Online Caching with Optimistic Learning

Naram Mhaisen[1] , George Iosifidis[1] , Douglas Leith[2]

[1]Department of Software Technology, EEMCS, TU Delft, Netherlands.

[2]School of Computer Science and Statistics, Trinity College Dublin, Ireland.





Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Table of Contents

1. Online Caching: System Model & Problem Formulation
2. Optimistic Learning via Recommendations
3. Optimistic Learning with a Stable Predictor
4. Optimistic Learning with Constraints
5. Summary & Outlook

Table of Contents

1 Online Caching: System Model & Problem Formulation

2 Optimistic Learning via Recommendations

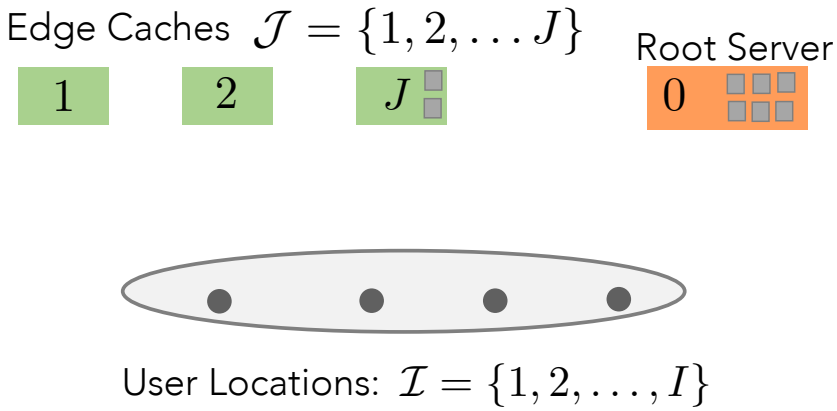
3 Optimistic Learning with a Stable Predictor

4 Optimistic Learning with Constraints

5 Summary & Outlook

Online Caching: System Model

Bipartite edge caching model[1]



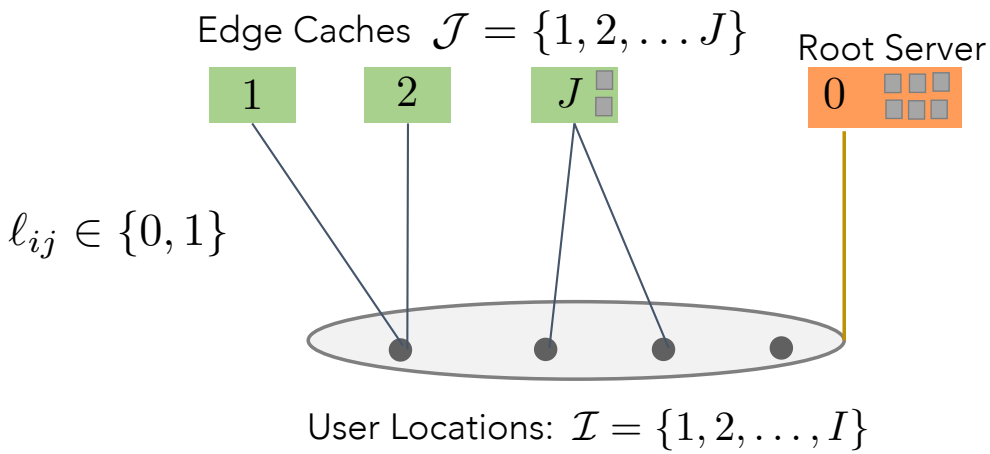
for $t = 1, 2, \dots, T$ do:

- A request $q_t \in \{0, 1\}^{N \times I}$ arrives.
- Serve according to the current policy $x_t = (y_t, z_t)$:
  - $y_t = (y_{nj} \in [0, 1])$: Proportion of file n stored at cache j.
  - $z_t = (z_{nij} \in [0, 1])$: Proportion of file n routed to user i from cache j.
- Get utility $f_t(x_t) = \sum_{n,i,j} w_{nij} q_{ni}^t z_{nij}^t$.
- Compute x_{t+1}

[1] K. Shanmugam *et. al.* "FemtoCaching: Wireless Content Delivery Through Distributed Caching Helpers" in IEEE Trans. on Inf. Theory, 2013.

Online Caching: System Model

Bipartite edge caching model[1]



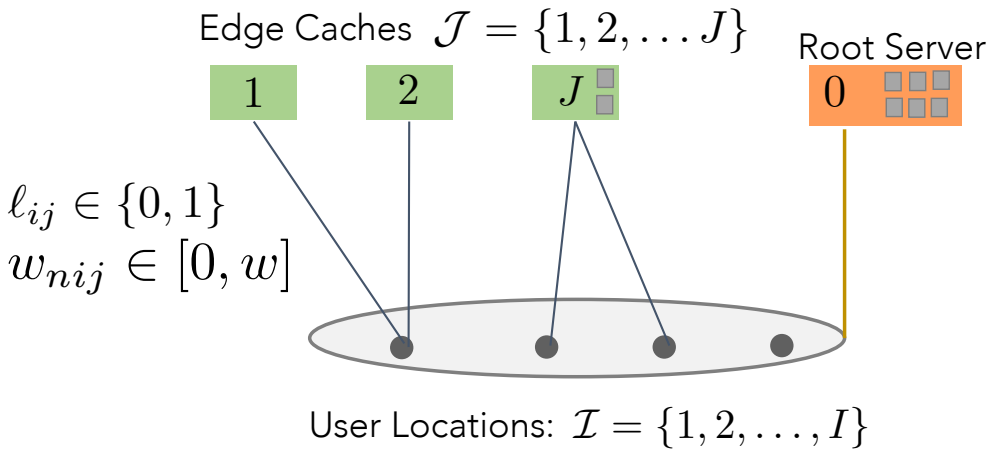
for $t = 1, 2, \dots, T$ do:

- A request $q_t \in \{0, 1\}^{N \times I}$ arrives.
- Serve according to the current policy $x_t = (y_t, z_t)$:
  - $y_t = (y_{nj} \in [0, 1])$: Proportion of file n stored at cache j.
  - $z_t = (z_{nij} \in [0, 1])$: Proportion of file n routed to user i from cache j.
- Get utility $f_t(x_t) = \sum_{n,i,j} w_{nij} q_{ni}^t z_{nij}^t$.
- Compute x_{t+1}

[1] K. Shanmugam *et. al.* "FemtoCaching: Wireless Content Delivery Through Distributed Caching Helpers" in IEEE Trans. on Inf. Theory, 2013.

Online Caching: System Model

Bipartite edge caching model[1]



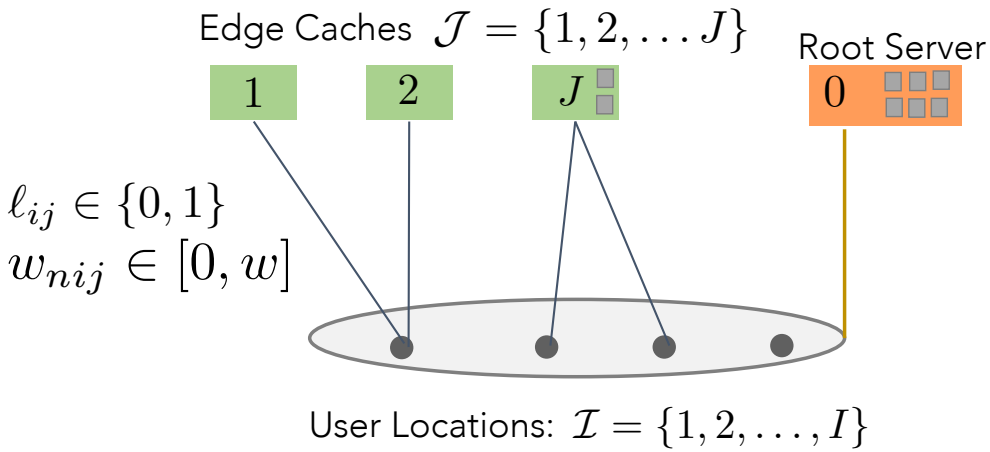
for $t = 1, 2, \dots, T$ do:

- A request $q_t \in \{0, 1\}^{N \times I}$ arrives.
- Serve according to the current policy $x_t = (y_t, z_t)$:
  - $y_t = (y_{nj} \in [0, 1])$: Proportion of file n stored at cache j.
  - $z_t = (z_{nij} \in [0, 1])$: Proportion of file n routed to user i from cache j.
- Get utility $f_t(x_t) = \sum_{n,i,j} w_{nij} q_{ni}^t z_{nij}^t$.
- Compute x_{t+1}

[1] K. Shanmugam *et. al.* "FemtoCaching: Wireless Content Delivery Through Distributed Caching Helpers" in IEEE Trans. on Inf. Theory, 2013.

