Online Decision-making with a Expert Committee and Its Application in FahsionFlow

Zalando Search Team

Zalando SE

hanchen.xiong@zalando.de

May 1, 2016

Overview

1. Online prediction with experts,repeated game playing and convex optimization
   - Prediction with Experts' advice
   - Online repeated game playing
   - Online convex optimization
2. Bandit Optimization in metric spaces
   - Bandit: play games with limited feedbacks
   - Gaussian process bandit optimization
   - General Bayesian optimization
3. Concept drift in online decision
   - Stability v.s. adaptivity
   - Explicit detection of concept drifts
   - Time-varying surface-response bandit optimization
   - Adaptive regret for tracking the best expert
4. New sku proposal plans in FashionFlow
   - Expert setting v.s. Bandit setting
   - Algorithm evaluation

Online prediction with experts, repeated game playing and convex optimization

A gentle start

- Task: online prediction of a binary sequence, *i.e.* sequentially forecast a value $y^{(t)} \in \{-1, 1\}$ at time t based on historical values $\{y^{(i)}\}_{i=1}^{(t-1};$

A gentle start

- Task: online prediction of a binary sequence, *i.e.* sequentially forecast a value $y^{(t)} \in \{-1, 1\}$ at time t based on historical values $\{y^{(i)}\}_{i=1}^{(t-1}$;
- Expert committee: an expert E_n is a man/woman who can make a prediction, $f_n^{(t)}$, using different strategies (algorithms/heuristics/data resources); assume there are N experts in the committee;

A gentle start

- Task: online prediction of a binary sequence, *i.e.* sequentially forecast a value $y^{(t)} \in \{-1, 1\}$ at time t based on historical values $\{y^{(i)}\}_{i=1}^{(t-1}$;
- Expert committee: an expert E_n is a man/woman who can make a prediction, $f_n^{(t)}$, using different strategies (algorithms/heuristics/data resources); assume there are N experts in the committee;
- Side information: other task-relevant information may available, *e.g.* $\{\mathbf{x}^{(t)}\}_{i=1}^{t}$;

A gentle start

- Task: online prediction of a binary sequence, *i.e.* sequentially forecast a value $y^{(t)} \in \{-1, 1\}$ at time t based on historical values $\{y^{(i)}\}_{i=1}^{(t-1}$;
- Expert committee: an expert E_n is a man/woman who can make a prediction, $f_n^{(t)}$, using different strategies (algorithms/heuristics/data resources); assume there are N experts in the committee;
- Side information: other task-relevant information may available, *e.g.* $\{\mathbf{x}^{(t)}\}_{i=1}^{t}$;
- Forecaster: an aggregating policy π is a function which map experts' advices $\Longrightarrow$ final decision $\hat{y}_t$: $\hat{y}_t = \pi(f_1^{(t)}, f_2^{(t)}, \cdots, f_N^{(t)})$;

A gentle start

- Task: online prediction of a binary sequence, *i.e.* sequentially forecast a value $y^{(t)} \in \{-1, 1\}$ at time t based on historical values $\{y^{(i)}\}_{i=1}^{(t-1)}$;
- Expert committee: an expert E_n is a man/woman who can make a prediction, $f_n^{(t)}$, using different strategies (algorithms/heuristics/data resources); assume there are N experts in the committee;
- Side information: other task-relevant information may available, *e.g.* $\{\mathbf{x}^{(t)}\}_{i=1}^{t}$;
- Forecaster: an aggregating policy π is a function which map experts' advices $\Longrightarrow$ final decision $\hat{y}_t$: $\hat{y}_t = \pi(f_1^{(t)}, f_2^{(t)}, \cdots, f_N^{(t)})$;
- Weights update: at the beginning, each expert is assigned a (uniform) weight $w_{n,0} = 1$, and it will be updated along the sequence;

A gentle start

- Task: online prediction of a binary sequence, *i.e.* sequentially forecast a value $y^{(t)} \in \{-1, 1\}$ at time t based on historical values $\{y^{(i)}\}_{i=1}^{(t-1)}$;
- Expert committee: an expert E_n is a man/woman who can make a prediction, $f_n^{(t)}$, using different strategies (algorithms/heuristics/data resources); assume there are N experts in the committee;
- Side information: other task-relevant information may available, *e.g.* $\{\mathbf{x}^{(t)}\}_{i=1}^{t}$;
- Forecaster: an aggregating policy π is a function which map experts' advices $\Longrightarrow$ final decision $\hat{y}_t$: $\hat{y}_t = \pi(f_1^{(t)}, f_2^{(t)}, \cdots, f_N^{(t)})$;
- Weights update: at the beginning, each expert is assigned a (uniform) weight $w_{n,0} = 1$, and it will be updated along the sequence;
- Loss: $l(\hat{y}^{(t)}, y^{(t)}) = \mathbf{1}_{\hat{y}^{(t)} \neq y^{(t)}}$, $l(f_n^{(t)}, y^{(t)}) = \mathbf{1}_{f_n^{(t)} \neq y^{(t)}}$

A gentle start, cont.

A simple policy

The final decision is made with the majority voting from the expert committee: $\hat{y}^{(t)} = \mathbf{sign}(\frac{\sum_{n=1}^{N} f_i^{(t)}}{N})$;

A gentle start, cont.

A simple policy

The final decision is made with the majority voting from the expert committee: $\hat{y}^{(t)} = \textbf{sign}(\frac{\sum_{n=1}^{N} f_i^{(t)}}{N})$;

An ideal scenario

we know that there exist some experts who are perfect in the given task;

A gentle start, cont.

