COMS 4771 Lecture 26

1. Online learning

A Sequential Prediction Task

- Over a sequence of T days, you want to predict whether it will be sunny or rainy

A Sequential Prediction Task

- Over a sequence of T days, you want to predict whether it will be sunny or rainy
- You have access to N "experts": `weather.com`, `wunderground.com`, ..., groundhog
- Each expert makes it's own prediction, sunny or rainy

A Sequential Prediction Task

- Over a sequence of T days, you want to predict whether it will be sunny or rainy
- You have access to N "experts": `weather.com`, `wunderground.com`, ..., groundhog
- Each expert makes it's own prediction, sunny or rainy
- Based on their prediction, you make your own prediction

A Sequential Prediction Task

- Over a sequence of T days, you want to predict whether it will be sunny or rainy
- You have access to N "experts": `weather.com`, `wunderground.com`, ..., groundhog
- Each expert makes it's own prediction, sunny or rainy
- Based on their prediction, you make your own prediction
- **Question:** without knowing which expert is the best, can you predict almost as well as the best expert?

A Sequential Prediction Task

- Over a sequence of T days, you want to predict whether it will be sunny or rainy
- You have access to N "experts": `weather.com`, `wunderground.com`, ..., groundhog
- Each expert makes it's own prediction, sunny or rainy
- Based on their prediction, you make your own prediction
- **Question:** without knowing which expert is the best, can you predict almost as well as the best expert?
- More precisely: can you ensure that

$$\#(\text{mistakes you make}) \leq \#(\text{mistakes best expert makes}) + o(T)?$$

A Sequential Prediction Task

- Over a sequence of T days, you want to predict whether it will be sunny or rainy
- You have access to N "experts": `weather.com`, `wunderground.com`, ..., groundhog
- Each expert makes it's own prediction, sunny or rainy
- Based on their prediction, you make your own prediction
- **Question:** without knowing which expert is the best, can you predict almost as well as the best expert?
- More precisely: can you ensure that

$$\#(\text{mistakes you make}) \leq \#(\text{mistakes best expert makes}) + o(T)?$$

- $\Rightarrow$ your error rate $\to$ best expert's error rate as $T \to \infty$.

Prediction setup

- Feature space: $\mathcal{X}$
- Label space: $\mathcal{Y}$

Prediction setup

- Feature space: $\mathcal{X}$
- Label space: $\mathcal{Y}$
- Loss function: $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. E.g.
  - Zero-one loss in classification: $\ell(y, \hat{y}) = \mathbb{1}\{\hat{y} \neq y\}$
  - Square loss in regression: $\ell(y, \hat{y}) = (\hat{y} - y)^2$
  - Absolute loss in regression: $\ell(y, \hat{y}) = |\hat{y} - y|$
- Given a labeled example $(x, y) \in \mathcal{X} \times \mathcal{Y}$, error of predicting $\hat{y}$ is measured by $\ell(y, \hat{y})$

Prediction setup

- Feature space: $\mathcal{X}$
- Label space: $\mathcal{Y}$
- Loss function: $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. E.g.
  - Zero-one loss in classification: $\ell(y, \hat{y}) = \mathbb{1}\{\hat{y} \neq y\}$
  - Square loss in regression: $\ell(y, \hat{y}) = (\hat{y} - y)^2$
  - Absolute loss in regression: $\ell(y, \hat{y}) = |\hat{y} - y|$
- Given a labeled example $(x, y) \in \mathcal{X} \times \mathcal{Y}$, error of predicting $\hat{y}$ is measured by $\ell(y, \hat{y})$
- Class $\mathcal{F}$ of functions $f : \mathcal{X} \to \mathcal{Y}$ (e.g. decision trees, nearest-neighbor classifiers, neural networks, linear predictor for regression, etc.)
- Informally: *Learning* over $\mathcal{F}$ means making predictions as well as the best predictor in $\mathcal{F}$.

Online learning Framework

Online learning algorithm

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

Online learning algorithm

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

1. Observe data point $x_t \in \mathcal{X}$.

Online learning algorithm

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

1. Observe data point $x_t \in \mathcal{X}$.
2. Choose prediction $\hat{y}_t \in \mathcal{Y}$.

Online learning algorithm

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

1. Observe data point $x_t \in \mathcal{X}$.
2. Choose prediction $\hat{y}_t \in \mathcal{Y}$.
3. Observe label $y_t \in \mathcal{Y}$.

Online learning algorithm

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

1. Observe data point $x_t \in \mathcal{X}$.
2. Choose prediction $\hat{y}_t \in \mathcal{Y}$.
3. Observe label $y_t \in \mathcal{Y}$.
4. Suffer loss $\ell(y_t, \hat{y}_t)$ (e.g., $\ell(y_t, \hat{y}_t) = \mathbb{1}\{\hat{y}_t \neq y_t\}$).

Online learning Framework

Online learning algorithm

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

1. Observe data point $x_t \in \mathcal{X}$.
2. Choose prediction $\hat{y}_t \in \mathcal{Y}$.
3. Observe label $y_t \in \mathcal{Y}$.
4. Suffer loss $\ell(y_t, \hat{y}_t)$ (e.g., $\ell(y_t, \hat{y}_t) = \mathbb{1}\{\hat{y}_t \neq y_t\}$).

Underlying (non-probabilistic) assumption

The function class $\mathcal{F}$ models the data well:
i.e. best-in-hindsight (average) loss $\min_{f \in \mathcal{F}} \frac{1}{T} \sum_{t=1}^{T} \ell(y_t, f(x_t))$ is small.

Online learning Framework

Online learning algorithm

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

1. Observe data point $x_t \in \mathcal{X}$.
2. Choose prediction $\hat{y}_t \in \mathcal{Y}$.
3. Observe label $y_t \in \mathcal{Y}$.
4. Suffer loss $\ell(y_t, \hat{y}_t)$ (e.g., $\ell(y_t, \hat{y}_t) = \mathbb{1}\{\hat{y}_t \neq y_t\}$).

Underlying (non-probabilistic) assumption

The function class $\mathcal{F}$ models the data well:
i.e. best-in-hindsight (average) loss $\min_{f \in \mathcal{F}} \frac{1}{T} \sum_{t=1}^{T} \ell(y_t, f(x_t))$ is small.

All "data" (e.g., x_t, y_t) could be controlled by an adversary!

Online learning Framework

Online learning algorithm

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

1. Observe data point $x_t \in \mathcal{X}$.
2. Choose prediction $\hat{y}_t \in \mathcal{Y}$.
3. Observe label $y_t \in \mathcal{Y}$.
4. Suffer loss $\ell(y_t, \hat{y}_t)$ (e.g., $\ell(y_t, \hat{y}_t) = \mathbb{1}\{\hat{y}_t \neq y_t\}$).

