Nesterov Accelerated Stochastic Broyden-Flecther-Goldfarb-Shanno

Yuval Lavie & Asaf Ahi Mordechai

February 24, 2020

Abstract

Acceleration methods are used to increase the convergence rate of the optimizer and to avoid exhausting its iterations or achieving its tolerance ahead of expected time. we introduce the Acceleration methods which are commonly used for first order minimization and apply them on a variant of the BFGS algoritm. we test our Accelerated BFGS on several functions and show its performance on a classification task with Support Vector Machines as measured against the Steepest Descent algorithm

Part I
Introduction

1 Supervised Machine Learning

A supervised machine learning problem is defined by the tuple $\Big(\phi, \mathbb{P}, S, H\Big)$ where $\mathbb{P}$ is an unknown distribution which can be sampled from and S is a sample space which acts as a window for the learner to view $\mathbb{P}.H$ is a hypothesis class which repesents some form of prior belief on P and is supposed to act as a decision-maker for the learner and $\phi : \mathbb{R} \to \mathbb{R}^+$ is a psuedo-measure of error when applying $h \in H$ to the sample space.

1. $S = \Big((\overline{x}_1, y_2), (\overline{x}_2, y_3), ...(\overline{x}_m, y_m)\Big)$ where y_i serves as the ground truth we are supposed to learn on $\overline{x}_i$ and may be generated deterministically or stochastically.
2. $\psi : \mathbb{R} \to \mathbb{R}^+$ is a pesudo measure of error for a specific example from the sample space

The goal of this task is the minimize the generalization error known as $\mathbb{E}_{(\overline{x},y)\sim\mathbb{P}}\Big[\psi\big(h(\overline{x}), y\big)\Big]$

but since the probability distribution is unknown to us the best we can hope for is to minimize the empirical approximation $\frac{1}{m}\sum_{i=1}^{m}\psi\big(h(\overline{x}_i), y_i\big)$

2 Cast as optimization

A supervised machine learning problem is an unconstrained optimization task where the objective is differentiable

$$\underset{\overline{W}\in\mathbb{R}^d}{argmin}\left[\phi\left(\overline{W}, S\right)\right] = \frac{1}{m}\sum_{i=1}^{m}\psi\left(\overline{W}, (\overline{x}_i, y_i)\right)$$

we have no hope of finding a global minima but for practical reasons a local minimizer is sufficient and has shown incredible results in prediction tasks such as regression and classification.

3 Stochastic Approximation

When our sample space is too big we may encounter a computationally infeasiable problem even when using the less accurate but more efficient first order algorithms such as Steepest Descent.a dominant solution today is to use an approximation of the gradient where instead of passing through the whole sample space we sample it and calculate a loose gradient.Stochastic Gradient Descent is a key algorithm in almost all supervised machine learning schemes.

Algorithm 1 Stochastic Gradient Descent

```
INPUT $\nabla\psi, S, \alpha, T$

PROCEDURE

  • While $t < T$ :
      1. Sample $(\overline{x}, y) \sim S$ or $B = \big\{(\overline{x}_1, y_1), ..., (\overline{x}_m, y_m)\big\} \sim S$
      2. $\overline{d} = -\nabla\psi\left(\overline{W}_t, (\overline{x}, y\right)$ or $-\frac{1}{|B|}\sum_{(x,y)\in B}\nabla\psi\left(\overline{W}_t, (\overline{x}, y)\right)$
      3. $\overline{W}_{t+1} = \overline{W}_t + \alpha \cdot \overline{d}$
```

4 Momentum

Using an approximation of the gradient does not assure us a decreasing direction in the objective and exact / inexact line searches become redundant.each step occurs regardless of the acceptable step length and we may suffer increases,exhasution of iterations or very slow advancements due to low curvature areas.we seek to help the optimizer to avoid these errors using a momentum which counts on previous steps to either adjust to step length or the direction.we now present several known methods

- Heavy Ball Momentum

$$\overline{W}_{t+1} = \overline{W}_t + \alpha \cdot \overline{d} + \beta\Big(\overline{W}_t - \overline{W}_{t-1}\Big)$$

- Force Momentum

$$\begin{gathered}\overline{V}_{t+1} = \mu \cdot \overline{V}_t + \alpha \cdot \overline{d} \\ \overline{W}_{t+1} = \overline{W}_t + \overline{V}_{t+1}\end{gathered}$$

