COS 511: Theoretical Machine Learning

Lecturer: Rob Schapire
Scribe: Allison Chaney

Lecture # 20
April 18, 2013

Recall that in the previous lecture, our task was modeling or estimating a probability distribution. We are given a large (but not "gigantic") space $\mathbf{X}$ where $|\mathbf{X}| = N$. We also have points $x_1, \ldots, x_m \in \mathbf{X}$ where $x_i \sim \mathcal{D}$ and $\mathcal{D}$ is an unknown distribution. Further, we have features $f_1, \ldots, f_n$ where $f_j : \mathbf{X} \to \mathbb{R}$; in other words, each feature is a real-valued function. Our goal is to estimate the distribution $\mathcal{D}$.

We have already discussed two possible approaches; the first is using maximum entropy, or $q^* = \arg\max_{q \in P} H(q)$, where

$$P = \{q \mid \forall j : \mathbf{E}_q[f_j] = \hat{\mathbf{E}}[f_j]\}$$

$$\mathbf{E}_q[f] = \mathbf{E}_{x \sim q}[f(x)]$$

$$\hat{\mathbf{E}}[f] = \frac{1}{m} \sum_{i=1}^{m} f(x).$$

The second is by picking a parametric form, or $q^* = \arg\max_{p \in \overline{Q}} \sum_i \ln q(x_i)$ where

$$Q = \left\{ q : q \text{ has form } q(x) = \frac{\exp\left(\sum_j \lambda_j f_j(x)\right)}{Z_\lambda} \right\}.$$

Theorem 1. The following are equivalent:

$$\begin{array}{lll} (1) & q^* & = \arg\max_{q \in P} H(q) \\ (2) & q^* & = \arg\max_{q \in \overline{Q}} \sum_i \ln q(x_i) \\ (3) & q^* & \in P \cap \overline{Q} \end{array}$$

Any one of these uniquely defines q^*. In today's lecture, we want to show how to numerically find solution to this by deriving an algorithm and proving it converges to the solution.

Expression (2) in theorem 1 is the easiest one to manage, so we want to find the λ_j's to minimize the objective loss function

$$L(\boldsymbol{\lambda}) = -\frac{1}{m} \sum_{i=1}^{m} \ln q_{\boldsymbol{\lambda}}(x_i)$$

where

$$q_{\boldsymbol{\lambda}}(x) = \frac{\exp\left(\sum_j \lambda_j f_j(x)\right)}{Z_{\boldsymbol{\lambda}}}.$$

For shorthand, let's call $g_{\boldsymbol{\lambda}}(x) = \sum_j \lambda_j f_j(x)$.

Now our approach can be that we want to find a new $\boldsymbol{\lambda}_t$ with each iteration of the algorithm such that the loss converges as follows:

$$L(\boldsymbol{\lambda}_t) \to \inf_{\boldsymbol{\lambda}} L(\boldsymbol{\lambda})$$

The basic shell of this algorithm would be:

choose $\boldsymbol{\lambda}_1$ (e.g., $\mathbf{0}$)
for $t = 1, 2 \ldots$ **do**
  compute $\boldsymbol{\lambda}_{t+1}$ from $\boldsymbol{\lambda}_t$
end for

We also want to assume that for all x:

$$\forall j : f_j(x) \geq 0 \quad \text{and} \quad \sum_{j=1}^{n} f_j(x) = 1.$$

Thus for any x, the features form a probability distribution.

Preliminary 1. We can assume that for all x and for all j: $f_j(x) \geq 0$ and $\sum_{j=1}^{n} f_j(x) = 1$ without loss of generality.