A simple policy

The final decision is made with the majority voting from the expert committee: $\hat{y}^{(t)} = \mathbf{sign}(\frac{\sum_{n=1}^{N} f_i^{(t)}}{N})$;

An ideal scenario

we know that there exist some experts who are perfect in the given task;

A simple weight update scheme

$w_n^{(t)} \leftarrow 0$ if expert E_n makes a mistake at time $t-1$, *i.e.* kick E_n out of the committee;

A gentle start, cont.

A simple policy

The final decision is made with the majority voting from the expert committee: $\hat{y}^{(t)} = \mathbf{sign}(\frac{\sum_{n=1}^{N} f_i^{(t)}}{N})$;

An ideal scenario

we know that there exist some experts who are perfect in the given task;

A simple weight update scheme

$w_n^{(t)} \leftarrow 0$ if expert E_n makes a mistake at time $t-1$, *i.e.* kick E_n out of the committee;

Cummulative loss:

$$\hat{L}^{(t)} = \sum_{i=t}^{t} l(\hat{y}^{(t)}, y^{(t)}) \leq \log_2 N; \qquad (1)$$

Generalized committee

A more realistic scenario

we know that in the committee there exists a best expert who can work better than others in the given task;

Generalized committee

A more realistic scenario

we know that in the committee there exists a best expert who can work better than others in the given task;

Regret $R_n^{(t)}$: the extra losses the forecaster made without exclusively following the expert E_n up to time t:

$$R_n^{(t)} = \hat{L}^{(t)} - L_n^{(t)} = \sum_{i=t}^{t} l(\hat{y}^{(t)}, y^{(t)}) - \sum_{i=t}^{t} l(f_n^{(t)}, y^{(t)}) \qquad (2)$$

Generalized committee

A more realistic scenario

we know that in the committee there exists a best expert who can work better than others in the given task;

Regret $R_n^{(t)}$: the extra losses the forecaster made without exclusively following the expert E_n up to time t:

$$R_n^{(t)} = \hat{L}^{(t)} - L_n^{(t)} = \sum_{i=t}^{t} l(\hat{y}^{(t)}, y^{(t)}) - \sum_{i=t}^{t} l(f_n^{(t)}, y^{(t)}) \tag{2}$$

the upper bound of regret

$R^{(t)*} = \max_{n \in [1,N]} R_n^{(t)} = \hat{L}^{(t)} - \min_{n \in [1,N]} L_n^{(t)}$

Weighted majority algorithm

- Task: online prediction of a binary sequence, *i.e.* sequentially forecast a value $y^{(t)} \in \{-1, 1\}$ at time t based on historical values $\{y^{(i)}\}_{i=1}^{(t-1)}$;
- Expert committee: an expert E_n is a man/woman who can make a prediction, $f_n^{(t)}$, using different strategies (algorithms/heuristics/data resources); assume there are N experts in the committee;
- Loss: $l(\hat{y}^{(t)}, y^{(t)}) = \mathbf{1}_{\hat{y}^{(t)} \neq y^{(t)}}$, $l(f_n^{(t)}, y^{(t)}) = \mathbf{1}_{f_n^{(t)} \neq y^{(t)}}$

Weighted majority algorithm

- Task: online prediction of a binary sequence, *i.e.* sequentially forecast a value $y^{(t)} \in \{-1, 1\}$ at time t based on historical values $\{y^{(i)}\}_{i=1}^{(t-1)}$;
- Expert committee: an expert E_n is a man/woman who can make a prediction, $f_n^{(t)}$, using different strategies (algorithms/heuristics/data resources); assume there are N experts in the committee;
- Loss: $l(\hat{y}^{(t)}, y^{(t)}) = \mathbf{1}_{\hat{y}^{(t)} \neq y^{(t)}}$, $l(f_n^{(t)}, y^{(t)}) = \mathbf{1}_{f_n^{(t)} \neq y^{(t)}}$

- Forecaster: $\hat{y}_t = \mathbf{sign}(\sum_{n:f_{n,t}=1} w_n^{(t-1)} - \sum_{m:f_{m,t}=-1} w_m^{(t-1)})$;

Weighted majority algorithm

- Task: online prediction of a binary sequence, *i.e.* sequentially forecast a value $y^{(t)} \in \{-1, 1\}$ at time t based on historical values $\{y^{(i)}\}_{i=1}^{(t-1)}$;
- Expert committee: an expert E_n is a man/woman who can make a prediction, $f_n^{(t)}$, using different strategies (algorithms/heuristics/data resources); assume there are N experts in the committee;
- Loss: $l(\hat{y}^{(t)}, y^{(t)}) = \mathbf{1}_{\hat{y}^{(t)} \neq y^{(t)}}$, $l(f_n^{(t)}, y^{(t)}) = \mathbf{1}_{f_n^{(t)} \neq y^{(t)}}$

- Forecaster: $\hat{y}_t = \mathbf{sign}(\sum_{n:f_{n,t}=1} w_n^{(t-1)} - \sum_{m:f_{m,t}=-1} w_m^{(t-1)})$;
- Weights update: if one expert E_n predicts wrongly, decrease its weight

$$w_n^{(t+1)} = (1-\eta) w_n^{(t)} \tag{3}$$

where $\eta \leq \frac{1}{2}$.

