Convex Optimization for Machine Learning and Computer Vision

Lecture: Dr. Tao Wu
Exercises: Zhenzhang Ye
Winter Semester 2019/20

Computer Vision Group
Institut für Informatik
Technische Universität München

Weekly Exercises 6

Room: 02.09.023
Wednesday, 04.12.2019, 12:15-14:00
Submission deadline: Monday, 02.12.2019, 16:15, Room 02.09.023

Proximal operator (10+6 Points)

Exercise 1 (4 Points)**.** Assume function $J : \mathbb{R}^n \to \mathbb{R}$ is convex and subdifferentiable on its domain. Show that u^* minimizes J if and only if $u^* = \text{prox}_J(u^*)$.

Solution. If u^* minimizes J, we have $J(u) \geq J(u^*)$ for any $u \in \text{domJ}$. Therefore,

$$J(u) + \frac{1}{2}\|u - u^*\|_2^2 \geq J(u^*) = J(u^*) + \frac{1}{2}||u^* - u^*||_2^2$$

for any u, which means $u^* = \text{prox}_J(u^*)$.
Conversely, as J is a convex function, a point $u = \text{prox}_J(u^*)$ if and only if

$$0 \in \partial J(u) + (u - u^*)$$

Replace u with u^*, we can get the optimality condition. Since J is convex, we know that u^* is the minimizer.

Exercise 2 (4 Points)**.** Prove following properties of proximal operator:

- If $J(u) = \alpha f(u) + b$, with $\alpha > 0$, then $\text{prox}_{\lambda J}(v) = \text{prox}_{\alpha\lambda f}(v)$.
- If $J(u) = f(Qu)$, where Q is an orthogonal matrix, then $\text{prox}_{\lambda J}(v) = Q^\top \text{prox}_{\lambda f}(Qv)$

Solution.

- $$\begin{aligned}\text{prox}_{\lambda J}(v) &= \text{argmin}_u\, J(u) + \frac{1}{2\lambda}||u - v||^2 \\ &= \text{argmin}_u\, \alpha f(u) + b + \frac{1}{2\lambda}||u - v||^2 \\ &= \text{argmin}_u\, \alpha(f(u) + \frac{1}{2\lambda\alpha}||u - v||^2) \\ &= \text{argmin}_u\, f(u) + \frac{1}{2\lambda\alpha}||u - v||^2 \\ &= \text{prox}_{\alpha\lambda f}(v)\end{aligned}$$

- $$\begin{aligned}\mathrm{prox}_{\lambda J}(v) &= \mathrm{argmin}_u\, J(u) + \frac{1}{2\lambda}||u-v||^2 \\ &= \mathrm{argmin}_u\, f(Qu) + \frac{1}{2\lambda}||u-v||^2 \\ &= \mathrm{argmin}_u\, f(Qu) + \frac{1}{2\lambda}||Qu-Qv||^2 \\ &\overset{t=Qu}{=} Q^\top\, \mathrm{argmin}_t\, f(t) + \frac{1}{2\lambda}||t-Qv||^2 \\ &= Q^\top \mathrm{prox}_{\lambda f}(Qv)\end{aligned}$$

Exercise 3 (4 Points)**.** Show that the ℓ_1-norm proximal operator of $v \in \mathbb{R}^n$ is given as

$$\mathrm{prox}_{\lambda\|\cdot\|_1}(v) = u \in \mathbb{R}^n, \quad u_i := \begin{cases} v_i + \lambda & \text{if } v_i < -\lambda \\ 0 & \text{if } v_i \in [-\lambda, \lambda] \\ v_i - \lambda & \text{if } v_i > \lambda. \end{cases}$$

Solution. We begin reformulating the optimality condition

$$0 \in \partial \left(\frac{1}{2\lambda}(u_i - v_i)^2 + |u_i| \right)$$

of the optimal u_i

$$0 = \frac{1}{\lambda}(u_i - v_i) + p, \quad p \in \partial|u_i| := \begin{cases} -1 & \text{if } u_i < 0 \\ [-1,1] & \text{if } u_i = 0 \\ 1 & \text{if } u_i > 0 \end{cases}$$

$$v_i \in u_i + \begin{cases} -\lambda & \text{if } u_i < 0 \\ [-\lambda,\lambda] & \text{if } u_i = 0 \\ \lambda & \text{if } u_i > 0. \end{cases}$$

Recall that we are looking for a u_i that satisfies the condition above given a fixed v_i. We distinguish the following cases:

1. Assume $v_i \in [-\lambda, \lambda]$. Choosing $u_i := 0$ satisfies the condition above.

2. Assume $v_i > \lambda$. Choosing $u_i := v_i - \lambda$ again satisfies the condition.

3. Assume $v_i < -\lambda$. Choosing $u_i := v_i + \lambda$ is the right choice.

Exercise 4 (4 points)**.** Compute the proximal operator of the 1, 2-norm, i.e.

$$\mathrm{prox}_{\tau||X||_{1,2}},$$

where $X \in \mathbb{R}^{m \times n}$ is a matrix .

