HOMEWORK SET 4

CSCI 5525 Advanced Machine Learning (Spring 2021)

Due 11:59 pm, Apr 28 2021

Instruction Typesetting your homework 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). No late submission will be accepted.

For each problem, your should acknowledge your collaborators if any. For problems containing multiple subproblems, there are often close logic connections between the subproblems. So always remember to build on previous ones, rather than work from scratch.

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.

Problem 1 (Kernel methods; 8/12) Let our input data points be $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N \in \mathbb{R}^d$. Recall that in kernel methods we *implicitly* map all $\boldsymbol{x}_i$'s into a higher-, often times infinite-, dimensional space $\mathbb{H}$—which is a Hilbert space and hence an inner product can be defined—so that nonlinearity unfolds into linearity in $\mathbb{H}$. The mapping Φ often is not explicitly constructed, but induced by a *kernel function* K that takes any pair of points $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$ and maps them to the inner product $\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}') \rangle$ in $\mathbb{H}$, i.e.,

$$K(\boldsymbol{x}, \boldsymbol{x}') = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}') \rangle \quad \forall\, \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d. \tag{1}$$

(a) K has to satisfy certain conditions to be a valid kernel function, i.e., able to induce a Φ and a valid inner product as described above. One sufficient condition is *symmetric positive definiteness* (SPD): K is said to be *symmetric positive definite* if: 1) $K(\boldsymbol{x}, \boldsymbol{x}') = K(\boldsymbol{x}', \boldsymbol{x})$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$, and 2) for all $m \in \mathbb{N}$ and for all $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m$, the Gram matrix $[K(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{ij}$ is *positive semidefinite*[1].

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

$$\overline{K}(\boldsymbol{x}, \boldsymbol{x}') = \begin{cases} 0 & K(\boldsymbol{x}, \boldsymbol{x}) = 0 \text{ or } K(\boldsymbol{x}', \boldsymbol{x}') = 0 \\ \frac{K(\boldsymbol{x}, \boldsymbol{x}')}{\sqrt{K(\boldsymbol{x}, \boldsymbol{x}) K(\boldsymbol{x}', \boldsymbol{x}')}} & \text{otherwise} \end{cases} \tag{2}$$

is also SPD. (Hint: one possibility is to express the Gram matrix $[\overline{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{ij}$ in terms of the product of $[K(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{ij}$ with certain other matrices; 1/12)

[1] Yes, this is not a typo but an inconsistency of conventions between different fields: kernels and their positive definiteness are notions commonly used in functional analysis and operator theory, whereas positive (semi)definiteness of matrices in linear algebra and matrix analysis. Warning: different authors use different conventions of positive definiteness in the kernel method literature; make sure you understand their conventions before trying to digest their results.

(ii) Recall that the popular Gaussian (or radial basis function—RBF) kernel is defined as

$$K\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|_2^2}{2\sigma^2}\right) \quad \forall\, \boldsymbol{x}, \boldsymbol{x}'. \tag{3}$$

The Gaussian kernel is a normalized kernel, as

$$\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|_2^2}{2\sigma^2}\right) = \frac{\exp\left(\left\langle \boldsymbol{x}, \boldsymbol{x}'\right\rangle/\sigma^2\right)}{\sqrt{\exp\left(\|\boldsymbol{x}\|_2^2/\sigma^2\right)\exp\left(\|\boldsymbol{x}'\|_2^2/\sigma^2\right)}}. \tag{4}$$

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

$$K^\circ \doteq \exp\left(\left\langle \boldsymbol{x}, \boldsymbol{x}'\right\rangle/\sigma^2\right) \tag{5}$$

is SPD. SPD kernels have several closure properties:

- **Summation**: $K_1 + K_2$ is SPD if K_1 and K_2 are SPD;
- **Product**: $K_1 K_2$ is SPD if K_1 and K_2 are SPD;
- **Pointwise limit**: the limit $K = \lim_{n\to\infty} K_n$ is SPD if all K_n's are SPD;
- **Power series composition**: $\sum_{n=0}^{\infty} a_n K^n$ is SPD if K is SPD, $a_n \geq 0$ for all n, and K takes values inside the convergence radius of the power series $\sum_{n=0}^{\infty} a_n x^n$ (this can be easily shown from closure under pointwise limit);
- **Tensor product/summation**: if K_1 is a SPD kernel on $\mathbb{R}^d$ and K_2 is a SPD kernel on $\mathbb{R}^{d'}$. Then, both $K_1 K_2$ and $K_1 + K_2$ are SPD kernels on $\mathbb{R}^d \times \mathbb{R}^{d'}$.

