COS324: Introduction to Machine Learning

Lecture 2: Online learning

Prof. Elad Hazan & Prof. Yoram Singer

Learning from experts' advice

- Previous lecture introduced setting called online learning
- Numerous applications in temporal prediction problems
- Target need not be binary ($\{-1, +1\}$), e.g. $y^t \in \mathbb{R}^d$
  - Weather (temperature, precipitation, wind, ...)
  - Seismic activities
  - Financial markets
  - Reactive systems (drones, self-driving cars, ...)

Online learning

- Initialize $\mathbf{w}^1$; $\mathcal{L}^1 = 0$
- For $t = 1, 2, \ldots, T, \ldots$
  1. Predict $\hat{y}^t$
  2. Observe true outcome y^t
  3. Endure loss: $\ell^t = \ell(y^t, \hat{y}^t)$; $\mathcal{L}^{t+1} = \mathcal{L}^t + \ell^t$
  4. Update $\mathbf{w}^{t+1} := F(\mathbf{w}^t, \mathbf{x}^t, y^t)$

- Classification loss $\ell(y, y') = 1$ if $y \neq y'$ and 0 o.w.
- When none of the experts is always consistent
  - `Consistent` will get out-of-bound
  - `Halving` will end with a zero vector ($\exists T$ s.t. $\forall t \geq T\ \mathbf{w}^t = \mathbf{0}$)

Online learning: what can be said in general?

- With consistent expert $\Rightarrow$ can achieve low # of errors
- Analoguous statement for experts which make errors?

Weighted Majority

- Classification learning, $0 - 1$ loss
- Assign real-valued weight to experts $w_j^t \in [0, 1]$
- Rather than eliminating erronous experts demote them
$$x_j^t \neq y^t \;\Rightarrow\; w_j^{t+1} < w_j^t$$
- Replace simple majority rule with weighted majority
- Pro: no expert left behind...
- Con: need to introduce demotion parameter $0 < \eta \ll 1$

WM algorithm by Littlestone and Warmuth, 1989

Weighted Majority Algorithm

- Initialize $\mathbf{w}^1 = \mathbf{1}$; $\mathcal{L}^1 = 0$
- For $t = 1, 2, \ldots, T, \ldots$
  1. Observe predictions $\mathbf{x}^t \in \{-1, +1\}^n$
  2. Predict $\hat{y}^t := \text{sign}(\mathbf{w}^t \cdot \mathbf{x}^t)$
  3. Observe true outcome y^t
  4. Endure loss: $\ell^t = \mathbf{1}\left[y^t \neq \hat{y}^t\right]$; $\mathcal{L}^{t+1} = \mathcal{L}^t + \ell^t$
  5. Update:
$$w_j^{t+1} = \begin{cases} w_j^t & x_j^t = y^t \\ (1-\eta)w_j^t & x_j^t \neq y^t \end{cases}$$

Intuitive explanation

- Suppose there exists an accurate expert (AE) albeit not perfect
- Four possible cases on round t:

1. AE correct & WM incorrect
   total mass of erroneous experts decreases by $1 - \eta$
   AE stayed the same and improved it relative standing
2. AE correct & WM correct
   some mass of erroneous experts decreases by $1 - \eta$
   AE stayed the same and may improved a little
3. AE incorrect & WM correct
   does not happen quite often & WM is still fine
4. AE & WM predicted incorrectly
   AE is in the same boat with other accurate experts

Analysis: bounding the number of mistakes

$\mathcal{L}_i^T$ number of mistakes made by expert i during $t = 1, ..., T$

$\mathcal{L}^T$ number of mistakes WM made during during $t = 1, ..., T$

Theorem: For every sequence $(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^T, y^T)$ the number of mistakes of WM is at most,

$$\forall i \in [n] : \quad \mathcal{L}^T \ \leq \ 2(1+\eta)\mathcal{L}_i^T \ + \ \frac{2\log(\mathrm{n})}{\eta}$$

Bound holds for all i in particular for the best during $t = 1, \ldots, T$

Multiplicative factor $2(1+\eta)$; Additive factor $O(\log(n))$

Analysis: bounding the number of mistakes

- Let i^* be the best expert in hindsight – the one who made the least number of mistakes in retrospect

- Let $\Phi^t = \sum_{i=1}^{n} w_i^t$ denote the total mass at round t

  Clearly, $\Phi^t \geq \Phi^{t+1} \geq \Phi^{t+2} \ldots$

- Use shorthand $m \stackrel{\text{def}}{=} \mathcal{L}^T$, $m^\star \stackrel{\text{def}}{=} \mathcal{L}_{i^\star}^T$

