Academia Sinica
Study Group

Online Algorithm

Presented by **Ting-An Chen**

Advisor: De-Nian Yang, Ming-Syan Chen

Jul 2, 2020

Outline

- Introduction - online algorithm
- Online v.s. Offline algorithm
- Online algorithm performance – competitiveness analysis
- Case study
  - Case 1. Ski Rental Problem
  - Case 2. Deterministic Paging Problem

Online algorithm

- Sequence of data
- Limited memory
- A sketch of data   Summary
- Return output at each time stamp
- Never know the nature of the coming data, but expected!
- OPT is unknown
- ➔ v.s. Offline OPT (known)   *How to compare?*

Online algorithm (A) v.s. Offline OPT (OPT)

- Competitiveness analysis
  - $\frac{\text{Cost(A)}}{\text{Cost(OPT)}} \leq$ bound, then A is *competitive*.
  - $[Def.]\ \alpha - competitive\ online\ algorithm.$
  - σ: $an\ input\ sequence$
  - c: $a\ cost\ function$
  - ➔
  - $A\ is\ said\ to\ be\ \alpha - competitive\ \ if\ c_A(\sigma) \leq \alpha \cdot c_{OPT}\ (\sigma).$
  - α: $competitive\ ratio.$

Case 1. Ski-rental problem

- Ski everyday
- Rent or buy the skiing equipment (daily decision)
  - Rent one day, \$1.
  - Buy, \$C.
- Assumption: might get hurt each day then cannot ski.
- Let d be the total number of days skiing.
- Algorithm: “Rent for C days, then buy on (C+1)-th day.”
  - $[pf.]$ $2-competitive$ $online\ algorithm,$ $\mathrm{i.e.,}$ $c_A(\sigma) \leq 2 \cdot c_{OPT}(\sigma).$

case	$c_A(\sigma)$	$c_{OPT}(\sigma)$
If d ≤ C	**d**	**d**
If d > C	**2C**	**C**

Online algorithm (A) v.s. Offline OPT (OPT)

- Competitiveness analysis
  - $\frac{\text{Cost(A)}}{\text{Cost(OPT)}} \leq$ bound, then A is *competitive*.

    Approximation ratio?

  - $[Def.]\ \alpha - competitive\ online\ algorithm.$
  - $\sigma: an\ input\ sequence$
  - $c: a\ cost\ function$
  - ➔
  - $A\ is\ said\ to\ be\ \alpha - competitive\ \ if\ c_A(\sigma) \leq \alpha \cdot c_{OPT}\ (\sigma).$
  - $\alpha: competitive\ ratio.$

Online v.s. Offline (traditional) algorithm

		Online	Offline
	Compare to Offline OPT	Competitive ratio	Approximation ratio
events	Cost(A) related to… 1. **Inputs**	a. Unknown but expected b. Random w/o known patterns ➔ Online alg. Cost(A): ➔ fluctuate	a. Known b. Not random –OR- Random w/. Distribution ➔ Offline alg. Cost(A): ➔ stable
strategy	Cost(A) related to… 2. **Algorithm**	At different states… Same strategy - Deterministic Different strategies – Random	Single strategy
	Inputs to the **Alg.**	Hard –OR- easy ➔ average case	Hard ➔ worst case

Studying worst case in Online algorithm?

- Adversary!!

To consider the case: the inputs make the algorithm worst

Online algorithm – random v.s. adversarial inputs

	Online (**random** inputs)	Online (**adversarial** inputs)	Offline
Compare to Offline OPT	Competitive ratio ≤	Competitive ratio	Approximation ratio
Cost(A) related to… 1. Inputs	a. Unknown but expected b. Random w/o known patterns ➔ Online alg. Cost(A): ➔ fluctuate	a. Known (simulated) b. Not random ➔ Online alg. Cost(A)': ➔ Stable ➔ Online alg. Cost(A) ≤ Cost(A)'	a. Known b. Not random –OR- Random w/. Distribution ➔ Offline alg. Cost(A): ➔ stable
Inputs to the Alg.	Hard –OR- easy ➔ average case	Hard ➔ worst case	Hard ➔ worst case

Case 2. Paging problem

- Hard disk – large memory, slow access
- Cache – small memory, fast access
- A sequence of page requests (from cache)
- *Page fault* if requested info. is not in cache
- ➔ access from hard disk
- ➔ large access costs
- Problem:
  - what data is to be stored in cache s.t. fewest page faults
  - more precisely, which data in cache is to be evicted when a new data is requested

最不近(最久之前) request 的，先 evict 拔除

Paging problem – Least Recently Used (LRU)

- Example

request|	cache elements|	page fault|	evicted item|
a	-,-,-	True	-
b	a,-,-	True	-
c	a,b,-	True	-
d	a,b,c	True	a
a	d,b,c	True	b
e	d,a,c	True	c
b	d,a,e	True	d
a	b,a,e	False	
c	b,a,e	True	e
e	b,a,c	True	b

__Deterministic Online Algorithm:__
With specific strategy

Paging problem

- Claim: If A is a deterministic online algorithm that is $\alpha - competitive$, then $\alpha \geq k$, where k is the cache size. (at most k pages in cache), and total (k+1) distinct pages.

Summary

- Introduction - online algorithm
- Online v.s. Offline algorithm
- Online algorithm performance – competitiveness analysis
- Case study
  - Case 1. Ski Rental Problem
  - Case 2. Deterministic Paging Problem
- Adversary – worst case of online algorithm

Randomized Online Algorithm