Online Caching: System Model

Bipartite edge caching model[1]



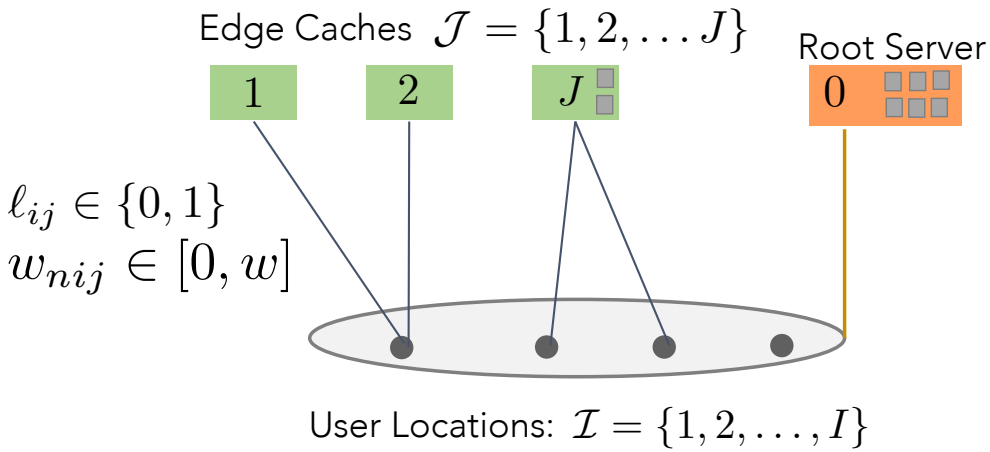
for $t = 1, 2, \dots, T$ do:

- A request $q_t \in \{0, 1\}^{N \times I}$ arrives.
- Serve according to the current policy $x_t = (y_t, z_t)$:
  - $y_t = (y_{nj} \in [0, 1])$: Proportion of file n stored at cache j.
  - $z_t = (z_{nij} \in [0, 1])$: Proportion of file n routed to user i from cache j.
- Get utility $f_t(x_t) = \sum_{n,i,j} w_{nij} q_{ni}^t z_{nij}^t$.
- Compute x_{t+1}

[1] K. Shanmugam *et. al.* "FemtoCaching: Wireless Content Delivery Through Distributed Caching Helpers" in IEEE Trans. on Inf. Theory, 2013.

Online Caching: System Model

Bipartite edge caching model[1]



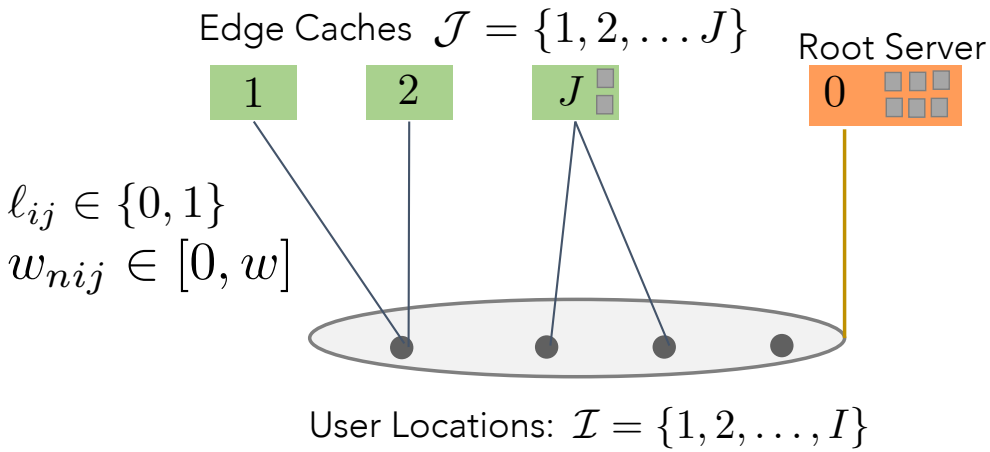
for $t = 1, 2, \dots, T$ do:

- A request $q_t \in \{0, 1\}^{N \times I}$ arrives.
- Serve according to the current policy $x_t = (y_t, z_t)$:
  - $y_t = (y_{nj} \in [0, 1])$: Proportion of file n stored at cache j.
  - $z_t = (z_{nij} \in [0, 1])$: Proportion of file n routed to user i from cache j.
- Get utility $f_t(x_t) = \sum_{n,i,j} w_{nij} q_{ni}^t z_{nij}^t$.
- Compute x_{t+1}

[1] K. Shanmugam *et. al.* "FemtoCaching: Wireless Content Delivery Through Distributed Caching Helpers" in IEEE Trans. on Inf. Theory, 2013.

Online Caching: System Model

Bipartite edge caching model[1]



for $t = 1, 2, \dots, T$ do:

- A request $q_t \in \{0, 1\}^{N \times I}$ arrives.
- Serve according to the current policy $x_t = (y_t, z_t)$:
  - $y_t = (y_{nj} \in [0, 1])$: Proportion of file n stored at cache j.
  - $z_t = (z_{nij} \in [0, 1])$: Proportion of file n routed to user i from cache j.
- Get utility $f_t(x_t) = \sum_{n,i,j} w_{nij} q_{ni}^t z_{nij}^t$.
- Compute x_{t+1}

[1] K. Shanmugam *et. al.* "FemtoCaching: Wireless Content Delivery Through Distributed Caching Helpers" in IEEE Trans. on Inf. Theory, 2013.

Online Caching: Performance Metric

- $\mathcal{X} = \left\{ \{\mathcal{Y} \times \mathcal{Z}\} \cap \{z_{nij} \leq y_{nj}\ell_{ij}, \quad \forall i,j,n\} \right\}$
  - $\mathcal{Y} = \left\{ \sum_{n\in\mathcal{N}} y_{nj} \leq C_j,\ j \in \mathcal{J} \right\}$. (Respect capacity of each cache)
  - $\mathcal{Z} = \left\{ \sum_{j\in\mathcal{J}} z_{nij} \leq 1,\ n \in \mathcal{N}, i \in \mathcal{I} \right\}$. (No redundant parts)
- If *file requests* and *links' quality* were known in advance, we compute

$$x^\star = \arg\max_{x\in\mathcal{X}} \sum_{t=1}^{T} f_t(x), \qquad \text{and select } x^\star,\ \forall t.$$

- Given $\{x_t\}_T = \{x_0, x_1, \ldots, x_t, \ldots, x_T\}$ according to some online alg.

Definition (Regret Metric)

$$R_T = \sum_{t=1}^{T} \left(f_t(x^\star) - f_t(x_t) \right)$$

Online Caching: Performance Metric

- $\mathcal{X} = \left\{ \{\mathcal{Y} \times \mathcal{Z}\} \cap \{z_{nij} \leq y_{nj}\ell_{ij}, \quad \forall i,j,n\} \right\}$
  - $\mathcal{Y} = \left\{ \sum_{n\in\mathcal{N}} y_{nj} \leq C_j,\ j \in \mathcal{J} \right\}$. (Respect capacity of each cache)
  - $\mathcal{Z} = \left\{ \sum_{j\in\mathcal{J}} z_{nij} \leq 1,\ n \in \mathcal{N}, i \in \mathcal{I} \right\}$. (No redundant parts)
- If *file requests* and *links' quality* were known in advance, we compute

$$x^\star = \arg\max_{x\in\mathcal{X}} \sum_{t=1}^{T} f_t(x), \qquad \text{and select } x^\star,\ \forall t.$$

- Given $\{x_t\}_T = \{x_0, x_1, \ldots, x_t, \ldots, x_T\}$ according to some online alg.

