studied
   - CUCB algorithm
   - Application specifically for CUCB
   - PMC Bandit
   - Social Influence maximization bandit
   - ESCB Algorithm

Combinatorial UCB algorithm

Key features of CUCB algorithm:

- Does not assume the direct knowledge on how super arms are formed from underlying arms or how the reward is computed.

Combinatorial UCB algorithm

Key features of CUCB algorithm:

- Does not assume the direct knowledge on how super arms are formed from underlying arms or how the reward is computed.
- Assume the availability of an offline computation oracle that takes such knowledge as well as the expectations of outcomes of all arms as input and computes the optimal super arm with respect to the input.

Combinatorial UCB algorithm

Key features of CUCB algorithm:

- Does not assume the direct knowledge on how super arms are formed from underlying arms or how the reward is computed.
- Assume the availability of an offline computation oracle that takes such knowledge as well as the expectations of outcomes of all arms as input and computes the optimal super arm with respect to the input.
- Such an existence oracle is also used in other approximate algorithms which are computationally intractable (NP Hard) and approximate solutions are suggested.

Combinatorial UCB algorithm

Key features of CUCB algorithm:

- Does not assume the direct knowledge on how super arms are formed from underlying arms or how the reward is computed.
- Assume the availability of an offline computation oracle that takes such knowledge as well as the expectations of outcomes of all arms as input and computes the optimal super arm with respect to the input.
- Such an existence oracle is also used in other approximate algorithms which are computationally intractable (NP Hard) and approximate solutions are suggested.
- Regret Upper Bounds
  - CUCB (Chen 2013) : $\mathcal{O}(\frac{m^2 d}{\Delta_{min}} log(T))$

Combinatorial UCB algorithm – Oracle Business

How does the analysis work with oracle:

- Framework separates the online learning task and the offline computation task

Combinatorial UCB algorithm – Oracle Business

How does the analysis work with oracle:

- Framework separates the online learning task and the offline computation task
- Oracle does offline computation task, which uses the domain knowledge of the problem instance

Combinatorial UCB algorithm – Oracle Business

How does the analysis work with oracle:

- Framework separates the online learning task and the offline computation task
- Oracle does offline computation task, which uses the domain knowledge of the problem instance
- CMAB algorithm takes care of the online learning task, and is oblivious to the domain knowledge of the problem instance

Combinatorial UCB algorithm – Oracle Business

How does the analysis work with oracle:

- Framework separates the online learning task and the offline computation task
- Oracle does offline computation task, which uses the domain knowledge of the problem instance
- CMAB algorithm takes care of the online learning task, and is oblivious to the domain knowledge of the problem instance
- $\alpha\beta$ Approximate oracle: It takes an input of μ and outputs a super arm $\in S$ such that $\Pr[r_\mu(S) \geq \alpha \text{opt}\mu] \geq \beta$. β is the probability of success for the oracle.

Combinatorial UCB algorithm – Setup

Assumptions:

- A superarm S is the set of base arms, $S \subset [m]$

Combinatorial UCB algorithm – Setup

Assumptions:

- A superarm S is the set of base arms, $S \subset [m]$
- In round t, superarm S_t^A is played according to algo A

Combinatorial UCB algorithm – Setup

Assumptions:

- A superarm S is the set of base arms, $S \subset [m]$
- In round t, superarm S_t^A is played according to algo A
- **Outcomes of all played base arms are observed**

Combinatorial UCB algorithm – Setup

Assumptions:

- A superarm S is the set of base arms, $S \subset [m]$
- In round t, superarm S_t^A is played according to algo A
- **Outcomes of all played base arms are observed**
- Outcome of an arm i has an unknown distribution with unknown mean μ_i

Combinatorial UCB algorithm – Rewards

Notations used :

- Reward of super-arm S_t^A played in round t, $R_t(S_t^A)$ is the function of outcomes of al played arms

Combinatorial UCB algorithm – Rewards

Notations used :

- Reward of super-arm S_t^A played in round t, $R_t(S_t^A)$ is the function of outcomes of al played arms
- Expected reward of playing an arm S, $E(R_t(S_t^A))$ $(= r_\mu(S)$ $)$depends only on S and mean vector of arms μ

