Convex Optimization for Machine Learning and Computer Vision

Lecture: T. Wu
Exercises: E. Laude, T. Möllenhoff
Summer Semester 2017

Computer Vision Group
Institut für Informatik
Technische Universität München

Weekly Exercises 5

Room: 02.09.023
Monday, 12.06.2017, 12:15-14:00
Submission deadline: Wednesday, 31.05.2017, Room 02.09.023

Proximal mapping (8 Points + 4 Bonus)

Exercise 1 (4 Points)**.** Compute the proximity operator of the ℓ_2-norm, i.e.

$$\operatorname{prox}_{\|\cdot\|_2}.$$

Solution. From Moreau's identity we know that for any convex, proper lsc $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ and any $x \in \mathbb{R}^n$

$$x = \operatorname{prox}_{\tau f}(v) + \operatorname{prox}_{\tau f^*}(\tfrac{v}{\tau}).$$

Further we know that the convex conjugate of any norm is the indicator function of the unit ball wrt the dual norm. Since the dual norm of the ℓ_2-norm is again the ℓ_2-norm, we have that:

$$\|\cdot\|_2^* = \delta\{\|\cdot\|_2 \leq 1\}.$$

Overall we have

$$\operatorname{prox}_{\tau\|\cdot\|_2}(x) = x - \operatorname{prox}_{\delta\{\|\cdot\|_2 \leq 1\}}(\tfrac{v}{\tau}),$$

where

$$\operatorname{prox}_{\delta\{\|\cdot\|_2\leq 1\}}(y) = \operatorname{proj}_{\{x:\|x\|_2\leq 1\}}(y) = \begin{cases} y & \text{if } \|y\|_2 \leq 1 \\ \frac{y}{\|y\|_2} & \text{otherwise.} \end{cases}$$

Exercise 2 (4 Points)**.** Prove that the proximal operator of the nuclear norm is the proximal operator of the ℓ_1-norm applied to the singular values of the input argument. Formally, let $Y \in \mathbb{R}^{n\times n}$ and let $Y = U\Sigma V^\top$ be the singular value decomposition of Y. Prove that

$$\operatorname{prox}_{\tau\|\cdot\|_{\text{nuc}}}(Y) = U\operatorname{diag}(\{(\sigma_i - \tau)_+\})V^\top,$$

where $\operatorname{diag}(\{\sigma_i - \tau)_+\}) := \operatorname{diag}(\{\max\{0, \sigma_i - \tau\}\}) = \operatorname{prox}_{\tau\|\cdot\|_1}(\{\sigma_i\})$ is the shrinkage (or soft thresholding) operator applied to the singular values σ_i of Y.

Solution. Let $Y \in \mathbb{R}^{n\times n}$. We are interested in the solution of

$$\operatorname{argmin}_X \frac{1}{2}\|X - Y\|_F^2 + \tau\|X\|_{\text{nuc}}. \tag{1}$$

Since the above problem is strictly convex there exists a unique solution $\hat{X}$. The optimality condition of the problem is given as

$$0 \in \hat{X} - Y + \partial\|\cdot\|_{\text{nuc}}(\hat{X}). \tag{2}$$

where $\partial\|\cdot\|_{\text{nuc}}(X)$ is the subdifferential of the nuclear norm at X characterized on exercise sheet 3. Our aim is to show that $\hat{X} := U\text{diag}(\{(\sigma_i - \tau)_+\})V^\top$ meets the optimality condition. To this end we decompose $V = [V_1\ V_2]$, $U = [U_1\ U_2]$ and $\Sigma = \begin{bmatrix}\Sigma_1 & 0\\ 0 & \Sigma_2\end{bmatrix}$ so that

$$Y = U_1\Sigma_1 V_1^\top + U_2\Sigma_2 V_2^\top,$$

where Σ_1 contains all singular values $\sigma_i > \tau$ and Σ_2 all singular values $\sigma_i \leq \tau$. We may then write $\hat{X}$ as

$$\hat{X} = U\text{diag}(\{(\sigma_i - \tau)_+\})V^\top = U_1 \underbrace{(\Sigma_1 - \tau I)}_{\sigma_i>0} V_1^\top + U_2 \underbrace{\text{diag}(\{0\})}_{\sigma_i=0} V_2^\top.$$

