Chao Lan
EN 4084A
clan@uwyo.edu

Online Learning

In batch learning task, a model is learned from a batch of data. In online learning task, a model is continuously learned (or, updated) from a sequence of data. Let $x_1, x_2, \ldots, x_n$ be a sequence of instances, and y_t be the label of x_t. A general framework of online learning is as follows:

[0] initialize f

[1] f receives x_t at time t

[2] f predicts label $\hat{y}_t$ for x_t

[3] f receives true label y_t

[4] f updates itself based on some loss $\ell(y_t, \hat{y}_t)$

[5] repeat step 1-4 until convergence

After n rounds, f will incur a cumulative loss

$$L_n(f) = \frac{1}{n}\sum\nolimits_{t=1}^{n} \ell(\hat{y}_t, y_t). \tag{1}$$

The goal of online learning is to find a model f that can minimize $L_n(f)$. The challenge is one does not store all received data and continuously use them to retrain the model – that's too inefficient in terms of computation and memory. Instead, one can keep updating the model using newly received instances – this can be implemented using stochastic gradient descent (SGD).

Consider the following loss

$$L_n(f) = \sum\nolimits_{t=1}^{n} \ell(y_t, f(x_t)). \tag{2}$$

Ordinary gradient descent updates f using the gradient of L_n on all instances, i.e.,

$$f = f - \eta \cdot \frac{\partial}{\partial f} L_n(f) = f - \eta \cdot \sum\nolimits_{t=1}^{n} \frac{\partial}{\partial f} \ell(y_t, f(x_t)). \tag{3}$$

SGD approximates (3) using the gradient on a single instance x_t at time t, i.e.,

$$f \;=\; f - \eta \cdot \sum\nolimits_{t=1}^{n} \frac{\partial}{\partial f} \;\approx\; f - \eta \cdot \frac{\partial}{\partial f} \ell(y_t, f(x_t)). \tag{4}$$

Therefore, Step 4 in the online learning framework can be implemented as

[4] f updates itself based on $f = f - \eta \cdot \frac{\partial}{\partial f} \ell(\hat{y}_t; y_t)$.

Stochastic gradient descent will converge if f is a bit more than convex. In general, it converges more slowly than ordinary gradient descent.

The Peceptron Algorithm

Consider binary classification task with label set$Y = \{-1, +1\}$. Let $\text{sign}(z)$ be a sign function outputting $+1$ if $z \geq 0$ and -1 otherwise. Perceptron is an online learning algorithm for model

$$f(x) = \text{sign}(x^T \beta). \tag{5}$$

It finds an f that can minimize the following hinge loss (that only counts misclassified instances)

$$L_n(f) = \sum\nolimits_{t=1}^{n} \max\{0, -y_t(x_t^T \beta)\}. \tag{6}$$

[*Discussion*] How does the hinge loss count only misclassified instances?

Let I_{mis} be the index set of misclassified instances. The gradient of L_n is

$$L_n'(f) = \sum\nolimits_{t \in I_{mis}} - y_t x_t. \tag{7}$$

[*Exercise*] Derive (7).

Perceptron applies SGD to optimize L_n and therefore approximates (7) by

$$L_n' \;\approx\; -y_t x_t, \qquad \text{if } x_t \text{ is misclassified.} \tag{8}$$

Perceptron learning is summarized in Algorithm 1. A common choice of learning rate is $\eta = 1$.

Algorithm 1 The Perceptron Learning Algorithm

Input: learning rate η
Initialization: (randomly) initialize model β
for t = 1 to n **do**
1: model receives instance x_t
2: model predicts $\hat{y}_t = \text{sign}(x_t^T \beta)$
3: model receives true label y_t
4: if $\hat{y}_t \neq y_t$, then model updates itself by

$$\beta = \beta - \eta \cdot (-y_t \cdot x_t) = \beta + \eta y_t x_t. \tag{9}$$

otherwise, model does nothing.
end for