Online isotonic regression

Wojciech Kotłowski

Joint work with:
Wouter Koolen (CWI, Amsterdam)
Alan Malek (MIT)

Poznań University of Technology
06.06.2017

Outline

Outline

Motivation I – house pricing

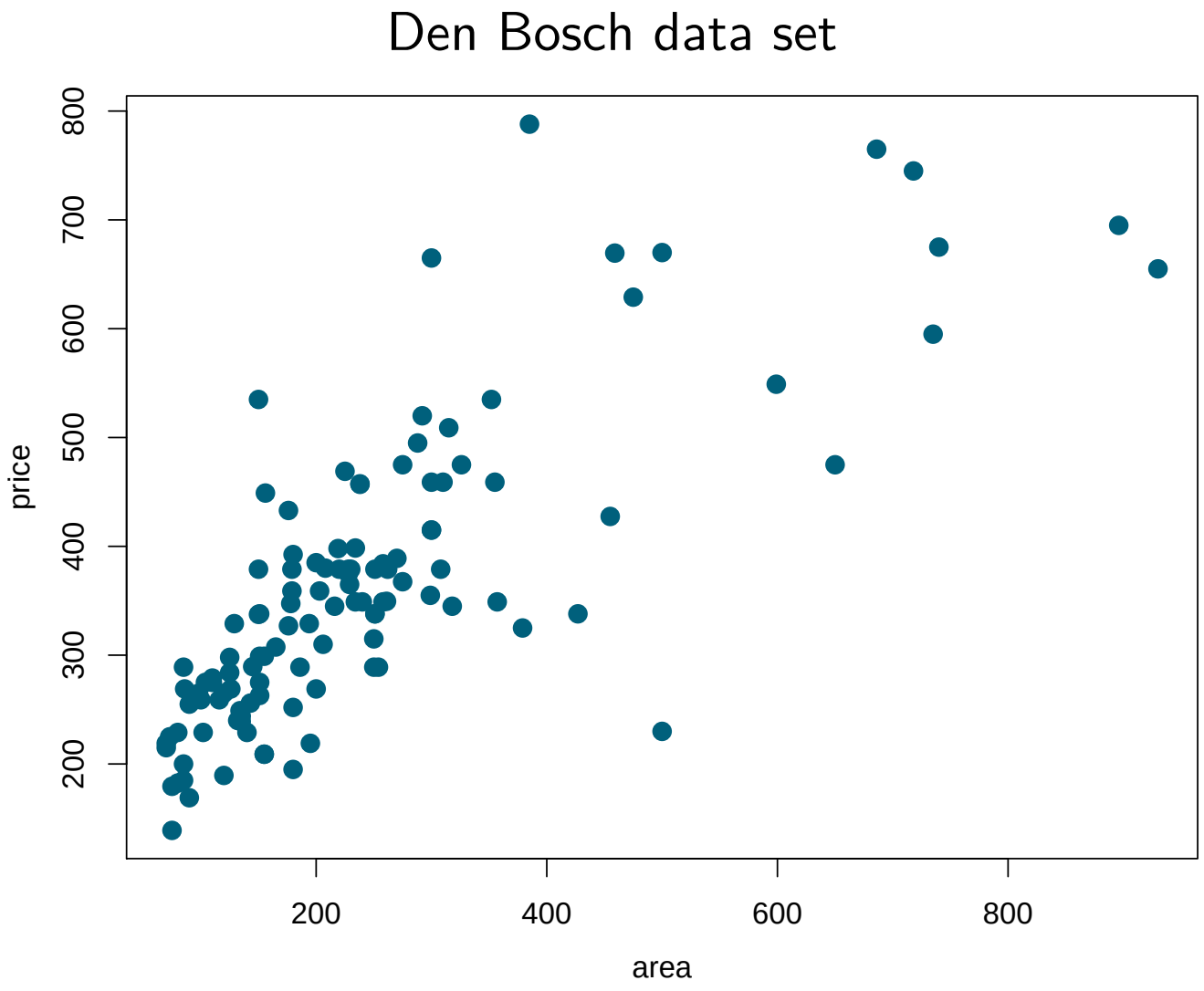
Assess the selling price of a house based on its attributes.



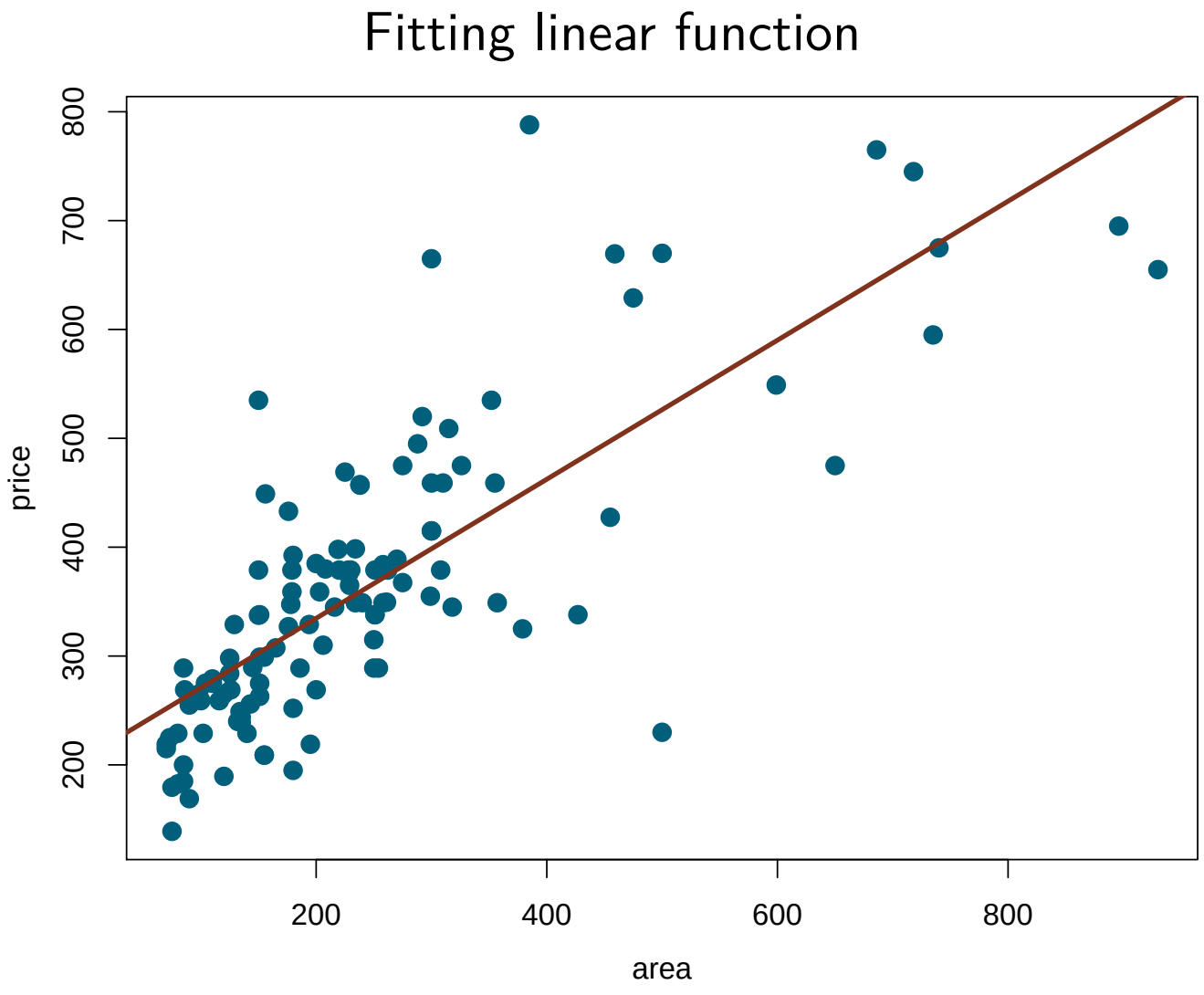




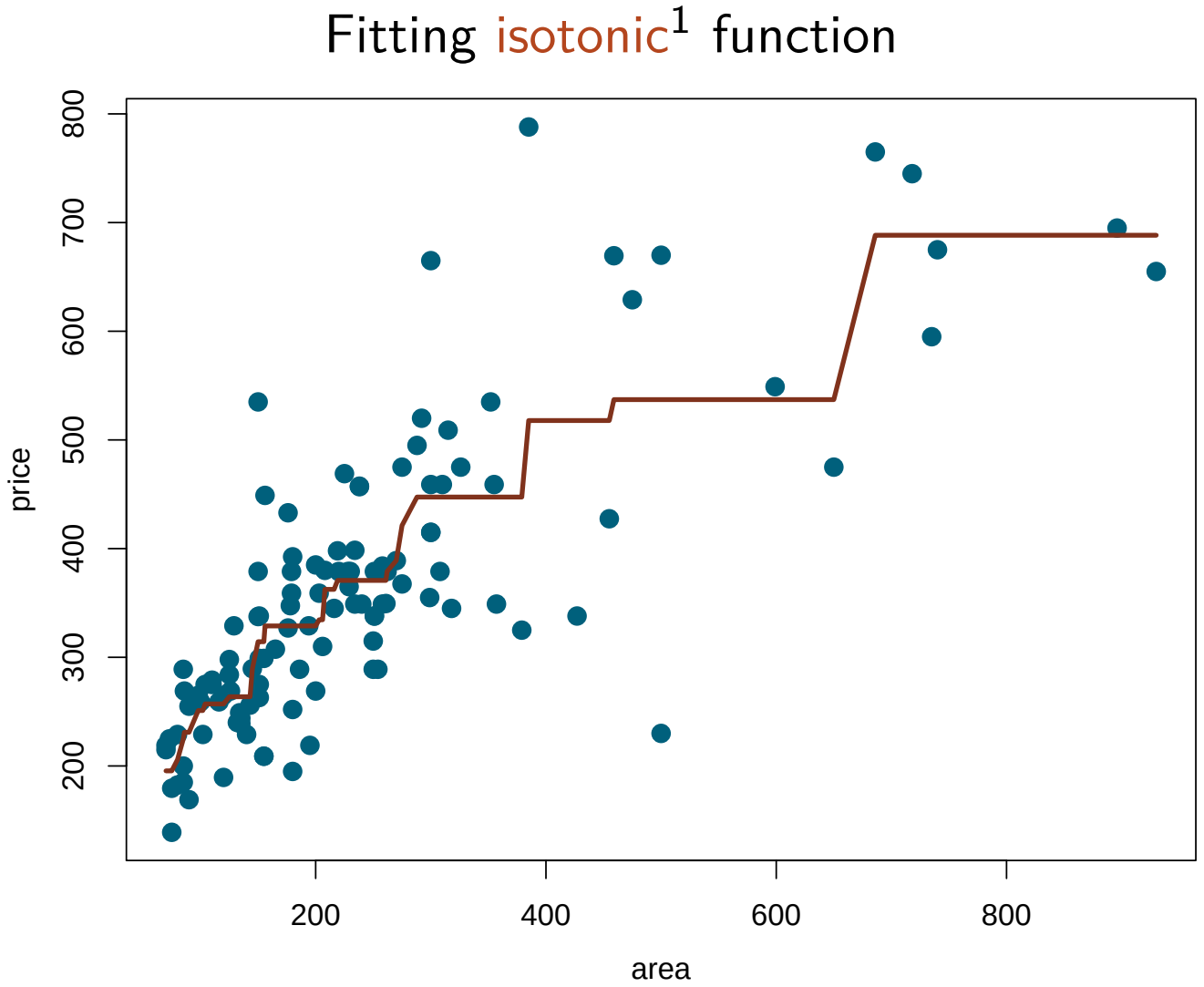
Motivation I – house pricing



Motivation I – house pricing

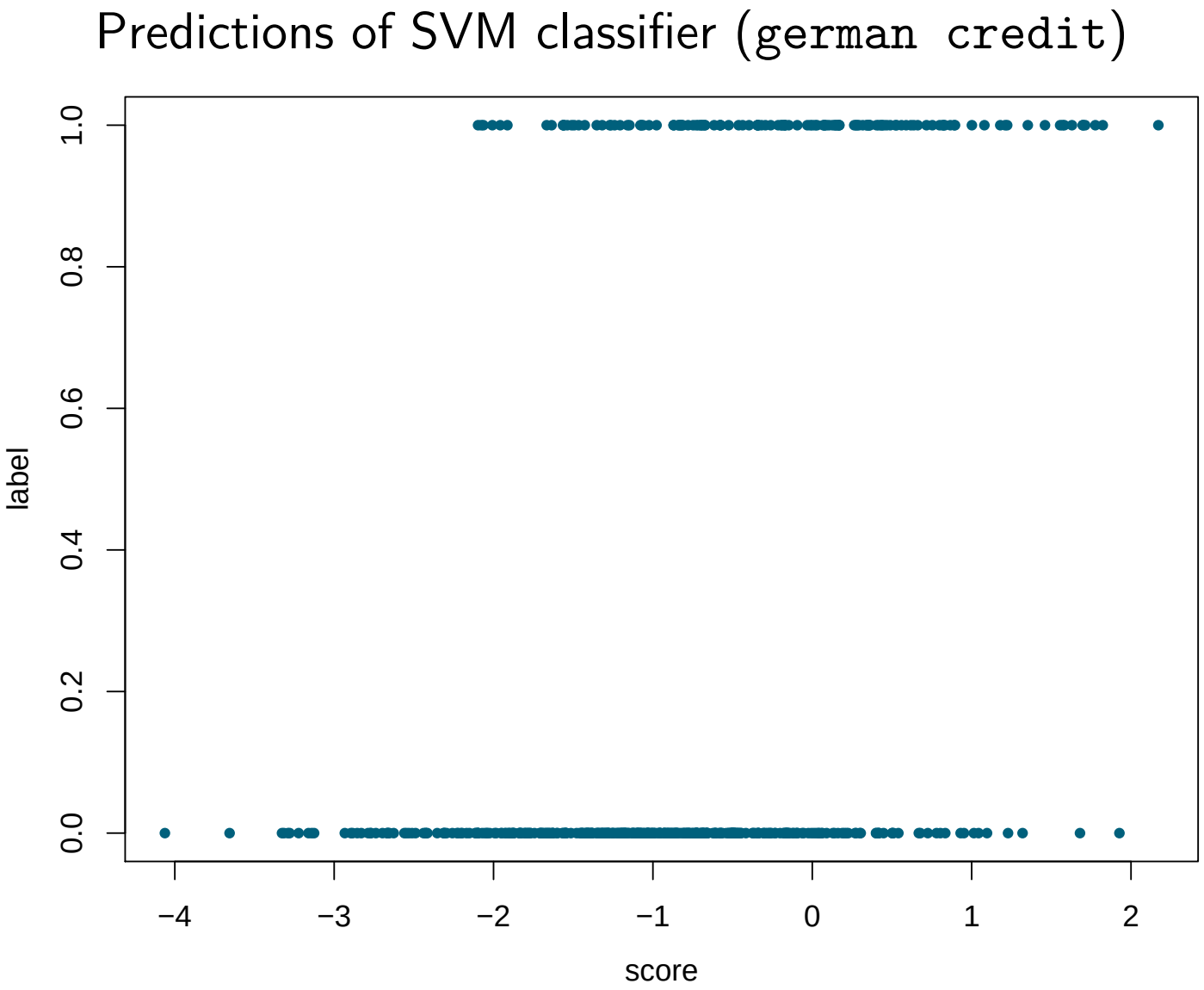


Motivation I – house pricing



[1] isotonic – non-decreasing, order-preserving

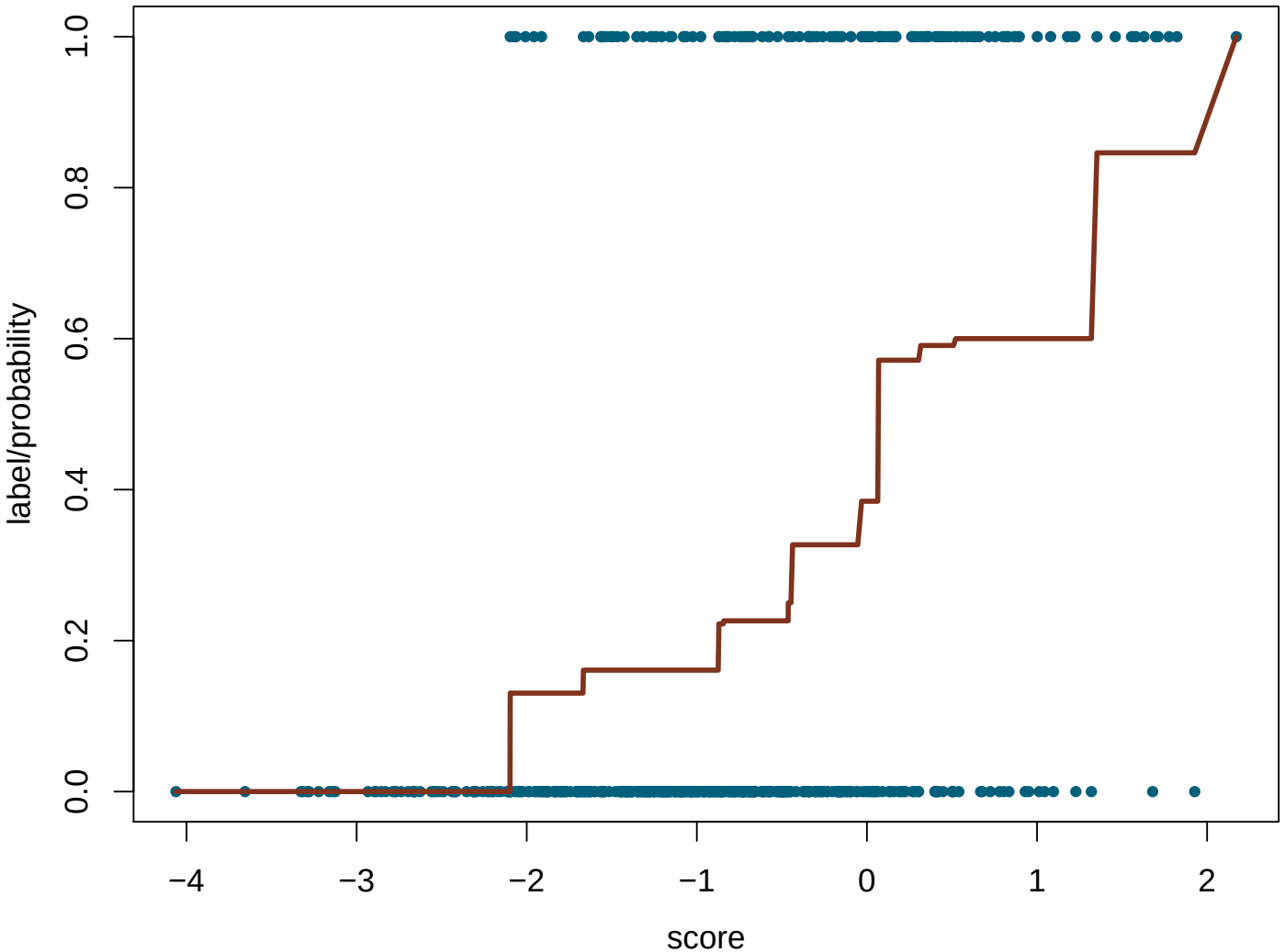
Motivation II – predicting good probabilities



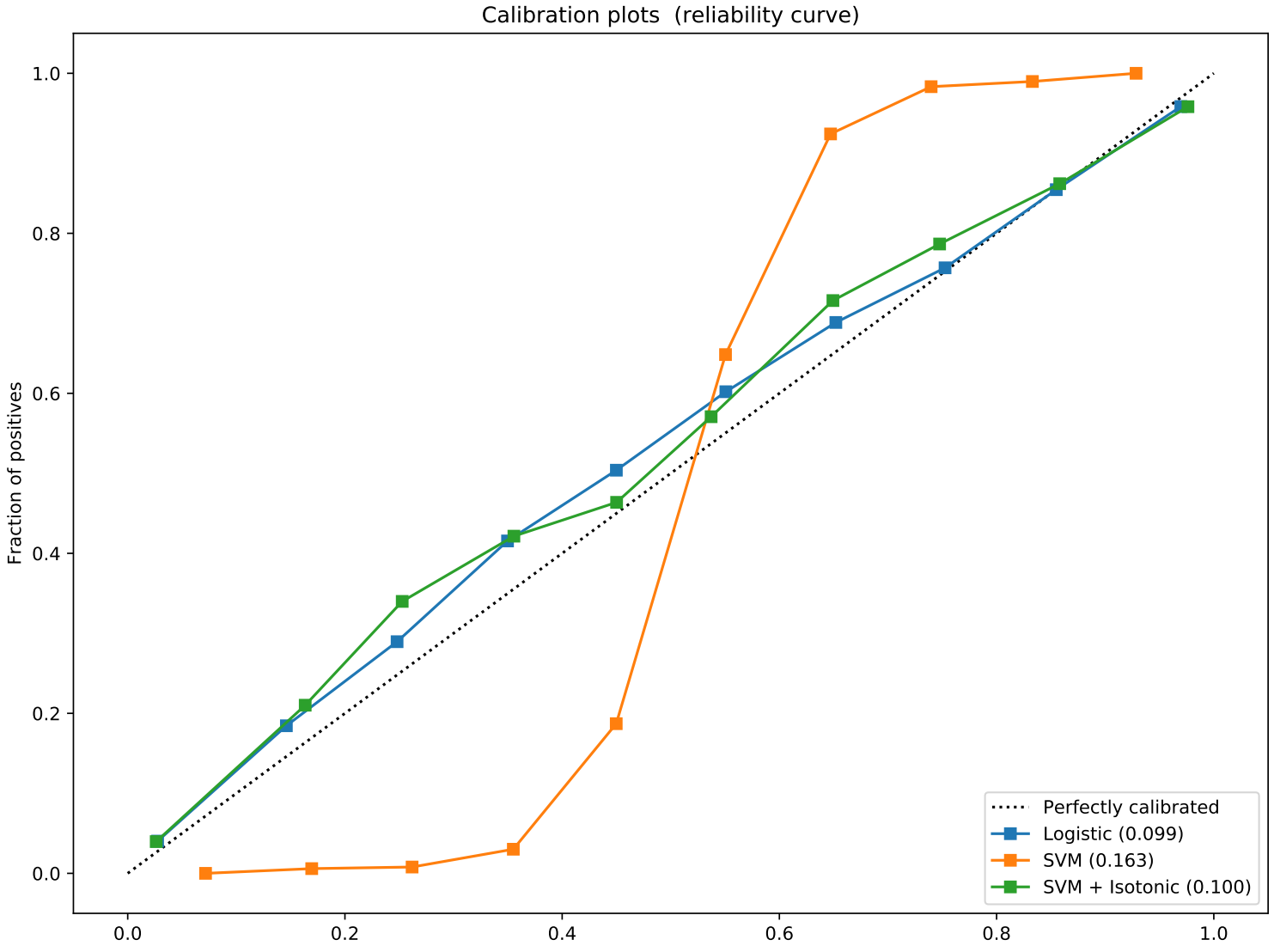
Can we turn score values into conditional probabilities $P(y|x)$?

Motivation II – predicting good probabilities

Fitting isotonic function to the labels [Zadrozny & Elkan, 2002]

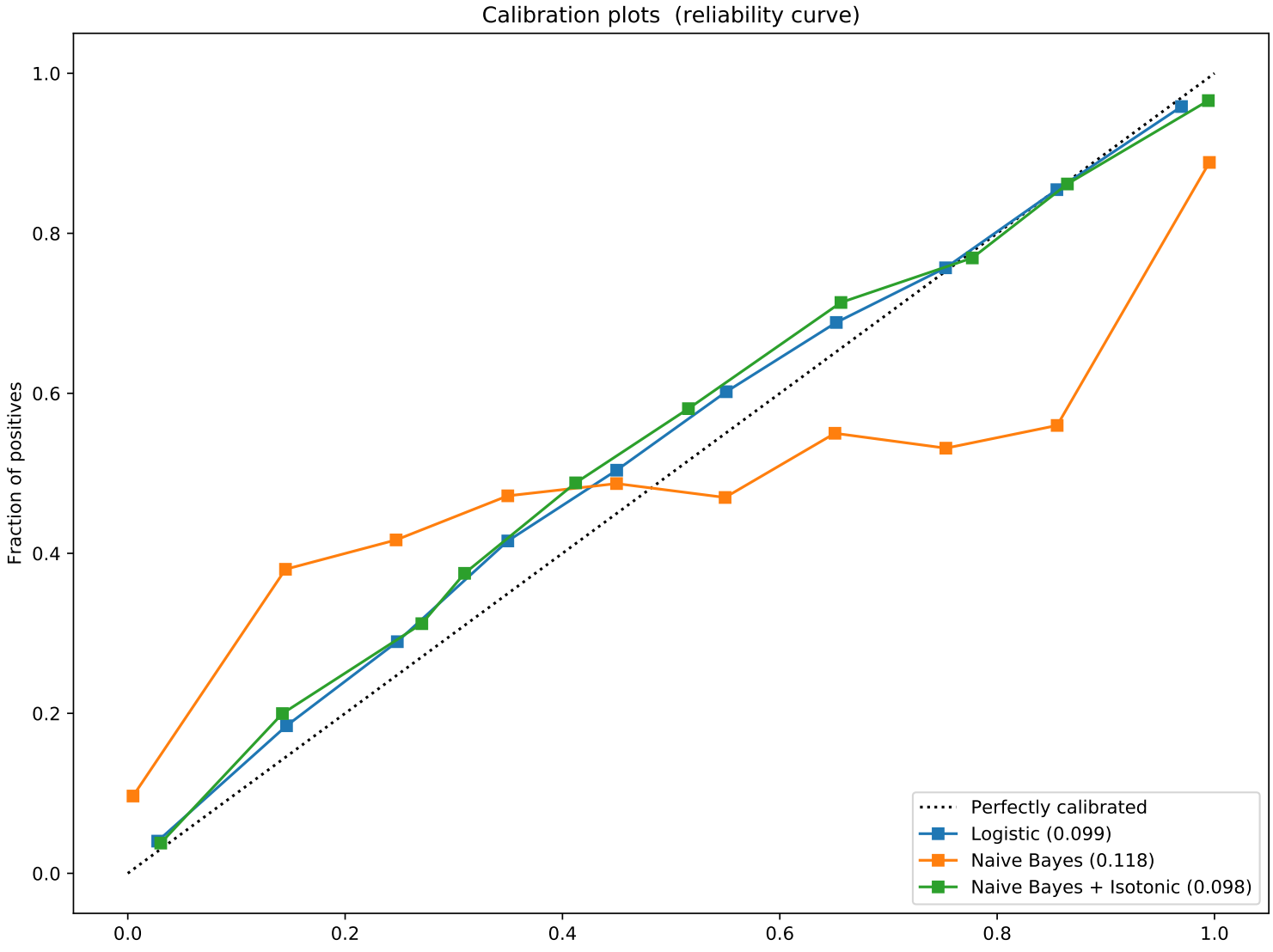


Motivation II – predicting good probabilities



(generated by a script from `scikit-learn.org`)

Motivation II – predicting good probabilities



(generated by a script from `scikit-learn.org`)

Outline

Isotonic regression

Definition

Fit an isotonic (monotonically increasing) function to the data.

Extensively studied in statistics [Ayer et al., 55; Brunk, 55; Robertson et al., 98].

Numerous applications:

- Biology, medicine, psychology, etc.
- Multicriteria decision support.
- Hypothesis tests under order constraints.
- Multidimensional scaling.
- Machine learning: probability calibration, ROC analysis.