We know that each feature f_j maps $\mathbf{X} \to \mathbb{R}$; this can be rescaled to $f_j : \mathbf{X} \to [0, 1]$, because the domain $\mathbf{X}$ is finite. Then, f_j can be replaced by f_j/n. Now,

$$\forall x : \sum_{j=1}^{n} f_j(x) \leq 1.$$

Finally, we can create a new feature:

$$f_0(x) := 1 - \sum_{j=1}^{n} f_j(x)$$

such that now $\forall x$: $\sum_{j=0}^{n} f_j(x) = 1$. In addition, we have not changed the space of functions we are working with since a linear combination involving f_0 will still be linear in the original features $f_1, \ldots, f_n$.

Returning to our loss function $L(\boldsymbol{\lambda})$, we can reframe the problem of minimizing the change in loss between iterations, which is equivalent. Using the shorthand of $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}$ and $\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}'$, which will make notation easier hereafter, we can write this change in loss as $\Delta L = L(\boldsymbol{\lambda}') - L(\boldsymbol{\lambda})$. We can then write our update to $\boldsymbol{\lambda}$ as $\lambda'_j = \lambda_j + \alpha_j$.

We want to figure out an approximation on ΔL and minimize that instead. In doing so, we have two major considerations: (1) that the approximation is tractable (easy to compute) and (2) that it is tight enough (so that it is close to what we actually want).

We start by writing out by filling in the equation with definitions:

$$\begin{aligned}
\Delta L &= \frac{1}{m} \sum_i \left[-\ln \left(\frac{\exp(g_{\boldsymbol{\lambda}'}(x_i)}{Z_{\boldsymbol{\lambda}'}} \right) + \ln \left(\frac{\exp(g_{\boldsymbol{\lambda}}(x_i)}{Z_{\boldsymbol{\lambda}}} \right) \right] \\
&= \left[\frac{1}{m} \sum_i \left(g_{\boldsymbol{\lambda}}(x_i) - g_{\boldsymbol{\lambda}'}(x_i) \right) \right] + \ln \left(\frac{Z_{\boldsymbol{\lambda}'}}{Z_{\boldsymbol{\lambda}}} \right) \\
&= (\text{term 1}) + (\text{term 2})
\end{aligned}$$

$$
\begin{aligned}
\text{(term 1)} &= \frac{1}{m}\sum_i \left(\sum_j \lambda_j f_j(x_i) - \sum_j \lambda'_j f_j(x_i)\right) \\
&= -\frac{1}{m}\sum_i \sum_j \alpha_j f_j(x_i) \\
&= -\sum_j \alpha_j \left(\frac{1}{m}\sum_i f_j(x_i)\right) \\
&= -\sum_j \alpha_j \hat{\mathbf{E}}[f_j]
\end{aligned}
$$

Everything up until this point has been exact. Now let's examine a part of term 2, where $\sum_x$ is over the entire space $\mathbf{X}$:

$$
\begin{aligned}
\frac{Z_{\boldsymbol{\lambda}'}}{Z_{\boldsymbol{\lambda}}} &= \frac{\sum_x \exp\left(\sum_j \lambda'_j f_j(x)\right)}{Z_{\boldsymbol{\lambda}}} \\
&= \frac{\sum_x \exp\left(\sum_j (\lambda_j + \alpha_j) f_j(x)\right)}{Z_{\boldsymbol{\lambda}}} \\
&= \frac{\sum_x \exp\left(\sum_j \lambda_j f_j(x) + \sum_j \alpha_j f_j(x)\right)}{Z_{\boldsymbol{\lambda}}} \\
&= \sum_x q_{\boldsymbol{\lambda}}(x) \exp\left(\sum_j \alpha_j f_j(x)\right)
\end{aligned}
$$

Because of convexity we can approximate the exponential of an average of terms as being less than or equal to the average of the exponentials of those same terms. This approximation gives us

$$
\frac{Z_{\boldsymbol{\lambda}'}}{Z_{\boldsymbol{\lambda}}} \leq \sum_x q_{\boldsymbol{\lambda}}(x) \sum_j f_j(x) e^{\alpha_j} = \sum_j \left(\sum_x q_{\boldsymbol{\lambda}}(x) f_j(x)\right) e^{\alpha_j} = \sum_j \mathbf{E}_{q_{\boldsymbol{\lambda}}}[f_j]\ e^{\alpha_j}
$$

