DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

OPTIMIZATION METHODS FOR LARGE-SCALE MACHINE LEARNING

November 15, 2017

Aditi Gangrade (20162059)

Shreya Singh (20162060)

Rohit Tiwari (20162085)

Dishant Chauhan (20162109)

(under the guidance of Dr. Pawan Kumar)

Introduction

As the training data sets become large, the classical gradient methods tend to be slow in convergence. The goal of this project is to identify a suitable optimization method for a given cost function, and then to investigate a class of parallel optimization solvers that are suitable for large scale machine learning applications. Effectiveness of our algorithms will be tested on existing datasets.

Approach

Almost every machine learning problem revolves around minimizing a convex cost function. Traditional approaches use Gradient Descent method to accomplish this. However, as gradient descent is based on single derivative (i.e. the slope of the function at the given point), the convergence is slow.

To overcome this issue, we move to a better approximation that takes into account the curvature details of the given cost function by computing the double derivative or the Hessian matrix.

Another issue that crops up while using the double derivative is the large memory space required by the Hessian matrix. As a solution to the memory issue, we use do not store the complete Hessian and instead, calculate the matrix-vector product on the go, as required by the iterative solution of Conjugate Gradient Method.

Algorithm : Subsampled Hessian-Free Newton Methods

1: Choose an initial iterate w_1.

2: Choose constants $\rho \in (0,1)$, $\gamma \in (0,1)$, $\eta \in (0,1)$, and $max_{cg} \in \mathbb{N}$.

3: **for** k = 1,2,... **do**

4: Generate realizations of ξ_k and ξ_k^H corresponding to $S_k^H \subseteq S_k$.

5: Compute s_k by applying Hessian-free CG to solve

. $$\nabla^2 f_{S_k^H}(w_k;\xi_k^H)\mathrm{s} = -\nabla f_{S_k}(w_k;\xi_k)$$

. until max_{cg} iterations have been performed or a trial solution yields

. $$\| r_k \|_2 := \|\nabla^2 f_{S_k^H}(w_k;\xi_k^H)\mathrm{s} + \nabla f_{S_k}(w_k;\xi_k)\|_2 \leq \rho \| \nabla f_{S_k}(w_k;\xi_k)\|_2.$$

6: Set $w_{k+1} \leftarrow w_k\ a_k s_k$, where $a_k \in \{ \gamma^0, \gamma^1, \gamma^2, ... \}$ is the largest element with

. $$f_{S_k}(w_{k+1};\xi_k) \leq f_{S_k}(w_k;\xi_k) + \eta a_k \nabla f_{S_k}(w_k;\xi_k)^T s_k.$$

Gradient Logistic Regression

$$
\begin{aligned}
h_\theta(x) &= g(\theta^T x) \\
g(z) &= \frac{1}{1+e^{-z}} \\
z &= \theta^T x \\
\frac{\partial h_\theta(x)}{\partial \theta} &= (h_\theta(x))(1-h_\theta(x)).x
\end{aligned}
$$

$$
\begin{aligned}
J(\theta) &= \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)} log(h_\theta(x^{(i)})) - (1-y^{(i)}) log(1-h_\theta(x^{(i)}))] \\
\frac{\partial J(\theta)}{\partial \theta_j} &= \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}\frac{1}{h_\theta(x^{(i)})}h_\theta(x^{(i)})(1-h_\theta(x^{(i)}))x_j^{(i)} \\
&\quad -(1-y^{(i)})\frac{1}{(1-h_\theta(x^{(i)}))}h_\theta(x^{(i)})(1-h_\theta(x^{(i)}))(-x_j^{(i)})] \\
&= \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}(1-h_\theta(x^{(i)}))x_j^{(i)}) + (1-y^{(i)})h_\theta(x^{(i)})x_j^{(i)}] \\
&= \frac{1}{m}\sum_{i=1}^{m}[-x_j^{(i)}y^{(i)} + x_j^{(i)}y^{(i)}h_\theta(x^{(i)}) + x_j^{(i)}h_\theta(x^{(i)}) - x_j^{(i)}y^{(i)}h_\theta(x^{(i)})] \\
&= \frac{1}{m}\sum_{i=1}^{m}[h_\theta(x^{(i)}) - y^{(i)}]x_j^{(i)}
\end{aligned}
$$

Hessian

$$
\begin{aligned}
\frac{\partial J(\theta)}{\partial \theta_j} &= \frac{1}{m}\sum_{i=1}^{m}[h_\theta(x^{(i)}) - y^{(i)}]x_j^{(i)} \\
\frac{\partial^2 J(\theta)}{\partial \theta_j \partial \theta_k} &= \frac{\partial \frac{1}{m}\sum_{i=1}^{m}[h_\theta(x^{(i)})x_j^{(i)} - y^{(i)}x_j^{(i)}]}{\partial \theta_k} \\
&= \frac{1}{m}\sum_{i=1}^{m}[h_\theta(x_j^{(i)})(1-h_\theta(x_j^{(i)}))x_k^{(i)}x_j^{(i)}]
\end{aligned}
$$