Isotonic regression

Isotonic regression

Definition

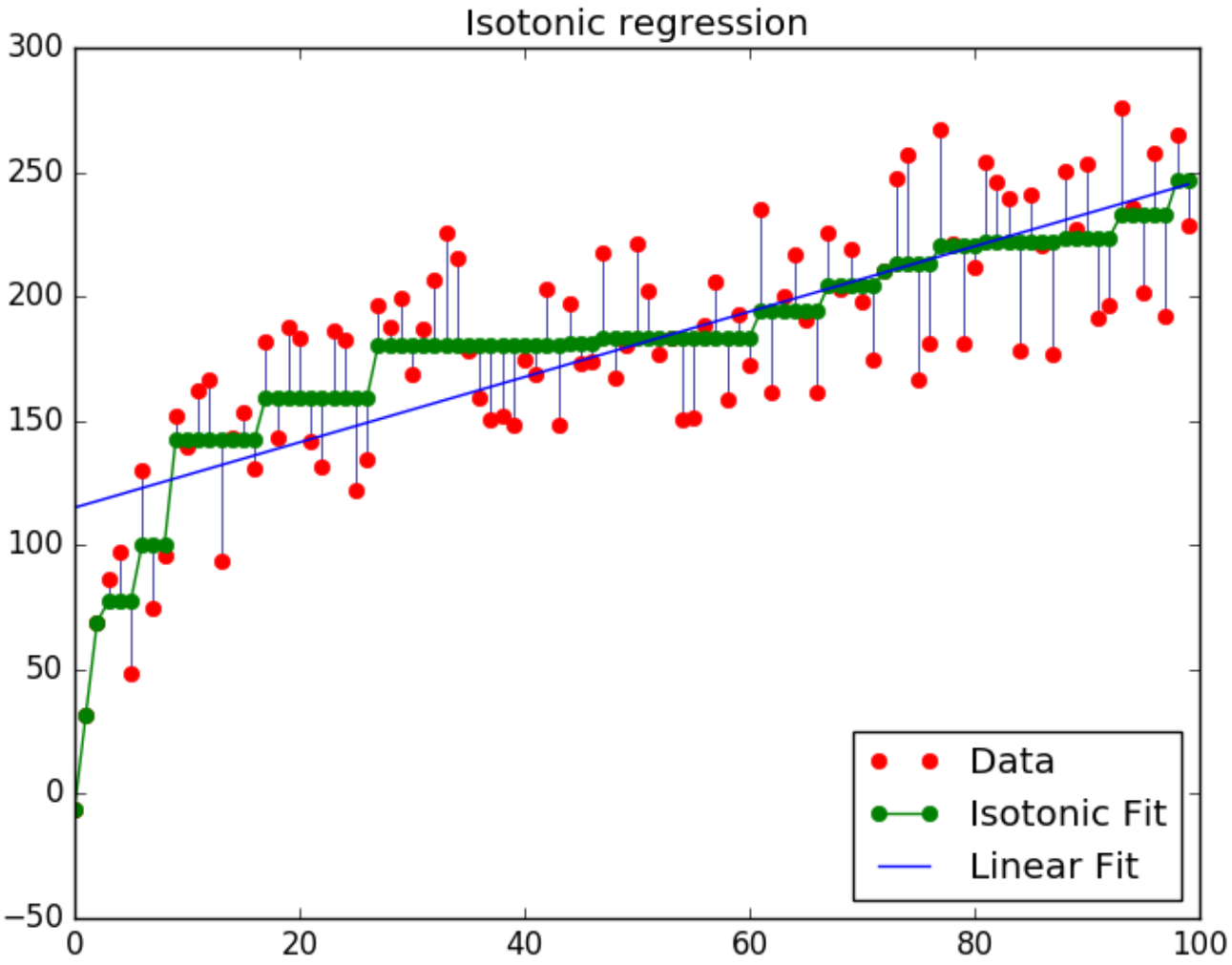
Given data $\{(x_t, y_t)\}_{t=1}^{T} \subset \mathbb{R} \times \mathbb{R}$, find isotonic (nondecreasing) $f^*: \mathbb{R} \to \mathbb{R}$, which minimizes squared error over the labels:

$$\begin{aligned} \min_f : \quad & \sum_{t=1}^{T} (y_t - f(x_t))^2, \\ \text{subject to} : \quad & x_t \geq x_q \implies f(x_t) \geq f(x_q), \quad q, t \in \{1, \ldots, T\}. \end{aligned}$$

The optimal solution f^* is called isotonic regression function.

What only matters are values $f(x_t)$, $t = 1, \ldots, T$.

Isotonic regression example



(source: `scikit-learn.org`)

Properties of isotonic regression

- Depends on instances (x) only through their order relation.
- Only defined at points $\{x_1, \ldots, x_T\}$.
  - Often extended to $\mathbb{R}$ by linear interpolation.
- Piecewise constants (splits the data into level sets).
- Self-averaging property: the value of f^* in a given level set equals the average of labels in that level set. For any v:

$$v = \frac{1}{|S_v|} \sum_{t \in S_v} y_t \qquad \text{where} \quad S_v = \{t \colon f^*(x_t) = v\}.$$

- If $y_t \in [a, b]$ for all t, then $f^*(x_t) \in [a, b]$ for all t.

Isotonic regression gives calibrated probabilities

Definition

Let $y \in \{0, 1\}$. A probability estimator $\widehat{p}$ of y is calibrated if

$$\mathbb{E}[y|\widehat{p} = v] = v$$

Isotonic regression gives calibrated probabilities

Definition

Let $y \in \{0, 1\}$. A probability estimator $\widehat{p}$ of y is calibrated if

$$\mathbb{E}[y|\widehat{p} = v] = v$$

Fact

For binary labels, isotonic regression f^* is a calibrated probability estimator on the data set.

Proof: Let $S_v = \{t : f^*(x_t) = v\}$. By self-averaging:

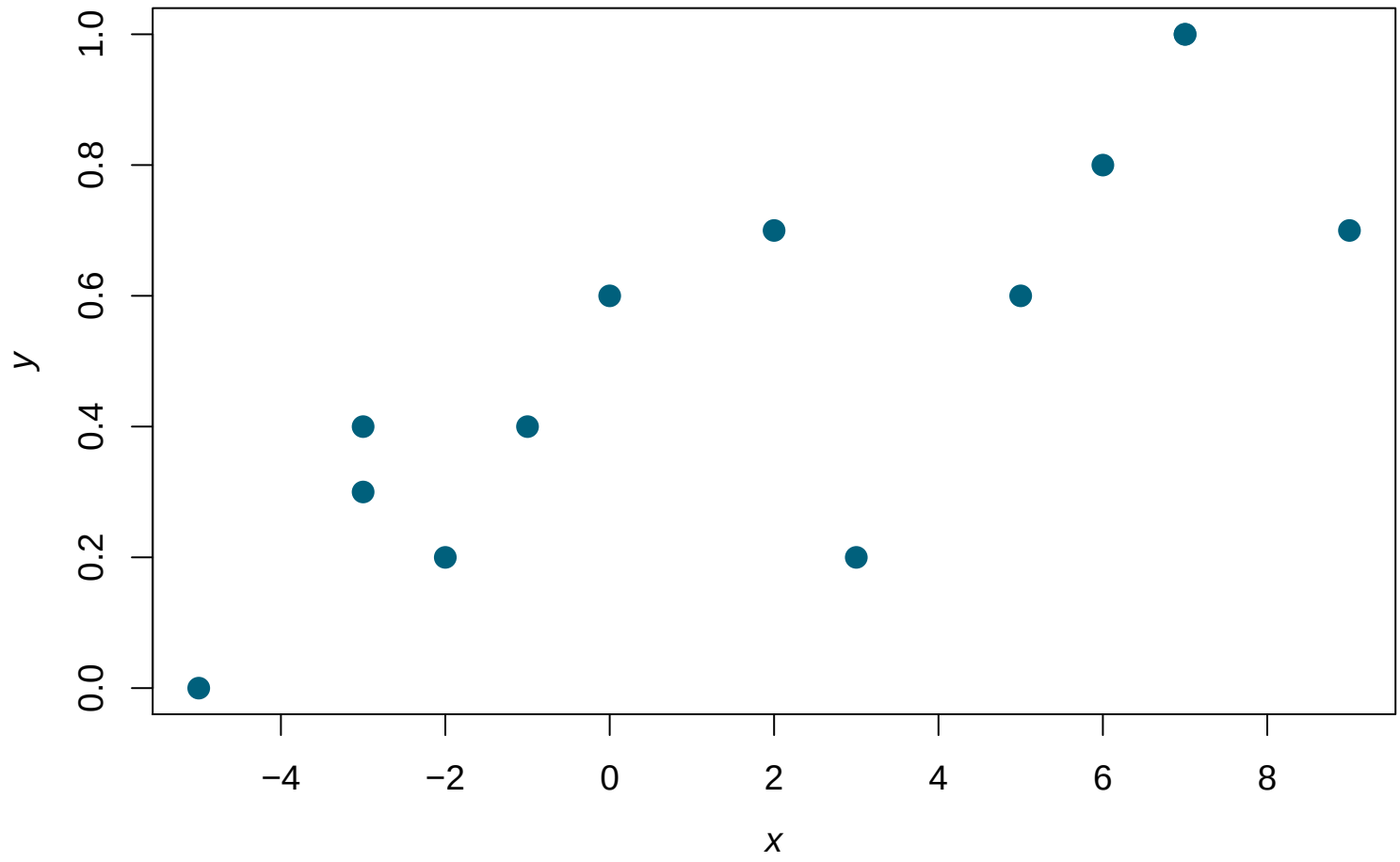
$$\mathbb{E}[y|f^*(x) = v] \;=\; \frac{1}{|S_v|} \sum_{t \in S_v} y_t \;=\; v.$$

Pool Adjacent Violators Algorithm (PAVA)

- Iterative merging of of data points into blocks until no violators of isotonic constraints exist.
- The values assigned to each block is the average over labels in this block.
- The final assignments to blocks corresponds to the level sets of isotonic regression.
- Works in linear $O(T)$ time, but requires the data to be sorted.

PAVA: example

x	7	-1	-2	9	2	0	6	3	-3	5	-3	7	-5
y	1	0.4	0.2	0.7	0.7	0.6	0.8	0.2	0.3	0.6	0.4	1	0



PAVA: example

Step 1: Sort the data in the increasing order of x.

x	7	-1	-2	9	2	0	6	3	-3	5	-3	7	-5
y	1	0.4	0.2	0.7	0.7	0.6	0.8	0.2	0.3	0.6	0.4	1	0

$\Downarrow \quad \Downarrow \quad \Downarrow$

x	-5	-3	-3	-2	-1	0	2	3	5	6	7	7	9
y	0	0.4	0.3	0.2	0.4	0.6	0.7	0.2	0.6	0.8	1	1	0.7

PAVA: example

Step 2: Split the data into blocks $B_1, \ldots, B_r$, such that points with the same x_t fall into the same block.

Assign value f_i to each block ($i = 1, \ldots, r$) which is the average of labels in this block.

x	-5	-3	-3	-2	-1	0	2	3	5	6	7	7	9
y	0	0.4	0.3	0.2	0.4	0.6	0.7	0.2	0.6	0.8	1	1	0.7

$\Downarrow \quad \Downarrow \quad \Downarrow$

block	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}
data	$\{1\}$	$\{2,3\}$	$\{4\}$	$\{5\}$	$\{6\}$	$\{7\}$	$\{8\}$	$\{9\}$	$\{10\}$	$\{11,12\}$	$\{13\}$
f_i	0	0.35	0.2	0.4	0.6	0.7	0.2	0.6	0.8	1	0.7

PAVA: example

Step 3: While there exists a violator, i.e. a pair of blocks B_i, B_{i+1} such that $f_i > f_{i+1}$:

- Merge B_i and B_{i+1} and assign a weighted average:

$$f_i = \frac{|B_i| f_i + |B_{i+1}| f_{i+1}}{|B_i| + |B_{i+1}|}.$$

block	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}
data	$\{1\}$	$\{2,3\}$	$\{4\}$	$\{5\}$	$\{6\}$	$\{7\}$	$\{8\}$	$\{9\}$	$\{10\}$	$\{11,12\}$	$\{13\}$
f_i	0	0.35	0.2	0.4	0.6	0.7	0.2	0.6	0.8	1	0.7

$\Downarrow \quad \Downarrow \quad \Downarrow$

block	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}
data	$\{1\}$	$\{2,3,4\}$	$\{5\}$	$\{6\}$	$\{7\}$	$\{8\}$	$\{9\}$	$\{10\}$	$\{11,12\}$	$\{13\}$
f_i	0	0.3	0.4	0.6	0.7	0.2	0.6	0.8	1	0.7

PAVA: example

Step 3: While there exists a violator, i.e. a pair of blocks B_i, B_{i+1} such that $f_i > f_{i+1}$:

- Merge B_i and B_{i+1} and assign a weighted average:

$$f_i = \frac{|B_i| f_i + |B_{i+1}| f_{i+1}}{|B_i| + |B_{i+1}|}.$$

block	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}
data	$\{1\}$	$\{2,3,4\}$	$\{5\}$	$\{6\}$	$\{7\}$	$\{8\}$	$\{9\}$	$\{10\}$	$\{11,12\}$	$\{13\}$
f_i	0	0.3	0.4	0.6	0.7	0.2	0.6	0.8	1	0.7

$\Downarrow \quad \Downarrow \quad \Downarrow$

block	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9
data	$\{1\}$	$\{2,3,4\}$	$\{5\}$	$\{6\}$	$\{7,8\}$	$\{9\}$	$\{10\}$	$\{11,12\}$	$\{13\}$
f_i	0	0.3	0.4	0.6	0.45	0.6	0.8	1	0.7

PAVA: example

Step 3: While there exists a violator, i.e. a pair of blocks B_i, B_{i+1} such that $f_i > f_{i+1}$:

- Merge B_i and B_{i+1} and assign a weighted average:

$$f_i = \frac{|B_i| f_i + |B_{i+1}| f_{i+1}}{|B_i| + |B_{i+1}|}.$$

block	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9
data	$\{1\}$	$\{2,3,4\}$	$\{5\}$	$\{6\}$	$\{7,8\}$	$\{9\}$	$\{10\}$	$\{11,12\}$	$\{13\}$
f_i	0	0.3	0.4	0.6	0.45	0.6	0.8	1	0.7

$\Downarrow \quad \Downarrow \quad \Downarrow$

block	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
data	$\{1\}$	$\{2,3,4\}$	$\{5\}$	$\{6,7,8\}$	$\{9\}$	$\{10\}$	$\{11,12\}$	$\{13\}$
f_i	0	0.3	0.4	0.5	0.6	0.8	1	0.7

PAVA: example

Step 3: While there exists a violator, i.e. a pair of blocks B_i, B_{i+1} such that $f_i > f_{i+1}$:

- Merge B_i and B_{i+1} and assign a weighted average:

$$f_i = \frac{|B_i| f_i + |B_{i+1}| f_{i+1}}{|B_i| + |B_{i+1}|}.$$

block	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
data	$\{1\}$	$\{2,3,4\}$	$\{5\}$	$\{6,7,8\}$	$\{9\}$	$\{10\}$	$\{11,12\}$	$\{13\}$
f_i	0	0.3	0.4	0.5	0.6	0.8	1	0.7

$\Downarrow \quad \Downarrow \quad \Downarrow$

block	B_1	B_2	B_3	B_4	B_5	B_6	B_7
data	$\{1\}$	$\{2,3,4\}$	$\{5\}$	$\{6,7,8\}$	$\{9\}$	$\{10\}$	$\{11,12,13\}$
f_i	0	0.3	0.4	0.5	0.6	0.8	0.9

No more violators – finished.

PAVA: example

Reading out the solution.

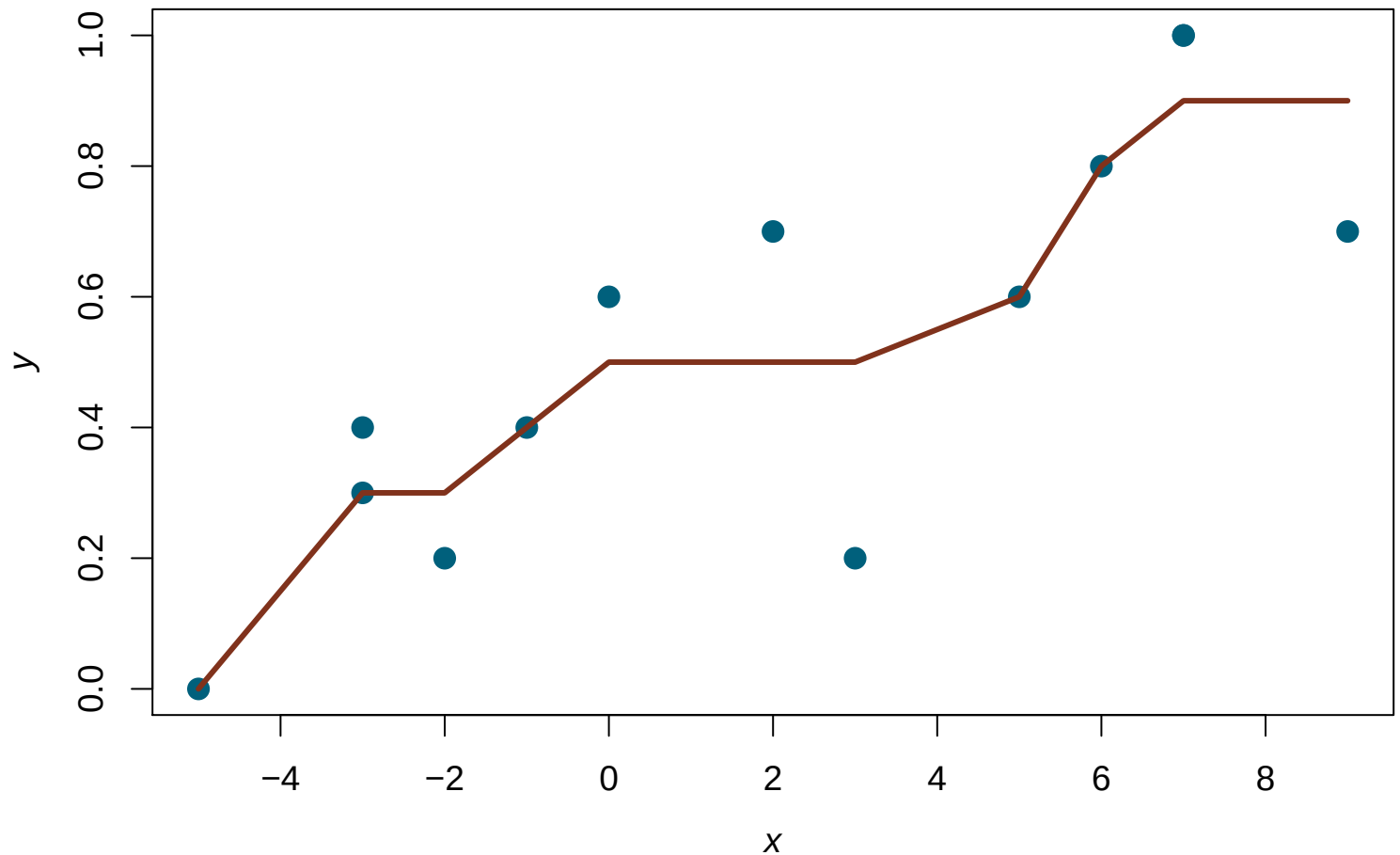
block	B_1	B_2	B_3	B_4	B_5	B_6	B_7
data	$\{1\}$	$\{2,3,4\}$	$\{5\}$	$\{6,7,8\}$	$\{9\}$	$\{10\}$	$\{11,12,13\}$
f_i	0	0.3	0.4	0.5	0.6	0.8	0.9

$\Downarrow \quad \Downarrow \quad \Downarrow$

x	-5	-3	-3	-2	-1	0	2	3	5	6	7	7	9
y	0	0.4	0.3	0.2	0.4	0.6	0.7	0.2	0.6	0.8	1	1	0.7
f^*	0	0.3	0.3	0.3	0.4	0.5	0.5	0.5	0.6	0.8	0.9	0.9	0.9

PAVA: example

x	-5	-3	-3	-2	-1	0	2	3	5	6	7	7	9
y	0	0.4	0.3	0.2	0.4	0.6	0.7	0.2	0.6	0.8	1	1	0.7
f^*	0	0.3	0.3	0.3	0.4	0.5	0.5	0.5	0.6	0.8	0.9	0.9	0.9



Generalized isotonic regression

Definition

Given data $\{(x_t, y_t)\}_{t=1}^T \subset \mathbb{R} \times \mathbb{R}$, find isotonic $f^* : \mathbb{R} \to \mathbb{R}$ which minimizes:

$$\min_{\text{isotonic } f} \quad \sum_{t=1}^{T} \Delta(y_t, f(x_t)).$$

Squared loss $(y_t - f(x_t))^2$ replaced with general loss $\Delta(y_t, f(x_t))$.

Generalized isotonic regression

Definition

Given data $\{(x_t, y_t)\}_{t=1}^T \subset \mathbb{R} \times \mathbb{R}$, find isotonic $f^* : \mathbb{R} \to \mathbb{R}$ which minimizes:

$$\min_{\text{isotonic } f} \quad \sum_{t=1}^{T} \Delta(y_t, f(x_t)).$$

Squared loss $(y_t - f(x_t))^2$ replaced with general loss $\Delta(y_t, f(x_t))$.

Theorem [Robertson et al., 1998]

All loss functions of the form:

$$\Delta(y, z) = \Psi(y) - \Psi(z) - \Psi'(z)(y - z)$$

for some strictly convex Ψ result in the same isotonic regression function f^*.

Generalized isotonic regression – examples

$$\Delta(y, z) = \Psi(y) - \Psi(z) - \Psi'(z)(y - z)$$

Squared function $\Psi(y) = y^2$:

$$\Delta(y, z) \;=\; y^2 - z^2 - 2f(y - z) \;=\; (y - z)^2 \qquad \text{(squared loss)}.$$

Entropy $\Psi(y) = -y \log y - (1 - y) \log(1 - y)$, $y \in [0, 1]$

$$\Delta(y, z) \;=\; -\, y \log z - (1 - y) \log(1 - z) \qquad \text{(cross-entropy)}.$$

Negative logarithm $\Psi(y) = -\log y$, $y > 0$

$$\Delta(y, z) \;=\; \frac{y}{z} - \log \frac{y}{z} \qquad \text{(Itakura-Saito distance / Burg entropy)}.$$

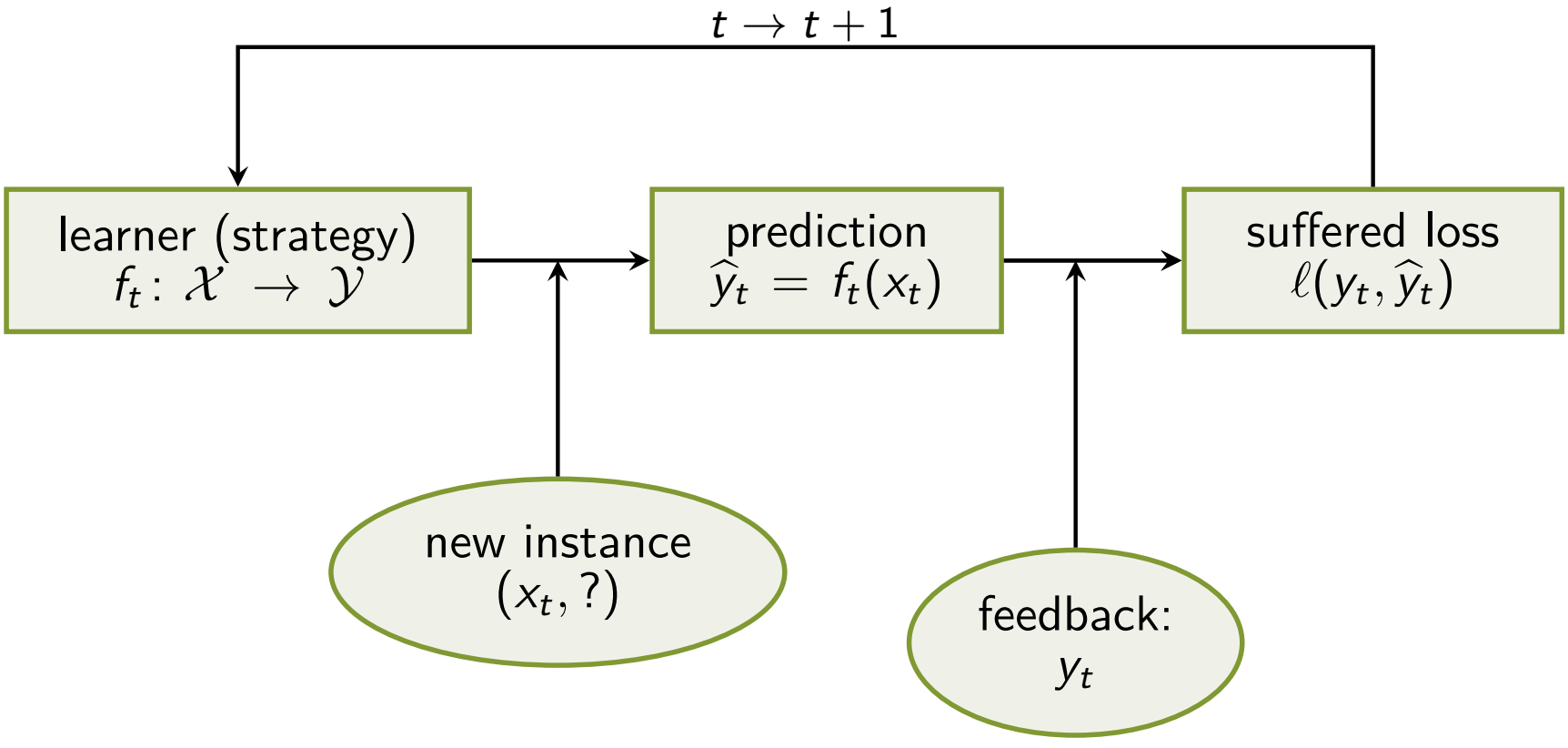
Outline

Online learning framework

A theoretical framework for the analysis of online algorithms.

- Learning process by its very nature is incremental.
- Avoids stochastic (e.g., i.i.d.) assumptions on the data sequence, designs algorithms which work well for any data.
- Meaningful performance guarantees based on observed quantities: regret bounds.

Online learning framework



Online learning framework

Set of strategies (actions) $\mathcal{F}$; known loss function ℓ.

Learner starts with some initial strategy (action) f_1.

For $t = 1, 2, \ldots$:

1. Learner observes instance x_t.
2. Learner predicts with $\widehat{y}_t = f_t(x_t)$.
3. The environment reveals outcome y_t.
4. Learner suffers loss $\ell(y_t, \widehat{y}_t)$.
5. Learner updates its strategy $f_t \to f_{t+1}$.

Online learning framework

The goal of the learner is to be close to the best f in hindsight.

Cumulative loss of the learner:

$$\widehat{L}_T = \sum_{t=1}^{T} \ell(y_t, \widehat{y}_t).$$

Cumulative loss of the best strategy f in hindsight:

$$L_T^* = \min_{f \in \mathcal{F}} \sum_{t=1}^{T} \ell(y_t, f(x_t)).$$

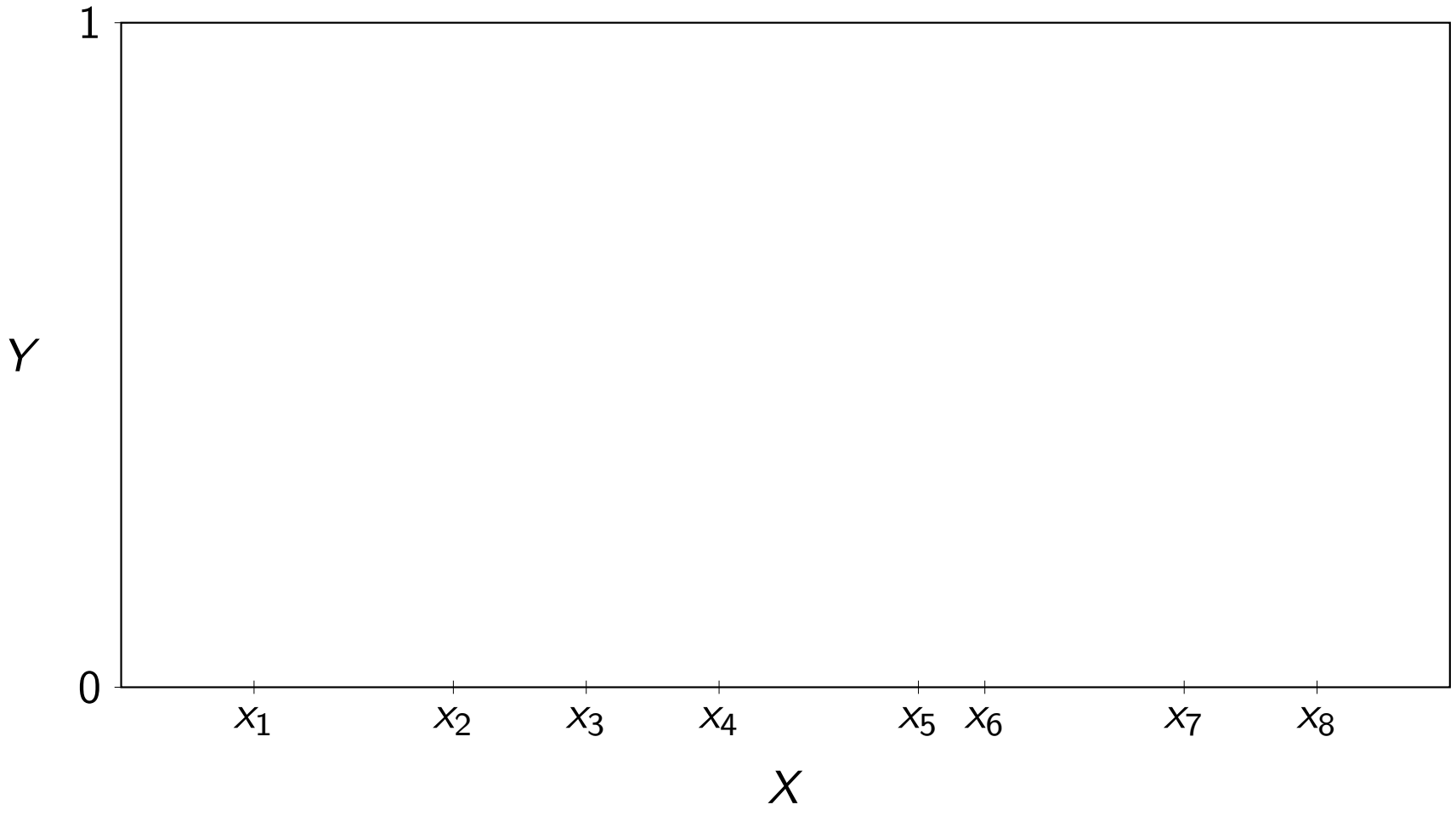
Regret of the learner:

$$\text{regret}_T = \widehat{L}_T - L_T^*.$$

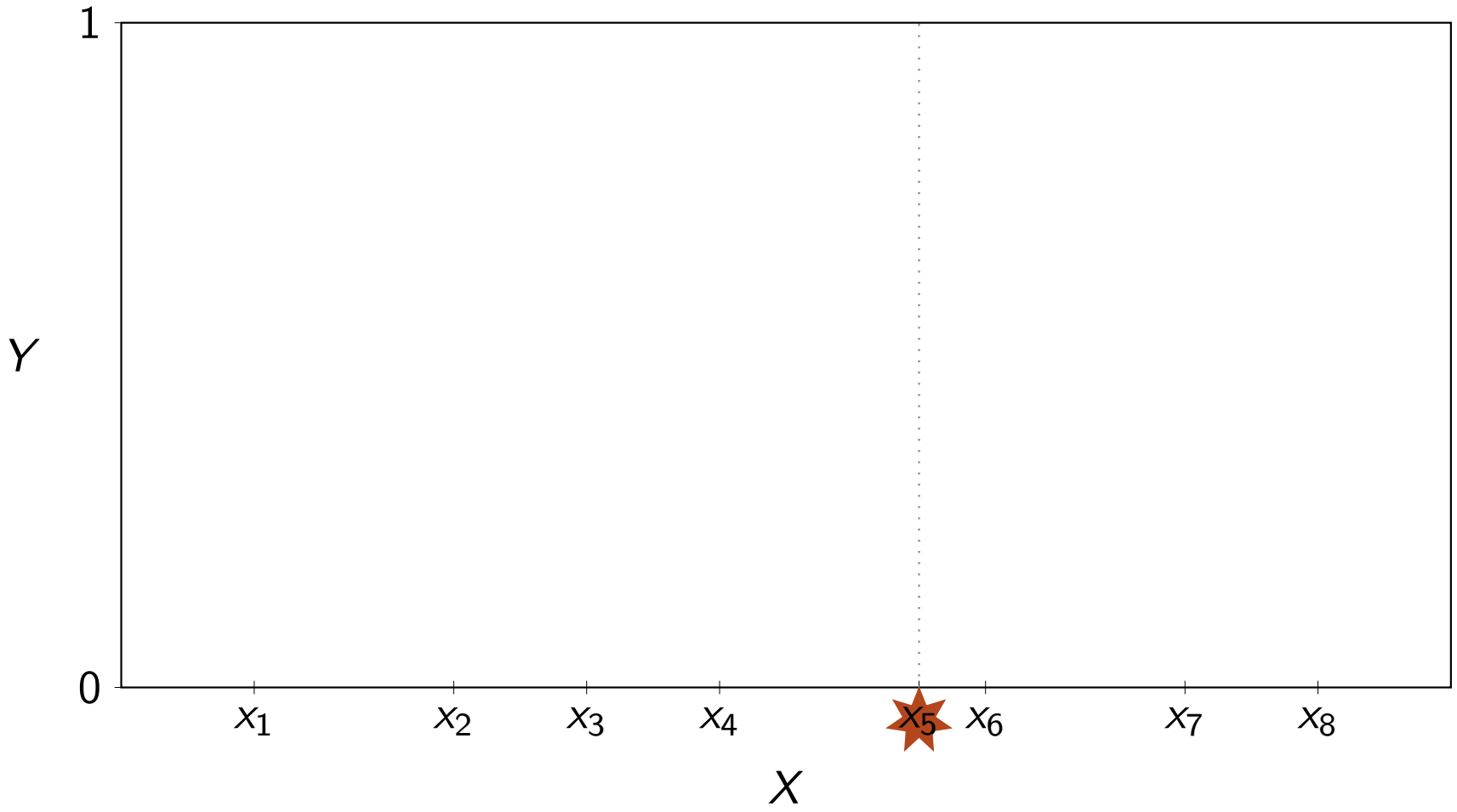
The goal is to minimize regret over all possible data sequences.

