Online convex optimization

Source: David Childer's 73-423 note

Online learning

- Often, prediction must be updated in light of new data, and decisions made on an ongoing basis

Online learning

- Often, prediction must be updated in light of new data, and decisions made on an ongoing basis

- May even need to choose a procedure in which process is entirely automated

Online learning

- Often, prediction must be updated in light of new data, and decisions made on an ongoing basis
- May even need to choose a procedure in which process is entirely automated
- Example use cases
  - Web services: may need to update page every day or even every visitor

Online learning

- Often, prediction must be updated in light of new data, and decisions made on an ongoing basis
- May even need to choose a procedure in which process is entirely automated
- Example use cases
  - Web services: may need to update page every day or even every visitor
  - Large retail operation: need to predict inventories for each outlet each day

Online learning

- Want a procedure which is both automated and robust
  - Not enough time for practitioner to apply judgment each time model must be changed
  - If something goes wrong, might not be able to intervene immediately

Online learning

- Want a procedure which is both automated and robust
  - Not enough time for practitioner to apply judgment each time model must be changed
  - If something goes wrong, might not be able to intervene immediately
- These goals motivate **online learning** methods
  - A class of algorithms for making a sequence of repeated forecasting decisions

Online learning problem

Consider the **sequential prediction** task:

Online learning problem

Consider the **sequential prediction** task:

- At each time step t, we have a model M_t and receive a predictor $x_t \in \mathbb{R}^d$

Online learning problem

Consider the **sequential prediction** task:

- At each time step t, we have a model M_t and receive a predictor $x_t \in \mathbb{R}^d$
- Output a prediction $M_t(x_t)$

Online learning problem

Consider the **sequential prediction** task:

- At each time step t, we have a model M_t and receive a predictor $x_t \in \mathbb{R}^d$
- Output a prediction $M_t(x_t)$
- Receive the true label y_t and we compute our **loss** $\ell(M_t(x_t), y_t)$

Online learning problem

Consider the **sequential prediction** task:

- At each time step t, we have a model M_t and receive a predictor $x_t \in \mathbb{R}^d$
- Output a prediction $M_t(x_t)$
- Receive the true label y_t and we compute our **loss** $\ell(M_t(x_t), y_t)$
- Update our model to M_{t+1}, repeat for $t = 1, \ldots, T$

Online learning problem

Consider the **sequential prediction** task:

- At each time step t, we have a model M_t and receive a predictor $x_t \in \mathbb{R}^d$
- Output a prediction $M_t(x_t)$
- Receive the true label y_t and we compute our **loss** $\ell(M_t(x_t), y_t)$
- Update our model to M_{t+1}, repeat for $t = 1, \ldots, T$

What do we want to optimize?

- Let M_* be the **best** model if we were to observe $(x_1, y_1), \ldots, (x_T, y_T)$ from the beginning

Online learning problem

Consider the **sequential prediction** task:

- At each time step t, we have a model M_t and receive a predictor $x_t \in \mathbb{R}^d$
- Output a prediction $M_t(x_t)$
- Receive the true label y_t and we compute our **loss** $\ell(M_t(x_t), y_t)$
- Update our model to M_{t+1}, repeat for $t = 1, \ldots, T$

What do we want to optimize?

- Let M_* be the **best** model if we were to observe $(x_1, y_1), \ldots, (x_T, y_T)$ from the beginning
- Then, M_* minimizes the **total loss**

$$\sum_{t=1}^{T} \ell(M_*(x_t), y_t)$$

Regret

We want out models $M_1, \ldots, M_T$ to perform well compared to the best model M_*

Regret

We want out models $M_1, \ldots, M_T$ to perform well compared to the best model M_*

The performance is measured in total loss, so we want to minimize the so-called **regret**:

$$\text{Regret}_T = \max_{y_1,\ldots,y_T} \underbrace{\sum_{t=1}^{T} \ell(M_t(x_t), y_t)}_{\text{loss from online learning}} - \underbrace{\sum_{t=1}^{T} \ell(M_*(x_t), y_t)}_{\text{loss from the best model}}$$