Results

Cost Function :

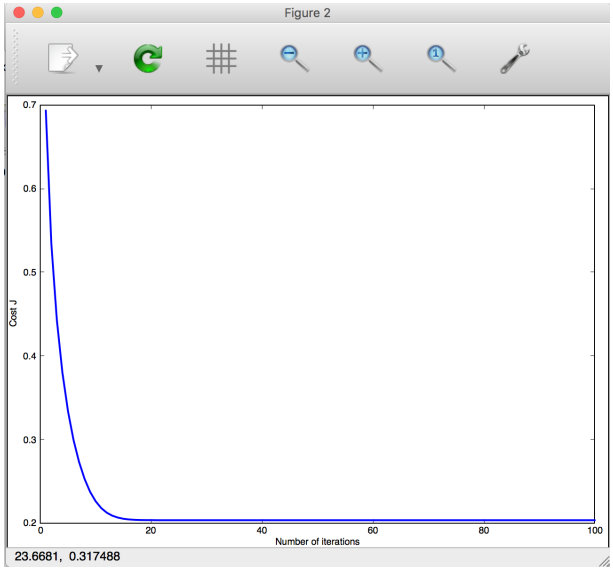
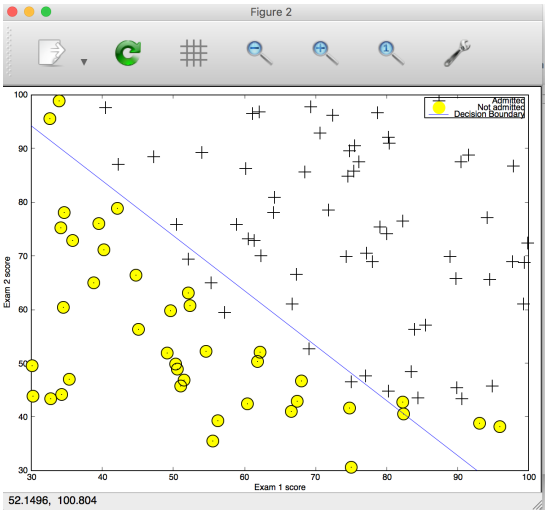
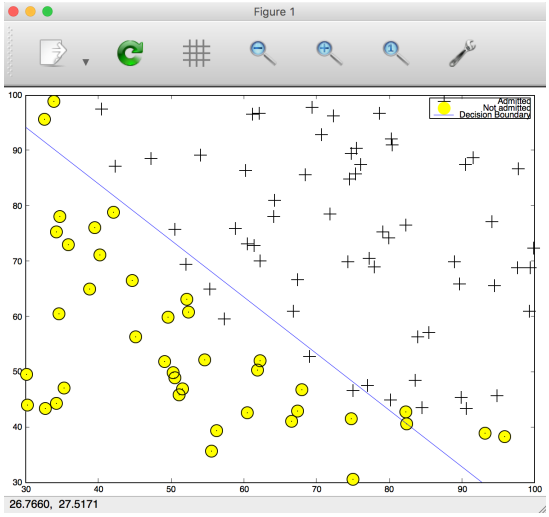


Figure 1: Cost function of proposed method

Decision Boundary :



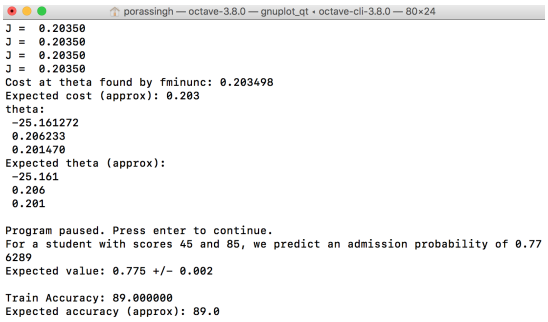
(a) Existing Method



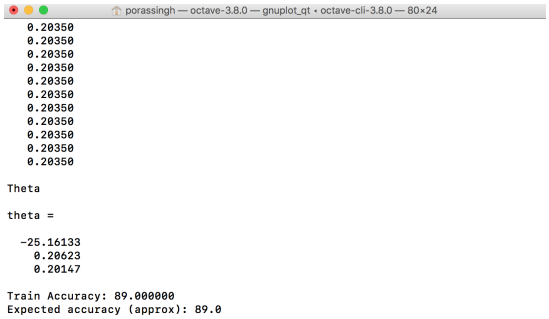
(b) Proposed Method

Figure 2: Decision boundary

Parameter Values (θ) :



(a) Existing Method



(b) Proposed Method

Figure 3: Parameter Values

Comparison of Existing Method and Proposed Method :

	Existing Method	Proposed Method
# iteration reqd for convergence	67	25
CPU time(in sec)	0.45	2.13

References

Leon Bottou, Frank E. Curtis, Jorge Nocedal, *Optimization Methods for Large-Scale Machine Learning,* June 6, 2017