Underlying (non-probabilistic) assumption

The function class $\mathcal{F}$ models the data well:
i.e. best-in-hindsight (average) loss $\min_{f \in \mathcal{F}} \frac{1}{T} \sum_{t=1}^{T} \ell(y_t, f(x_t))$ is small.

All "data" (e.g., x_t, y_t) could be controlled by an adversary!

Goal: perform (almost) as well as the best-in-hindsight $f \in \mathcal{F}$.

Online learning Framework

Online learning algorithm

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

1. Observe data point $x_t \in \mathcal{X}$.
2. Choose prediction $\hat{y}_t \in \mathcal{Y}$.
3. Observe label $y_t \in \mathcal{Y}$.
4. Suffer loss $\ell(y_t, \hat{y}_t)$ (e.g., $\ell(y_t, \hat{y}_t) = \mathbb{1}\{\hat{y}_t \neq y_t\}$).

Underlying (non-probabilistic) assumption

The function class $\mathcal{F}$ models the data well:
i.e. best-in-hindsight (average) loss $\min_{f \in \mathcal{F}} \frac{1}{T} \sum_{t=1}^{T} \ell(y_t, f(x_t))$ is small.

All "data" (e.g., x_t, y_t) could be controlled by an adversary!

Goal: perform (almost) as well as the best-in-hindsight $f \in \mathcal{F}$.

(Average) **Regret**:

$$\underbrace{\frac{1}{T} \sum_{t=1}^{T} \ell(y_t, \hat{y}_t)}_{\text{your avg. loss}} - \underbrace{\min_{f \in \mathcal{F}} \frac{1}{T} \sum_{t=1}^{T} \ell(y_t, f(x_t))}_{\text{avg. loss of best } f \in \mathcal{F}}$$

Example

$\mathcal{X} = \mathbb{N}$, $\mathcal{Y} = \{0, 1\}$, $\ell(y, \hat{y}) := \mathbb{1}\{\hat{y} \neq y\}$, $\mathcal{F} =$ threshold functions.

t	1	2	3	4	5	6	7	8	9
x_t									
$\hat{y}_t$									
y_t									

(Average) **Regret**:

$$\frac{1}{9}\sum_{t=1}^{9} \ell(y_t, \hat{y}_t) \quad - \quad \min_{f \in \mathcal{F}} \frac{1}{9}\sum_{t=1}^{9} \ell(y_t, f(x_t))$$

Example

$\mathcal{X} = \mathbb{N}$, $\mathcal{Y} = \{0, 1\}$, $\ell(y, \hat{y}) := \mathbb{1}\{\hat{y} \neq y\}$, $\mathcal{F} =$ threshold functions.

t	1	2	3	4	5	6	7	8	9
x_t	7								
$\hat{y}_t$									
y_t									

(Average) **Regret**:

$$\frac{1}{9}\sum_{t=1}^{9} \ell(y_t, \hat{y}_t) \quad - \quad \min_{f \in \mathcal{F}} \frac{1}{9}\sum_{t=1}^{9} \ell(y_t, f(x_t))$$

Example

$\mathcal{X} = \mathbb{N}$, $\mathcal{Y} = \{0, 1\}$, $\ell(y, \hat{y}) := \mathbb{1}\{\hat{y} \neq y\}$, $\mathcal{F} =$ threshold functions.

t	1	2	3	4	5	6	7	8	9
x_t	7								
$\hat{y}_t$	0								
y_t									

(Average) **Regret**:

$$\frac{1}{9}\sum_{t=1}^{9} \ell(y_t, \hat{y}_t) \quad - \quad \min_{f \in \mathcal{F}} \frac{1}{9}\sum_{t=1}^{9} \ell(y_t, f(x_t))$$

Example

$\mathcal{X} = \mathbb{N}$, $\mathcal{Y} = \{0, 1\}$, $\ell(y, \hat{y}) := \mathbb{1}\{\hat{y} \neq y\}$, $\mathcal{F} =$ threshold functions.

t	1	2	3	4	5	6	7	8	9
x_t	7								
$\hat{y}_t$	0								
y_t	1								

(Average) **Regret**:

$$\frac{1}{9}\sum_{t=1}^{9} \ell(y_t, \hat{y}_t) \quad - \quad \min_{f \in \mathcal{F}} \frac{1}{9}\sum_{t=1}^{9} \ell(y_t, f(x_t))$$

Example

$\mathcal{X} = \mathbb{N}$, $\mathcal{Y} = \{0, 1\}$, $\ell(y, \hat{y}) := \mathbb{1}\{\hat{y} \neq y\}$, $\mathcal{F} =$ threshold functions.

t	1	2	3	4	5	6	7	8	9
x_t	7	8							
$\hat{y}_t$	0								
y_t	1								

(Average) **Regret**:

$$\frac{1}{9}\sum_{t=1}^{9} \ell(y_t, \hat{y}_t) \quad - \quad \min_{f \in \mathcal{F}} \frac{1}{9}\sum_{t=1}^{9} \ell(y_t, f(x_t))$$

Example

$\mathcal{X} = \mathbb{N}$, $\mathcal{Y} = \{0, 1\}$, $\ell(y, \hat{y}) := \mathbb{1}\{\hat{y} \neq y\}$, $\mathcal{F} =$ threshold functions.

t	1	2	3	4	5	6	7	8	9
x_t	7	8							
$\hat{y}_t$	0	1							
y_t	1								

(Average) **Regret**:

$$\frac{1}{9}\sum_{t=1}^{9} \ell(y_t, \hat{y}_t) \quad - \quad \min_{f \in \mathcal{F}} \frac{1}{9}\sum_{t=1}^{9} \ell(y_t, f(x_t))$$

Example

$\mathcal{X} = \mathbb{N}$, $\mathcal{Y} = \{0, 1\}$, $\ell(y, \hat{y}) := \mathbb{1}\{\hat{y} \neq y\}$, $\mathcal{F} =$ threshold functions.

t	1	2	3	4	5	6	7	8	9
x_t	7	8							
$\hat{y}_t$	0	1							
y_t	1	1							

(Average) **Regret**:

$$\frac{1}{9}\sum_{t=1}^{9} \ell(y_t, \hat{y}_t) \quad - \quad \min_{f \in \mathcal{F}} \frac{1}{9}\sum_{t=1}^{9} \ell(y_t, f(x_t))$$

Example

$\mathcal{X} = \mathbb{N}$, $\mathcal{Y} = \{0, 1\}$, $\ell(y, \hat{y}) := \mathbb{1}\{\hat{y} \neq y\}$, $\mathcal{F} =$ threshold functions.

