TAKE-HOME MID-TERM EXAM

CSCI 5525 Advanced Machine Learning (Spring 2021)

Due 11:59 pm, May 02 2021 (No Extension)

Instruction Typesetting your solutions in LaTeX is optional but encouraged, and you need to submit it as a single PDF file in Canvas. For programming, include all your codes and running results in a single Jupyter notebook file and submit it alongside the main PDF (since Jupyter notebook also allows text editing, feel free to put your textual answers inside the Jupyter notebook sometimes).

Important NO collaboration or discussion with others is allowed, except for posting clarification questions in the Canvas discussion forum. Any detected violation will be considered as plagiarism, resulting in zero scores and reporting to the department.

Problem 1 (5/15) Let $\boldsymbol{y} \in \mathbb{R}^N$ and $\boldsymbol{A} \in \mathbb{R}^{N \times d}$. Consider the Lasso problem

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} f(\boldsymbol{x}) \doteq \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_1 \tag{1}$$

for a certain fixed $\lambda > 0$. We showed in HW3 that the objective $f(\boldsymbol{x})$ is convex in $\boldsymbol{x}$. But note that $\|\boldsymbol{x}\|_1 = \sum_{j=1}^d |x_i|$ is non-differentiable, and hence we cannot directly apply gradient descent methods to solve problem (1).

Subdifferential generalizes gradient of differential functions to non-differentiable functions. Here we focus on convex functions. For a convex function $g(\boldsymbol{w}) : W \to \mathbb{R}$ where $W \subset \mathbb{R}^n$ is a convex set, the subdifferential at any $\boldsymbol{w}_0 \in W$ is the set

$$\partial g(\boldsymbol{w}_0) \doteq \{\boldsymbol{z} \in \mathbb{R}^n : g(\boldsymbol{w}) \ge g(\boldsymbol{w}_0) + \langle \boldsymbol{z}, \boldsymbol{w} - \boldsymbol{w}_0 \rangle \ \forall\, \boldsymbol{w} \in W\}, \tag{2}$$

which is inspired by the fact that if g is *differentiable* and convex over W,

$$g(\boldsymbol{w}) \ge g(\boldsymbol{w}_0) + \langle \nabla g(\boldsymbol{w}_0), \boldsymbol{w} - \boldsymbol{w}_0 \rangle \ \forall\, \boldsymbol{w}, \boldsymbol{w}_0 \in W. \tag{3}$$

Any element $\boldsymbol{v} \in \partial g(\boldsymbol{w}_0)$ is called a subgradient of g at $\boldsymbol{w}_0$. The following properties are of interest about subdifferentials:

- If g is differentiable at $\boldsymbol{w} \in W$, then $\partial g(\boldsymbol{w}) = \{\nabla g(\boldsymbol{w})\}$, i.e., the subdifferential is a singleton that only contains the usual gradient.
- Positive sum rule: Let both $g, h : W \to \mathbb{R}$ be convex. Then:

$$\partial (t_1 g + t_2 h)(\boldsymbol{w}) = t_1 \partial g(\boldsymbol{w}) + t_2 \partial h(\boldsymbol{w}) \ \ \forall\, \boldsymbol{w} \in W \ \ \forall\, t_1, t_2 \ge 0, \tag{4}$$

  where the second $+$ denotes the set summation, i.e., $A + B = \{\boldsymbol{x} + \boldsymbol{y} : \boldsymbol{x} \in A, \boldsymbol{y} \in B\}$.
- Composition with an affine mapping: let g be convex over the range of $\ell(\boldsymbol{w}) \doteq \boldsymbol{A}_0 \boldsymbol{w} + \boldsymbol{b}_0$ for certain constant $\boldsymbol{A}_0$ and $\boldsymbol{b}_0$, then

$$\partial (g \circ \ell)(\boldsymbol{w}) = \boldsymbol{A}_0^\intercal \partial g(\boldsymbol{A}_0 \boldsymbol{w} + \boldsymbol{b}_0), \tag{5}$$

  i.e., a simple chain rule holds in this case.

