HOMEWORK SET 2

CSCI5525 Machine Learning: Analysis and Methods (Fall 2024)

Due 11:59 pm, Oct 27 2024

Instruction Your writeup, either typeset or scanned, should be a single PDF file. For problems requiring coding, organize all codes for all problems into **ONE** Jupyter notebook file (i.e., `.ipynb` file) with cell execution outputs. Your submission to Gradescope should include the single PDF and the one notebook file—**please DO NOT zip them**! Please assign page(s) to each question to reduce the TA's navigation time. If your notebook submission doesn't display on Gradescope due to a large file error, please try remove images or figures from your cell outputs and re-upload. No late submission will be accepted. For each problem, you should acknowledge your collaborators—**including AI tools**, if any.

About the use of AI tools You are strongly encouraged to use AI tools—they are becoming our workspace friends, such as ChatGPT (`https://chat.openai.com/`), Claude (`https://claude.ai/chats`), and Github Copilot (`https://github.com/features/copilot`), to help you when trying to solve problems. It takes a bit of practice to ask the right and effective questions/prompts to these tools; we highly recommend that you go through this popular free short course **ChatGPT Prompt Engineering for Developers** offered by `https://learn.deeplearning.ai/` to get started.

If you use any AI tools for any of the problems, you should include screenshots of your prompting questions and their answers in your writeup. The answers provided by such AI tools often contain factual errors and reasoning gaps. **So, if you only submit an AI answer with such bugs for any problem, you will obtain a zero score for that problem**. You obtain the scores only when you explain the bugs and also correct them in your own writing. You can also choose not to use any of these AI tools, in which case we will grade based on the efforts you have made.

Reminder about notations We will use small letters (e.g., u) for scalars, small boldface letters (e.g., $\boldsymbol{a}$) for vectors, and capital boldface letters (e.g., $\boldsymbol{A}$) for matrices. For a matrix $\boldsymbol{A}$, $\boldsymbol{a}^i$ (supscripting) means its i-th row as a *row vector*, and $\boldsymbol{a}_j$ (subscripting) means the j-the column as a column vector, and a_{ij} means its (i,j)-th element. $\mathbb{R}$ is the set of real numbers. $\mathbb{R}^n$ is the space of n-dimensional real vectors, and similarly $\mathbb{R}^{m\times n}$ is the space of $m \times n$ real matrices. The dotted equal sign $\doteq$ means defining.

It is important that you review the related section of the supplementary notes before you attempt the questions.

Problem 1 (Convex analysis & KKT conditions; 5/15)

(a) Prove that a strictly convex function has a unique global minimizer (Hint: it might be useful to consider proof by contradiction). $(0.5/15)$

Is $f(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2$ strictly convex? Why or why not? Is the Hinge loss $\phi(z) = \max(0, 1-z)$ strictly convex? Why or why not? $(0.5/15)$

(b) Show that if $f(\boldsymbol{x}) : X \mapsto \mathbb{R}$ is a convex function, the sublevel set $\{\boldsymbol{x} \in X : f(\boldsymbol{x}) \le 0\}$ is a convex set. $(0.5/15)$

In general, is $\{\boldsymbol{x} \in X : f(\boldsymbol{x}) = 0\}$ a convex set if $f(x) : X \mapsto \mathbb{R}$ is convex? Why or why not? What about when $f(\boldsymbol{x})$ is a linear function, i.e., $f(\boldsymbol{x}) = \boldsymbol{w}^\intercal \boldsymbol{x} + b$ for certain $\boldsymbol{w}$ and b? $(0.5/15)$

(c) (i) Prove that the Lasso objective

$$f\left(\boldsymbol{x}\right)=\left\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\right\|_2^2+\lambda\left\|\boldsymbol{x}\right\|_1 \tag{1}$$

is convex. **You should work from the first principles and potentially use the operation rules**. (0.5/15)

(ii) For any convex subset $S \subset \mathbb{R}^n$, the distance function to S induced by the ℓ_p norm is defined as

$$d_{\ell_p}\left(\boldsymbol{x};S\right) \doteq \inf_{\boldsymbol{y}\in S}\ \left\|\boldsymbol{x}-\boldsymbol{y}\right\|_p . \tag{2}$$

Show that $d_{\ell_p}\left(\boldsymbol{x};S\right)$ is convex. **You should work from the first principles and potentially use the operation rules**. (0.5/15)

(d) Make use of the KKT conditions to solve the following projection problems.

