Online optimization and learning under long-term convex constraints and objective

Shipra Agrawal

Industrial Engineering and Operations Research
Data Science Institute
Columbia University

Based on joint work with Nikhil R. Devanur.

Outline of the talk

Online stochastic convex programming

Generalization of online stochastic packing/covering

Multi-armed Bandits

with concave rewards and convex knapsacks

Linear contextual bandits

with global convex constraints and objective

Outline of the talk

Online stochastic convex programming

Generalization of online stochastic packing/covering

Multi-armed Bandits

with concave rewards and convex knapsacks

Linear contextual bandits

with global convex constraints and objective

The online allocation problem in display advertising

Advertisers specify target user profiles, delivery goals, budgets

- user opens a page at time t, matches target profile of many ads
- for each ad j, there is a value v_{tj}
- Pick one

(Uncertainty in future user profiles/values/matching of user-ads)

The online allocation problem in display advertising

Advertisers specify target user profiles, delivery goals, budgets

- user opens a page at time t, matches target profile of many ads
- for each ad j, there is a value v_{tj}
- Pick one

(Uncertainty in future user profiles/values/matching of user-ads)

- Maximize the total value of served ads while not exceeding budgets.

Online budgeted matching

At every time t,

- a request arrives matches set A_t of ads.

Online budgeted matching

At every time t,

- a request arrives matches set A_t of ads.
- Observe value v_{tj} of every ad $j \in A_t$.

  Full information. "Before" making the decision.

Online budgeted matching

At every time t,

- a request arrives matches set A_t of ads.
- Observe value v_{tj} of every ad $j \in A_t$.

  Full information. "Before" making the decision.
- Pick an ad j_t from A_t,

  Online decisions: use A_t and history before time t

Online budgeted matching

At every time t,

- a request arrives matches set A_t of ads.
- Observe value v_{tj} of every ad $j \in A_t$.

  Full information. "Before" making the decision.
- Pick an ad j_t from A_t,

  Online decisions: use A_t and history before time t
- Goal: Given budget B_j for advertiser j

$$\begin{aligned} \textit{Maximize} \quad & \sum_j \sum_{t:j=j_t} v_{tj} \\ s.t. \quad & \sum_{t:j=j_t} v_{tj} \leq B_j \quad \forall j \end{aligned}$$

Online packing

[DH 2009, AWY 2009, DCCJS 2010, FHKMS 2010, DJSW 2011, KRTV 2014]

- At every time t, we have a set A_t of options.

Online packing

[DH 2009, AWY 2009, DCCJS 2010, FHKMS 2010, DJSW 2011, KRTV 2014]

- At every time t, we have a set A_t of options.
- Cost/rewards associated with option $j \in A_t$ is given by vector $\mathbf{v}_{tj} = (r_{tj}, \mathbf{c}_{tj})$.

Online packing

[DH 2009, AWY 2009, DCCJS 2010, FHKMS 2010, DJSW 2011, KRTV 2014]

- At every time t, we have a set A_t of options.
- Cost/rewards associated with option $j \in A_t$ is given by vector $\mathbf{v}_{tj} = (r_{tj}, \mathbf{c}_{tj})$.
- Pick an option j_t from A_t, $(r_t^\dagger, \mathbf{c}_t^\dagger) := (r_{tj_t}, \mathbf{c}_{tj_t})$.

  Online decisions: use only history before time t

Online packing

[DH 2009, AWY 2009, DCCJS 2010, FHKMS 2010, DJSW 2011, KRTV 2014]

- At every time t, we have a set A_t of options.
- Cost/rewards associated with option $j \in A_t$ is given by vector $\mathbf{v}_{tj} = (r_{tj}, \mathbf{c}_{tj})$.
- Pick an option j_t from A_t, $(r_t^\dagger, \mathbf{c}_t^\dagger) := (r_{tj_t}, \mathbf{c}_{tj_t})$.

