MATHEMATICS OF DATA

Marco Pietro Abrate, 292996

HOMEWORK 1

TRAINING DATA SET OF n POINTS

$(a_1, b_1) \dots (a_n, b_n)$

$a_i \in \mathbb{R}^p \qquad b_i \in \{-1; 1\}$

$$l_{MH}(x) = \frac{1}{2n} \sum_{i=1}^{n} q_i(x) \qquad \mu = 0$$

$$q_i(x) = \begin{cases} 0 & \text{if } b_i(a_i^T x) > 1+h \\ (1+h-b_i(a_i^T x))^2/4h & \text{if } |1-b_i(a_i^T x)| \le h \\ 1-b_i(a_i^T x) & \text{if } b_i(a_i^T x) < 1-h \end{cases}$$

$$f(x) = l_{MH}(x) + \frac{\lambda}{2}\|x\|^2 \qquad x \in \mathbb{R}^p$$

$$x^* \in \arg\min_x f(x)$$

PROBLEM 1:

$I_0, I_a, I_L \in \mathbb{R}^{n \times n} \qquad \tilde{A} = (b_1 a_1 \ \dots \ b_n a_n)^T \in \mathbb{R}^{n \times p}$

a) $$l_{MH}(x) = \frac{1}{2n}\left\{ \frac{1}{4h} \| I_a \overset{n\times 1}{\mathbb{1}}(1+h) - I_a \tilde{A} x \|^2 + 1 - \overset{1\times n}{\mathbb{1}} I_L \tilde{A} x \right\}$$

$$\nabla l_{MH}(x) = \frac{1}{2n}\left\{ \frac{1}{4h} \nabla\left((I_a \overset{n\times 1}{\mathbb{1}}(1+h) - I_a\tilde{A}x)^T (I_a \overset{n\times 1}{\mathbb{1}}(1+h) - I_a\tilde{A}x) \right) + \nabla\left(1 - \overset{1\times n}{\mathbb{1}} I_L \tilde{A} x\right) \right\}$$

$$= \frac{1}{2n}\left\{ \frac{1}{4h} \nabla\left((I_a \overset{n\times 1}{\mathbb{1}}(1+h))^T (I_a \overset{n\times 1}{\mathbb{1}}(1+h)) + (I_a\tilde{A}x)^T(I_a\tilde{A}x) - 2(I_a \overset{n\times 1}{\mathbb{1}}(1+h))^T (I_a\tilde{A}x) \right) + \right.$$

$$\left. + \nabla\left[- \overset{1\times n}{\mathbb{1}} I_L \tilde{A} x \right] \right\}$$

$$= \frac{1}{2n}\left\{ \frac{1}{4h} \nabla\left[(I_a\tilde{A}x)^T (I_a\tilde{A}x) \right] - \frac{1}{2h} \nabla\left[(I_a \overset{n\times 1}{\mathbb{1}}(1+h))^T (I_a\tilde{A}x) \right] - \nabla\left[\overset{1\times n}{\mathbb{1}} I_L \tilde{A} x \right] \right\} =$$

$$= \frac{1}{2n}\left\{ \frac{1}{4h} 2 (I_a\tilde{A})^T (I_a\tilde{A}x) - \frac{1}{2h}(1+h)\left[(I_a \overset{n\times 1}{\mathbb{1}})^T (I_a\tilde{A}) \right]^T - \left(\overset{1\times n}{\mathbb{1}} I_L \tilde{A} \right)^T \right\} =$$

$$= \frac{1}{4nh}\left\{\tilde{A}^T \underbrace{I_Q^T I_Q}_{=I_Q} \tilde{A}x - (1+h)\tilde{A}^T \underbrace{I_Q I_Q}_{=I_Q} \overset{nx1}{\mathbb{1}}\right\} - \frac{1}{2n}\tilde{A}^T I_L^T \overset{nx1}{\mathbb{1}} =$$

$$= \frac{1}{4nh}\tilde{A}^T I_Q\left\{\tilde{A}x - (1+h)\overset{nx1}{\mathbb{1}}\right\} - \frac{1}{2n}\tilde{A}^T I_L \overset{nx1}{\mathbb{1}}$$

$$\nabla f(x) = \nabla \ell_{MH}(x) + \frac{\lambda}{2}\nabla(\|x\|^2) = \nabla \ell_{MH}(x) + \lambda x$$

$\nabla f(x)$ IS L-LIPSCHITZ CONTINUOUS IF THERE EXISTS $L>0$ S.T.

