Not completed
Proximal gradient descent
& ILP.

Principles of Machine Learning

Exercises

Victor Verreet *victor.verreet@cs.kuleuven.be*
Laurens Devos *laurens.devos@cs.kuleuven.be*

Fall, 2021

Exercise Session 3: Optimization and ILP

3.1 Huber Loss Gradient

Huber loss is defined as

$$\mathcal{L}_{H,\delta}(y,\hat{y}) = \begin{cases} \frac{1}{2}(y-\hat{y})^2 & \text{for } |y-\hat{y}| \leq \delta, \\ \delta|y-\hat{y}| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

a. Draw this curve.

b. Huber is made up of two other loss functions we've seen in the course. Why would you use Huber instead the other losses?

c. Compute the gradient of the Huber loss. Is this function C^1 continuous? Is it C^2 continuous? Do the answers to the previous two questions limit the possible optimization methods?

d. Huber loss has a parameter δ. Which value would you choose for this parameter?

3.2 Squared Loss with Squared Regularization

a. Write down the loss function $\mathcal{L}(y, \boldsymbol{X}w)$ for a linear regression problem with L2 regularization with input matrix $\boldsymbol{X}$ containing the instances in its rows, a column vector with weights w, and a column vector y with target values.

b. Compute the derivative of this loss function.[1]

3.3 Non-Differentiable Functions: Subgradients

The following definition for convexity was given in the lecture: for any $x, y \in \text{dom}(f)$ and $\alpha \in [0,1]$:

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y).$$

If a function f is differentiable, f is convex if: [2]

Fix x

$$f(y) \geq f(x) + \nabla f(x)^T(y-x) \quad \text{for all } x, y \in \text{dom}(f).$$

a. Visually show what these definitions mean for $f(x) = x^2$. Explain how what you have drawn relates to the definition.

[1] The *Dive Into Deep Learning* book has a great chapter on multivariate calculus.
[2] A great resource for convex optimization is: `http://www.stat.cmu.edu/~ryantibs/convexopt`.

b. A **subgradient** of a convex function (also non-differentiable) at a value x is any vector $g \in \mathbb{R}^n$ such that:

$$f(y) \geq f(x) + g^T(y - x) \quad \text{for all } y \in \text{dom}(f).$$

The **subgradient set** or **subdifferential** is defined as all vectors g for which the above holds:

$$\partial f(x) = \{g \mid g \text{ is a subgradient of } f \text{ at } x\}.$$

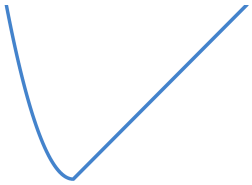
Find the subgradient set $\partial f(x)$ for all $x \in \mathbb{R}$ of the function:

$$f(x) = \begin{cases} x^2, & x \leq 0, \\ x, & x > 0. \end{cases}$$

$f(y) \geq f(x) + g^T(y-x)$

$g = 0$

$\Rightarrow f(y) \geq f(x)$



A convex function is minimized at a value x^* if $0 \in \partial f(x^*)$. Can you verify this in the above example?

3.4 First-Order Methods Using Parabolas

We've seen that a second-order approximation of a differentiable function can be used to optimize a function (e.g. Newton-Raphson). The general format of such second-order approximation in a is the following:

$$f(x) \approx f(a) + \nabla f(a)^T(x - a) + \frac{1}{2}(x - a)^T \boldsymbol{H}(a)(x - a),$$

where $\boldsymbol{H}(a)$ is the Hessian of f evaluated in a. We can turn this into a first-order approximation by replacing the Hessian by the scaled identity matrix $\frac{1}{\rho}\boldsymbol{I}$, i.e., we still use a parabolic approximation, but we limit the family of possible approximating functions to the parabolas for which the coefficients for the second-order terms are fixed. You can view ρ as a scaling factor of the parabola. We get:

$$\begin{aligned} f(x) &\approx f(a) + \nabla f(a)^T(x - a) + \frac{1}{2\rho}(x - a)^T \boldsymbol{I}(x - a) \\ &= f(a) + \nabla f(a)^T(x - a) + \frac{1}{2\rho}\|x - a\|_2^2. \end{aligned}$$