Analysis on weighted majority algorithm

Theorem (cummulative loss bound using weighted majority)

After T steps, $\hat{L}^{(T)} \leq 2(1+\eta)\min_{1\in[1,N]} L_n^{(T)} + \frac{2\ln N}{\eta}$

Analysis on weighted majority algorithm

Theorem (cummulative loss bound using weighted majority)

After T steps, $\hat{L}^{(T)} \leq 2(1+\eta)\min_{1\in[1,N]} L_n^{(T)} + \frac{2\ln N}{\eta}$

Proof: Let $\Gamma^{(t)} = \sum_{n\in[1,N]} w_n^{(t)}$, then $\Gamma^{(1)} = N$. Also, if the forecaster makes a mistake, $\hat{y}^{(t)} \neq y^{(t)}$

$$\Gamma^{(t+1)} \leq \Gamma^{(t)}(\frac{1}{2} + \frac{1}{2}(1-\eta) = \Gamma^{(t)}(1-\frac{\eta}{2}) \tag{4}$$

therefore:

$$\Gamma^{(T+1)} \leq N(1-\frac{\eta}{2})^{\hat{L}^{(T)}} \tag{5}$$

for any individual expert n

$$w_n^{(T+1)} = (1-\eta)^{L_n^{(T)}} \tag{6}$$

since $w_n^{(T+1)} \leq \Gamma^{(T+1)} \Longrightarrow (1-\eta)^{L_n^{(T)}} \leq N(1-\frac{\eta}{2})^{\hat{L}^{(T)}}$

Analysis on weighted majority algorithm, cont.

$$
\begin{aligned}
(1-\eta)^{L_n^{(T)}} &\leq N(1-\tfrac{\eta}{2})^{\hat{L}^{(T)}} \\
\Leftrightarrow L_n^{(T)} \ln(1-\eta) &\leq \ln N + \hat{L}^{(T)} \ln(1-\tfrac{\eta}{2}) \\
\Leftrightarrow -\ln(1-\tfrac{\eta}{2}) &\leq -L_n^{(T)} \ln(1-\eta) + \ln N \\
\xRightarrow{x \leq -\ln(1-x)} \tfrac{\eta}{2}\hat{L}^{(T)} &\leq -L_n^{(T)} \ln(1-\eta) + \ln N \\
\xRightarrow{-\ln(1-x)\leq x+x^2,\text{when } x\leq 1/2} \tfrac{\eta}{2}\hat{L}^{(T)} &\leq L_n^{(T)}\eta(1+\eta) + \ln N \\
\Leftrightarrow \hat{L}^{(T)} &\leq 2(1+\eta)L_n^{(T)} + \frac{2\ln N}{\eta}
\end{aligned}
\tag{7}
$$

Randomized weighted majority algorithm

- Task: online prediction of a binary sequence, *i.e.* sequentially forecast a value $y^{(t)} \in \{-1, 1\}$ at time t based on historical values $\{y^{(i)}\}_{i=1}^{(t-1)}$;
- Expert committee: an expert E_n is a man/woman who can make a prediction, $f_n^{(t)}$, using different strategies (algorithms/heuristics/data resources); assume there are N experts in the committee;
- Loss: $l(\hat{y}^{(t)}, y^{(t)}) = \mathbf{1}_{\hat{y}^{(t)} \neq y^{(t)}}$, $l(f_n^{(t)}, y^{(t)}) = \mathbf{1}_{f_n^{(t)} \neq y^{(t)}}$
- Weights update: if one expert E_n predicts wrongly, decrease its weight

$$w_n^{(t+1)} = (1 - \eta) w_n^{(t)} \tag{8}$$

where $\eta \leq 1/2$.

Randomized weighted majority algorithm

- Task: online prediction of a binary sequence, *i.e.* sequentially forecast a value $y^{(t)} \in \{-1, 1\}$ at time t based on historical values $\{y^{(i)}\}_{i=1}^{(t-1)}$;
- Expert committee: an expert E_n is a man/woman who can make a prediction, $f_n^{(t)}$, using different strategies (algorithms/heuristics/data resources); assume there are N experts in the committee;
- Loss: $l(\hat{y}^{(t)}, y^{(t)}) = \mathbf{1}_{\hat{y}^{(t)} \neq y^{(t)}}$, $l(f_n^{(t)}, y^{(t)}) = \mathbf{1}_{f_n^{(t)} \neq y^{(t)}}$
- Weights update: if one expert E_n predicts wrongly, decrease its weight

$$w_n^{(t+1)} = (1 - \eta) w_n^{(t)} \tag{8}$$

where $\eta \leq 1/2$.

- Forecaster: $\hat{y}_t \sim \textbf{Bernoulli}\left(\frac{\sum_{n:f_{n,t}=1} w_n^{(t-1)}}{\sum_n w_n^{(t-1)}}, \frac{\sum_{n:f_{n,t}=-1} w_n^{(t-1)}}{\sum_n w_n^{(t-1)}}\right)$;

Analysis on randomized weighted majority algorithm

Theorem (cummulative loss bound using randomized weighted majority)

After T steps, $\hat{L}^{(T)} \leq (1+\eta) \min_{1 \in [1,N]} L_n^{(T)} + \frac{\ln N}{\eta}$

Analysis on randomized weighted majority algorithm

Theorem (cummulative loss bound using randomized weighted majority)

After T steps, $\hat{L}^{(T)} \leq (1+\eta)\min_{1\in[1,N]} L_n^{(T)} + \frac{\ln N}{\eta}$

Proof: Let $\Gamma^{(t)} = \sum_{n\in[1,N]} w_n^{(t)}$, then $\Gamma^{(1)} = N$.