Combinatorial UCB algorithm – Rewards

Notations used :

- Reward of super-arm S_t^A played in round t, $R_t(S_t^A)$ is the function of outcomes of al played arms
- Expected reward of playing an arm S, $E(R_t(S_t^A))$ $(= r_\mu(S)$)depends only on S and mean vector of arms μ
- Optimal Reward: Optμ $= Max_S\ r_\mu(S)$

Handling Non-Linear Rewards

Two mild assumptions on $r_\mu(S)$

- Monotonicity: If $\mu \leq \mu^{'}$ (pairwise), $r_\mu(S) \leq r_{\mu^{'}}(S)$ for all superarms S

Handling Non-Linear Rewards

Two mild assumptions on $r_\mu(S)$

- Monotonicity: If $\mu \leq \mu^{'}$ (pairwise), $r_\mu(S) \leq r_{\mu'}(S)$ for all superarms S
- Boundedness: There exists a strictly increasing function $f()$ such that for any two expectation vectors μ and $\mu_{'}$,
  $|r_\mu(S) - r_{\mu'}(S)| \leq f(\Delta)$, where $\Delta = \text{Max } |\mu_i - \mu_i^{'}|$

Handling Non-Linear Rewards

Two mild assumptions on $r_\mu(S)$

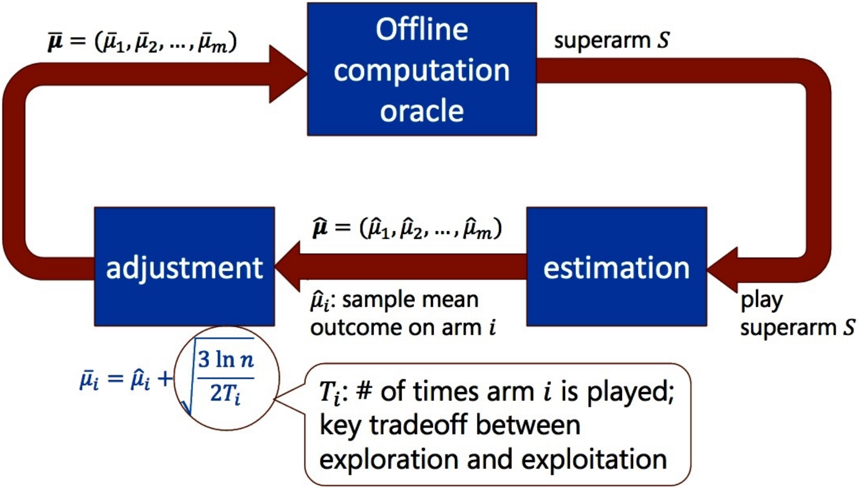
- Monotonicity: If $\mu \leq \mu^{'}$ (pairwise), $r_\mu(S) \leq r_{\mu'}(S)$ for all superarms S
- Boundedness: There exists a strictly increasing function $f()$ such that for any two expectation vectors μ and $\mu_{'}$,
  $|r_\mu(S) - r_{\mu'}(S)| \leq f(\Delta)$, where $\Delta = \text{Max } |\mu_i - \mu_i^{'}|$
- A large class of reward functions satisfy these conditions (linear and non-linear)

(α, β) Approximation Regret

We compare against the $\alpha\beta$ fraction of the optimal:

$$Regret = n * \alpha * \beta * opt_{\mu} - E(\sum_{i=1}^{n} r_{\mu}(S_t^A)) \tag{1}$$

The Algorithm



Regret bound

Theorem: The $(\alpha\beta)$-approximation regret of CUCB algorithm in *n* rounds using an approximation oracle is at most:

$$\sum_{i\in[m],\Delta^i_{\min}>0}\left(\frac{6\ln n\cdot\Delta^i_{\min}}{(f^{-1}(\Delta^i_{\min}))^2}+\int_{\Delta^i_{\min}}^{\Delta^i_{\max}}\frac{6\ln n}{(f^{-1}(x))^2}\mathrm{d}x\right)+\left(\frac{\pi^2}{3}+1\right)\cdot m\cdot\Delta_{\max}.$$

- Δ^i_{min} is defined as the minimum gap between αopt_μ and reward of a bad super arm containing i
- $\Delta_{min} = Min_i\ \Delta^i_{min}$

Outline

1. Introduction
   - Motivation
   - Exploration - Exploitation Dilemma
2. Combinatorial Multi Arm Bandit
   - CMAB Setting
   - Stochastic CMAB: Prior work
3. Mid term plan of action
4. Two main algorithms studied
   - CUCB algorithm
   - Application specifically for CUCB
   - PMC Bandit
   - Social Influence maximization bandit
   - ESCB Algorithm

CUCB with non linear rewards

- Online submodular maximization problem

CUCB with non linear rewards

- Online submodular maximization problem
  - Probabilistic maximum coverage bandit
  - Social influence maximization bandit

Outline

1 Introduction
- Motivation
- Exploration - Exploitation Dilemma

2 Combinatorial Multi Arm Bandit
- CMAB Setting
- Stochastic CMAB: Prior work

3 Mid term plan of action

4 Two main algorithms studied
- CUCB algorithm
- Application specifically for CUCB
- PMC Bandit
- Social Influence maximization bandit
- ESCB Algorithm

PMC Bandit details

- Probabilistic Maximum coverage problem has an input of a weighted bipartite graph G = (L,R,E)
- Aim is to choose a set of edges $S \subset L$ of size k that maximizes the activated number of nodes in R

PMC Bandit details

- Probabilistic Maximum coverage problem has an input of a weighted bipartite graph G = (L,R,E)
- Aim is to choose a set of edges $S \subset L$ of size k that maximizes the activated number of nodes in R
- A node u activates a connected node v with a probability $p(u, v)$ (which is the edge weight)
- Example: User - Advt in webpages

PMC Bandit details

- Probabilistic Maximum coverage problem has an input of a weighted bipartite graph G = (L,R,E)
- Aim is to choose a set of edges $S \subset L$ of size k that maximizes the activated number of nodes in R
- A node u activates a connected node v with a probability $p(u, v)$ (which is the edge weight)
- Example: User - Advt in webpages
- PMC Bandit: Probability weights are unknown
- Submodular set function maximization technique shows the existence of oracle and hence we can use methods of CUCB

Outline

1. Introduction
   - Motivation
   - Exploration - Exploitation Dilemma
2. Combinatorial Multi Arm Bandit
   - CMAB Setting
   - Stochastic CMAB: Prior work
3. Mid term plan of action
4. Two main algorithms studied
   - CUCB algorithm
   - Application specifically for CUCB
   - PMC Bandit
   - Social Influence maximization bandit
   - ESCB Algorithm

SIM Bandit details

- Social influence maximization problem has an input of a Directed graph G = (V,E)
- Works as a diffusion process with probability of activation being $p(u, v)$

SIM Bandit details

- Social influence maximization problem has an input of a Directed graph G = (V,E)
- Works as a diffusion process with probability of activation being $p(u, v)$
- Reward is is the total number of activated nodes in the end.
- Each edge is an arm and superedge is a set of outgoing edges from at most k nodes

Outline

1 Introduction
- Motivation
- Exploration - Exploitation Dilemma

2 Combinatorial Multi Arm Bandit
- CMAB Setting
- Stochastic CMAB: Prior work

3 Mid term plan of action

4 Two main algorithms studied
- CUCB algorithm
- Application specifically for CUCB
- PMC Bandit
- Social Influence maximization bandit
- ESCB Algorithm

- Assigns index to each arm: arm with largest index chosen for exploratio

- Assigns index to each arm: arm with largest index chosen for exploratio
- Indices are the natural extension of Upper Confidence Bound (UCB) and KL-UCB algorithms

- Assigns index to each arm: arm with largest index chosen for exploratio
- Indices are the natural extension of Upper Confidence Bound (UCB) and KL-UCB algorithms
- ESCB improve over LLR and and CUCB by the multiplicative factor of $\sqrt{m}$

Summary

- Presented Multi Arm Bandit Problem and Combinatorial setting in MAB framework
- Overview of CUCB and ESCB algorithm
- Presented two application where CUCB with non linear rewards can be applied
- Tried to understand and improve bounds, but the direction of application of CUCB looks more promising