Show that:

$$\arg\min_x f(a) + \nabla f(a)^T(x - a) + \frac{1}{2\rho}\|x - a\|_2^2$$

is equivalent to

$$\arg\min_x \frac{1}{2\rho}\|(x - (a - \rho\nabla f(a))\|_2^2.$$

Note that the above formula can be solved exactly: simply take x equal to $a - \rho\nabla f(a)$, which corresponds to the **update step of gradient descent**, with ρ the step size or learning rate.

3.5 Proximal Methods: Finding Weights for LASSO

Proximal gradient descent is used to optimize convex functions of the form $f(x) = g(x) + h(x)$, where f is non-differentiable, but it can be decomposed into a convex differentiable g and a convex non-differentiable h. We cannot apply the gradient update step from the previous exercise to $f(x)$ in its

entirety because it is non-differentiable. However, we can use it for $g(x)$. Consider the following update step, where $g(x)$ is replaced by its quadratic approximation $\tilde{g}(x)$ and $h(x)$ is left unchanged:

$$\begin{aligned} x^{(k+1)} &= \arg\min_z \tilde{g}(z) + h(z) \\ &= \arg\min_z g(x^{(k)}) + \nabla g(x^{(k)})^T (z - x^{(k)}) + \frac{1}{2\rho}\|z - x^{(k)}\|_2^2 + h(z) \\ &= \arg\min_z \frac{1}{2\rho} \underbrace{\|z - (x^{(k)} - \rho\,\nabla g(x^{(k)}))\|_2^2}_{(1)} + \underbrace{h(z)}_{(2)} \end{aligned} \tag{1}$$

We can interpret (1) as: *choose an update z as close to the gradient update $x - \rho\,\nabla g(x)$*, and (2) as: *but also make $h(z)$ small.* Proximal gradient descent uses a proximal mapping to optimize functions of this form. We won't be looking at the general formulation, rather, we will look at an example.

Proximal gradient descent can be used to optimize the LASSO criterion for linear regression. Linear regression with L1 regularization (LASSO) is defined as:

$$\begin{aligned} &\frac{1}{2}\sum_i (y_i - \boldsymbol{X}_{i*}w)^2 + \lambda\|w\|_1 \\ &= \frac{1}{2}\|y - \boldsymbol{X}w\|_2^2 + \lambda\|w\|_1. \end{aligned} \tag{2}$$

The regularization term is not differentiable at zero, so standard gradient descent models aren't applicable. Luckily, the problem can be decomposed into $f = g + h$, where g is the convex differentiable part $\sum_i (y_i - \boldsymbol{X}_{i*}w)^2$, and h is the convex, non-differentiable part $\lambda\|w\|_1$.

We will use the proximal mapping of the L1 norm. But what exactly is a proximal mapping? The general definition of a *proximal mapping* or *proximal operator* of a convex function ϕ is:

$$\text{prox}_\phi(v) = \arg\min_z \left[\phi(z) + \frac{1}{2}\|z - v\|_2^2\right]$$

Complete books have been written on the subject of proximal methods, so the full details are beyond the scope of this course. What you need to know is that $\text{prox}_\phi(v)$ is well-defined: it has a unique solution because $\|\cdot\|_2$ is strictly convex and we required ϕ to be convex. The sum is therefore also strictly convex, so a unique solution exists. Intuitively, you can view the proximal mapping as a projection to a point in the neighborhood of v that is (1) close to v and (2) minimizes ϕ.