Definition (Regret Metric)

$$R_T = \sum_{t=1}^{T} \left(f_t(x^\star) - f_t(x_t) \right)$$

Online Caching: Performance Metric

- $\mathcal{X} = \left\{ \{\mathcal{Y} \times \mathcal{Z}\} \cap \{z_{nij} \leq y_{nj}\ell_{ij}, \quad \forall i,j,n\} \right\}$
  - $\mathcal{Y} = \left\{ \sum_{n\in\mathcal{N}} y_{nj} \leq C_j,\ j \in \mathcal{J} \right\}$. (Respect capacity of each cache)
  - $\mathcal{Z} = \left\{ \sum_{j\in\mathcal{J}} z_{nij} \leq 1,\ n \in \mathcal{N}, i \in \mathcal{I} \right\}$. (No redundant parts)
- If *file requests* and *links' quality* were known in advance, we compute

$$x^\star = \arg\max_{x\in\mathcal{X}} \sum_{t=1}^{T} f_t(x), \qquad \text{and select } x^\star,\ \forall t.$$

- Given $\{x_t\}_T = \{x_0, x_1, \ldots, x_t, \ldots, x_T\}$ according to some online alg.

Definition (Regret Metric)

$$R_T = \sum_{t=1}^{T} \left(f_t(x^\star) - f_t(x_t) \right)$$

Online Caching: Performance Metric

- $\mathcal{X} = \left\{ \{\mathcal{Y} \times \mathcal{Z}\} \cap \{z_{nij} \leq y_{nj}\ell_{ij}, \quad \forall i,j,n\} \right\}$
  - $\mathcal{Y} = \left\{ \sum_{n\in\mathcal{N}} y_{nj} \leq C_j,\ j \in \mathcal{J} \right\}$. (Respect capacity of each cache)
  - $\mathcal{Z} = \left\{ \sum_{j\in\mathcal{J}} z_{nij} \leq 1,\ n \in \mathcal{N}, i \in \mathcal{I} \right\}$. (No redundant parts)
- If *file requests* and *links' quality* were known in advance, we compute

$$x^\star = \arg\max_{x\in\mathcal{X}} \sum_{t=1}^{T} f_t(x), \qquad \text{and select } x^\star,\ \forall t.$$

- Given $\{x_t\}_T = \{x_0, x_1, \ldots, x_t, \ldots, x_T\}$ according to some online alg.

Definition (Regret Metric)

$$R_T = \sum_{t=1}^{T} \left(f_t(x^\star) - f_t(x_t) \right)$$

Online Caching: Performance Metric

- $\mathcal{X} = \left\{ \{\mathcal{Y} \times \mathcal{Z}\} \cap \{z_{nij} \leq y_{nj}\ell_{ij}, \quad \forall i,j,n\} \right\}$
  - $\mathcal{Y} = \left\{ \sum_{n\in\mathcal{N}} y_{nj} \leq C_j,\ j \in \mathcal{J} \right\}$. (Respect capacity of each cache)
  - $\mathcal{Z} = \left\{ \sum_{j\in\mathcal{J}} z_{nij} \leq 1,\ n \in \mathcal{N}, i \in \mathcal{I} \right\}$. (No redundant parts)
- If *file requests* and *links' quality* were known in advance, we compute

$$x^\star = \arg\max_{x\in\mathcal{X}} \sum_{t=1}^{T} f_t(x), \qquad \text{and select } x^\star,\ \forall t.$$

- Given $\{x_t\}_T = \{x_0, x_1, \ldots, x_t, \ldots, x_T\}$ according to some online alg.

Definition (Regret Metric)

$$R_T = \sum_{t=1}^{T} \left(f_t(x^\star) - f_t(x_t) \right)$$

Online Caching: Performance Metric

- $\mathcal{X} = \left\{ \{\mathcal{Y} \times \mathcal{Z}\} \cap \{z_{nij} \leq y_{nj}\ell_{ij}, \quad \forall i,j,n\} \right\}$
  - $\mathcal{Y} = \left\{ \sum_{n\in\mathcal{N}} y_{nj} \leq C_j,\ j \in \mathcal{J} \right\}$. (Respect capacity of each cache)
  - $\mathcal{Z} = \left\{ \sum_{j\in\mathcal{J}} z_{nij} \leq 1,\ n \in \mathcal{N}, i \in \mathcal{I} \right\}$. (No redundant parts)
- If *file requests* and *links' quality* were known in advance, we compute

$$x^\star = \arg\max_{x\in\mathcal{X}} \sum_{t=1}^{T} f_t(x), \qquad \text{and select } x^\star,\ \forall t.$$

- Given $\{x_t\}_T = \{x_0, x_1, \ldots, x_t, \ldots, x_T\}$ according to some online alg.

Definition (Regret Metric)

$$R_T = \sum_{t=1}^{T} \left(f_t(x^\star) - f_t(x_t) \right)$$

Online Caching: Performance Metric

- $\mathcal{X} = \left\{ \{\mathcal{Y} \times \mathcal{Z}\} \cap \{z_{nij} \leq y_{nj}\ell_{ij}, \quad \forall i,j,n\} \right\}$
  - $\mathcal{Y} = \left\{ \sum_{n\in\mathcal{N}} y_{nj} \leq C_j,\ j \in \mathcal{J} \right\}$. (Respect capacity of each cache)
  - $\mathcal{Z} = \left\{ \sum_{j\in\mathcal{J}} z_{nij} \leq 1,\ n \in \mathcal{N}, i \in \mathcal{I} \right\}$. (No redundant parts)
- If *file requests* and *links' quality* were known in advance, we compute

$$x^\star = \arg\max_{x\in\mathcal{X}} \sum_{t=1}^{T} f_t(x), \qquad \text{and select } x^\star,\ \forall t.$$

- Given $\{x_t\}_T = \{x_0, x_1, \ldots, x_t, \ldots, x_T\}$ according to some online alg.

Definition (Regret Metric)

$$R_T = \sup_{\{f_t\}_T} \left[\sum_{t=1}^{T} \left(f_t(x^\star) - f_t(x_t) \right) \right]$$

Online Caching: Problem Formulation & Algorithms

- *No-regret* policy: A policy $\{x_t\}_T$ with $R_T = o(T)$: $\lim_{T\to\infty} \frac{R_T}{T} = 0$
- *Regret-optimal* policy: $R_T = O\left(\sqrt{T}\right)$
- $R_T = \sum_{t=1}^{T} \left(f_t(x^*) - f_t(x_t)\right) \leq \sum_{t=1}^{T} \left(\langle c_t, x^* - x_t\rangle\right), \quad c_t = \nabla f_t(x_t)$
- Algorithms: Follow The Regularized Leader (FTRL)

$$x_{t+1} = \arg\min \left\{\langle -c_{1:t}, x\rangle + r_{1:t}(x)\right\}$$

Example: $r_0 = \frac{1}{2\eta}\|x\|^2$, $r_t(x) = 0$ leads to $x_{t+1} = x_t + \eta c_t$ $\leftarrow$ OGA

Recent advances:

- G. Paschos *et. al.* ,"Learning to Cache With No Regrets," in IEEE Infocom, 2019.
- Y. Li *et. al.* , "Online Caching Networks with Adversarial Guarantees," in ACM Sigmetrics, 2021.
- D. Paria *et. al.* , "LeadCache: Regret-Optimal Caching in Networks," in NeurIPS, 2021".