t	1	2	3	4	5	6	7	8	9
x_t	7	8	1						
$\hat{y}_t$	0	1							
y_t	1	1							

(Average) **Regret**:

$$\frac{1}{9}\sum_{t=1}^{9} \ell(y_t, \hat{y}_t) \quad - \quad \min_{f \in \mathcal{F}} \frac{1}{9}\sum_{t=1}^{9} \ell(y_t, f(x_t))$$

Example

$\mathcal{X} = \mathbb{N}$, $\mathcal{Y} = \{0, 1\}$, $\ell(y, \hat{y}) := \mathbb{1}\{\hat{y} \neq y\}$, $\mathcal{F} =$ threshold functions.

t	1	2	3	4	5	6	7	8	9
x_t	7	8	1						
$\hat{y}_t$	0	1	1						
y_t	1	1							

(Average) **Regret**:

$$\frac{1}{9}\sum_{t=1}^{9} \ell(y_t, \hat{y}_t) \quad - \quad \min_{f \in \mathcal{F}} \frac{1}{9}\sum_{t=1}^{9} \ell(y_t, f(x_t))$$

Example

$\mathcal{X} = \mathbb{N}$, $\mathcal{Y} = \{0, 1\}$, $\ell(y, \hat{y}) := \mathbb{1}\{\hat{y} \neq y\}$, $\mathcal{F} =$ threshold functions.

t	1	2	3	4	5	6	7	8	9
x_t	7	8	1						
$\hat{y}_t$	0	1	1						
y_t	1	1	0						

(Average) **Regret**:

$$\frac{1}{9}\sum_{t=1}^{9} \ell(y_t, \hat{y}_t) \quad - \quad \min_{f \in \mathcal{F}} \frac{1}{9}\sum_{t=1}^{9} \ell(y_t, f(x_t))$$

Example

$\mathcal{X} = \mathbb{N}$, $\mathcal{Y} = \{0, 1\}$, $\ell(y, \hat{y}) := \mathbb{1}\{\hat{y} \neq y\}$, $\mathcal{F} =$ threshold functions.

t	1	2	3	4	5	6	7	8	9
x_t	7	8	1	9	2	6	3	1	4
$\hat{y}_t$	0	1	1	1	1	1	1	0	1
y_t	1	1	0	1	0	1	1	0	0

(Average) **Regret**:

$$\frac{1}{9}\sum_{t=1}^{9} \ell(y_t, \hat{y}_t) \quad - \quad \min_{f \in \mathcal{F}} \frac{1}{9}\sum_{t=1}^{9} \ell(y_t, f(x_t))$$

Example

$\mathcal{X} = \mathbb{N}$, $\mathcal{Y} = \{0, 1\}$, $\ell(y, \hat{y}) := \mathbb{1}\{\hat{y} \neq y\}$, $\mathcal{F} =$ threshold functions.

t	1	2	3	4	5	6	7	8	9
x_t	7	8	1	9	2	6	3	1	4
$\hat{y}_t$	0	1	1	1	1	1	1	0	1
y_t	1	1	0	1	0	1	1	0	0

(Average) **Regret**:

$$\frac{1}{9}\sum_{t=1}^{9} \ell(y_t, \hat{y}_t) \quad - \quad \min_{f \in \mathcal{F}} \frac{1}{9}\sum_{t=1}^{9} \ell(y_t, f(x_t))$$

$$= \quad \frac{4}{9} \quad - \quad \frac{1}{9}$$

($f(x) = \mathbb{1}\{x > 2\}$ makes just one mistake.)

Example

$\mathcal{X} = \mathbb{N}$, $\mathcal{Y} = \{0, 1\}$, $\ell(y, \hat{y}) := \mathbb{1}\{\hat{y} \neq y\}$, $\mathcal{F} =$ threshold functions.

t	1	2	3	4	5	6	7	8	9
x_t	7	8	1	9	2	6	3	1	4
$\hat{y}_t$	0	1	1	1	1	1	1	0	1
y_t	1	1	0	1	0	1	1	0	0

(Average) **Regret**:

$$\frac{1}{9}\sum_{t=1}^{9} \ell(y_t, \hat{y}_t) \quad - \quad \min_{f \in \mathcal{F}} \frac{1}{9}\sum_{t=1}^{9} \ell(y_t, f(x_t))$$

$$= \quad \frac{4}{9} \quad - \quad \frac{1}{9}$$

($f(x) = \mathbb{1}\{x > 2\}$ makes just one mistake.)

Goal: Regret $\to 0$ as $T \to \infty$ (i.e., "vanishing regret" or "no regret").

Some easy cases

Simplifying assumptions

- $N := |\mathcal{F}|$ is finite, $\mathcal{Y} = \{0, 1\}$, $\ell(y, \hat{y}) = \mathbb{1}\{\hat{y} \neq y\}$ (zero-one loss).
- *Realizability*: assume there is a $f \in \mathcal{F}$ with $y_t = f(x_t)$ for all t.

Simplifying assumptions

- $N := |\mathcal{F}|$ is finite, $\mathcal{Y} = \{0, 1\}$, $\ell(y, \hat{y}) = \mathbb{1}\{\hat{y} \neq y\}$ (zero-one loss).
- *Realizability*: assume there is a $f \in \mathcal{F}$ with $y_t = f(x_t)$ for all t.

  So **Regret** is simply Average Loss (i.e., average number of mistakes):

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{1}\{\hat{y}_t \neq y_t\} .$$

Some easy cases

Simplifying assumptions

- $N := |\mathcal{F}|$ is finite, $\mathcal{Y} = \{0, 1\}$, $\ell(y, \hat{y}) = \mathbb{1}\{\hat{y} \neq y\}$ (zero-one loss).
- *Realizability*: assume there is a $f \in \mathcal{F}$ with $y_t = f(x_t)$ for all t.

  So **Regret** is simply Average Loss (i.e., average number of mistakes):

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{1}\{\hat{y}_t \neq y_t\} .$$

Trivial algorithm: follow-the-leader

In round t, pick any $f_t \in \mathcal{F}$ that is *perfect-so-far*, and choose $\hat{y}_t := f_t(x_t)$.

Some easy cases

Simplifying assumptions

- $N := |\mathcal{F}|$ is finite, $\mathcal{Y} = \{0, 1\}$, $\ell(y, \hat{y}) = \mathbb{1}\{\hat{y} \neq y\}$ (zero-one loss).
- *Realizability*: assume there is a $f \in \mathcal{F}$ with $y_t = f(x_t)$ for all t.

  So **Regret** is simply Average Loss (i.e., average number of mistakes):

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{1}\{\hat{y}_t \neq y_t\} .$$

Trivial algorithm: follow-the-leader

In round t, pick any $f_t \in \mathcal{F}$ that is *perfect-so-far*, and choose $\hat{y}_t := f_t(x_t)$.

What is the average loss after T rounds?