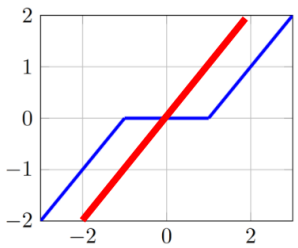
Replacing ϕ in the general definition of a proximal mapping with the L1 norm gives (with weight $\lambda > 0$):

$$\text{prox}_{\lambda\|\cdot\|_1}(v) = \arg\min_z \left[\lambda\|z\|_1 + \frac{1}{2}\|z - v\|_2^2\right] \tag{3}$$

a. Show that the above is minimized for:

$$\begin{aligned} z^* &= S_\lambda(v) \\ [S_\lambda(v)]_i &= \begin{cases} v_i - \lambda & \text{if } v_i > \lambda \\ 0 & \text{if } -\lambda \le v_i \le \lambda \\ v_i + \lambda & \text{if } v_i < -\lambda \end{cases} \end{aligned} \tag{4}$$

Remember that a convex function ρ is minimized for an input z^* when $0 \in \partial\rho(z^*)$. The S_λ function is called the **soft-thresholding operator**:



b. Show that, when $\boldsymbol{X}$ is taken to be the identity matrix $\boldsymbol{I}$, LASSO in Equation 2 is equivalent to $\text{prox}_{\lambda\|\cdot\|_1}$ from Equation 3.

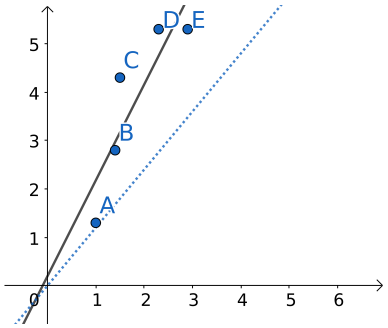
c. Rewrite the update step in Equation 1 for LASSO (Equation 2). Show that this is equivalent to:

$$\begin{aligned} w^{(k+1)} &= S_{\lambda\rho}\left(w^{(k)} - \rho\nabla g(w^{(k)})\right) \\ &= S_{\lambda\rho}\left(w^{(k)} + \rho\boldsymbol{X}^T(y - \boldsymbol{X}w^{(k)})\right), \end{aligned}$$

where $\nabla g(w) = -\boldsymbol{X}^T(y - \boldsymbol{X}w)$ (see previous exercise). This is the updating step of the **iterative soft-thresholding algorithm (ISTA)**.

d. Given the dataset below, $\lambda = 0.2$, learning rate $\rho = 0.01$, and $w^{(0)} = [0.0; 0.5]$, compute an update step using iterative soft thresholding. Compute the objective function before and after applying the soft-thresholding operator. Is the objective function minimized?

Feel free to use Python, Matlab, Julia, etc. for this exercise. This also allows you to execute multiple iterations.



	x	y
A	1.0	1.3
B	1.4	2.8
C	1.5	4.3
D	2.3	5.3
E	2.9	5.3

3.6 Theta Subsumption

Order these clauses according to theta subsumption. Which clauses are reduced? What are the reductions of the non-reduced clauses?

1. $f(a, b) \leftarrow m(a), m(b), p(a, b)$.
2. $f(X, Y) \leftarrow m(X), p(X, Y)$.
3. $f(X, X) \leftarrow m(X), p(X, Y), p(X, Z)$.
4. $f(X, X) \leftarrow m(X), p(X, X), m(X)$.
5. $f(X, X) \leftarrow m(X), p(X, X)$.
6. $f(X, X) \leftarrow m(X), p(X, Y)$.

3.7 Least General Generalization

Compute the LGG of the following literals and clauses.

a. $m(a, c(a, nil))$
$m(b, c(c, nil))$

b. $add(s(s(0)), s(s(0)), s(s(s(s(0)))))$
$add(s(0), s(0), s(s(0)))$

c. $m(X, c(X, Y)) \leftarrow c(X), list(Y)$.
$m(X, c(Y, Z)) \leftarrow c(X), c(Y), list(Z), m(X, Z)$.

3.8 Subsumption and Implication

Show that if $A \leq_\theta B$ then $A \implies B$ holds for any clauses A and B.

