Some Comments for Distributed SDCA

Mingrui Liu

April 28, 2017

Abstract

In this note, we will give some comments for Tianbao's nips 2013 paper: trading compututation with communication: distributed SDCA.

Suggestions

Reading section 3.1 of the paper before going into this note is highly recommended. For intuitive explanations, please refer to the page 104 of Tianbao's tutorial:
https://homepage.cs.uiowa.edu/ tyng/kdd15-tutorial.pdf.

Notations for the Algorithms

In this section, the detailed comments for the Algorithm **SDCA-mR** and the Algorithm **IncDual** are provided.

For SDCA-mR

SDCA-mR is an algorithm that should be implemented on the k-th machine, where k can be taken value from $1, \ldots, K$.

- n is the total number of samples.
- For the input part, m examples are processed on the k-th machines.
- n_k is a number which is less than or equal to m. Normally choose $n_k = m$.
- The choice of g in the procedure **SDCA-mR**:

$$g(w) = \frac{1}{2}\|w\|^2,$$

  We may not need to consider a general g. g being a quadratic function is actually the same with the original SDCA paper (Shai Shalev-Shwartz and Tong Zhang).

- g^* is the convex conjugate of g, which is defined as

$$g^*(x) = \sup_y (x^\top y - g(y)).$$

You can take the derivate with respect to y and set it to be zero to get the closed form solution of g^*. Actually, in our project, if $g(x) = \frac{1}{2}\|x\|^2$, then $g^*(x) = \frac{1}{2}\|x\|^2$, and $\nabla g^*(x) = x$. (**Note that x is a column vector instead of a scalar.**)

- Note that both $\alpha_{k,i}^{(t)}$ and $\triangle\alpha_{k,i}$ are scalars.
- The inner loop iteration for $j = 1, \ldots, m$ ends before the **Reduce** step.
- At the **Reduce** step, add the sum part, which is $\frac{1}{\lambda n}\sum_{j=1}^{m} \triangle\alpha_{k,i_j} x_{k,i_j}$ of each process to v^{t-1}. More formally,

$$v^t \leftarrow v^{t-1} + \sum_{k=1}^{K} \frac{1}{\lambda n} \sum_{j=1}^{m} \triangle\alpha_{k,i_j} x_{k,i_j}$$

  After finishing the update, broadcast the updated value to all K processes until the T-th iteration.

For IncDual

Option I at the Routine **IncDual**(w, scl):

For Hinge Loss:

$$\triangle\alpha_{k,i} = y_{k,i} \max(0, \min(1, \frac{1 - x_{k,i}^\top w^{t-1} y_i}{scl\|x_{k,i}\|^2/\lambda n} + \alpha_{k,i}^{t-1} y_{k,i})) - \alpha_{k,i}^{t-1}$$

More options

You can also consider the practical variant (at Figure 2) of distributed SDCA, which may be faster in practice.