Simplifying assumptions

- $N := |\mathcal{F}|$ is finite, $\mathcal{Y} = \{0, 1\}$, $\ell(y, \hat{y}) = \mathbb{1}\{\hat{y} \neq y\}$ (zero-one loss).
- *Realizability*: assume there is a $f \in \mathcal{F}$ with $y_t = f(x_t)$ for all t.

  So **Regret** is simply Average Loss (i.e., average number of mistakes):

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{1}\{\hat{y}_t \neq y_t\} .$$

Trivial algorithm: follow-the-leader

In round t, pick any $f_t \in \mathcal{F}$ that is *perfect-so-far*, and choose $\hat{y}_t := f_t(x_t)$.

What is the average loss after T rounds?

Answer: every mistake eliminates one function. So at most $N - 1$ mistakes, and average loss is at most $\frac{N-1}{T}$.

"Halving" algorithm

In round t,

$$\hat{y}_t := \arg\max_{y \in \{0,1\}} |\{\text{perfect-so-far } f : f(x_t) = y\}|$$

(i.e., go with majority's prediction).

A better algorithm

"Halving" algorithm

In round t,

$$\hat{y}_t := \arg\max_{y \in \{0,1\}} |\{\text{perfect-so-far } f : f(x_t) = y\}|$$

(i.e., go with majority's prediction).

What is the average loss after T rounds?

"Halving" algorithm

In round t,

$$\hat{y}_t := \arg\max_{y \in \{0,1\}} |\{\text{perfect-so-far } f : f(x_t) = y\}|$$

(i.e., go with majority's prediction).

What is the average loss after T rounds?

Answer: every mistake eliminates half the remaining functions. So, at most $\log_2(N)$ mistakes, and average loss is at most $\frac{\log_2(N)}{T}$.

"Halving" algorithm

In round t,

$$\hat{y}_t := \arg\max_{y \in \{0,1\}} |\{\text{perfect-so-far } f : f(x_t) = y\}|$$

(i.e., go with majority's prediction).

What is the average loss after T rounds?

Answer: every mistake eliminates half the remaining functions. So, at most $\log_2(N)$ mistakes, and average loss is at most $\frac{\log_2(N)}{T}$.

What if we drop realizability assumption?

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$									
y_t									
$L_t(f_0)$									
$L_t(f_1)$									
leader									

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1								
y_t									
$L_t(f_0)$									
$L_t(f_1)$									
leader									

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1								
y_t	$2/3$								
$L_t(f_0)$									
$L_t(f_1)$									
leader									

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1								
y_t	$2/3$								
$L_t(f_0)$	$2/3$								
$L_t(f_1)$	$1/3$								
leader	f_1								

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1	1							
y_t	$2/3$								
$L_t(f_0)$	$2/3$								
$L_t(f_1)$	$1/3$								
leader	f_1								

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1	1							
y_t	$2/3$	0							
$L_t(f_0)$	$2/3$								
$L_t(f_1)$	$1/3$								
leader	f_1								

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1	1							
y_t	$2/3$	0							
$L_t(f_0)$	$2/3$	$2/3$							
$L_t(f_1)$	$1/3$	$4/3$							
leader	f_1	f_0							

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1	1	0						
y_t	$2/3$	0							
$L_t(f_0)$	$2/3$	$2/3$							
$L_t(f_1)$	$1/3$	$4/3$							
leader	f_1	f_0							

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1	1	0						
y_t	$2/3$	0	1						
$L_t(f_0)$	$2/3$	$2/3$							
$L_t(f_1)$	$1/3$	$4/3$							
leader	f_1	f_0							

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1	1	0						
y_t	$2/3$	0	1						
$L_t(f_0)$	$2/3$	$2/3$	$5/3$						
$L_t(f_1)$	$1/3$	$4/3$	$4/3$						
leader	f_1	f_0	f_1						

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1	1	0	1					
y_t	$2/3$	0	1						
$L_t(f_0)$	$2/3$	$2/3$	$5/3$						
$L_t(f_1)$	$1/3$	$4/3$	$4/3$						
leader	f_1	f_0	f_1						

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1	1	0	1					
y_t	$2/3$	0	1	0					
$L_t(f_0)$	$2/3$	$2/3$	$5/3$						
$L_t(f_1)$	$1/3$	$4/3$	$4/3$						
leader	f_1	f_0	f_1						

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1	1	0	1					
y_t	$2/3$	0	1	0					
$L_t(f_0)$	$2/3$	$2/3$	$5/3$	$5/3$					
$L_t(f_1)$	$1/3$	$4/3$	$4/3$	$7/3$					
leader	f_1	f_0	f_1	f_0					

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1	1	0	1	0				
y_t	$2/3$	0	1	0					
$L_t(f_0)$	$2/3$	$2/3$	$5/3$	$5/3$					
$L_t(f_1)$	$1/3$	$4/3$	$4/3$	$7/3$					
leader	f_1	f_0	f_1	f_0					

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1	1	0	1	0				
y_t	$2/3$	0	1	0	1				
$L_t(f_0)$	$2/3$	$2/3$	$5/3$	$5/3$					
$L_t(f_1)$	$1/3$	$4/3$	$4/3$	$7/3$					
leader	f_1	f_0	f_1	f_0					

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1	1	0	1	0				
y_t	$2/3$	0	1	0	1				
$L_t(f_0)$	$2/3$	$2/3$	$5/3$	$5/3$	$8/3$				
$L_t(f_1)$	$1/3$	$4/3$	$4/3$	$7/3$	$7/3$				
leader	f_1	f_0	f_1	f_0	f_1				

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1	1	0	1	0	1	0	1	0
y_t	$2/3$	0	1	0	1	0	1	0	1
$L_t(f_0)$	$2/3$	$2/3$	$5/3$	$5/3$	$8/3$	$8/3$	$11/3$	$11/3$	$14/3$
$L_t(f_1)$	$1/3$	$4/3$	$4/3$	$7/3$	$7/3$	$10/3$	$10/3$	$13/3$	$13/3$
leader	f_1	f_0	f_1	f_0	f_1	f_0	f_1	f_0	f_1

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1	1	0	1	0	1	0	1	0
y_t	$2/3$	0	1	0	1	0	1	0	1
$L_t(f_0)$	$2/3$	$2/3$	$5/3$	$5/3$	$8/3$	$8/3$	$11/3$	$11/3$	$14/3$
$L_t(f_1)$	$1/3$	$4/3$	$4/3$	$7/3$	$7/3$	$10/3$	$10/3$	$13/3$	$13/3$
leader	f_1	f_0	f_1	f_0	f_1	f_0	f_1	f_0	f_1

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Regret of follow-the-leader is constant (i.e. not going to 0).