(a) Show that

$$\partial_z |z| = \begin{cases} 1 & z > 0 \\ -1 & z < 0 \\ [-1, 1] & z = 0 \end{cases} . \tag{6}$$

For convenience, henceforth we will modify the standard definition of sign function and define $\mathrm{sign}(z) \doteq \partial_z |z|$. (0.5/15)

(b) Show that

$$\partial_{\boldsymbol{x}} f(\boldsymbol{x}) = \boldsymbol{A}^{\intercal}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}) + \lambda \,\mathrm{sign}(\boldsymbol{x}), \tag{7}$$

where $\mathrm{sign}(\boldsymbol{x})$ means applying the sign function elementwise. Justify each of your key steps. (0.5/15)

(c) For a vector $\boldsymbol{z} \in \mathbb{R}^n$, what's $\partial_{\boldsymbol{z}} \|\boldsymbol{z}\|_2$? Justify all your steps. (1/15)

(d) For an unconstrained $g(\boldsymbol{w})$, we can generalize gradient descent methods into *subgradient methods*—there is no word "descent", as a negative subgradient direction may or may not be a descent direction, different from the differentiable case. The update step looks like:

$$\boldsymbol{w}^{(k)} = \boldsymbol{w}^{(k-1)} - \eta^{(k)} \boldsymbol{v}^{(k)} \quad \text{for some} \quad \boldsymbol{v}^{(k)} \in \partial g(\boldsymbol{w}^{(k-1)}). \tag{8}$$

Here the selection rule of the subgradient $\boldsymbol{v}^{(k)}$ is typically predefined, but in principle any element of the subdifferential is allowed. The step size $\eta^{(k)}$ is either taken to be very small, or follow a predefined decay schedule, e.g., the popular square-root schedule, $\eta^{(k)} = \frac{c}{\sqrt{k}}$, where c is a sufficiently small constant depending on the problem structure. Unfortunately, line search backtracking cannot be applied here, as there is no Taylor expansion—which the backtracking algorithm hinges on—for non-differentiable functions. Also, there is no strong stopping criterion based on the gradient norm, as the first-order optimality condition now reads $\mathbf{0} \in \partial g(\boldsymbol{w})$, i.e., there can still be subgradients with large magnitudes even at a minimizer. A weak criterion is the change of function value $\delta \doteq |g(\boldsymbol{w}^{(k)}) - g(\boldsymbol{w}^{(k-1)})|$, and the iteration process is stopped when δ is sufficiently small.

(i) Fix a random seed, and set $N = 30, d = 100$. Generate $\boldsymbol{A}$ as iid standard normal, a groundtruth $\boldsymbol{x}_0$ as iid Bernoulli with rate 0.2 (i.e., assuming 1 with probability 0.2, and otherwise assuming 0), and so $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is iid normal with mean 0 and variance 0.5, i.e., $\mathsf{N}(0, 0.5)$. Implement Lasso to estimate $\boldsymbol{x}$ using the subgradient method. Try at least 3 different values for λ, ideally with different orders of magnitude. Plot your recovered results against the groundtruth $\boldsymbol{x}_0$ on a stem plot. (1/15)

(ii) Consider square-root Lasso:

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} f_{\mathrm{sqrt}}(\boldsymbol{x}) \doteq \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2 + \lambda \|\boldsymbol{x}\|_1, \tag{9}$$

i.e., there is no square for the ℓ_2 norm in the first term. Solve square-root Lasso using subgradient methods and experiment with the same data as in (i). Please also try at least 3 different λ's (not necessarily the same values as in (i)) and produce a similar stem plot. (1/15)

(iii) Now fix reasonably good λ values for Lasso and square-root Lasso through experimentation in (i) and (ii). Try at least 3 different noise levels with variance below 5, and compare Lasso and square-root Lasso in terms of the recovery performance. What do you observe? (1/15)