(i) **Projection onto a subspace**. Prove that the ℓ_2 projection of a point $\boldsymbol{y}$ onto an affine subspace $\{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}\}$, where $\boldsymbol{A}$ has full row rank, is $(\boldsymbol{I}-\boldsymbol{A}^\intercal\left(\boldsymbol{A}\boldsymbol{A}^\intercal\right)^{-1}\boldsymbol{A})\boldsymbol{y}+\boldsymbol{A}^\intercal\left(\boldsymbol{A}\boldsymbol{A}^\intercal\right)^{-1}\boldsymbol{b}$, i.e., the global minimizer to

$$\min_{\boldsymbol{x}}\ \left\|\boldsymbol{x}-\boldsymbol{y}\right\|_2^2 \quad \text{s.t. } \boldsymbol{A}\boldsymbol{x}=\boldsymbol{b} \tag{3}$$

is $\boldsymbol{x}_*=(\boldsymbol{I}-\boldsymbol{A}^\intercal\left(\boldsymbol{A}\boldsymbol{A}^\intercal\right)^{-1}\boldsymbol{A})\boldsymbol{y}+\boldsymbol{A}^\intercal\left(\boldsymbol{A}\boldsymbol{A}^\intercal\right)^{-1}\boldsymbol{b}$. (1/15)

(ii) **Projection onto the ℓ_2 norm ball**. Prove that the global minimizer $\boldsymbol{x}_*$ to

$$\min_{\boldsymbol{x}}\ \left\|\boldsymbol{x}-\boldsymbol{y}\right\|_2^2 \quad \text{s.t. } \left\|\boldsymbol{x}\right\|_2^2\le 1 \tag{4}$$

is

$$\boldsymbol{x}_*=\begin{cases}\frac{\boldsymbol{y}}{\left\|\boldsymbol{y}\right\|_2} & \left\|\boldsymbol{y}\right\|_2\ge 1\\ \boldsymbol{y} & \left\|\boldsymbol{y}\right\|_2<1\end{cases}. \tag{5}$$

You can draw a picture on $\mathbb{R}^2$ to see if this solution makes sense. (1/15)

Problem 2 (ν-SVM; 3/15) A major issue with the soft-margin SVM

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}}\ \frac{1}{2}\left\|\boldsymbol{w}\right\|_2^2+C\sum_{i=1}^N\xi_i \quad \text{s.t. } y_i\left(\boldsymbol{w}^\intercal\boldsymbol{x}_i+b\right)\ge 1-\xi_i,\ \xi_i\ge 0\ \forall\ i \tag{6}$$

is that the tradeoff parameter C is not interpretable, i.e., although there is a relationship between C and the tradeoff between the margin and the support vectors, it is not explicit or quantitative. Now, consider the following modification:

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi},\rho}\ \frac{1}{2}\left\|\boldsymbol{w}\right\|_2^2-\nu\rho+\sum_{i=1}^N\xi_i \quad \text{s.t. } y_i\left(\boldsymbol{w}^\intercal\boldsymbol{x}_i+b\right)\ge \rho-\xi_i,\ \xi_i\ge 0\ \forall\ i \quad \rho\ge 0 \tag{7}$$

where $\nu>0$ is a user-specified parameter.