  Online decisions: use only history before time t
- Goal: Given budget vector $\mathbf{B}$,

$$\begin{aligned} \textit{Maximize} \quad & \textstyle\sum_t r_t^\dagger \\ s.t. \quad & \textstyle\sum_t \mathbf{c}_t^\dagger \le \mathbf{B} \end{aligned}$$

Nonlinear constraints and utilities

- Fairness

$$\text{Maximize} \quad \min_j \Big(\sum_{t:j=j_t} 1 \Big)$$

Nonlinear constraints and utilities

- Fairness

$$\text{Maximize} \quad \min_j \Big(\sum_{t: j = j_t} 1 \Big)$$

- Under-delivery penalty. (goal G_j for advertiser j)

$$\text{Minimize} \quad \sum_j \Big(G_j - \sum_{t: j = j_t} 1 \Big)^+$$

Nonlinear constraints and utilities

- Fairness

$$\text{Maximize} \quad \min_j \Big(\sum_{t: j = j_t} 1 \Big)$$

- Under-delivery penalty. (goal G_j for advertiser j)

$$\text{Minimize} \quad \sum_j \Big(G_j - \sum_{t: j = j_t} 1 \Big)^+$$

- Diversity. Let there are m types of users, $0 - 1$ vector w_t gives type of user t.

$$\text{Minimize} \quad \sum_j \Big|\Big| \sum_{t: j = j_t} w_t \Big|\Big|^2$$

Online Stochastic Convex Programming

[A., Devanur 2015]

- At every time t, we have a set A_t of options.

Online Stochastic Convex Programming

[A., Devanur 2015]

- At every time t, we have a set A_t of options.
- Observe vector $\mathbf{v}_{tj} \in [0,1]^d$ associated with every $j \in A_t$:

Online Stochastic Convex Programming

[A., Devanur 2015]

- At every time t, we have a set A_t of options.
- Observe vector $\mathbf{v}_{tj} \in [0,1]^d$ associated with every $j \in A_t$:
- Pick an option j_t from A_t, $\mathbf{v}_t^{\dagger} := \mathbf{v}_{tj_t}$.

Online Stochastic Convex Programming

[A., Devanur 2015]

- At every time t, we have a set A_t of options.
- Observe vector $\mathbf{v}_{tj} \in [0,1]^d$ associated with every $j \in A_t$:
- Pick an option j_t from A_t, $\mathbf{v}_t^\dagger := \mathbf{v}_{tj_t}$.
- Goal: Given concave function f, convex set S

$$\begin{aligned} \textit{Maximize} \quad & f\left(\tfrac{1}{T}\sum_t \mathbf{v}_t^\dagger\right) \\ s.t. \quad & \tfrac{1}{T}\sum_t \mathbf{v}_t^\dagger \in S \end{aligned}$$

Online Stochastic Convex Programming

[A., Devanur 2015]

- At every time t, we have a set A_t of options.
- Observe vector $\mathbf{v}_{tj} \in [0,1]^d$ associated with every $j \in A_t$:
- Pick an option j_t from A_t, $\mathbf{v}_t^\dagger := \mathbf{v}_{tj_t}$.
- Goal: Given concave function f, convex set S

$$\begin{aligned} \mathit{Maximize} \quad & f\left(\tfrac{1}{T} \textstyle\sum_t \mathbf{v}_t^\dagger\right) \\ s.t. \quad & \tfrac{1}{T} \textstyle\sum_t \mathbf{v}_t^\dagger \in S \end{aligned}$$

E.g., Under-delivery penalty: set $\mathbf{v}_{tj} = \mathbf{1}_j$.

$$\frac{1}{T} \|\mathbf{G} - \sum_t \mathbf{v}_t^\dagger)^+\|_1 =: h(\frac{1}{T} \sum_t \mathbf{v}_t^\dagger)$$

for a convex function h.

Other examples

- Objective $\sum_t f_t(\mathbf{u}_t^\dagger)$ or constraint $\sum_t h_t(\mathbf{u}_t^\dagger) \leq B$
  - Use
    $$\mathbf{v}_{tj} := f_t(\mathbf{u}_{tj})$$
  - Objective $\sum_t \mathbf{v}_t^\dagger$, constraint $\sum_t \mathbf{u}_t^\dagger \leq B$

Other examples

- Objective $\sum_t f_t(\mathbf{u}_t^\dagger)$ or constraint $\sum_t h_t(\mathbf{u}_t^\dagger) \leq B$
  - Use
$$\mathbf{v}_{tj} := f_t(\mathbf{u}_{tj})$$
  - Objective $\sum_t \mathbf{v}_t^\dagger$, constraint $\sum_t \mathbf{u}_t^\dagger \leq B$
- $\mathbf{v}_{tj} \in [-1, 1]$
  - Replace
$$\mathbf{v}_{tj} := (\mathbf{v}_{tj} + 1)/2$$
Change f and S accordingly. Remains concave/convex.