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x-y\| \qquad \forall x, y \in \operatorname{dom} f$$

FOR THE LINEAR PART $LP(x) = -\frac{1}{2n}\tilde{A}^T I_L \overset{nx1}{\mathbb{1}}$

$$\|LP(x) - LP(y)\| = \left\| -\frac{1}{2n}\tilde{A}^T I_L \overset{nx1}{\mathbb{1}} + \frac{1}{2n}\tilde{A}^T I_L \overset{nx1}{\mathbb{1}} \right\| = 0 \quad \rightarrow L_{LP} = 0$$

FOR THE NON-LINEAR PART $NLP(x) = \frac{1}{4nh}\tilde{A}^T I_Q\left\{\tilde{A}x - (1+h)\overset{nx1}{\mathbb{1}}\right\} + \lambda x$

$$\|NLP(x) - NLP(y)\| = \left\| \frac{1}{4nh}\tilde{A}^T\tilde{A}x - \cancel{\frac{1}{4nh}\tilde{A}^T(1+h)\overset{nx1}{\mathbb{1}}} + \lambda x - \frac{1}{4nh}\tilde{A}^T\tilde{A}y + \right.$$

$$\left. + \cancel{\frac{1}{4nh}\tilde{A}^T(1+h)\overset{nx1}{\mathbb{1}}} - \lambda y \right\| = \left\| \frac{1}{4nh}\tilde{A}^T A(x-y) + \lambda(x-y) \right\| \leq$$

$$\leq \frac{1}{4nh}\|\tilde{A}^T\tilde{A}(x-y)\| + \lambda\|x-y\| \leq \left\{\frac{1}{4nh}\|\tilde{A}^T\|\|\tilde{A}\| + \lambda\right\}\|x-y\|$$

$$L_{NLP} = L = \lambda + \frac{1}{4nh}\|\tilde{A}^T\|\cdot\|\tilde{A}\| = \lambda + \frac{1}{4nh}\|A^T\|\cdot\|A\|$$

b) SUPPOSE $I_Q = \mathbb{I}$ AND LET'S CONSIDER g AS THE ELEMENTS OF $\nabla f(x)$ THAT DEPENDS ON x:

$$g(x) = \lambda x + \frac{1}{4nh}\tilde{A}^T\tilde{A}x \qquad \text{LET'S CALL } Q = \tilde{A}^T\tilde{A}\frac{1}{4nh}$$

HESSIAN MATRIX OF f: $H(f) = J(\nabla f(x))^T = J(g(x))^T$

$$g(x) = \begin{pmatrix} \lambda x_1 + Q_{1,2}x_2 + \dots + Q_{1,p}x_p \rightarrow g_1(x) \\ \vdots \\ \lambda x_p + Q_{p,2}x_1 + \dots + Q_{p,p}x_p \rightarrow g_p(x) \end{pmatrix}$$

$$J(g(x)) = \begin{pmatrix} \partial g_1/\partial x_1 & \cdots & \partial g_1/\partial x_p \\ \partial g_p/\partial x_1 & \cdots & \partial g_p/\partial x_p \end{pmatrix} =$$

$$= \begin{pmatrix} \lambda + Q_{1,1} & Q_{1,2} & \cdots & Q_{1,p} \\ \vdots & & & \vdots \\ Q_{p,1} & Q_{p,2} & \cdots & \lambda + Q_{p,p} \end{pmatrix} = \lambda \overset{p\times p}{\mathbb{I}} + \frac{1}{4nh}\tilde{A}^T\tilde{A}$$

$$J(g(x))^T = J(g(x)) = J(\nabla f(x)) = H(f(x)) = \lambda \overset{p\times p}{\mathbb{I}} + \frac{1}{4nh}\tilde{A}^T\tilde{A}$$