We will now show that $\hat{X}$ meets (2): $Y - \hat{X}$ is given as

$$Y - \hat{X} = \tau(U_1V_1^\top + U_2\tfrac{1}{\tau}\Sigma_2 V_2^\top).$$

By construction $\|\frac{1}{\tau}\Sigma_2\|_{\text{spec}} \leq 1$. And therefore and due to sheet 3

$$Y - \hat{X} \in \tau\partial\|\cdot\|_{\text{nuc}}(\hat{X})$$

Definition. A function $g : \mathbb{R}^n \to \mathbb{R}\cup\{\infty\}$ is called 1-homogeneous if

$$g(\alpha x) = \alpha g(x),$$

for all $\alpha \geq 0$.

Exercise 3 (4 Points)**.** Let $g : \mathbb{R}^n \to \mathbb{R}\cup\{\infty\}$ be convex, closed, proper and 1-homogeneous. Show that the proximity operator of the sum $\|\cdot\|_2 + g$ is the composition of the proximity operators of $\|\cdot\|_2$ and g, i.e.

$$\text{prox}_{\|\cdot\|_2+g} = \text{prox}_{\|\cdot\|_2} \circ \text{prox}_g.$$

Solution. Let $y \in \text{dom}(g)$. We have the following optimality conditions for $\text{prox}_{\|\cdot\|_2+g}(y)$, $\text{prox}_{\|\cdot\|_2\circ g}(y)$ and $\text{prox}_g(y)$:

$$0 \in \text{prox}_{\|\cdot\|_2+g}(y) - y + \partial(\|\cdot\|_2 + g)(\text{prox}_{\|\cdot\|_2+g}(y)) \tag{3}$$
$$0 \in \text{prox}_{\|\cdot\|_2}(\text{prox}_g(y)) - \text{prox}_g(y) + \partial(\|\cdot\|_2)(\text{prox}_{\|\cdot\|_2}(\text{prox}_g(y))) \tag{4}$$
$$0 \in \text{prox}_g(y) - y + \partial g(\text{prox}_g(y)) \tag{5}$$

Adding the last two inclusions yields:

$$0 \in \operatorname{prox}_{\|\cdot\|_2}(\operatorname{prox}_g(y)) - y + \partial g(\operatorname{prox}_g(y)) + \partial(\|\cdot\|_2)(\operatorname{prox}_{\|\cdot\|_2}(\operatorname{prox}_g(y))). \tag{6}$$

Assume that it holds for all $x \in \mathbb{R}^n$

$$\partial g(\operatorname{prox}_{\|\cdot\|_2}(x)) \supseteq \partial g(x). \tag{7}$$

Then for $x := \operatorname{prox}_g(y)$, and due to (6) and the sum rule of the subdifferential $\partial(\|\cdot\|_2 + g) \supseteq \partial\|\cdot\|_2 + \partial g$ we have that

$$\begin{aligned} 0 &\in \operatorname{prox}_{\|\cdot\|_2}(\operatorname{prox}_g(y)) - y + \partial g(\operatorname{prox}_{\|\cdot\|_2}(\operatorname{prox}_g(y))) + \partial(\|\cdot\|_2)(\operatorname{prox}_{\|\cdot\|_2}(\operatorname{prox}_g(y))) \\ &\subset \operatorname{prox}_{\|\cdot\|_2}(\operatorname{prox}_g(y)) - y + \partial(g + \|\cdot\|_2)(\operatorname{prox}_{\|\cdot\|_2}(\operatorname{prox}_g(y))). \end{aligned}$$

This shows that $\operatorname{prox}_{\|\cdot\|_2}(\operatorname{prox}_g(y))$ satisfies (3) and therefore $\operatorname{prox}_{\|\cdot\|_2}(\operatorname{prox}_g(y)) = \operatorname{prox}_{\|\cdot\|_2+g}(y)$.

