Predictors that Specialize

Yoav Freund

February 16, 2006

Outline

The specialists setup

bounding cumulative loss using relative entropy

Applications of specialists

The specialists setup

- Up till now we assumed that each expert makes a prediction at each iteration.
- Imagine that experts are specialists, they predict only some of the time.
- Gives the designer a lot of flexibility.
- Generalizes the switching experts setup.

The specialists game

On each iteration $t = 1, 2, 3, \ldots$

- Adversary chooses a set $E^t \subseteq \{1, \ldots, N\}$ of awake specialists.
- Adversary chooses predictions for specialists in E^t
- Algorithm chooses it's prediction.
- Adversary chooses outcome.
- Algorithm suffers loss. Specialists in E^t suffer loss. Sleeping specialists suffer no loss.

Desired bound

- Algorithm has to predict on each iteration
- Each specialist might sleep some of the time.
- $\Rightarrow$ makes no sense to compare to total loss of best specialist.
- **u**: a probability distributions, $u_i \geq 0, \;\; \sum_i u_i = 1$.
- Average loss w.r.t. **u**: $\ell_{\mathbf{u}}^t \doteq \frac{\sum_{i \in E^t} u_i \ell_i^t}{\sum_{i \in E^t} u_i}$
- Goal: $L_A \leq \min_{\mathbf{u}} \sum_{t=1}^{T} \ell_{\mathbf{u}}^t +$ something small

Applying Vovk-style algs to specialists

- We use normalized weights:

$$v_i^t = \frac{w_i^t}{\sum_{j=1}^N w_i^t}, \quad \mathbf{v}^t = \frac{\mathbf{w}^t}{W^t}$$

- Algorithm: treat the set E_t as the set of experts.
- Normalize the weights of specialists in E_t so that

$$\sum_{i \in E^t} v_i^t = \sum_{i \in E^t} v_i^{t+1}$$

- In particular: total weight is always 1.

Bound for log-loss case

- Bound for log loss (Theorem 1), for any distribution $\mathbf{u}$:
  $\sum_{t=1} u(E^t)\ell_A^t \leq \sum_{t=1}^T \sum_{i \in E^t} u_i \ell_i^t + \mathbf{RE}(\mathbf{u}||\mathbf{v}^1)$
- $\mathbf{RE}(\mathbf{u}||\mathbf{v}) \doteq \sum_i u_i \log \frac{u_i}{v_i}$
- $u(E^t) \doteq \sum_{i \in E^t} u_i$
- If we assume that $u(E^t) = U$ is constant, we get

$$L_A \leq \sum_{t=1}^{T} \ell_{\mathbf{u}}^t + \frac{\mathbf{RE}(\mathbf{u}||\mathbf{v}^1)}{U}$$

Cumulative loss vs. Final total weight

Total weight: $W^t \doteq \sum_{i=1}^N w_i^t$

$$\frac{W^{t+1}}{W^t} = \frac{\sum_{i=1}^N w_i^t e^{\log p_i^t(c^t)}}{\sum_{i=1}^N w_i^t} = \frac{\sum_{i=1}^N w_i^t p_i^t(c^t)}{\sum_{i=1}^N w_i^t} = p_A^t(c^t)$$

$$-\log \frac{W^{t+1}}{W^t} = -\log p_A^t(c^t)$$

$$-\log W^{T+1} = -\log \frac{W^{T+1}}{W^1} = -\sum_{t=1}^{T} \log p_A^t(c^t) = L_A^T$$

EQUALITY not bound!

Relative Entropy

- $\mathbf{u}, \mathbf{v}$: probability distributions, $u_i \geq 0, \ \sum_i u_i = 1$.
- $$\mathbf{RE}(\mathbf{u}||\mathbf{v}) \doteq \sum_i u_i \log \frac{u_i}{v_i}$$
- $\mathbf{RE}(\mathbf{u}||\mathbf{v}) \geq 0$, $\mathbf{RE}(\mathbf{u}||\mathbf{v}) = 0$ iff $\mathbf{u} = \mathbf{v}$
- $\exists \mathbf{u}, \mathbf{v}, \quad \mathbf{RE}(\mathbf{u}||\mathbf{v}) \neq \mathbf{RE}(\mathbf{v}||\mathbf{u})$
- $\exists \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \quad \mathbf{RE}(\mathbf{u}_1||\mathbf{u}_3) > \mathbf{RE}(\mathbf{u}_1||\mathbf{u}_2) + \mathbf{RE}(\mathbf{u}_2||\mathbf{u}_3)$

Normalized weights notation

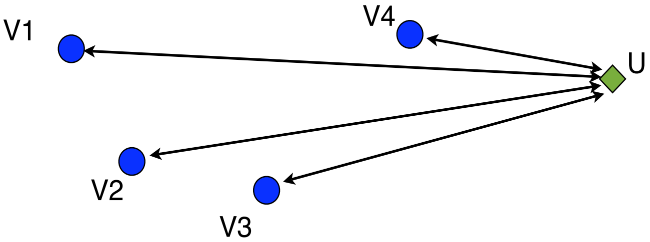
- p_i^t: distribution (of letters) predicted by expert i at time t
- Experts losses at time t:
  $\ell^t = \langle \ell_1^t, \ldots, \ell_N^t \rangle = -\langle \log p_1^t(c^t), \ldots, \log p_N^t(c^t) \rangle$
- Prediction of algorithm: $p_A^t = \sum_{i=1}^N v_i^t p_i^t$
- Loss of algorithm at time t: $\ell_A^t = -\log p_A^t(c^t)$