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1	1	0	1	0	1	0	1	0
y_t	$2/3$	0	1	0	1	0	1	0	1
$L_t(f_0)$	$2/3$	$2/3$	$5/3$	$5/3$	$8/3$	$8/3$	$11/3$	$11/3$	$14/3$
$L_t(f_1)$	$1/3$	$4/3$	$4/3$	$7/3$	$7/3$	$10/3$	$10/3$	$13/3$	$13/3$
leader	f_1	f_0	f_1	f_0	f_1	f_0	f_1	f_0	f_1

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Regret of follow-the-leader is constant (i.e. not going to 0).

Can any learner do much better?

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1	1	0	1	0	1	0	1	0
y_t	$2/3$	0	1	0	1	0	1	0	1
$L_t(f_0)$	$2/3$	$2/3$	$5/3$	$5/3$	$8/3$	$8/3$	$11/3$	$11/3$	$14/3$
$L_t(f_1)$	$1/3$	$4/3$	$4/3$	$7/3$	$7/3$	$10/3$	$10/3$	$13/3$	$13/3$
leader	f_1	f_0	f_1	f_0	f_1	f_0	f_1	f_0	f_1

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Regret of follow-the-leader is constant (i.e. not going to 0).

Can any learner do much better?

Cover's impossibility theorem: No *deterministic* learner can ensure regret $< \frac{1}{2}$

Example: failure of follow-the-leader

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$\hat{y}_t$	1	1	0	1	0	1	0	1	0
y_t	$2/3$	0	1	0	1	0	1	0	1
$L_t(f_0)$	$2/3$	$2/3$	$5/3$	$5/3$	$8/3$	$8/3$	$11/3$	$11/3$	$14/3$
$L_t(f_1)$	$1/3$	$4/3$	$4/3$	$7/3$	$7/3$	$10/3$	$10/3$	$13/3$	$13/3$
leader	f_1	f_0	f_1	f_0	f_1	f_0	f_1	f_0	f_1

Follow-the-leader: go with the function $f \in \mathcal{F}$ that is best-so-far.

Regret of follow-the-leader is constant (i.e. not going to 0).

Can any learner do much better?

Cover's impossibility theorem: No *deterministic* learner can ensure regret $< \frac{1}{2}$

However:

If the learner can be randomized, then can make regret $\to 0$ as $T \to \infty$.

Using randomness

Assume $N := |\mathcal{F}|$ is finite, and $\ell(y, \hat{y}) \in [0, 1]$.

Let $L_t(f) := \sum_{\tau=1}^{t} \ell(y_\tau, f(x_\tau))$ be cumulative loss of f after t rounds.

Using randomness

Assume $N := |\mathcal{F}|$ is finite, and $\ell(y, \hat{y}) \in [0, 1]$.

Let $L_t(f) := \sum_{\tau=1}^{t} \ell(y_\tau, f(x_\tau))$ be cumulative loss of f after t rounds.

Randomized Weighted Majority (Littlestone and Warmuth, 1994)

In round t, randomly pick f_t from distribution $\boldsymbol{w}_t$ on $\mathcal{F}$ given by

$$w_t(f) \;\propto\; \exp(-\eta \cdot L_{t-1}(f))\,;$$

predict $\hat{y}_t := f_t(x_t)$.

Using randomness

Assume $N := |\mathcal{F}|$ is finite, and $\ell(y, \hat{y}) \in [0, 1]$.

Let $L_t(f) := \sum_{\tau=1}^{t} \ell(y_\tau, f(x_\tau))$ be cumulative loss of f after t rounds.

Randomized Weighted Majority (Littlestone and Warmuth, 1994)

In round t, randomly pick f_t from distribution $\boldsymbol{w}_t$ on $\mathcal{F}$ given by

$$w_t(f) \;\propto\; \exp(-\eta \cdot L_{t-1}(f))\,;$$

predict $\hat{y}_t := f_t(x_t)$.

Expected regret after T rounds:

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} \ell(y_t, \hat{y}_t) - \min_{f \in \mathcal{F}} \frac{1}{T}\sum_{t=1}^{T} \ell(y_t, f(x_t))\right] \;\leq\; \frac{1}{\eta} \cdot \frac{\ln(N)}{T} + \eta \cdot \frac{1}{2}\,.$$

Using randomness

Assume $N := |\mathcal{F}|$ is finite, and $\ell(y, \hat{y}) \in [0, 1]$.

Let $L_t(f) := \sum_{\tau=1}^{t} \ell(y_\tau, f(x_\tau))$ be cumulative loss of f after t rounds.

Randomized Weighted Majority (Littlestone and Warmuth, 1994)

In round t, randomly pick f_t from distribution $\boldsymbol{w}_t$ on $\mathcal{F}$ given by

$$w_t(f) \;\propto\; \exp(-\eta \cdot L_{t-1}(f))\,;$$

predict $\hat{y}_t := f_t(x_t)$.

Expected regret after T rounds:

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\ell(y_t, \hat{y}_t) - \min_{f\in\mathcal{F}}\frac{1}{T}\sum_{t=1}^{T}\ell(y_t, f(x_t))\right] \;\leq\; \frac{1}{\eta}\cdot\frac{\ln(N)}{T} + \eta\cdot\frac{1}{2}\,.$$

Optimize bound with $\eta := \sqrt{\frac{2\ln(N)}{T}}$ to get regret bound $\sqrt{\frac{2\ln(N)}{T}}$.

Using randomness

Assume $N := |\mathcal{F}|$ is finite, and $\ell(y, \hat{y}) \in [0, 1]$.

Let $L_t(f) := \sum_{\tau=1}^{t} \ell(y_\tau, f(x_\tau))$ be cumulative loss of f after t rounds.

Randomized Weighted Majority (Littlestone and Warmuth, 1994)

In round t, randomly pick f_t from distribution $\boldsymbol{w}_t$ on $\mathcal{F}$ given by

$$w_t(f) \;\propto\; \exp(-\eta \cdot L_{t-1}(f))\,;$$

predict $\hat{y}_t := f_t(x_t)$.

Expected regret after T rounds:

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\ell(y_t, \hat{y}_t) - \min_{f\in\mathcal{F}}\frac{1}{T}\sum_{t=1}^{T}\ell(y_t, f(x_t))\right] \;\leq\; \frac{1}{\eta}\cdot\frac{\ln(N)}{T} + \eta\cdot\frac{1}{2}\,.$$

Optimize bound with $\eta := \sqrt{\frac{2\ln(N)}{T}}$ to get regret bound $\sqrt{\frac{2\ln(N)}{T}}$.

Could also use $\eta_t := \sqrt{\frac{2\ln(N)}{t}}$ in round t.

