Machine learning theory

Ranking

Hamid Beigy

Sharif university of technology

May 18, 2020

Table of contents

Introduction

- The learning to rank problem is how to learn an ordering.
- Application in very large datasets
  - search engines,
  - information retrieval
  - fraud detection
  - movie recommendation

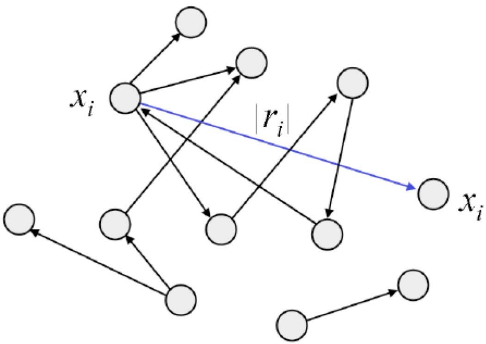
main motivation for ranking over classification in the binary case is the limitation of resources

1. it may be impractical or even impossible to display or process all items labeled as relevant by a classifier.
2. we need to show more relevant ones or prioritize them.

- In these applications, ranking is more desirable than classification.
- **Problem:** Can we learn to predict ranking accurately?
- Ranking scenarios
  1. score-based setting
  2. preference-based setting

Score-based setting

- General supervised learning problem of ranking,
  1. the learner receives labeled sample of pairwise preferences,
  2. the learner outputs a scoring function $h : \mathcal{X} \mapsto \mathbb{R}$.
- Drawbacks
  1. h induces a linear ordering for full set $\mathcal{X}$
  2. does not match a query-based scenario.
- Advantages
  1. efficient algorithms
  2. good theory,
  3. VC bounds,
  4. margin bounds,
  5. stability bounds



Shivani Agarwal

- The score-based setting is defined as
  1. $\mathcal{X}$ is input space.
  2. $\mathcal{D}$ is unknown distribution over $\mathcal{X} \times \mathcal{X}$.
  3. $f : \mathcal{X} \times \mathcal{X} \mapsto \{-1, 0, +1\}$ is arget labeling function or preference function, where

$$f(\mathbf{x}, \mathbf{x}') = \begin{cases} -1 & \text{if } \mathbf{x}' \prec_{pref} \mathbf{x} \\ 0 & \text{if } \mathbf{x}' =_{pref} \mathbf{x} \\ +1 & \text{if } \mathbf{x} \prec_{pref} \mathbf{x}' \end{cases}$$

- No assumption is made about the transitivity of the order induced by f.

$$f(\mathbf{x}, \mathbf{x}') = +1 \quad \text{and} \quad f(\mathbf{x}', \mathbf{x}'') = +1 \quad \text{and} \quad f(\mathbf{x}'', \mathbf{x}) = +1$$

- No assumption is made about the antisymmetry of the order induced

$$f(\mathbf{x}, \mathbf{x}') = +1 \quad \text{and} \quad f(\mathbf{x}', \mathbf{x}) = +1 \quad \text{and} \quad \mathbf{x} \neq \mathbf{x}'$$

Definition (Learning to rank (score-based setting))

1. Learner receives $S = \{(\mathbf{x}_1, \mathbf{x}'_1, y_1), \ldots, (\mathbf{x}_m, \mathbf{x}'_m, y_m)\} \in (\mathcal{X} \times \mathcal{X} \mapsto \{-1, 0, +1\})^m$, where $(\mathbf{x}_i, \mathbf{x}'_i) \sim \mathcal{D}$ and $y_i = f(\mathbf{x}_i, \mathbf{x}'_i)$.
2. Given a hypothesis set $H = \{h : \mathcal{X} \mapsto \mathbb{R}\}$, ranking problem consists of selecting a hypothesis $h \in H$ with small expected pairwise misranking or generalization error $\mathbf{R}(h)$ with respect to the target f

$$\mathbf{R}(h) = \underset{(\mathbf{x}, \mathbf{x}') \sim \mathcal{D}}{\mathbb{P}} \left[(f(\mathbf{x}, \mathbf{x}') \neq 0) \wedge (f(\mathbf{x}, \mathbf{x}')(h(\mathbf{x}) - h(\mathbf{x}')) \leq 0) \right]$$

3. The empirical pairwise misranking or empirical error of h is defined by

$$\hat{\mathbf{R}}(h) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I} \left[(y_i \neq 0) \wedge (y_i(h(\mathbf{x}_i) - h(\mathbf{x}'_i)) \leq 0) \right]$$