Stochastic input models

- Random Permutation (RP)
  - $A_1, A_2, \ldots, A_T$ chosen adversarially, arrive in random order.
- IID
  - A_t at every time t is generated i.i.d. from fixed but *unknown* distribution (over sets of options)

Performance Measures

(Notation) $\mathbf{v}^{\dagger}_{\text{avg}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{v}^{\dagger}_{t}$

Regret (Competitive difference)

- Regret in objective $\text{OPT} - f(\mathbf{v}^{\dagger}_{\text{avg}})$
  - OPT: offline optimal in RP model
  - expected optimal in IID, bounded by best static policy
- Regret in constraints $d(\mathbf{v}^{\dagger}_{\text{avg}}, S)$

Performance Measures

(Notation) $\mathbf{v}^{\dagger}_{\text{avg}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{v}^{\dagger}_{t}$

Regret (Competitive difference)

- Regret in objective $\text{OPT} - f(\mathbf{v}^{\dagger}_{\text{avg}})$
  - OPT: offline optimal in RP model
  - expected optimal in IID, bounded by best static policy
- Regret in constraints $d(\mathbf{v}^{\dagger}_{\text{avg}}, S)$

Performance Measures

(Notation) $\mathbf{v}^{\dagger}_{\text{avg}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{v}^{\dagger}_{t}$

Regret (Competitive difference)

- Regret in objective $\text{OPT} - f(\mathbf{v}^{\dagger}_{\text{avg}})$
  - OPT: offline optimal in RP model
  - expected optimal in IID, bounded by best static policy
- Regret in constraints $d(\mathbf{v}^{\dagger}_{\text{avg}}, S)$

Competitive ratio

Performance Measures

(Notation) $\mathbf{v}^\dagger_{\text{avg}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{v}^\dagger_t$

Regret (Competitive difference)

- Regret in objective $\text{OPT} - f(\mathbf{v}^\dagger_{\text{avg}})$
  - OPT: offline optimal in RP model
  - expected optimal in IID, bounded by best static policy
- Regret in constraints $d(\mathbf{v}^\dagger_{\text{avg}}, S)$

Competitive ratio

- The ratio of OPT to $f(\mathbf{v}^\dagger_{\text{avg}})$

  constraints need to be satisfied at all times
  popular measure for online packing
  too strong for online convex programming

Our results [A., Devanur SODA 2015]

- Fast algorithms with regret of $\tilde{O}\left(\sqrt{\frac{1}{T}}\right)$ for both RP and IID

Our results [A., Devanur SODA 2015]

- Fast algorithms with regret of $\tilde{O}\left(\sqrt{\frac{1}{T}}\right)$ for both RP and IID

$$\begin{aligned} \text{Regret in objective in time } T &= (Z+L)\cdot O\left(\sqrt{\frac{C}{T}}\right) \\ \text{Regret in constraints in time } T &= O\left(\sqrt{\frac{C}{T}}\right) \end{aligned}$$

  - High probability results.
  - f is L-Lipschitz, $C = \log(d)$ for $\|\cdot\|_\infty$, $C = d\log(d)$ for $\|\cdot\|_2$
  - Z is a parameter of problem

Special cases

Online Packing: Competitive ratio of $1 - O(\frac{\log(d)}{\sqrt{B}})$ for both RP and IID

- Matches the upper bound. [A., Wang, Ye 2009]
- Long line of previous work [DH 2009, AWY 2009, DCCJS 2010, FHKMS 2010, DJSW 2011, KRTV 2014]
- Simultaneous to our work [Gupta, Molinaro 2014]

Smooth objective and constraints Even better **logarithmic** regret of $\tilde{O}\left(\frac{\log(T)}{T}\right)$ in IID case

Qualitative contributions

- Online learning as blackbox (to learn dual variables)
- Analysis techniques modularize role of IID vs. RP stochastic model
- Fast algorithm with incremental updates

Overall idea

- Consider no constraints, maximize concave function

$$\text{maximize } f\left(\frac{1}{T}\sum_t \mathbf{v}_t^\dagger\right)$$

Overall idea

- Consider no constraints, maximize concave function

$$\text{maximize } f\left(\frac{1}{T}\sum_t \mathbf{v}_t^\dagger\right)$$

- Main issue: non-separability
  - $\frac{1}{T}\sum_t f_t(\mathbf{v}_t^\dagger)$ is easy
  - Simply, $\mathbf{v}_t^\dagger = \arg\max_{j \in A_t} f_t(\mathbf{v}_{tj})$.