Example: RWM

$\mathcal{Y} = [0, 1]$, $\ell(y, \hat{y}) := |\hat{y} - y|$ (absolute loss),

t	1	2	3	4	5	6	7	8	9
$f_0(x_t)$	0	0	0	0	0	0	0	0	0
$f_1(x_t)$	1	1	1	1	1	1	1	1	1
$w_t(f_1)$	.500	.569	.389	.549	.413	.540	.426	.535	.435
y_t	$2/3$	0	1	0	1	0	1	0	1
$L_t(f_0)$	$2/3$	$2/3$	$5/3$	$5/3$	$8/3$	$8/3$	$11/3$	$11/3$	$14/3$
$L_t(f_1)$	$1/3$	$4/3$	$4/3$	$7/3$	$7/3$	$10/3$	$10/3$	$13/3$	$13/3$
leader	f_1	f_0	f_1	f_0	f_1	f_0	f_1	f_0	f_1

Expected regret of RWM is $O(1/\sqrt{T})$.

Why RWM works

Why RWM works

- Randomization lets us precisely "hedge" over functions in $\mathcal{F}$.

  (A generalization of RWM called "Hedge" is due to Freund and Schapire; proposed in same paper as AdaBoost.)

Why RWM works

- Randomization lets us precisely "hedge" over functions in $\mathcal{F}$.

  (A generalization of RWM called "Hedge" is due to Freund and Schapire; proposed in same paper as AdaBoost.)

- Minimizer of cumulative loss L_{t-1} can change in every round.
  But the distribution given by

$$w_t(f) \;\propto\; \exp(-\eta \cdot L_{t-1}(f))$$

  (as in RWM) changes relatively slowly when η is small.

Usual approach

Split training data S into two parts, S' and S'':

- Use first part S' to **grow tree $\mathcal{T}$ until all leaves are pure**.
- Use second part S'' to **choose a good pruning of tree $\mathcal{T}$**.

Application to learning decision trees

Usual approach

Split training data S into two parts, S' and S'':

- Use first part S' to **grow tree $\mathcal{T}$ until all leaves are pure**.
- Use second part S'' to **choose a good pruning of tree $\mathcal{T}$**.

Alternative approach (Helmbold and Schapire, 1995)

Split training data S into two parts, S' and S'':

- Use first part S' to **grow tree $\mathcal{T}$ until all leaves are pure**.
- Use second part S'' to **learn weighting over all prunings of tree $\mathcal{T}$**:
  Let $\mathcal{F}$ = all prunings of $\mathcal{T}$, and run RWM using S''.

Application to learning decision trees

Usual approach

Split training data S into two parts, S' and S'':

- Use first part S' to **grow tree $\mathcal{T}$ until all leaves are pure**.
- Use second part S'' to **choose a good pruning of tree $\mathcal{T}$**.

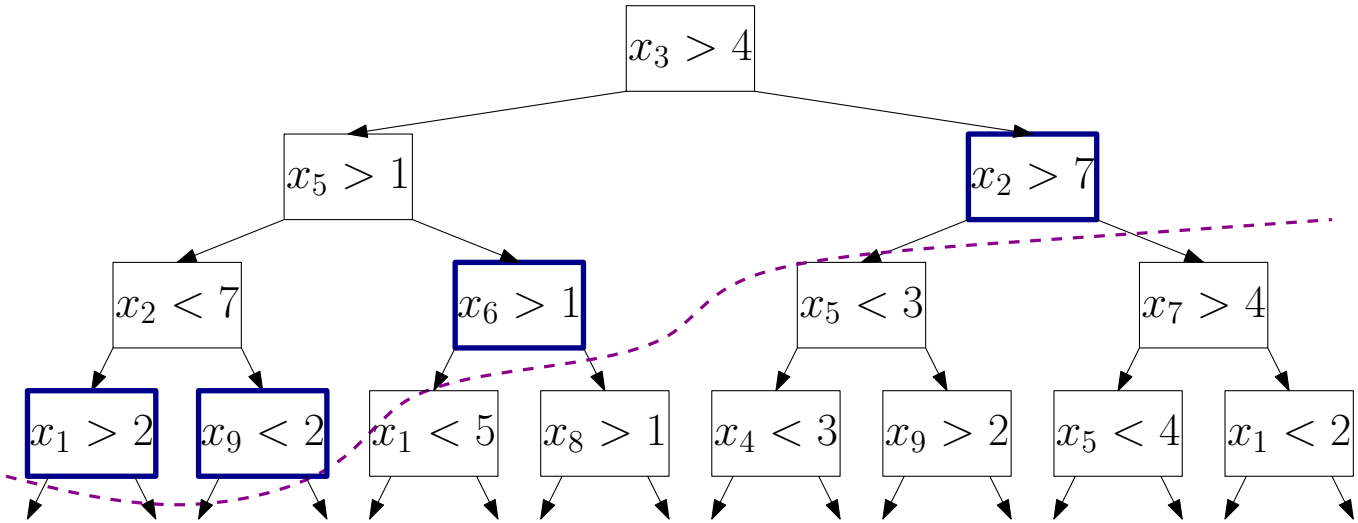
Alternative approach (Helmbold and Schapire, 1995)

Split training data S into two parts, S' and S'':

- Use first part S' to **grow tree $\mathcal{T}$ until all leaves are pure**.
- Use second part S'' to **learn weighting over all prunings of tree $\mathcal{T}$**:
  Let $\mathcal{F}$ = all prunings of $\mathcal{T}$, and run RWM using S''.

  Problem: $|\mathcal{F}|$ is exponentially large!

Dynamic programming



- Each $f \in \mathcal{F}$ corresponds to a pruning of the tree $\mathcal{T}$.
- $L_t(f)$ depends on prediction error statistics at leaves of f; weight of f is just $\exp(-\eta L_t(f))$.
- Maintain error statistics for all nodes in $\mathcal{T}$.
- Can use dynamic programming to combine weights of all $f \in \mathcal{F}$ to make weighted prediction.
- See (Helmbold & Schapire, 1995) for details.

Infinite function classes

- RWM only works for finite function classes

Infinite function classes

- RWM only works for finite function classes
- For infinite function classes, bounded VC dimension is *not enough*, unlike the batch case

Infinite function classes

- RWM only works for finite function classes
- For infinite function classes, bounded VC dimension is *not enough*, unlike the batch case
- E.g. the class of thresholds in $\mathbb{R}$ has VC-dimension 2, but no online learning algorithm can achieve vanishing regret

Infinite function classes

- RWM only works for finite function classes
- For infinite function classes, bounded VC dimension is *not enough*, unlike the batch case
- E.g. the class of thresholds in $\mathbb{R}$ has VC-dimension 2, but no online learning algorithm can achieve vanishing regret
- Different combinatorial parameter called **Littlestone dimension** characterizes online learnability

Infinite function classes

- RWM only works for finite function classes
- For infinite function classes, bounded VC dimension is *not enough*, unlike the batch case
- E.g. the class of thresholds in $\mathbb{R}$ has VC-dimension 2, but no online learning algorithm can achieve vanishing regret
- Different combinatorial parameter called **Littlestone dimension** characterizes online learnability
- For $\mathcal{F}$ with bounded Littlestone dimension, generic online learning algorithm: RWM on carefully constructed "experts"

