Random Permutation Online Isotonic Regression

Wojciech Kotłowski   Wouter M. Koolen   Alan Malek

Motivation

? distant goal: online isotonic regression on **partial orders**

. . . Current solution for **linear orders** does not scale

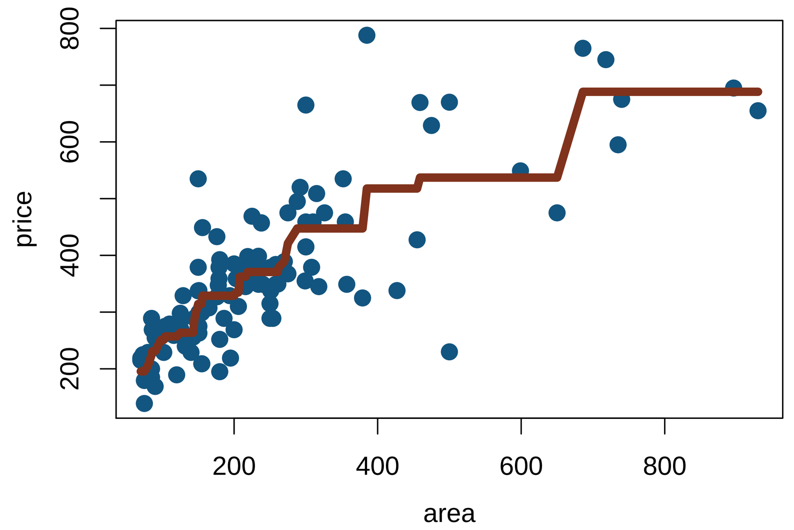
! New model and algorithms for **linear case**

(Offline) isotonic regression

Fit an *isotonic* (non-decreasing) function to the data:

$$f^* = \operatorname*{argmin}_{\text{isotonic } f} \sum_{t=1}^{T} (y_t - f(x_t))^2$$

isotonic regression function



Pool Adjacent Violators Algorithm (PAVA) [Ayer et al., 1955]:

- Iteratively merge data into blocks until no violator of isotonic constraints exists
- Assign to data in each block the average of their labels y_t
- Blocks correspond to *level sets* of f^*

Online Isotonic Regression

At trial $t = 1 \ldots T$:
  Adversary chooses covariate x_t
  Learner predicts $\widehat{y}_t \in [0,1]$
  Adversary reveals label $y_t \in [0,1]$
  Learner suffers squared loss $(y_t - \widehat{y}_t)^2$

- Regret: $\sum_{t=1}^{T} (y_t - \widehat{y}_t)^2 - \underbrace{\min_{\text{isotonic } f} \sum_{t=1}^{T} (y_t - f(x_t))^2}_{\text{total loss of offline IR function}}$
- Linear regret without restriction on x_t

Random Permutation Model

Random Permutation Model

- Adversary chooses data instances $x_1 < \ldots < x_T, y_1, \ldots, y_T$
- Sample UAR a permutation $\sigma = (\sigma_1, \ldots, \sigma_T)$ of $\{1, \ldots, T\}$
- Round t: covariate x_{σ_t}, true label y_{σ_t}, and loss $(\widehat{y}_{\sigma_t} - y_{\sigma_t})^2$

Learner minimizes *expected regret*,

$$R_T := \mathbb{E}_\sigma \left[\sum_{t=1}^{T} (y_{\sigma_t} - \widehat{y}_{\sigma_t})^2 \right] - L_T^* = \sum_{t=1}^{T} r_t,$$

where $r_t := \mathbb{E}_\sigma \left[(y_{\sigma_t} - \widehat{y}_{\sigma_t})^2 - L_t^* + L_{t-1}^* \right]$ is the *per-round regret* and $L_t^* = L^*(\{(x_{\sigma_1}, y_{\sigma_1}), \ldots, (x_{\sigma_t}, y_{\sigma_t})\})$ is the optimal loss of the first t labeled instances.