Putting everything back together, we get

$$
\Delta L \leq -\sum_j \alpha_j \hat{\mathbf{E}}[f_j] + \ln\left(\sum_j \mathbf{E}_{q_{\boldsymbol{\lambda}}}[f_j]\ e^{\alpha_j}\right)
$$

Now we can optimize with respect to α_j. (We'll use yet another shorthand of $\hat{\mathbf{E}}[f_j] = \hat{\mathbf{E}}_j$ and $\mathbf{E}q_{\boldsymbol{\lambda}}[f_j] = \mathbf{E}_j$.)

$$
\frac{\partial \Delta L}{\partial \alpha_j} = -\hat{\mathbf{E}}_j + \frac{\mathbf{E}_j e^{\alpha_j}}{\sum_j \mathbf{E}_j e^{\alpha_j}} = 0
$$

We can also do $\alpha_j + c$ instead of α_j because of cancelation; if α'_j is a solution, then $\alpha_j = \alpha'_j + c$ is also a solution. Then, we can just choose c such that the denominator is equal to 1. Thus,

$$
\alpha_j = \ln\left(\frac{\hat{\mathbf{E}}_j}{\mathbf{E}_j}\right).
$$

Our algorithm, which is called "iterative scaling" can now have the updates

$$\lambda_{t+1,j} \leftarrow \lambda_{t,j} + \ln\left(\frac{\hat{\mathbf{E}}[f_j]}{\mathbf{E}_{q_{\hat{\lambda}_t}}[f_j]}\right).$$

If we switch to using the notation $p_t = q_{\boldsymbol{\lambda}_t}$, then we can rewrite this in terms of p, which gives us

$$p_{t+1}(x) \propto p_t(x) \prod_j \left(\frac{\hat{\mathbf{E}}[f_j]}{\mathbf{E}_{p_t}[f_j]}\right)^{f_j(x)}.$$

We are looking for p such that for all $t : \mathbf{E}_{p_t}[f_j] = \hat{\mathbf{E}}[f_j]$. Say on some round t this isn't true, e.g., $\mathbf{E}_{p_t}[f_j] < \hat{\mathbf{E}}[f_j]$; then, $\hat{\mathbf{E}}[f_j]/\mathbf{E}_{p_t}[f_j] > 1$; so we want $\mathbf{E}_{p_t}[f_j]$ to be larger. Since the base of the exponent is greater than 1 in this case, larger values of $f_j(x)$ will tend to get greater weight under p_{t+1}, so that the expectation will tend to get larger.

This gives us an intuition, but we need to show that it actually works, i.e., we need to prove that it converges to the distribution that we want.

Theorem 2. *p_t converges to q^* as $t \to \infty$.*

The technique for this proof is using an auxiliary or helper function.

Proof We define an auxiliary function A to be a real-valued function defined on the space of probability distributions on $\mathbf{X}$ which has three key properties:

(1) A is continuous
(2) A gives upper bound of change in loss
$L(\boldsymbol{\lambda}_{t+1}) - L(\boldsymbol{\lambda}_t) \leq A(p_t) \leq 0$
(3) $A(p) = 0 \Rightarrow p \in P$

We want to first show that if A exists, then it proves the desired result. Then, we'll show that A exists.

Say A exists. By property 2, we know that $L \geq 0$ and $L(\boldsymbol{\lambda}_t)$ never increases and never goes below 0. This implies that $\Delta L \to 0$, or $L(\boldsymbol{\lambda}_{t+1}) - L(\boldsymbol{\lambda}_t) \to 0$. By this and property (2), $A(p_t) \to 0$, because it is squeezed between the two bounds.

Say p is the limit of p_t, or $p = \lim_{t\to\infty} p_t$. Then we know three things:

(1) $p \in \overline{Q}$, since $p_t \in Q$ because our algorithm uses distributions of the form that defines Q and $\overline{Q}$ contains all limits of sequences in Q (by definition of closure).