Outline

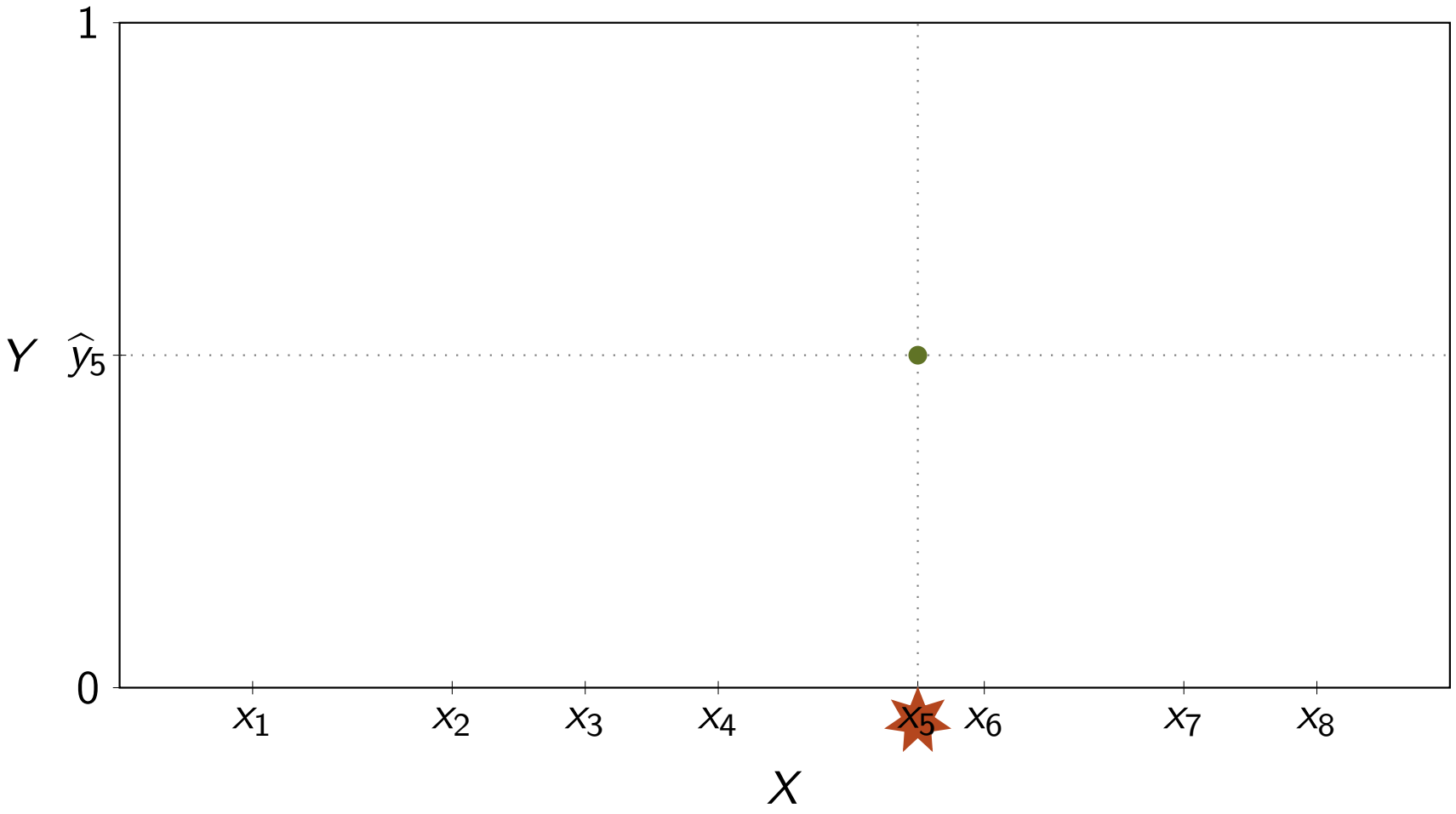
Online isotonic regression



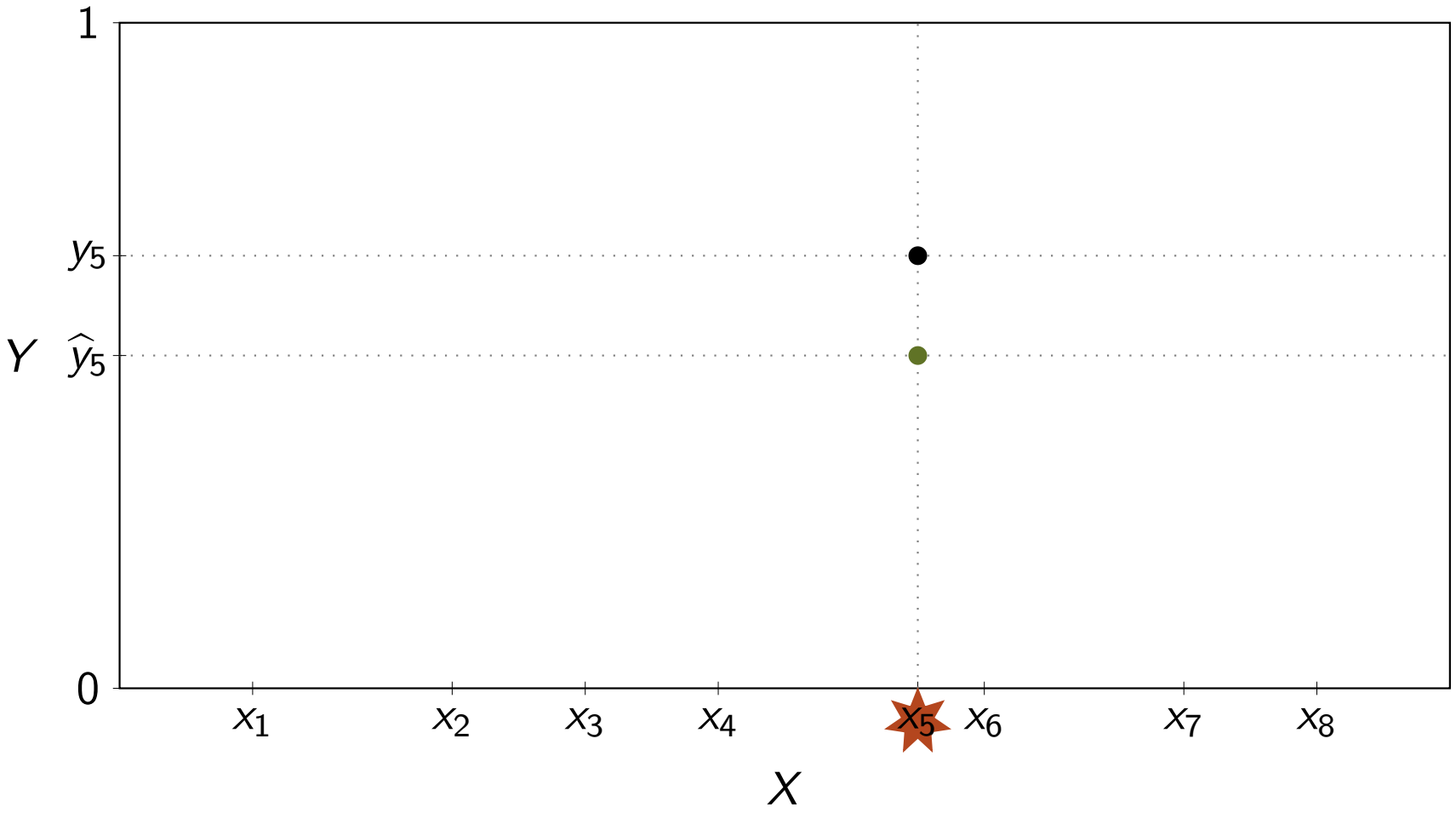
Online isotonic regression



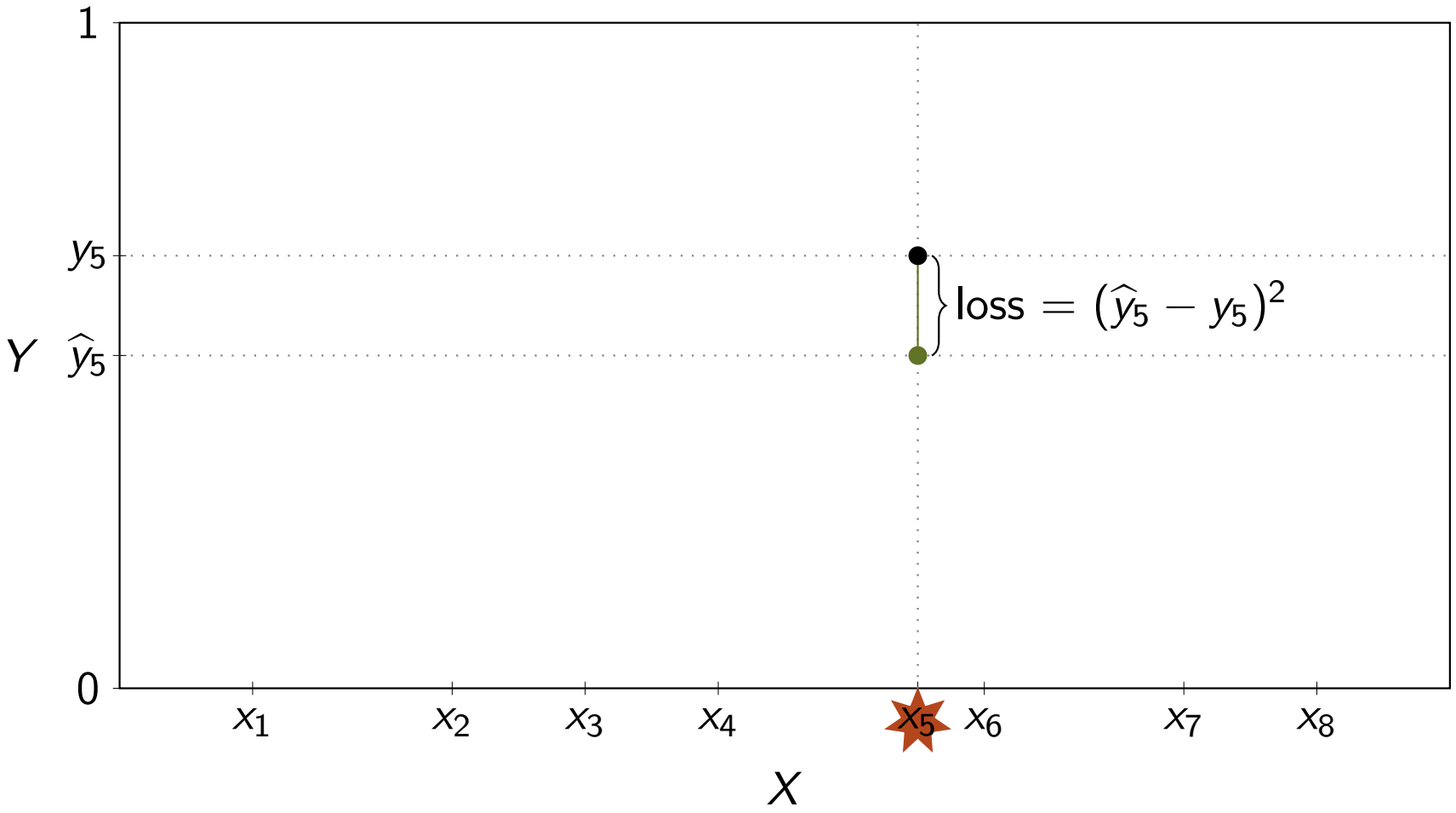
Online isotonic regression



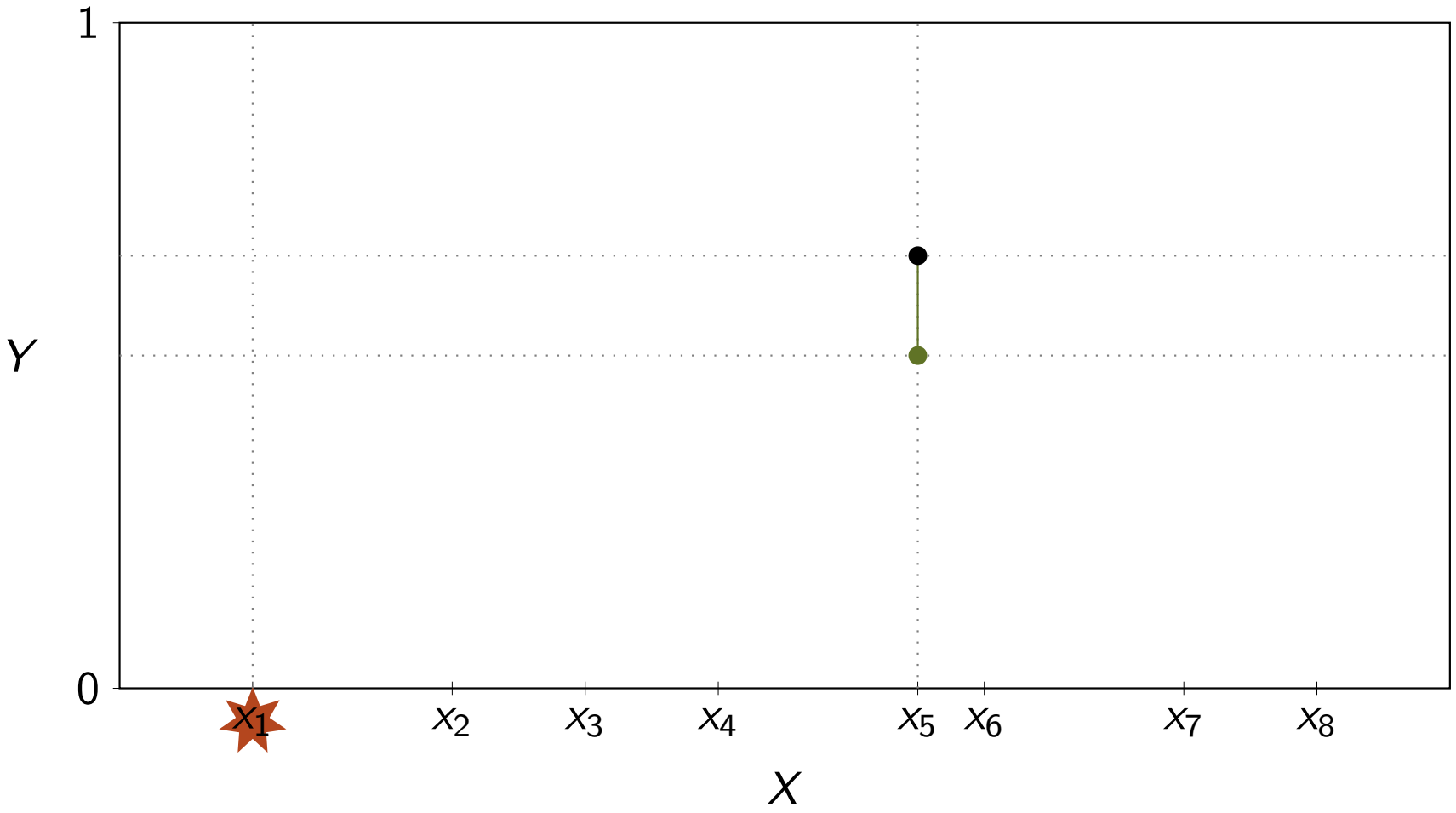
Online isotonic regression



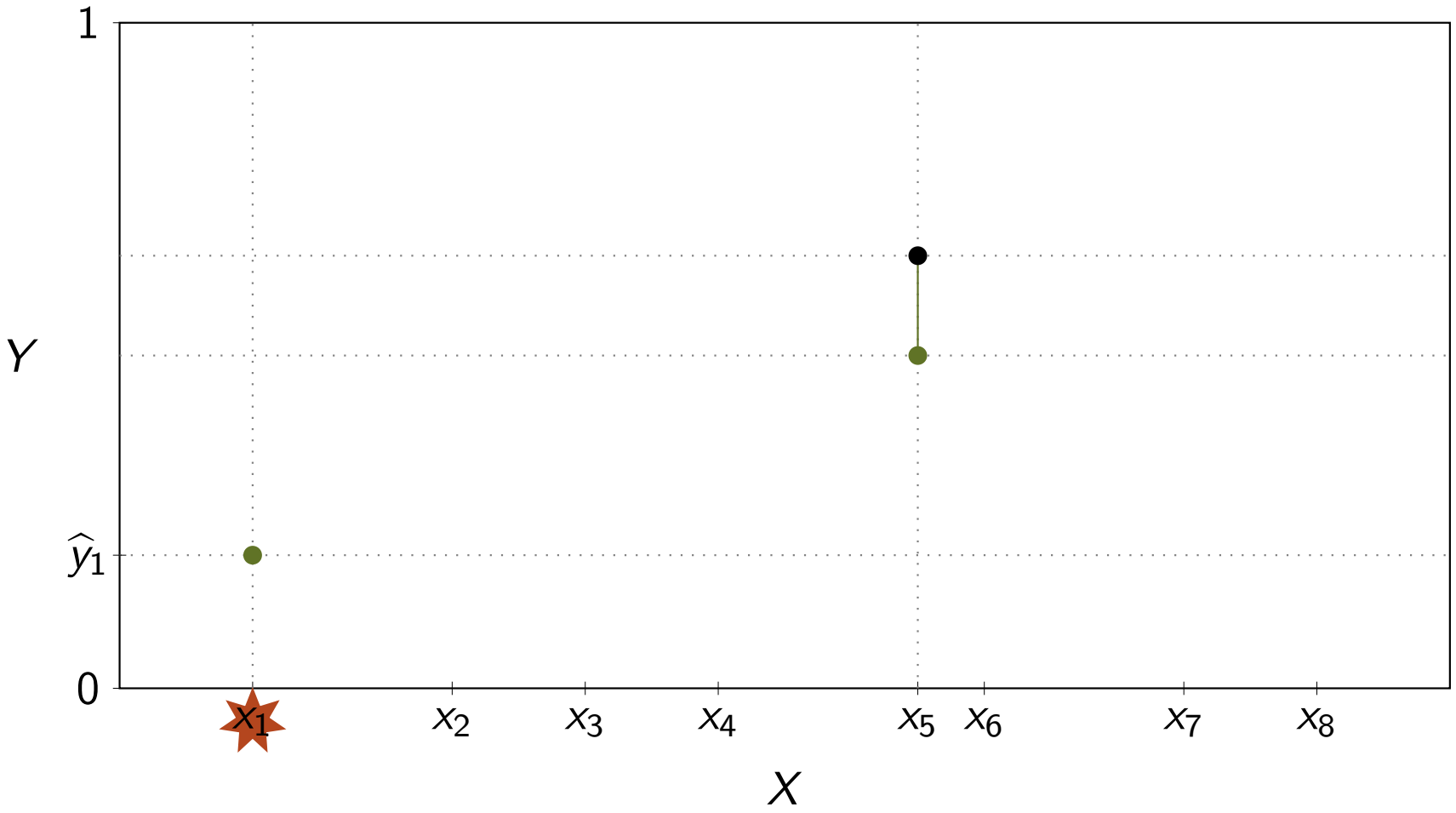
Online isotonic regression



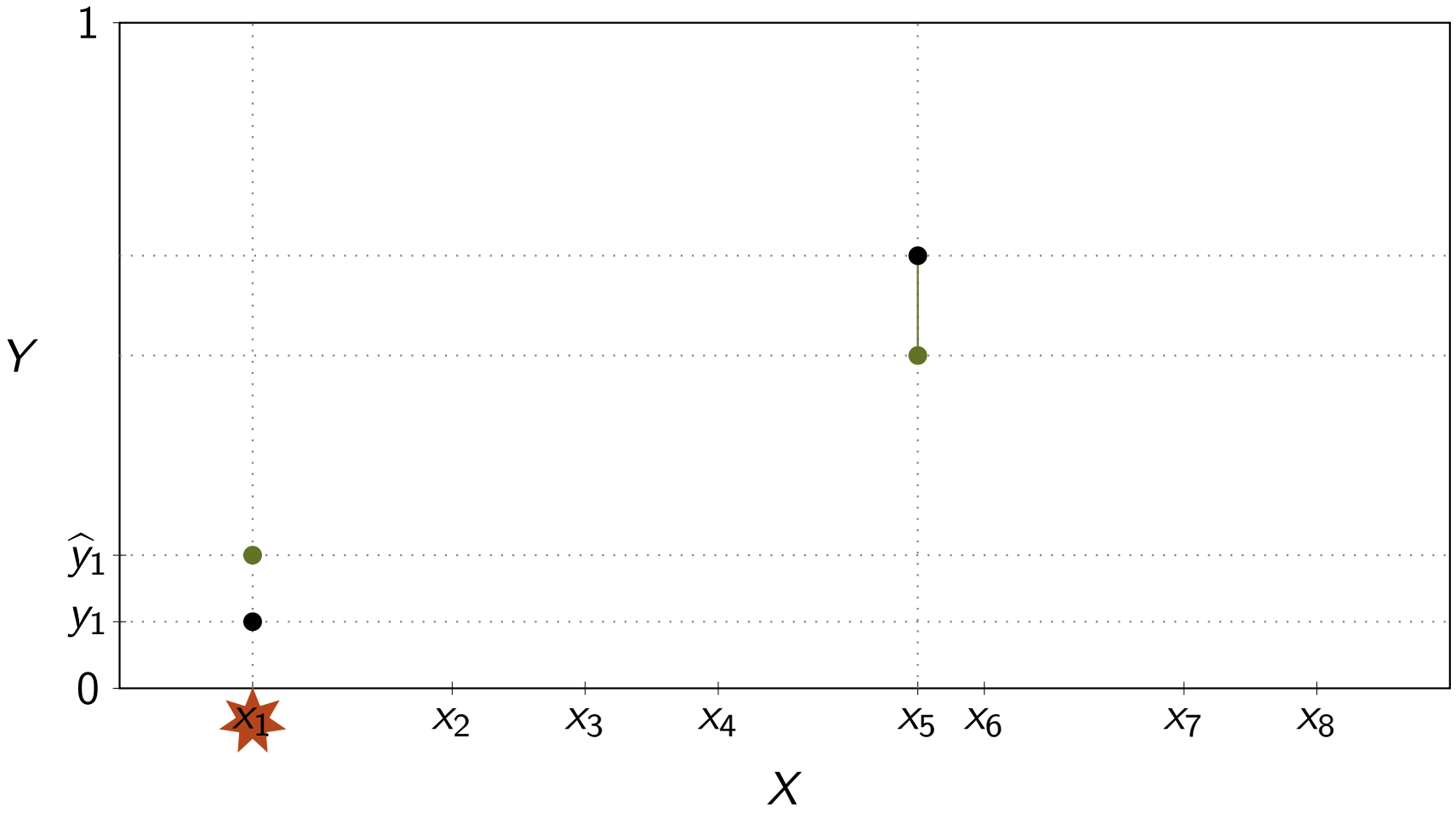
Online isotonic regression



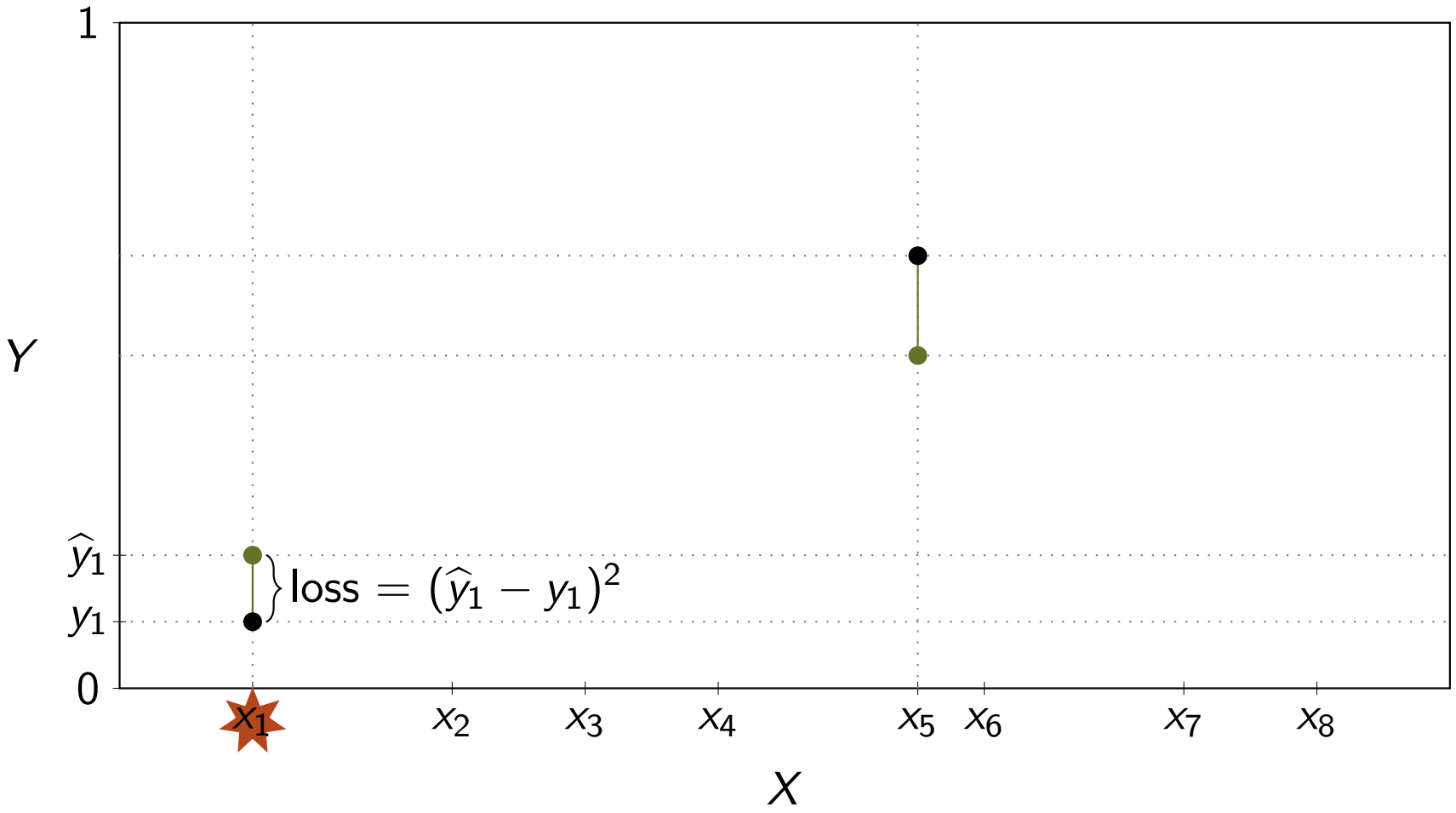
Online isotonic regression



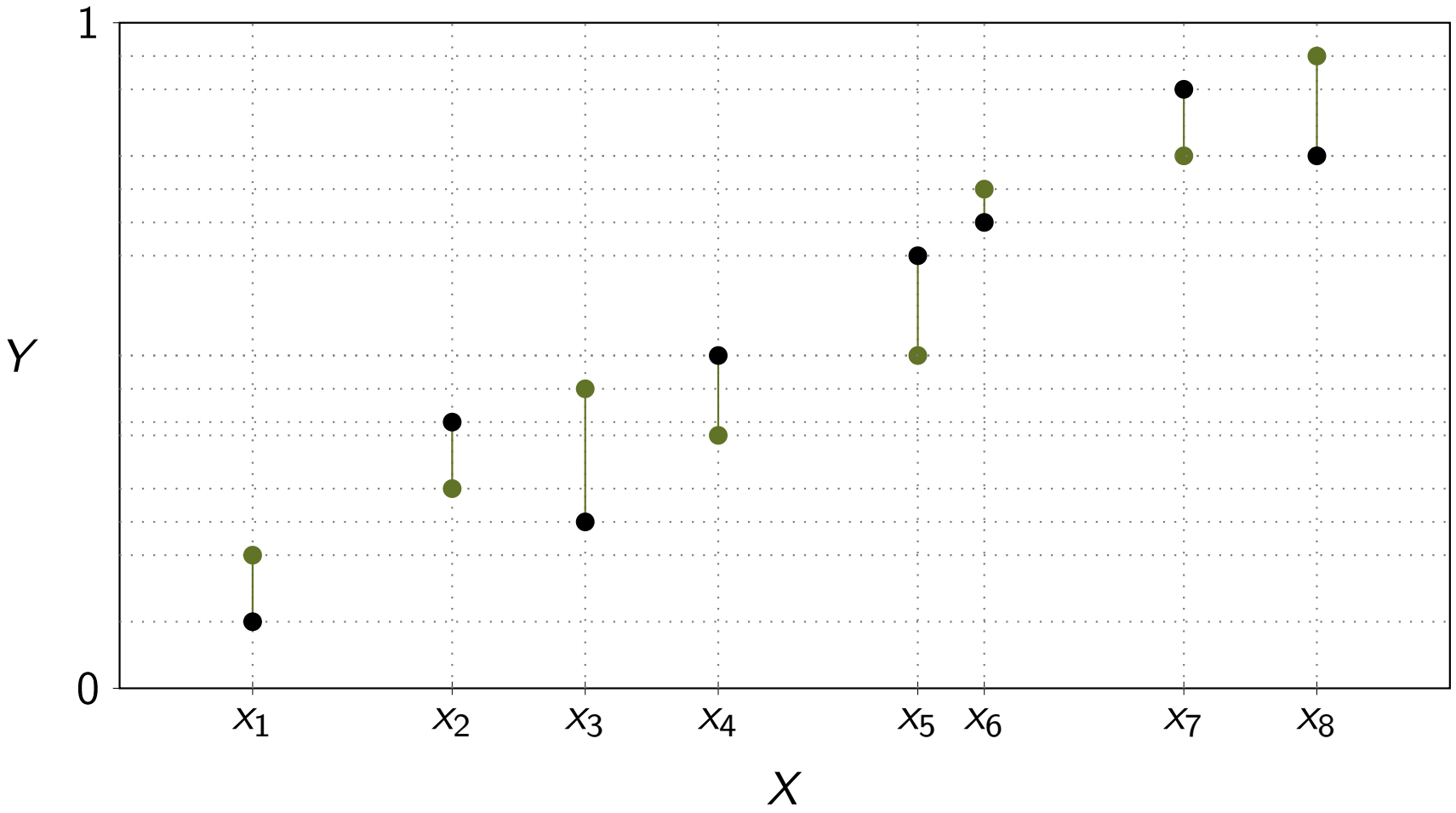
Online isotonic regression



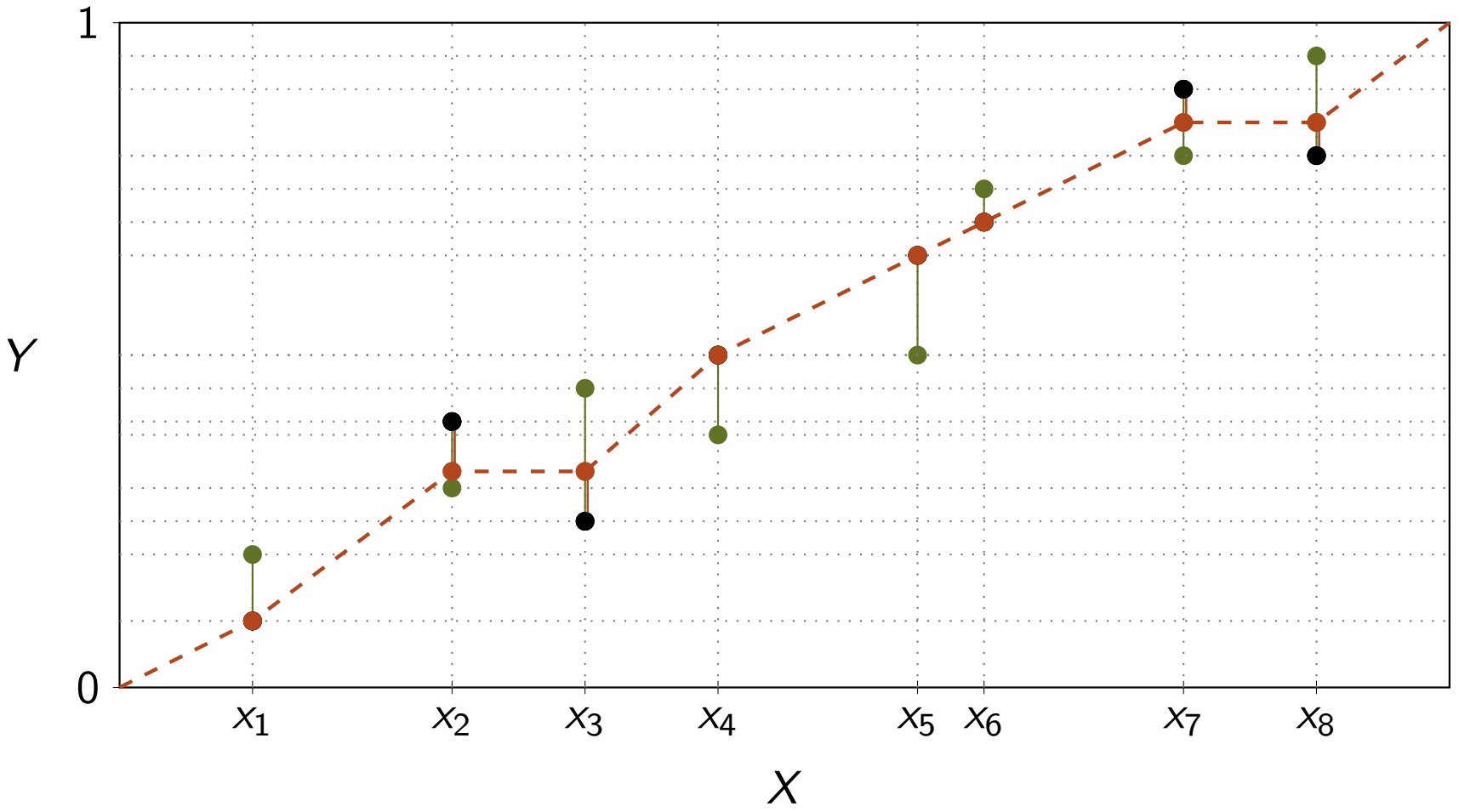
Online isotonic regression



Online isotonic regression



Online isotonic regression



Online isotonic regression

The protocol

Given: $x_1 < x_2 < \ldots < x_T$.

At trial $t = 1, \ldots, T$:

- Environment chooses a yet unlabeled point x_{i_t}.
- Learner predicts $\widehat{y}_{i_t} \in [0, 1]$.
- Environment reveals label $y_{i_t} \in [0, 1]$.
- Learner suffers squared loss $(y_{i_t} - \widehat{y}_{i_t})^2$.

Online isotonic regression

The protocol

Given: $x_1 < x_2 < \ldots < x_T$.

At trial $t = 1, \ldots, T$:

- Environment chooses a yet unlabeled point x_{i_t}.
- Learner predicts $\widehat{y}_{i_t} \in [0, 1]$.
- Environment reveals label $y_{i_t} \in [0, 1]$.
- Learner suffers squared loss $(y_{i_t} - \widehat{y}_{i_t})^2$.

Strategies = isotonic functions:

$$\mathcal{F} = \{f : \ f(x_1) \leq f(x_2) \leq \ldots \leq f(x_T)\}$$

Online isotonic regression

The protocol