Leave-one-out loss

With Data $D = \{(x_1, y_1), \ldots, (x_t, y_t)\}$, the ℓoo of a t round game is

$$\ell oo_t(D) := \frac{1}{t} \left(\left(\sum_{i=1}^{t} (y_i - \widehat{y}_i(D \setminus (x_i, y_i)))^2 \right) - L^*(D) \right).$$

Lemma 1. *$r_t(D) \leq \ell oo_t(D)$ for any t and any data set* $D = \{(x_1, y_1), \ldots, (x_t, y_t)\}$.

Lower bound

Adversarial lower bound [Kotłowski, Koolen, and Malek, 2016] applies to random permutation model: $\ell oo_t = \Omega(t^{-2/3})$.

Matching bounds

Theorem 2. *There is an algorithm for the random-permutation model with excess leave-one-out loss $\ell oo_t = \tilde{O}(t^{-\frac{2}{3}})$ and hence expected regret $R_T \leq \sum_t \tilde{O}(t^{-\frac{2}{3}}) = \tilde{O}(T^{\frac{1}{3}})$, which matches the lower bound of $\ell oo_t = \Omega\left(t^{-2/3}\right)$.*

Caveat: algorithm is not efficient (on partial orders)!

Forward Algorithms

Two observations:

- PAVA is efficient and generalizes to partial orders
- *Follow The Leader* algorithms are common in practice

Forward Algorithm: To predict at x_t, imagine $y_t' \in [0,1]$, compute f^* on $\{(x_1, y_1) \ldots (x_{t-1}, y_{t-1})\} \cup \{(x_t, y_t')\}$, and play $\widehat{y}_t = f^*(x_t)$.

Forward Algorithm Examples

- IR-Int: Compute f^* on past data. Predict with average of f^* at nearest x_i.
- Interpolation $\widehat{y}_i = \lambda_i \widehat{y}_i^0 + (1 - \lambda_i)\widehat{y}_i^1$ (where $\widehat{y}^0$ and $\widehat{y}^1$ are „plug-in $y_t' = 0$" and „plug-in $y_t' = 1$")
- Last step minimax:

$$\widehat{y}_i = \operatorname*{argmin}_{\widehat{y} \in [0,1]} \max_{y_i \in [0,1]} \left\{ (\widehat{y} - y_i)^2 - L^*(\boldsymbol{y}) \right\}$$

- IVAP predictors [Vovk et al., 2015]:

$$\widehat{y}_i^{\log} = \frac{\widehat{y}_i^1}{\widehat{y}_i^1 + 1 - \widehat{y}_i^0}, \qquad \widehat{y}_i^{\text{Brier}} = \frac{1 + (\widehat{y}_i^0)^2 - (1 - \widehat{y}_i^1)^2}{2}$$

Regret Bounds

Theorem 3. *Any forward algorithm has $\ell oo_t = O\left(\sqrt{\frac{\log t}{t}}\right)$.*

Lemma 4. *The IR-Int algorithm has $\ell oo_t = \Omega(t^{-\frac{1}{2}})$ when run on*

$$\underbrace{10000}_{1/5} \; \underbrace{11000}_{2/5} \; \underbrace{11100}_{3/5} \; \underbrace{11110}_{4/5} \; \underbrace{11111}_{5/5}.$$

Heavy-γ

Parameters: Weight $c > 0$ and label $\gamma \in [0,1]$.
Algorithm: To predict at x_t

- Compute isotonic regression f' on **weighted** dataset

$$D' := \{(x_s, y_s, 1) | 1 \leq s < t\} \cup \{(x_t, \gamma, c)\}$$

- Predict $y_t = f'(x_t)$

Efficient weighted algorithms available [Kyng et al., 2015].

Tuning Heavy-γ

Any fixed label γ works. We like $\gamma = 1$.
(Not all adaptive labels work. Fixed point + lower bound.)

Theorem 5. *Heavy-γ has sub-optimal ℓoo_t loss unless $c = \Theta(t^{\frac{1}{3}})$.*

Conjecture 6. *Heavy-γ with weight $c = \Theta(t^{\frac{1}{3}})$ has $\ell oo_t = \tilde{O}(t^{-\frac{2}{3}})$.*