Infinite function classes

- RWM only works for finite function classes
- For infinite function classes, bounded VC dimension is *not enough*, unlike the batch case
- E.g. the class of thresholds in $\mathbb{R}$ has VC-dimension 2, but no online learning algorithm can achieve vanishing regret
- Different combinatorial parameter called **Littlestone dimension** characterizes online learnability
- For $\mathcal{F}$ with bounded Littlestone dimension, generic online learning algorithm: RWM on carefully constructed "experts"
- Caveat: generic online learning algorithm is *inefficient*

Online Convex Optimization

- Suppose now that $\mathcal{F}$ is a parametric function class:
  - $\boldsymbol{W} \subseteq \mathbb{R}^d$ is a **convex set** of allowable parameters.
  - For every $\boldsymbol{w} \in \boldsymbol{W}$, $f_{\boldsymbol{w}} : \mathcal{X} \to \mathcal{Y}$ is predictor parameterized by $\boldsymbol{w}$
  - $\mathcal{F} = \{f_{\boldsymbol{w}} \mid \boldsymbol{w} \in \boldsymbol{W}\}$

Online Convex Optimization

- Suppose now that $\mathcal{F}$ is a parametric function class:
  - $\boldsymbol{W} \subseteq \mathbb{R}^d$ is a **convex set** of allowable parameters.
  - For every $\boldsymbol{w} \in \boldsymbol{W}$, $f_{\boldsymbol{w}} : \mathcal{X} \to \mathcal{Y}$ is predictor parameterized by $\boldsymbol{w}$
  - $\mathcal{F} = \{f_{\boldsymbol{w}} \mid \boldsymbol{w} \in \boldsymbol{W}\}$
- Assume, further, that for any example $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$, the mapping $\boldsymbol{w} \mapsto \ell(y, f_{\boldsymbol{w}}(\boldsymbol{x}))$ is **convex** in $\boldsymbol{w}$.

Online Convex Optimization

- Suppose now that $\mathcal{F}$ is a parametric function class:
  - $\boldsymbol{W} \subseteq \mathbb{R}^d$ is a **convex set** of allowable parameters.
  - For every $\boldsymbol{w} \in \boldsymbol{W}$, $f_{\boldsymbol{w}} : \mathcal{X} \to \mathcal{Y}$ is predictor parameterized by $\boldsymbol{w}$
  - $\mathcal{F} = \{f_{\boldsymbol{w}} \mid \boldsymbol{w} \in \boldsymbol{W}\}$
- Assume, further, that for any example $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$, the mapping $\boldsymbol{w} \mapsto \ell(y, f_{\boldsymbol{w}}(\boldsymbol{x}))$ is **convex** in $\boldsymbol{w}$.
- Example: linear regression setting
  - $\mathcal{X} = \mathbb{R}^d$
  - $\mathcal{Y} = \mathbb{R}$
  - $\boldsymbol{W} = \{\boldsymbol{w} \in \mathbb{R}^d \mid \|\boldsymbol{w}\|_2 \leq R\}$, a convex set
  - $f_{\boldsymbol{w}}(\boldsymbol{x}) := \langle \boldsymbol{w}, \boldsymbol{x} \rangle$
  - Square loss: $\ell(y, \hat{y}) = (\hat{y} - y)^2$
  - For any $(\boldsymbol{x}, y) \in \mathbb{R}^d \times \mathbb{R}$, $\boldsymbol{w} \mapsto (\langle \boldsymbol{w}, \boldsymbol{x} \rangle - y)^2$ is *convex*

Online Convex Optimization

- Suppose now that $\mathcal{F}$ is a parametric function class:
  - $\boldsymbol{W} \subseteq \mathbb{R}^d$ is a **convex set** of allowable parameters.
  - For every $\boldsymbol{w} \in \boldsymbol{W}$, $f_{\boldsymbol{w}} : \mathcal{X} \to \mathcal{Y}$ is predictor parameterized by $\boldsymbol{w}$
  - $\mathcal{F} = \{f_{\boldsymbol{w}} \mid \boldsymbol{w} \in \boldsymbol{W}\}$
- Assume, further, that for any example $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$, the mapping $\boldsymbol{w} \mapsto \ell(y, f_{\boldsymbol{w}}(\boldsymbol{x}))$ is **convex** in $\boldsymbol{w}$.
- Example: linear regression setting
  - $\mathcal{X} = \mathbb{R}^d$
  - $\mathcal{Y} = \mathbb{R}$
  - $\boldsymbol{W} = \{\boldsymbol{w} \in \mathbb{R}^d \mid \|\boldsymbol{w}\|_2 \leq R\}$, a convex set
  - $f_{\boldsymbol{w}}(\boldsymbol{x}) := \langle \boldsymbol{w}, \boldsymbol{x} \rangle$
  - Square loss: $\ell(y, \hat{y}) = (\hat{y} - y)^2$
  - For any $(\boldsymbol{x}, y) \in \mathbb{R}^d \times \mathbb{R}$, $\boldsymbol{w} \mapsto (\langle \boldsymbol{w}, \boldsymbol{x} \rangle - y)^2$ is *convex*
- This setting is called *online convex optimization*, and admits *efficient* algorithms

Online Gradient Descent

Algorithm for online convex optimization: an online form of gradient descent!

Online Gradient Descent

Algorithm for online convex optimization: an online form of gradient descent!

Seen before: online perceptron and stochastic gradient descent

Online Gradient Descent

Algorithm for online convex optimization: an online form of gradient descent!

Seen before: online perceptron and stochastic gradient descent

Online Gradient Descent (OGD) (Zinkevich, 2003)

1: Choose learning rate η.
2: Initialize weights $\boldsymbol{w}_1 \in \boldsymbol{W}$ arbitrarily.
3: **for** $t = 1, 2, \ldots, T$ **do**
4: Obtain $\boldsymbol{x}_t \in \mathbb{R}^d$
5: Predict $\hat{y}_t = f_{\boldsymbol{w}_t}(\boldsymbol{x}_t)$
6: Observe y_t, and incur loss $\ell(y_t, \hat{y}_t)$.
7: Compute the gradient $\nabla_{\boldsymbol{w}} \ell(y_t, f_{\boldsymbol{w}}(\boldsymbol{x}_t))$ at $\boldsymbol{w}_t$, call it $\boldsymbol{g}_t$.
8: Update:

$$\boldsymbol{w}_{t+1} \;=\; \Pi_{\boldsymbol{W}}(\boldsymbol{w}_t - \eta \boldsymbol{g}_t)$$

9: **end for**

Here, $\Pi_{\boldsymbol{W}}(\boldsymbol{w}') := \arg\min_{\boldsymbol{w} \in \boldsymbol{W}} \|\boldsymbol{w} - \boldsymbol{w}'\|_2^2$. Also known as *projection* on $\boldsymbol{W}$.