Overall idea

- Consider no constraints, maximize concave function

$$\text{maximize } f\left(\frac{1}{T}\sum_t \mathbf{v}_t^{\dagger}\right)$$

- Main issue: non-separability
  - $\frac{1}{T}\sum_t f_t(\mathbf{v}_t^{\dagger})$ is easy
  - Simply, $\mathbf{v}_t^{\dagger} = \arg\max_{j \in A_t} f_t(\mathbf{v}_{tj})$.
- What is contribution of $\mathbf{v}_t^{\dagger}$ to entire objective?

Using Fenchel duality

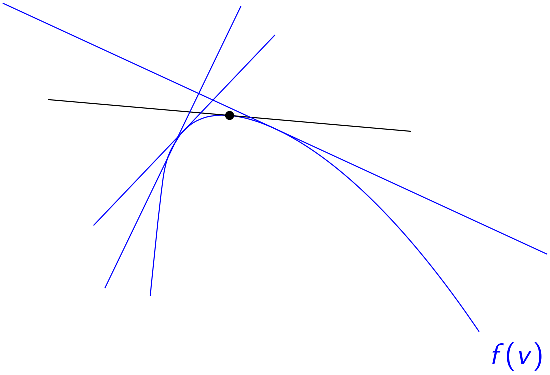
- Fenchel duality: concave function as min of linear functions

$$f(\mathbf{v}) = \min_{\|\boldsymbol{\theta}\|_* \leq L} f^*(\boldsymbol{\theta}) - \boldsymbol{\theta} \cdot \mathbf{v}$$

Using Fenchel duality

- Fenchel duality: concave function as min of linear functions

$$f(\mathbf{v}) = \min_{\|\boldsymbol{\theta}\|_* \leq L} f^*(\boldsymbol{\theta}) - \boldsymbol{\theta} \cdot \mathbf{v}$$



Using Fenchel duality

- $$f\left(\frac{1}{T}\sum_t \mathbf{v}_t^\dagger\right) = f^*(\boldsymbol{\theta}^*) - \frac{1}{T}\sum_t \boldsymbol{\theta}^* \cdot \mathbf{v}_t^\dagger$$

  for some $\boldsymbol{\theta}^*$ *in hindsight*

Using Fenchel duality

- $$f\left(\frac{1}{T}\sum_t \mathbf{v}_t^\dagger\right) = f^*(\boldsymbol{\theta}^*) - \frac{1}{T}\sum_t \boldsymbol{\theta}^* \cdot \mathbf{v}_t^\dagger$$

  for some $\boldsymbol{\theta}^*$ *in hindsight*
- Use $\boldsymbol{\theta}^* \cdot \mathbf{v}_t^\dagger$ as share of $\mathbf{v}_t^\dagger$?

Using Fenchel duality

- $$f(\tfrac{1}{T}\sum_t \mathbf{v}_t^\dagger) = f^*(\boldsymbol{\theta}^*) - \tfrac{1}{T}\sum_t \boldsymbol{\theta}^* \cdot \mathbf{v}_t^\dagger$$

  for some $\boldsymbol{\theta}^*$ *in hindsight*
- Use $\boldsymbol{\theta}^* \cdot \mathbf{v}_t^\dagger$ as share of $\mathbf{v}_t^\dagger$?

Predict dual variable $\boldsymbol{\theta}^*$.

Online Learning or Online Convex Optimization (OCO)

- At time t,
  - pick $\boldsymbol{\theta}_t$,
  - observe convex function $g_t(\cdot)$
  - Loss $g_t(\boldsymbol{\theta}_t)$

Online Learning or Online Convex Optimization (OCO)

- At time t,
  - pick $\boldsymbol{\theta}_t$,
  - observe convex function $g_t(\cdot)$
  - Loss $g_t(\boldsymbol{\theta}_t)$
- Goal: Minimize total loss, compete with any single $\boldsymbol{\theta}$ in hindsight

$$\sum_{t=1}^{T} g_t(\boldsymbol{\theta}_t) \leq \arg\min_{\boldsymbol{\theta}} \sum_{t=1}^{T} g_t(\boldsymbol{\theta}) + R(T)$$

Online Learning or Online Convex Optimization (OCO)