Problem 2 (3/15) Given a training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ where $\boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}$, and a positive definite kernel K on $\mathbb{R}^d$ which induces a feature mapping $\Phi : \mathbb{R}^d \to \mathbb{H}$ for a certain Hilbert space $\mathbb{H}$. Consider the following regularized regression formulation

$$\min_{\boldsymbol{w} \in \mathbb{H}} \sum_{i=1}^N \left(\langle \boldsymbol{w}, \Phi\left(\boldsymbol{x}_i\right)\rangle - y_i\right)^2 + \lambda \left\|\boldsymbol{w}\right\|_{\mathbb{H}}^2 \tag{10}$$

with a certain $\lambda > 0$. Here, one can think of the $\|\cdot\|_{\mathbb{H}}$ as the ℓ_2 norm in $\mathbb{H}$. The final predictor will take the form $\langle \boldsymbol{w}_*, \Phi\left(\boldsymbol{x}\right)\rangle$ for any new input $\boldsymbol{x} \in \mathbb{R}^d$, where $\boldsymbol{w}_*$ is any global minimizer to problem (10).

(a) Show that problem (10) can be solved via solving an optimization problem that only involve the Gram matrix $\boldsymbol{G}$, y_i's as given data. In other words, show how to implement the kernel trick here. Justify the key steps you take. (1/15)

(b) Does the new optimization problem has a unique global minimizer? Why or why not? (0.5/15) Derive a closed-form solution for it. (0.5/15)

(c) Derive a closed-form expression for the predictor, i.e., it should only involve the known data $\boldsymbol{G}$, y_i's, and the new input $\boldsymbol{x}$. (1/15)

Problem 3 (5/15) A major issue with the soft-margin SVM

$$\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^N \xi_i \quad \text{s. t.} \;\; y_i\left(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b\right) \ge 1 - \xi_i, \;\; \xi_i \ge 0 \;\forall\, i \tag{11}$$

is that the tradeoff C is unintuitive, i.e., there is a qualitative relationship between C and the tradeoff of the two kinds of support vectors, but the control is not very explicit or quantitative. Consider the following modification:

$$\min_{\boldsymbol{w}, b, \boldsymbol{\xi}, \rho} \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \nu\rho + \sum_{i=1}^N \xi_i \quad \text{s. t.} \;\; y_i\left(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b\right) \ge \rho - \xi_i, \;\; \xi_i \ge 0 \;\forall\, i \quad \rho \ge 0 \tag{12}$$

where $\nu > 0$ is a fixed parameter. Note that the C has gone, and there is a new penalty term $-\nu\rho$ in the objective, and the margin lower bounds have changed from $1 - \xi_i$ into $\rho - \xi_i$ for an optimizable $\rho \ge 0$.

(a) Is problem (12) convex or not? Why or why not? (0.5/15)

(b) Verify the Slater condition, and write down the KKT optimality condition. (1/15)

(c) Similar to the soft-margin SVM, support vectors are $\boldsymbol{x}_i$'s with $y_i\left(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b\right) = \rho - \xi_i$, and outliers are those support vectors with $\xi_i > 0$. Suppose we obtain a global minimizer with $\rho > 0$. Use the KKT condition to show that ν is an upper bound on the number of outliers, and also a lower bound on the number of support vectors. (1/15)

(d) Let's set $\nu = \frac{1}{2}$ and suppose the training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, where $y_i \in \{+1, -1\}$ for all i, is linearly separable. Show that solving problem (12) yields the same binary classifier as that of hard-margin SVM. (Hint: consider $\rho > 0$ and $\rho = 0$ separably, and argue that for a linearly separable dataset, only one of the two cases is possible.) (1.5/15)