Online Caching: Problem Formulation & Algorithms

- *No-regret* policy: A policy $\{x_t\}_T$ with $R_T = o(T)$: $\lim_{T\to\infty} \frac{R_T}{T} = 0$
- *Regret-optimal* policy: $R_T = O\left(\sqrt{T}\right)$
- $R_T = \sum_{t=1}^{T} \left(f_t(x^*) - f_t(x_t)\right) \leq \sum_{t=1}^{T} \left(\langle c_t, x^* - x_t\rangle\right), \quad c_t = \nabla f_t(x_t)$
- Algorithms: Follow The Regularized Leader (FTRL)

$$x_{t+1} = \arg\min \left\{\langle -c_{1:t}, x\rangle + r_{1:t}(x)\right\}$$

Example: $r_0 = \frac{1}{2\eta}\|x\|^2$, $r_t(x) = 0$ leads to $x_{t+1} = x_t + \eta c_t$ $\leftarrow$ OGA

Recent advances:

- G. Paschos *et. al.* ,"Learning to Cache With No Regrets," in IEEE Infocom, 2019.
- Y. Li *et. al.* , "Online Caching Networks with Adversarial Guarantees," in ACM Sigmetrics, 2021.
- D. Paria *et. al.* , "LeadCache: Regret-Optimal Caching in Networks," in NeurIPS, 2021".

Online Caching: Problem Formulation & Algorithms

- *No-regret* policy: A policy $\{x_t\}_T$ with $R_T = o(T)$: $\lim_{T\to\infty} \frac{R_T}{T} = 0$
- *Regret-optimal* policy: $R_T = O\left(\sqrt{T}\right)$
- $R_T = \sum_{t=1}^{T} \left(f_t(x^\star) - f_t(x_t)\right) \leq \sum_{t=1}^{T} \left(\langle c_t, x^\star - x_t\rangle\right), \quad c_t = \nabla f_t(x_t)$
- Algorithms: Follow The Regularized Leader (FTRL)

$$x_{t+1} = \arg\min \left\{\langle -c_{1:t}, x\rangle + r_{1:t}(x)\right\}$$

Example: $r_0 = \frac{1}{2\eta}\|x\|^2$, $r_t(x) = 0$ leads to $x_{t+1} = x_t + \eta c_t$ $\leftarrow$ OGA

Recent advances:

- G. Paschos *et. al.* ,"Learning to Cache With No Regrets," in IEEE Infocom, 2019.
- Y. Li *et. al.* , "Online Caching Networks with Adversarial Guarantees," in ACM Sigmetrics, 2021.
- D. Paria *et. al.* , "LeadCache: Regret-Optimal Caching in Networks," in NeurIPS, 2021".

Online Caching: Problem Formulation & Algorithms

- *No-regret* policy: A policy $\{x_t\}_T$ with $R_T = o(T)$: $\lim_{T\to\infty} \frac{R_T}{T} = 0$
- *Regret-optimal* policy: $R_T = O\left(\sqrt{T}\right)$
- $R_T = \sum_{t=1}^{T} \left(f_t(x^\star) - f_t(x_t)\right) \leq \sum_{t=1}^{T} \left(\langle c_t, x^\star - x_t\rangle\right), \quad c_t = \nabla f_t(x_t)$
- Algorithms: Follow The Regularized Leader (FTRL)

$$x_{t+1} = \arg\min \left\{\langle -c_{1:t}, x\rangle + r_{1:t}(x)\right\}$$

Example: $r_0 = \frac{1}{2\eta}\|x\|^2$, $r_t(x) = 0$ leads to $x_{t+1} = x_t + \eta c_t \leftarrow$ OGA

Recent advances:

- G. Paschos *et. al.* ,"Learning to Cache With No Regrets," in IEEE Infocom, 2019.
- Y. Li *et. al.* , "Online Caching Networks with Adversarial Guarantees," in ACM Sigmetrics, 2021.
- D. Paria *et. al.* , "LeadCache: Regret-Optimal Caching in Networks," in NeurIPS, 2021".

Online Caching: Problem Formulation & Algorithms

- *No-regret* policy: A policy $\{x_t\}_T$ with $R_T = o(T)$: $\lim_{T\to\infty} \frac{R_T}{T} = 0$
- *Regret-optimal* policy: $R_T = O\left(\sqrt{T}\right)$
- $R_T = \sum_{t=1}^{T} \left(f_t(x^\star) - f_t(x_t)\right) \leq \sum_{t=1}^{T} \left(\langle c_t, x^\star - x_t\rangle\right), \quad c_t = \nabla f_t(x_t)$
- Algorithms: Follow The Regularized Leader (FTRL)

$$x_{t+1} = \arg\min \left\{\langle -c_{1:t}, x\rangle + r_{1:t}(x)\right\}$$

Example: $r_0 = \frac{1}{2\eta}\|x\|^2$, $r_t(x) = 0$ leads to $x_{t+1} = x_t + \eta c_t \leftarrow$ OGA

Recent advances:

- G. Paschos *et. al.* ,"Learning to Cache With No Regrets," in IEEE Infocom, 2019.
- Y. Li *et. al.* , "Online Caching Networks with Adversarial Guarantees," in ACM Sigmetrics, 2021.
- D. Paria *et. al.* , "LeadCache: Regret-Optimal Caching in Networks," in NeurIPS, 2021".

Online Caching: Problem Formulation & Algorithms

- *No-regret* policy: A policy $\{x_t\}_T$ with $R_T = o(T)$: $\lim_{T\to\infty} \frac{R_T}{T} = 0$
- *Regret-optimal* policy: $R_T = O\left(\sqrt{T}\right)$
- $R_T = \sum_{t=1}^{T} \left(f_t(x^\star) - f_t(x_t)\right) \leq \sum_{t=1}^{T} \left(\langle c_t, x^\star - x_t\rangle\right), \quad c_t = \nabla f_t(x_t)$
- Algorithms: Follow The Regularized Leader (FTRL)

$$x_{t+1} = \arg\min \left\{\langle -c_{1:t}, x\rangle + r_{1:t}(x)\right\}$$

Example: $r_0 = \frac{1}{2\eta}\|x\|^2$, $r_t(x) = 0$ leads to $x_{t+1} = x_t + \eta c_t \leftarrow$ OGA

Recent advances:

- G. Paschos *et. al.* ,"Learning to Cache With No Regrets," in IEEE Infocom, 2019.
- Y. Li *et. al.* , "Online Caching Networks with Adversarial Guarantees," in ACM Sigmetrics, 2021.
- D. Paria *et. al.* , "LeadCache: Regret-Optimal Caching in Networks," in NeurIPS, 2021".

Online Caching: Problem Formulation & Algorithms

- *No-regret* policy: A policy $\{x_t\}_T$ with $R_T = o(T)$: $\lim_{T\to\infty} \frac{R_T}{T} = 0$
- *Regret-optimal* policy: $R_T = O\left(\sqrt{T}\right)$
- $R_T = \sum_{t=1}^{T} \left(f_t(x^\star) - f_t(x_t)\right) \leq \sum_{t=1}^{T} \left(\langle c_t, x^\star - x_t\rangle\right), \quad c_t = \nabla f_t(x_t)$
- Algorithms: Follow The Regularized Leader (FTRL)

$$x_{t+1} = \arg\min \left\{\langle -c_{1:t}, x\rangle + r_{1:t}(x)\right\}$$

Example: $r_0 = \frac{1}{2\eta}\|x\|^2$, $r_t(x) = 0$ leads to $x_{t+1} = x_t + \eta c_t \leftarrow$ OGA

Recent advances:

- G. Paschos *et. al.* ,"Learning to Cache With No Regrets," in IEEE Infocom, 2019.
- Y. Li *et. al.* , "Online Caching Networks with Adversarial Guarantees," in ACM Sigmetrics, 2021.
- D. Paria *et. al.* , "LeadCache: Regret-Optimal Caching in Networks," in NeurIPS, 2021".