- At time t,
  - pick $\boldsymbol{\theta}_t$,
  - observe convex function $g_t(\cdot)$
  - Loss $g_t(\boldsymbol{\theta}_t)$
- Goal: Minimize total loss, compete with any single $\boldsymbol{\theta}$ in hindsight

$$\sum_{t=1}^{T} g_t(\boldsymbol{\theta}_t) \leq \arg\min_{\boldsymbol{\theta}} \sum_{t=1}^{T} g_t(\boldsymbol{\theta}) + R(T)$$

- Algorithms with $R(T) \leq \tilde{O}(\sqrt{T})$
  - Online gradient descent [Zinkevich 2003], Online mirror descent, multiplicative weight update algorithm [OCO book by Elad Hazan].
  - Fast update of $\boldsymbol{\theta}_t$!

Our algorithm: Online learning to predict Fenchel dual variables

Initialize $\boldsymbol{\theta}_1$.

At time t,

- Primal decision: Pick

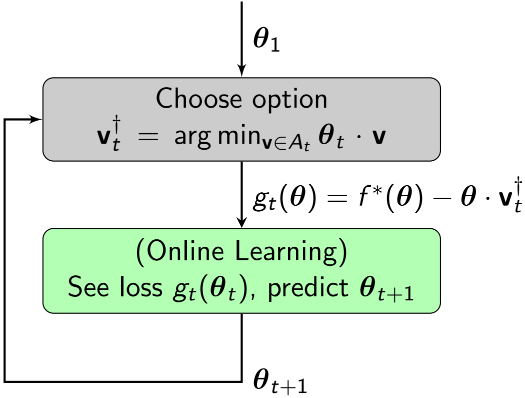
$$\mathbf{v}_t^\dagger = \arg\max_{\mathbf{v}\in A_t} f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbf{v}$$

- Online learning observes loss

$$g_t(\boldsymbol{\theta}_t) = f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbf{v}_t^\dagger$$

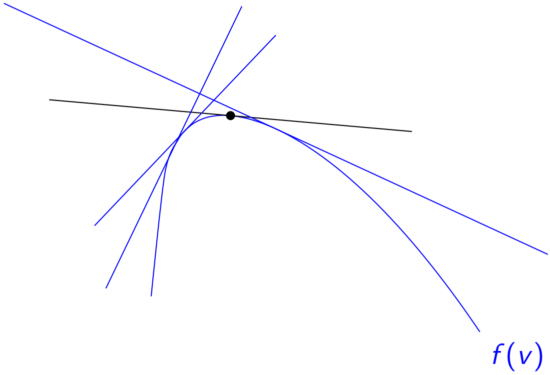
Updates dual variable $\boldsymbol{\theta}_t$ to get $\boldsymbol{\theta}_{t+1}$,

Our algorithm: online learning as blackbox



Analysis: optimism

Fenchel conjugate over-estimates



Algorithm uses optimistic estimates of per-step contribution (useful later for bandit problems)
Online learning controls the over-estimation

Details for IID

- Algorithm maximizes estimated per-step contribution

$$f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbf{v}_t^\dagger \geq f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbf{v}_t^*$$

Details for IID

- Algorithm maximizes estimated per-step contribution

$$f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbf{v}_t^\dagger \geq f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbf{v}_t^*$$

- For IID, you can get optimal in expectation at every step,

$$\mathbb{E}[\mathbf{v}_t^* | H_{t-1}] = \mathbf{v}_{\text{avg}}^*$$

(Not satisfied exactly for RP)

Details for IID

- Algorithm maximizes estimated per-step contribution

$$f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbf{v}_t^\dagger \geq f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbf{v}_t^*$$

- For IID, you can get optimal in expectation at every step,

$$\mathbb{E}[\mathbf{v}_t^* | H_{t-1}] = \mathbf{v}_{\text{avg}}^*$$

(Not satisfied exactly for RP)

- Every step's estimated contribution is at least optimal!

$$f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbb{E}[\mathbf{v}_t^\dagger | \boldsymbol{\theta}_t] \geq f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbf{v}_{\text{avg}}^*$$

Details for IID

- Algorithm maximizes estimated per-step contribution

$$f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbf{v}_t^\dagger \geq f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbf{v}_t^*$$

- For IID, you can get optimal in expectation at every step,

$$\mathbb{E}[\mathbf{v}_t^* | H_{t-1}] = \mathbf{v}_{\text{avg}}^*$$

(Not satisfied exactly for RP)

- Every step's estimated contribution is at least optimal!

