Algorithms for Online Optimization Problems

Adam Kalai
Santosh Vempala

IMA
May 2003
(Static) Online Optimization
Example

Online shortest paths [TW02]:

For any sequence of bounded times,

\[ E[\text{time}] \leq (1 + \epsilon) \times (\text{best in hindsight}) + \frac{c}{\epsilon} \]
Summary of results

Similar problems
Splay trees, list update [ST85]
Adaptive Huffman coding [F73,G78,K85]
Predicting from experts [LW87,...]
Online decision tree pruning [HS95]
Online shortest paths [TW02]
Online linear optimization [KV02]
...

Simple algorithm, single analysis gives: [KV02]

\[ E[\text{cost}] \leq (1 + \epsilon) \times (\text{best in hindsight}) + \frac{c}{\epsilon} \]

Efficient alg, \( \epsilon \) probability of recomputing.

Previous work: not (1 + \( \epsilon \))-optimal, or inefficient, or problem specific.
Online Linear Optimization

\[
\max_{\bar{x} \in S} \bar{x} \cdot \bar{c}
\]

state $\bar{x} = (\# \text{ chairs, } \# \text{ tables})$

objective $\bar{c} = (\text{chair profit, table profit})$

<table>
<thead>
<tr>
<th>Day</th>
<th>(chair profit, table profit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>($40,$60)</td>
</tr>
<tr>
<td>2</td>
<td>($30,$70)</td>
</tr>
<tr>
<td>3</td>
<td>($35,$66)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Online Optimization Model

- $S = \text{feasible state set}$

- $\vec{c}_1, \vec{c}_2, \ldots$, daily objective vectors, $|\vec{c}_j|_1 \leq 1$

- On day $j$, we choose state $\vec{x}_j \in S$ knowing only $S$ and $\vec{c}_1, \vec{c}_2, \ldots, \vec{c}_{j-1}$

- Get score $= \vec{x}_j \cdot \vec{c}_j$

- Maximize total score $= \vec{x}_1 \cdot \vec{c}_1 + \cdots + \vec{x}_t \cdot \vec{c}_t$
Static Offline/Online Problems

$\text{best state score} = \max_{\tilde{x} \in S} \tilde{x} \cdot (\tilde{c}_1 + \tilde{c}_2 + \cdots + \tilde{c}_t)$

$t = \# \text{ days}, \ n = \# \text{ dimensions}$

$d = L_1 \text{ diameter of feasible state set } S$

our score $\geq$ best state score $- \epsilon t - \frac{d^2}{\epsilon}$

For non-negative scores $\tilde{x} \cdot \tilde{c}_j \geq 0, \forall \tilde{x} \in S$,

our score $\geq (1 - \epsilon)(\text{best state score}) - \frac{d \log n}{\epsilon}$
Follow the Leader

\[
\begin{array}{cccc}
\tilde{c}_j & \text{best so far} & \text{Follow's} & \text{best score} \\
(0.5,0) & (1,0) & 0? & 0.5 \\
(0,1) & (0,1) & 0 & 1.0 \\
(1,0) & (1,0) & 0 & 1.5 \\
(0,1) & (0,1) & 0 & 2.0 \\
(1,0) & (1,0) & 0 & 2.5 \\
\vdots & \vdots & \vdots & \vdots \\
\end{array}
\]
Maximization oracle

- Oracle $M : \mathbb{R}^n \rightarrow \mathbb{R}^n$

- $M(\vec{c})$ is any $\vec{x} \in S$ that maximizes $= \vec{x} \cdot \vec{c}$

- best state score is

$$M(\vec{c}_1 + \cdots + \vec{c}_t) \cdot (\vec{c}_1 + \cdots + \vec{c}_t)$$
Be the Leader

On day \( j \), Follow uses \( M(\bar{c}_1 + \cdots + \bar{c}_{j-1}) \).

On day \( j \), **Be the Leader** uses \( M(\bar{c}_1 + \cdots + \bar{c}_j) \).

Be the leader’s score \( \geq \) best score in hindsight

Proof: Best score in hindsight =

\[
M(\bar{c}_1 + \cdots + \bar{c}_t) \cdot (\bar{c}_1 + \cdots + \bar{c}_t) \leq \\
M(\bar{c}_1 + \cdots + \bar{c}_{t-1}) \cdot (\bar{c}_1 + \cdots + \bar{c}_{t-1}) + \\
M(\bar{c}_1 + \cdots + \bar{c}_t) \cdot \bar{c}_t \leq \\
\vdots \\
= \text{Be the leader’s score}
\]
Follow the Leader

On day $j$ use $M(\tilde{c}_1 + \cdots + \tilde{c}_{j-1} + \tilde{r})$, where $\tilde{r}$ is random from cube of side $L$.

