The Online Bayes algorithm

Yoav Freund

January 15, 2018

Outline

- Classical Bayesian Statistics
- Combining experts in the log loss framework
- Review: The online Bayes Algorithm
  - Comparison to **Hedge**(η)
- Review: The performance bound
  - Comparison to **Hedge**(η)

The Bayesian Generative Process

- Let Θ be a set of distrubutions over a space X.
  Example: a d dimensional Gaussian distribution over R^d.
  $\theta = (\vec{\mu}, \Sigma)$
- Let D be the **prior** distribution over Θ
- **Selecting Model:** $\theta \in \Theta$ is chosen according to the prior D
- **Generating Data:** $x_1, x_2, \ldots, x_n$ are generated IID according to θ

The Bayes optimal prediction

- The **Posterior distribution**: the conditional probability of the model θ given the data $x_1, x_2, \ldots, x_n$.

$$P(\theta|x_1, x_2, \ldots, x_n) = \frac{1}{Z} D(\theta) \prod_{i=1}^{n} P(x_i|\theta)$$

- **Posterior average:** predict the distribution of a new example x_{n+1} with the conditional probability:

$$P(x_{n+1}|x_1, x_2, \ldots, x_n) = \sum_{\theta \in \Theta} P(x_{n+1}|\theta) P(\theta|x_1, x_2, \ldots, x_n)$$

In what sense is the posterior average optimal?

- It is the optimal perdiction if the data is generated according to the Bayesian generative process.
- What if the data is not generated by any of the models?
- Classical analysis cannot be used.
- **We will show** a tight bound on the regret!.

The log-loss framework

- Algorithm A predicts a sequence $c^1, c^2, \ldots, c^T$ over alphabet $\Sigma = \{1, 2, \ldots, k\}$
- The prediction for the c^tth is a distribution over Σ: $\mathbf{p}_A^t = \langle p_A^t(1), p_A^t(2), \ldots, p_A^t(k)\rangle$
- When c^t is revealed, the loss we suffer is $-\log p_A^t(c^t)$
- The cumulative log loss, which we wish to minimize, is ${L_A}^T = -\sum_{t=1}^T \log p_A^t(c^t)$
- $\lceil {L_A}^T \rceil$ is the code length if A is combined with arithmetic coding.

The game

- Prediction algorithm A has access to N experts.
- The following is repeated for $t = 1, \ldots, T$
  - Experts generate predictive distributions: $\mathbf{p}_1^t, \ldots, \mathbf{p}_N^t$
  - Algorithm generates its own prediction $\mathbf{p}_A^t$
  - c^t is revealed.
- **Goal:** minimize regret:

$$-\sum_{t=1}^{T} \log p_A^t(c^t) + \min_{i=1,\ldots,N} \left(-\sum_{t=1}^{T} \log p_i^t(c^t) \right)$$

The online Bayes Algorithm

- Total loss of expert i

$$L_i^t = -\sum_{s=1}^{t} \log p_i^s(c^s); \quad L_i^0 = 0$$

- Weight of expert i

$$w_i^t = w_i^1 e^{-L_i^{t-1}} = w_i^1 \prod_{s=1}^{t-1} p_i^s(c^s)$$

- Freedom to choose initial weights.
  $w_t^1 \geq 0$, $\sum_{i=1}^{n} w_i^1 = 1$
- Prediction of algorithm A

$$\mathbf{p}_A^t = \frac{\sum_{i=1}^{N} w_i^t \mathbf{p}_i^t}{\sum_{i=1}^{N} w_i^t}$$

The **Hedge**(η)Algorithm

Consider action i at time t

- Total loss:

$$L_i^t = \sum_{s=1}^{t-1} \ell_i^s$$

- Weight:

$$w_i^t = w_i^1 e^{-\eta L_i^t}$$

Note freedom to choose initial weight (w_i^1) $\sum_{i=1}^{n} w_i^1 = 1$.

- $\eta > 0$ is the learning rate parameter. Halving: $\eta \to \infty$
- Probability:

$$p_i^t = \frac{w_i^t}{\sum_{j=1}^{N} w_i^t}, \quad \mathbf{p}^t = \frac{\mathbf{w}^t}{\sum_{j=1}^{N} w_i^t}$$

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!

Simple Bound

- Use uniform initial weights $w_i^1 = 1/N$
- Total Weight is at least the weight of the best expert.

$$
\begin{aligned}
L_A^T &= -\log W^{T+1} = -\log \sum_{i=1}^{N} w_i^{T+1} \\
&= -\log \sum_{i=1}^{N} \frac{1}{N} e^{-L_i^T} = \log N - \log \sum_{i=1}^{N} e^{-L_i^T} \\
&\leq \log N - \log \max_i e^{-L_i^T} = \log N + \min_i L_i^T
\end{aligned}
$$

- Dividing by T we get $\frac{L_A^T}{T} = \min_i \frac{L_i^T}{T} + \frac{\log N}{T}$

Regret bound for **Hedge**(η)

- Tuning η as a function of T (uniform prior).
- trivially $\min_i L_i \leq T$, yielding

$$L_{\mathbf{Hedge}(\eta)} \leq \min_i L_i + \sqrt{2T \ln N} + \ln N$$

- per iteration we get:

$$\frac{L_{\mathbf{Hedge}(\eta)}}{T} \leq \min_i \frac{L_i}{T} + \sqrt{\frac{2 \ln N}{T}} + \frac{\ln N}{T}$$

- Compare to regret bund for Bayes Algorithm: $\frac{L_A^T}{T} = \min_i \frac{L_i^T}{T} + \frac{\log N}{T}$