- A simple approach is to project instances into a vector $\mathbf{w}$
- Let to define the ranking function as

$$h((\mathbf{x}_1, \ldots, \mathbf{x}_m)) = (\langle \mathbf{w}, \mathbf{x}_1 \rangle, \ldots, \langle \mathbf{w}, \mathbf{x}_m \rangle)$$

- Then use the distance of the point to classifier $\langle \mathbf{w}, \mathbf{x} \rangle$ as the score of $\mathbf{x}$.
- We assume that $y_i \neq 0$, then the empirical error is defined as

$$\hat{\mathbf{R}}(h) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\left[(y_i(h(\mathbf{x}_i) - h(\mathbf{x}'_i)) \leq 0)\right]$$

- if we define $h(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$, we have

$$\hat{\mathbf{R}}(h) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\left[(y_i \left\langle \mathbf{w}, (\mathbf{x}_i - \mathbf{x}'_i) \right\rangle \leq 0)\right]$$

- Then, we can use the following ERM algorithm to rank items.

$$\mathbf{w} = \operatorname*{argmin}_{w'} \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\left[(y_i \left\langle \mathbf{w}', (\mathbf{x}_i - \mathbf{x}'_i) \right\rangle \leq 0)\right]$$

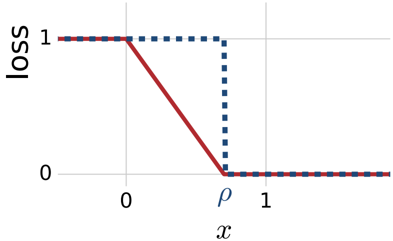
- Assume that labels are chosen from $\{-1, +1\}$.
- **Homework:** Generalize the result to the label set $\{-1, 0, +1\}$.
- Same as classification, for any $\rho > 0$, empirical margin loss of a hypothesis h for pairwise ranking is

$$\hat{\mathbf{R}}_\rho(h) = \frac{1}{m}\sum_{i=1}^{m} \Phi_\rho(y_i(h(\mathbf{x}_i') - h(\mathbf{x}_i)))$$

where

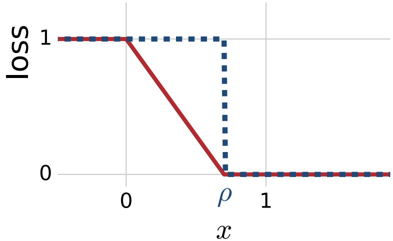
$$\Phi_\rho(u) = \begin{cases} 1 & \text{if } u \le 0 \\ 1 - \frac{u}{\rho} & \text{if } 0 \le u \le \rho \\ 0 & \text{if } \rho \ge u \end{cases}$$

- The parameter $\rho > 0$ can be interpreted as the confidence margin demanded from a hypothesis h.



- The upper bound of empirical margin loss of a hypothesis h is

$$\hat{\mathbf{R}}_\rho(h) \leq \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}\left[y_i(h(\mathbf{x}_i') - h(\mathbf{x}_i)) \leq \rho\right]$$



- Define
  1. $\mathcal{D}_1$ as the marginal distribution of the first element of the pairs $\mathcal{X} \times \mathcal{X}$ derived from $\mathcal{D}$,
  2. $\mathcal{D}_2$ as the marginal distribution of the secons element of the pairs $\mathcal{X} \times \mathcal{X}$ derived from $\mathcal{D}$,
  3. $S_1 = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ and $\mathcal{R}_m^{\mathcal{D}_1}(H)$ as the Rademacher complexity of H with respect to $\mathcal{D}_1$,
  4. $S_2 = \{(\mathbf{x}_1', y_1), \ldots, (\mathbf{x}_m', y_m)\}$ and $\mathcal{R}_m^{\mathcal{D}_2}(H)$ as the Rademacher complexity of H with respect to $\mathcal{D}_2$,
- We also have $\mathcal{R}_m^{\mathcal{D}_1}(H) = \mathbb{E}\left[\hat{\mathcal{R}}_{S_1}(H)\right]$ and $\mathcal{R}_m^{\mathcal{D}_2}(H) = \mathbb{E}\left[\hat{\mathcal{R}}_{S_2}(H)\right]$.
- If $\mathcal{D}$ is symmetric, then $\mathcal{R}_m^{\mathcal{D}_1}(H) = \mathcal{R}_m^{\mathcal{D}_2}(H)$.