3.9 Cyclic Clauses

Define for every $n > 0$ a clause γ_n as

$$a \impliedby \bigwedge_{i=0}^{n-1} c(X_i, X_{i+1})$$

where we define $X_n = X_0$. Give the LGG of two clauses γ_n and γ_m for general n and m and prove this result.

3.10 Inverse Resolution

Find an inverse resolution path that derives the clause

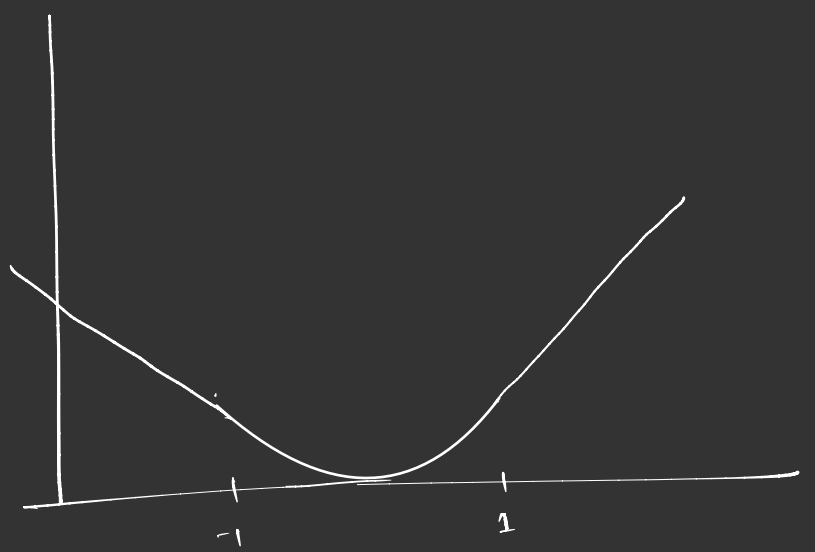
$$\text{daughter}(X, Y) \impliedby \text{parent}(Y, X), \text{female}(X)$$

from the observed facts

$$\begin{aligned} \text{daughter}(\text{lisa}, \text{marge}) \\ \text{parent}(\text{marge}, \text{lisa}) \\ \text{female}(\text{lisa}) \end{aligned}$$

① $L_{H,\delta}(y,\hat{y}) = \begin{cases} \frac{1}{2}(y-\hat{y})^2 & \text{for } |y-\hat{y}| \le \delta \\ \delta|y-\hat{y}| - \frac{1}{2}\delta^2 & \text{oth} \end{cases}$

Let $\delta = 1$

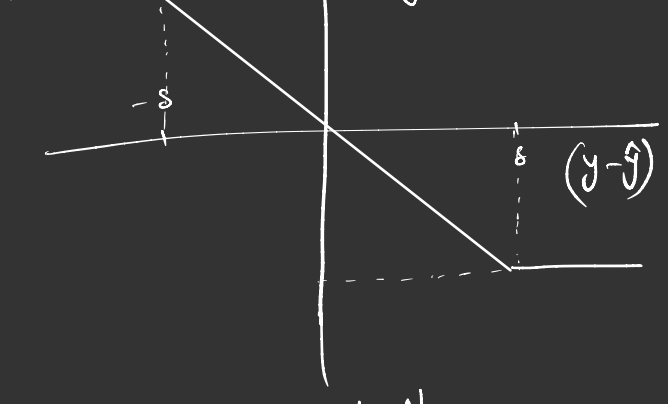


Huber Loss has robustness like L1 and also differentiable at 0.