- Nesterov Momentum

$$\begin{gathered}\overline{V}_{t+1} = \mu \cdot \overline{V}_t + \alpha \cdot \Big(- \nabla\varphi(\overline{W} + \mu\overline{V}_t)\Big) \\ \overline{W}_{t+1} = \overline{W}_t + \overline{V}_{t+1}\end{gathered}$$

Part II

Quasi Newton with Momentum

Quasi-Newton algorithms offer an attractive alternative as they do not require computation of the Hessian and may be used in medium sized machine learning problems where it is still applicable to calculate the gradients efficiently.we build upon the BFGS algorithm and introduce a stochastic variant called o-BFGS which works even in large-scale machine learning problems which are characterized by a convex loss function.

4.1 General Framework

We start by stating that most known unconstrained minimization algorithms in fact use the same update rule defined by

$$\overline{W}_{t+1} = \overline{W}_t + \alpha \cdot \Big[- B_k^{-1} \nabla\psi_k\Big]$$

1. Steepest Descent $\iff B_k = I$
2. Newton's Method $\iff B_k = \nabla^2\psi_k$
3. Quasi Newton $\iff B_k \approx \nabla^2\psi_k$

4.2 Broyden-Flecther-Goldfarb-Shanno Algorithm

Instead of calculating the Hessian at each iterate the BFGS algorithm maintains an approximation and updates it with each iteration while keeping some attractive properties namely a positive definite matrix satisfying the secant equation and is closest as possible to the last approximation. we form the problem of approximating the Inverse Hessian as following:

$$\min_{H} ||H - H_k||_F$$

$$s.t : H = H^T, \langle B, y_k \rangle = s_k$$

$$where : s_k = \overline{W}_t - \overline{W}_{t-1}, y_k = \nabla\phi(\overline{W}_t) - \nabla\phi(\overline{W}_{t-1})$$

and its unique solution given by

$$H_{t+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T : \rho_k = \frac{1}{\langle y_k, s_k \rangle}$$

Algorithm 2 BFGS

```
INPUT $\nabla\phi, \overline{W}_0$ convergence tolerance $\epsilon > 0$
PROCEDURE
  • Initial Approximation of the inverse hessian $H_0 = I$
  • While $||\nabla\phi(\overline{W}_t)|| > \epsilon$ :
      1. Calculate the descent direction $\bar{d}_t = -H_t \nabla\phi_t$
      2. Determine $a_t$ using a line search that satisfies Wolfe's conditions
      3. $\overline{W}_{t+1} = \overline{W}_t + \alpha_t \cdot \bar{d}_t$
      4. $s_k = \overline{W}_{t+1} - \overline{W}_t$ , $y_k = \nabla\phi(\overline{W}_t) - \nabla\phi(\overline{W}_{t-1})$ , $\rho_k = \frac{1}{\langle y_k, s_k \rangle}$
      5. $H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T$
```

4.3 Stochastic Broyden-Flecther-Goldfarb-Shanno Algorithm

We restrict our discussion to the family of convex functions and present the following stochastic variations

1. Stochastic BFGS

2. Nesterov Accelerated Stochastic BFGS

Both algorithms replace the line search by a gain schedule which iteratively decreases the step size and convexity of the function promises the restrictions on H are kept as with the original algorithm.