Given: $x_1 < x_2 < \ldots < x_T$.

At trial $t = 1, \ldots, T$:

- Environment chooses a yet unlabeled point x_{i_t}.
- Learner predicts $\widehat{y}_{i_t} \in [0, 1]$.
- Environment reveals label $y_{i_t} \in [0, 1]$.
- Learner suffers squared loss $(y_{i_t} - \widehat{y}_{i_t})^2$.

Strategies = isotonic functions:

$$\mathcal{F} = \{f : \ f(x_1) \leq f(x_2) \leq \ldots \leq f(x_T)\}$$

$$\text{regret}_T = \sum_{t=1}^{T} (y_{i_t} - \widehat{y}_{i_t})^2 \ - \ \min_{f \in \mathcal{F}} \sum_{t=1}^{T} (y_{i_t} - f(x_{i_t}))^2$$

Online isotonic regression

$$\mathcal{F} = \{f : \ f(x_1) \leq f(x_2) \leq \ldots \leq f(x_T)\}$$

$$\text{regret}_T = \sum_{t=1}^{T} (y_{i_t} - \widehat{y}_{i_t})^2 \ - \ \min_{f \in \mathcal{F}} \sum_{t=1}^{T} (y_{i_t} - f(x_{i_t}))^2$$

Cumulative loss of the learner should not be much larger than the loss of (optimal) isotonic regression function in hindsight.

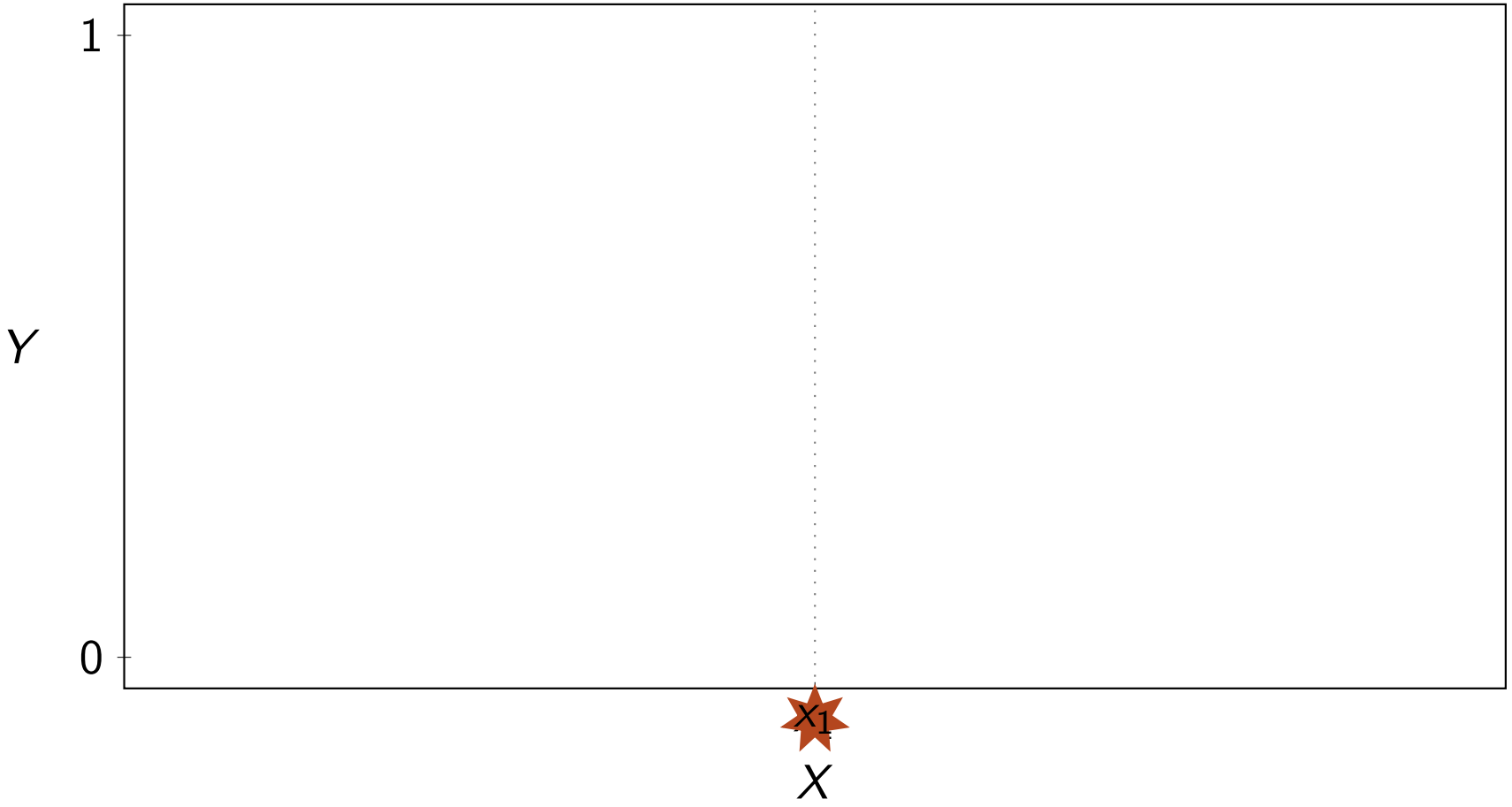
Only the order $x_1 < \ldots < x_T$ matters, not the values.

The adversary is too powerful!

Every algorithm will have $\Omega(T)$ regret

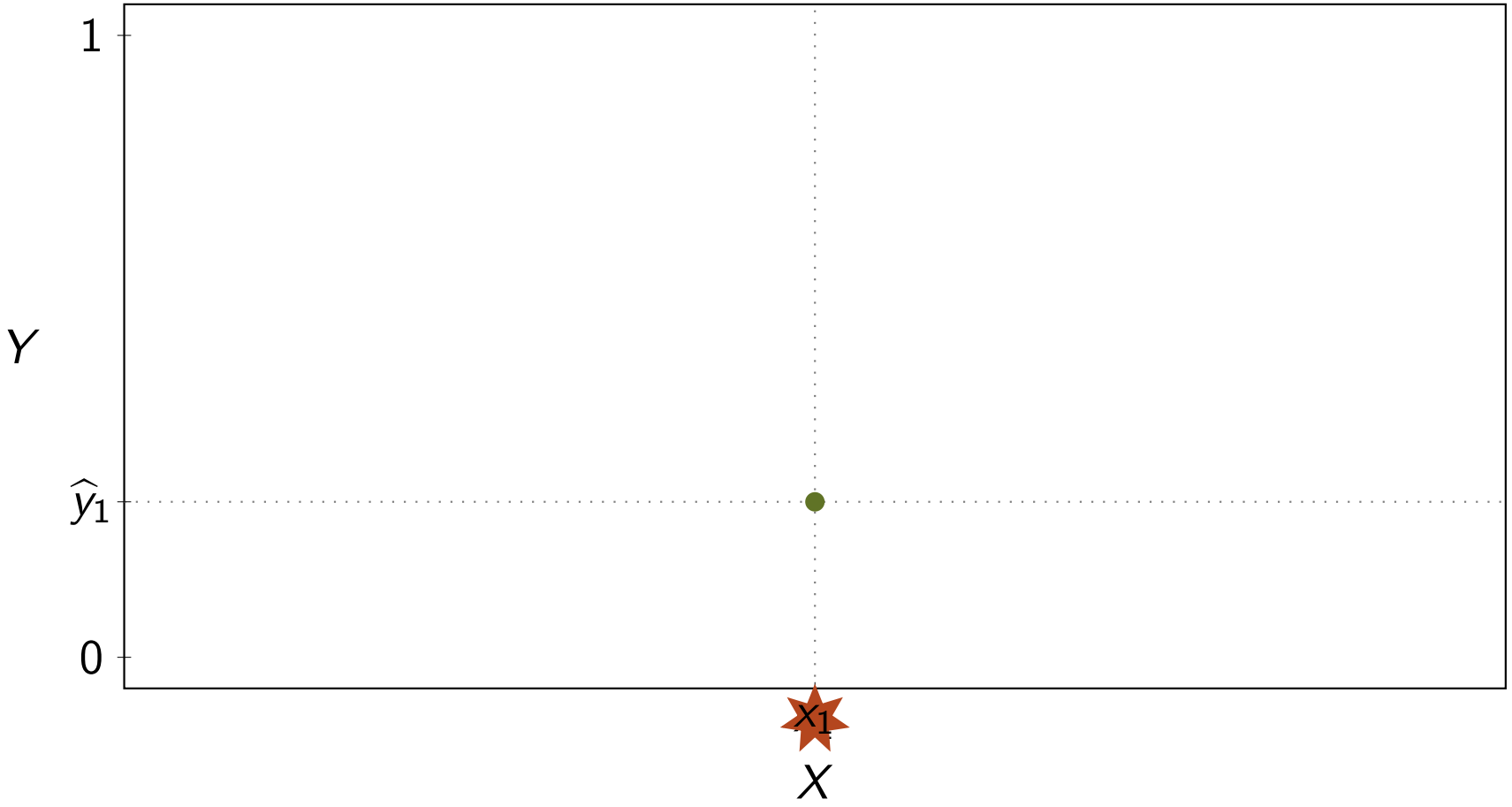
The adversary is too powerful!

Every algorithm will have $\Omega(T)$ regret



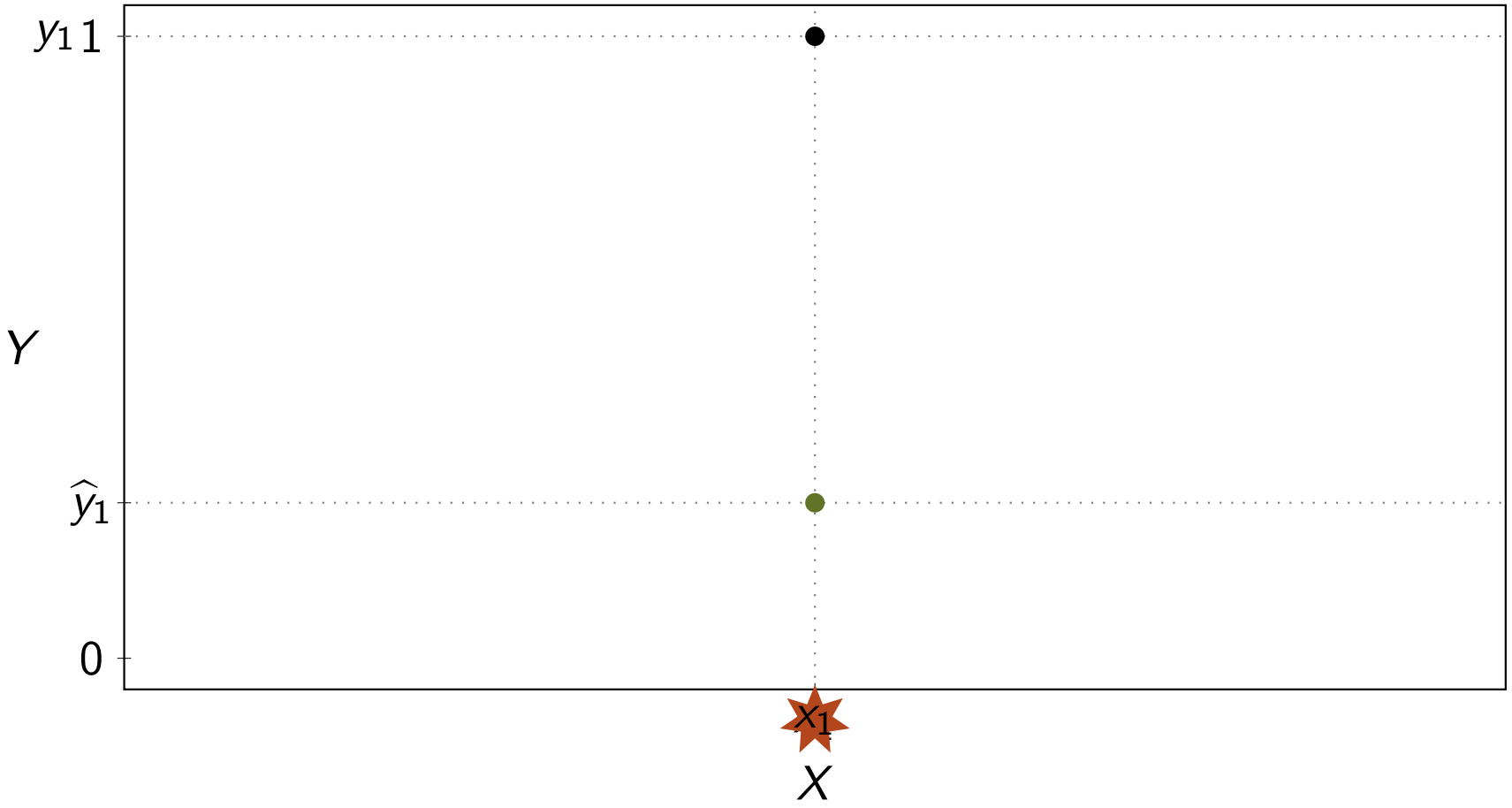
The adversary is too powerful!

Every algorithm will have $\Omega(T)$ regret



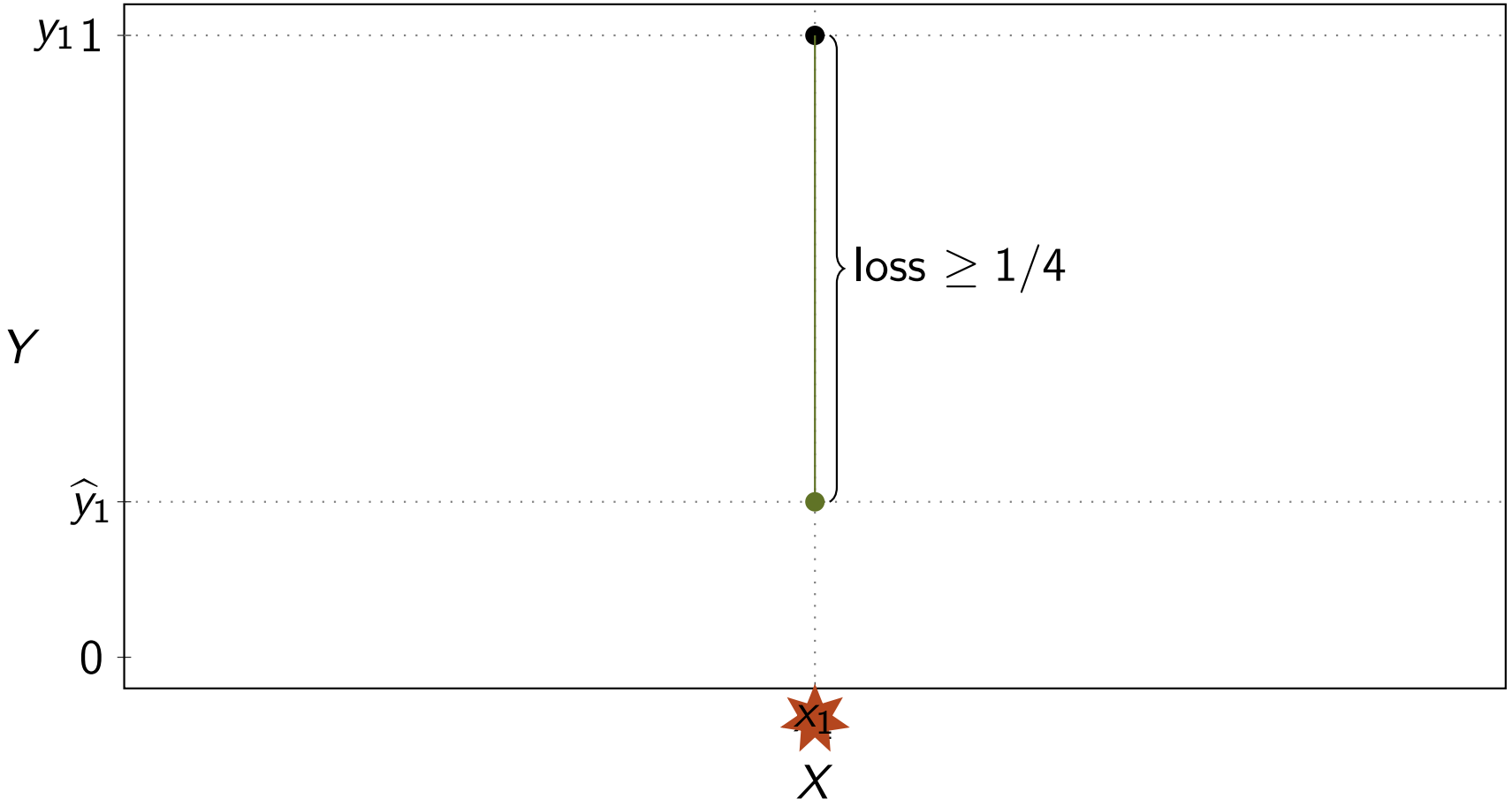
The adversary is too powerful!

Every algorithm will have $\Omega(T)$ regret



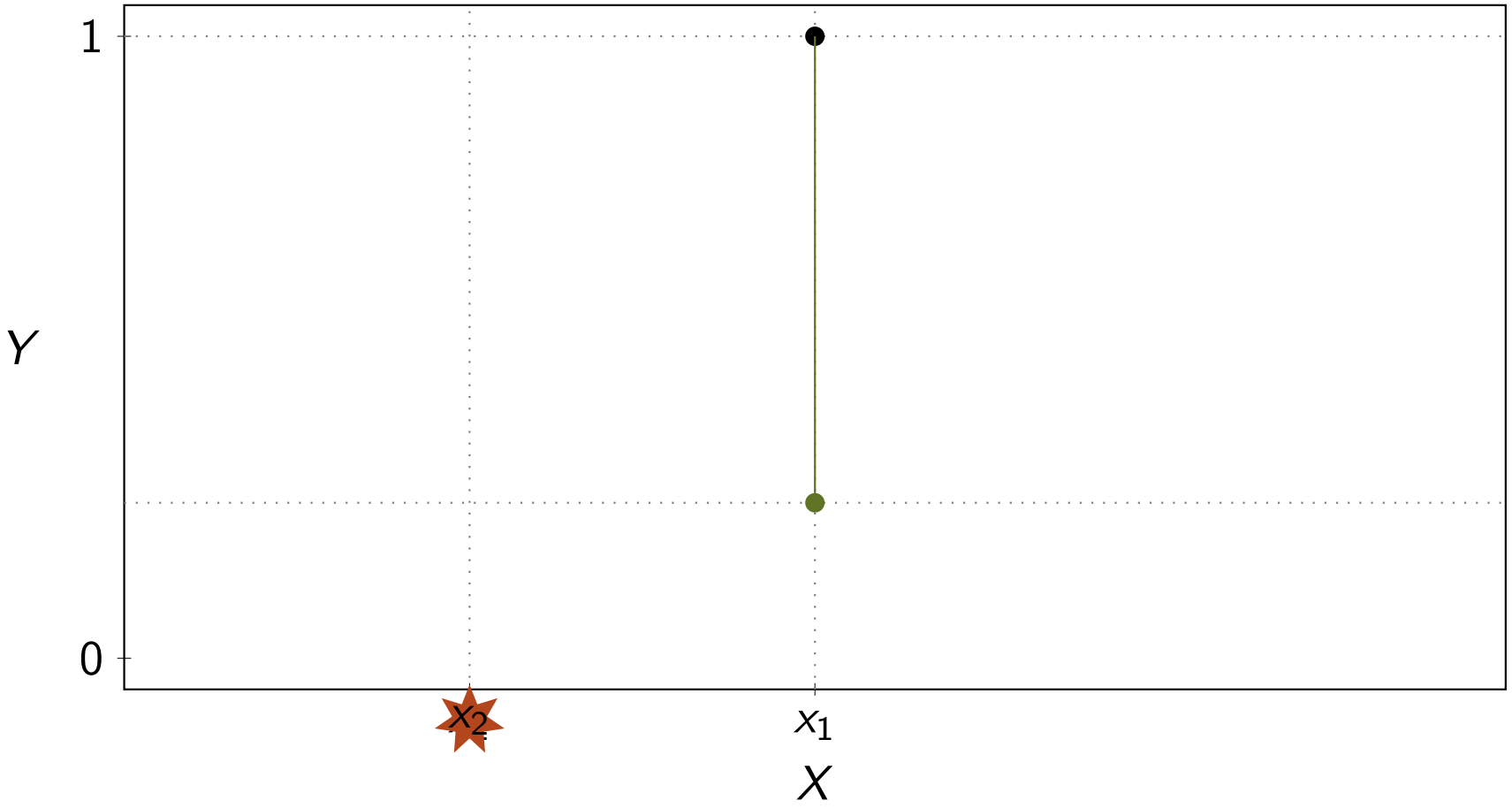
The adversary is too powerful!