ALL THE PARTIAL DERIVATIVES OF ∇f DO NOT DEPENDS ON x, THUS THEY ARE ALL CONTINUOUS. THIS PROVES THAT f IS TWICE DIFFERENTIABLE FOR EVERY $x \in \text{dom } f$. ⚠ SUPPOSING $I_Q = \mathbb{I}$

c) $f(x)$ STRONGLY CONVEX IF $h(x)$ CONVEX |

$$h(x) = f(x) - \frac{M}{2}\|x\|^2$$

LET'S TRY $M = \lambda$ SO THAT $h(x) = \ell_{MH}(x)$

$$\nabla h(x) = \frac{1}{4nh}\tilde{A}^T I_Q\{\tilde{A}x - (1+h)\overset{n\times 1}{\mathbb{1}}\} - \frac{1}{2n}\tilde{A}^T I_L \overset{n\times 1}{\mathbb{1}}$$

$$H(h(x)) = J(\nabla h(x))^T = \frac{1}{4nh}\tilde{A}^T I_Q \tilde{A}$$

IF $H(h(x))$ POSITIVE SEMI-DEFINITE, THEN $h(x)$ CONVEX

SO, WE NEED TO SHOW THAT $z^T H(h(x)) z \geq 0 \quad \forall z \in \mathbb{R}^p$

LET'S WRITE $H = H(h(x)) = \frac{1}{4nh}\tilde{A}^T I_Q \tilde{A} = \frac{1}{4nh}\tilde{A}^T I_Q^T I_Q \tilde{A}$

$$z^T H z = \frac{1}{4nh} z^T \tilde{A}^T I_Q^T I_Q \tilde{A} z = \frac{1}{4nh}\|I_Q \tilde{A} z\|^2 \geq 0$$

$\longrightarrow h(x)$ CONVEX

$\longrightarrow f(x)$ λ-STRONGLY CONVEX

PROBLEM 2:

a)
Implementing the Gradient Descent (GD) algorithm, using as the search direction the additive inverse of the gradient of *f* multiplied by the step-size, the following results are observed, after 8000 iterations:

f(x) = 0.037009811 (~10^{-5} precision)
Error w.r.t. 0-1 loss: 0.029197080

Changing the step-size to 2/(L+ƛ) because of the strong convexity of the function and iterating the same number of times, results in a more precise solution:

f(x) = 0.037003440 (~10^{-7} precision)
Error w.r.t. 0-1 loss: 0.029197080

b)
Accelerated Gradient Descent (AGD) can be used to ensure faster convergence. With 4000 iterations the minimum is reached, even if with an oscillatory behaviour:

f(x) = 0.03700341 (>10^{-9} precision)
Error w.r.t. 0-1 loss: 0.029197080

Exploiting the strong convexity, and halving the number of iterations, the same result is obtained.

c)
The step-size can be adapted to the local geometry of the function at each iteration, updating the Lipschitz constant. This is done because the global constant does not contain useful information about the local geometry of the function, which can be discovered through Line Search at each iteration, while approaching the stationary point. This technique has been implemented in both the GD and AGD algorithms, performing 450 and 400 iterations, respectively. The minimum is reached in both cases:

f(x) = 0.03700341 (>10^{-9} precision)
Error w.r.t. 0-1 loss: 0.029197080

d)
As stated before, the AGD (as well as the LSAGD) method is non-monotonic. This means that at some iterations the direction used to update *x* can move away from the minimum of the function. To prevent such behaviour an Adaptive Restart strategy can be applied to both the AGD and LSAGD algorithms at each iteration, checking improvements. Executing 500 and 100 iterations respectively, the stationary point is reached:

f(x) = 0.03700341 (>10^{-9} precision)
Error w.r.t. 0-1 loss: 0.029197080

PROBLEM 3:

a)
Using both the Hessian and the Gradient of the objective function *f* to compute the optimal direction, the Newton method converge to the minimum in just 20 iterations.