Algorithm 3 Stochastic BFGS

```
INPUT $\nabla\psi, S, \alpha_0 > 0, \tau \geq 0, \lambda > 0, T$

PROCEDURE

  • Initial Approximation of the inverse hessian $H_0 = I$
  • Initial guess $\overline{W} = \overline{0}$
  • While $t < T$ :
      1. Sample a mini-batch $B_t$
      2. Approximate the derivate by $g_t^1 = \frac{1}{|B_t|} \sum_{i=1}^{|B_t|} \nabla\psi(\overline{W}_t, B_t^i)$
      3. Set the descent direction $\overline{d} = \frac{-H_t g_t^1}{||H_t g_t^1||_2}$
      4. Determine $a_t = \frac{\tau}{\tau + t} \cdot \alpha_0$
      5. $\overline{W}_{t+1} = \overline{W}_t + \alpha_t \cdot \overline{d}_t$
      6. Calculate the another gradient $g_t^2 = \frac{1}{|B_t|} \sum_{i=1}^{|B_t|} \nabla\psi(\overline{W}_{t+1}, B_t^i)$
      7. $s_k = \overline{W}_{t+1} - \overline{W}_t$ , $y_k = g_t^2 - g_t^1 + \lambda s_k$ , $\rho_k = \frac{1}{\langle y_k, s_k \rangle}$
      8. $H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T$
```

```
**Algorithm 4** Nesterov Accelerated Stochastic BFGS
**INPUT** $\nabla\psi, S, \alpha_0 > 0, \tau \geq 0, \lambda > 0, \mu > 0T$
**PROCEDURE**
  • Initial Approximation of the inverse hessian $H_0 = I$
  • Initial guess $\overline{W} = \overline{0}$
  • Initial velcoity $\overline{V}_0 = \overline{0}$
  • While $t < T$ :
      1. Sample a mini-batch $B_t$
      2. Approximate the derivate by $g_t^1 = \frac{1}{|B_t|} \sum_{i=1}^{|B_t|} \nabla\psi(W_t + \mu\overline{V}_t, B_t^i)$
      3. Set the descent direction $\overline{d} = \frac{-H_t g_t^1}{||H_t g_t^1||_2}$
      4. Determine $a_t = \frac{\tau}{\tau + t} \cdot \alpha_0$
      5. Set the velocity $\overline{V}_t = \mu\overline{V}_{t-1} + \alpha_t \cdot \overline{d}_t$
      6. $\overline{W}_{t+1} = \overline{W}_t + \overline{V}_{t-1}$
      7. Calculate the another gradient $g_t^2 = \frac{1}{|B_t|} \sum_{i=1}^{|B_t|} \nabla\psi(W_{t+1}, B_t^i)$
      8. $s_k = \overline{W}_{t+1} - \Bigg(\overline{W}_t + \mu\overline{V}_t\Bigg)$ , $y_k = g_t^2 - g_t^1 + \lambda s_k$ , $\rho_k = \frac{1}{\langle y_k, s_k \rangle}$
      9. $H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T$
```

4.4 Convex Functions

We present a quick reminder of convex functions and few lemmas which prove our SVM learning problem is convex.

4.4.1 Definition

A function $\psi : \mathbb{R} \to \mathbb{R}$ is convex $\iff \forall x, y \in \mathbb{R}, \alpha \in [0,1] : \psi\Bigg(\alpha x + (1-\alpha) y\Bigg) \leq \alpha\psi(x) + (1-\alpha)\psi$

4.4.2 Lemma 1

Assume that $f : \mathbb{R}^d \to \mathbb{R}$ can be written as $f(W) = g\Bigg(\langle W, x \rangle + y\Bigg)$ for some $x \in \mathbb{R}^d, y \in \mathbb{R}, g : \mathbb{R} \to \mathbb{R}$ then convexity of g implies convexity of f

4.4.3 Lemma 2

Let $f_i : \mathbb{R}^d \to \mathbb{R} : i \in [1, n]$ be convex functions then $\forall i : \alpha_i \in \mathbb{R}^+ \implies g(x) = \sum_{1}^{n}$ $\alpha_i f_i(x)$ is convex

Part III
Results

4.5 Support Vector Machines

The soft SVM learning problem is a binary classification task where our linear predictor suffers losses with respect not only to the correct classification but also to a margin criterion.

1. $H = \Bigg\{ sign\Bigg(\langle W, X \rangle + b \Bigg) : W \in \mathbb{R}^d, b \in \mathbb{R} \Bigg\}$
2. $\psi(W, x, y) = max\big(0, 1 - y\langle W, x \rangle\big) + \frac{\lambda}{2} ||W||_2^2$
3. $\nabla\psi(W, x, y) = \mathbb{I}\big(1 > y\langle W, x \rangle\big) \cdot \big(-yx\big) + \lambda W$

