Stochastic Average Gradient on Logistic Regression and Markov Random Field

Eric Xin Zhou

March 15, 2015

1 Summary

Schmidt et al. [3] proposed the Stochastic Average Gradient(SAG) algorithm as a faster solver for optimizing the sum of finite number of smooth convex functions. Our project aims to create a SAG R package of optimizer solver which applies SAG algorithm to l-2 logistic regression and Markov random field.

2 Description

2.1 Minimizing finite sums problems

In large-scale machine learning, there are plethora of optimization problems based on empirical risk minimization principle in statistical learning theory, this class of problems involve computing a minimizer of a finite sum of a set of smooth functions, since the sum structure is a natural form of loss function over large numbers of data points (1). Therefore, our goal is to minimize $g(x)$ function with respect to optimization variable x, which is a vector of real number($x \in \mathbb{R}^p$).

$$g(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \tag{1}$$

Additionally the most widely successful class of algorithms to minimize sum structure problems are stochastic gradient methods. While stochastic average method is a faster algorithm which achieves $\mathcal{O}(\rho^k)$ coverage rate when summed function is strongly-convex[2, 3], it reduces the cost of iteration of gradient descent by keeping last iteration's gradient in memory.

2.2 SAG applied to logistic regression

We will implement the SAG R package, which will contain a solver for L-2 norm regularized logistic regression. For multiple data logistic regression, its loss function can be written as (2).

$$g(x) := \frac{\lambda||x||^2}{2} + \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp(-b_i a_i^T x)\right) \tag{2}$$

Where a_i is predictor in logistic regression problem and $a_i \in \mathbb{R}^p$, and b_i is response of this problem. For categorical problem, $b_i \in \{-1, 1\}$. Furthermore, x is the coefficient we will estimate and regularize parameter λ is to control our fitting parameters and we replace $f_i(x)$ in (1) by right-hand-side of equation (4). So the L-2 regularize logistic regression problem is an optimization problem for finite sum and SAG can be applied to solving this class of problems. In SAG algorithm, randomly selected $f_i(x)$'s gradient will be evaluated by (5).

$$\min_{x \in \mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp(-b_i a_i^T x)\right) + \frac{\lambda||x||^2}{2} \tag{3}$$

$$f_i(x) = \log\left(1 + \exp(-b_i a_i^T x)\right) + \frac{\lambda||x||^2}{2} \tag{4}$$

$$\nabla f_i(x) = \lambda x - \frac{b_i a_i \exp(-b_i a_i^T x)}{1 + \exp(-b_i a_i^T x)} \tag{5}$$

Then we will compare the convergency rate and result of SAG solver with the L-2 norm regularized logistic regression in package glmnet[1]/optimx. We will use `system.time` to record the time each solver consume.

2.3 SAG applied to CRF

Also, Schmidt et al. [4] also claimed that SAG can be implemented to train conditional random fields(CRF).

CRF is an undirected graph model for observations **a** and latent variables **b**. For example, in a chain graph model of words recognition, **a** represents a sequence of observations, such as different image of different letter. **b** represents a sequence of hidden state variables(real letters of words) that needs to be inferred given the observations. The **b**(hidden letter states) are structured to form a chain, with an edge between b_j and b_{j+1}. The observation $\mathbf{a}_j$ depends on $\mathbf{b}_j$. Therefore, we can define the conditional dependence of **a** and **b** through a set of *feature functions* $F_k(b_j, b_{j-1}, a_j)$, which are indicator functions of each edge in the graph model. Based on these *feature functions* and their correspond parameters x. A conditional probability $p(\mathbf{b}|\mathbf{a}, x)$ is built to estimate the hidden letter structure of our observation for traning and inference.

Therefore in general CRF chain model, we are considering a multiple observed conditional distribution $\mathbf{P}(\mathbf{b}|\mathbf{a}) = \prod_{i=1}^{n} \mathbf{P}(\mathbf{b}^{(i)}|\mathbf{a}^{(i)})$ in (6), for each observation, $\mathbf{a}^{(i)} = (a_1^{(i)}, a_2^{(i)}, ..., a_T^{(i)})$ means that we have a T-state structure to estimate, also $\mathbf{b}^{(i)} = (b_1^{(i)}, b_2^{(i)}, ..., b_T^{(i)})$ is a vector of T hidden states. Furthermore, $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, ..., x_p^{(i)})$ is the correspond $\mathbb{R}^p$ parameter vector.