Theorem (Margin bound for ranking)

Let H be a set of real-valued functions. Fix $\rho > 0$, then, for any $\delta > 0$, with probability at least $(1-\delta)$ over the choice of a sample S of size m, each of the following holds for all $h \in H$

$$\mathbf{R}(h) \leq \hat{\mathbf{R}}_\rho(h) + \frac{2}{\rho}\left(\mathcal{R}_m^{\mathcal{D}_1}(H) + \mathcal{R}_m^{\mathcal{D}_2}(H)\right) + \sqrt{\frac{\log(1/\delta)}{2m}}$$

$$\mathbf{R}(h) \leq \hat{\mathbf{R}}_\rho(h) + \frac{2}{\rho}\left(\hat{\mathcal{R}}_{S_1}(H) + \hat{\mathcal{R}}_{S_2}(H)\right) + 3\sqrt{\frac{\log(2/\delta)}{2m}}$$

Proof (Margin bound for ranking).

1. Consider the family of functions $\tilde{H} = \left\{\Phi_\rho \circ f \;\middle|\; f \in \tilde{H}\right\}$.
2. From margin-loss bounds we have

$$\mathbb{E}\left[\Phi_\rho(y[h(\mathbf{x}') - h(\mathbf{x}))\right] \leq \hat{\mathbf{R}}_\rho(h) + 2\mathcal{R}_m(\Phi_\rho \circ \tilde{H}) + \sqrt{\frac{\log(1/\delta)}{2m}}.$$

3. Since for all $u \in \mathbb{R}$, we have $\mathbb{I}\left[u \leq 0\right] \leq \Phi_\rho(u)$, then we have

$$\mathbf{R}(h) = \mathbb{E}\left[\mathbb{I}\left[y(h(\mathbf{x}') - h(\mathbf{x})) \leq 0\right]\right] \leq \mathbb{E}\left[\Phi_\rho(y[h(\mathbf{x}') - h(\mathbf{x}))\right]$$

4. Hence, we can write

$$\mathbf{R}(h) \leq \hat{\mathbf{R}}_\rho(h) + 2\mathcal{R}_m(\Phi_\rho \circ \tilde{H}) + \sqrt{\frac{\log(1/\delta)}{2m}}.$$

Proof (Margin bound for ranking)(cont.).

5. Since Φ_ρ is $1/\rho-$ *Lipschitz*, by Talagrand's lemma $\mathcal{R}_m(\Phi_\rho \circ \tilde{H}) \leq \frac{1}{\rho}\mathcal{R}_m(\tilde{H})$.
6. Here, $\mathcal{R}_m(\tilde{H})$ can be upper bounded as

$$\begin{aligned}
\mathcal{R}_m(\tilde{H}) &= \frac{1}{m} \mathop{\mathbb{E}}_{S,\sigma}\left[\sup_{h\in H}\sum_{i=1}^{m} \sigma_i y_i (h(\mathbf{x}_i') - h(\mathbf{x}_i))\right] \\
&= \frac{1}{m} \mathop{\mathbb{E}}_{S,\sigma}\left[\sup_{h\in H}\sum_{i=1}^{m} \sigma_i (h(\mathbf{x}_i') - h(\mathbf{x}_i))\right] && \sigma_i y_i \text{ and } \sigma_i \text{ : same distribution} \\
&\leq \frac{1}{m} \mathop{\mathbb{E}}_{S,\sigma}\left[\sup_{h\in H}\sum_{i=1}^{m} \sigma_i h(\mathbf{x}_i') + \sup_{h\in H}\sum_{i=1}^{m} \sigma_i h(\mathbf{x}_i)\right] && \text{by sub-additivity of sup} \\
&\leq \mathop{\mathbb{E}}_{S}\left[\hat{\mathcal{R}}_{S_1}(H) + \hat{\mathcal{R}}_{S_2}(H)\right] && \text{definition of } S_1 \text{ and } S_2 \\
&\leq \mathcal{R}_m^{\mathcal{D}_1}(H) + \mathcal{R}_m^{\mathcal{D}_2}(H).
\end{aligned}$$

7. The second inequality, can be derived in the same way.

□

These bounds can be generalized to hold uniformly for any $\rho > 0$ at cost of an additional term $\sqrt{(\log\log_2(2/\rho))/m}$.

Corollary (Margin bounds for ranking with kernel-based hypotheses)

Let $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ be a PDS kernel with $r = \sup_{\mathbf{x}\in\mathcal{X}} K(\mathbf{x},\mathbf{x})$. Let also $\Phi : \mathcal{X} \mapsto \mathbb{H}$ be a feature mapping associated to K and let $H = \left\{\mathbf{x} \mapsto \langle \mathbf{w}, \Phi(\mathbf{x})\rangle \;\middle|\; \|\mathbf{w}\|_{\mathbb{H}} \leq \Lambda\right\}$ for some $\Lambda \geq 0$. Fix $\rho > 0$. Then, for any $\delta > 0$, the following pairwise margin bound holds with probability at least $(1-\delta)$ for any $h \in H$:

$$\mathbf{R}(h) \leq \hat{\mathbf{R}}_\rho(h) + 4\sqrt{\frac{r^2\Lambda^2/\rho^2}{m}} + \sqrt{\frac{\log(1/\delta)}{2m}}$$

- This bound can be generalized to hold uniformly for any $\rho > 0$ at cost of an additional term $\sqrt{(\log \log_2(2/\rho))/m}$.
- This bound suggests that a small generalization error can be achieved
  1. when $\frac{\rho}{r}$ is large (small second term),
  2. while the empirical margin loss is relatively small (first term).

