COS 511: Foundations of Machine Learning

Rob Schapire
Scribe: Umar Syed

Lecture #21
April 27, 2006

1 Recap Of Last Time

Last time, we covered Bayes' Algorithm for online learning of probability distributions, and its analysis. Recall the setting: at each time t, we must predict a distribution for variable x_t. We are given N experts, where expert i predicts distribution $p_{t,i}(x)$ at time t. Our goal is to constuct a "master" distribution $q_t(x)$ that has log-loss over the training examples (in hindsight) not much worse than the log-loss of the best expert distibution.

Here is Bayes' algorithm.

1. Initialize $w_{1,i} = \pi_i$, where $\pi_i \geq 0$ and $\sum_i \pi_i = 1$.
2. For $t = 1, \ldots, T$
3. Expert i predicts distribution $p_{t,i}(x)$.
4. Master predicts $q_t(x) = \sum_i w_{t,i} p_{t,i}(x)$.
5. Observe x_t.
6. Update $w_{t+1,i}$ as follows:

$$w_{t+1,i} \leftarrow \frac{w_{t,i} p_{t,i}(x_t)}{Z_{t+1}}$$

where Z_{t+1} is a normalizing constant so that $\sum_i w_{t+1,i} = 1$.

We showed that

$$\underbrace{-\sum_{t=1}^{T} \log q_t(x_t)}_{(1)} \leq \min_i \left[\underbrace{-\sum_{t=1}^{T} \log p_{t,i}(x_t)}_{(2)} - \log \pi_i \right].$$

Note that (1) is the log-loss of the master, and (2) is the log-loss of expert i.

Recall that we "pretended" the x_t's were generated by the following probabalistic process:

1. Choose expert i^* according to $\Pr[i^* = i] \triangleq \pi_i$.
2. Choose x_t according to $\Pr\left[x \mid x_1^{t-1}, i^* = i\right] \triangleq p_{t,i}(x)$.[1]

Then we defined $q_t(x) \triangleq \Pr\left[x \mid x_1^{t-1}\right]$, and used Bayes' Rule to derive the above algorithm and bound. Importantly, the analysis of the algorithm did *not* require that a probabilistic process actually generated the data; it applies even when the x_t's are chosen arbitrarily (even adversarially).

[1] We write x_1^{t-1} to denote the sequence $x_1, \ldots, x_{t-1}$

2 Switching Experts

Now suppose that different experts do better on different parts of the sequence of x_t's. For example, expert 3 may be the best for the first 277 steps, followed by expert 7 for the next 173 steps, and so on. It is natural to ask whether we can construct a q_t that predicts almost as well as the best *sequence* of experts, i.e. the best sequence $e_1 \cdots e_T$, where expert e_t predicts the distribution at time t.

How often should the 'current' expert be allowed to change during the sequence? If we allow a change at every time step, then it's possible for only two experts — one that always predicts 0 with probability 1, and another that always predicts 1 with probability 1 — to predict any sequence of x_t's with no log-loss (this is assuming the x_t's are binary variables). More generally, if the x_t's come from a set $\mathcal{X}$, and if we allow a change every step, then an arbitrary switching sequence over $|\mathcal{X}|$ experts can predict any sequence with no log-loss. We cannot hope to bound the log-loss of q_t in this case.

So we will allow only k switches during the sequence, where k remains fixed even as T grows. The approach is to define a 'meta-expert' for each possible switching sequence of experts. If there are N experts and T time steps, then a simple combinatorial argument shows that there are $M = \binom{T}{k} N^{k+1}$ possible switching sequences.

Now we can just apply Bayes algorithm and its analysis. If we use a uniform prior over the meta-experts (i.e. $\pi_i = \frac{1}{M}$ for all i), then we have

$$-\sum_{t=1}^{T} \log q_t(x_t) \leq \min_i \left[-\sum_{t=1}^{T} \log p_{t,i}(x_t) + \log M \right],$$

where $\log M \approx (k+1)\log N + k \log \frac{T}{k}$. This is a reasonable bound; the trouble is that the algorithm now requires time exponential in k to compute q_t. A better approach is needed.

3 Switching Experts (Reprise)

The new approach will be to define a prior over the meta-experts that allows for efficient computation of q_t. Let $\mathbf{e} = e_1 \cdots e_T$ denote a meta-expert, where e_t is the expert predicting at time t. Note that we allow all possible switching sequences. The prior $\Pr[\mathbf{e}^* = \mathbf{e}] \triangleq \pi(\mathbf{e})$ is defined as

$$\Pr[e_1^* = i] = \tfrac{1}{N}$$
$$\Pr\left[e_{t+1}^* = i \mid e_t^* = j\right] = \begin{cases} 1-\alpha & \text{if } i = j \\ \frac{\alpha}{N-1} & \text{otherwise} \end{cases}$$

Note that the probability of choosing an expert at time t depends only on the expert chosen at time $t-1$. In other words, the sequence of experts is a *Markov chain*.

Again, we assume that a fictitious probabilistic process generates the data:

1. Choose meta-expert $\mathbf{e}^*$ according to $\Pr[\mathbf{e}^* = \mathbf{e}] \triangleq \pi(\mathbf{e})$ (defined above).
2. Choose x_t according to $\Pr\left[x \mid x_1^{t-1}, e_t^* = i\right] \triangleq p_{t,i}(x)$.