Every algorithm will have $\Omega(T)$ regret



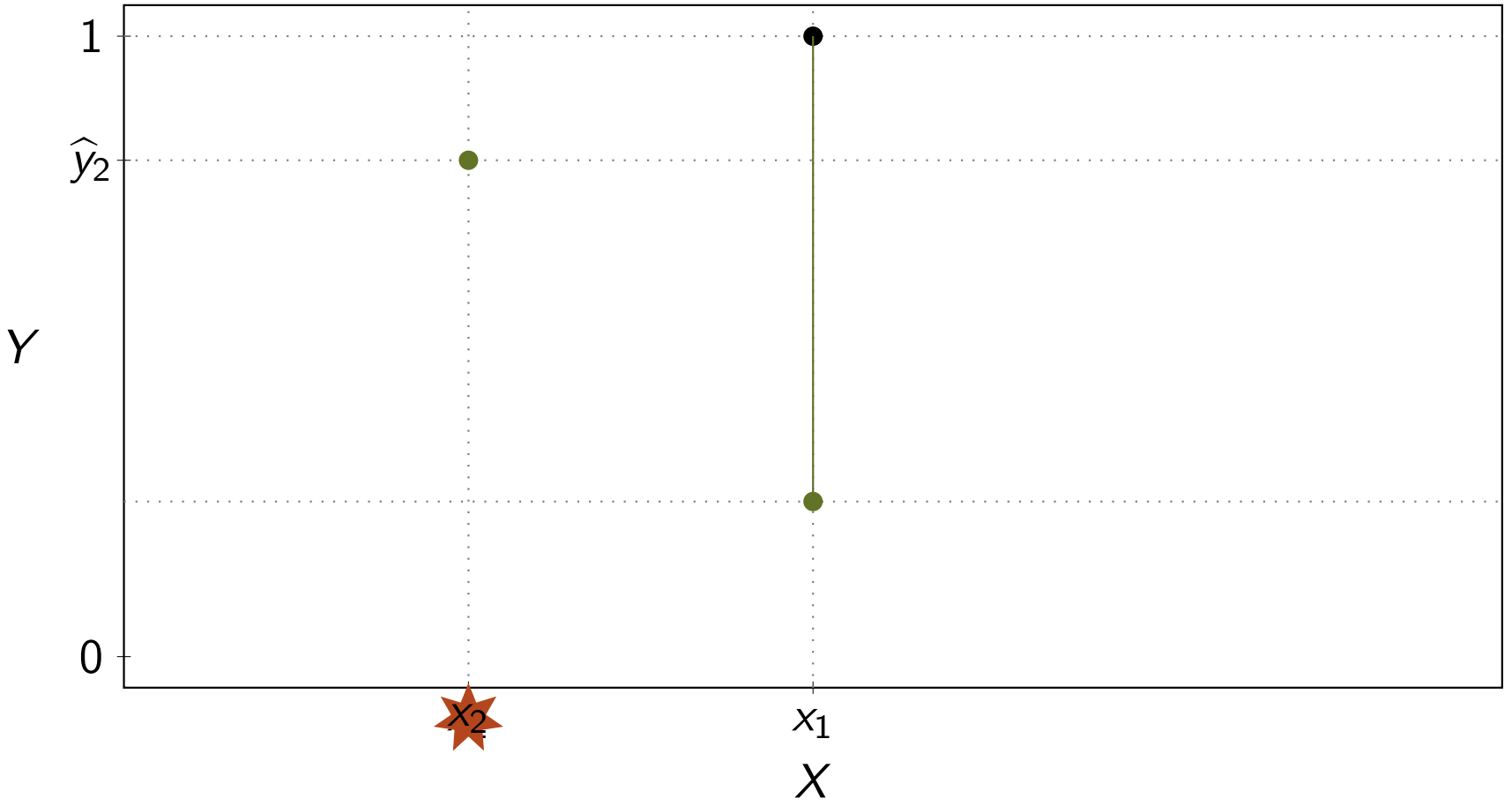
The adversary is too powerful!

Every algorithm will have $\Omega(T)$ regret



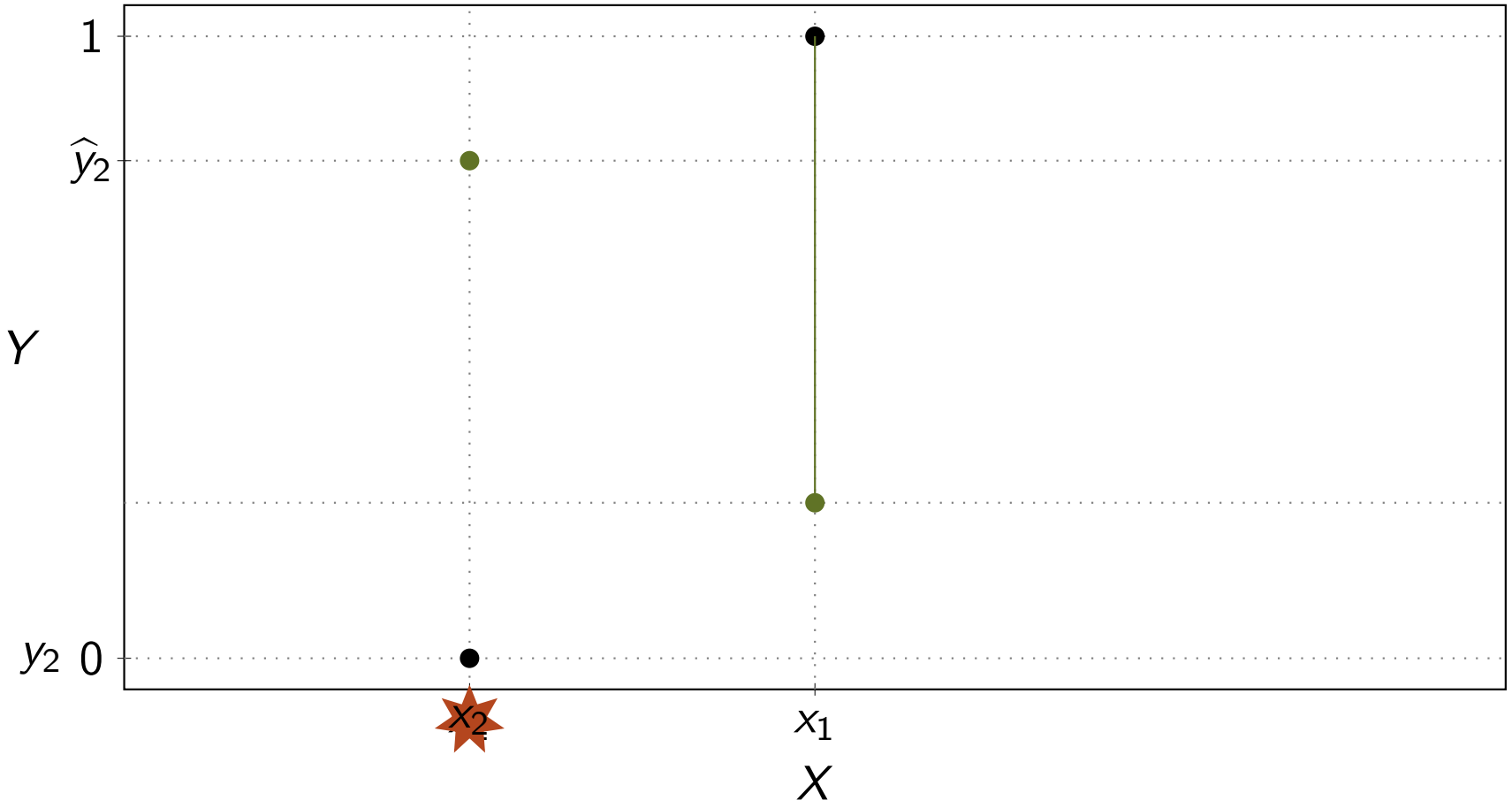
The adversary is too powerful!

Every algorithm will have $\Omega(T)$ regret



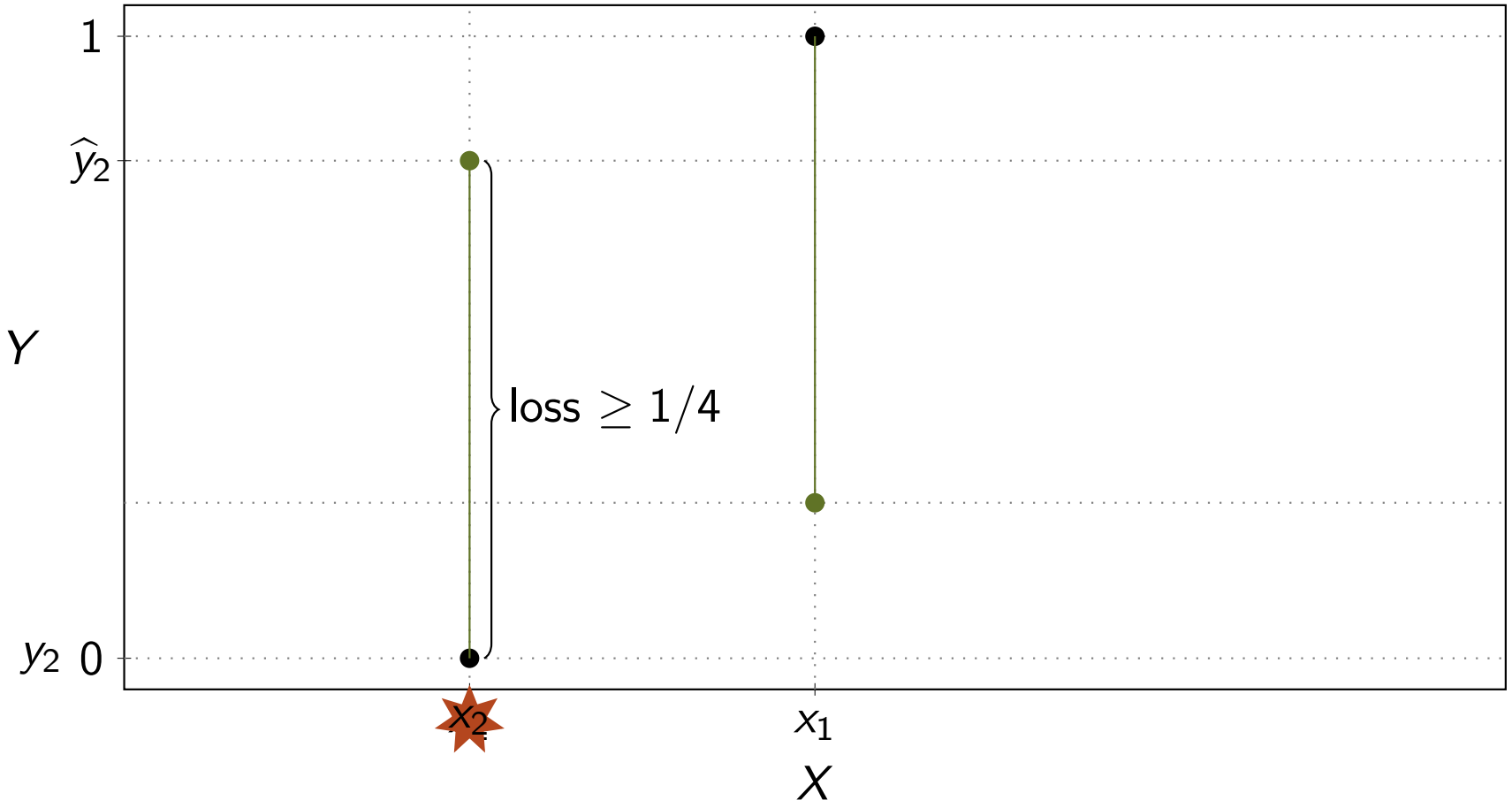
The adversary is too powerful!

Every algorithm will have $\Omega(T)$ regret



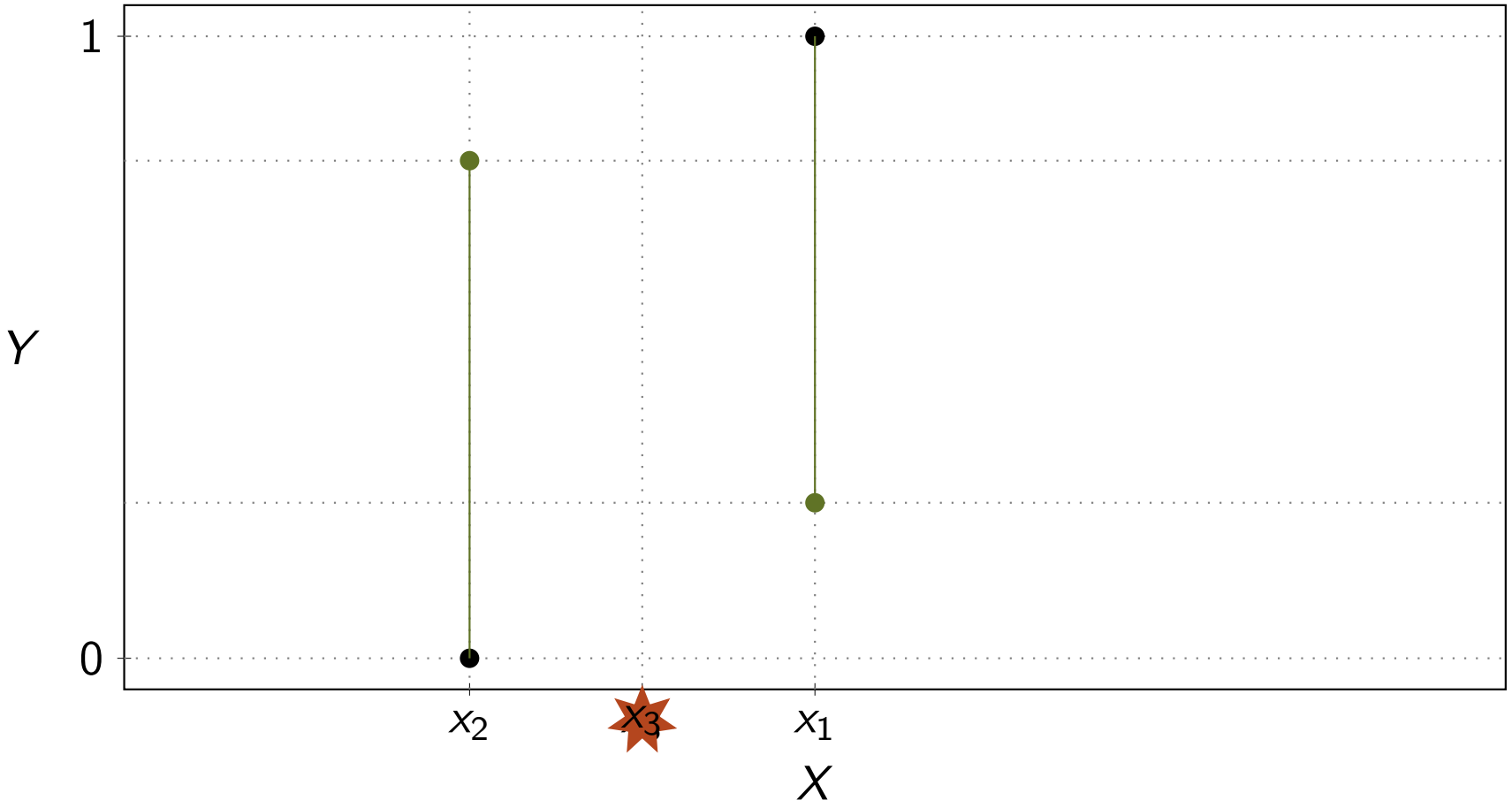
The adversary is too powerful!

Every algorithm will have $\Omega(T)$ regret



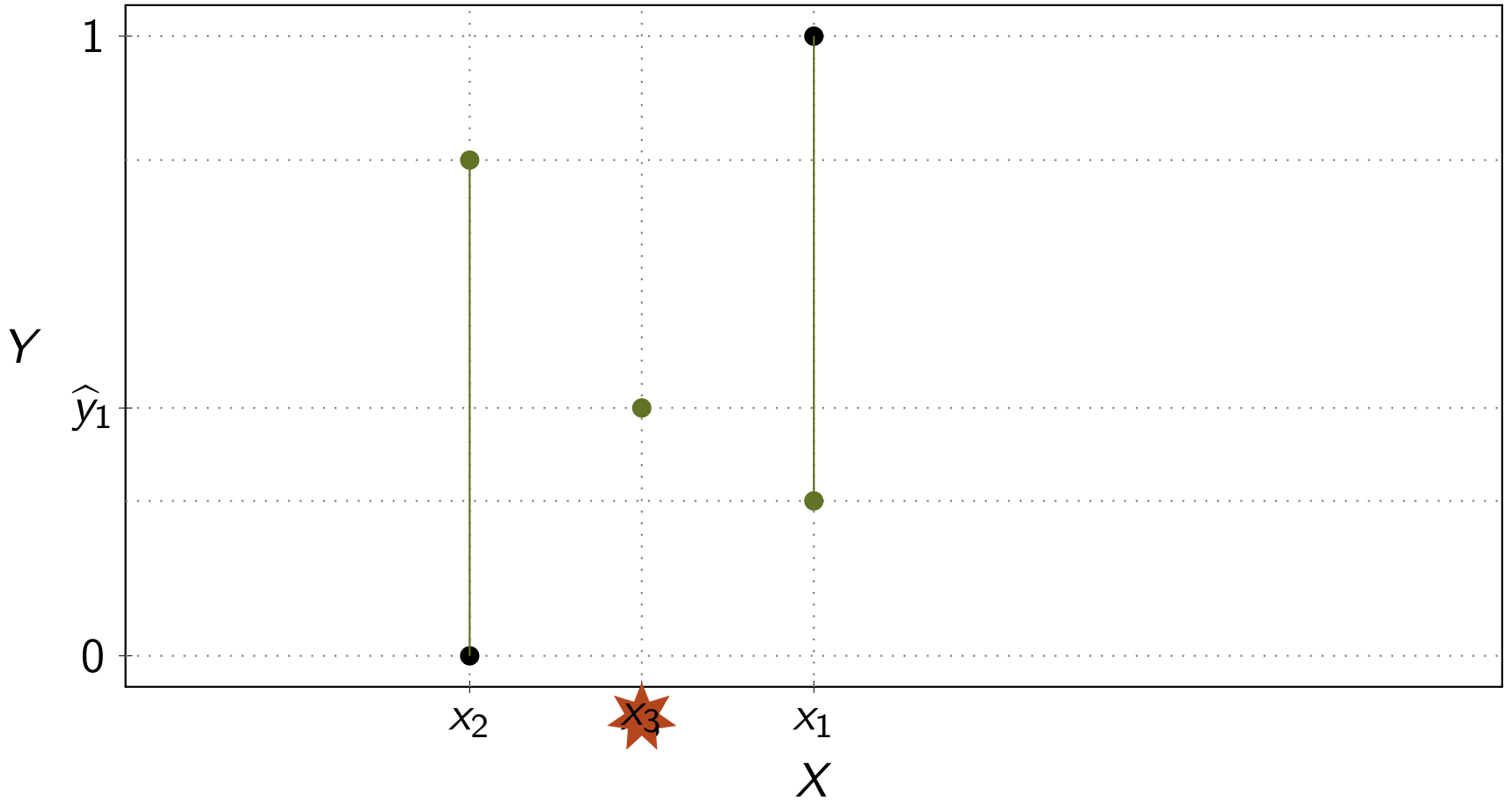
The adversary is too powerful!

Every algorithm will have $\Omega(T)$ regret



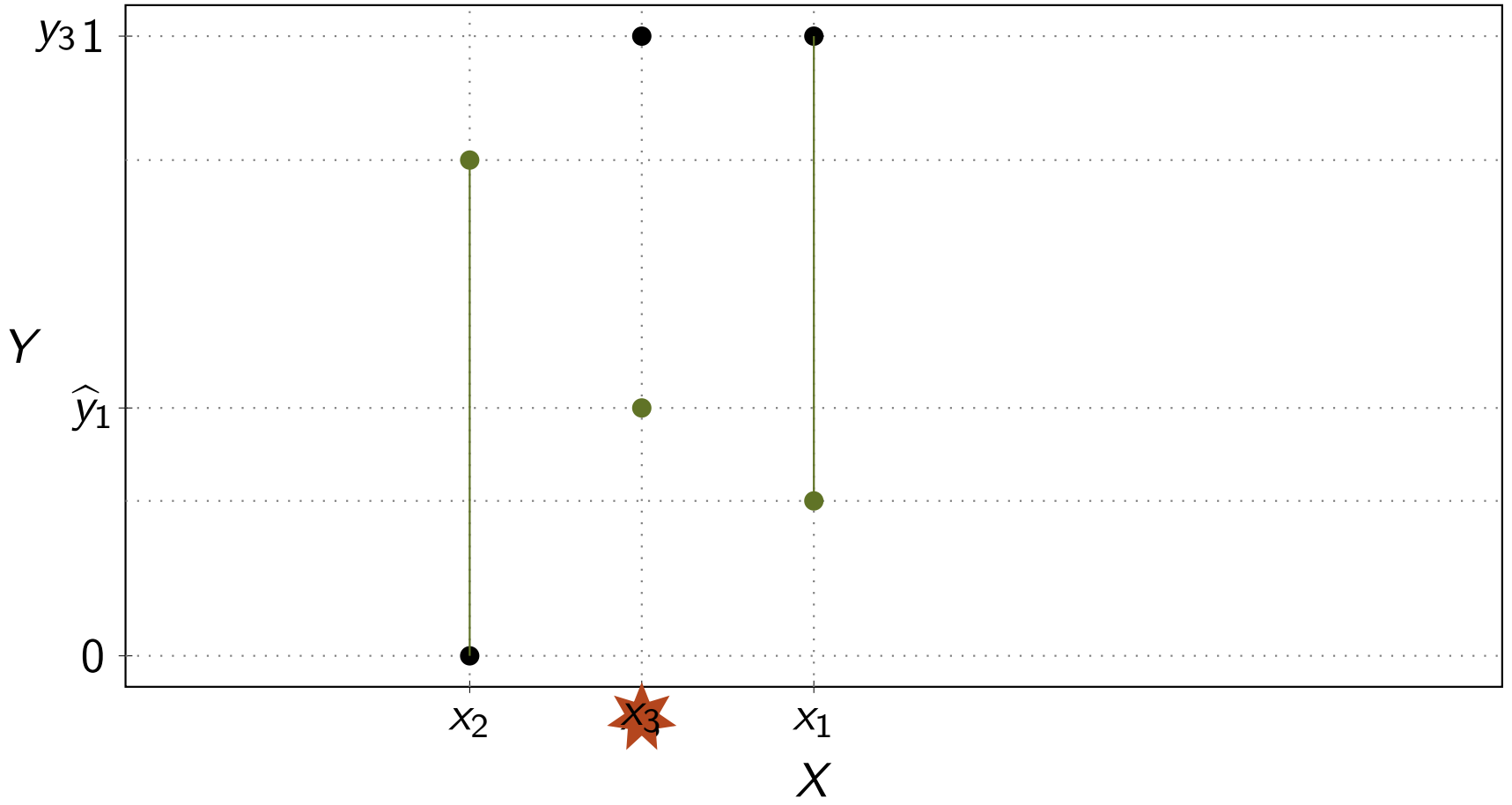
The adversary is too powerful!

Every algorithm will have $\Omega(T)$ regret



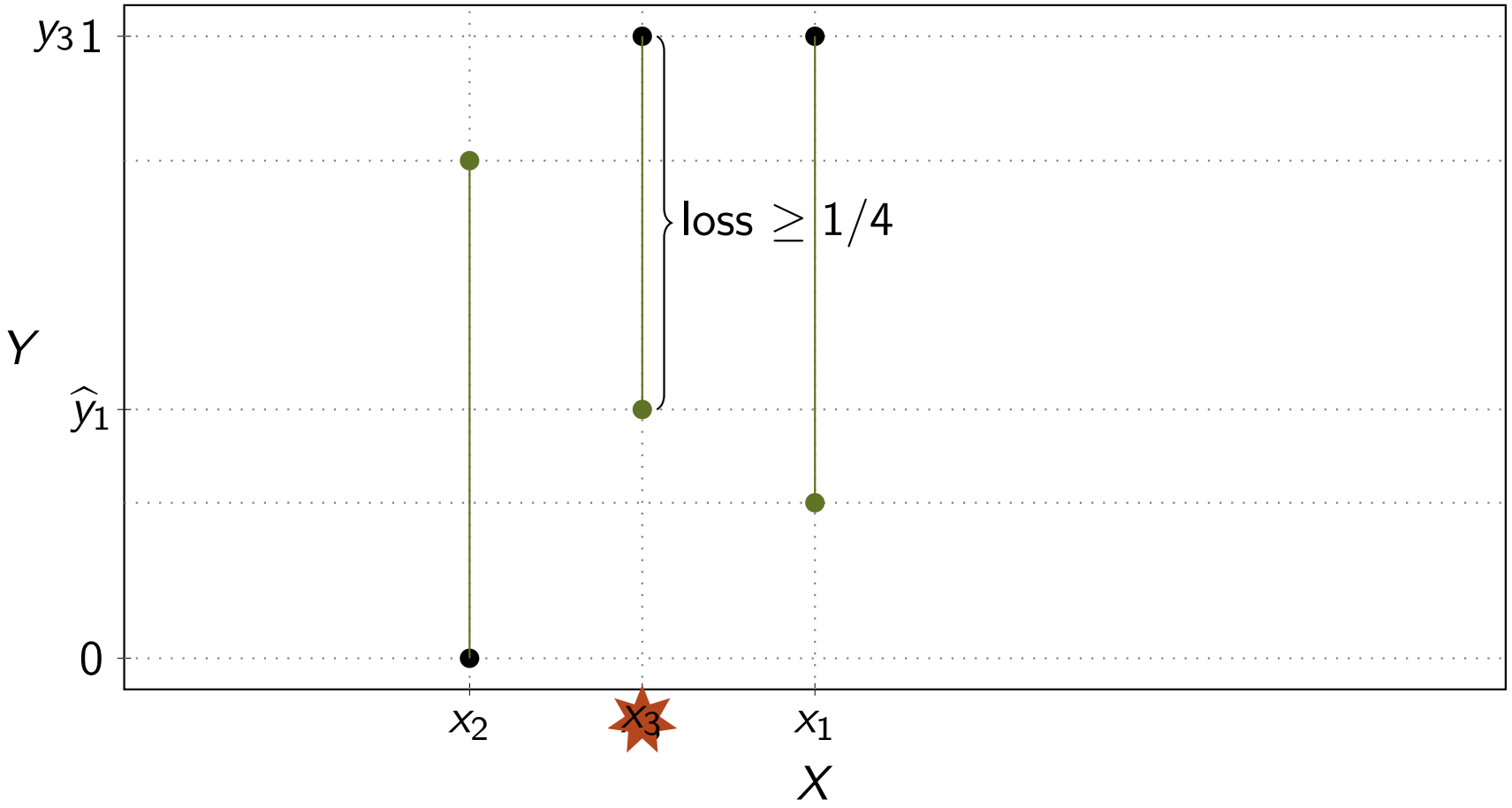
The adversary is too powerful!