$$f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbb{E}[\mathbf{v}_t^\dagger | \boldsymbol{\theta}_t] \geq f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbf{v}_{\text{avg}}^* \geq f(\mathbf{v}_{\text{avg}}^*)$$

Details for IID

- Algorithm maximizes estimated per-step contribution

$$f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbf{v}_t^\dagger \geq f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbf{v}_t^*$$

- For IID, you can get optimal in expectation at every step,

$$\mathbb{E}[\mathbf{v}_t^* | H_{t-1}] = \mathbf{v}_{\text{avg}}^*$$

(Not satisfied exactly for RP)

- Every step's estimated contribution is at least optimal!

$$f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbb{E}[\mathbf{v}_t^\dagger | \boldsymbol{\theta}_t] \geq f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbf{v}_{\text{avg}}^* \geq f(\mathbf{v}_{\text{avg}}^*)$$

LHS over-estimating $f(\frac{1}{T}\sum_t \mathbf{v}_t^\dagger)$ too much?

Details for IID

Remains to bound over-estimation error: use Online Learning regret bounds

- Recall loss function for online learning

$$g_t(\boldsymbol{\theta}) = f^*(\boldsymbol{\theta}) - \boldsymbol{\theta} \cdot \mathbf{v}_t^\dagger$$

Details for IID

Remains to bound over-estimation error: use Online Learning regret bounds

- Recall loss function for online learning

$$g_t(\boldsymbol{\theta}) = f^*(\boldsymbol{\theta}) - \boldsymbol{\theta} \cdot \mathbf{v}_t^\dagger$$

- Over-estimation =

$$\begin{aligned}
& \left(\frac{1}{T} \sum_t f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbf{v}_t^\dagger \right) - f\left(\frac{1}{T} \sum_t \mathbf{v}_t^\dagger\right) \\
= \quad & \frac{1}{T} \sum_t g_t(\boldsymbol{\theta}_t) - \min_{\boldsymbol{\theta}} \frac{1}{T} \sum_t g_t(\boldsymbol{\theta}) \\
\leq \quad & \frac{R(T)}{T} = \tilde{O}\left(\frac{1}{\sqrt{T}}\right)
\end{aligned}$$

Details for IID

Remains to bound over-estimation error: use Online Learning regret bounds

- Recall loss function for online learning

$$g_t(\boldsymbol{\theta}) = f^*(\boldsymbol{\theta}) - \boldsymbol{\theta} \cdot \mathbf{v}_t^\dagger$$

- Over-estimation =

$$\begin{aligned} & \left(\frac{1}{T} \sum_t f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \mathbf{v}_t^\dagger \right) - f\left(\frac{1}{T} \sum_t \mathbf{v}_t^\dagger\right) \\ = \; & \frac{1}{T} \sum_t g_t(\boldsymbol{\theta}_t) - \min_{\boldsymbol{\theta}} \frac{1}{T} \sum_t g_t(\boldsymbol{\theta}) \\ \leq \; & \frac{R(T)}{T} = \tilde{O}\left(\frac{1}{\sqrt{T}}\right) \end{aligned}$$

This bounds the regret in objective!

Analysis summary

- Optimistic Fenchel-dual estimate of algorithm's per-step contribution is at least OPT
- Online learning regret bounds the gap between actual contribution and optimistic estimate

Objective + constraints

- Constraints only problem $f(\frac{1}{T}\sum_t \mathbf{v}_t) = -d(\frac{1}{T}\sum_t \mathbf{v}_t, S)$

Objective + constraints

- Constraints only problem $f(\frac{1}{T}\sum_t \mathbf{v}_t) = -d(\frac{1}{T}\sum_t \mathbf{v}_t, S)$

Combining objectives and constraints

Objective + constraints

- Constraints only problem $f(\frac{1}{T}\sum_t \mathbf{v}_t) = -d(\frac{1}{T}\sum_t \mathbf{v}_t, S)$

Combining objectives and constraints

- Two sets of Fenchel dual variables: $\boldsymbol{\theta}_t$ for distance function, $\boldsymbol{\phi}_t$ for objective function

Objective + constraints

- Constraints only problem $f(\frac{1}{T}\sum_t \mathbf{v}_t) = -d(\frac{1}{T}\sum_t \mathbf{v}_t, S)$