- From the generalization bound for SVM, Corollary Margin bounds for ranking with kernel-based hypotheses can be expressed as

Corollary (Margin bounds for ranking with SVM)

Let $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ be a PDS kernel with $r = \sup_{\mathbf{x} \in \mathcal{X}} K(\mathbf{x}, \mathbf{x})$. Let also $\Phi : \mathcal{X} \mapsto \mathbb{H}$ be a feature mapping associated to K and let $H = \left\{ \mathbf{x} \mapsto \langle \mathbf{w}, \Phi(\mathbf{x}) \rangle \mid \|\mathbf{w}\|_{\mathbb{H}} \leq \Lambda \right\}$ for some $\Lambda \geq 0$. Then, for any $\delta > 0$, the following pairwise margin bound holds with probability at least $(1 - \delta)$ for any $h \in H$:

$$\mathbf{R}(h) \leq \frac{1}{m} \sum_{i=1}^{m} \xi_i + 4\sqrt{\frac{r^2 \Lambda^2}{m}} + \sqrt{\frac{\log(1/\delta)}{2m}} \tag{1}$$

where $\xi = \max\left(1 - y_i \left[\Phi(\mathbf{x}_i') - \Phi(\mathbf{x}_i)\right], 0\right)$

- Minimizing the right-hand side of inequality (1) is minimizing an objective function with a term corresponding to the sum of the slack variables ξ_i, and another one minimizing $\|\mathbf{w}\|$ or equivalently $\|\mathbf{w}\|^2$.
- This optimization problem can thus be formulated as

$$\min_{\mathbf{w}, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \xi_i$$

$$\text{subject to } y_i \left[\left\langle \mathbf{w}, \left(\Phi(\mathbf{x}_i') - \Phi(\mathbf{x}_i) \right) \right\rangle \right] \geq 1 - \xi_i$$

$$\xi_i \geq 0 \quad \forall 1 \leq i \leq m.$$

- This coincides exactly with the primal optimization problem of SVMs, with a feature mapping

$$\Psi : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{H}$$

defined by

$$\Psi(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x}) - \Phi(\mathbf{x}')$$

for all

$$(\mathbf{x}, \mathbf{x}') \in \mathcal{X} \times \mathcal{X}$$

and with a hypothesis set of functions of the form

$$(\mathbf{x}, \mathbf{x}') \mapsto \langle \mathbf{w}, \Psi(\mathbf{x}, \mathbf{x}') \rangle .$$

- Clearly, all the properties already presented for SVMs apply in this instance.
- In particular, the algorithm can benefit from the use of PDS kernels.
- This can be used with kernels

$$K'((\mathbf{x}_i, \mathbf{x}_i'), (\mathbf{x}_j, \mathbf{x}_j')) = \langle \Psi(\mathbf{x}_i, \mathbf{x}_i'), \Psi(\mathbf{x}_j, \mathbf{x}_j') \rangle = K(\mathbf{x}_i, \mathbf{x}_j) + K(\mathbf{x}_i', \mathbf{x}_j') - K(\mathbf{x}_i', \mathbf{x}_j) - K(\mathbf{x}_i, \mathbf{x}_j').$$

Boosting for ranking

- Use weak ranking algorithm and create stronger ranking algorithm:
- Ensemble method: combine base rankers returned by weak ranking algorithm
- Finding simple relatively accurate base rankers often not hard.
- How should base rankers be combined?
- Let H defined as
$$H = \{h : \mathcal{X} \mapsto \{0, 1\}\}$$
where H is the hypothesis set from which the base rankers are selected.
- For any $s \in \{-1, 0, +1\}$, we define
$$\epsilon_t^s = \sum_{i=1}^{m} D_t(i) \, \mathbb{I}\left[y_i(h_t(\mathbf{x}_i') - h_t(\mathbf{x}_i)) = s\right] = \mathop{\mathbb{E}}_{i \sim D_t}\left[\mathbb{I}\left[y_i(h_t(\mathbf{x}_i') - h_t(\mathbf{x}_i)) = s\right]\right]$$
- Hence, we have
$$\epsilon_t^+ + \epsilon_t^- + \epsilon_t^0 = 1$$
- We assume that $y_i \neq 0$.
- **Homework:** Show that the derivation of the algorithm.