Solution. Firstly recall the subdifferential of $1, 2$-norm computed in previous sheet:

$$\partial \left\|X\right\|_{1,2} = \{P \in \mathbb{R}^{m\times n} : P_i \in \partial ||X_i||_2\}$$

where P_i and X_i are the i-th row of coressponding matrix and

$$\partial \left\|X_i\right\| = \begin{cases} \frac{X_i}{\|X_i\|_2}, & X_i \neq 0 \\ \{P_i \in \mathbb{R}^n : \|P_i\|_2 \leq 1\}, & X_i = 0 \end{cases}$$

Now we use the definition of proximity operator:

$$\text{prox}_{\tau||X||_{1,2}}(Y) = \text{argmin}_X \, ||X||_{1,2} + \frac{1}{2\tau}||X - Y||_2^2$$

which gives us the following by using the optimality condition:

$$0 \in \partial||X||_{1,2} + \frac{1}{\tau}(X - Y).$$

Since each row is independently, we can solve it for each row and get:

$$0 \in \partial||X_i||_2 + \frac{1}{\tau}(X_i - Y_i).$$

If $\|X_i\| \neq 0$, we have $Y_i = \tau\frac{X_i}{\|X_i\|} + X_i$. If we denote $X_i = te_i$ where $e_i := \frac{X_i}{\|X_i\|}$, previous equation becomes $Y_i = \tau e_i + te_i$. Hence, $\|Y_i\|_2 = \tau + t$, which implies $\|Y_i\| > \tau$.
If $\|X_i\| = 0$, we have $Y_i \in \{P_i \in \mathbb{R}^n : \|P_i\|_2 \leq \tau\}$.
To summary we have:

$$\text{prox}_{\tau||X||_{1,2}}(Y) = \left\{ X \in \mathbb{R}^{m\times n} : X_i = \begin{cases} 0, & \text{if } \|Y_i\| \leq \tau \\ (\|Y_i\|_2 - \tau)\frac{Y_i}{\|Y_i\|}, & \text{if } \|Y_i\| > \tau \end{cases} \right\}.$$

Multinomial Logistic Regression (Due:16.12) (16 Points)

Exercise 5 (16 Points)**.** In this exercise you are asked to train a linear model for a multiclass classification task with Logistic regression. The idea is as follows: You are given a set of training samples $\mathcal{I} = \{1, \dots, N\}$ that are represented by their feature vectors $x_i \in \mathbb{R}^d$, for $i \in \mathcal{I}$. Each training sample i is associated with a class label $y_i \in \{1, \dots, C\}$. The aim is to estimate a linear classifier parameterized by $W^* \in \mathbb{R}^{d \times C}, b^* \in \mathbb{R}^C$ so that $y_i = \text{argmax}_{1 \leq j \leq C}\, x_i^\top W_j^* + b_j^*$ for most training samples i. Once you have obtained this "optimal" classifier the hope is, that you are able to classify new unseen and unlabeled samples $x \in \mathbb{R}^d$. In machine learning this is called generalization. For this task you may query your trained model via the classifier rule

$$y = \text{argmax}_{1 \leq j \leq C}\, x^\top W_j^* + b_j^* \tag{1}$$

and y probably is the true class label of x if your model generalizes well. In order to estimate the model we solve an optimization problem of the form

$$\min_{W \in \mathbb{R}^{d \times C}, b \in \mathbb{R}^C} \frac{1}{N} \sum_{i=1}^{N} \ell(W, b, x_i, y_i) + \frac{\lambda_1}{2} \|W\|_2^2 + \frac{\lambda_2}{2} \|b\|_2^2, \tag{2}$$

where

$$\ell(W, b, x_i, y_i) = -\log \left(\frac{\exp(\langle W_{y_i}, x_i \rangle + b_{y_i})}{\sum_{j=1}^{C} \exp(\langle W_j, x_i \rangle + b_j)} \right) \tag{3}$$

is called the softmax loss. Note that the above problem is smooth and strongly convex and can be solved with gradient descent. In practice however, it may happen, that some features (i.e. components of the vector x_i) do not contain any information about the true class labels, i.e. components that are just noise. In order to filter out the useless features we modify the norm on W. So we have

$$\min_{W \in \mathbb{R}^{d \times C}, b \in \mathbb{R}^C} \frac{1}{N} \sum_{i=1}^{N} \ell(W, b, x_i, y_i) + \frac{\lambda_1}{2} \|W\|_{1,2} + \frac{\lambda_2}{2} \|b\|_2^2 \tag{4}$$

You are asked to do the following:

- Download the toy data template from the homepage
- Implement a proximal gradient descent algorithm to optimize above objective function (4) (Avoid for-loops)
- Make sure that your objective monotonically decreases. Plot the objective values. Stop your code if the difference of two successive iterates is less than 10^{-12}.