Bounding cumulative log loss using relative entropy

- Let $\mathbf{u}$ be an arbitrary distribution vector over experts.
- Lemma: $\mathbf{RE}(\mathbf{u}||\mathbf{v}^t) - \mathbf{RE}(\mathbf{u}||\mathbf{v}^{t+1}) = \ell_A^t - \mathbf{u} \cdot \ell^t$
- Summing over $t = 1, \ldots, T$ we get:
  $\mathbf{RE}(\mathbf{u}||\mathbf{v}^1) - \mathbf{RE}(\mathbf{u}||\mathbf{v}^{T+1}) = L_A - \mathbf{u} \cdot \sum_{t=1}^{T} \ell^t$
- $L_A \leq \min_{\mathbf{u}} \left(\mathbf{u} \cdot \sum_{t=1}^{T} \ell^t + \mathbf{RE}(\mathbf{u}||\mathbf{v}^1) \right)$
- For the special case $\mathbf{u} = \langle 0, \ldots, 0, 1, 0, \ldots, 0 \rangle$ and $\mathbf{v}^1 = \langle 1/N, \ldots, 1/N \rangle$ we get the old bound:
  $L_A \leq \min_i L_i + \log N$

Visual intuition

$$\mathbf{RE}(\mathbf{u}\|\mathbf{v}^t) - \mathbf{RE}(\mathbf{u}\|\mathbf{v}^{t+1}) = \ell^t_A - \mathbf{u} \cdot \ell^t$$



$\mathbf{v}^{t+1}$ is chosen to minimize $\mathbf{RE}(\mathbf{v}^{t+1}\|\mathbf{v}^t) + \mathbf{v}^{t+1} \cdot \ell^t$
Last line is confusing! I don't understand it!
But Manfred Warmuth does!

Proof of Lemma

- $\mathbf{RE}(\mathbf{u}||\mathbf{v}^t) - \mathbf{RE}(\mathbf{u}||\mathbf{v}^{t+1}) = \ell_A^t - \mathbf{u} \cdot \ell^t$
-

$$\begin{aligned}
&\mathbf{RE}(\mathbf{u}||\mathbf{v}^t) - \mathbf{RE}\left(\mathbf{u}||\mathbf{v}^{t+1}\right) \\
&= \sum_i u_i \log \frac{u_i}{v_i^t} - \sum_i u_i \log \frac{u_i}{v_i^{t+1}} = \sum_i u_i \log \frac{v_i^{t+1}}{v_i^t} \\
&= \sum_i u_i \log \left(\frac{W^t}{W^{t+1}} \frac{w_i^{t+1}}{w_i^t} \right) \\
&= \log \frac{W^t}{W^{t+1}} + \sum_i u_i \log e^{-\ell_i^t} = \ell_A^T - \sum_i u_i \ell_i^t
\end{aligned}$$

bounding general loss using relative entropy

- Suppose that loss is (a, c)-achievable.
- Achievable with Vovk algorithm, learning rate $\eta = \frac{a}{c}$
- Let $\mathbf{u}$ be an arbitrary distribution vector over experts.
- Lemma: $\mathbf{RE}(\mathbf{u}||\mathbf{v}^t) - \mathbf{RE}(\mathbf{u}||\mathbf{v}^{t+1}) \geq \frac{1}{c}\ell_A^t - \frac{a}{c}\mathbf{u} \cdot \ell^t$
- Summing over $t = 1, \ldots, T$ we get:
  $\mathbf{RE}(\mathbf{u}||\mathbf{v}^1) - \mathbf{RE}(\mathbf{u}||\mathbf{v}^{T+1}) = \frac{1}{c}L_A - \frac{a}{c}\mathbf{u} \cdot \sum_{t=1}^{T} \ell^t$
- $L_A \leq \min_{\mathbf{u}} \left(a\mathbf{u} \cdot \sum_{t=1}^{T} \ell^t + c\mathbf{RE}(\mathbf{u}||\mathbf{v}^1) \right)$
- For any mixable loss, $a = 1$, using
  $\mathbf{u} = \langle 0, \ldots, 0, 1, 0, \ldots, 0 \rangle$ and $\mathbf{v}^1 = \langle 1/N, \ldots, 1/N \rangle$ we get the old bound: $L_A \leq \min_i L_i + c \log N$

Example Application

- Consider the context algorithm.
- Let each node in the tree be a specialist.
- Gives an inferior algorithm (regret bound is twice as large)
- But much easier to generalize.

Generic Example

- Partition the input space. Assign each part to a specialist.
- Use several partitions, of different fineness.
- Can partition time in addition to space.
- Parts do not have to be disjoint.
- Partitions can adapt to data.
- Your idea here...