Online Gradient Descent (OGD)

1: Choose learning rate η.
2: Initialize weights $\boldsymbol{w}_1 \in \boldsymbol{W}$ arbitrarily.
3: **for** $t = 1, 2, \ldots, T$ **do**
4: Obtain $\boldsymbol{x}_t \in \mathbb{R}^d$
5: Predict $\hat{y}_t = f_{\boldsymbol{w}_t}(\boldsymbol{x}_t)$
6: Observe y_t, and incur loss $\ell(y_t, \hat{y}_t)$.
7: Compute the gradient $\nabla_{\boldsymbol{w}} \ell(y_t, f_{\boldsymbol{w}}(\boldsymbol{x}_t))$ at $\boldsymbol{w}_t$, call it $\boldsymbol{g}_t$.
8: Update:

$$\boldsymbol{w}_{t+1} \;=\; \Pi_{\boldsymbol{W}}(\boldsymbol{w}_t - \eta \boldsymbol{g}_t)$$

9: **end for**

Regret of OGD after T rounds: Suppose for any $\boldsymbol{w}, \boldsymbol{w}' \in \boldsymbol{W}$, we have $\|\boldsymbol{w} - \boldsymbol{w}'\|_2 \le D$ and $\|\boldsymbol{g}_t\|_2 \le G$ for all t. Then

$$\frac{1}{T}\sum_{t=1}^{T} \ell(y_t, \hat{y}_t) - \min_{f \in \mathcal{F}} \frac{1}{T}\sum_{t=1}^{T} \ell(y_t, f(\boldsymbol{x}_t)) \;\le\; \frac{1}{\eta} \cdot \frac{D^2}{2T} + \eta \cdot \frac{G^2}{2}.$$

Online Gradient Descent (OGD)

1: Choose learning rate η.
2: Initialize weights $\boldsymbol{w}_1 \in \boldsymbol{W}$ arbitrarily.
3: **for** $t = 1, 2, \ldots, T$ **do**
4: Obtain $\boldsymbol{x}_t \in \mathbb{R}^d$
5: Predict $\hat{y}_t = f_{\boldsymbol{w}_t}(\boldsymbol{x}_t)$
6: Observe y_t, and incur loss $\ell(y_t, \hat{y}_t)$.
7: Compute the gradient $\nabla_{\boldsymbol{w}} \ell(y_t, f_{\boldsymbol{w}}(\boldsymbol{x}_t))$ at $\boldsymbol{w}_t$, call it $\boldsymbol{g}_t$.
8: Update:

$$\boldsymbol{w}_{t+1} \;=\; \Pi_{\boldsymbol{W}}(\boldsymbol{w}_t - \eta \boldsymbol{g}_t)$$

9: **end for**

Regret of OGD after T rounds: Suppose for any $\boldsymbol{w}, \boldsymbol{w}' \in \boldsymbol{W}$, we have $\|\boldsymbol{w} - \boldsymbol{w}'\|_2 \le D$ and $\|\boldsymbol{g}_t\|_2 \le G$ for all t. Then

$$\frac{1}{T}\sum_{t=1}^{T} \ell(y_t, \hat{y}_t) - \min_{f \in \mathcal{F}} \frac{1}{T}\sum_{t=1}^{T} \ell(y_t, f(\boldsymbol{x}_t)) \;\le\; \frac{1}{\eta} \cdot \frac{D^2}{2T} + \eta \cdot \frac{G^2}{2}.$$

Optimize bound with $\eta := \sqrt{\frac{D}{GT}}$ to get regret bound $\sqrt{\frac{GD}{4T}}$.

Online Gradient Descent (OGD)

1: Choose learning rate η.
2: Initialize weights $\boldsymbol{w}_1 \in \boldsymbol{W}$ arbitrarily.
3: **for** $t = 1, 2, \ldots, T$ **do**
4: Obtain $\boldsymbol{x}_t \in \mathbb{R}^d$
5: Predict $\hat{y}_t = f_{\boldsymbol{w}_t}(\boldsymbol{x}_t)$
6: Observe y_t, and incur loss $\ell(y_t, \hat{y}_t)$.
7: Compute the gradient $\nabla_{\boldsymbol{w}} \ell(y_t, f_{\boldsymbol{w}}(\boldsymbol{x}_t))$ at $\boldsymbol{w}_t$, call it $\boldsymbol{g}_t$.
8: Update:

$$\boldsymbol{w}_{t+1} \;=\; \Pi_{\boldsymbol{W}}(\boldsymbol{w}_t - \eta \boldsymbol{g}_t)$$

9: **end for**

Regret of OGD after T rounds: Suppose for any $\boldsymbol{w}, \boldsymbol{w}' \in \boldsymbol{W}$, we have $\|\boldsymbol{w} - \boldsymbol{w}'\|_2 \le D$ and $\|\boldsymbol{g}_t\|_2 \le G$ for all t. Then

$$\frac{1}{T}\sum_{t=1}^{T} \ell(y_t, \hat{y}_t) - \min_{f \in \mathcal{F}} \frac{1}{T}\sum_{t=1}^{T} \ell(y_t, f(\boldsymbol{x}_t)) \;\le\; \frac{1}{\eta} \cdot \frac{D^2}{2T} + \eta \cdot \frac{G^2}{2}.$$

Optimize bound with $\eta := \sqrt{\frac{D}{GT}}$ to get regret bound $\sqrt{\frac{GD}{4T}}$.

Could also use $\eta_t := \sqrt{\frac{D}{Gt}}$ in round t.

Application to "big data" learning

- Online learning algorithms can be easily converted to batch learning algorithms
- **Online-to-batch conversion:** output a random predictor

Application to "big data" learning

- Online learning algorithms can be easily converted to batch learning algorithms
- **Online-to-batch conversion:** output a random predictor
- Can have high variance, so some form of averaging is useful (if possible)
- E.g. in online convex optimization settings, output average parameter vector.

Application to "big data" learning

- Online learning algorithms can be easily converted to batch learning algorithms
- **Online-to-batch conversion:** output a random predictor
- Can have high variance, so some form of averaging is useful (if possible)
- E.g. in online convex optimization settings, output average parameter vector.
- Utility: can process "big data" by learning example-by-example
- In fact, SGD analysis directly obtained from online learning techniques.

More on online learning

Many problems and algorithms in online learning:

- Partial feedback ("multi-armed bandit")
- Bayesian inference
- Tracking / filtering
- ...

Many connections between online learning and other subjects:

- Game theory, calibration, boosting
- Stochastic optimization
- Compression, coding theory
- ...