Every algorithm will have $\Omega(T)$ regret



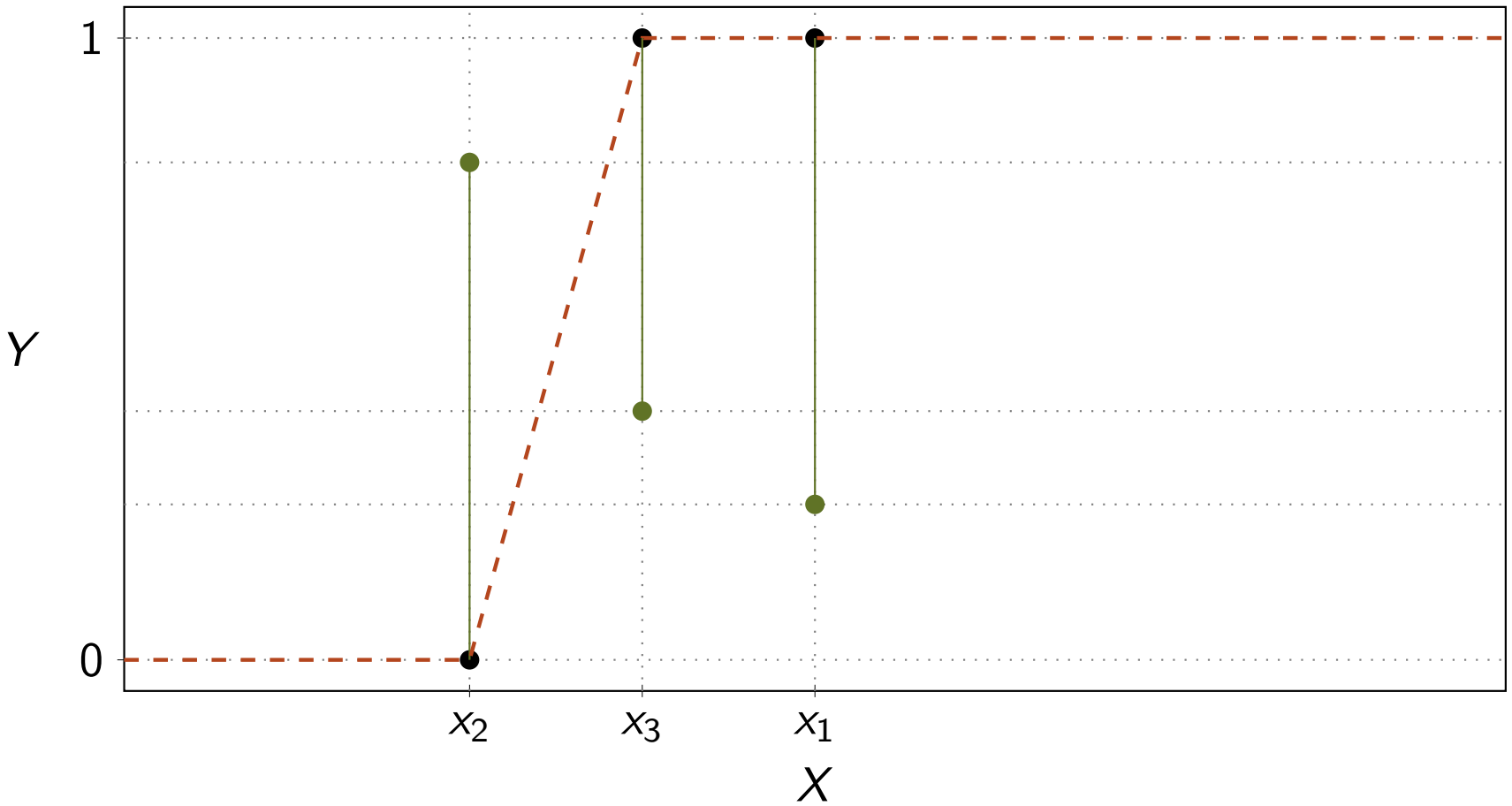
The adversary is too powerful!

Every algorithm will have $\Omega(T)$ regret



The adversary is too powerful!

Every algorithm will have $\Omega(T)$ regret



Algorithms' loss $\geq \frac{1}{4}$ per trial, loss of best isotonic function $= 0$.

Outline

Fixed design

Data $x_1, \ldots, x_T$ is known in advance to the learner

We will show that in such model, efficient online algorithms exist.

K., Koolen, Malek: *Online Isotonic Regression*. Proc. of Conference on Learning Theory (COLT), pp. 1165–1189, 2016.

Off-the-shelf online algorithms

Algorithm	General bound	Bound for online IR
Stochastic Gradient Descent	$G_2 D_2 \sqrt{T}$	T
Exponentiated Gradient	$G_\infty D_1 \sqrt{T \log d}$	$\sqrt{T \log T}$
Follow the Leader	$G_2 D_2 d \log T$	$T^2 \log T$
Exponential Weights	$d \log T$	$T \log T$

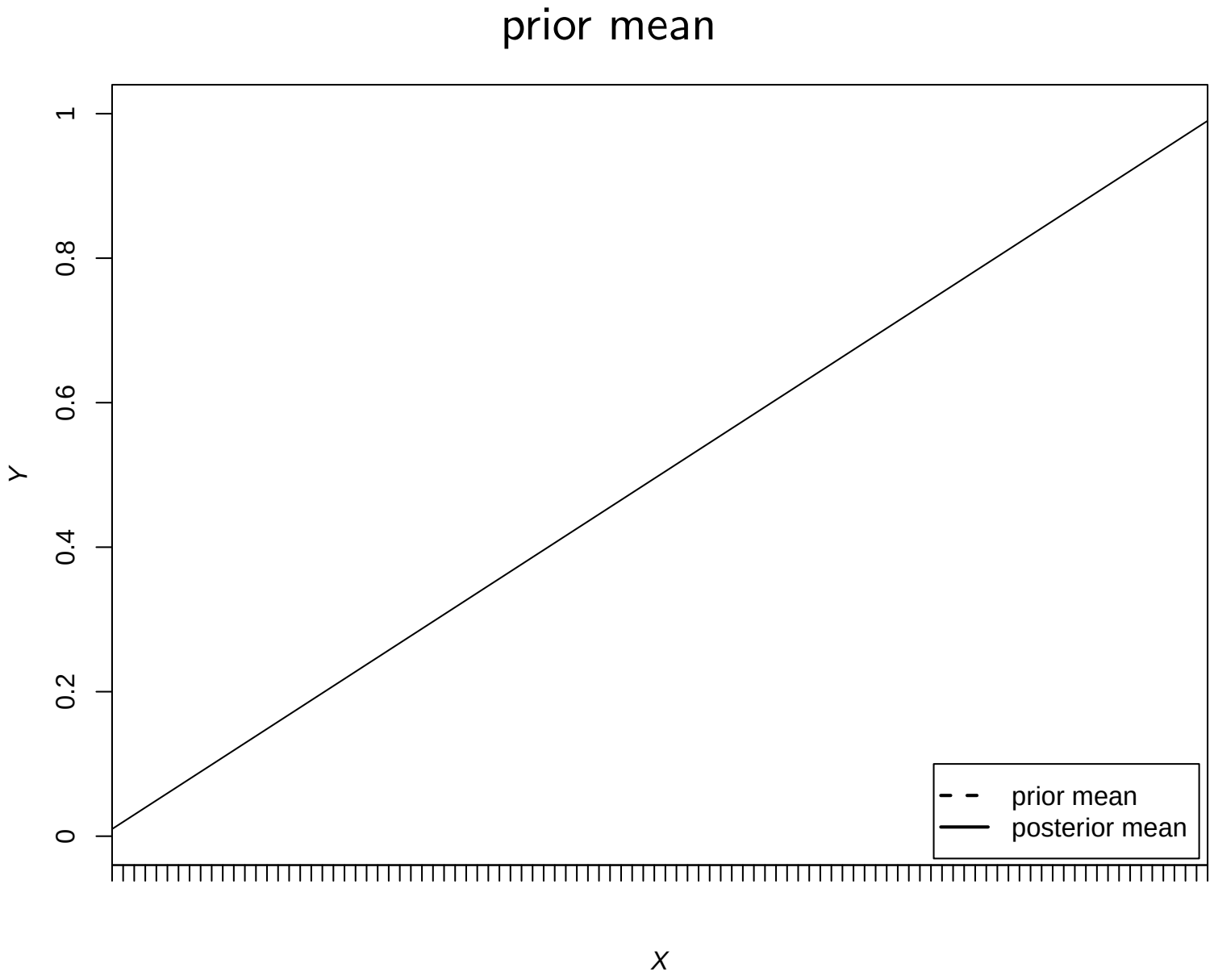
These bounds are tight (up to logarithmic factor).

Exponential Weights (Bayes) with uniform prior

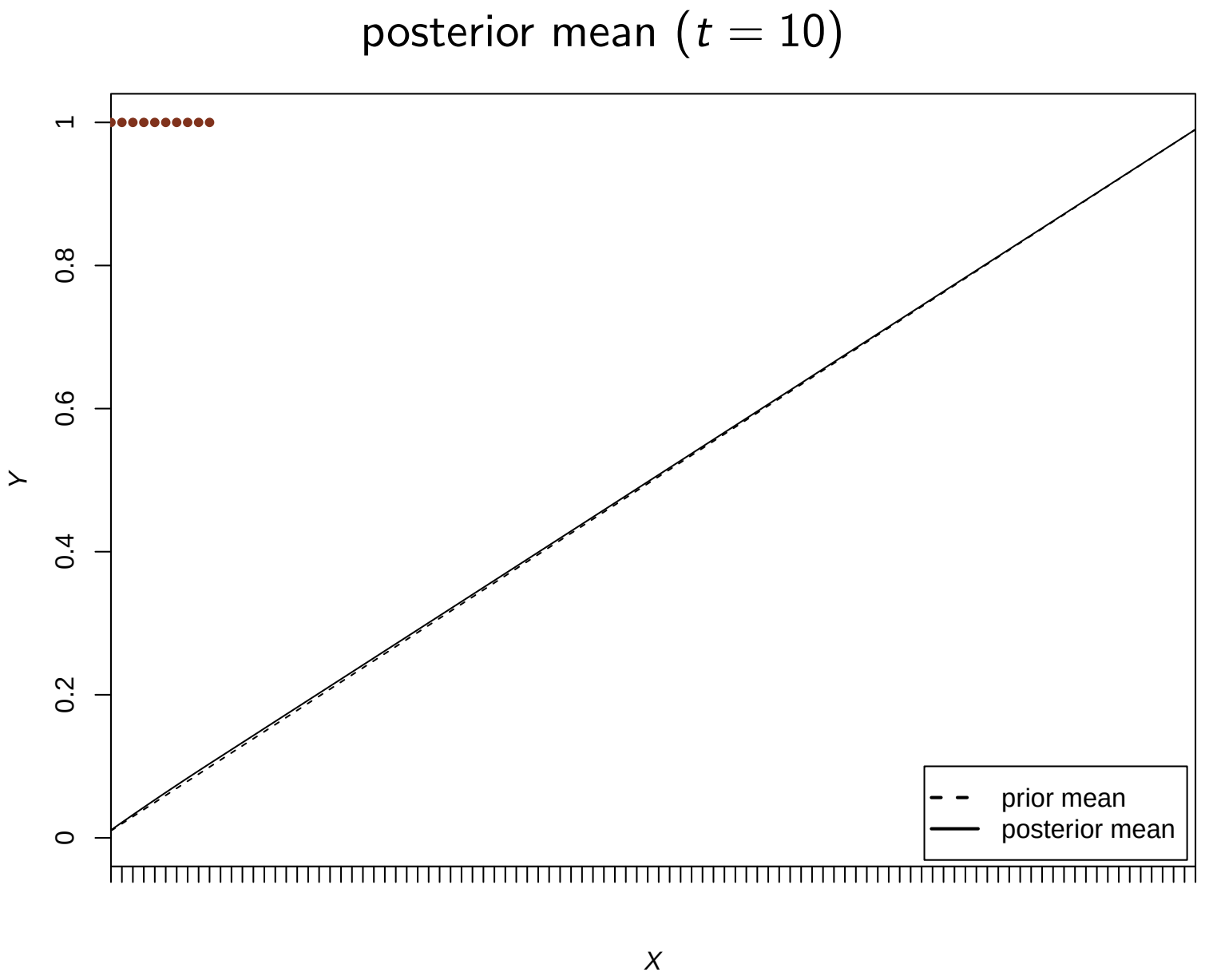
Let $\boldsymbol{f} = (f_1, \ldots, f_T)$ denote values of f at $(x_1, \ldots, x_T)$.

$$\begin{aligned}
\pi(\boldsymbol{f}) &= \text{const}, \qquad \text{for all } \boldsymbol{f}: f_1 \leq \ldots \leq f_T, \\
P(\boldsymbol{f}|y_{i_1}, \ldots, y_{i_t}) &\propto \pi(\boldsymbol{f}) e^{-\frac{1}{2}\text{loss}_{1\ldots t}(\boldsymbol{f})}, \\
\widehat{y}_{i_{t+1}} &= \underbrace{\int f_{i_{t+1}} P(\boldsymbol{f}|y_{i_1}, \ldots, y_{i_t}) \mathrm{d}\boldsymbol{f}}_{= \text{posterior mean}}.
\end{aligned}$$

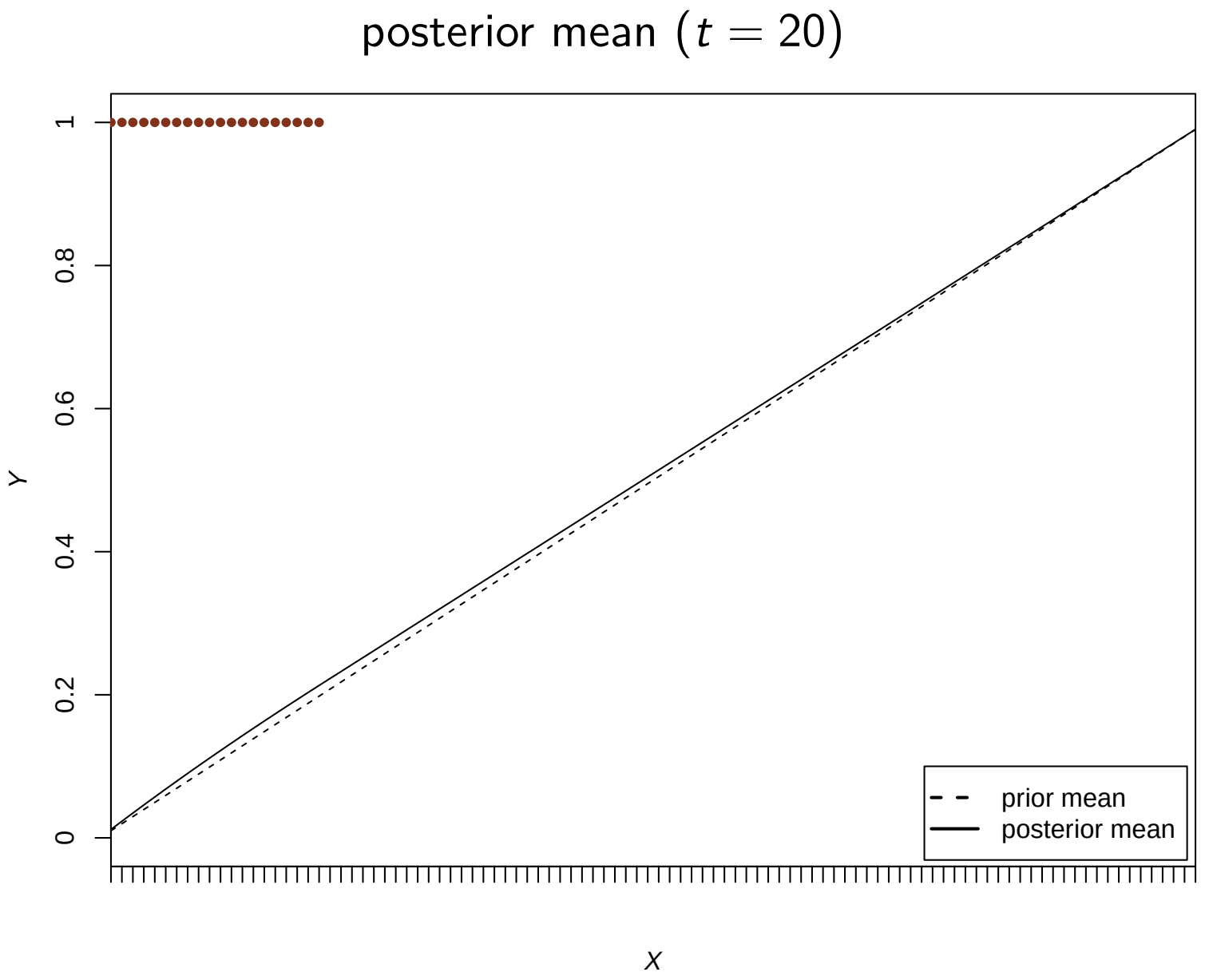
Exponential Weights with uniform prior does not learn



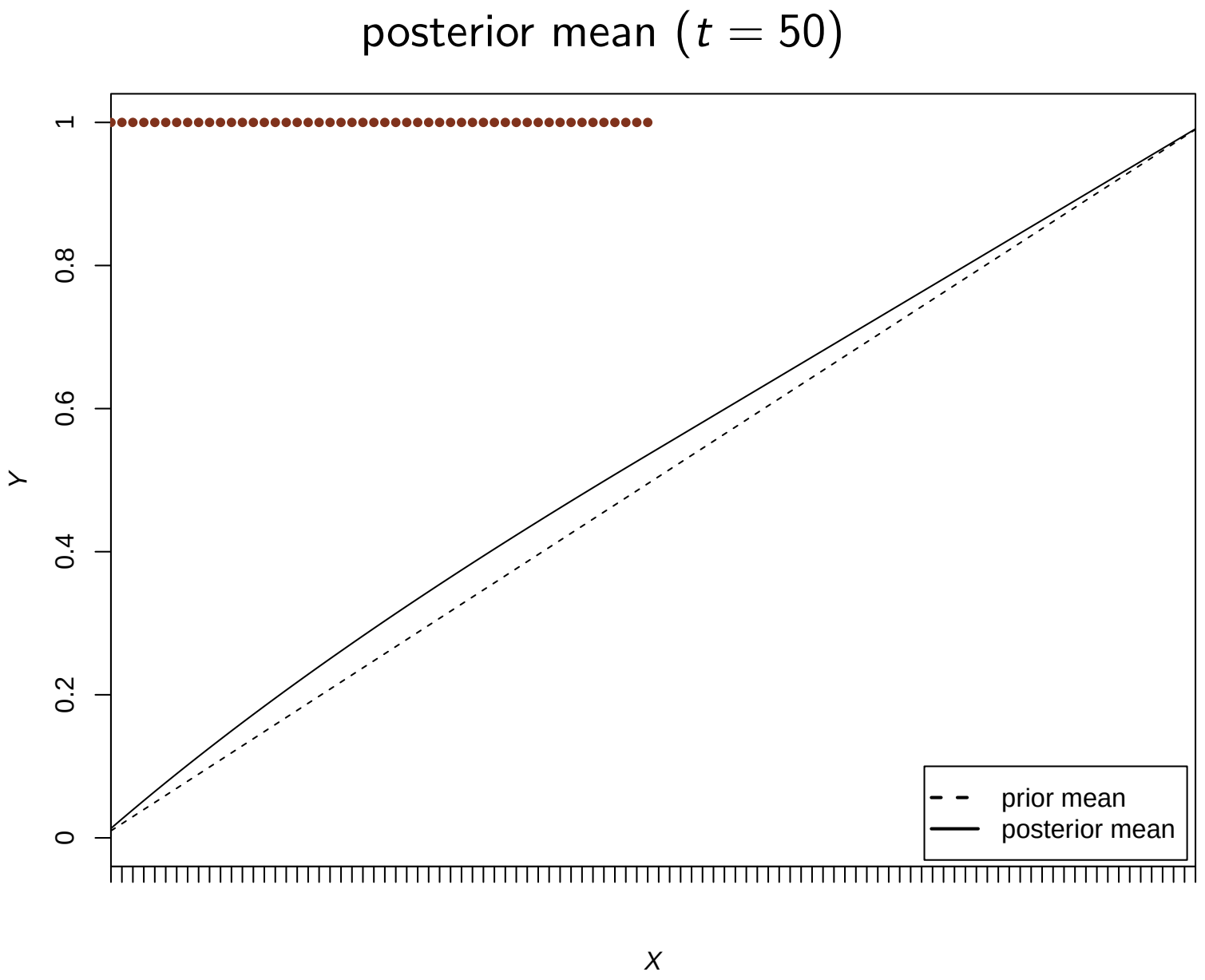
Exponential Weights with uniform prior does not learn



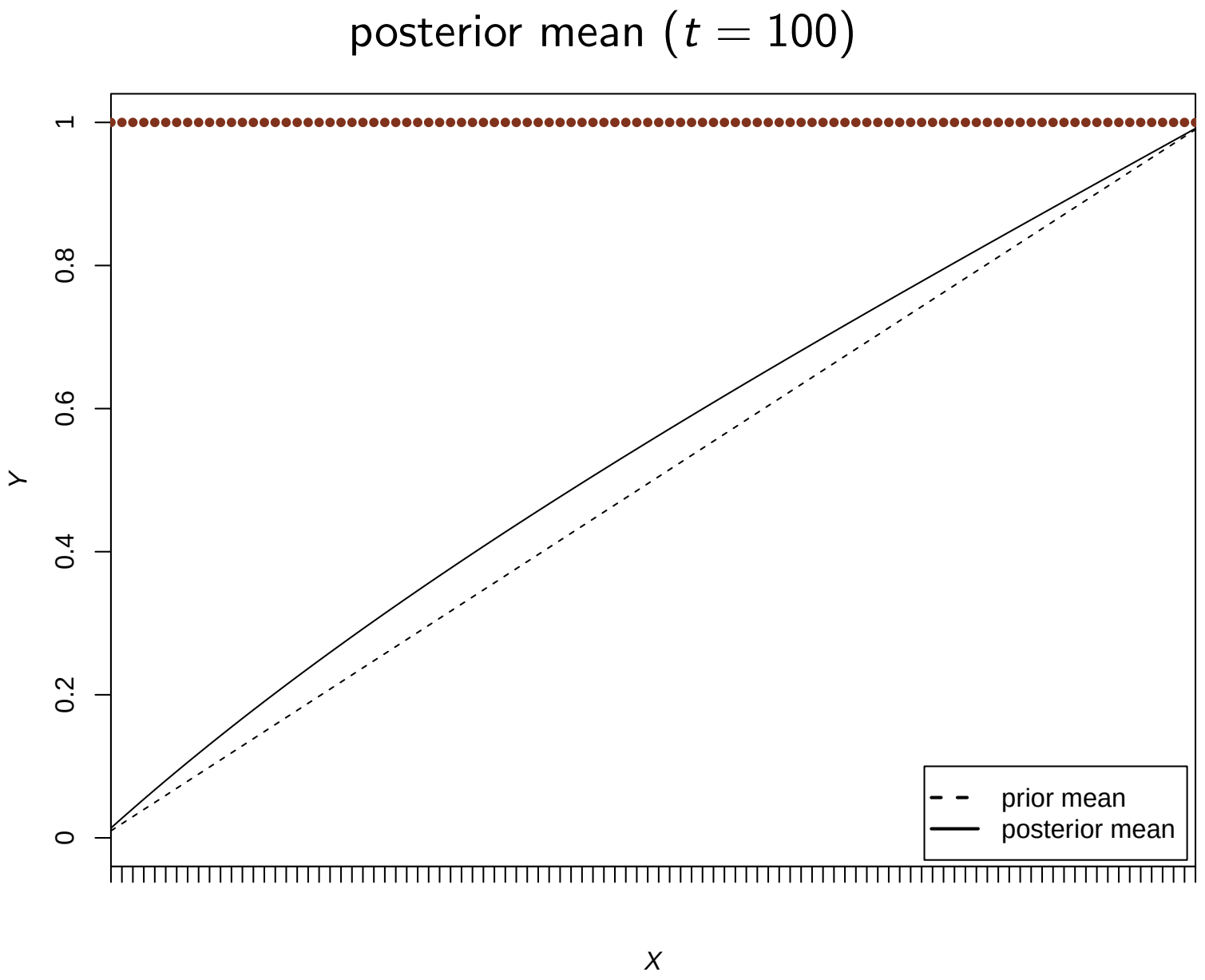
Exponential Weights with uniform prior does not learn



Exponential Weights with uniform prior does not learn



Exponential Weights with uniform prior does not learn



The algorithm

Exponential Weights on a covering net

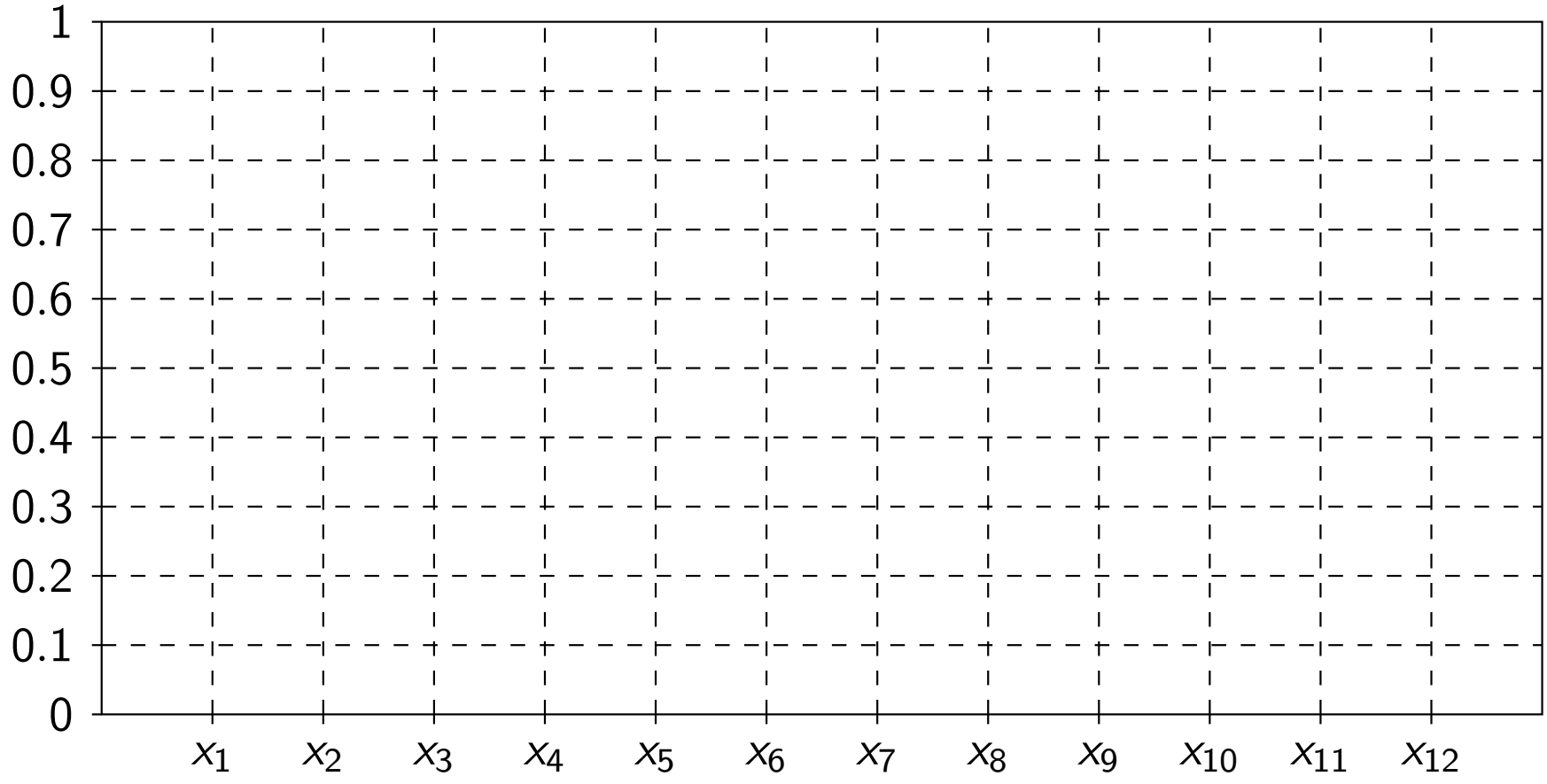
$$\mathcal{F}_K = \left\{ \boldsymbol{f} : f_t = \frac{k_t}{K}, k \in \{0, 1, \ldots, K\}, \quad f_1 \leq \ldots \leq f_T \right\},$$

$$\pi(\boldsymbol{f}) \text{ uniform on } \mathcal{F}_K.$$

- Efficient implementation by dynamic programming: $O(Kt)$ at trial t.
- Speed-up to $O(K)$ if the data revealed in isotonic order.

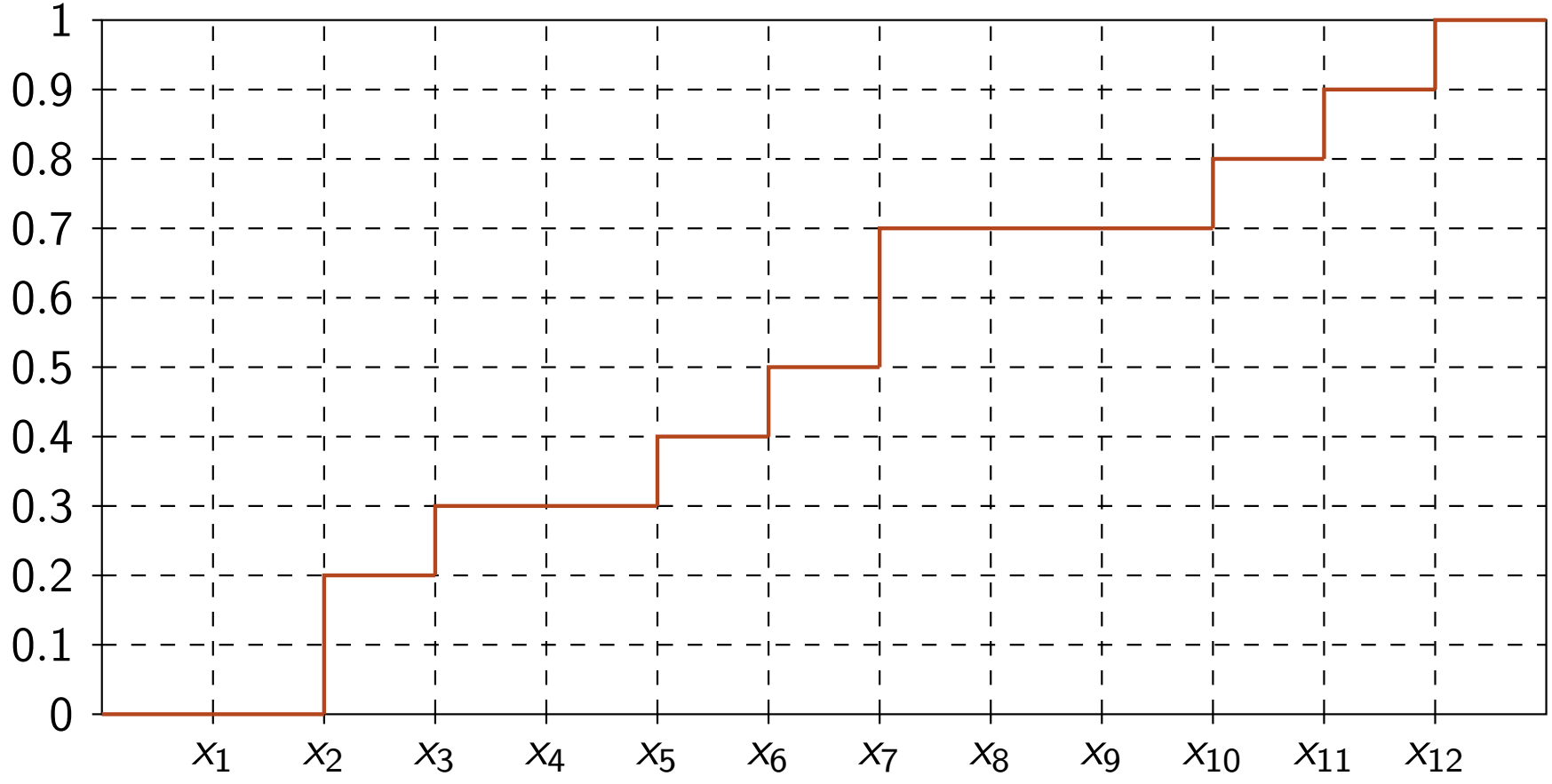
Covering net

A finite set of isotonic functions on a discrete grid of y values.