Prove that K° is SPD. (Hint: recall that the series expansion $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$ holds for all $x \in \mathbb{R}$. 1/12)

(iii) Polynomial kernel of degree-d is $K\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \left(\left\langle \boldsymbol{x}, \boldsymbol{x}'\right\rangle + c\right)^d$ where $c \geq 0$ is a parameter to be set. Prove that polynomial kernels are SPD. (Hint: apply the closure properties above. 1/12)

(b) The soft-margin SVM

$$\min_{\boldsymbol{w}, b, \xi_i' s} \ \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{N} \xi_i \quad \text{s. t. } \ y_i\left(\left\langle \boldsymbol{w}, \boldsymbol{x}_i\right\rangle + b\right) \geq 1 - \xi_i, \xi_i \geq 0\ \forall\, i \tag{6}$$

has a convex quadratic objective and linear constraints. So it is a *quadratic programming* problem and can be solved using off-the-shelf quadratic programming solvers. Alternatively, one can also derive its *dual problem*, which is again a convex quadratic problem.

A modern and scalable approach is to write it as an equivalent unconstrained problem. Note that the constraints in Eq. (6) are equivalent to

$$\xi_i \geq 1 - y_i\left(\left\langle \boldsymbol{w}, \boldsymbol{x}_i\right\rangle + b\right), \quad \xi_i \geq 0\ \forall i \Longleftrightarrow \xi_i \geq \max\left(0, 1 - y_i\left(\left\langle \boldsymbol{w}, \boldsymbol{x}_i\right\rangle + b\right)\right)\ \forall\, i. \tag{7}$$

So problem in Eq. (6) is equivalent to

$$\min_{\boldsymbol{w}, b, \xi_i' s} \ \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{N} \xi_i \quad \text{s. t. } \ \xi_i \geq \max\left(0, 1 - y_i\left(\left\langle \boldsymbol{w}, \boldsymbol{x}_i\right\rangle + b\right)\right)\ \forall\, i. \tag{8}$$

Since for any fixed $(\boldsymbol{w}_0, b_0)$, one can set $\xi_i = \max(0, 1 - y_i(\langle \boldsymbol{w}_0, \boldsymbol{x}_i \rangle + b_0))$ for all i to minimize $\sum_{i=1}^N \xi_i$ while remaining feasible, the problem Eq. (8) is in turn equivalent to

$$\min_{\boldsymbol{w}, b} \ \frac{1}{2} \|\boldsymbol{w}\|_2^2 + C \sum_{i=1}^N \max(0, 1 - y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b)). \tag{9}$$

Problem Eq. (9) can be written compactly as

$$\min_{\boldsymbol{w}, b} \ \frac{1}{2} \|\boldsymbol{w}\|_2^2 + C \sum_{i=1}^N \ell_h(y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b)), \tag{10}$$

where the $\ell_h(z) = \max(0, 1 - z)$ is the *hinge loss*. In principle, we can use gradient-type methods to optimize Eq. (10). But the hinge loss is not differentiable and hence we need to develop *subgradient methods*. Here for simplicity, we instead consider a smoothed version:

$$\min_{\boldsymbol{w}, b} \ \frac{1}{2} \|\boldsymbol{w}\|_2^2 + C \sum_{i=1}^N \ell_h^2(y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b)). \tag{11}$$

(i) Show that the objective in Eq. (11) is continuous differentiable. (0.5/12)

(ii) Derive the gradient and implement gradient descent with backtracking linear search to optimize Eq. (11). (0.5/12)

(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 fair comparison, and 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 the performance as much as possible. (1/12)

(c) Now let's apply the kernel trick to move beyond the linear kernel. Suppose we choose a kernel K which induces a feature mapping Φ. Then Eq. (11) becomes

$$\min_{\boldsymbol{w}, b} \ \frac{1}{2} \|\boldsymbol{w}\|_2^2 + C \sum_{i=1}^N \ell_h^2(y_i \langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i) \rangle), \tag{12}$$

where we omit the bias term. The celebrated *representer theorem* says that for any function f and any monotonically nondecreasing function $R: \mathbb{R}_+ \to \mathbb{R}$, any optimization problem of the form

$$\min f(\langle \boldsymbol{w}, \Phi(\boldsymbol{x}_1) \rangle, \ldots, \langle \boldsymbol{w}, \Phi(\boldsymbol{x}_N) \rangle) + R(\|\boldsymbol{w}\|_2) \tag{13}$$

has a global minimizer of the form $\boldsymbol{w} = \sum_{i \in [N]} \alpha_i \Phi(\boldsymbol{x}_i)$. If R is strictly increasing, then all global minimizers take this form. Easy to see that we can apply the representer theorem to Eq. (12), and turn the problem into:

$$\min_{\boldsymbol{\alpha}} \ \frac{1}{2} \left\| \sum_{j \in [N]} \alpha_j \Phi(\boldsymbol{x}_j) \right\|_2^2 + C \sum_{i=1}^N \ell_h^2 \left(y_i \left\langle \sum_{j \in [N]} \alpha_j \Phi(\boldsymbol{x}_j), \Phi(\boldsymbol{x}_i) \right\rangle \right) \tag{14}$$

$$\equiv \min_{\boldsymbol{\alpha}} \ \frac{1}{2}\boldsymbol{\alpha}^\intercal \boldsymbol{G}\boldsymbol{\alpha} + C\sum_{i=1}^{N} \ell_h^2 \left(y_i \left\langle \boldsymbol{g}_i, \boldsymbol{\alpha}\right\rangle\right), \tag{15}$$

where $\boldsymbol{G}$ is the Gram matrix generated from K on our training data points $\{\boldsymbol{x}_i\}_{i=1}^N$.