Regret

We want out models $M_1, \ldots, M_T$ to perform well compared to the best model M_*

The performance is measured in total loss, so we want to minimize the so-called **regret**:

$$\text{Regret}_T = \max_{y_1,\ldots,y_T} \underbrace{\sum_{t=1}^{T} \ell(M_t(x_t), y_t)}_{\text{loss from online learning}} - \underbrace{\sum_{t=1}^{T} \ell(M_*(x_t), y_t)}_{\text{loss from the best model}}$$

Of course Regret_T grows as a function of T but how slow does it grow?

For example, $\text{Regret}_T \approx \sqrt{T}$ is better than $\text{Regret}_T \approx T$

Example: online classification

Suppose that we have a stream of data $x_1, y_1, x_2, y_2, \ldots, x_T, y_T$, where $x_t \in \mathbb{R}^d$ and $y_t \in \{-1, 1\}$

Example: online classification

Suppose that we have a stream of data $x_1, y_1, x_2, y_2, \ldots, x_T, y_T$, where $x_t \in \mathbb{R}^d$ and $y_t \in \{-1, 1\}$

We will consider the $0-1$ loss:

$$\ell(M(x_t), y_t) = \begin{cases} 0 & \text{if } M(y_t) = y_t \\ 1 & \text{if } M(y_t) \neq y_t \end{cases}$$

that is, a loss occurs when M makes a wrong prediction

Example: online classification

Suppose that we have a stream of data $x_1, y_1, x_2, y_2, \ldots, x_T, y_T$, where $x_t \in \mathbb{R}^d$ and $y_t \in \{-1, 1\}$

We will consider the $0-1$ loss:

$$\ell(M(x_t), y_t) = \begin{cases} 0 & \text{if } M(y_t) = y_t \\ 1 & \text{if } M(y_t) \neq y_t \end{cases}$$

that is, a loss occurs when M makes a wrong prediction

We will use a linear model: $M_t(x_t) = \text{sign}(\theta_t^T x_t)$. Here, θ_t is the vector of coefficients at time t

Perceptron Algorithm (Rosenblatt, 1958)

Input: A stream of data $x_1, y_1, x_2, y_2, \ldots, x_T, y_T$, where $x_t \in \mathbb{R}^d$ and $y_t \in \{-1, 1\}$

Initialize the coefficients $\theta_1 \in \mathbb{R}^d$

Perceptron Algorithm (Rosenblatt, 1958)

Input: A stream of data $x_1, y_1, x_2, y_2, \ldots, x_T, y_T$, where $x_t \in \mathbb{R}^d$ and $y_t \in \{-1, 1\}$

Initialize the coefficients $\theta_1 \in \mathbb{R}^d$

for $t = 1$ **to** T **do**

Perceptron Algorithm (Rosenblatt, 1958)

Input: A stream of data $x_1, y_1, x_2, y_2, \ldots, x_T, y_T$, where $x_t \in \mathbb{R}^d$ and $y_t \in \{-1, 1\}$

Initialize the coefficients $\theta_1 \in \mathbb{R}^d$

for $t = 1$ **to** T **do**

1. Receive $x_t \in \mathbb{R}^d$

Perceptron Algorithm (Rosenblatt, 1958)

Input: A stream of data $x_1, y_1, x_2, y_2, \ldots, x_T, y_T$, where $x_t \in \mathbb{R}^d$ and $y_t \in \{-1, 1\}$

Initialize the coefficients $\theta_1 \in \mathbb{R}^d$

for $t = 1$ **to** T **do**

1. Receive $x_t \in \mathbb{R}^d$
2. Predict $\hat{y}_t = \text{sign}(\theta_t^T x_t)$

Perceptron Algorithm (Rosenblatt, 1958)