Combining objectives and constraints

- Two sets of Fenchel dual variables: $\boldsymbol{\theta}_t$ for distance function, $\boldsymbol{\phi}_t$ for objective function
- Lagrangian dual variable Z to combine objective and distance

Objective + constraints

- Constraints only problem $f(\frac{1}{T}\sum_t \mathbf{v}_t) = -d(\frac{1}{T}\sum_t \mathbf{v}_t, S)$

Combining objectives and constraints

- Two sets of Fenchel dual variables: $\boldsymbol{\theta}_t$ for distance function, $\boldsymbol{\phi}_t$ for objective function
- Lagrangian dual variable Z to combine objective and distance
- Z needs to be large enough, appears in regret, constant factor approximation suffices

Objective + constraints

- Constraints only problem $f(\frac{1}{T}\sum_t \mathbf{v}_t) = -d(\frac{1}{T}\sum_t \mathbf{v}_t, S)$

Combining objectives and constraints

- Two sets of Fenchel dual variables: $\boldsymbol{\theta}_t$ for distance function, $\boldsymbol{\phi}_t$ for objective function
- Lagrangian dual variable Z to combine objective and distance
- Z needs to be large enough, appears in regret, constant factor approximation suffices
- Sample average approximation every doubling epoch

Outline of the talk

Online stochastic convex programming

Generalization of online stochastic packing/covering

Multi-armed Bandits

with concave rewards and convex knapsacks

Linear contextual bandits

with global convex constraints and objective

Bandit Model: Pay-per-click advertising

Advertiser pays only if the user clicks on the ad

- user opens a page, matches target profile of many ads
- pick ad j
- observe if user clicks or not: value $v_{tj} = b_j$ if the user clicks

(Uncertainty in future user profiles, and user click behavior)

Bandit Model: Pay-per-click advertising

Advertiser pays only if the user clicks on the ad

- user opens a page, matches target profile of many ads
- pick ad j
- observe if user clicks or not: value $v_{tj} = b_j$ if the user clicks

(Uncertainty in future user profiles, and user click behavior)

- Click behavior can be observed only on *after* picking the ad
- Bandit feedback, Exploration-exploitation tradeoff

Online decisions with bandit feedback

We study a framework combining the

multi-armed bandit problem	with	global convex constraints and objective

Combining MAB with online convex programming [A., Devanur EC 2014]

- There are N arms, pick one arm to pull at every time step
- Observe the value vector $\mathbf{v}_t$ **for the pulled arm only**, generated i.i.d.
  (Show an ad, observe click,conversion)

Combining MAB with online convex programming [A., Devanur EC 2014]

- There are N arms, pick one arm to pull at every time step
- Observe the value vector $\mathbf{v}_t$ **for the pulled arm only**, generated i.i.d.
  (Show an ad, observe click,conversion)
- Overall goal:

$$\text{maximize } f\left(\frac{1}{T}\sum_t \mathbf{v}_t\right) \quad \text{s.t.} \quad \frac{1}{T}\sum_t \mathbf{v}_t \in S.$$

Combining MAB with online convex programming [A., Devanur EC 2014]

- There are N arms, pick one arm to pull at every time step
- Observe the value vector $\mathbf{v}_t$ **for the pulled arm only**, generated i.i.d.
  (Show an ad, observe click,conversion)
- Overall goal:

$$\text{maximize } f\left(\frac{1}{T}\sum_t \mathbf{v}_t\right) \quad \text{s.t.} \quad \frac{1}{T}\sum_t \mathbf{v}_t \in S.$$

- Regret in objective and constraints
  - (average) Regret in objective value $\text{OPT} - f(\mathbf{v}^{\dagger}_{\text{avg}})$
  - (average) Regret in constraints $d(\mathbf{v}^{\dagger}_{\text{avg}}, S)$

Our algorithm: simple extension

Optimism under uncertainty

- Same algorithm, but work with high confidence estimates $\tilde{\mathbf{v}}_{t1}, \ldots, \tilde{\mathbf{v}}_{tN}$

$$\tilde{\mathbf{v}}_{jt} = \arg \min_{\mathbf{v} \in \text{confidence interval}_j} \boldsymbol{\theta}_t \cdot \mathbf{v}$$

- $f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \tilde{\mathbf{v}}_{tj}$ is UCB estimate of per-step contribution

Our algorithm: simple extension

Initialize $\boldsymbol{\theta}_1$. At time t,

- Primal algorithm picks

$$j_t := \arg\max_{j \in A_t} f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \tilde{\mathbf{v}}_{tj}$$