(2) $A(p) = 0$ since A is continuous. The center equality in the following expression is the definition of continuous.

$$A(p) = A(\lim_{t\to\infty} p_t) = \lim_{t\to\infty} A(p_t) = 0.$$

(3) $p \in P$ by property 3. This gives us that $p \in P \cap \overline{Q}$ and therefore $p = q^*$.

Here we have assumed that $\lim p_t$ exists, which might not be the case. To argue this more carefully, note that the p_t's are in a compact space. Therefore, they have a convergent subsequence. By the argument that we just gave, that subsequence must converge to q^*. Since every convergent subsequence converges to this same unique point q^*, it can be argued that the entire sequence converges to q^*.

Recall from previously in this lecture:

$$\Delta L \leq -\sum_j \alpha_j \hat{\mathbf{E}}[f_j] + \ln\left(\sum_j \mathbf{E}_{q_\lambda}[f_j]\ e^{\alpha_j}\right)$$

and

$$\alpha_j = \ln\left(\frac{\hat{\mathbf{E}}_j}{\mathbf{E}_j}\right).$$

We also know that $\sum_j \mathbf{E}_{q_\lambda}[f_j]\ e^{\alpha_j} = 1$, thus giving us:

$$\Delta L = -\sum_j \ln\left(\frac{\hat{\mathbf{E}}_j}{\mathbf{E}_j}\right)\hat{\mathbf{E}}[f_j]$$

which looks a lot like relative entropy.

We also claim that $\mathbf{E}_q$ is a distribution; this is true by linearity of expectation: we know that $\mathbf{E}_q[\sum_j f_j] = 1$, therefore $\sum_j \mathbf{E}_q[f_j] = 1$ and $\mathbf{E}_q$ is a distribution.

Therefore,

$$\Delta L = -RE(\langle\hat{\mathbf{E}}[f_j]\rangle \mid\mid \langle\mathbf{E}_{p_t}[f_j]\rangle),$$

where the notation $\langle\hat{\mathbf{E}}[f_j]\rangle$ is shorthand for the vector $\langle\hat{\mathbf{E}}[f_1], \ldots, \hat{\mathbf{E}}[f_n]\rangle$.

For any distribution p, we can define

$$A(p) = -RE(\langle\hat{\mathbf{E}}[f_j]\rangle \mid\mid \langle\mathbf{E}_p[f_j]\rangle).$$

Let's review the three necessary properties. Relative entropy guarantees property (1)–continuity. Property (2) is satisfied because relative entropy is always ≥ 0, thus negative relative entropy must be ≤ 0. Property (3) is satisfied because $A(p) = 0$ implies, using the fact that relative entropy is zero if and only if the two distributions that are involved are identical, that for all $j : \hat{\mathbf{E}}[f_j] = \mathbf{E}_p[f_j]$ and therefore $p \in P$.

We've been looking at the batch setting. Next time, we'll take a look at the analogous problem in the online setting. Suppose one is placing bets at a horserace. We use the probability of a horse winning a race in order to place bets. Experts give advice on these probabilities, and you combine the expert advice into your own distribution, where your own is not too much worse than the best expert. More concretely,

for $t = 1, \ldots, T$ **do**
  each expert gives estimated probability distribution $p_{t,i}$ over space $\mathbf{X}$
  learner combines these into own probability distribution q_t
  observe outcome $x_t \in \mathbf{X}$
  loss suffered by learner is log loss: $-\ln q_t(x_t)$
end for

We get a cumulative loss function $-\sum_t \ln q_t(x_t)$ and expert loss $-\sum_t \ln p_{t,i}(x_t)$. Next time, we'll look for a bound like:

$$-\sum_{t=1}^{T} \ln q_t(x_t) \leq \min_i \left(-\sum_{t=1}^{T} \ln p_{t,i}(x_t) \right) + (\text{small amount}).$$