Input: A stream of data $x_1, y_1, x_2, y_2, \ldots, x_T, y_T$, where $x_t \in \mathbb{R}^d$ and $y_t \in \{-1, 1\}$

Initialize the coefficients $\theta_1 \in \mathbb{R}^d$

for $t = 1$ **to** T **do**

1. Receive $x_t \in \mathbb{R}^d$
2. Predict $\hat{y}_t = \text{sign}(\theta_t^T x_t)$
3. Receive y_t and pay $\ell_t = 1$ if $\hat{y}_t \neq y_t$, $\ell_t = 0$ otherwise

Perceptron Algorithm (Rosenblatt, 1958)

Input: A stream of data $x_1, y_1, x_2, y_2, \ldots, x_T, y_T$, where $x_t \in \mathbb{R}^d$ and $y_t \in \{-1, 1\}$

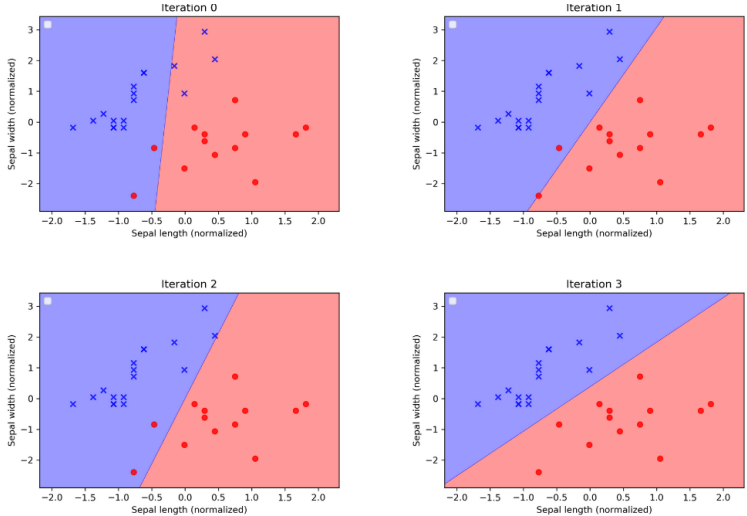
Initialize the coefficients $\theta_1 \in \mathbb{R}^d$

for $t = 1$ **to** T **do**

1. Receive $x_t \in \mathbb{R}^d$
2. Predict $\hat{y}_t = \text{sign}(\theta_t^T x_t)$
3. Receive y_t and pay $\ell_t = 1$ if $\hat{y}_t \neq y_t$, $\ell_t = 0$ otherwise
4. $\theta_{t+1} = \theta_t + \ell_t y_t x_t$

end

Example: Iris dataset



Source: Simone Alberto Peirone (2019)

Follow the leader

Suppose that we have a stream of data $x_1, y_1, x_2, y_2, \ldots, x_T, y_T$, where $x_t \in \mathbb{R}^d$ and $y_t \in \mathbb{R}$

Let M_{θ_t} be our model with parameters θ_t at step t

Follow the leader

Suppose that we have a stream of data $x_1, y_1, x_2, y_2, \ldots, x_T, y_T$, where $x_t \in \mathbb{R}^d$ and $y_t \in \mathbb{R}$

Let M_{θ_t} be our model with parameters θ_t at step t

- For example, M_{θ_t} is a linear regression with coefficients θ_t at step t: $M_{\theta_t}(x) = \theta_t^T x$

Follow the leader

Suppose that we have a stream of data $x_1, y_1, x_2, y_2, \ldots, x_T, y_T$, where $x_t \in \mathbb{R}^d$ and $y_t \in \mathbb{R}$

Let M_{θ_t} be our model with parameters θ_t at step t

- For example, M_{θ_t} is a linear regression with coefficients θ_t at step t: $M_{\theta_t}(x) = \theta_t^T x$

Why not pick coefficients θ_t which have performed best so far?