- Observe $\mathbf{v}_{tj_t}$, update UCB estimate for j_t.
- Observe online learning loss

$$g_t(\boldsymbol{\theta}_t) = f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \tilde{\mathbf{v}}_{tj}$$

Update dual variables to get $\boldsymbol{\theta}_{t+1}$,

Our Contributions [A., Devanur EC 2014]

Over-estimation by Fenchel dual fits perfectly with optimistic UCB estimates

- Provably optimal performance
  - regret goes down as $T^{-1/2}$

Our Contributions [A., Devanur EC 2014]

Over-estimation by Fenchel dual fits perfectly with optimistic UCB estimates

- Provably optimal performance
  - regret goes down as $T^{-1/2}$
- Known lower bound of $T^{-1/2}$ on regret for the classic multi-armed bandit problem

Our Contributions [A., Devanur EC 2014]

Over-estimation by Fenchel dual fits perfectly with optimistic UCB estimates

- Provably optimal performance
  - regret goes down as $T^{-1/2}$
- Known lower bound of $T^{-1/2}$ on regret for the classic multi-armed bandit problem
- Matches regret lower bound of $\tilde{O}(\frac{\mathrm{OPT}}{\sqrt{B}})$ for bandits with knapsack constraints.
  - Simplifies earlier work on bandits with knapsacks [Badanidiyuru, Kleinberg, Slivkins 2013] and extends to nonlinear

Outline of the talk

Online stochastic convex programming

Generalization of online stochastic packing/covering

Multi-armed Bandits

with concave rewards and convex knapsacks

Linear contextual bandits

with global convex constraints and objective

Linear Contextual bandits: Pay-per click advertising

Advertisers specify target user profiles, payment per click

- user opens a page at time t, matches target profile of many ads
- pick one ad
- "if the user clicks" on the shown ad, publisher gets paid

Uncertainty in future user profiles, uncertainty in clicks

"Click-through rate" depends on a combination of user profile and ad features.

Linear regression Model

Click-through rates as a linear function of user and ad features.

- Let $x_{t,j}$ be a vector of features of (user t, ad j) combination
- chances of getting clicked is $v_{tj} = w^T x_{t,j}$ for some unknown vector w.

Linear contextual bandit problem: explore-exploit in the feature space to learn w quickly, even when number of ad user combinations are large.

Linear contextual bandits with global convex constraints and objective

In every round t, pick one of the many options (arms) in set A_t.

- For every $j \in A_t$, observe "context vector" $x_{t,j} \in \mathbb{R}^d$ before making the choice.
- On pulling arm j, observe vector $\mathbf{v}_t \in [0,1]^m$

Stochastic assumptions:

- Given that arm j is pulled, vector $\mathbf{v}_t$ is i.i.d. from distribution with mean $W^T x_{tj}$, matrix W is unknown.
- Set A_t of context vectors is generated i.i.d. from some *unknown* distribution over collection of context vectors

Our algorithm: simple extension

- Same algorithm, but work with LinUCB estimates $\tilde{W}_t^T x_{tj}$ for every j

Initialize $\boldsymbol{\theta}_1$. At time t,

- Primal algorithm picks

$$j_t := \arg\max_{j \in A_t} f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \tilde{W}_t^T x_{tj}$$

- Observe $\mathbf{v}_t = W^T x_{t,j_t} + noise$, update UCB estimate for W.
- Observe online learning loss

$$g_t(\boldsymbol{\theta}_t) = f^*(\boldsymbol{\theta}_t) - \boldsymbol{\theta}_t \cdot \tilde{W}_t^T x_{tj}$$

Update dual variables to get $\boldsymbol{\theta}_{t+1}$,

Our results

- $\tilde{O}(d\sqrt{T})$ regret for only constraints or only objective
- Tricky to estimate Z even for knapsack problem due to context uncertainty
- $\tilde{O}(d\frac{\text{OPT}}{B}\sqrt{T})$ regret bounds for linear contextual bandits with knapsack constraints when $B \geq dT^{3/4}$.
- Important: no dependence on number of arms (possible user+ad types, which is exponential in d)

Conclusion

Sequential decision making: Online learning as black-box

- Fast algorithm
- Modular techniques that work for RP and IID, linear and convex, full information and bandit
- Any progress in learning gets translated, e.g., smooth functions
- First formal connection, conjectured since [Mehta et al. 2007]