(a) Is problem (7) convex or not? Why or why not? (0.5/15)

(b) Verify the Slater condition and write down the KKT optimality conditions. $(0.5/15)$

(c) Similarly to soft-margin SVM, here support vectors are $\boldsymbol{x}_i$'s with $y_i\left(\boldsymbol{w}^\intercal \boldsymbol{x}_i + b\right) = \rho - \xi_i$, and outliers are those support vectors with $\xi_i > 0$. Suppose that we obtain a global minimizer with $\rho > 0$. Use the KKT conditions 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) Assume that $(\boldsymbol{w}_*, b_*, \boldsymbol{\xi}_*, \rho_*)$ is a global minimizer to problem (7) and $\rho_* > 0$. Can you construct a global minimizer to the soft-margin SVM with $C = \rho_*^{-1}$? $(1/15)$

Problem 3 (Kernel methods; $7/15$)

(a) Recall that typical kernels that we work with are symmetric positive definite (SPD).

(i) Prove that if K is a SPD kernel, then the normalized kernel

$$\overline{K}\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \begin{cases} 0 & K\left(\boldsymbol{x}, \boldsymbol{x}\right) = 0 \text{ or } K\left(\boldsymbol{x}', \boldsymbol{x}'\right) = 0 \\ \frac{K(\boldsymbol{x}, \boldsymbol{x}')}{\sqrt{K(\boldsymbol{x}, \boldsymbol{x}) K(\boldsymbol{x}', \boldsymbol{x}')}} & \text{otherwise} \end{cases} \tag{8}$$

is also SPD. (Hint: one possibility is to express the Gram matrix $[\overline{K}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)]_{ij}$ in terms of the product of $[K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)]_{ij}$ with certain other matrices; $0.5/15$)

(ii) The polynomial kernel of degree-d is $K\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \left(\langle \boldsymbol{x}, \boldsymbol{x}' \rangle + c\right)^d$ where $c \geq 0$ is a parameter to set. Prove that polynomial kernels are SPD. (Hint: think about the operation rules that preserve SPD kernels; $0.5/15$)

(iii) Recall that the popular Gaussian kernel is defined as

$$K\left(\boldsymbol{x}, \boldsymbol{x}'\right) = e^{-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2}{2\sigma^2}} \quad \forall\, \boldsymbol{x}, \boldsymbol{x}'. \tag{9}$$

The Gaussian kernel is a normalized kernel, as

$$e^{-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2}{2\sigma^2}} = \frac{e^{\langle \boldsymbol{x}, \boldsymbol{x}' \rangle / \sigma^2}}{\sqrt{e^{\|\boldsymbol{x}\|_2^2/\sigma^2}} \sqrt{e^{\|\boldsymbol{x}'\|_2^2/\sigma^2}}}. \tag{10}$$

So, to prove that the Gaussian kernel is SPD, it is sufficient to prove that the kernel

$$K^\circ \doteq e^{\langle \boldsymbol{x}, \boldsymbol{x}' \rangle / \sigma^2} \tag{11}$$

is SPD. Prove that K° is SPD. (Hint: recall that the series expansion $e^x = \sum_{n=0}^\infty \frac{x^n}{n!}$ holds for all $x \in \mathbb{R}$. $0.5/15$)

(b) Recall that the soft-margin SVM is equivalent to

$$\min_{\boldsymbol{w}, b} \; \frac{1}{2} \|\boldsymbol{w}\|_2^2 + C \sum_{i=1}^N \max\left(0, 1 - y_i\left(\boldsymbol{w}^\intercal \boldsymbol{x}_i + b\right)\right), \tag{12}$$

which can be written compactly as

$$\min_{\boldsymbol{w}, b} \; \frac{1}{2} \|\boldsymbol{w}\|_2^2 + C \sum_{i=1}^N \ell_h\left(y_i\left(\boldsymbol{w}^\intercal \boldsymbol{x}_i + b\right)\right), \tag{13}$$

where the $\ell_h(z) = \max(0, 1-z)$ is the *hinge loss*. In principle, we can use gradient-type methods to optimize Eq. (13). But the hinge loss is not differentiable, and hence we need to develop *subgradient methods*. Here for simplicity, we instead consider a smoothed version:

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{N} \ell_h^2\left(y_i\left(\boldsymbol{w}^\intercal \boldsymbol{x}_i + b\right)\right). \tag{14}$$

(i) Show that the objective in Eq. (14) is continuous differentiable. $(0.5/15)$

(ii) Derive the gradient and implement gradient descent with backtracking linear search to optimize Eq. (14). $(1/15)$

(iii) Load the breast cancer dataset from scikit-learn (https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html#sklearn.datasets.load_breast_cancer), and compare your implementation of soft-margin SVM with that of scikit-learn: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html for the binary classification on breast cancer. Note that you should set the linear kernel in SVC for a fair comparison. Also, due to our smoothing above, you may not get the same level of performance with the same C parameter for the two implementations. Try to tune around C for both to optimize performance as much as possible. $(0.5/15)$

(c) Now let us apply the kernel trick to move beyond the linear kernel. Suppose that we choose a kernel K that induces a feature mapping Φ. Then Eq. (14) becomes

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{N} \ell_h^2\left(y_i(\boldsymbol{w}^\intercal \Phi\left(\boldsymbol{x}_i\right) + b)\right). \tag{15}$$

(i) Describe how to implement the kernel trick here. $(1/15)$

(ii) Implement the mini-batch stochastic gradient descent (SGD) algorithm to solve the resulting problem from (i).

Then, on the MNIST dataset (you can follow the instruction from https://www.kaggle.com/datasets/hojjatk/mnist-dataset or any other source you find to load the data in Python), perform a binary classification task: $\{0,1,2,3,4\}$ as the first class and $\{5,6,7,8,9\}$ as the second class, with the Gaussian kernel. You may want to set up a small validation set to finetune the kernel parameters as well as the optimization parameters in mini-batch SGD.

Optionally, you may want to compare the performance of your implementation with that of the SGDClassifier function in scikit-learn: https://scikit-learn.org/1.5/modules/generated/sklearn.linear_model.SGDClassifier.html#sklearn.linear_model.SGDClassifier, although unfortunately this function only supports the linear kernel and also its SGD algorithm samples only one sample each time—vs. the mini-batch SGD that we described in class (more details about this function can be found from https://scikit-learn.org/1.5/modules/sgd.html#mathematical-formulation). $(1/15)$

(d) Recall that in soft-margin support vector regression (SVR),

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum\nolimits_i \zeta_i \quad \text{s.t. } |y_i - \langle \boldsymbol{w}, \Phi\left(\boldsymbol{x}_i\right)\rangle - b| \le \varepsilon + \zeta_i, \zeta_i \ge 0 \; \forall\, i. \tag{16}$$

(i) Is Eq. (16) a convex optimization problem or not? Why? $(0.5/15)$

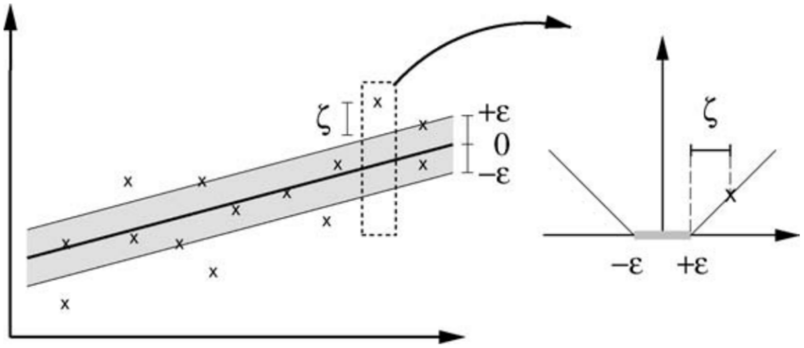

Figure 1: Illustration of SVR and the loss it uses

(ii) Reformulate Eq. (16) into an unconstrained problem. $(0.5/15)$

(iii) How do we implement the kernel trick for SVR? $(0.5/15)$