- After step t, find θ_t that minimizes the total loss:

$$\sum_{s=1}^{t-1} \ell(M_{\theta_t}(x_s), y_s)$$

This method of choosing θ_t is called **follow the leader (FTL)**

Follow The Leader

$$\text{Choose } \theta_t \text{ that minimizes } \sum_{s=1}^{t-1} \ell(M_{\theta_t}(x_s), y_s)$$

- Exactly equivalent to convex optimization with data $(x_1, y_1), \ldots, (x_{t-1}, y_{t-1})$

- So does it work? Can we just do what we were already doing and get good results in the online setting?

Follow The Leader

$$\text{Choose } \theta_t \text{ that minimizes } \sum_{s=1}^{t-1} \ell(M_{\theta_t}(x_s), y_s)$$

- Exactly equivalent to convex optimization with data $(x_1, y_1), \ldots, (x_{t-1}, y_{t-1})$

- So does it work? Can we just do what we were already doing and get good results in the online setting?
  - In general, **no**
  - For general convex loss functions, regret can grow linearly in T

Worst Case Behavior of Follow The Leader

Consider a stream of data

$$x_t : -0.5, 1, -1, 1, -1, 1, -1, \ldots$$

Worst Case Behavior of Follow The Leader

Consider a stream of data

$$x_t : -0.5, 1, -1, 1, -1, 1, -1, \ldots$$

We restrict $\theta \in [-1, 1]$ and use the loss $\ell_\theta(x_t) = \theta x_t$

Worst Case Behavior of Follow The Leader

Consider a stream of data

$$x_t : -0.5, 1, -1, 1, -1, 1, -1, \ldots$$

We restrict $\theta \in [-1, 1]$ and use the loss $\ell_\theta(x_t) = \theta x_t$

Then, at step t,

$$\sum_{s=1}^{t-1} \ell_{\theta_t}(x_s) = (-0.5 + 1 - 1 + \ldots)\theta_t = \begin{cases} -0.5\theta_t & t \text{ even} \\ 0.5\theta_t & t \text{ odd} \end{cases}$$

Worst Case Behavior of Follow The Leader

Consider a stream of data

$$x_t : -0.5, 1, -1, 1, -1, 1, -1, \ldots$$

We restrict $\theta \in [-1, 1]$ and use the loss $\ell_\theta(x_t) = \theta x_t$

Then, at step t,

$$\sum_{s=1}^{t-1} \ell_{\theta_t}(x_s) = (-0.5 + 1 - 1 + \ldots)\theta_t = \begin{cases} -0.5\theta_t & t \text{ even} \\ 0.5\theta_t & t \text{ odd} \end{cases}$$

θ_t that minimizes this function is $\theta_t = 1$ when t is even, and $\theta_t = -1$ when t is odd

Worst Case Behavior of Follow The Leader

Consider a stream of data

$$x_t : -0.5, 1, -1, 1, -1, 1, -1, \ldots$$

We restrict $\theta \in [-1, 1]$ and use the loss $\ell_\theta(x_t) = \theta x_t$

Then, at step t,

$$\sum_{s=1}^{t-1} \ell_{\theta_t}(x_s) = (-0.5 + 1 - 1 + \ldots)\theta_t = \begin{cases} -0.5\theta_t & t \text{ even} \\ 0.5\theta_t & t \text{ odd} \end{cases}$$

θ_t that minimizes this function is $\theta_t = 1$ when t is even, and $\theta_t = -1$ when t is odd

So the loss is

$$\sum_{t=1}^{T} \ell_{\theta_t}(x_t) = \begin{cases} -0.5\theta_1 + T - 1 & t \text{ even} \\ -0.5\theta_1 + T - 1 & t \text{ odd} \end{cases}$$

Worst Case Behavior of Follow The Leader

But the best choice of θ is 0, since then we would have

$$\sum_{s=1}^{t} \ell_\theta(x_t) = (-0.5 + 1 - 1 + \ldots)\theta = 0$$

Thus, our regret is $(-0.5\theta_1 + T - 1) - 0 = O(T)$