At each time t, let $F^(t) = \frac{\sum_{n:f_n^{(t)}\neq y^{(t)}} w_n^{(t)}}{\sum_n w_n^{(t)}}$, then

$$\Gamma^{(t+1)} = \Gamma^{(t)}\Big(1 - F^{(t)} + F^{(t)}(1-\eta)\Big) = \Gamma^{(t)}(1 - F^{(t)}\eta) \tag{9}$$

therefore:

$$\Gamma^{(T+1)} = N\prod_{t=1}^{T}(1 - F^{(t)}\eta) \tag{10}$$

for any individual expert n : $w_n^{(T+1)} = (1-\eta)^{L_n^{(T)}}$,
since $w_n^{(T+1)} \leq \Gamma^{(T+1)} \Longrightarrow (1-\eta)^{L_n^{(T)}} \leq N\prod_{t=1}^{T}(1 - F^{(t)}\eta)$

Go beyond binary sequence and 0-1 loss

- Expert committee: an expert E_n is a man/woman who can make a prediction, $f_n^{(t)}$, using different strategies (algorithms/heuristics/data resources); assume there are N experts in the committee;
- Forecaster: $\hat{y}^{(t)} = \pi(f_1^{(t)}, f_2^{(t)}, \cdots, f_N^{(t)})$;

- Task: general online prediction, *i.e.* sequentially forecast a value $y^{(t)} \in \mathbb{R}$ at time t based on historical values $\{y^{(i)}\}_{i=1}^{(t-1)}$;
- Loss: a cost vector $[m_1^{(t)}, m_2^{(t)} i, \cdots, m_N^{(t)}]$ is incurred to experts' predictions at time t;
- Weights update: multiplicative manner

$$w_n^{(t+1)} = w_n^{(t)} g(m_n^{(t)}) \tag{11}$$

where $g(x)$ is a decreasing function w.r.t. x.

Revisit aggregation policy in binary case

Forecaster: $\hat{y}_t \sim \textbf{Bernoulli}\left(\frac{\sum_{n:f_{n,t}=1} w_n^{(t-1)}}{\sum_n w_n^{(t-1)}}, \frac{\sum_{n:f_{n,t}=-1} w_n^{(t-1)}}{\sum_n w_n^{(t-1)}}\right)$;

Let $p_n^{(t-1)} = \frac{w_n^{(t-1)}}{\sum_{n=1}^N w_n^{(t-1)}}$, then

$$\begin{aligned}
\mathbb{E}\left\{\hat{y}^{(t)}\right\} &= (-1) \cdot \sum_{n:f_{n,t}=-1} p_n^{(t-1)} + 1 \cdot \sum_{n:f_{n,t}=1} p_n^{(t-1)} \\
&= \sum_{n=1} f_n^{(t)} p_n^{(t-1)} \qquad (12) \\
&= \mathbb{E}_{p^{(t-1)}}\left\{f^{(t)}\right\}
\end{aligned}$$

A further randomized version:

$$\hat{y}^{(t)} = f_{n_\dagger}^{(t)} \text{ with } n_\dagger \sim [p_1^{(t-1)}, p_2^{(t-1)}, \cdots, p_N^{(t-1)}] \qquad (13)$$

Randomized weighted majority algorithm IN GENERAL

- Expert committee: an expert E_n is a man/woman who can make a prediction, $f_n^{(t)}$, using different strategies (algorithms/heuristics/data resources); assume there are N experts in the committee;

- Task: general online prediction, *i.e.* sequentially forecast a value $y^{(t)} \in \mathbb{R}$ at time t based on historical values $\{y^{(i)}\}_{i=1}^{(t-1)}$;
- Loss: a cost vector $[m_1^{(t)}, m_2^{(t)} i, \cdots, m_N^{(t)}]$ is incurred to experts' predictions at time t;
- Forecaster: $\hat{y}^{(t)} = f_{n_\dagger}^{(t)}$ with $n_\dagger \sim [p_1^{(t-1)}, p_2^{(t-1)}, \cdots, p_N^{(t-1)}]$
- Weights update: multiplicative manner

$$w_n^{(t+1)} = w_n^{(t)}(1 - \eta m_n^{(t)}) \tag{14}$$

where $\eta \leq 1/2$

Analysis

Theorem (Regret bound using randomized weighted majority)

When $m_n^{(t)} \in [-1, 1], \forall n, t$, *after* T *steps,*
$\hat{L}^{(T)} \leq \min_{1 \in [1,N]} L_n^{(T)} + \eta \sum_{t=1}^{T} |m_n^{(t)}|_1 + \frac{\ln N}{\eta}$

Regret bound : $\mathbf{R}^* \leq \eta T + \frac{\ln N}{\eta}$

Hedge algorithm

Weights update:

$$w_n^{(t+1)} = w_n^{(t)} \exp(-\eta m_n^{(t)}) \tag{15}$$

where $\eta \in [0, 1]$.

Theorem (Regret bound using randomized weighted majority)

When $m_n^{(t)} \in [-1, 1], \forall n, t$, after T steps,
$\hat{L}^{(T)} \leq \min_{1 \in [1,N]} L_n^{(T)} + \eta \sum_{t=1}^{T} |m_n^{(t)}|_1 + \frac{\ln N}{\eta}$

One-player game

- One player plays a game, at time t he takes one action $a^{(t)} = i \in [1, N]$, then the environment releases the cost for each action $\mathbf{m}^{(t)} = [m_1^{(t)}, m_2^{(t)}, \cdots, m_N^{(t)}]^\top, m_n^{(t)} \in [-1, 1]$,
- Note that the loss function $\mathbf{m}^{(t)}$ can change over time, i.e. the environment is changing.