Boosting for ranking

RankBoost Algorithm

function RankBoost(S, H, T)
  for $i \leftarrow 1$ to m **do**
    $D_1(i) \leftarrow \frac{1}{m}$
  end for
  for $t \leftarrow 1$ to T **do**
    Let $h_t = \underset{h \in H}{\text{argmin}} \left(\epsilon_t^- - \epsilon_t^+\right) \triangleq \underset{h \in H}{\text{argmin}} \left\{-\mathbb{E}_{i \sim D_t}\left[y_i \left(h(\mathbf{x}_i') - h(\mathbf{x}_i)\right)\right]\right\}$
    $\alpha_t \leftarrow \frac{1}{2} \log \frac{\epsilon_t^+}{\epsilon_t^-}$
    $Z_t \leftarrow \epsilon_t^0 + 2\sqrt{\epsilon_t^+ \epsilon_t^-}$
    for $i \leftarrow 1$ to m **do**
      $D_{t+1}(i) \leftarrow \frac{D_t(i) \exp\left[-\alpha_t y_i \left(h_t(\mathbf{x}_i') - h_t(\mathbf{x}_i)\right)\right]}{Z_t}$
    end for
  end for
  return $f \triangleq \sum_{t=1}^{T} \alpha_t h_t$
end function

Theorem (Bound on the empirical error of RankBoost)

The empirical error of the hypothesis $H = \{h : \mathcal{X} \mapsto \{0, 1\}\}$ *returned by RankBoost verifies:*

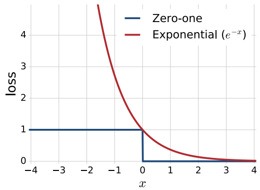
$$\hat{\mathbf{R}}(h) \leq \exp\left[-2\sum_{t=1}^{T}\left(\frac{\epsilon_t^+ - \epsilon_t^-}{2}\right)^2\right]$$

Furthermore, if there exists γ *such that for all* $1 \leq t \leq T$*, condition* $0 \leq \gamma \leq \frac{\epsilon_t^+ - \epsilon_t^-}{2}$*, then*

$$\hat{\mathbf{R}}(h) \leq \exp\left[-2\gamma^2 T\right].$$

Proof of (Bound on the empirical error of RankBoost).

1. The empirical error equals to $\hat{\mathbf{R}}(h) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}\left[y_i(f(\mathbf{x}_i') - f(\mathbf{x}_i)) \leq 0\right]$.
2. On the other hand, for all $u \in \mathbb{R}$, we have $\mathbb{I}\left[u \leq 0\right] \leq \exp(-u)$.



Proof of (Bound on the empirical error of RankBoost) (cont.).

3. Hence, we can write

$$\begin{aligned}\hat{\mathbf{R}}(h) &= \frac{1}{m}\sum_{i=1}^{m}\mathbb{I}\left[y_i(f(\mathbf{x}_i') - f(\mathbf{x}_i)) \leq 0\right] \\ &\leq \frac{1}{m}\sum_{i=1}^{m}\exp\left[-y_i(f(\mathbf{x}_i') - f(\mathbf{x}_i))\right] \\ &\leq \frac{1}{m}\sum_{i=1}^{m}\left[m\prod_{t=1}^{T}Z_t\right]D_{t+1}(i) = \prod_{t=1}^{T}Z_t.\end{aligned}$$

4. From definition of

$$Z_t = \sum_{i=1}^{m} D_t(i) exp\left[-y_i(h_t(\mathbf{x}_i') - h_t(\mathbf{x}_i))\right]$$

,

5. By grouping together the indices i for which $y_i(h_t(\mathbf{x}_i') - h_t(\mathbf{x}_i))$ take values in -1, 0, or $+1$, Z_t can be written as

$$\begin{aligned}Z_t &= \epsilon_t^{+}e^{-\alpha_t} + \epsilon_t^{-}e^{+\alpha_t} + \epsilon_t^{0} \\ &= \epsilon_t^{+}\sqrt{\frac{\epsilon_t^{-}}{\epsilon_t^{+}}} + \epsilon_t^{-}\sqrt{\frac{\epsilon_t^{+}}{\epsilon_t^{-}}} + \epsilon_t^{0} \\ &= 2\sqrt{\epsilon_t^{+}\epsilon_t^{-}} + \epsilon_t^{0}\end{aligned}$$

Proof of (Bound on the empirical error of RankBoost) (cont.).