Table of Contents

1 Online Caching: System Model & Problem Formulation

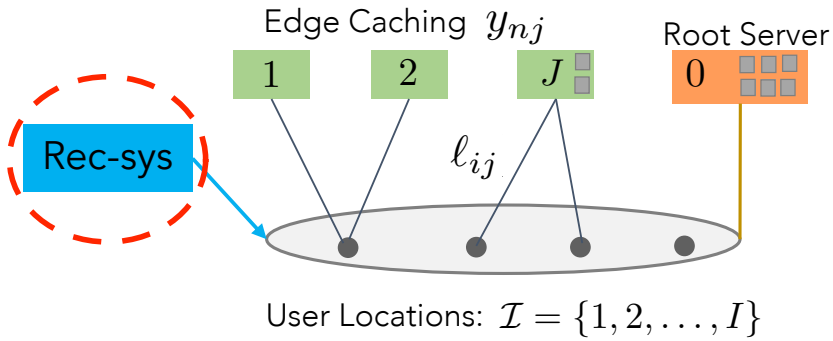
2 Optimistic Learning via Recommendations

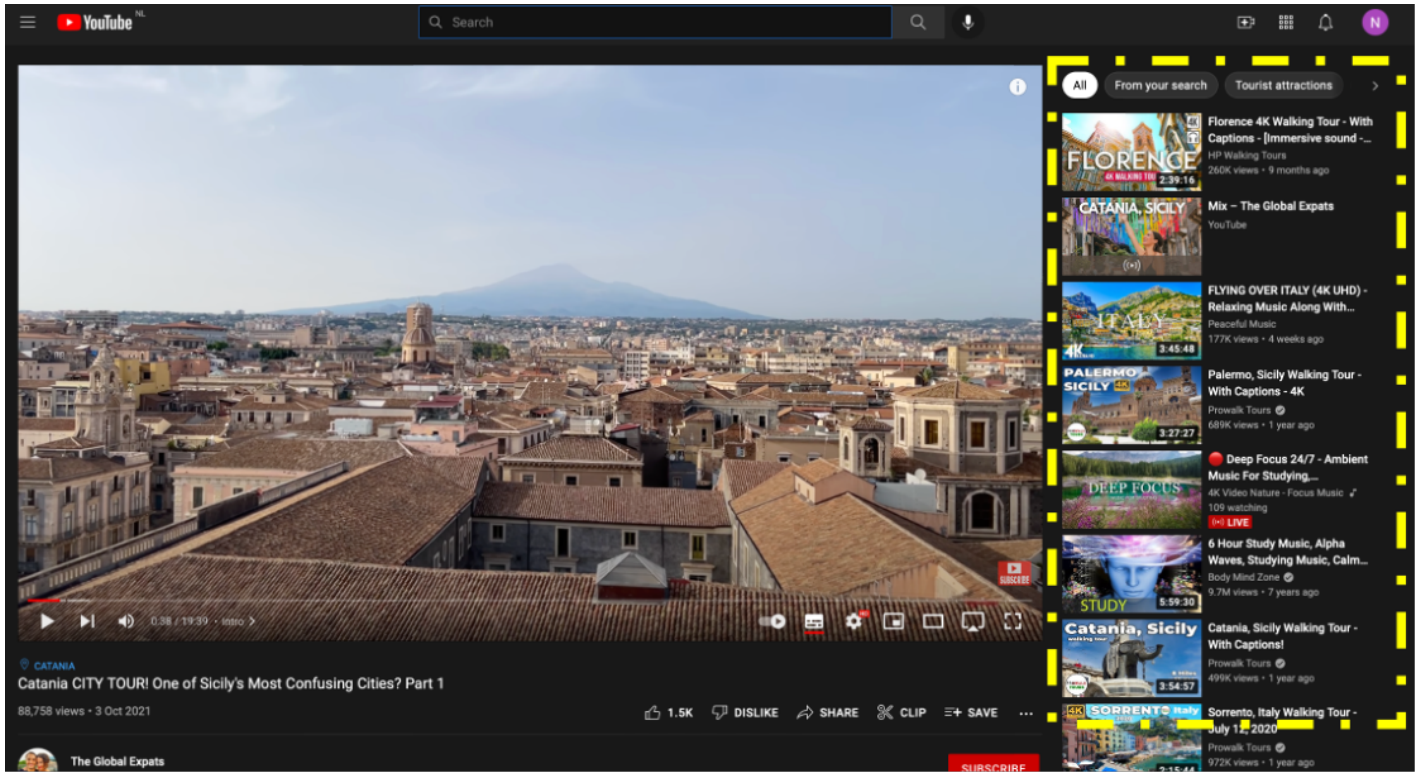
3 Optimistic Learning with a Stable Predictor