One-player game

- One player plays a game, at time t he takes one action $a^{(t)} = i \in [1, N]$, then the environment releases the cost for each action $\mathbf{m}^{(t)} = [m_1^{(t)}, m_2^{(t)}, \cdots, m_N^{(t)}]^\top, m_n^{(t)} \in [-1, 1]$,
- Note that the loss function $\mathbf{m}^{(t)}$ can change over time, i.e. the environment is changing.
- if the player selects actions by the probabilities $\mathbf{p}^{(t)}$ computed by randomized weighted majority and update them accordingly, then

$$\sum_{t=1}^{T} \langle \mathbf{m}^{(t)}, \mathbf{p}^{(t)} \rangle \leq \sum_{t=1}^{T} m_i^{(t)} + \eta T + \frac{\ln n}{\eta} \tag{16}$$

Two-player game

Two-person zero-sum: two players ($K = 2$) play a game which is defined by a $N \times N$ cost matrix $\mathbf{C}$ (N is the number of possible actions for players) , where each entry c_{ij} defines the loss to the row player when the row player takes the action $i \in [1, N]$ and the column player takes the action $j \in [1, N]$.

An example cost matrix:
$$\begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{bmatrix} \end{array}$$

The row player's goal is to minimize its loss while the objective of the column player is to maximize it

Nash Equilibrium

- if the row player chooses his action from a distribution $\mathbf{p}$, then the most adversary action the column player should take is

$$j := \arg\max_{j \in [1,N]} \mathbb{E}_{i \sim \mathbf{p}} \{c_{ij}\} \tag{17}$$

Nash Equilibrium

- if the row player chooses his action from a distribution **p**, then the most adversary action the column player should take is

$$j := \arg\max_{j \in [1,N]} \mathbb{E}_{i \sim \mathbf{p}}\{c_{ij}\} \tag{17}$$

- similarly, if column player use a distribution **q** , then the row player should take:

$$i := \arg\min_{i \in [1,N]} \mathbb{E}_{j \sim \mathbf{q}}\{c_{ij}\} \tag{18}$$

Nash Equilibrium

- if the row player chooses his action from a distribution $\mathbf{p}$, then the most adversary action the column player should take is

$$j := \arg \max_{j \in [1,N]} \mathbb{E}_{i \sim \mathbf{p}} \{c_{ij}\} \tag{17}$$

- similarly, if column player use a distribution $\mathbf{q}$, then the row player should take:

$$i := \arg \min_{i \in [1,N]} \mathbb{E}_{j \sim \mathbf{q}} \{c_{ij}\} \tag{18}$$

von Neumann's min-max theorem

$\min_{\mathbf{p}} \max_j \mathbb{E}_{i \sim \mathbf{p}} \{c_{ij}\} = \max_{\mathbf{q}} \max_i \mathbb{E}_{j \sim \mathbf{q}} \{c_{ij}\} = \lambda^*$

-
- no player has an incentive of changing his strategy (distribution) if the player does not change his, i.e. every player is happy about current status;

Hanan's algorithm for two-player zero-sum game

Follow the perturbed leading expert

Forecaster: select the action $i^{(t)} = \arg\min_{i \in [1,N]} \left\{ L_i^{(t-1)} + \tau_i \right\}$ where τ_i are randomly sampled from $[0, 1/\epsilon]$

Hanan's algorithm for two-player zero-sum game

Follow the perturbed leading expert

Forecaster: select the action $i^{(t)} = \arg\min_{i \in [1,N]} \left\{ L_i^{(t-1)} + \tau_i \right\}$ where τ_i are randomly sampled from $[0, 1/\epsilon]$

Equivalence to exponential weighted majority

In Hanan's algorithm, when $\tau_i = \frac{1}{\eta} \ln \ln \frac{1}{u_i}$, where $u_i \sim [0,1]$,

$$Pr[i^{(t)} = j] = \frac{e^{-\eta L_j^{(t)}}}{\sum_{k=1}^{N} e^{-\eta L_k^{(t)}}}$$

after $T = \frac{4 \ln n}{\epsilon}$ iterations, the algorithm can converge to a $\tilde{\mathbf{p}}$ which yield $\lambda^* + \epsilon$

Arbitrary convex loss function

- a convex function family on $\mathbf{p}$: $g(\mathbf{p})$
- the whole objective function is presented sequentially: $G(\cdot) = g^{(1)}(\cdot) + g^{(2)}(\cdot) + \cdots g^{(T)}(\cdot)$

Arbitrary convex loss function

- a convex function family on $\mathbf{p}$: $g(\mathbf{p})$
- the whole objective function is presented sequentially: $G(\cdot) = g^{(1)}(\cdot) + g^{(2)}(\cdot) + \cdots g^{(T)}(\cdot)$
- re-design a loss $\mathbf{m}^{(t)} = \frac{1}{\rho}\Delta f^{(t)}(\mathbf{p})$
- $\sum_{t=1}^{T} f^{(t)}(\mathbf{p}^{(t)}) - \min_{\mathbf{p}} G(\mathbf{p}) \leq 2\rho\sqrt{\ln nT}$