Similar to Hannan’s 1951 idea in game theory.
Be the Leader

On day $j$ use $M(\hat{c}_1 + \cdots + \hat{c}_j + \vec{r})$, where $\vec{r}$ is random from cube of side $L$.

Intuition: added randomness makes

\[
\text{follow the leader} \approx \text{be the leader}
\]
Nonoverlap

\[ Pr(\text{nonoverlap}) \leq \frac{|\vec{c}|_1}{L} \leq \frac{1}{L} \]

\[ E[M(\vec{c}_1 + \cdots + \vec{c}_j + \vec{r}) \cdot \vec{c}_j - M(\vec{c}_1 + \cdots + \vec{c}_{j-1} + \vec{r}) \cdot \vec{c}_j] \]

\[ \leq (1 - \frac{1}{L})0 + \frac{1}{L}(\vec{v} \cdot \vec{c}_j - \vec{w} \cdot \vec{c}_j) \leq \frac{d}{L} \]
Follow’s Analysis

\[ \tilde{\text{follow leader’s score}} \geq \tilde{\text{be leader’s score}} - \frac{d}{L} t \]

(from previous slide)

\[ \tilde{\text{be leader’s score}} \geq \text{best score in hindsight} - dL \]

Proof: With pretend initial perturbation \( \tilde{c}_0 = r \), \( \tilde{\text{be}} \) the leader is as good as \( \tilde{\text{be}} \) in hindsight. Difference is at most \( (\tilde{v} - \tilde{w}) \cdot \tilde{r}' \leq dL \).

With \( L = \frac{d}{\epsilon} \),

\[ \tilde{\text{follow’s score}} \geq \text{best score} - \frac{d}{L} t - dL \]

\[ = \text{best score} - \epsilon t - \frac{d^2}{\epsilon} \]
Lazy algorithm

$E[\text{lazy’s score}] = E[\text{follow’s score}]$

Lazy recomputes with probability $\leq 1/L = \epsilon/d$

Cost of switching is no problem.
High probability bounds

1. Choose \( \vec{r}_1, \ldots, \vec{r}_k \in \mathbb{R}^n \) from \( \text{cube}_n(d/\epsilon) \)
2. On day \( j \) use average

\[
\left( \frac{1}{k} \sum_{i=1}^{k} M(\vec{r}_i + \vec{c}_1 + \cdots + \vec{c}_{j-1}) \right) \in S
\]

For \( k \geq \frac{d^2}{\epsilon^2 \delta t} \), with prob. at least \( 1 - \delta \),
our score \( \geq \) best state score \(- 2\epsilon t - \frac{d^2}{\epsilon} \)
\((1 - \epsilon)\) algorithm

1. Choose \(\vec{r}'\) with density \(\propto (1 - \epsilon)|\vec{r}'|_1\)

2. On day \(j\), use state \(M(\vec{c}_1 + \cdots + \vec{c}_{j-1} + \vec{c}_r)\)

If \(\vec{c}_j \cdot \vec{x} \geq 0\) \(\forall \vec{x} \in S, j \geq 1,)\)

our score \(\geq (1 - \epsilon)(\text{best state score}) - \frac{d \log n}{\epsilon}\)
Shortest path application

- A dimension for each edge

- Objective vector = time on each edge

- Path = 0-1 vector with 1’s on path edges

- Feasible set = set of paths

- Optimization oracle M = shortest path

- Lazy algorithm recomputes only with probability $\epsilon/d$, and finds the shortest path given totals on all edges + randomness.
Conclusions

- Optimization $\Rightarrow (1+\epsilon)$-online optimization

- Extensions
  1. High probability bounds
  2. Lazy algorithm
  3. Partial information (bandits version)
  4. Tracking (dynamic bounds)

- Approximate optimization $\Rightarrow$ approximate online optimization?

- Online nonlinear optimization? [Z03]
Online Binary Search Trees

![Binary Tree Diagram](image)

<table>
<thead>
<tr>
<th>Access</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

“splaying” cost $\leq 4.75$ (best tree cost) $+ O(\log n)$
[Sleator, Tarjan 85]
Air Conditioned Trees

1. Initially, for $1 \leq i \leq n$, choose $a_i \in \{0, 1, \ldots, N\}$

2. After access to element $i$,
   if (number of accesses to $i$) $\geq a_i + N$ then
   
   (a) $a_i \leftarrow a_i + N$

   (b) Change trees to $M(a_1, a_2, \ldots, a_n)$

   $N = n^2/\epsilon$ ($L_1$ diameter of trees $\leq n^2$)

   $Pr$(changing trees) $\leq \epsilon/n^2$

   Best tree $M(\cdot)$ can be computed in time $O(n^2)$

   $E[our \ cost] \leq (best \ tree \ cost) + \epsilon t + \frac{n^4}{\epsilon}$

   $\leq (1 + \epsilon)(best \ tree \ cost) + \frac{n^4}{\epsilon}$
Online $k$-median problem