b)
Considering that evaluating the Hessian at each iteration is computationally expensive, it can be approximated using the quasi-Newton method. In this case 200 iterations assure optimality, even if 140 are sufficient, considering the results:

iteration = 140
f(x) = 0.03700341 ($>10^{-9}$ precision)
Error w.r.t. 0-1 loss: 0.029197080

PROBLEM 4:

$$f(x) = \frac{1}{n}\sum_{i=1}^{n}\{\underbrace{\frac{1}{2}c_i(x) + \frac{\lambda}{2}\|x\|^2}_{f_i(x)}\}$$

$$\nabla f_i(x) = \lambda x + \mathbb{1}_{\{|1-b_i a_i^T x| \le h\}}\frac{1}{4h}a_i(a_i^T x - b_i(1+h)) - \mathbb{1}_{\{b_i a_i^T x < 1-h\}}\frac{1}{2}b_i a_i$$

PICK $j \in \{1,\dots,n\}$ UNIFORMLY AT RANDOM

a) SHOW THAT $\nabla f_j(x)$ IS AN UNBIASED ESTIMATOR OF $\nabla f(x)$

$\nabla f(x) = \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x)$, WE NEED TO SHOW THAT $\mathbb{E}_j[\nabla f_j(x)] = \nabla f(x)$

$$\mathbb{E}_j[\nabla f_j(x)] = \mathbb{E}_j[\mathbb{1}_{\{j=i\}}\nabla f_i] = \sum_{i=1}^{n}\mathbb{P}(j=i)\nabla f_i(x) = \cdot$$

$$= \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x) = \nabla f(x)$$

WHY $\nabla f_j(x)$ IS LIPSCHITZ CONTINUOUS WITH $L(f_j) = \frac{1}{4h}\|a_j\|^2 + \lambda$?

SINCE THE LINEAR PART OF $\nabla f_j(x)$ HAS $L = 0$, WE CAN CONSIDER JUST THE NON-LINEAR PART SUBSTITUTING THE $\mathbb{1}_{\{\text{predicate}\}}$ WITH 1:

$$NL_j(x) = \lambda x + \frac{1}{4h}a_j(a_j^T x - b_j(1+h))$$

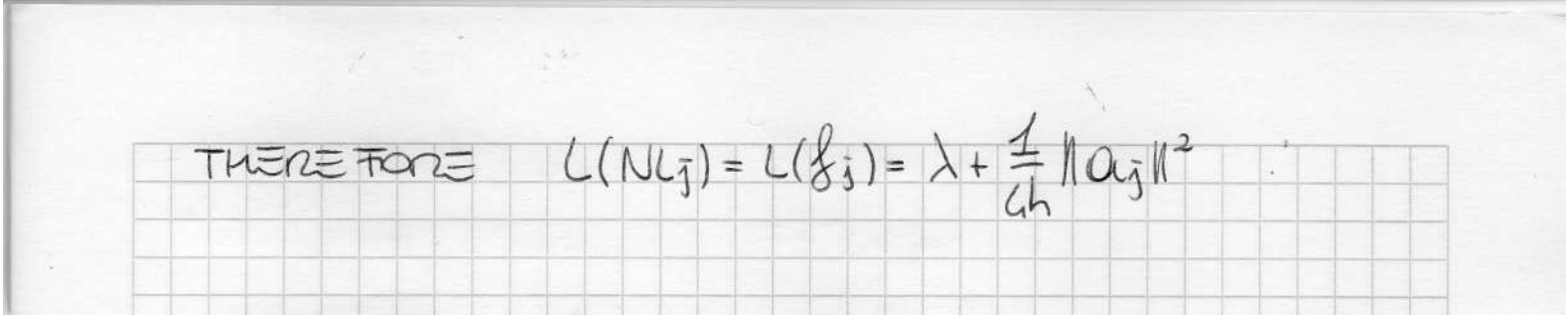
THEN, WE CAN UPPER BOUND

$$\|NL_j(x) - NL_j(y)\| = \|\lambda x + \frac{1}{4h}a_j a_j^T x - \cancel{\frac{1}{4h}a_j b_j(1+h)} - \lambda y +$$