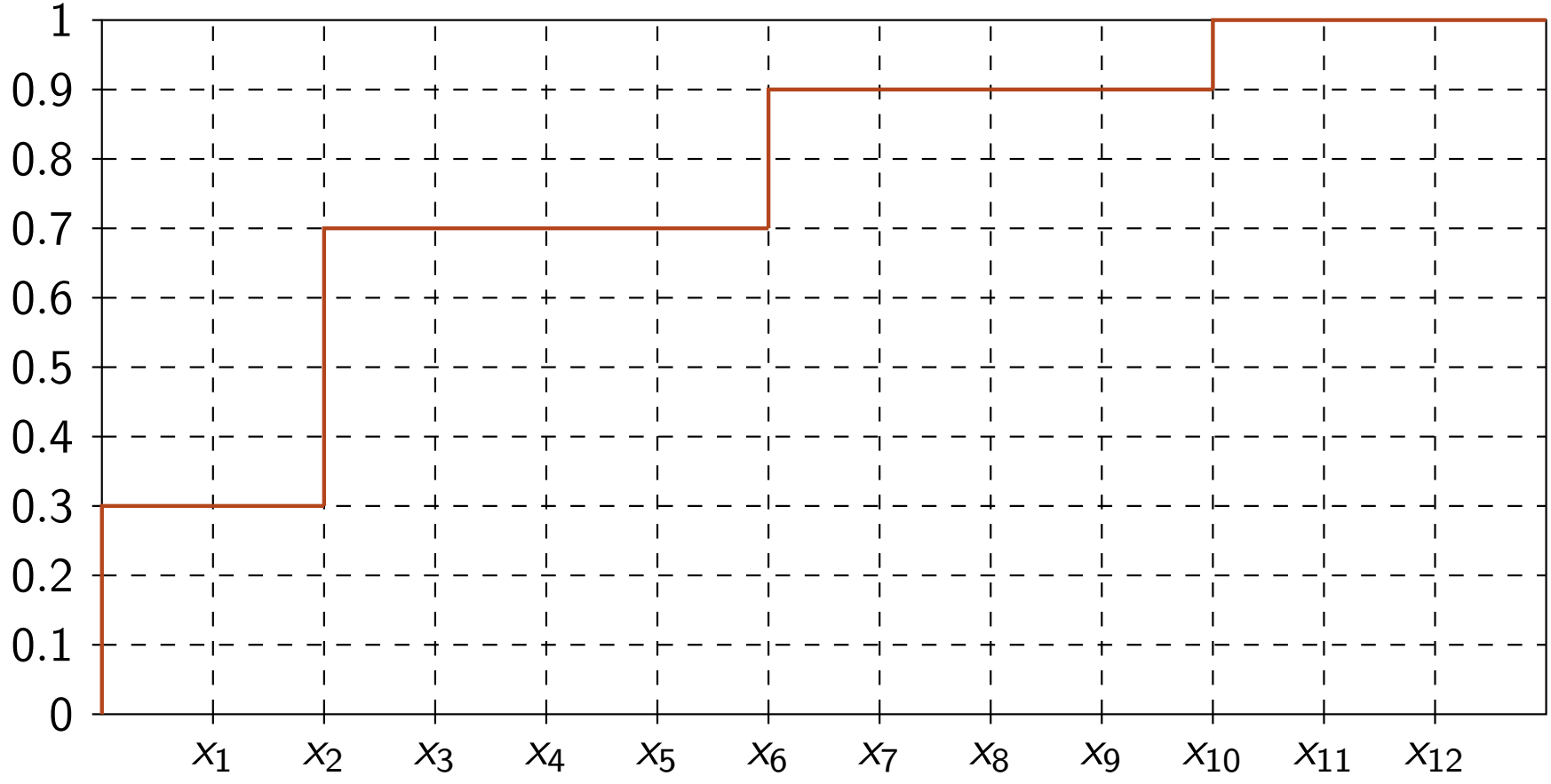
Covering net

A finite set of isotonic functions on a discrete grid of y values.



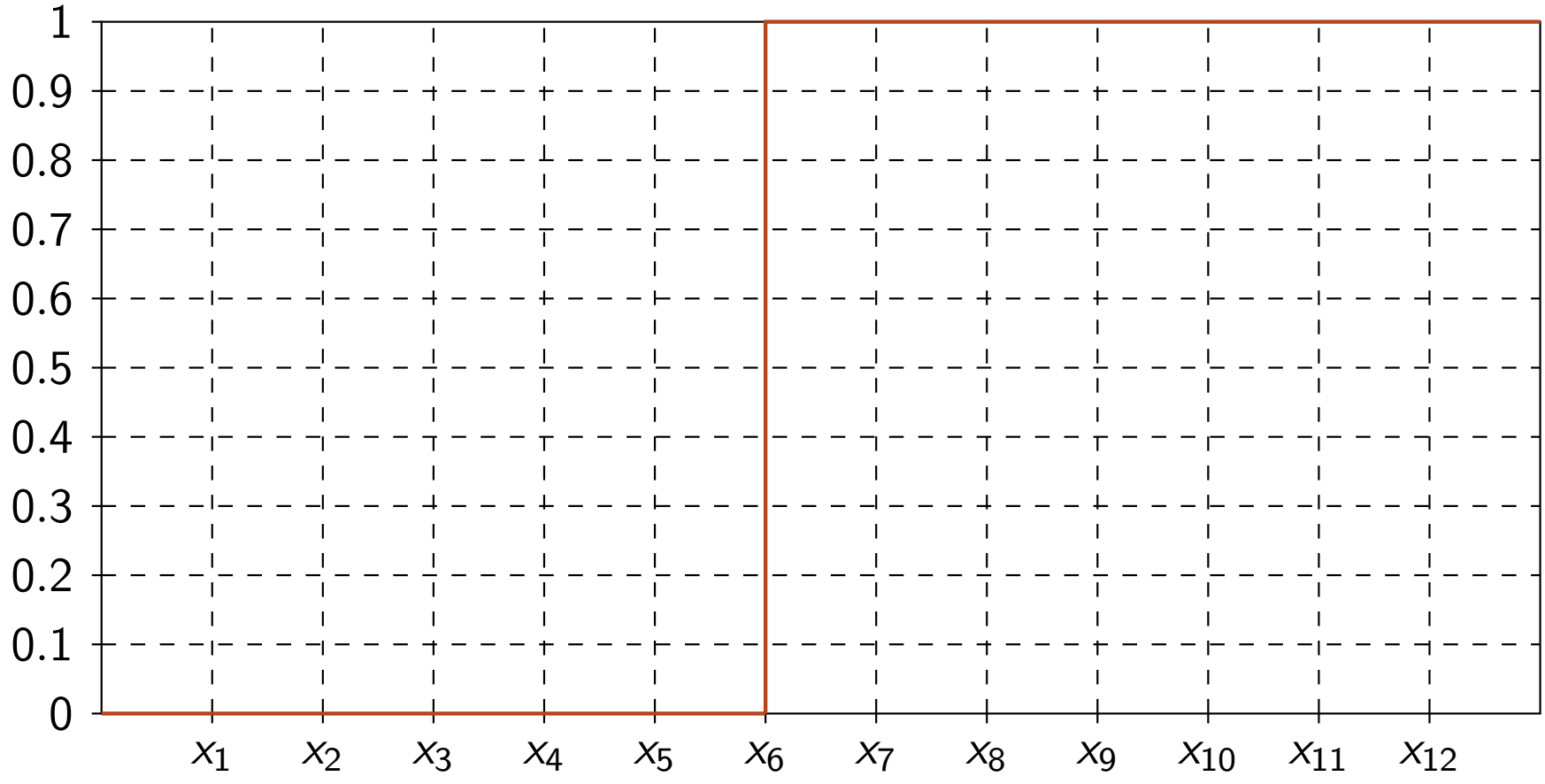
Covering net

A finite set of isotonic functions on a discrete grid of y values.



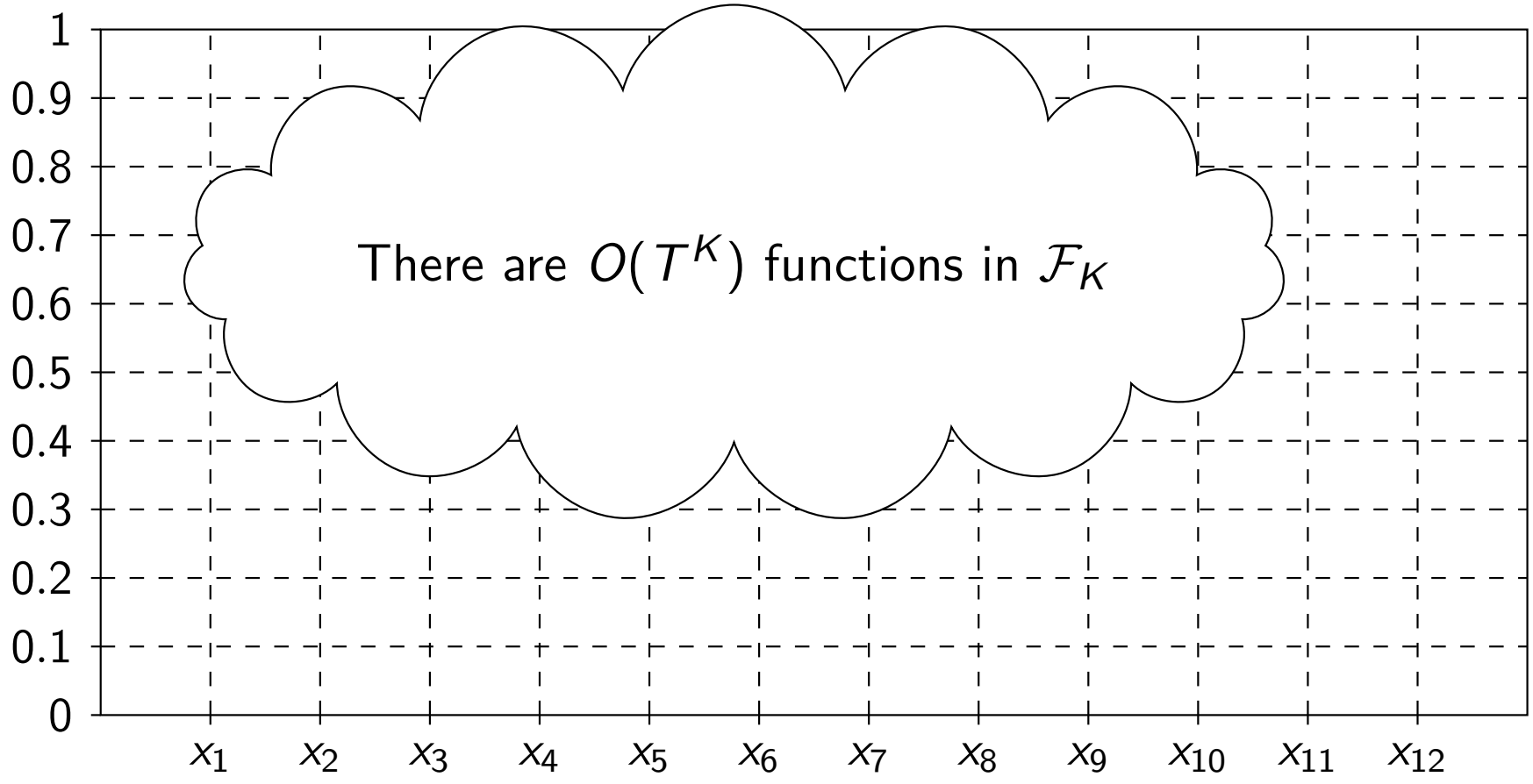
Covering net

A finite set of isotonic functions on a discrete grid of y values.



Covering net

A finite set of isotonic functions on a discrete grid of y values.



Performance of the algorithm

Regret bound

When $K = \Theta\left(T^{1/3}\log^{-1/3}(T)\right)$,

$$\text{Regret} = O\left(T^{1/3}\log^{2/3}(T)\right)$$

Performance of the algorithm

Regret bound

When $K = \Theta\left(T^{1/3}\log^{-1/3}(T)\right)$,

$$\text{Regret} = O\left(T^{1/3}\log^{2/3}(T)\right)$$

- Matching lower bound $\Omega(T^{1/3})$ (up to log factor).

Performance of the algorithm

Regret bound

When $K = \Theta\left(T^{1/3}\log^{-1/3}(T)\right)$,

$$\text{Regret} = O\left(T^{1/3}\log^{2/3}(T)\right)$$

- Matching lower bound $\Omega(T^{1/3})$ (up to log factor).

Proof idea

$$\text{Regret} = \text{Loss(alg)} - \min_{\boldsymbol{f}\in\mathcal{F}_K} \text{Loss}(\boldsymbol{f})$$

$$+ \min_{\boldsymbol{f}\in\mathcal{F}_K} \text{Loss}(\boldsymbol{f}) - \min_{\text{isotonic } f} \text{Loss}(f)$$

Performance of the algorithm

Regret bound

When $K = \Theta\left(T^{1/3}\log^{-1/3}(T)\right)$,

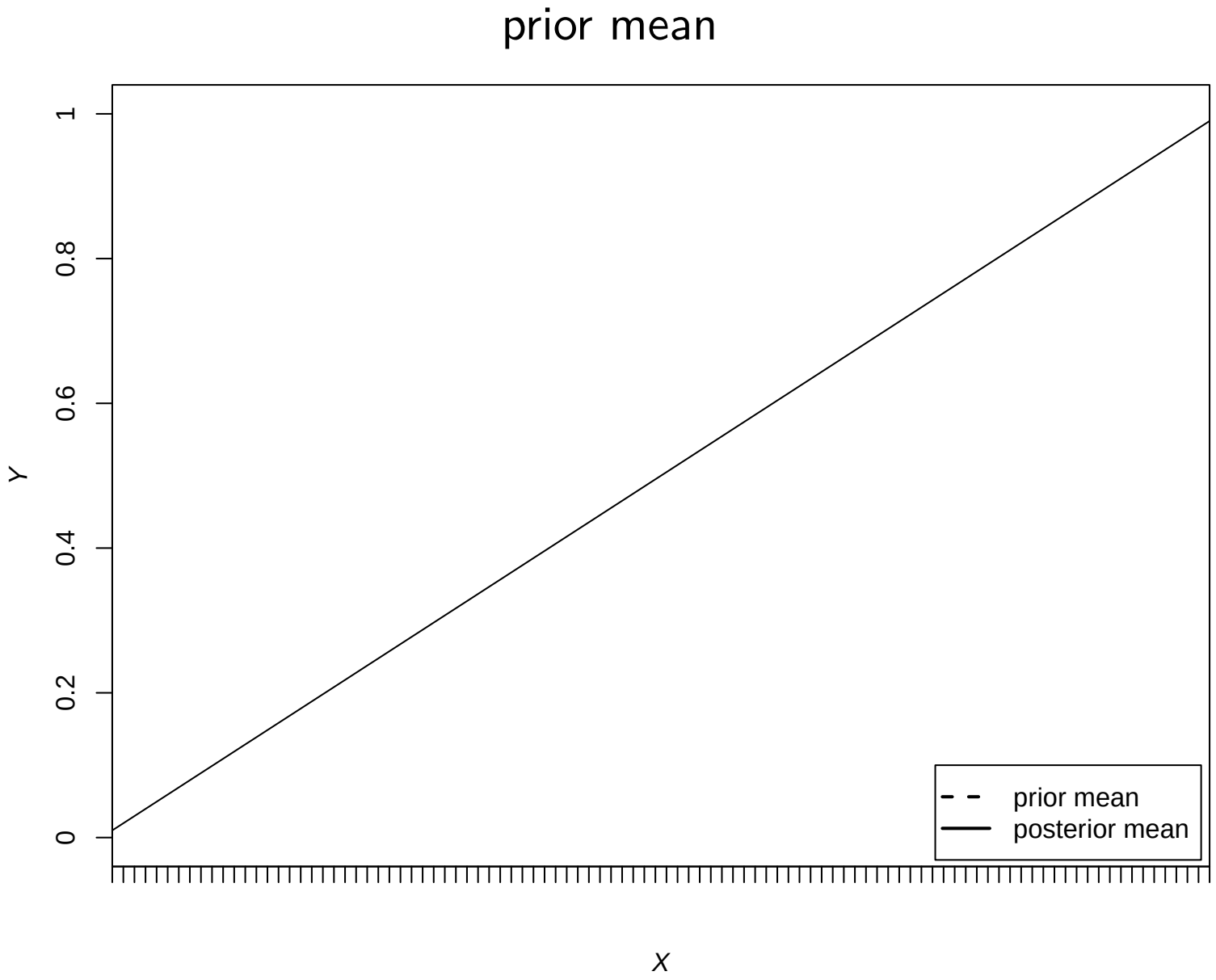
$$\text{Regret} = O\left(T^{1/3}\log^{2/3}(T)\right)$$

- Matching lower bound $\Omega(T^{1/3})$ (up to log factor).

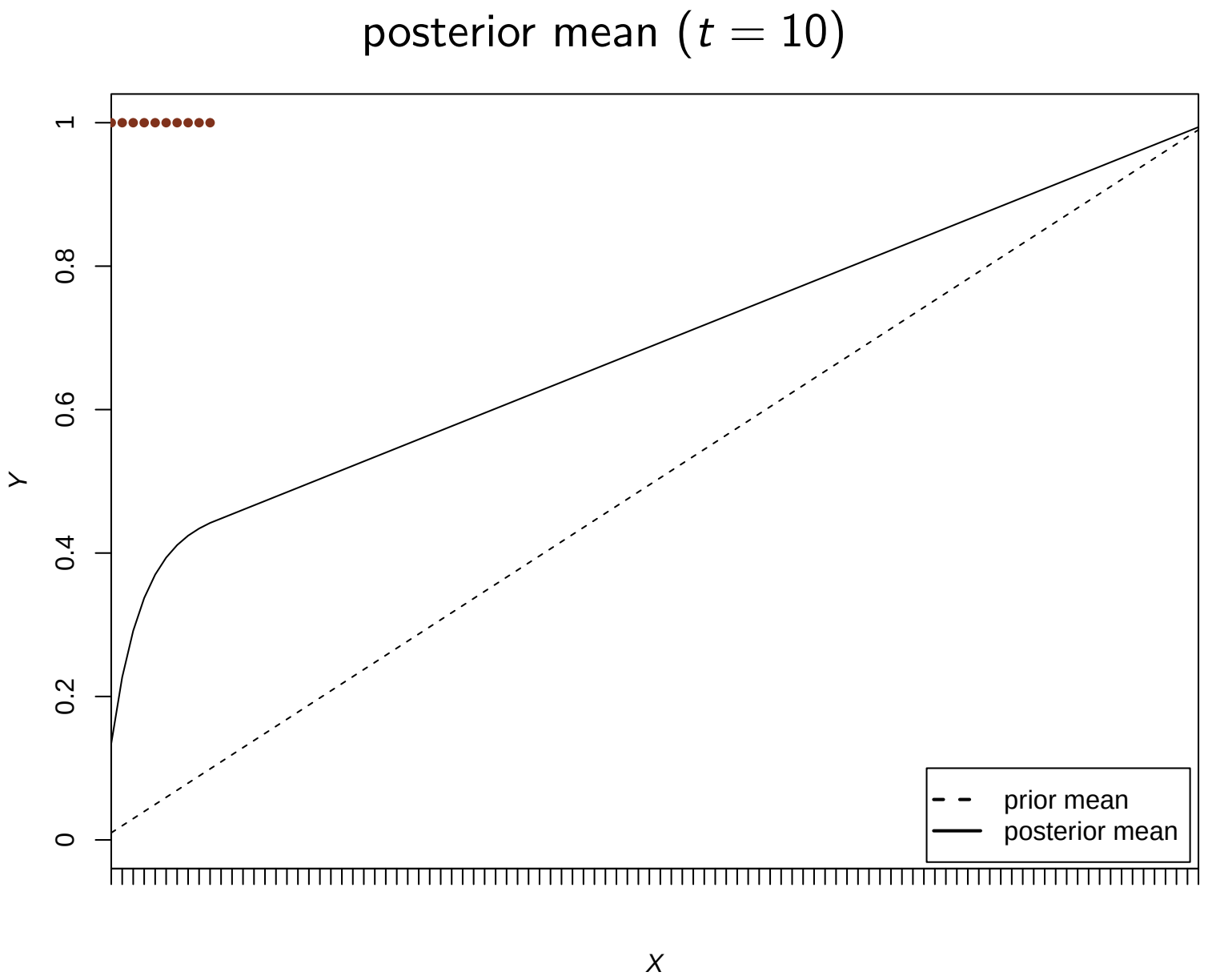
Proof idea

$$\text{Regret} = \underbrace{\text{Loss(alg)} - \min_{\boldsymbol{f}\in\mathcal{F}_K}\text{Loss}(\boldsymbol{f})}_{=2\log|\mathcal{F}_K|=O(K\log T)} + \underbrace{\min_{\boldsymbol{f}\in\mathcal{F}_K}\text{Loss}(\boldsymbol{f}) - \min_{\text{isotonic } f}\text{Loss}(f)}_{=\frac{T}{4K^2}}$$