4 Optimistic Learning with Constraints

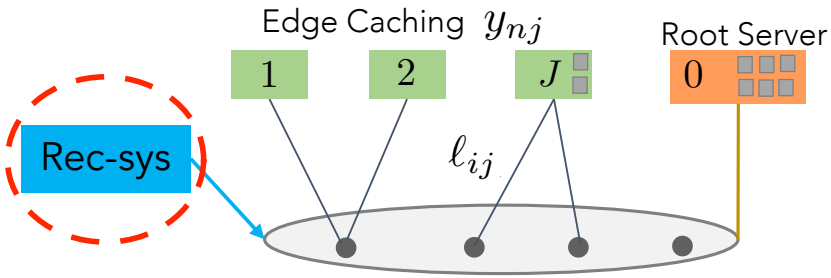
5 Summary & Outlook

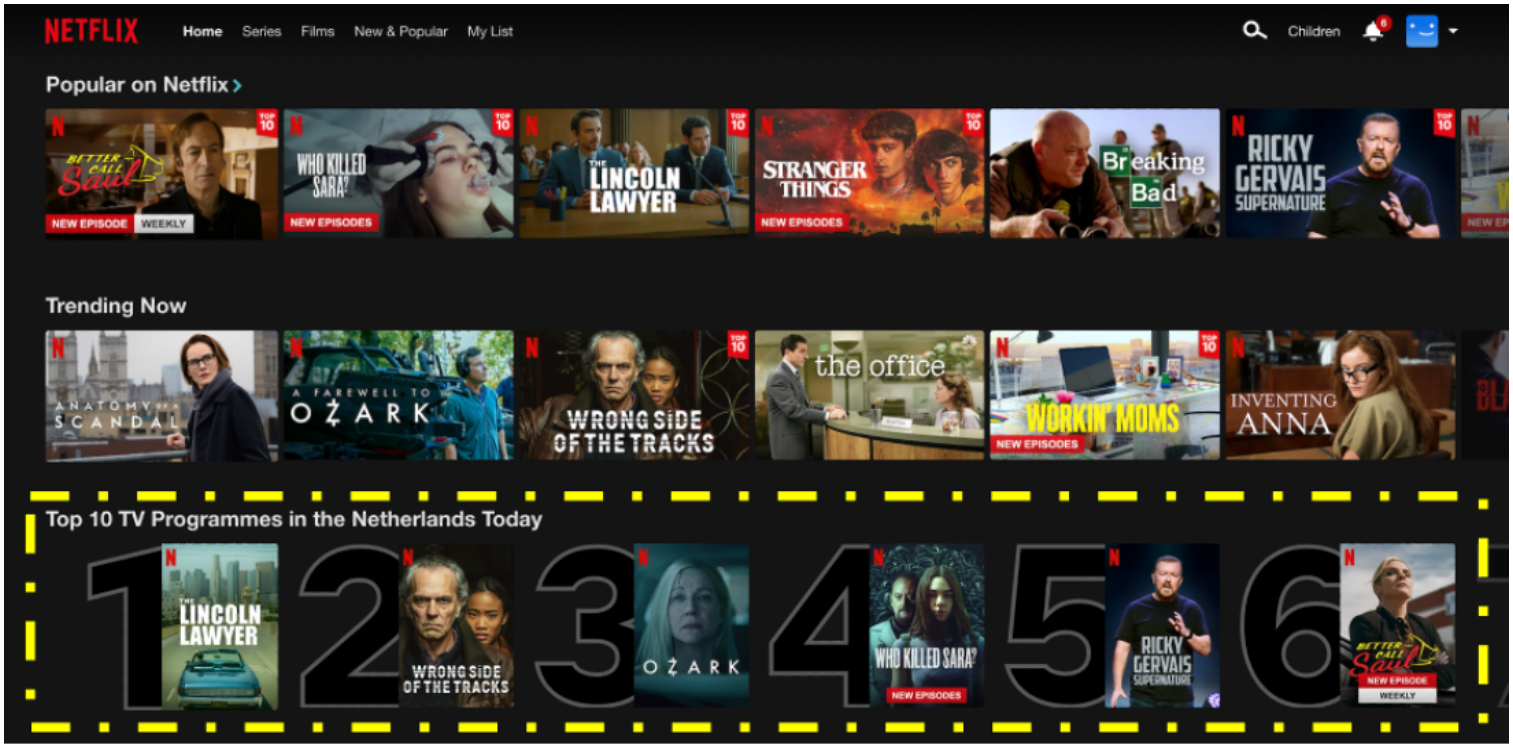
Optimistic Learning via Recommendations



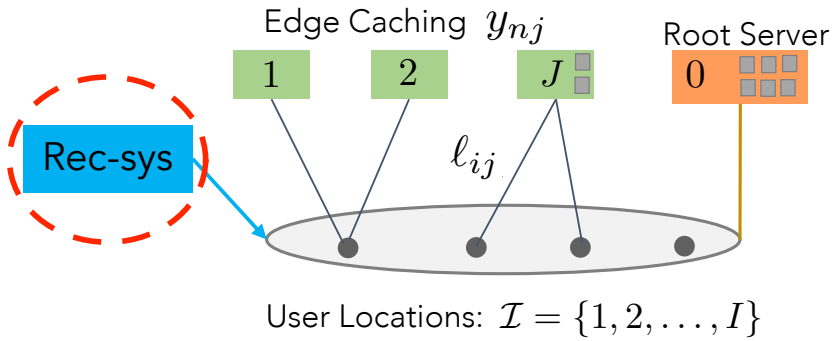


Optimistic Learning via Recommendations





Optimistic Learning via Recommendations



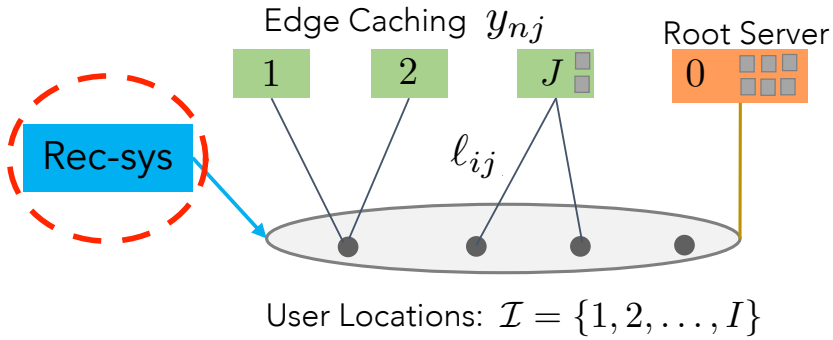
Research Question

Can we leverage *recommendations* to minimize the regret beyond $O(\sqrt{T})$, while still maintaining the $O(\sqrt{T})$ guarantee regardless of their quality?

for $t = 1, 2, \ldots, T$ do:

- A request $q_t \in \{0, 1\}^{I \times N}$ arrives.
- Served according to x_t; a utility is gained $f_t(x_t) = \sum_{n,i,j} w_{nij} q^t_{ni} z^t_{nij}$.
- **Predict next slot's utility:** $\tilde{c}_{t+1} = w_{nij} \tilde{q}_{t+1}$
- Compute x_{t+1}

Optimistic Learning via Recommendations



Research Question

Can we leverage *recommendations* to minimize the regret beyond $O(\sqrt{T})$, while still maintaining the $O(\sqrt{T})$ guarantee regardless of their quality?

for $t = 1, 2, \ldots, T$ do:

- A request $q_t \in \{0, 1\}^{I \times N}$ arrives.
- Served according to x_t; a utility is gained $f_t(x_t) = \sum_{n,i,j} w_{nij} q_{ni}^t z_{nij}^t$.
- **Predict next slot's utility:** $\tilde{c}_{t+1} = w_{nij} \tilde{q}_{t+1}$
- Compute x_{t+1}

Optimistic Learning via Recommendations

- Optimistic update[2]: $x_{t+1} = \arg\min_{x \in \mathcal{X}} \left\{ -\langle c_{1:t} + \tilde{c}_{t+1}, x \rangle + r_{1:t}(x) \right\}$
- Prediction adaptive regularizers:

$$h_t = \|c_t - \tilde{c}_t\|^2, \qquad r_t(x) \approx \left(\sqrt{h_{1:t}} - \sqrt{h_{1:t-1}} \right) \|x - x_t\|^2$$

- Regret optimal refined guarantee:

Result 1

$$R_T \leq D_{\mathcal{X}} \sqrt{\sum_{t=1}^{T} \|c_t - \tilde{c}_t\|} \qquad , D_{\mathcal{X}} = \sqrt{2(1 + JC)}$$

- Best case: $R_T \leq 0$
- Worst case: $R_T \leq O(\sqrt{T})$
- Plausible case: $R_T \leq O(\text{constant})$

[2] M.Mohri *et. al.* "Accelerating Online Convex Optimization via Adaptive Prediction", AISTATS 2016.

Optimistic Learning via Recommendations

- Optimistic update[2]: $x_{t+1} = \arg\min_{x \in \mathcal{X}} \left\{ -\langle c_{1:t} + \tilde{c}_{t+1}, x \rangle + r_{1:t}(x) \right\}$
- Prediction adaptive regularizers:

$$h_t = \|c_t - \tilde{c}_t\|^2, \qquad r_t(x) \approx \left(\sqrt{h_{1:t}} - \sqrt{h_{1:t-1}} \right) \|x - x_t\|^2$$

- Regret optimal refined guarantee:

Result 1

$$R_T \leq D_{\mathcal{X}} \sqrt{\sum_{t=1}^{T} \|c_t - \tilde{c}_t\|} \qquad , D_{\mathcal{X}} = \sqrt{2(1 + JC)}$$

- Best case: $R_T \leq 0$
- Worst case: $R_T \leq O(\sqrt{T})$
- Plausible case: $R_T \leq O(\text{constant})$

[2] M.Mohri *et. al.* "Accelerating Online Convex Optimization via Adaptive Prediction", AISTATS 2016.

Optimistic Learning via Recommendations

- Optimistic update[2]: $x_{t+1} = \arg\min_{x \in \mathcal{X}} \left\{ -\langle c_{1:t} + \tilde{c}_{t+1}, x \rangle + r_{1:t}(x) \right\}$
- Prediction adaptive regularizers:

$$h_t = \|c_t - \tilde{c}_t\|^2, \qquad r_t(x) \approx \left(\sqrt{h_{1:t}} - \sqrt{h_{1:t-1}} \right) \|x - x_t\|^2$$

- Regret optimal refined guarantee:

Result 1

$$R_T \leq D_{\mathcal{X}} \sqrt{\sum_{t=1}^{T} \|c_t - \tilde{c}_t\|} \qquad , D_{\mathcal{X}} = \sqrt{2(1 + JC)}$$

- Best case: $R_T \leq 0$
- Worst case: $R_T \leq O(\sqrt{T})$
- Plausible case: $R_T \leq O(\text{constant})$

[2] M.Mohri *et. al.* "Accelerating Online Convex Optimization via Adaptive Prediction", AISTATS 2016.

Optimistic Learning via Recommendations

- Optimistic update[2]: $x_{t+1} = \arg\min_{x \in \mathcal{X}} \left\{ -\langle c_{1:t} + \tilde{c}_{t+1}, x \rangle + r_{1:t}(x) \right\}$
- Prediction adaptive regularizers:

$$h_t = \|c_t - \tilde{c}_t\|^2, \qquad r_t(x) \approx \left(\sqrt{h_{1:t}} - \sqrt{h_{1:t-1}} \right) \|x - x_t\|^2$$

- Regret optimal refined guarantee:

Result 1

$$R_T \leq D_{\mathcal{X}} \sqrt{\sum_{t=1}^{T} \|c_t - \tilde{c}_t\|} \qquad , D_{\mathcal{X}} = \sqrt{2(1 + JC)}$$

- Best case: $R_T \leq 0$
- Worst case: $R_T \leq O(\sqrt{T})$
- Plausible case: $R_T \leq O(\text{constant})$

[2] M.Mohri *et. al.* "Accelerating Online Convex Optimization via Adaptive Prediction", AISTATS 2016.