$$- \frac{1}{4h}a_j a_j^T y + \cancel{\frac{1}{4h}a_j b_j(1+h)}\| = \|\lambda(x-y) + \frac{1}{4h}a_j a_j^T(x-y)\| \le$$

$$\le \|\lambda(x-y)\| + \|\frac{1}{4h}a_j a_j^T(x-y)\| = \lambda\|x-y\| + \frac{1}{4h}\|a_j a_j^T(x-y)\| \le$$

$$\le \lambda\|x-y\| + \frac{1}{4h}\|a_j a_j^T\|\cdot\|x-y\| \le \{\lambda + \frac{1}{4h}\|a_j\|^2\}\|x-y\|$$



b)
Implementing the Stochastic Gradient Descent (SGD) algorithm, it can be seen that a precision of ~10^{-1} is reached very quickly. After that point the convergence does not seem to get better, even if iterating for 5 epochs:

> *f(x)* = 0.103201462 (~10^{-1} precision)
> Error w.r.t. 0-1 loss: 0.080291971

The problem can be that the step-size is taken as 1/*k*, where *k* is the number of the current iteration. Since α is decreasing each step, at some point the algorithm is moving very slowly towards the minimum. Fixing the step-size to 1/*n* from the *n-th* iteration, the method seems to work better, having as final results:

> *f(x)* = 0.057310298 (~10^{-2} precision)
> Error w.r.t. 0-1 loss: 0.036496350

c)
Changing the step-size to 1/(16 * Lmax) and averaging iterates to reduce the oscillation effect, better results are obtained with the same number of epochs:

> *f(x)* = 0.053582986 (~10^{-2} precision)
> Error w.r.t. 0-1 loss: 0.036496350

d)
Finally, implementing a variance reduction technique (i.e. computing the real gradient from time to time to adjust the direction) to ensure convergence and setting the step-size to a predetermined value to converge faster, a solution very close to the minimum is reached after 0.3 epochs:

> *f(x)* = 0.037578456 (~10^{-3} precision)
> Error w.r.t. 0-1 loss: 0.021897810

In Figure 1 and Figure 2 the results of every methods are shown. Note that the scales of the two figures are different.

