Proposal for "Stochastic Gradient Descent: Implementation and Application"

Minjie Fan, 998585352

Let us consider the following scenario: $(\boldsymbol{X}_i, Y_i), i = 1, \cdots, n$ are samples from a population $(\boldsymbol{X}, Y)$, where Y is a scalar and $\boldsymbol{X}$ can be a vector. A parametric predictive function $f_{\boldsymbol{\theta}}$ is used to predict Y based on $\boldsymbol{X}$. The predictive accuracy is measured by the loss function $l(\widehat{Y}, Y) = l(f_{\boldsymbol{\theta}}(\boldsymbol{X}), Y)$. The parameter vector $\boldsymbol{\theta}$ is estimated by minimizing the risk function

$$E(l(\widehat{Y}, Y)) = \int_{\Omega} l(f_{\boldsymbol{\theta}}(\boldsymbol{X}), Y) dP.$$

However, since the distribution of $(\boldsymbol{X}, Y)$ is unknown, we usually approximate the risk by its empirical version

$$E_n(l(\widehat{Y}, Y)) = \frac{1}{n} \sum_{i=1}^{n} l(f_{\boldsymbol{\theta}}(\boldsymbol{X}_i), Y_i) = \frac{1}{n} \sum_{i=1}^{n} l_i(\boldsymbol{\theta}).$$

This kind of objective function, i.e., $\sum_i l_i(\boldsymbol{\theta})$, is very common in the framework of supervised learning. Additionally, it is closely related with the M-estimator in statistics.

Traditional numerical optimization methods, such as gradient descent, Newton's method and BFGS, can be used to minimize the objective function. For example, at each iteration, the gradient descent method updates $\boldsymbol{\theta}$ as follows:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \alpha \sum_{i=1}^{n} \nabla l_i(\boldsymbol{\theta}^{(t)}).$$

However, when the sample size is very large, we need to evaluate the gradients of all the summand functions, which can be very time-consuming. Given the limitation of the traditional methods, stochastic gradient descent (see Bottou (2012) for details) is proposed to speed up the computation by approximating the true gradient by the sum of a randomly selected subset of the summand functions. As the simplest example, we may update $\boldsymbol{\theta}$ as follows:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \alpha \nabla l_i(\boldsymbol{\theta}^{(t)}),$$

where the index i is randomly selected at each iteration.

The stochastic gradient descent algorithm can be applied to a number of classic machine learning schemes. I plan to apply it to linear regression, logistic regression and the Lasso (Tibshirani, 1996), and compare its performance with the L-BFGS (Limited-memory BFGS) algorithm.

References

Bottou, L. (2012) Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade*, 421–436. Springer.

Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288.