Optimistic Learning via Recommendations

- Optimistic update[2]: $x_{t+1} = \arg\min_{x \in \mathcal{X}} \left\{ -\langle c_{1:t} + \tilde{c}_{t+1}, x \rangle + r_{1:t}(x) \right\}$
- Prediction adaptive regularizers:

$$h_t = \|c_t - \tilde{c}_t\|^2, \qquad r_t(x) \approx \left(\sqrt{h_{1:t}} - \sqrt{h_{1:t-1}} \right) \|x - x_t\|^2$$

- Regret optimal refined guarantee:

Result 1

$$R_T \leq 2 D_{\mathcal{X}} \sqrt{\sum_{t=1}^{T} \|c_t - \tilde{c}_t\|} \qquad , D_{\mathcal{X}} = \sqrt{2(1 + JC)}$$

- Best case: $R_T \leq 0$
- Worst case: $R_T \leq O(\sqrt{T})$
- Plausible case: $R_T \leq O(\text{constant})$

[2] M.Mohri *et. al.* "Accelerating Online Convex Optimization via Adaptive Prediction", AISTATS 2016.

Optimistic Learning via Recommendations

- Optimistic update[2]: $x_{t+1} = \arg\min_{x \in \mathcal{X}} \left\{ -\langle c_{1:t} + \tilde{c}_{t+1}, x \rangle + r_{1:t}(x) \right\}$
- Prediction adaptive regularizers:

$$h_t = \|c_t - \tilde{c}_t\|^2, \qquad r_t(x) \approx \left(\sqrt{h_{1:t}} - \sqrt{h_{1:t-1}} \right) \|x - x_t\|^2$$

- Regret optimal refined guarantee:

Result 1

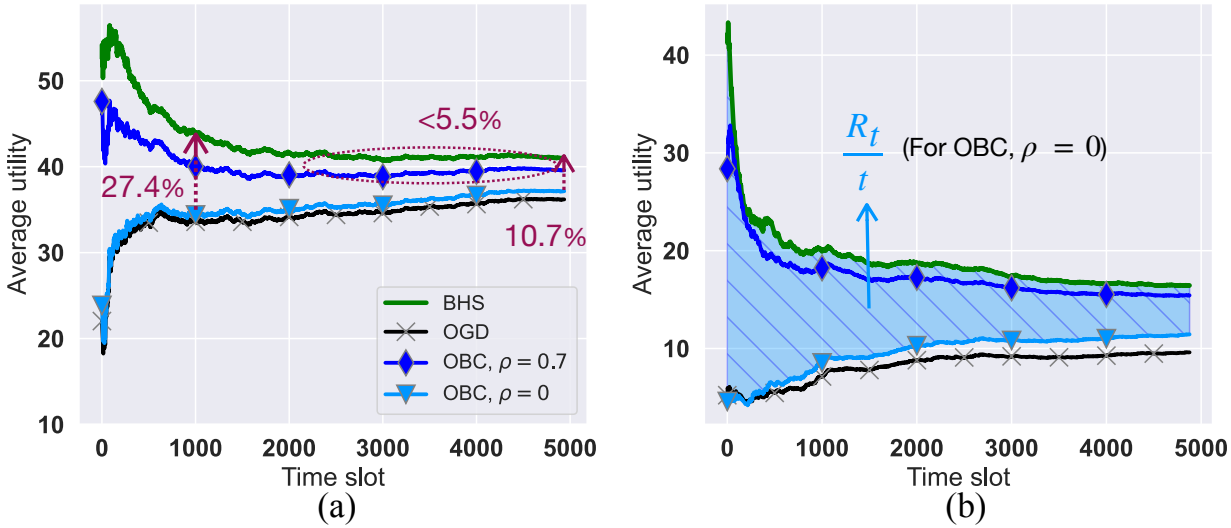
$$R_T \leq \sqrt{2} D_{\mathcal{X}} \sqrt{\sum_{t=1}^{T} \|c_t - \tilde{c}_t\|} \qquad , D_{\mathcal{X}} = \sqrt{2(1 + JC)}$$

- Best case: $R_T \leq 0$
- Worst case: $R_T \leq O(\sqrt{T})$
- Plausible case: $R_T \leq O(\text{constant})$

[2] M.Mohri *et. al.* "Accelerating Online Convex Optimization via Adaptive Prediction", AISTATS 2016.

Simulations

- Scenario (a): Stationary zipf requests.
- Scenario (b): Actual YouTube request trace.



- BHS: The benchmark policy, set with full future knowledge.
- OGD: The comparator, does not use predictions[3].
- OBC: Optimitic (ρ is the probability of following the recom.).

[3] G. Paschos *et. al.* ,"Learning to Cache With No Regrets," in IEEE Infocom, 2019.

Table of Contents

1 Online Caching: System Model & Problem Formulation

2 Optimistic Learning via Recommendations

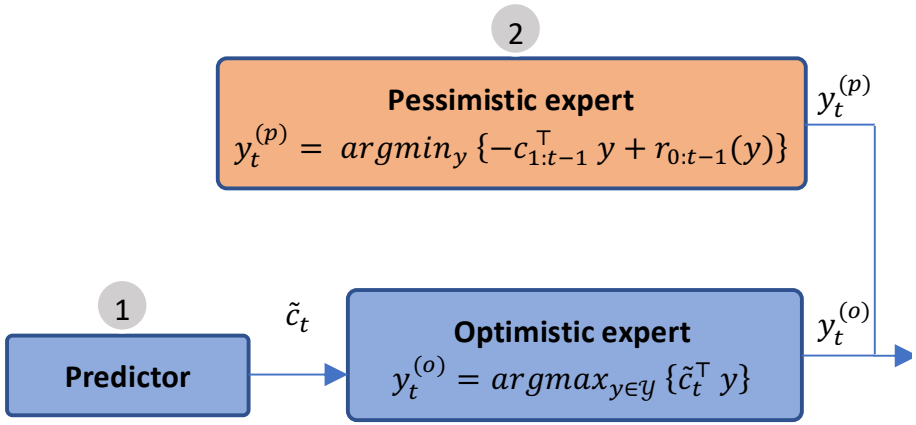
3 Optimistic Learning with a Stable Predictor

4 Optimistic Learning with Constraints

5 Summary & Outlook

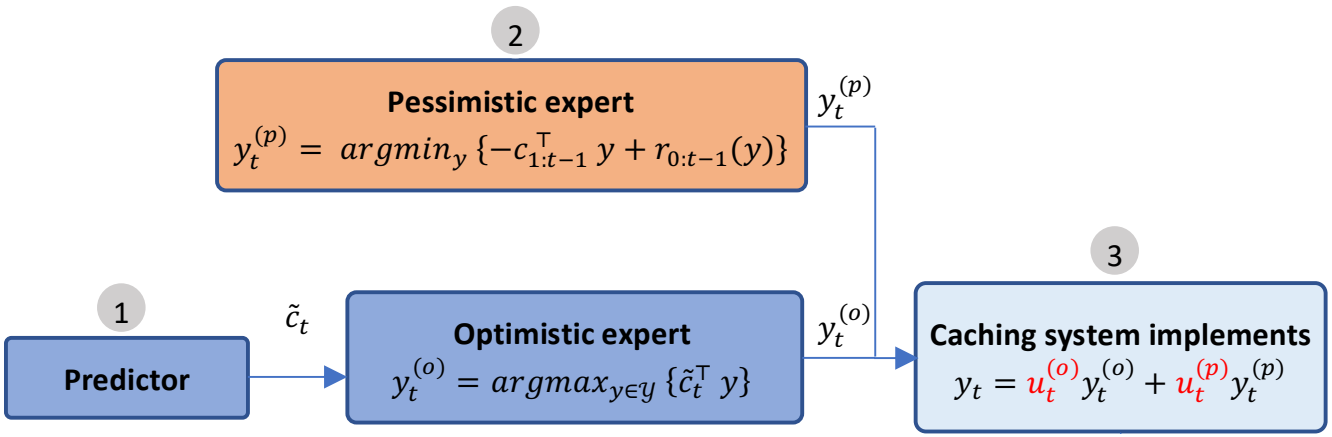
Optimistic Online Caching: Stable predictors

- Experts model. Objective: caching $R_T = R_T$ of the best expert.
- Meta-policy: Convex combination via u_t.
- Meta-learner: OGD on u_{t+1}.