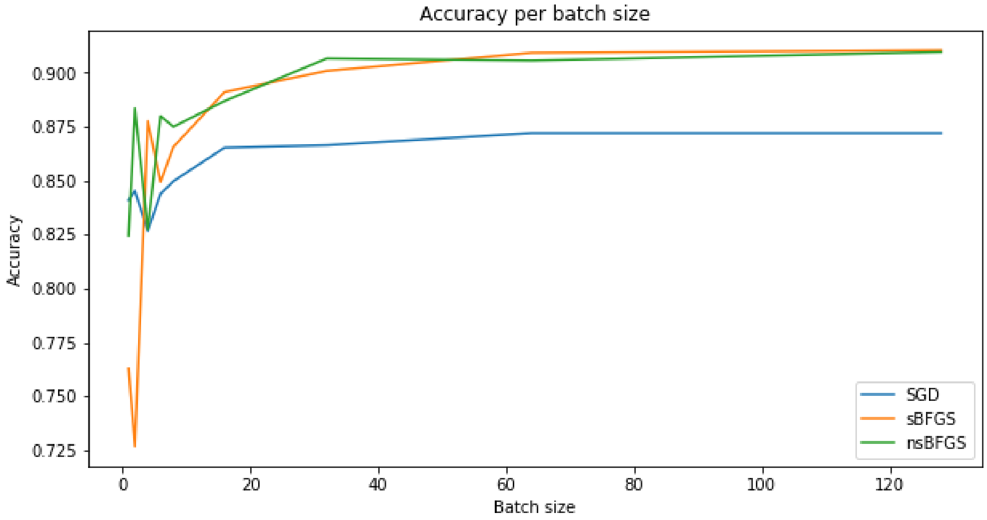
4.6 Data & Results

We show that both the stochastic variants of BFGS dominates the Stochastic Gradient Descent algorithm on synthetic data and a heart disease dataset where we desire to predict a patients medical condition by several measurements.

4.6.1 Synthetic Data

This is a classification task on data generated by scikit-learn which consists of 50,000 samples. each sample has 14 features of which only 11 are informative for the decision and is not linearly separable. on this dataset its clear that both the algorithms increasingly dominate SGD when the batch size increases and the hessian approximation becomes more accurate.

Figure 1: Synthetic dataset accuracy per batch size

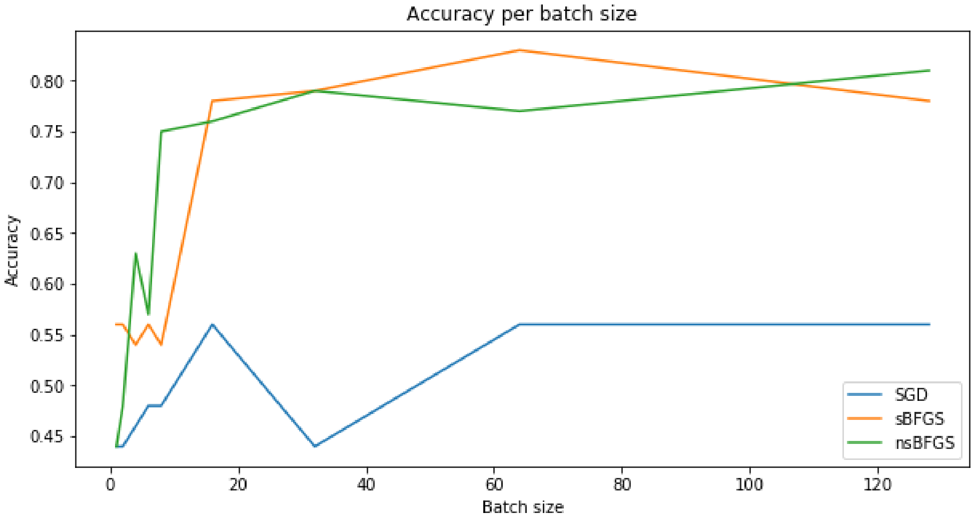


4.6.2 Heart disease data

We solve the problem of classifying heart disease patients based on their respective medical measurements by applying an SVM Predictor with all three algorithms: SGD, sBFGS, nBFGS

- age, sex, chest pain type (4 values), resting blood pressure, serum cholestoral in mg/dl, fasting blood sugar $>$ 120 mg/dl, resting electrocardiographic results (values 0,1,2), maximum heart rate achieved, exercise induced angina, oldpeak = ST depression induced by exercise relative to rest, the slope of the peak exercise ST segment number of major vessels (0-3) colored by flourosopy, thal: 3 = normal; 6 = fixed defect; 7 = reversable defect

Figure 2: Heart rate dataset accuracy per batch size



-