(e) Now suppose the training set is general, i.e., either linearly separable or not. Assume $(\boldsymbol{w}_*, b_*, \boldsymbol{\xi}_*, \rho_*)$ is a global minimizer to problem (12) and $\rho_* > 0$. Can you construct a global minimizer to the soft-margin SVM with $C = \rho_*^{-1}$? (1/15)

Problem 4 (2/15) Recall that the definition of (empirical) Rademacher complexity for function classes is induced by the Radamacher complexity for sets. For a set $X \subset \mathbb{R}^d$, the Radamacher complexity is defined as

$$R(X) \doteq \mathbb{E}_{\boldsymbol{r} \sim_{iid} \mathrm{Rad}} \sup_{\boldsymbol{x} \in X} \langle \boldsymbol{x}, \boldsymbol{r} \rangle, \tag{13}$$

where Rad denotes the Rademacher distribution. We also talked about Gaussian width/complexity, which just replaces the iid Rademacher vector with an iid Gaussian vector, i.e.,

$$G(X) \doteq \mathbb{E}_{\boldsymbol{g} \sim_{iid} \mathsf{N}(0,1)} \sup_{\boldsymbol{x} \in X} \langle \boldsymbol{x}, \boldsymbol{g} \rangle. \tag{14}$$

These two complexity measures are closely related, and it can be shown that

$$\sqrt{\frac{2}{\pi}} R(X) \le G(X) \le 2\sqrt{\log d}\, R(X). \tag{15}$$

But the Gaussian complexity may be easier to estimate in most cases, due to the rich collection of results on Gaussian random processes in the literature, see, e.g., [Ver18].

Given N data points $\{\boldsymbol{x}_i\}_{i=1}^N$, or in matrix form $\boldsymbol{X} \in \mathbb{R}^{N \times d}$, consider estimating the empirical Rademacher complexity of the set of linear functions with bounded norms.

(a) The bounded ℓ_2 norm case, i.e.,

$$\mathcal{H} \doteq \{\boldsymbol{x} \mapsto \langle \boldsymbol{w}, \boldsymbol{x} \rangle : \|\boldsymbol{w}\|_2 \le 1\}, \tag{16}$$

boils down to estimating

$$\mathbb{E}_{\boldsymbol{r} \sim_{iid} \mathrm{Rad}} \sup_{\|\boldsymbol{w}\|_2 \le 1} \langle \boldsymbol{X}\boldsymbol{w}, \boldsymbol{r} \rangle. \tag{17}$$

The process is provided in the proof of Lemma 26.10 (Section 26.2) of the book [SSS14][1]. Modify the proof to estimate, i.e., provide a reasonably tight upper bound for, the Gaussian complexity. (Hint: Jensen's inequality implies that for any function f and random variable $\boldsymbol{v}$, $\mathbb{E}_{\boldsymbol{v}} \|f(\boldsymbol{v})\|_2 \le (\mathbb{E}_{\boldsymbol{v}} \|f(\boldsymbol{v})\|_2^2)^{1/2}$). (1/12)

(b) Consider the bounded ℓ_∞ case, i.e.,

$$\mathcal{H} \doteq \{\boldsymbol{x} \mapsto \langle \boldsymbol{w}, \boldsymbol{x} \rangle : \|\boldsymbol{w}\|_\infty \le 1\}. \tag{18}$$

Estimate the Gaussian complexity

$$\mathbb{E}_{\boldsymbol{g} \sim_{iid} \mathsf{N}(0,1)} \sup_{\|\boldsymbol{w}\|_\infty \le 1} \langle \boldsymbol{X}\boldsymbol{w}, \boldsymbol{g} \rangle. \tag{19}$$

(Hint: The proof of Lemma 26.11 (Section 26.2) of the book [SSS14] is helpful.) (1/15)

END

References

[SSS14] Shai Ben-David Shai Shalev-Shwartz, *Understanding machine learning*, Cambridge University Press, 2014.

[Ver18] Roman Vershynin, *High-dimensional probability*, Cambridge University Pr., 2018.

[1] Available online: https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/copy.html