It remains to prove the sufficient condition (7). Clearly, for any $x, y \in \mathbb{R}^n$ with $x \perp y$ we have that $\|x+y\|_2 \geq \|y\|_2$, since $x \perp y$ implies $\langle x, y\rangle = 0$. Then we have that

$$\begin{aligned} \min_x \frac{1}{2}\|x-y\|_2^2 + \|x\|_2 &= \min_{\lambda, z\perp y} \frac{1}{2}\|z+\lambda y - y\|_2^2 + \|z+\lambda y\|_2 \\ &= \min_\lambda \frac{1}{2}\|\lambda y - y\|_2^2 + \|\lambda y\|_2 \\ &= \min_{\lambda\geq 0} \frac{1}{2}(\lambda-1)^2\|y\|_2^2 + |\lambda|\|y\|_2. \end{aligned}$$

The constraint in the last equality can be seen as follows: Suppose $\lambda < 0$. Then increasing it to zero decreases both summands of the objective. Therefore, we have that $\operatorname{prox}_{\|\cdot\|_2}(y) = \lambda y$ for some $\lambda \geq 0$ and clearly $\operatorname{prox}_{\|\cdot\|_2}(y) = 0 \iff y = 0$. Since g is 1-homogeneous, its subdifferential is scaling invariant, meaning that $p \in \partial g(y) \implies p \in \partial g(\lambda y)$ for $\lambda > 0$, we have that (for $y \neq 0$) there exists $\lambda > 0$ so that,

$$\partial g(y) \subseteq \partial g(\lambda y) = \partial g(\operatorname{prox}_{\|\cdot\|_2}(y)).$$

It remains to prove the scaling invariance of the subdifferential for 1-homogeneous g. Let $\lambda > 0$: Via the substitution $z' = \frac{1}{\lambda}z$ we obtain that

$$\begin{aligned} p \in \partial g(y) &\implies \langle p, z-y\rangle + g(y) \leq g(z), \quad \forall z \in \operatorname{dom}(g) \\ &\implies \langle p, \lambda z - \lambda y\rangle + \lambda g(y) \leq \lambda g(z), \quad \forall z \in \operatorname{dom}(g) \\ &\implies \langle p, \lambda z - \lambda y\rangle + g(\lambda y) \leq g(\lambda z), \quad \forall z \in \operatorname{dom}(g) \\ &\implies \langle p, z' - \lambda y\rangle + g(\lambda y) \leq g(z'), \quad \forall z' \in \operatorname{dom}(g) \\ &\implies p \in \partial g(\lambda y). \end{aligned}$$

Multinomial Logistic Regression (16 Points)

Exercise 4 (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 $y_i = \text{argmax}_{1 \le j \le C} \, x_i^\top W_j^* + b_j^*$ 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

$$y = \text{argmax}_{1 \le j \le C} \, x^\top W_j^* + b_j^* \tag{8}$$

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

$$\min_{W \in \mathbb{R}^{d \times C}, b \in \mathbb{R}^C} \; \frac{1}{N} \sum_{i=1}^{N} \ell(W, b, x_i, y_i) + \frac{\lambda_1}{2} \|W\|_2^2 + \frac{\lambda_1}{2} \|b\|_2^2, \tag{9}$$

where

$$\ell(W, b, x_i, y_i) = -\log \left(\frac{\exp(\langle W_{y_i}, x_i \rangle + b_{y_i})}{\sum_{j=1}^{C} \exp(\langle W_j, x_i \rangle + b_j)} \right) \tag{10}$$

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 add the nonsmooth sparsity inducing ℓ_1-norm term on W. So overall we would like to optimize

$$\min_{W \in \mathbb{R}^{d \times C}, b \in \mathbb{R}^C} \; \frac{1}{N} \sum_{i=1}^{N} \ell(W, b, x_i, y_i) + \frac{\lambda_1}{2} \|W\|_2^2 + \frac{\lambda_1}{2} \|b\|_2^2 + \lambda_2 \|W\|_1. \tag{11}$$

You are asked to do the following:

- Download the toy data template from the homepage
- Implement a proximal gradient descent algorithm to optimize the above objective (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 (9) with MATLABs *fminunc* with the options *'GradObj'*, *'On'* and *'DerivativeCheck'*, *'On'*.
- Iteratively compute the test error in percent, i.e. how many test samples are not classified correctly via the rule (8).
- Play around with different parameter settings for λ_1, λ_2. What do you observe? Can you identify the useless features? Explain why the model generalizes better to unseen test data if you add a sparsity inducing term.
- 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.

Solution. We apply the proximal gradient descent scheme to our objective (11). 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:

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