Inexact Newton Method Implementation

Distributed Optimization and Games Project

***Abstract***: Optimization is one of the most important part of the machine learning, and the most famous used algorithms are the 1st order methods gradient-based methods. Recently, second order methods have become attractive, however those methods can be not so efficient due to the difficulties related with the approximation of the Hessian. In this project we are going to study, implement and analyze one of those second order methods, the Inexact Newton Method and one of its implementations, the Subsampled Hessian-Free Newton Method.

***Introduction:*** The simplest algorithm used to perform iterative minimization is the gradient descent. This method is part of the first order methods due to the fact that it take advantage of the gradient properties in order to reach a local minimum. Sometimes this could be not enough imposing significant limitations in particular because the first order methods don’t take into account the curvature of the error function causing a slow model training. Another reason that could lead us to avoid the first order methods is that they are not scale invariant and so changing the scale of data, also the behavior of the algorithm will change.

.

.

.

***Inexact Newton Method:*** Thanks to the scale invariance and quadratic rate convergence properties, Newton method is a good candidate to be an optimization algorithm. Problems raise with this method when it is applied to a multidimensional problem due to the calculation and inversion of the second derivative of the gradient that create the Hessian matrix. There exist some variants of this method that allow to scale the computation while maintaining the benefit of the second order methods. This method is called Inexact Newton Method and execute the update of the parameters on this way where satisfies . Finding the requires lot of resource in terms of memory and storage, so instead of directly solve the equation, it can be solved in an inexact way using the conjugate gradient method (described below). The benefits of this approach are that is not required to access the Hessian matrix but is just needed the hessian-vector. This family of algorithm that the inexact Newton Method take part are called Hessian-free. Taking into account as example:

Parameter vector and we will have and then We have so presented an example that can find without accessing

***Conjugate gradient method:*** The Conjugate gradient (CG) is an algorithm to find numerical solutions on the form , where A is a positive defined matrix, is the vector we want to fund and is an already known vector. In our case, and .

…

***Subsampled Hessian-Free Newton Method:*** The Conjugate gradient method is a flexible solution because allow us to control the number of the CG iteration, but if we don’t have preconditions this method can be expensive because one Hessian-vector product is as same cost as computing one gradient. A proposed, more efficient method is the Subsampled Hessian-Free Newton Method.

This method is based on the fact that computing the Hessian-vectors in all the dataset is still computationally expensive. Instead we will use a smaller sample , this subsample should be taken small enough so that the computation cost can be significantly reduced but on the other hand it should be large enough in order to have a good quality of the curvature information obtained by the Hessian-vector. We are

***Implementation:***