6. Since, $\epsilon_t^+ = 1 - \epsilon_t^- - \epsilon_t^0$, we have

$$4\epsilon_t^+\epsilon_t^- = \left(\epsilon_t^+ + \epsilon_t^-\right)^2 - \left(\epsilon_t^+ - \epsilon_t^-\right)^2 = \left(1 - \epsilon_t^0\right)^2 - \left(\epsilon_t^+ - \epsilon_t^-\right)^2$$

7. Thus, assuming that $\epsilon_t^0 < 1$, Z_t can be upper bounded as

$$Z_t = \sqrt{(1-\epsilon_t^0)^2 - \left(\epsilon_t^+ - \epsilon_t^-\right)^2} + \epsilon_t^0 = \left(1 - \epsilon_t^0\right)\sqrt{1 - \frac{\left(\epsilon_t^+ - \epsilon_t^-\right)^2}{(1-\epsilon_t^0)^2}} + \epsilon_t^0$$

$$\leq \left(1 - \epsilon_t^0\right)\exp\left(-\frac{\left(\epsilon_t^+ - \epsilon_t^-\right)^2}{2\,(1-\epsilon_t^0)^2}\right) + \epsilon_t^0$$

By using inequality $1 - x \leq e^{-x}$

$$\leq \exp\left(-\frac{\left(\epsilon_t^+ - \epsilon_t^-\right)^2}{2}\right)$$

exp is concave and $0 < \left(1 - \epsilon_t^0\right) \leq 1$

$$\leq \exp\left(-2\left[\frac{\left(\epsilon_t^+ - \epsilon_t^-\right)}{2}\right]^2\right)$$

□

- Assume that the pairwise labels are in $-1, +1$.
- We showed that $\hat{\mathcal{R}}_S(conv(H)) = \hat{\mathcal{R}}_S(H)$.

Corollary (Margin bound for ensemble methods in ranking)

Let H be a set of real-valued functions. Fix $\rho > 0$; then, for any $\delta > 0$, with probability at least $(1-\delta)$ over the choice of a sample S of size m, each of the following ranking guarantees holds for all $h \in conv(H)$

$$\mathbf{R}(h) \leq \hat{\mathbf{R}}_\rho(h) + \frac{2}{\rho}\left(\mathcal{R}_m^{\mathcal{D}_1}(H) + \mathcal{R}_m^{\mathcal{D}_2}(H)\right) + \sqrt{\frac{\log(1/\delta)}{2m}}$$

$$\mathbf{R}(h) \leq \hat{\mathbf{R}}_\rho(h) + \frac{2}{\rho}\left(\hat{\mathcal{R}}_{S_1}(H) + \hat{\mathcal{R}}_{S_2}(H)\right) + 3\sqrt{\frac{\log(2/\delta)}{2m}}$$

- For RankBoost, these bounds apply to $f/\left\|\alpha\right\|_1$, where f and $f/\left\|\alpha\right\|_1$ induce the same ordering of the points.
- Then, or any $\delta > 0$, the following holds with probability at least $(1-\delta)$

$$\mathbf{R}(f) \leq \hat{\mathbf{R}}_\rho(f/\left\|\alpha\right\|_1) + \frac{2}{\rho}\left(\mathcal{R}_m^{\mathcal{D}_1}(H) + \mathcal{R}_m^{\mathcal{D}_2}(H)\right) + \sqrt{\frac{\log(1/\delta)}{2m}}$$

- Note that T does not appear in this bound.

Bipartite ranking

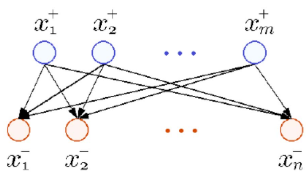
- Bipartite ranking problem is an important ranking scenario within the score-based setting.
- In this scenario, the set of points $\mathcal{X}$ is partitioned into
  1. the class of positive points $\mathcal{X}_+$
  2. the class of negative points $\mathcal{X}_-$
- In this setting, positive points must rank higher than negative ones and the learner receives
  1. a sample $S_+ = (\mathbf{x}'_1, \ldots, \mathbf{x}'_m)$ drawn i.i.d. according to some distribution $\mathcal{D}_+$ over $\mathcal{X}_+$ and
  2. a sample $S_- = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ drawn i.i.d. according to some distribution $\mathcal{D}_-$ over $\mathcal{X}_-$.
- The learning problem consists of selecting a hypothesis $h \in H$ with small expected bipartite misranking or generalization error $\mathbf{R}(h)$:

$$\mathbf{R}(h) = \underset{\substack{\mathbf{x} \sim \mathcal{D}_+ \\ \mathbf{x}' \sim \mathcal{D}_-}}{\mathbb{P}} \left[h(\mathbf{x}') < h(\mathbf{x}) \right]$$

- The empirical pairwise misranking or empirical error of h is

$$\hat{\mathbf{R}}_{S_+, S_-}(h) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbb{I}\left[h(\mathbf{x}'_i) < h(\mathbf{x}_j) \right]$$

- The learning algorithm must typically deal with mn pairs.