Arbitrary convex loss function

- a convex function family on $\mathbf{p}$: $g(\mathbf{p})$
- the whole objective function is presented sequentially: $G(\cdot) = g^{(1)}(\cdot) + g^{(2)}(\cdot) + \cdots g^{(T)}(\cdot)$
- re-design a loss $\mathbf{m}^{(t)} = \frac{1}{\rho}\Delta f^{(t)}(\mathbf{p})$
- $\sum_{t=1}^{T} f^{(t)}(\mathbf{p}^{(t)}) - \min_{\mathbf{p}} G(\mathbf{p}) \leq 2\rho\sqrt{\ln nT}$
- applications
  1. online portfolio management: $g^{(t)}(\mathbf{p}) = \log(-\mathbf{p}^\top \Delta \mathbf{v}^{(t)})$
  2. online learning algorithms: $g^{(t)}(\mathbf{p}) = ||y^{(t)} - \mathbf{p}^\top \mathbf{x}^{(t)}||_2^2$

Bandit Optimization in metric spaces

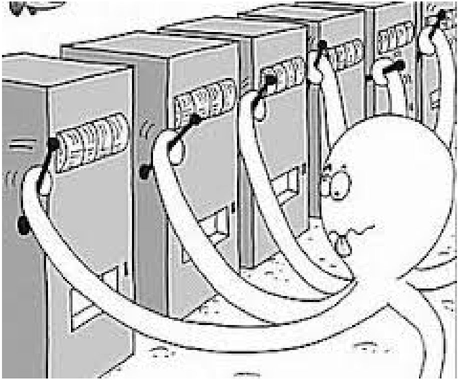
Observe only one cost

- One player plays a game, at time t he takes one action $a^{(t)} = i \in [1, N]$, then the environment releases the cost for only the selected action $m^{(t)}_{a^{(t)}}$,

Observe only one cost

- One player plays a game, at time t he takes one action $a^{(t)} = i \in [1, N]$, then the environment releases the cost for only the selected action $m^{(t)}_{a^{(t)}}$,
- playing games in a Bandit setting falls into classic exploitation v.s. exploration dilemma

Multi-armed bandit problem



Upper confidence bound

- at each time t, play the arm $i^{(t)}$ by

$$i^{(t)} = \arg\max_{i \in [1,N]} \left\{ \mathbf{avg.}[m_i] + \sqrt{\frac{2 \ln t}{T_i}} \right\} \tag{19}$$

Upper confidence bound

- at each time t, play the arm $i^{(t)}$ by

$$i^{(t)} = \arg\max_{i \in [1,N]} \left\{ \mathbf{avg.}[m_i] + \sqrt{\frac{2 \ln t}{T_i}} \right\} \tag{19}$$

- Regret bound:

$$\mathbf{R}^{(t)*} \leq \left[8 \sum_{i:\mu_i < \mu^*} \frac{\ln t}{\mu^* - \mu_i} \right] + (1 + \frac{\pi^2}{3})(\sum_{i=1}^{N} (\mu^* - \mu_i)) \tag{20}$$

where μ_i denotes the true expected reward for arm i, and μ^* denotes the best one.

Contextual bandit optimization

- in classic multi-armed bandit problem, all arms are independent;

Contextual bandit optimization

- in classic multi-armed bandit problem, all arms are independent;
- in many practical applications, there exists a context beneath arms, e.g. representation of arms is in a metric space;

Response surface optimization

- response surface optimization is an extension of contextual bandit optimization to infinite number of arms;

Response surface optimization

- response surface optimization is an extension of contextual bandit optimization to infinite number of arms;
- model the response of arms using a smooth surface function f

Response surface optimization

- response surface optimization is an extension of contextual bandit optimization to infinite number of arms;
- model the response of arms using a smooth surface function f
- a demo mimicking Fashionflow style to understand surface-response optimization;

Response surface optimization

- response surface optimization is an extension of contextual bandit optimization to infinite number of arms;
- model the response of arms using a smooth surface function f
- a demo mimicking Fashionflow style to understand surface-response optimization;
- a most general case, Lipschit continuity: if

$$d_x(\mathbf{x}_1, \mathbf{x}_2) \leq L \cdot d_f(f(\mathbf{x}_1), f(\mathbf{x}_2)) \tag{21}$$

then we say f is a L-Lipschit continuous function.

Gaussian process

Definition

A Gaussian process (GP) defines is a collection of random variables, any finite number of of which have joint Gaussian distribution.

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{22}$$

Gaussian process

Definition

A Gaussian process (GP) defines is a collection of random variables, any finite number of of which have joint Gaussian distribution.

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{22}$$

joint Gaussian distribution

$$\left[\begin{array}{c} f(\mathbf{X}_{train}) \\ f(\mathbf{x}_{test}) \end{array}\right] \sim \left(\left[\begin{array}{c} \mu(\mathbf{X}_{train}) \\ \mu(\mathbf{x}_{test}) \end{array}\right], \left[\begin{array}{cc} k(X_{train}, X_{train}), & k(\mathbf{X}_{train}, \mathbf{x}_{test}) \\ k(\mathbf{X}_{train}, x_{test})^\top, & k(\mathbf{x}_{test}, \mathbf{x}_{test}) \end{array}\right]\right) \tag{23}$$

Gaussian process, cont.

Conditional probability

$$f(\mathbf{x}_{test})|f(\mathbf{X}_{train}) \sim \mathcal{N}\left(\mu_{pos}(\mathbf{x}_{test}), k_{pos}(\mathbf{x}_{test}, \mathbf{x}')\right) \tag{24}$$

Gaussian process, cont.