$\frac{\partial L}{\partial \hat{y}} = -(y-\hat{y}) \quad \text{if } |(y-\hat{y})| \le \delta$

$\frac{\partial L}{\partial \hat{y}} = \begin{cases} -\delta & \text{if } (y-\hat{y}) > \delta \\ \delta & \text{if } (y-\hat{y}) < \delta \end{cases}$

$\frac{\partial L}{\partial \hat{y}} = \begin{cases} \delta & \text{if } (y-\hat{y}) < \delta \\ -(y-\hat{y}) & \text{if } |y-\hat{y}| \le \delta \\ -\delta & \text{if } (y-\hat{y}) > \delta \end{cases}$



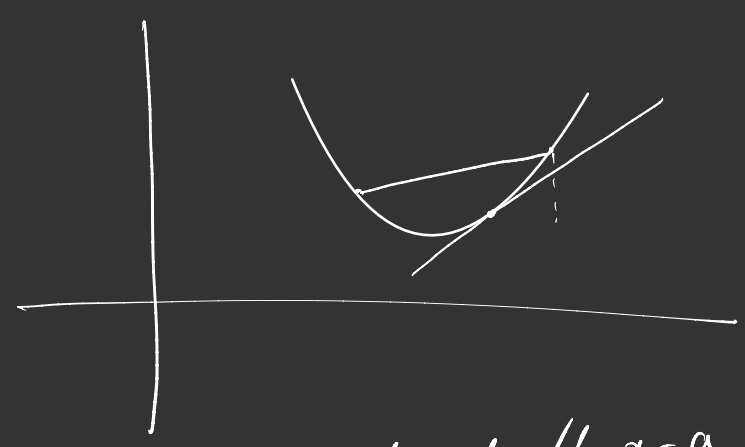
$\frac{\partial^2 L}{\partial \hat{y}^2} = \begin{cases} 0 & \text{if } |y-\hat{y}| \ge \delta \\ 1 & |y-\hat{y}| \le \delta \\ 0 & \end{cases}$

3.2 $L(y, Xw) = \frac{1}{2}\|y - Xw\|_2^2 + \lambda \|w\|^2$

$\frac{\partial L}{\partial w} = \begin{bmatrix} -2 \cdot \frac{1}{2}(y_i - x_{i1}w_i) \cdot x_{i1} \\ \vdots \\ -2 \cdot \frac{1}{2}(y_n - x_{n1}w_n) x_{n1} \end{bmatrix} + 2\lambda w$

$= X^T(Xw - y) + 2\lambda w$

$\frac{f(y)-f(x)}{y-x}$



Eqn of a Tangent at pt. $x = a$

$T(x) = f(a) + \nabla f(a)(x-a)$

$\forall x \in X$

$T(x) \le f(x)$

Tangent will be under the curve.

$f(y) = y^2$

$x^2 + 2x(y-x)$

$2xy - x^2$

$f(y) - 2xy + x^2 \ge 0$

$\Rightarrow (x-y)^2 \ge 0$

$y^2 \ge x^2$

b) $\partial f(x) = \begin{cases} 2x & x < 0 \\ [0,1] & x = 0 \\ 1 & x > 0 \end{cases}$

3.4

$$g(x) = f(a) + \nabla f(a)^T (x-a) + \frac{1}{2\rho}\|x-a\|_2^2$$

$$\frac{\partial g}{\partial x} = \nabla f(a)^T + \frac{1}{\rho}(x-a) = 0$$

$$\Rightarrow \quad x = a - \rho \nabla f(a)^T$$

$$x = a - \rho \nabla f(a)^T$$

$$\text{Prox}_\phi(v) = \underset{z}{\text{argmin}} \left(\phi(z) + \frac{1}{2}\|z-v\|^2 \right)$$

$$r(v) = \text{Prox}_{\lambda\|\cdot\|_1}(v) = \underset{z}{\text{argmin}} \left(\lambda\|z\|_1 + \frac{1}{2}\|z-v\|^2 \right)$$