$$\mathbf{P}(\mathbf{b}^{(i)}|\mathbf{a}^{(i)}) = \frac{\mathbf{P}(\mathbf{b}^{(i)},\mathbf{a}^{(i)})}{Z^{(i)}} = \frac{1}{Z^{(i)}}\prod_{t=1}^{T}\exp\,(\sum_{k=1}^{p} x_k F_k(b_{t-1}^{(i)}, b_t^{(i)}, a^{(i)})) \tag{6}$$

$$Z^{(i)} = \sum_{\mathbf{b'^{(i)}}} \exp\,(\sum_{t=1}^{T}\sum_{k=1}^{p} x_k F_k(b_{t-1}^{'(i)}, b_t^{'(i)}, a_t^{(i)}))$$

Where $F_k(b_{t-1}^{(i)}, b_t^{(i)}, a^{(i)})$ is a *feature function* which represents $\mathbf{1}_{b_t^{(i)}=i}\mathbf{1}_{b_{t-1}^{(i)}=j}$ or $\mathbf{1}_{b_t^{(i)}=i}\mathbf{1}_{a_t^{(i)}=o}$, where $\mathbf{1}_{x=a}$ is an indicator function to display whether $x = a$ or not. Therefore, we can treat $F_k(b_{t-1}^{(i)}, b_t^{(i)}, a_t^{(i)})$ as a function for each edge in factor graph G. CRF can provide a discriminative model to predict each input a's label b^* by maximizing likelihood(7), where $b^* = \underset{b}{\operatorname{argmax}}\,\mathbf{P}(b|a, \mathbf{x})$.

To estimate parameters $\mathbf{x} = \{x_k\} \in \mathbb{R}^p$ in the $\mathbf{P}(\mathbf{b}|\mathbf{a})$, we always train l2 regularized CRF by to obtain highest likelihood(7) on training data, where the λ is regularized parameter to control the fitting parameters.

$$L(x) = \frac{1}{n}\log[\mathbf{P}(\mathbf{b}|\mathbf{a})] = \frac{1}{n}\sum_{i=1}^{n}[\frac{\lambda}{2}||x||^2 - \sum_{t=1}^{T}\sum_{k=1}^{p} x_k F_k(b_{t-1}^{(i)}, b_t^{(i)}, a_t^{(i)}) + \log(Z^{(i)})] \tag{7}$$

CRF model will minimize $L(x)$ with respect to parameters $\mathbf{x}$ in (8), so CRF can be viewed as a special form of finite sum of smooth function in (9). For SAG, since $\nabla f_i(x) = (\frac{\partial f_i(x)}{\partial x_1}, \frac{\partial f_i(x)}{\partial x_2}, ..., \frac{\partial f_i(x)}{\partial x_p})^T$, we will estimate each $\frac{\partial f_i(x)}{\partial x_k}$, where $k \in [1, p]$ for $\nabla f_k(x)$ in (10).

$$\min_{x\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n}[\frac{\lambda}{2}||x||^2 - \sum_{t=1}^{T}\sum_{k=1}^{p} x_k F_k(b_{t-1}^{(i)}, b_t^{(i)}, a_t^{(i)}) + \log(Z^{(i)})] \tag{8}$$

$$f_i(x) = \frac{\lambda}{2}||x||^2 - \sum_{t=1}^{T}\sum_{k=1}^{p} x_k F_k(b_{t-1}^{(i)}, b_t^{(i)}, a_t^{(i)}) + \log(Z^{(i)}) \tag{9}$$

$$\frac{\partial f_i(x)}{\partial x_k} = \lambda x_k - \sum_{t=1}^{T} F_k(b_{t-1}^{(i)}, b_t^{(i)}, a_t^{(i)}) + \sum_{\mathbf{b'^{(i)}}}\sum_{t=1}^{T} F_k(b_{t-1}^{'(i)}, b_t^{'(i)}, a_t^{(i)})\mathbf{P}(\mathbf{b}^{'(i)}|\mathbf{a}^{(i)}) \tag{10}$$

Our implementation of the SAG R package will contain a solver for training L-2 regularize CRF model. Also with the R package CRF for UGM inference and traning, we will compare our SAG training function both the convergency rate and estimate parameter value. We will use `system.time` to record each solver's time consuming and compare two different method's result.

References

[1] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

[2] Nicolas L Roux, Mark Schmidt, and Francis R Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. pages 2663–2671, 2012.

[3] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *arXiv preprint arXiv:1309.2388*, 2013.

[4] Mark Schmidt, Ann Clifton, and Anoop Sarkar. Non-uniform stochastic average gradient method for training conditional random fields. 2014.