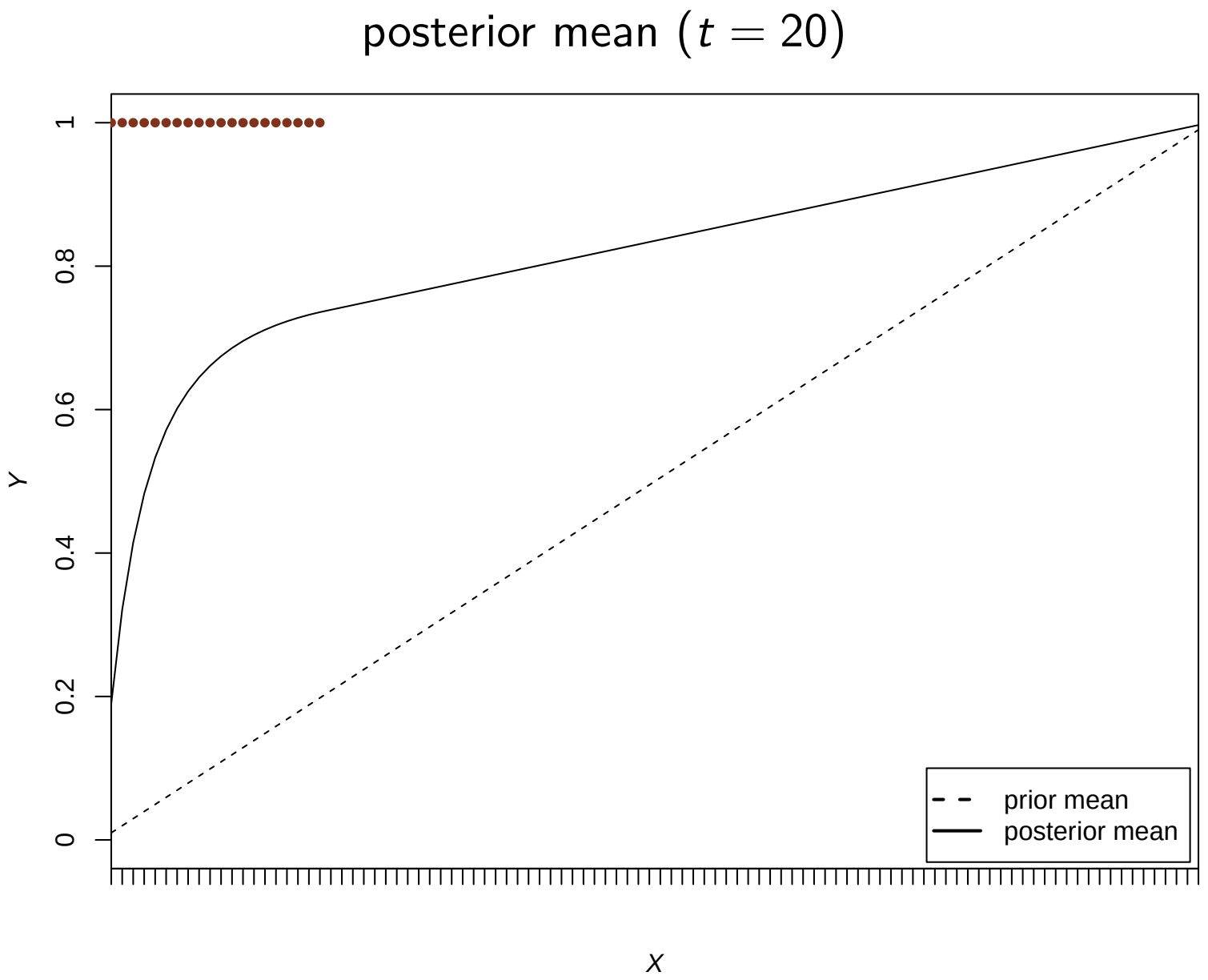
Performance of the algorithm



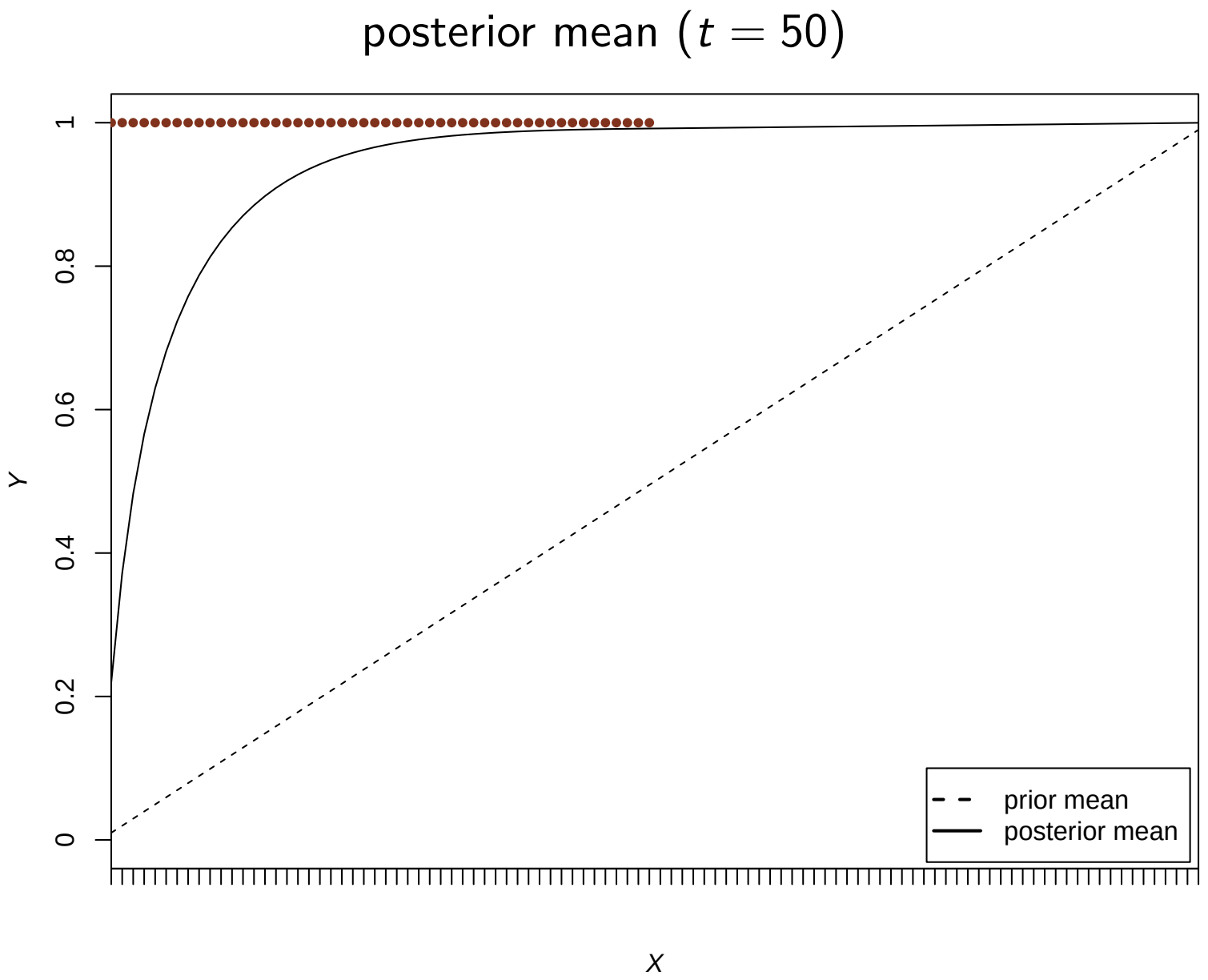
Performance of the algorithm



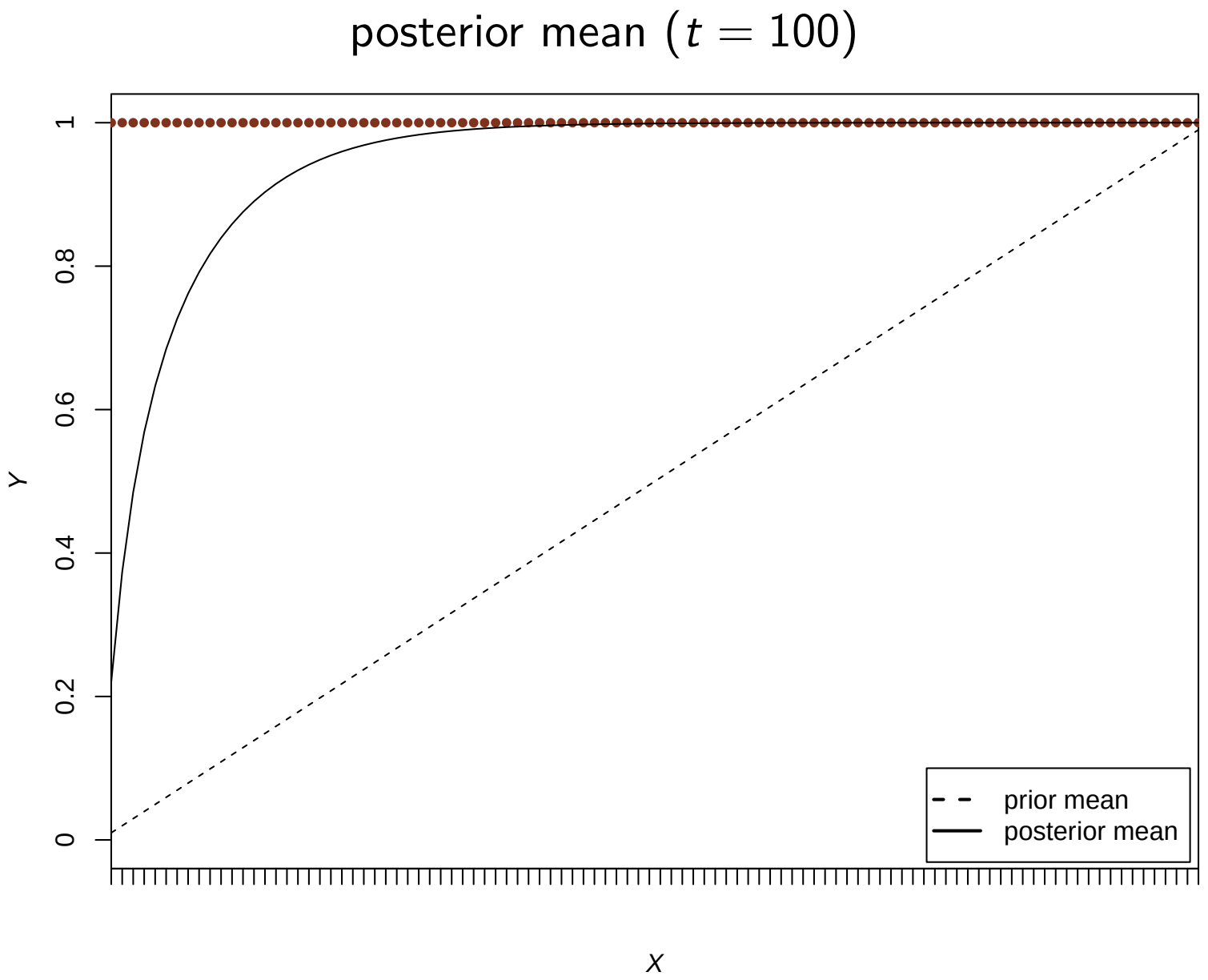
Performance of the algorithm



Performance of the algorithm



Performance of the algorithm



Other loss functions

Cross-entropy loss

$$\ell(y, \widehat{y}) = -y \log \widehat{y} - (1 - y) \log(1 - \widehat{y})$$

- The same bound $O\left(T^{1/3} \log^{2/3}(T)\right)$.
- Covering net $\mathcal{F}_K$ obtained by non-uniform discretization.

Other loss functions

Cross-entropy loss

$$\ell(y, \widehat{y}) = -y \log \widehat{y} - (1 - y) \log(1 - \widehat{y})$$

- The same bound $O\left(T^{1/3} \log^{2/3}(T)\right)$.
- Covering net $\mathcal{F}_K$ obtained by non-uniform discretization.

Absolute loss

$$\ell(y, \widehat{y}) = |y - \widehat{y}|$$

- $O\left(\sqrt{T \log T}\right)$ obtained by Exponentiated Gradient.
- Matching lower bound $\Omega(\sqrt{T})$ (up to log factor).

Outline

Random permutation model

A more realistic scenario for generating $x_1, \ldots, x_T$ which allows data to be unknown in advance.

Random permutation model

A more realistic scenario for generating $x_1, \ldots, x_T$ which allows data to be unknown in advance.

The data are chosen adversarially before the game begins, but then are presented to the learner in a random order

- Motivation: data gathering process is independent on the underlying data generation mechanism.
- Still very weak assumption.
- Evaluation: regret averaged over all permutations of data:

$$\mathbb{E}_\sigma \left[\text{regret}_T\right]$$

K., Koolen, Malek: *Random Permutation Online Isotonic Regression*. Submitted, 2017.

Leave-one-out loss

Definition

Given t labeled points $\{(x_i, y_i)\}_{i=1}^t$, for $i = 1, \ldots, t$:

- Take out i-th point and give remaining $t-1$ points to the learner as a training data.
- Learner predict $\widehat{y}_i$ on x_i and receives loss $\ell(y_i, \widehat{y}_i)$.

Evaluate the learner by $\ell oo_t = \frac{1}{t} \sum_{i=1}^t \ell(y_i, \widehat{y}_i)$

No sequential structure in the definition.

Leave-one-out loss

Definition

Given t labeled points $\{(x_i, y_i)\}_{i=1}^t$, for $i = 1, \ldots, t$:

- Take out i-th point and give remaining $t-1$ points to the learner as a training data.
- Learner predict $\widehat{y}_i$ on x_i and receives loss $\ell(y_i, \widehat{y}_i)$.

Evaluate the learner by $\ell oo_t = \frac{1}{t} \sum_{i=1}^t \ell(y_i, \widehat{y}_i)$

No sequential structure in the definition.

Theorem

If $\ell oo_t \leq g(t)$ for all t, then $\mathbb{E}_\sigma \left[\text{regret}_T\right] \leq \sum_{t=1}^T g(t)$.

Fixed design to random permutation conversion

Any algorithm for fixed-design can be used in the random permutation setup by being re-run from the scratch in each trial.

We have shown that:

$$\ell oo_t \leq \frac{1}{t}\mathbb{E}_\sigma\left[\text{fixed-design-regret}_t\right]$$

We thus get an optimal algorithm (Exponential Weights on a grid) with $\widetilde{O}(T^{-2/3})$ leave-one-out loss "for free", but it is complicated.

Can we get simpler algorithms to work in this setup?

Follow the Leader (FTL) algorithm

Definition

Given past $t-1$ data, compute the optimal (loss-minimizing) function f^* and predict on new instance x according to $f^*(x)$.

Follow the Leader (FTL) algorithm

Definition

Given past $t-1$ data, compute the optimal (loss-minimizing) function f^* and predict on new instance x according to $f^*(x)$.

FTL is undefined for isotonic regression.

x	-3	-1		2	3
y	0	0.2		0.7	1
$f^*(x)$	0	0.2		0.7	1

Follow the Leader (FTL) algorithm

Definition

Given past $t-1$ data, compute the optimal (loss-minimizing) function f^* and predict on new instance x according to $f^*(x)$.

FTL is undefined for isotonic regression.

x	-3	-1	0	2	3
y	0	0.2		0.7	1
$f^*(x)$	0	0.2	??	0.7	1

Foward Algorithm (FA)

Definition

Given past $t-1$ data and a new instance x, take any guess $y' \in [0,1]$ of the new label and predict according to the optimal function f^* on the past data including the new point (x, y').

x	-3	-1	0	2	3
y	0	0.2		0.7	1
$f^*(x)$					

Foward Algorithm (FA)

Definition

Given past $t-1$ data and a new instance x, take any guess $y' \in [0,1]$ of the new label and predict according to the optimal function f^* on the past data including the new point (x, y').

x	-3	-1	0	2	3
y	0	0.2	$y' = 1$	0.7	1
$f^*(x)$					

Foward Algorithm (FA)

Definition

Given past $t-1$ data and a new instance x, take any guess $y' \in [0,1]$ of the new label and predict according to the optimal function f^* on the past data including the new point (x, y').

x	-3	-1	0	2	3
y	0	0.2	$y' = 1$	0.7	1
$f^*(x)$	0	0.2	0.85	0.85	1

Foward Algorithm (FA)

Definition

Given past $t-1$ data and a new instance x, take any guess $y' \in [0,1]$ of the new label and predict according to the optimal function f^* on the past data including the new point (x, y').

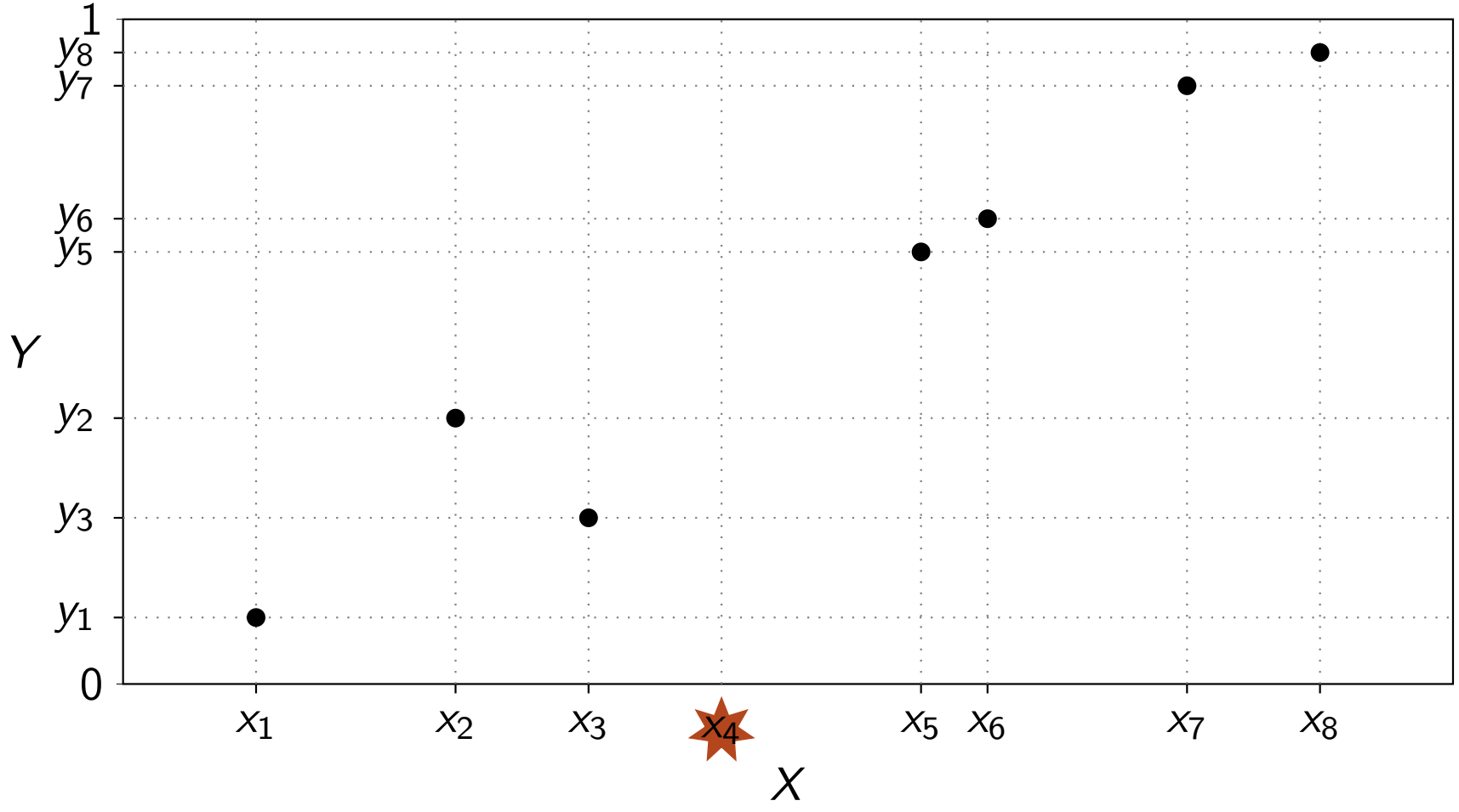
x	-3	-1	0	2	3
y	0	0.2	$y'=1$	0.7	1
$f^*(x)$	0	0.2	0.85	0.85	1

Various popular prediction algorithms for IR fall into this framework (including linear interpolation [Zadrozny & Elkan, 2002] and many others [Vovk et al., 2015]).

Foward Algorithm (FA)

Two extreme FA: guess-1 and guess-0, denoted f_1^* and f_0^*.

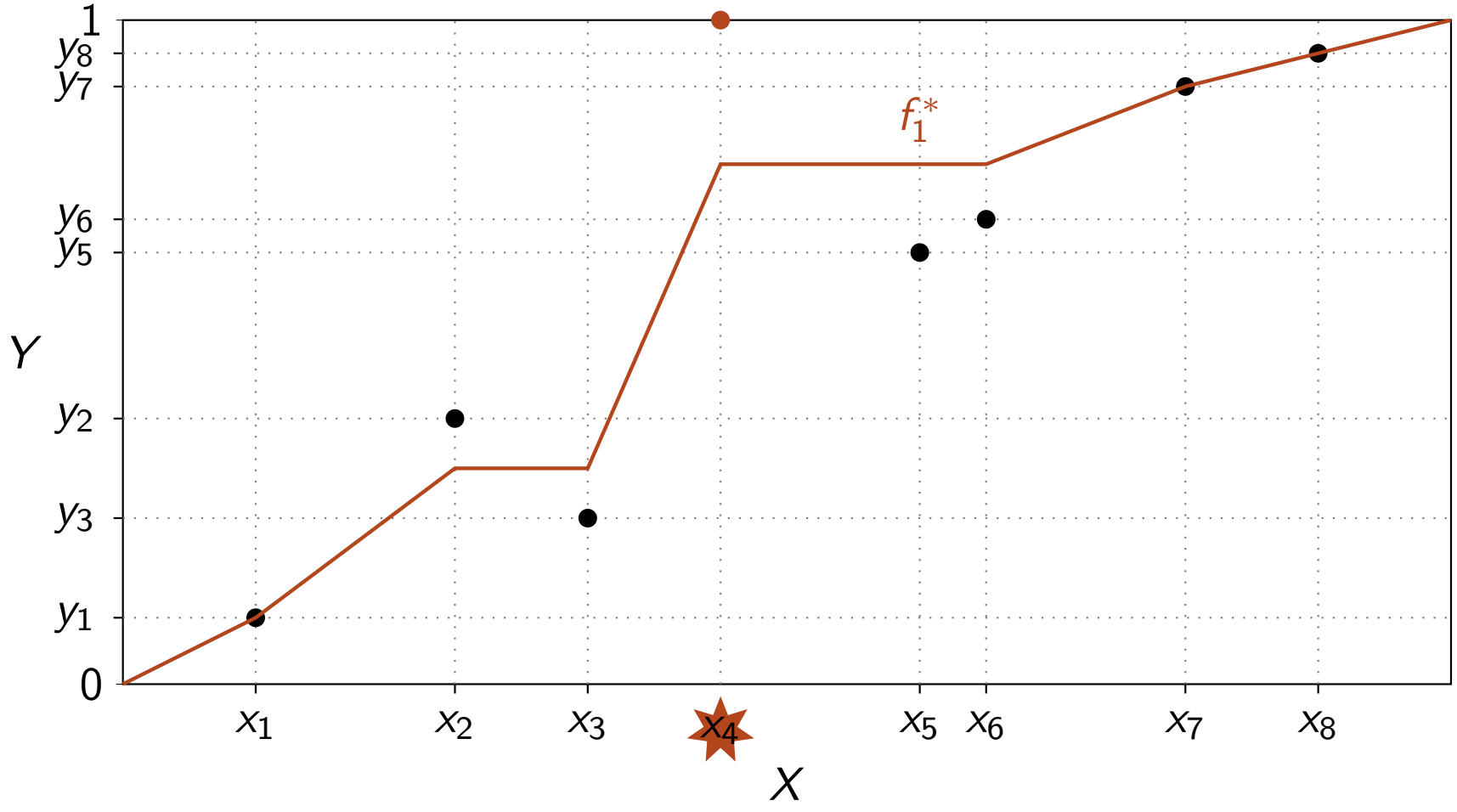
Prediction of any FA is always between: $f_0^*(x) \leq f^*(x) \leq f_1^*(x)$.



Foward Algorithm (FA)

Two extreme FA: guess-1 and guess-0, denoted f_1^* and f_0^*.

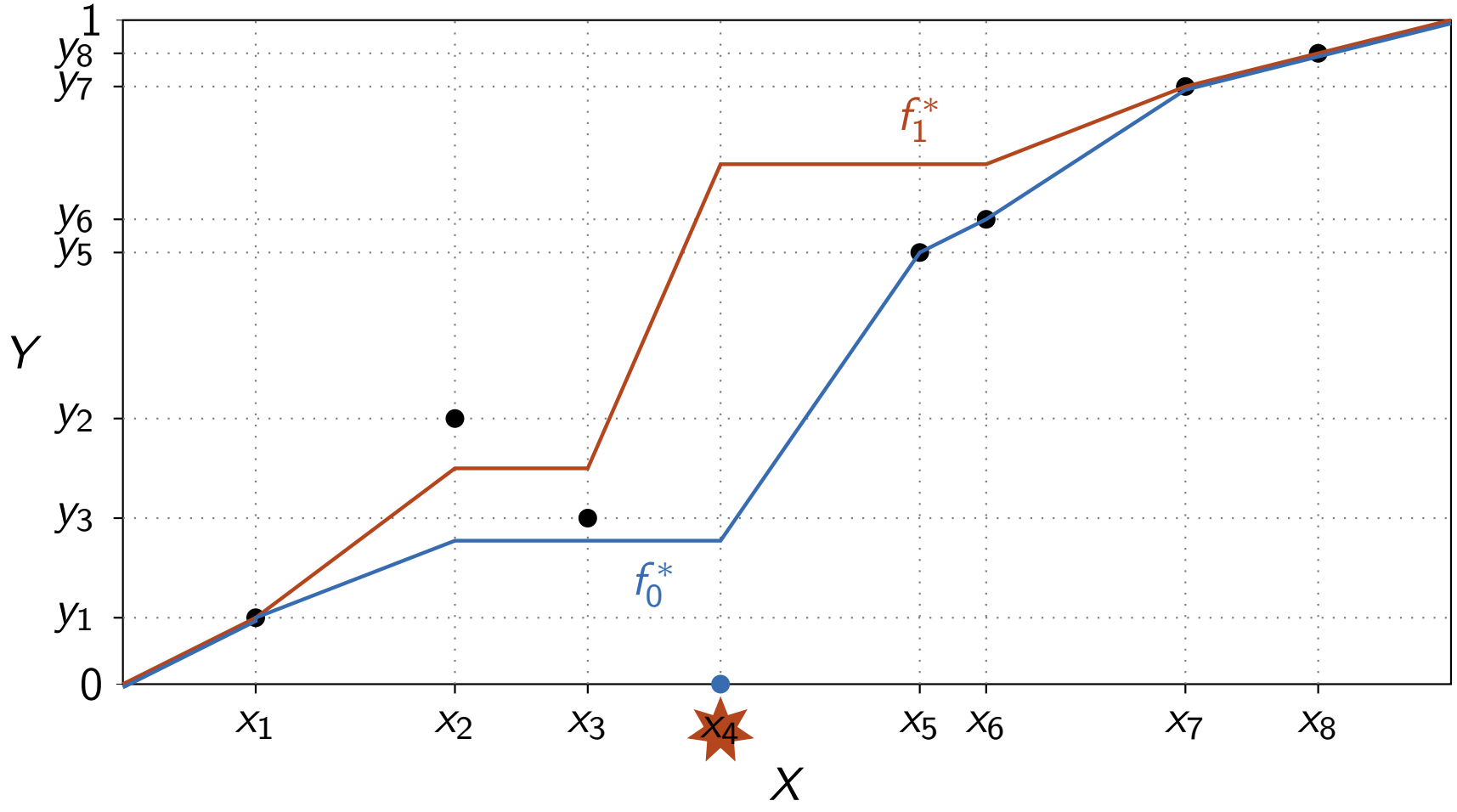
Prediction of any FA is always between: $f_0^*(x) \leq f^*(x) \leq f_1^*(x)$.



Foward Algorithm (FA)

Two extreme FA: guess-1 and guess-0, denoted f_1^* and f_0^*.

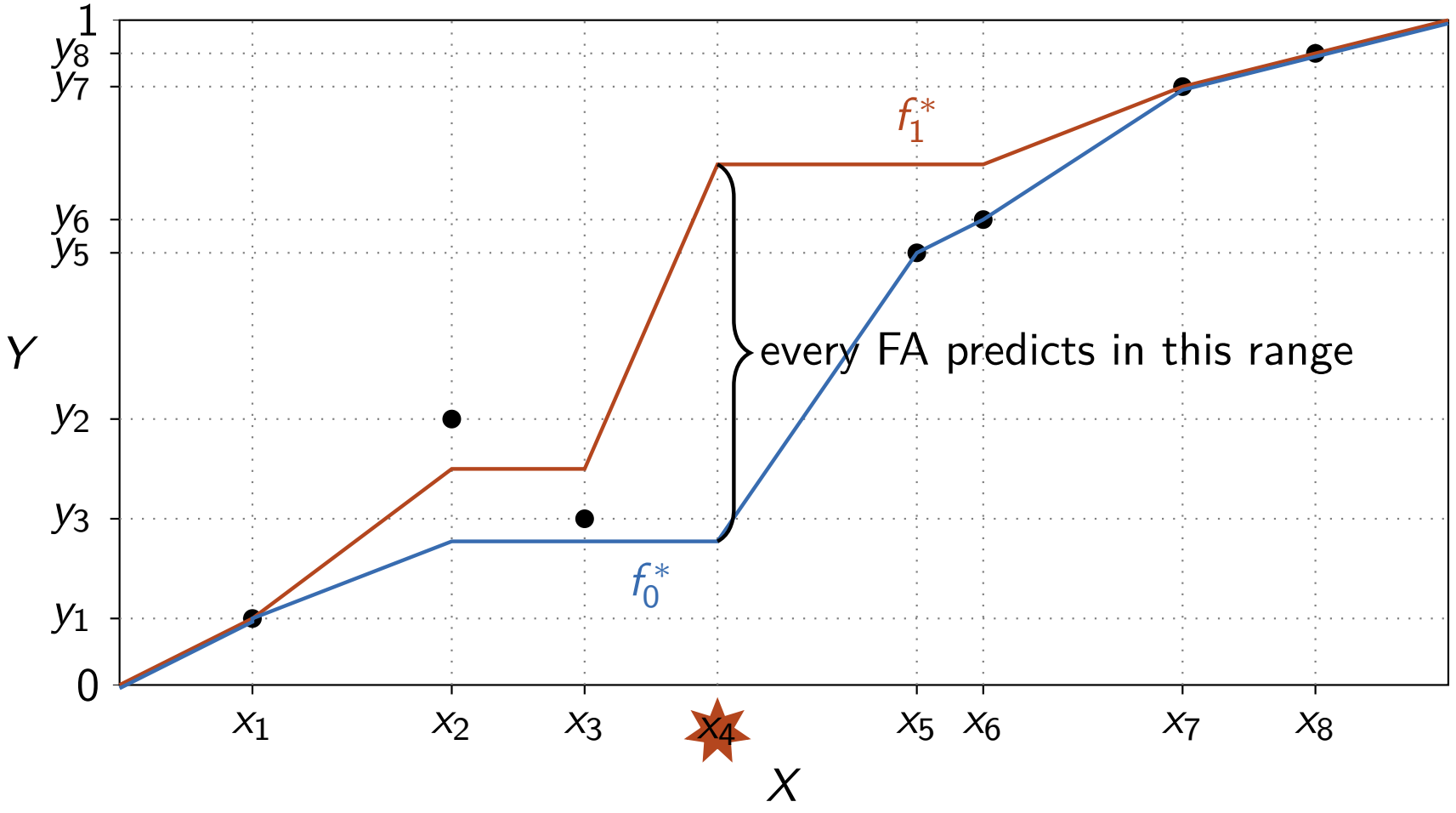
Prediction of any FA is always between: $f_0^*(x) \leq f^*(x) \leq f_1^*(x)$.



Foward Algorithm (FA)

Two extreme FA: guess-1 and guess-0, denoted f_1^* and f_0^*.

Prediction of any FA is always between: $f_0^*(x) \leq f^*(x) \leq f_1^*(x)$.



Performance of FA

Theorem

For squared loss, every forward algorithm has:

$$\ell oo_t = O\left(\sqrt{\frac{\log t}{t}}\right)$$

- The bound is suboptimal, but only a factor of $O(t^{1/6})$ off.
- For cross-entropy loss, the some bound holds but a more careful choice of the guess must be made.

Outline

Conclusions

- Two models for online isotonic regression: fixed design and random permutation.
- Optimal algorithm in both models: Exponential Weights (Bayes) on a grid.
- In the random permutation model, a class of forward algorithms with good bounds on the leave-one-out loss.

Open problem:
Extend any of these algorithms to the partial order case.

Bibliography

Statistics

M. Ayer, H. D. Brunk, G. M. Ewing, W. T. Reid, and E. Silverman. An empirical distribution function for sampling with incomplete information. *Annals of Mathematical Statistics*, 26(4):641–647, 1955

H. D. Brunk. Maximum likelihood estimates of monotone parameters. *Annals of Mathematical Statistics*, 26(4):607–616, 1955

J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964

R. E. Barlow and H. D. Brunk. The isotonic regression problem and its dual. *Journal of the American Statistical Association*, 67:140–147, 1972

T. Robertson, F. T. Wright, and R. L. Dykstra. *Order Restricted Statistical Inference*. John Wiley & Sons, 1998

Sara Van de Geer. Estimating a regression function. *Annals of Statistics*, 18:907–924, 1990

Cun-Hui Zhang. Risk bounds in isotonic regression. *The Annals of Statistics*, 30(2):528–555, 2002

Jan de Leeuw, Kurt Hornik, and Patrick Mair. Isotone optimization in R: Pool-adjacent-violators algorithm (PAVA) and active set methods. *Journal of Statistical Software*, 32:1–24, 2009

Bibliography

Machine Learning

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *KDD*, pages 694–699, 2002

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *ICML*, volume 119, pages 625–632. ACM, 2005

Tom Fawcett and Alexandru Niculescu-Mizil. PAV and the ROC convex hull. *Machine Learning*, 68(1):97–106, 2007

Vladimir Vovk, Ivan Petej, and Valentina Fedorova. Large-scale probabilistic predictors with and without guarantees of validity. In *NIPS*, pages 892–900, 2015

Aditya Krishna Menon, Xiaoqian Jiang, Shankar Vembu, Charles Elkan, and Lucila Ohno-Machado. Predicting accurate probabilities with a ranking loss. In *ICML*, 2012

Rasmus Kyng, Anup Rao, and Sushant Sachdeva. Fast, provable algorithms for isotonic regression in all ℓ_p-norms. In *NIPS*, 2015

Adam Tauman Kalai and Ravi Sastry. The isotron algorithm: High-dimensional isotonic regression. In *COLT*, 2009

T. Moon, A. Smola, Y. Chang, and Z. Zheng. Intervalrank: Isotonic regression with listwise and pairwise constraint. In *WSDM*, pages 151–160. ACM, 2010

Sham M Kakade, Varun Kanade, Ohad Shamir, and Adam Kalai. Efficient learning of generalized linear and single index models with isotonic regression. In *NIPS*, pages 927–935, 2011

Bibliography

Online isotonic regression

Alexander Rakhlin and Karthik Sridharan. Online nonparametric regression. In *COLT*, pages 1232–1264, 2014

Pierre Gaillard and Sébastien Gerchinovitz. A chaining algorithm for online nonparametric regression. In *COLT*, pages 764–796, 2015

Wojciech Kotłowski, Wouter M. Koolen, and Alan Malek. Online isotonic regression. In *COLT*, pages 1165–1189, 2016

Wojciech Kotłowski, Wouter M. Koolen, and Alan Malek. Random permutation online isotonic regression. submitted, 2017