As before, to derive Bayes' algorithm, we need a way to compute $q_t(x) \triangleq \Pr\left[x \mid x_1^{t-1}\right]$. But before doing so, we can directly apply the bound:

$$-\sum_{t=1}^{T} \log q_t(x_t) \leq \min_i \left[-\sum_{t=1}^{T} \log p_{t,i}(x_t) - \log \pi(\mathbf{e}) \right].$$

It's easy to see from its definition that

$$-\log \pi(\mathbf{e}) = -\log \left[\frac{1}{N} \left(\frac{\alpha}{N-1} \right)^k (1-\alpha)^{T-k-1} \right],$$

where k is the number of times the expert switches in $\mathbf{e}$. Minimizing this over α yields $\alpha = \frac{k}{T-1}$, which gives us

$$-\log \pi(\mathbf{e}) = \log N + k \log \left[\frac{(N-1)(T-1)}{k} \right] \underbrace{-(T-k-1) \log \left(1 - \frac{k}{T-1} \right)}_{(1)}.$$

The quantity labeled (1) is less than k, and so this bound is similar to the one for the first meta-experts Bayes algorithm (see previous section).

How do we compute $q_t(x)$? Here's how.

$$\begin{aligned} q_t(x) &\triangleq \Pr\left[x_t \mid x_1^{t-1}\right] \\ &= \sum_i \Pr\left[e_t^* = i \mid x_1^{t-1}\right] \Pr\left[x_t \mid x_1^{t-1}, e_t^* = i\right] \\ &= \sum_i \underbrace{\Pr\left[e_t^* = i \mid x_1^{t-1}\right]}_{w_{t,i}} p_{t,i}(x) \end{aligned}$$

So we just need a way to compute $w_{t,i}$. We know that $w_{1,i} = \frac{1}{N}$. Also

$$\begin{aligned} w_{t+1,i} &= \Pr\left[e_{t+1}^* = i \mid x_1^t\right] \\ &\overset{(1)}{=} \sum_j \Pr\left[e_t^* = j \mid x_1^t\right] \Pr\left[e_{t+1}^* = i \mid e_t^* = j, x_1^t\right] \\ &\overset{(2)}{=} \sum_j \Pr\left[e_t^* = j \mid x_1^t\right] \Pr\left[e_{t+1}^* = i \mid e_t^* = j\right] \\ &= \sum_j \Pr\left[e_t^* = j \mid x_1^t\right] \cdot \left\{ \begin{array}{ll} 1-\alpha & \text{if } i = j \\ \frac{\alpha}{N-1} & \text{otherwise} \end{array} \right\} \\ &= \sum_j \Pr\left[e_t^* = j \mid x_1^{t-1}, x^t\right] \cdot \left\{ \begin{array}{ll} 1-\alpha & \text{if } i = j \\ \frac{\alpha}{N-1} & \text{otherwise} \end{array} \right\} \\ &\overset{(3)}{=} \sum_j \frac{\Pr\left[x^t \mid x_1^{t-1}, e_t^* = j\right] \Pr\left[e_t^* = j \mid x_1^{t-1}\right]}{\Pr\left[x_t \mid x_1^{t-1}\right]} \cdot \left\{ \begin{array}{ll} 1-\alpha & \text{if } i = j \\ \frac{\alpha}{N-1} & \text{otherwise} \end{array} \right\} \\ &\overset{(4)}{=} \sum_j \underbrace{\frac{p_{t,j}(x_t) w_{t,j}}{q_t(x_t)}}_{C_j} \cdot \left\{ \begin{array}{ll} 1-\alpha & \text{if } i = j \\ \frac{\alpha}{N-1} & \text{otherwise} \end{array} \right\} \end{aligned}$$

Equality (1) holds by the Law of Total Probability. Equality (2) holds by the Markov property of the sequence of experts. Equality (3) is from Bayes' Rule, and equality (4) is just the application of several definitions.

By naively applying the last equality above, we can compute all the weights in $O(N^2)$ time. Actually, we can do better than this, since the last equality can be re-written

$$w_{t+1,i} = \underbrace{\frac{\alpha}{N-1} \Big(\sum_j C_j \Big)}_{(1)} + \Big(1 - \alpha - \frac{\alpha}{N-1} \Big) C_i,$$

where the quantity labeled (1) is independent of i. So we can actually compute all weights in only $O(N)$ time. (Moreover, it was pointed out after class that $\sum_j C_j = 1$, making the calculation even simpler.)

4 A Technical Comment For Next Time

So far, for Bayes' algorithm, we've assumed that all the expert distributions sum to 1 (i.e. they are probability distributions), which has meant that the 'master' distribution also sums to 1. In other words,

$$\forall t, i \ \sum_x p_{t,i}(x) = \sum_x q_t(x) = 1.$$

However, if all the $p_{t,i}$ distributions summed to some constant C instead, so would q_t. Moreover, Bayes' algorithm would not change at all (the C's would cancel in the derivation), nor would the bound from the analysis (a $\log C$ term would cancel from both sides of the bound). This fact will be useful for the next lecture.