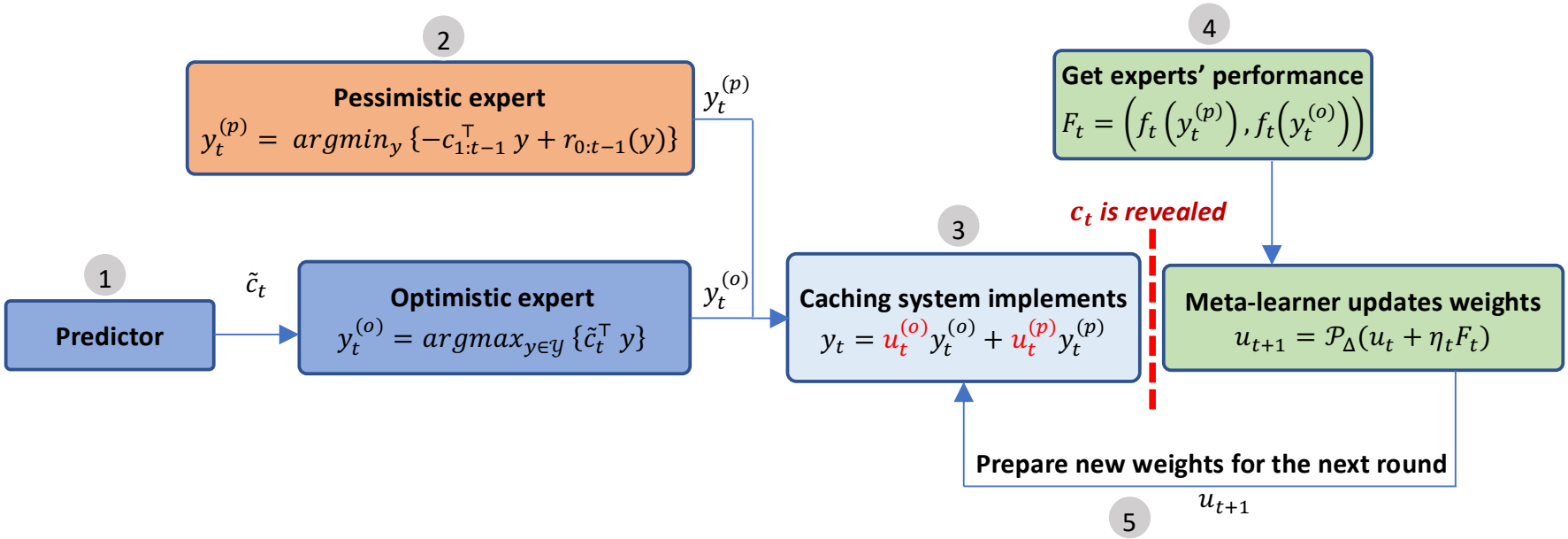
Optimistic Online Caching: Stable predictors

- Experts model. Objective: caching $R_T = R_T$ of the best expert.
- Meta-policy: Convex combination via u_t.
- Meta-learner: OGD on u_{t+1}.

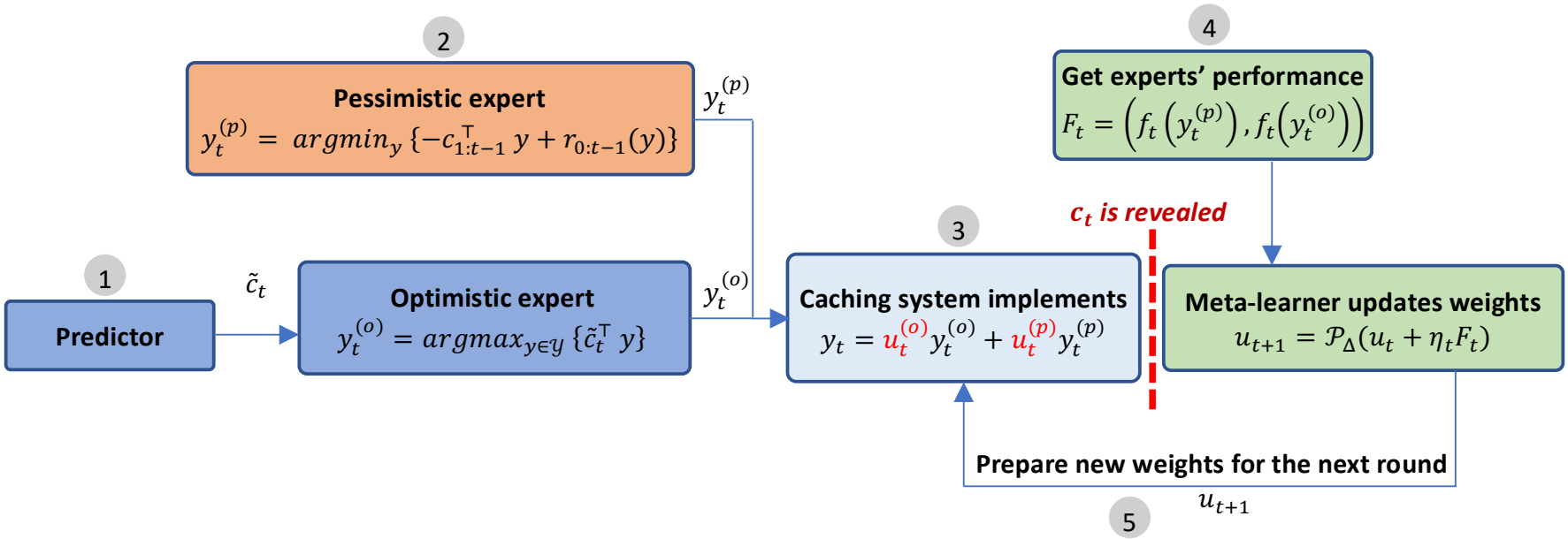


Optimistic Online Caching: Stable predictors

- Experts model. Objective: caching $R_T = R_T$ of the best expert.
- Meta-policy: Convex combination via u_t.
- Meta-learner: OGD on u_{t+1}.



Optimistic Online Caching: Stable predictors



Result 2

$$R_T = \leq R_T^{(u)} + \underbrace{\min\left\{R_T^{(p)}, R_T^{(o)}\right\}}_{A^\star}.$$

- Worst case: $R_T \leq O(\sqrt{T})$
- Best case: $R_T < 0$

Table of Contents

1 Online Caching: System Model & Problem Formulation

2 Optimistic Learning via Recommendations

3 Optimistic Learning with a Stable Predictor

4 Optimistic Learning with Constraints

5 Summary & Outlook

Optimistic Learning with Constraints

- Violation metric: $V_T = \left[\sum_{t=1}^{T} g_t(x_t)\right]_+$. Objective: $V_T = o(T)$
- Lagrangian-type function $\mathcal{L}_t(x, \lambda) = r_t + f_t(x) + \lambda g_t(x) - r'_t(x)$
- Descent ascent Steps

$$\lambda_{t+1} = \arg\max_{\lambda \geq 0} \left\{ - r'_{0:t}(\lambda) + \lambda \sum_{i=1}^{t} g_i(x_i) \right\},$$

$$x_{t+1} = \arg\min_{x} \left\{ r_{0:t}(x) + \left\langle \sum_{i=1}^{t+1} \lambda_i s_i - c_{1:t} - \tilde{c}_{t+1}, x \right\rangle \right\}.$$

Result 3

For $\beta = 1/2$:

$$R_T \leq O(\sqrt{T})$$

$$V_T \leq O(T^{\frac{3}{4}})$$

Table of Contents

1 Online Caching: System Model & Problem Formulation

2 Optimistic Learning via Recommendations

3 Optimistic Learning with a Stable Predictor

4 Optimistic Learning with Constraints

5 Summary & Outlook

Summary & Outlook

- Using recommendations in the framework of online caching to obtain $R_T \leq 0$ and maintain $R_T \leq O(\sqrt{T})$.
- Caching (network) optimization:

