Combinatorial Online Learning
组合在线学习
Wei Chen 陈卫
Microsoft Research Asia
What is Combinatorial Online Learning?
Consider GPS routing suggestion
Or news recommendation
Are these just recommender systems?

• No.

• Traditional recommender systems
  – Relatively static
  – Offline learn user and item features, then make online recommendation

• Online learning
  – Fast feedback loop: online learning features and online optimization
  – Iterative learning and optimization
Online learning: the iterative feedback loop

- Re-optimize based on updated statistics
- Observe feedback (road segment delay, news click), update learning statistics
- New solution (e.g., routes, news combinations)
- Updated statistics
Why combinatorial?

• The solution is not a simple item, it is a combinatorial item:
  – GPS routing: a combination of road segments
  – News recommendation: combination of different type of news a user may be interested in

• For many combinatorial optimization problems, when the input is uncertain, they may be turned into an online learning problem
Combinatorial online learning

• Iterative feedback loop between optimization and learning
  – Handle uncertainty in the environment

• Action to optimize is combinatorial

• (Combinatorial) online learning is the foundation of reinforcement learning (强化学习) in AI
  – Provide solid theoretical guidance to reinforcement learning
  – Theoretical treatment to the key tradeoff between exploration (探索) and exploitation (守成) in reinforcement learning
My Recent Research Effort

- ICML’13: general combinatorial multi-armed bandit (CMAB) framework, apply to non-linear rewards, approximation oracle
- ICML’14: combinatorial partial monitoring
- NIPS’14: combinatorial pure exploration
- NIPS’15: online greedy learning
- JMLR’16: CMAB with probabilistically triggered arms (CMAB-T)
- ICML’16: contextual combinatorial cascading bandits
- NIPS’16: CMAB with general reward functions
- NIPS’17: Improving the regret bound for CMAB-T
Background: Multi-armed Bandit
Multi-armed bandit: the canonical OL problem

• There are $m$ arms (machines)
• Arm $i$ has an unknown reward distribution on $[0,1]$ with unknown mean $\mu_i$
  – best arm $\mu^* = \max \mu_i$
• In each round, the player selects one arm to play and observes the reward
Multi-armed bandit problem

• Performance metric: Regret:
  – Regret after playing \( T \) rounds = \( T \mu^* - \mathbb{E} \left[ \sum_{t=1}^{T} R_t(i^A_t) \right] \)

• Objective: minimize regret in \( T \) rounds

• Balancing exploration-exploitation tradeoff
  – exploration (探索): try new arms
  – exploitation (守成): keep playing the best arm so far

• Known results:
  – UCB1 (Upper Confidence Bound) [Auer, Cesa-Bianchi, Fischer 2002]
    • Distribution-dependent bound \( O(\log T \sum_{i: \Delta_i > 0} 1/\Delta_i) \), \( \Delta_i = \mu^* - \mu_i \), match lower bound
    • Distribution-independent bound \( O(\sqrt{mT \log T}) \), tight up to a factor of \( \sqrt{\log T} \)
Combinatorial Multi-armed Bandit: Framework and the General Solution

Joint work with Yajun Wang (Microsoft), Yang Yuan (Cornell), Qinshi Wang (Princeton)
ICML’2013, JMLR’2016
Motivating application: Display ad placement

- Bipartite graph of pages and users who are interested in certain pages
  - Each edge has a click-through probability
- Find $k$ pages to put ads to maximize total number of users clicking through the ad
- When click-through probabilities are known, can be solved by approximation
- Question: how to learn click-through prob. while doing optimization?
Naïve application of MAB

• Every set of k webpages is treated as an arm
• Reward of an arm is the total click-through counted by the number of people
• Main issues
  – combinatorial explosion
  – ad-user click-through information is wasted
• Other possible issues
  – Offline optimization problem may already be hard
  – The reward of a combinatorial action may not be linear on its components
  – The reward may depend not only on the means of its component rewards
Combinatorial multi-armed bandit (CMAB) framework

- A super arm \( S \in \mathcal{S} \) is a set of (base) arms, \( S \subseteq [m] \)
  - \( \mathcal{S} \) is the set of possible super arms
- In round \( t \), a super arm \( S_t^A \) is played according algo \( A \)
- When a super arm \( S \) is played, all based arms in \( S \) are played
- Outcomes of all played base arms are observed --- semi-bandit feedback
- Outcome of arm \( i \in [m] \) has an unknown distribution on \([0,1]\) with unknown mean \( \mu_i \)
Rewards in CMAB

- Reward of super arm $S_t^A$ played in round $t$, $R_t(S_t^A)$, is a function of the outcomes of all played arms.
- Expected reward of playing arm $S$, $\mathbb{E}[R_t(S)]$, only depends on $S$ and the vector of mean outcomes of arms, $\mu = (\mu_1, \mu_2, \ldots, \mu_m)$, denoted $r_{\mu}(S)$
  - e.g. linear rewards, or independent Bernoulli random variables
  - generalization to be discussed later
- Optimal reward: $\text{opt}_\mu = \max_{S \in \mathcal{S}} r_{\mu}(S)$
Offline computation oracle ---
allow approximations and failure probabilities

• \((\alpha, \beta)\)-approximation oracle:
  – Input: vector of mean outcomes of all arms \(\mu = (\mu_1, \mu_2, \ldots, \mu_m)\),
  – Output: a super arm \(S\), such that with probability at least \(\beta\) the expected reward of \(S\) under \(\mu\), \(r_\mu(S)\), is at least \(\alpha\) fraction of the optimal reward:
    \[
    \Pr[r_\mu(S) \geq \alpha \cdot \text{opt}_\mu] \geq \beta
    \]
(\alpha, \beta)-Approximation regret

- Compare against the \( \alpha \beta \) fraction of the optimal

\[
\text{Regret} = T \cdot \alpha \beta \cdot \text{opt}_\mu - \mathbb{E}[\sum_{i=1}^{T} r_\mu(S_t^A)]
\]

- Oracle treatment: modular, ignore all following offline factors from the online learning part
  - combinatorial structure
  - reward function
  - how oracle computes the solution
Classical MAB as a special case

- Each super arm is a singleton
- Oracle is taking the max, $\alpha = \beta = 1$
Examples of CMAB instances

- Linear CMAB
  - $s$-$t$ Shortest path (for GPS routing)
    - Each edge is an arm, outcome is the random delay on the arm from an unknown distribution
    - Each $s$-$t$ path is a super arm, reward is the sum of edge delays
    - Each round selects an $s$-$t$ path, each edge on the path gives the delay feedback
    - Offline oracle is any shortest path algorithm
    - Minimize the cumulative delay over all rounds
  - Matching (e.g. for crowdsourcing platforms, wireless channel allocation)
  - Spanning tree (e.g. for wireless routing planning)
Examples of CMAB instances

• Nonlinear CMAB
  – Probabilistic max cover (for ad placement)
    • Bipartite graph $G = (L, R, E)$
    • Each edge is a base arm, with Bernoulli distribution
    • Each set of edges linking $k$ webpages is a super arm
    • Reward is the number of users a super covered
      – Nonlinear: 2 webpages covering the same user is counted as 1, not 2
    • Offline problem is NP hard, a greedy algorithm achieves $(1 - \frac{1}{e}, 1)$-approximation
Our solution: CUCB algorithm

\[ \bar{\mu} = (\bar{\mu}_1, \bar{\mu}_2, \ldots, \bar{\mu}_m) \]

Offline computation oracle

super arm \( S \)

observe arm feedback and update sample mean estimate

play super arm \( S \)

\[ \hat{\mu}_i : \text{sample mean outcome on arm } i, \text{ initially 1} \]

adjust sample mean to upper confidence bound

\[ \bar{\mu}_i = \min \left\{ \hat{\mu}_i + \frac{3 \ln t}{2T_i}, 1 \right\} \]

key tradeoff between exploration and exploitation

\( T_i : \# \text{ of times arm } i \text{ is played; initially 0;} \)

\( t : \text{current round number; initially 1} \)
Handling non-linear reward functions ---

two mild assumption on \( r_\mu(S) \)

• Monotonicity
  – if \( \mu \leq \mu' \) (pairwise), \( r_\mu(S) \leq r_\mu'(S) \), for all super arm \( S \)

• Bounded smoothness (a general Lipschitz continuity condition)
  – there exists a bounded smoothness constant \( B_\infty \), such that for any two expectation vectors \( \mu \) and \( \mu' \),
    \[
    |r_\mu(S) - r_\mu'(S)| \leq B_\infty \cdot \|\mu_S - \mu'_S\|_\infty, \text{ where } \|\mu_S - \mu'_S\|_\infty = \max_{i \in S} |\mu_i - \mu'_i| 
    \]
  – Small change in \( \mu_S \) lead to small changes in \( r_\mu(S) \)

• Rewards may not be linear, a large class of functions satisfy these assumptions
Theorem 1: Distribution-dependent bound

- The \((\alpha, \beta)\)-approximation regret of the CUCB algorithm in \(T\) rounds using an \((\alpha, \beta)\)-approximation oracle is at most

\[
\sum_{i \in [m], \Delta^i_{\text{min}} > 0} \frac{12B^2_{\infty} \ln T}{\Delta^i_{\text{min}}} + \left(\frac{\pi^2}{3} + 1\right) \cdot m \cdot \Delta_{\text{max}} = O\left(\sum_{i} \frac{1}{\Delta^i_{\text{min}}} B^2_{\infty} \ln T\right)
\]

- \(-\Delta^i_{\text{min}} (\Delta^i_{\text{max}})\) are defined as the minimum (maximum) gap between \(\alpha \cdot \text{opt}_{\mu}\) and the reward of a bad super arm containing \(i; \Delta_{\text{max}} = \max_i \Delta^i_{\text{max}}\)

- Here, we define the set of bad super arms as \(\mathcal{S}_B = \{S | r_{\mu}(S) < \alpha \cdot \text{opt}_{\mu}\}\)

- Match UCB regret for the classical MAB
Idea of regret analysis

• In each round $t$, if the played super $S$ is bad, count regret $\Delta_S = \alpha \cdot \text{opt}_\mu - r_\mu(S)$.
• Blame one arm $i \in S$ that has been played the least for this regret in round $t$, obtain pair $(i, S)$
• For each $(i, S)$ pair, separate all their occurrences in multiple rounds into two stages
  - Sufficiently-sampled part: $(i, S)$ has appeared more than $\frac{6B_\infty^2 \ln T}{\Delta_S^2}$ times
  - Under-sampled part: $(i, S)$ has appeared at most $\frac{6B_\infty^2 \ln T}{\Delta_S^2}$ times
• For sufficiently-sampled part, all arms in $S$ have enough samples, so
  - W.h.p, all arms in $S$ have good estimates, i.e. $\|\mu_S - \bar{\mu}_S\|_\infty$ and $\|\mu_S - \bar{\mu}_S\|_\infty$ are small
  - then by bounded smoothness, $r_{\bar{\mu}}(S)$ should be close to $r_\mu(S)$, actually $0 \leq r_{\bar{\mu}}(S) - r_\mu(S) < \Delta_S$
  - By monotonicity, and $S$ being the oracle output under $\bar{\mu}$ (with probability $\beta$), $r_{\bar{\mu}}(S) \geq \alpha \cdot \text{opt}_{\bar{\mu}} \geq \alpha \cdot \text{opt}_\mu$, since $\mu \leq \bar{\mu}$ w.h.p
  - So $S$ cannot be bad, unless either sample concentration is violated or offline oracle failed to return an $\alpha$ approximation --- bound regret in this way --- constant cumulative regret $(\frac{\pi^2}{3} + 1) \cdot m \cdot \Delta_{\text{max}}$

• For under-sampled part, each $(i, S)$ appearance causes $i$ to sampled one more time, so at most $\frac{6B_\infty^2 \ln T}{\Delta_S^2}$ appearances of $(i, S)$, and each has regret $\Delta_S$ --- with a careful summation, obtain cumulative regret $O\left(\sum_i \frac{1}{\Delta_{\text{min}}}B_\infty^2 \ln T\right)$
Theorem 2: Distribution-independent bound

- Consider a CMAB problem with an \((\alpha, \beta)\)-approximation oracle. The distribution-independent regret of CUCB in \(T\) round is at most:

\[
B_\infty \sqrt{12mT \ln T} + \left(\frac{\pi^2}{3} + 1\right) \cdot m \cdot \Delta_{\text{max}} = O\left(B_\infty \sqrt{mT \ln T}\right)
\]

- Revise the under-sampled part of Theorem 1: For each arm \(i \in [m],\)
  - if \(\Delta_{\text{min}}^i > \varepsilon_i\), under-sampled regret for \(i\) is \(O\left(\frac{1}{\varepsilon_i} B_\infty^2 \ln T\right)\)
  - if \(\Delta_{\text{min}}^i \leq \varepsilon\), under-sampled regret is for \(i\) is \(O(\varepsilon_i \cdot N_i)\)
    - \(N_i\) is the number of times \(i\) is blamed
  - The best \(\varepsilon_i\) is to make the two terms equal, so under-sampled regret is for \(i\) \(O\left(B_\infty \sqrt{N_i \ln T}\right)\)
  - Overall, under-sampled regret is \(O\left(B_\infty \sqrt{\ln T} \sum_i \sqrt{N_i}\right) = O\left(B_\infty \sqrt{mT \ln T}\right),\) by Jensen’s Inequality and the fact that \(\sum_i N_i = T.\)
Application to ad placement

- Bounded smoothness constant $B_{\infty} = |E|$
- $(1 - \frac{1}{e}, 1)$-approximation regret
  \[ \sum_{i \in E, \Delta^i_{\min} > 0} \frac{12 |E|^2 \ln T}{\Delta^i_{\min}} + \left( \frac{\pi^2}{3} + 1 \right) \cdot |E| \cdot \Delta_{\max} \]
- improvement based on clustered arms is available
Application to linear bandit problems

• Linear bandits: shortest path, matching, spanning tree (in networking literature)
  – Linear expected reward: \( r_\mu(S) = \sum_{i \in S} \mu_i \)

• Our result significantly improves the previous regret bound on linear rewards [Gai et al. 2012]
  – Also provide distribution-independent bound
  – When using 1-norm bounded smoothness condition, tight regret bound matching the lower bound
CMAB with Probabilistically Triggered Arms

Joint work with Yajun Wang (Microsoft), Yang Yuan (Cornell), Qinshi Wang (Princeton)
JMLR’2016, NIPS’2017
Motivation example: influence maximization

- Optimization problem:
  - Given influence parameters on edges
    - Diffusion follows independent cascade model
  - Find \( k \) nodes that generated the largest expected influence

- The online learning version:
  - Influence parameters are unknown
  - Repeatedly select \( k \) seed nodes, observe the cascade, update edge probability estimate, then iterate again
New challenge

• When treating every edge as an arm
  – Probabilistic triggering of arms: The play of some arms may trigger more arms to be played
  – The triggered arms affect the reward

• New dilemma:
  – We need to explore probabilistically triggered arms, since they affect the optimal solution
  – These arms are probabilistically triggered, need more time to learn
CMAB-T framework

- Super arms $S$ are abstracted to actions
- Each action $S$ may probabilistically trigger arms
  - $p_i^{\mu,S}$: probability of action $S$ triggering arm $i$
  - $p^* = \min\{p_i^{\mu,S}: i \in [m], S \in S, p_i^{\mu,S} > 0\}$, minimum positive triggering probability
  - $\tilde{S} = \{i \in [m]: p_i^{\mu,S} > 0\}$, all arms that can be possibly triggered by $S$

- Bounded smoothness: there exists a bounded smoothness constant $B_\infty$, such that for any two expectation vectors $\mu$ and $\mu'$,
  $|r_\mu(S) - r_{\mu'}(S)| \leq B_\infty \cdot \|\mu_{\tilde{S}} - \mu'_{\tilde{S}}\|_\infty$, where $\|\mu_{\tilde{S}} - \mu'_{\tilde{S}}\|_\infty = \max_{i \in \tilde{S}} |\mu_i - \mu'_i|$
  - All arms that may be triggered by $S$ should be considered

理論計算機年会, Oct. 14, 2017
Result on CMAB-T [Chen et al. JMLR’2016]

- Use the same CUCB algorithm
- Distribution-dependent regret: $O \left( \sum_i \frac{1}{p^* \Delta_{\min}} B_{\infty}^2 \ln T \right)$
- Distribution-independent regret: $O \left( B_{\infty} \sqrt{\frac{m T \ln T}{p^*}} \right)$
- Issue: $1/p^*$ could be exponentially large
Improving CMAB-T [Wang and Chen, NIPS’2017]

• Introducing a new triggering-probability modulated (TPM) bounded smoothness condition

• Show that with the TPM condition, $1/p^*$ term in the regret bound is eliminated

• Show that influence maximization bandit and combinatorial cascading bandit satisfy the TPM condition

• Provide a lower bound showing that $1/p^*$ is unavoidable in general CMAB-T instances
TPM condition

• 1-norm TPM bounded smoothness
  – there exists a bounded smoothness constant $B_1$, such that for any two expectation vectors $\mu$ and $\mu'$,

  $$|r_\mu(S) - r_{\mu'}(S)| \leq B_1 \sum_{i \in [m]} p_i^{\mu,S} |\mu_i - \mu'_i|$$

• Intuition: when $i$ is less likely to be triggered by $S$ ($p_i^{\mu,S}$ is small), $i$’s change in its mean has less impact to the change in the expected reward
Regret bounds

• Use the same CUCB algorithm

• Distribution-dependent regret: \( O\left(\sum_i \frac{1}{\Delta_i} B_1^2 K \ln T \right) \)
  
  \(-\ K = \max_{\mathcal{S} \in \mathcal{S}} |\mathcal{S}|, \) the maximum number of arms any action can trigger

• Distribution-independent regret: \( O\left(B_1 \sqrt{mKT \ln T} \right) \)

• Regret analysis is involved, need decomposition of triggering probabilities into geometrically separated bins
  
  \(-\ Also use a reverse amortization trick to improve the 1-norm based regret bound \)
Applications

• Influence maximization bandit
  – TPM condition constant: $B_1 = \tilde{C}$
    • $\tilde{C}$ is the largest number of nodes any node can reach
  – Analysis involves influence tree decomposition to handle loops in the graph, and then use a bottom-up modification technique

• Combinatorial cascading bandit
  – TPM condition constant: $B_1 = 1$
Other CMAB Extensions
What if estimating means of arms is not enough?
Motivating example: graph routing

- **Expected Utility Maximization (EUM) Model**
  - Each edge $i$ has a random delay $X_i$
  - Each routing path is a subset of edges, $S$
  - utility of a routing path $S$: $u(\sum_{i \in S} X_i)$
    - $u(\cdot)$ is nonlinear, modeling risk-averse or risk-prone behavior
  - Goal: maximize $\mathbb{E}[u(\sum_{i \in S} X_i)]$

- **Issue for online learning** (when distributions of $X_i$’s are unknown)
  - only estimating the mean of $X_i$ is not enough

- **Solution**: estimating the entire CDF distribution with DKW inequality
See NIPS’16: Combinatorial Multi-Armed Bandit with General Reward Functions

Joint work with Wei Hu (Princeton), Fu Li, (UT Austin), Jian Li (Tsinghua), Yu Liu (Tsinghua), Pinyan Lu (SUFU)
How to test base arms efficiently to find the best super arm?
Motivating example: Crowdsourcing

• Matching **workers** with **tasks** in a bipartite graph
  - Initial test period: adaptively test worker-task pair performance
  - Goal: at the end of test period, find the best worker-task matching
See NIPS’14: Combinatorial Pure Exploration in Multi-Armed Bandits

joint work with Shouyuan Chen (Microsoft), Tian Lin (Google), Irwin King (CUHK), Michael R. Lyu (CUHK)
Other of my studies

- ICML’14 [with Tian Lin (Google), Bruno Abraohao (Stanford), Robert Kleinberg (Cornell), John Lui (CUHK)]: combinatorial partial monitoring
  - Handling limited feedback
- NIPS’15 [with Tian Lin (Google), Jian Li (Tsinghua)]: online greedy learning
  - How to utilize offline greedy algorithm for online learning
- ICML’16 [with Shuai Li (CUHK), Baoxiang Wang (CUHK), Shengyu Zhang (CUHK)]: contextual combinatorial cascading bandits
  - How to incorporate contextual information
Summary and Future Directions
Overall summary

• Central theme
  – Iterative combinatorial optimization and combinatorial learning
  – modular approach: separate offline optimization with online learning
    • learning part does not need domain knowledge on optimization
Ongoing and Future Work

• Ongoing:
  – Thompson sampling for CMAB
  – Combinatorial pure exploration for nonlinear reward functions

• Possible future directions
  – Many other variants of combinatorial optimizations problems --- as long as it has unknown inputs need to be learned
  – What about adversarial CMAB?
  – More practical and more efficient solutions for particular problems
  – How to generalize CMAB to reinforcement learning tasks?
Acknowledgments to my collaborators

袁洋
Cornell

王亚军
Microsoft

林添
Google

Bruno Abraohao
Stanford

Robert Kleinberg
Cornell

John C.S. Lui
CUHK

陈首元
Microsoft

Irwin King
CUHK

Michael Lyu
CUHK

李建
Tsinghua

王钦石
Princeton

李帅
CUHK

王趵翔
CUHK

张胜誉
CUHK

胡威
Princeton

李孚
UT Austin

刘宇
Tsinghua

陆品燕
SUFE
Questions?

Search Wei Chen Microsoft