Observation I

$$
\begin{aligned}
\Phi^T &= \sum_{i=1}^{n} w_i^T \\
&\geq w_{i^*}^T \\
&= w_{i^*}^1 (1-\eta)^{m^\star} \\
&= (1-\eta)^{m^\star}
\end{aligned}
$$

Observation II

We prove that $\Phi^T \leq \Phi^1 (1-\eta/2)^m$

- WM predicted correctly on iteration t: $\Phi^{t+1} \leq \Phi^t$
- WM predicted incorrectly on iteration t:

$$\Phi^{t+1} = \sum_{i:x_i^t \neq y^t} w_i^{t+1} + \sum_{i:x_i^t = y^t} w_i^{t+1}$$
$$= \sum_{i:x_i^t \neq y^t} (1-\eta)\, w_i^t + \sum_{i:x_i^t = y^t} w_i^t$$

- Define $\sigma_\mathrm{e} = \sum_{i:x_i^t \neq y^t} w_i^t \quad \sigma_\mathrm{C} = \sum_{i:x_i^t = y^t} w_i^t$ and rewrite

$$\Phi^{t+1} = (1-\eta)\sigma_\mathrm{e} + \sigma_\mathrm{C} = (1-\eta)\Phi^t + \eta\sigma_\mathrm{C}$$

- Last, when WM predicts incorrectly $\sum_{i:x_i^t = y^t} w_i \leq \frac{1}{2}\Phi^t$

Analysis (cont.)

- We therefore get

$$\Phi^{t+1} \leq (1-\eta)\Phi^t + \eta\frac{1}{2}\Phi^t = (1-\eta/2)\Phi^t$$

- Unraveling the recursion for each round WM was mistaken

$$\Phi^{t+1} \leq \Phi^1(1-\eta/2)^m = n(1-\eta/2)^m$$

- Combining the observations we get

$$(1-\eta)^{m^\star} \leq \Phi^T \leq n(1-\eta/2)^m$$

- Taking the logarithm of both sides,

$$m^\star \log(1-\eta) \leq m \log(1-\eta/2) + \log(n)$$

Analysis – Refinment

- We use Taylor approximation for $a < \frac{1}{2}$:

$$-a - a^2 \leq \log(1-a) \leq -a$$

and bound

$$-\eta - \eta^2 \leq \log(1-\eta) \quad \log(1-\eta/2) \leq -\eta/2$$

- Using the lower and upper bounds:

$$m^\star \left(-\eta - \eta^2\right) \leq m\left(-\frac{\eta}{2}\right) + \log(n)$$

- Diving by $\eta/2$ and rearranging:

$$m \leq \frac{2\log(n)}{\eta} + 2(1+\eta)m^\star$$

which means

$$\mathcal{L}^T \leq 2(1+\eta)\mathcal{L}_i^T + \frac{2\log(n)}{\eta}$$

Comments on Weighted Majority

- The algorithm is deterministic and its running time linear
- Optimal choice of η (requires to knowledge of $m^\star$) gives:

$$\mathcal{L}^T \leq 2\mathcal{L}^T_{i^*} + 4\sqrt{\mathcal{L}^T_{i^*}\log(n)} + 2\log(n)$$

- It is possible to adaptively change η for each round and get:

$$\mathcal{L}^T \leq 2\mathcal{L}^T_{i^*} + 8\sqrt{\mathcal{L}^T_{i^*}\log(n)} + 4\log(n)$$

- Is possible to achieve lower error?

Comments on Weighted Majority

- Theorem [Littlestone-Warmuth]: any deterministic algorithm is bound to make at least $2\mathcal{L}_{i^*}^T$ mistakes
- Prove?
- Can we do better using a non-deterministic algorithm?

Bag Of Words (BOW) model

- Pre-defined dictionary of n tokens (words, html, arch-codes)

kale	1
plate	2
kohlrabi	3
ate	4
fork	5

- Represent a document as a vector $\mathbf{x} \in \{-1, +1\}^n$ s.t. $x_j = +1$ iff token j appears in document
- Tokens not in the dictionary are ignored
- Examples:

$$\text{"The kohlrabi ate kale on a plate"} \mapsto (+1, +1, +1, +1, -1)$$

$$\text{"A monkey ate a banana with a fork"} \mapsto (-1, -1, -1, +1, +1)$$

BOW + WM ⇒ Text Classifier

- Each dictionary word is an expert
- Initialize weight of experts $\mathbf{w}^1 = \mathbf{1}$
- For $t = 1, \ldots, m$: // m is #document
  - Convert document t to a vector $\mathbf{x}^t \in \{-1, +1\}^n$
  - Update weights using WM with provided tagging y^t: $\mathbf{w}^t \rightsquigarrow \mathbf{w}^{t+1}$
- Output $\mathbf{w}^{m+1}$

Wait, but what if $\nexists$ *single* accurate expert ?
Do we obtain a good classifier? **Yes!**