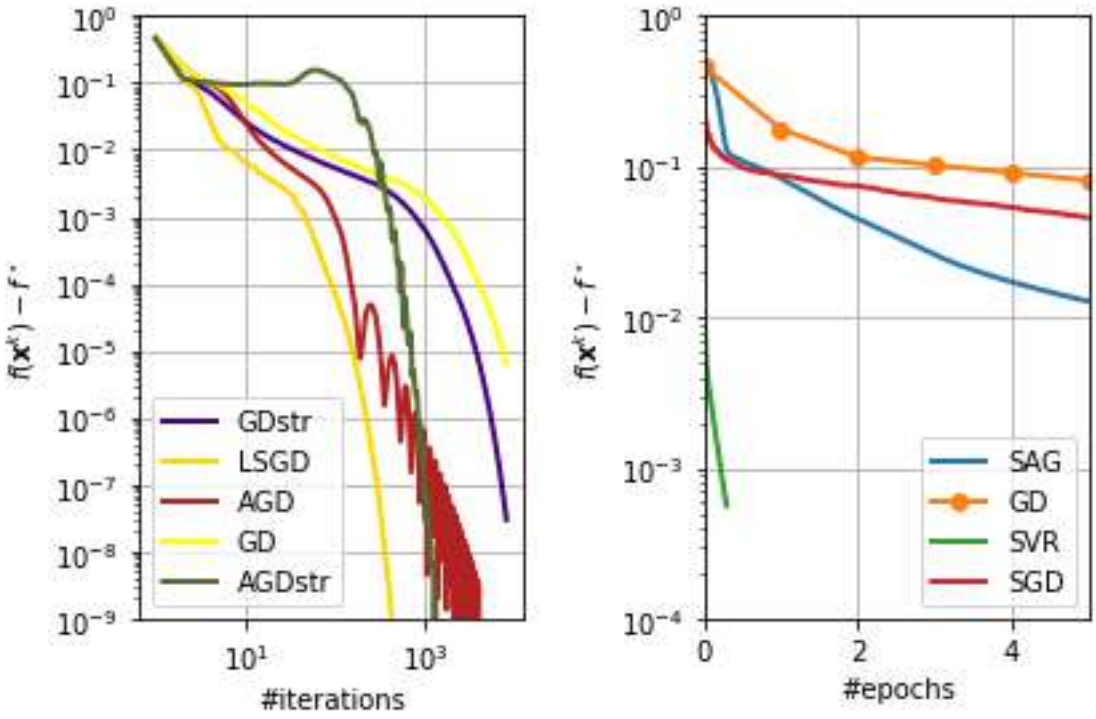


Figure 1: Results for methods described in points 2a, 2b, 2c, 4a, 4b, 4c

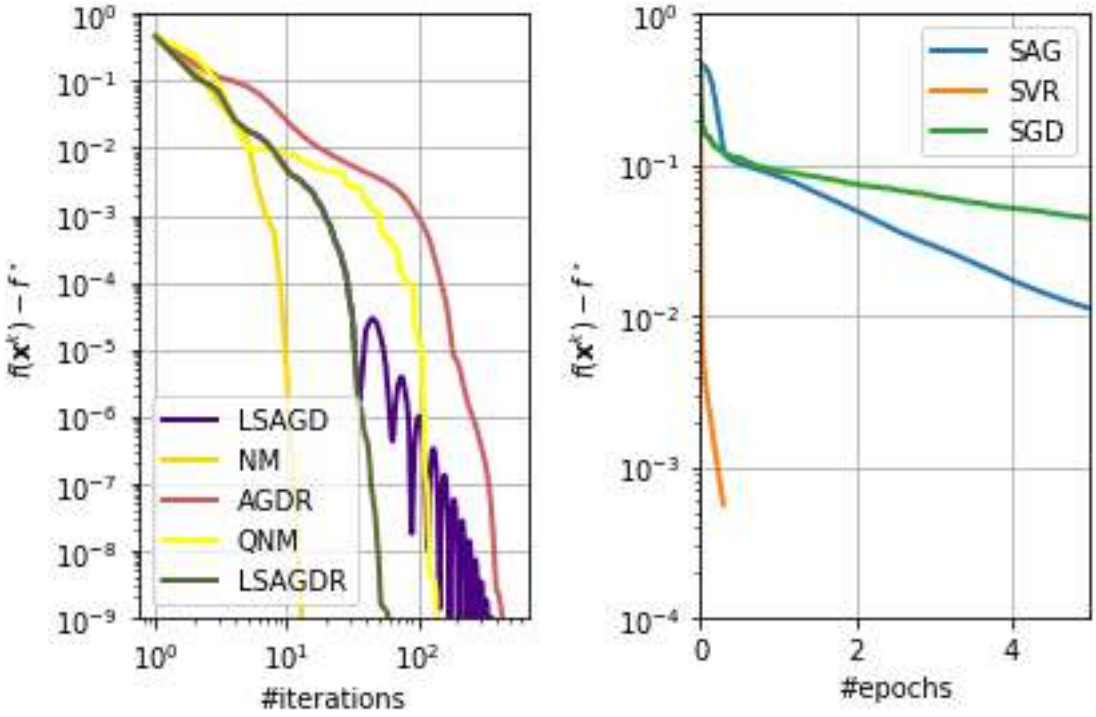


Figure 2: Results for methods described in points 2c, 2d, 3a, 3b, 4b, 4c, 4d