Conditional probability

$$f(\mathbf{x}_{test})|f(\mathbf{X}_{train}) \sim \mathcal{N}\left(\mu_{pos}(\mathbf{x}_{test}), k_{pos}(\mathbf{x}_{test}, \mathbf{x}')\right) \quad (24)$$

Posterior Gaussian process

$$f(\mathbf{x})|\mathcal{D}_{train} \sim \mathcal{GP}\left(\mu_{pos}(\mathbf{x}), k_{pos}(\mathbf{x}, \mathbf{x}')\right) \quad (25)$$

where $\mu_{pos}(\mathbf{x}) = \mu(\mathbf{x}) + k(\mathbf{X}_{train}, \mathbf{x})^\top k(X_{train}, X_{train})(f(\mathbf{X}_{train}) - \mathbf{X}_{train})$
and $k_{pos}(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - k(\mathbf{X}_{train}, \mathbf{x})^\top k(\mathbf{X}_{train}, \mathbf{X}_{train})^{-1} k(\mathbf{X}_{train}, \mathbf{x}')$.

Gaussian process bandit optimization

Kernel function

squared exponential kernel: $k(\mathbf{x}, \mathbf{x}') = \exp(-\frac{||\mathbf{x}-\mathbf{x}'||}{2l^2})$

Information gain

the informativeness of a set of points $A \in \mathcal{X}$ for learning f is defined as:

$$I(\mathbf{y}_A; \mathbf{f}_A) = H(\mathbf{y}_A) - H(\mathbf{y}_A|\mathbf{f}_A) \tag{26}$$

Maximum information gain after T iterations

$\gamma^{(T)} = \max_{A\in\mathcal{X}:|A|=T} I(\mathbf{y}_A; \mathbf{f}_A)$

- maximum information gain basically reflect the complexity of f;
- for RBF kernel, $\gamma^{(T)} = \mathcal{O}\left((\log T)^{d+1}\right)$

Upper confidence bound

- at each time t, chose:

$$\mathbf{x}^{(t)} = \arg\max_{\mathbf{x}\in\mathcal{X}} \mu^{(t-1)}(\mathbf{x}) + \kappa^{(t)}\sigma^{(t-1)}(\mathbf{x}) \tag{27}$$

where $\kappa^{(t)} = 2B + 300\gamma^{(t)}\log^3(t/\delta)$, $||f||^2 \leq B$

Upper confidence bound

- at each time t, chose:

$$\mathbf{x}^{(t)} = \arg\max_{\mathbf{x}\in\mathcal{X}} \mu^{(t-1)}(\mathbf{x}) + \kappa^{(t)}\sigma^{(t-1)}(\mathbf{x}) \tag{27}$$

  where $\kappa^{(t)} = 2B + 300\gamma^{(t)}\log^3(t/\delta)$, $||f||^2 \leq B$

- Regret bound:

$$Pr\left\{\mathbf{R}^{(T)*} \leq \sqrt{\frac{8}{\log(1+\sigma^{-1})} T\beta^{(T)}\gamma^{(T)}}\right\} \geq 1-\delta \tag{28}$$

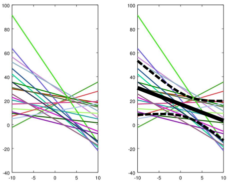
- Regret bound complexity: $\mathbf{R}^{(T)*} = \mathcal{O}\left(\sqrt{T(\log T)^{d+1}}\right)$.

Regression with ensemble models

- Gaussian process bandit optimization is an instance of Bayesian optimization;
- any function f with certain smoothness assumption and uncertainty measurement can be used as a pseudo Bayesian optimization;

Regression with ensemble models

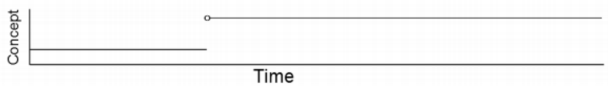
- Gaussian process bandit optimization is an instance of Bayesian optimization;
- any function f with certain smoothness assumption and uncertainty measurement can be used as a pseudo Bayesian optimization;
- one example: an ensemble of linear functions



  1. smoothness assumption: average of linear functions
  2. uncertainty measurement: std.of linear functions
- advantage: can better exploit task-relevant features instead of isotropic length-scale in kernel function

Concept drift in online decision

What do people do when the concept drifts ?



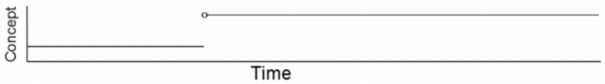
(a) Sudden concept drift.

(b) Gradual concept drift.

Assume that the concept drift can be successfully detected,

- forget the data before the new concept in learning algorithms;

What do people do when the concept drifts ?



(a) Sudden concept drift.

(b) Gradual concept drift.

Assume that the concept drift can be successfully detected,

- forget the data before the new concept in learning algorithms;
- change a new learning algorithm (strategy)

Stability and adaptivity when forgetting data

Definition

A learning algorithm $\mathcal{A}$ has error stability β_n with respect to the loss function l if

$$\forall Z_n \in \mathcal{X}^n, \forall i \in \{1, \cdots, n\} |\mathbb{E}\{l(\mathcal{A}_{Z_n})\} - \mathbb{E}\{l(\mathcal{A}_{Z_n}^{-i})\}|$$

Examples

for k-NN, SVM, support vector regression and ridge regression, $\beta_n = \mathcal{O}(\frac{1}{n})$

Detect concept drift

- Essentially, the detection has been conducted via hypothesis test sequentially.
  $I(\mathcal{A}_{\{\mathbf{x}_1,\cdots,\mathbf{x}_{n-t}\}}, \underbrace{\{\mathbf{x}_{n-t+1},\cdots,\mathbf{x}_n\}}_{\text{time window}}) \Longrightarrow$
  null hypothesis v.s.alternative hypothesis
- Can work well with good hyper-parameter settings.