- In order to ensure that your derivative is computed correctly you may first optimize the fully differentiable model (2) with MATLABs *fminunc* with the options *'GradObj'*, *'On'* and *'DerivativeCheck'*, *'On'*. (Python: check out e.g. `scipy.optimize.grad_check`. This step is optional.)

- Iteratively compute the test error in percent, i.e. how many test samples are not classified correctly via the rule (1).

- Play around with different parameter settings for λ_1, λ_2. Can you identify the useless features? Explain why the model generalizes better to unseen test data if you use $1,2$-norm on W^* (answer by comment at the end of your code).

- You may apply your code to the MNIST dataset `http://yann.lecun.com/exdb/mnist/` and see that your are now able to classify handwritten digits (Optional).

Solution. We apply the proximal gradient descent scheme to our objective (4). To this end we need compute the partial derivatives $\frac{\partial F(W,b)}{\partial W_{lk}}$ and $\frac{\partial F(W,b)}{\partial b_k}$ of the differentiable part of the objective

$$F(W,b) = \frac{1}{N}\sum_{i=1}^{N} \ell(W,b,x_i,y_i) + \frac{\lambda_1}{2}\|W\|_2^2 + \frac{\lambda_1}{2}\|b\|_2^2.$$

First we observe, that

$$\frac{\partial F(W,b)}{\partial W_{lk}} = \frac{1}{N}\sum_{i=1}^{N} \frac{\partial \ell(W,b,x_i,y_i)}{\partial W_{lk}} + \lambda_1 W_{lk}$$

and

$$\frac{\partial F(W,b)}{\partial b_k} = \frac{1}{N}\sum_{i=1}^{N} \frac{\partial \ell(W,b,x_i,y_i)}{\partial b_k} + \lambda_1 b_k.$$

For some class $1 \leq k \leq C$ define

$$h_k(W,b) = \frac{\exp(\langle W_{y_i}, x_i\rangle + b_{y_i})}{\sum_{j=1}^{C}\exp(\langle W_j, x_i\rangle + b_j)}$$

and

$$\mathbf{1}\{y_i = k\} = \begin{cases} 1 & \text{if } y_i = k \\ 0 & \text{otherwise.}\end{cases}$$

Via the one-dimensional chain rule and the quotient rule the partial derivatives of the individual loss terms are given as:

$$\begin{aligned}
&\frac{\partial \ell(W,b,x_i,y_i)}{\partial W_{lk}} \\
=&-\frac{1}{h_{y_i}(W,b)}\cdot\frac{\mathbf{1}\{y_i=k\}\cdot\exp(\langle W_{y_i},x_i\rangle+b_{y_i})\cdot x_{il}\cdot\left(\sum_{j=1}^{C}\exp(\langle W_j,x_i\rangle+b_j)\right)}{\left(\sum_{j=1}^{C}\exp(\langle W_j,x_i\rangle+b_j)\right)^2} \\
&+\frac{1}{h_{y_i}(W,b)}\cdot\frac{\exp(\langle W_{y_i},x_i\rangle+b_{y_i})\cdot\exp(\langle W_k,x_i\rangle+b_k)\cdot x_{il}}{\left(\sum_{j=1}^{C}\exp(\langle W_j,x_i\rangle+b_j)\right)^2} \\
=&-\frac{1}{h_{y_i}(W,b)}\cdot\mathbf{1}\{y_i=k\}\cdot x_{il}\cdot h_{y_i}(W,b)+\frac{1}{h_{y_i}(W,b)}\cdot h_{y_i}(W,b)\cdot h_k(W,b)\cdot x_{il} \\
=&(h_k(W,b)-\mathbf{1}\{y_i=k\})\cdot x_{il}.
\end{aligned}$$

Similarly we obtain for the derivative wrt. b_k:

$$\begin{aligned}
&\frac{\partial \ell(W,b,x_i,y_i)}{\partial b_k} \\
=&-\frac{1}{h_{y_i}(W,b)}\cdot\frac{\mathbf{1}\{y_i=k\}\cdot\exp(\langle W_{y_i},x_i\rangle+b_{y_i})\cdot\left(\sum_{j=1}^{C}\exp(\langle W_j,x_i\rangle+b_j)\right)}{\left(\sum_{j=1}^{C}\exp(\langle W_j,x_i\rangle+b_j)\right)^2} \\
&+\frac{1}{h_{y_i}(W,b)}\cdot\frac{\exp(\langle W_{y_i},x_i\rangle+b_{y_i})\cdot\exp(\langle W_k,x_i\rangle+b_k)}{\left(\sum_{j=1}^{C}\exp(\langle W_j,x_i\rangle+b_j)\right)^2} \\
=&h_k(W,b)-\mathbf{1}\{y_i=k\}.
\end{aligned}$$