If $z > 0$, $z = v - \lambda$, $v > \lambda$

$z = 0$ $\quad -\lambda \le v \le \lambda$

$z = v + \lambda$ $\quad v < -\lambda$

$$\underset{x}{\text{argmin}} \quad f(a) + \nabla f(a)^T (x-a) + \frac{1}{2\rho}\|x-a\|^2$$

$$\underset{x}{\text{argmin}} \sum \partial_i f(a)(x_i - a_i) + \frac{1}{2\rho}\sum (x_i - a_i)^2$$

$$\underset{x}{\text{argmin}} \frac{1}{2\rho}\sum \left(2\rho x_i \partial_i f(a) - 2\rho a_i \partial_i f(a) + x_i^2 - 2x_i a_i + a_i^2 \right)$$

$$= \underset{x}{\text{argmin}} \frac{1}{2\rho}\sum x_i^2 + 2\rho x_i \partial_i f(a) - 2\rho a_i \partial_i f(a) + a^2 - 2x_i a_i + \left(\rho \partial_i f(a)\right)^2 - \left(\rho \partial_i f(a)\right)^2$$

$$= \underset{x}{\text{argmin}} \frac{1}{2\rho}\sum \left\{ x_i^2 - 2x_i\left(a_i - \rho \partial_i f(a)\right) + \left(a_i - \rho \partial_i f(a)\right)^2 \right\}$$

$$= \underset{x}{\text{argmin}} \frac{1}{2\rho}\sum \left[x_i - \left(a_i - \rho \partial_i f(a)\right) \right]^2$$

$$= \underset{x}{\text{argmin}} \frac{1}{2\rho}\|x - (a - \rho \nabla f(a))\|_2^2$$

$$x^{k+1} = \underset{z}{\text{argmin}}\ \tilde{g}(z) + h(z)$$

$$= \underset{z}{\text{argmin}} \left(g(x^k) + \nabla g(x^k)^T (z - x^k) + \frac{1}{2\rho}\|z - x^k\|_2^2 \right) + h(z)$$

$$= \underset{z}{\text{argmin}} \left\| z - \left(x^k - \rho \nabla g(x^k)\right)^2 \right\|_2^2 + h(z)$$

$f = g + h$,

$$\rho(v) = \underset{z}{\text{argmin}} \left(\lambda\|z\|_1 + \frac{1}{2}\|z-v\|_2^2 \right)$$

$$o(z)$$

$$\partial o(z) = \left\{ \lambda s + (z - v) \mid s \in \partial(\|z\|_1) \right\}$$

$$s = \begin{cases} -1 & \text{if } z < 0 \\ [-1, 1] & \text{if } z = 0 \\ 1 & \text{if } z > 0 \end{cases}$$

If $v > \lambda > 0$, $v - \lambda s > 0$

$\lambda s + z - v = 0$ $\qquad -3 < -2$

$\Rightarrow z = v - \lambda s$ $\qquad -3 + 2$

$s = 1$, $\quad z = v - \lambda$,

If $v < -\lambda$, $z = v - \lambda s < -\lambda - \lambda s < 0$

$s = -1$

$z = v + \lambda$

If $-\lambda \le v \le \lambda$

$v - \lambda s = 0$ for $s = \frac{v}{\lambda} \in [-1, 1]$

$\therefore z = 0$

$$\frac{1}{2}\|y-Xw\|_2^2+\lambda\|w\|_1$$

$$=\frac{1}{2}\|y-w\|_2^2+\lambda\|w\|_1$$

$$w^{k+1}=\operatorname{argmin}\left(\lambda\|w\|_1+\frac{1}{2}\|y-w\|_2^2\right)$$

$$w^{k+1}=S_{\lambda\rho}\left(w^k-\rho\nabla g(w^k)\right)$$

$$=\operatorname*{argmin}_x\left(\lambda\|z\|_1+\frac{1}{2}\|z-\left(w^k-\rho\nabla g(w^k)\right)\|_2^2\right)$$

$$\nabla g(w)=-X^T(y-Xw)$$

$$w_{0,0}=\begin{pmatrix}0.0\\0.5\end{pmatrix},\ \lambda=0.2,\ \rho=0.01$$

---

① $f(a,b)\leftarrow m(a), m(b), p(a,b)$

$2\le\theta 1$