Temporal-spatial kernel

- Typical analysis of Gaussian process bandit optimization is in a stationary environment.

Temporal-spatial kernel

- Typical analysis of Gaussian process bandit optimization is in a stationary environment.
- When involving the concept drift, the kernel can be defined in a temporal-spatial manner, i.e. scale down the kernel values of old instances;

Temporal-spatial kernel

- Typical analysis of Gaussian process bandit optimization is in a stationary environment.
- When involving the concept drift, the kernel can be defined in a temporal-spatial manner, i.e. scale down the kernel values of old instances;
- Regret bound:

$$Pr\left\{\mathbf{R}^{(T)*} \leq \sqrt{\frac{8}{\log(1+\sigma^{-1})} T\beta^{(T)}\hat{\gamma}^{(T)}}\right\} \geq 1-\delta \tag{29}$$

Concept drift in expert committee

1. A expert committee
   - an expert which uses some side information;
   - an expert which uses short-memory of instances;
   - an expert which uses whole sequence of instances;
   - an expert which can detect concept drifts and forget old data;
2. concept drift == best expert shift

Adaptive regret and Fixed-share algorithm

- regular regret:
  $R^{(t)*} = \max_{n\in[1,N]} R_n^{(t)} = \hat{L}^{(t)} - \min_{n\in[1,N]} L_n^{(t)}$

Adaptive regret and Fixed-share algorithm

- regular regret:
  $R^{(t)*} = \max_{n\in[1,N]} R_n^{(t)} = \hat{L}^{(t)} - \min_{n\in[1,N]} L_n^{(t)}$
- adaptive regret:
  $\bar{R}^{(t)*} = \sup_{[t1,t2]\subseteq[1,t]} \left\{ \hat{L}^{[t1,t2]} - \min_{n\in[1,N]} L_n^{[t1,t2]} \right\}$

Adaptive regret and Fixed-share algorithm

- regular regret:
  $R^{(t)*} = \max_{n\in[1,N]} R_n^{(t)} = \hat{L}^{(t)} - \min_{n\in[1,N]} L_n^{(t)}$
- adaptive regret:
  $\bar{R}^{(t)*} = \sup_{[t1,t2]\subseteq[1,t]} \left\{ \hat{L}^{[t1,t2]} - \min_{n\in[1,N]} L_n^{[t1,t2]} \right\}$
- weight-share algorithm:

Fixed-share algorithm

Weight update if one expert E_n predicts wrongly, decrease its weight

$$\hat{w}_n^{(t+1)} = w_n^{(t)} \exp^{(-\eta m_n^{(t)})} \tag{30}$$

followed by a weight-share update:

$$w_n^{(t+1)} = (1-\alpha)\hat{w}_n^{(t+1)} + \frac{\alpha}{N-1} \sum_{m\neq n}^{N} \hat{w}_m^{(t+1)} \tag{31}$$

New sku proposal plans in FahsionFlow

Expert committee interpretation

by considering each expert is a sku proposer,

- fully observed feedback for all experts;
- can flexibly define cost function ;
- can flexibly add diverse experts;
- regret bound complexity: $\mathcal{O}(\sqrt{T \ln N})$
- clear understanding of performance in non-stationary environments;
- can be computational expensive by maintaining many experts;
- there exist some gap between sequential proposal and search

Bandit interpretation within ensemble learning

- partially observed feedback only on the selected sku;
- no explicit loss function;
- non-trivial analysis of the information gains for different types of functions;
- regret bound complexity: $\mathcal{O}\left(\sqrt{T(\log T)^{d+1}}\right)$
- difficult to analyze information gain in non-stationary environments;
- GP complexity $\mathcal{O}(n^3)$; also can be computational expensive by maintaining many models;
- it stands closer to the nature of the *Search* task;

Connections

if $\kappa = 0$ for all t, then Bandit setting == Expert committee setting in the aggregation policy:

A reminder

$$\hat{y}^{(t)} = \sum_{n=1}^{N} p_n^{(t-1)} f_n^{(t)}$$

Plans and outlooks

Under expert committee interpretation framework:

- define more diverse proposal experts (strategies);
- design an informative loss function;
    1. loss function for classifiers

    $$m_n^{(t)} = \exp^{-p_n^{(t)}(y^{(t)})}$$

    where $y^{(n)} = \{-1, 1\}$ and $p_n^{(t)}$ denotes the classification probability of E_n
    2. loss function for regressor (ranker)

    ??

Plans and outlooks

Under expert committee interpretation framework:

- define more diverse proposal experts (strategies);
- design an informative loss function;
  1. loss function for classifiers

     $$m_n^{(t)} = \exp^{-p_n^{(t)}(y^{(t)})}$$

     where $y^{(n)} = \{-1, 1\}$ and $p_n^{(t)}$ denotes the classification probability of E_n
  2. loss function for regressor (ranker)

     $$??$$

- enable some experts with concept drift detection and data forgetting mechanism;
- track the best expert using weight-share algorithm

Quantitative evaluation

- Expert committee interpretation:
  cumulative loss of the forecaster or regret;
- Bandit interpretation:
  use pseudo targets (wishlist);

References: Nicolo Cesa-Bianchi and Gabor Lugosi. 2006. Prediction, Learning, and Games. Cambridge University Press, New York, NY, USA.

The End