Shivani Agarwal

- A key property of RankBoost leading to an efficient algorithm for bipartite ranking is exponential form of its objective function.
- The objective function can be decomposed into the product of two functions,
  1. one depends on only the positive points.
  2. one depends on only the negative points.
- Similarly,

$$D_1(i,j) = \frac{1}{mn} = D_1^+(i)D_1^-(j) = \frac{1}{m} \times \frac{1}{n}$$

- Similarly,

$$D_{t+1}(i,j) = \frac{D_t(i,j)\exp\left(-\alpha_t\left[h_t(\mathbf{x}_i') - h_t(\mathbf{x}_j)\right]\right)}{Z_t} = \frac{D_t^+(i)\exp\left(-\alpha_t h_t(\mathbf{x}_i')\right)}{Z_t^+} = \frac{D_t^-(j)\exp\left(\alpha_t h_t(\mathbf{x}_j)\right)}{Z_t^-}$$

- The pairwise misranking of a hypothesis h

$$\left(\epsilon_t^- - \epsilon_t^+\right) = \mathop{\mathbb{E}}_{(i,j)\sim D_t}\left[h(\mathbf{x}_i') - h(\mathbf{x}_j)\right] \times \mathop{\mathbb{E}}_{j\sim D_t^-}\left[h(\mathbf{x}_j)\right] - \mathop{\mathbb{E}}_{i\sim D_t^+}\left[h(\mathbf{x}_i')\right]$$

- The time and space complexity of BipartiteRankBoost is $O(m+n)$.

Boosting for bipartite ranking

BipartiteRankBoost Algorithm

function BipartiteRankBoost(S, H, T)

  $D_1^+(i) \leftarrow \frac{1}{m} \quad \forall i \in 1, 2, \ldots, m$

  $D_1^-(j) \leftarrow \frac{1}{n} \quad \forall j \in 1, 2, \ldots, n$

  for $t \leftarrow 1$ to T **do**

    Let $h_t = \operatorname*{argmin}_{h \in H} \left(\epsilon_t^- - \epsilon_t^+\right) \triangleq \operatorname*{argmin}_{h \in H} \left\{\mathbb{E}_{j \sim D_t^-}[h(\mathbf{x}_j)] - \mathbb{E}_{i \sim D_t^+}[h(\mathbf{x}'_i)]\right\}$

    $\alpha_t \leftarrow \frac{1}{2} \log \frac{\epsilon_t^+}{\epsilon_t^-}$

    $Z_t^+ \leftarrow 1 - \epsilon_t^+ + \sqrt{\epsilon_t^+ \epsilon_t^-}$

    for $i \leftarrow 1$ to m **do**

      $D_{t+1}^+(i) \leftarrow \frac{D_t^+(i) \exp\left[-\alpha_t h_t(\mathbf{x}'_i)\right]}{Z_t^+}$

    end for

    $Z_t^- \leftarrow 1 - \epsilon_t^- + \sqrt{\epsilon_t^+ \epsilon_t^-}$

    for $j \leftarrow 1$ to n **do**

      $D_{t+1}^-(j) \leftarrow \frac{D_t^-(j) \exp\left[\alpha_t h_t(\mathbf{x}_j)\right]}{Z_t^-}$

    end for

  end for

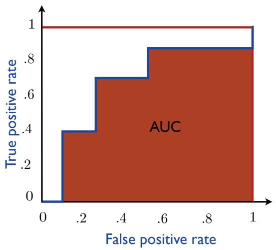
  return $f \triangleq \sum_{t=1}^{T} \alpha_t h_t$

end function

- The performance of a bipartite ranking algorithm is reported in terms of area ROC curve, or AUC.
- Let U be a test sample used for evaluating the performance of h

1. m positive points $\mathbf{z}'_1, \ldots, \mathbf{z}'_m$
2. n negative points $\mathbf{z}_1, \ldots, \mathbf{z}_n$
3. $AUC(h, u)$ equals to

$$AUC(h, U) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbb{I}\left[h(\mathbf{z}'_i) \geq h(\mathbf{z}_j)\right] = \underset{\substack{\mathbf{z} \sim D_U^- \\ \mathbf{z}' \sim D_U^+}}{\mathbb{P}} \left[h(\mathbf{z}') \geq h(\mathbf{z})\right]$$



- The average pairwise misranking of h over U denoted by $\hat{\mathbf{R}}(h, U)$

$$\hat{\mathbf{R}}(h, U) = 1 - AUC(h, U).$$

- AUC can be computed in time of $O(m + n)$ from a sorted array $h(\mathbf{z}'_i)$ and $h(\mathbf{z}_j)$.

