Online Learning using Kernel Concepts

S. Sumitra

In batch learning all the data points are available before training starts. However in on-line learning training points arrive sequentially and hence we cannot use the batch learning frame for online learning. The online learning frame work is described below.

Let (x_t, y_t) be the data arrived at time t. Let f_t is the model developed at time t. Let it belongs to RKHS $\mathcal{F}$ whose reproducing kernel is k. Also, we define the instantaneous regularized risk on a single example (x_{t+1}, y_{t+1}) at time $(t+1)$ as

$$\hat{g}_{t+1}(f) = \frac{1}{2}\|f(x_{t+1}) - y_{t+1}\|^2 + \frac{\mu}{2}||f||^2 \tag{1}$$

where, $\mu > 0$ is the regularisation parameter.

[*Here we have used regularized least square loss function. Instead of that, other appropriate loss functions can be used.*]

Now,

$$f_{t+1} = \arg\min_{f \in \mathcal{F}} \hat{g}_{t+1}(f) \tag{2}$$

We use stochastic gradient descent (SGD) to find the model at time $t+1, t = 0, 1, \ldots$.

Given an initial approximation, $f_0 \in \mathcal{F}$, the method of SGD is

$$f_{t+1} = f_t - \eta_{t+1} \left(\nabla \hat{g}_{t+1}(f)\right)_{f=f_t} \tag{3}$$

Now, $f_t(x_{t+1}) = \langle f_t, k_{x_{t+1}} \rangle, k_{x_{t+1}} \in \mathcal{F}$. Therefore

$$\nabla \hat{g}_{t+1}(f_t) = [f_t(x_{t+1}) - y_{t+1}]k_{x_{t+1}} + \mu f_t$$

Hence it is clear that $f_t = \sum_{i=1}^{t} \alpha_i k_{x_i}, \alpha_i \in \mathbb{R}, k_{x_i} \in \mathcal{F}$.

Now, one iteration of the method is

$$\begin{aligned} f_{t+1} &= f_t - \eta_{t+1}\left[(f_t(x_{t+1}) - y_{t+1})k_{x_{t+1}} + \mu f_t\right] \\ &= (1 - \eta_{t+1}\mu)f_t - \eta_{t+1}(f_t(x_{t+1}) - y_{t+1})k_{x_{t+1}} \\ &= (1 - \eta_{t+1}\mu)\sum_{i=1}^{t} \alpha_i k_{x_i} - \eta_{t+1}(f_t(x_{t+1}) - y_{t+1})k_{x_{t+1}} \end{aligned}$$

Hence at step $(t+1)$, the coefficients are updated as:

$$\alpha_{t+1} := -\eta_{t+1}(f_t(x_{t+1}) - y_{t+1}) \tag{4}$$

$$\alpha_i := (1 - \eta_{t+1}\mu)\alpha_i, i < (t+1) \tag{5}$$

Thus new model = $(1 - \eta_{t+1}\mu)$ * previous model + $\alpha_{t+1}k_{x_{t+1}}$, where, $\alpha_{t+1} := -\eta_{t+1}(f_t(x_{t+1}) - y_{t+1})$.

From the above equations it is clear that the amount of computation required for prediction grows linearly with the number of terms in the expansion and hence computation complexity increases with time. This could overcome with the aid of the regularization term, since at each iteration the coefficients α_i with $i \neq t$ contracts by $(1-\mu\eta)$ for constant learning rate $\eta_t = \eta$. Therefore, after k iterations, the coefficient will be reduced to by $(1 - \mu\eta)^k\alpha_i$. Hence the smaller terms can be dropped, which makes the algorithm more computationally effective.

References

(1) Online learning with kernels by Jyrki Kivinen, A. J. Smola and Robert C. Williamson.