Preference-based setting

Preference-based setting

- Assume that you receive a list $X \subseteq \mathcal{X}$ as a result of a query q.
- The goal is to rank items in list X not all items in $\mathcal{X}$.
- The advantage of preference-based setting over score-based setting is:
  The learning algorithm is not required to return a linear ordering of all points of $\mathcal{X}$, which may be impossible.
- The preference-based setting consists of two stages.
  1. A sample of labeled pairs S is used to learn a **preference function** $h : \mathcal{X} \times \mathcal{X} \mapsto [0, 1]$ (exactly as in the score-based setting).
  2. The given list $X \subseteq \mathcal{X}$, the preference function h is used to determine a ranking of X.
- How can h be used to generate an accurate ranking?
- The computational complexity of the second stage is also crucial.
- We will measure the time complexity in terms of the number of calls to h.

- Assume that a preference function h is given.
- h is not assumed to be transitive.
- We assume that h is pairwise consistent, that is

$$h(u,v) + h(v,u) = 1, \qquad \forall u, v \in \mathcal{X}$$

- Let $\mathcal{D}$ be an unknown distribution according to which pairs (X, σ^*) are drawn, where
  1. $X \subseteq \mathcal{X}$ is a query subset.
  2. σ^* is a target ranking.
- The objective of a second-stage algorithm A is using function h to return an accurate ranking $A(X)$ for any query subset X.
- The algorithm A may be deterministic or randomized.
- The loss function ℓ is used to measure disagreement between target ranking σ^* and ranking σ for set X with $n \geq 1$ elements.

$$\ell(\sigma, \sigma^*) = \frac{2}{n(n-1)} \sum_{u \neq v} \mathbb{I}\left[\sigma(u) < \sigma(v)\right] \mathbb{I}\left[\sigma^*(v) < \sigma^*(u)\right]$$

- The loss between target ranking σ^* and ranking h equals to

$$\ell(h, \sigma^*) = \frac{2}{n(n-1)} \sum_{u \neq v} h(u,v)\, \mathbb{I}\left[\sigma^*(v) < \sigma^*(u)\right]$$

- The expected loss for a deterministic algorithm A is

$$\mathop{\mathbb{E}}_{(X,\sigma^*)\sim\mathcal{D}}\left[\ell(A(X),\sigma^*)\right].$$

- Regret of algorithm A is the difference between its loss and loss of the best fixed global ranking.

$$Regret(A) = \mathop{\mathbb{E}}_{(X,\sigma^*)\sim\mathcal{D}}\left[\ell(A(X),\sigma^*)\right] - \min_{\sigma'} \mathop{\mathbb{E}}_{(X,\sigma^*)\sim\mathcal{D}}\left[\ell(\sigma'_{|X},\sigma^*)\right]$$

- Regret of the preference function is

$$Regret(h) = \mathop{\mathbb{E}}_{(X,\sigma^*)\sim\mathcal{D}}\left[\ell(h_{|X},\sigma^*)\right] - \min_{h'} \mathop{\mathbb{E}}_{(X,\sigma^*)\sim\mathcal{D}}\left[\ell(h'_{|X},\sigma^*)\right]$$

- For sort by degree algorithm, we can prove

$$Regret(A) \leq 2Regret(h)$$

Theorem (Lower bound for deterministic algorithms)

For any deterministic algorithm A, there is a bipartite distribution for which

$$Regret(A) \geq 2Regret(h)$$

- For randomized quick sort(RQS), we can prove

$$Regret(A_{RQS}) \leq Regret(h)$$

- **Homework:** Prove these bounds and the above theorem.

Summary

- We defined ranking problem.
- We can extend this by using other loss functions defined in terms of a weight function.
- We can extend this by using other criteria have been introduced in information retrieval such as $NDCG$, $P@n$.

Readings

1. Sections 17.4 and 17.5 of Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning : From theory to algorithms*. Cambridge University Press, 2014.
2. Chapter 10 of Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. Second Edition. MIT Press, 2018.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. Second Edition. MIT Press, 2018.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning : From theory to algorithms*. Cambridge University Press, 2014.

Questions?