(i) Implement gradient descent with backtracking linear search to optimize Eq. (15). (0.5/12)

(ii) Implement both Gaussian and polynomial kernels, and compare the classification performance on the breast cancer dataset with that of built-in Gaussian (i.e., RBF) and polynomial kernels with `SVC`. You may want to set up a small validation set to help you tune the hyperparameters. (1/12)

(d) One can also use SVM for regression, often called *support vector regression*, or SVR. In the hard-margin setting, we consider a predictor of the form $\langle \boldsymbol{w}, \Phi(\boldsymbol{x})\rangle + b$, where Φ is the nonlinear feature mapping we apply to the input. We require that the predictor makes at most ε error uniformly on all training samples, i.e., $|y_i - \langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i)\rangle - b| \le \varepsilon$ for all i. So the resulting optimization problem is

$$\min_{\boldsymbol{w}, b} \ \frac{1}{2}\|\boldsymbol{w}\|_2^2 \quad \text{s. t. } |y_i - \langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i)\rangle - b| \le \varepsilon \ \forall\, i. \tag{16}$$

Here obviously

$$|y_i - \langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i)\rangle - b| \le \varepsilon \Longleftrightarrow y_i - \langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i)\rangle - b \le \varepsilon, \ \langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i)\rangle + b - y_i \le \varepsilon. \tag{17}$$

To allow a bit of slackness, similar to the soft-margin SVM, one can introduce two groups of slackness variables: $\xi_i' s$ and ξ_i^*'s and consider the formulation

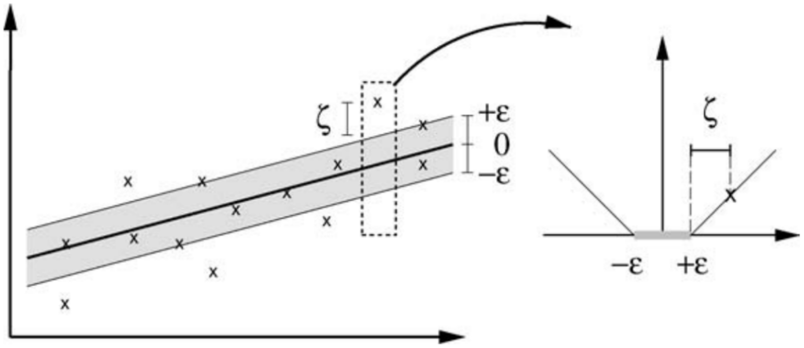

Figure 1: Illustration of SVR and the loss it uses. Figure taken from [SS04].

$$\min_{\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*} \ \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \quad \text{s. t. } y_i - \langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i)\rangle - b \le \varepsilon + \xi_i, \ \xi_i \ge 0 \\ \langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i)\rangle + b - y_i \le \varepsilon + \xi_i^*, \xi_i^* \ge 0 \ \forall i. \tag{18}$$

(i) Is Eq. (18) a convex optimization problem or not? Why? (0.5/12)

(ii) Show that Eq. (18) can be reformulated as an equivalent unconstrained problem

$$\min_{\boldsymbol{w},b} \frac{1}{2} \|\boldsymbol{w}\|_2^2 + C \sum_{i=1}^{N} \max\left(|y_i - \langle \boldsymbol{w}, \Phi\left(\boldsymbol{x}_i\right)\rangle - b| - \varepsilon, 0\right). \tag{19}$$

(0.5/12)

(iii) Let's omit the bias term b. How do we implement the kernel trick for SVR? (0.5/12)

Problem 2 (Learning theory) To apply the standard statistical learning theorems, it is crucial to estimate the complexity of the hypothesis class $\mathcal{H}$ under discussion.

(a) For $\mathcal{H}$ with finitely many elements, we only need to provide a reasonable upper bound to its cardinality $|\mathcal{H}|$.

(i) Consider the set of all conjunctions[2] of at most n Boolean literals (a literal is either x_i or the negation $\overline{x_i}$) from $x_1, \dots, x_n$, e.g., $x_1 \wedge \overline{x_3} \wedge x_4$. What's the cardinality of the set? (0.5/12)

(ii) Let X be the set of all Boolean functions $f : \{0,1\}^k \mapsto \{0,1\}$, and U be the set of all subsets of X. What's $|U|$? (0.5/12)

(b) Consider binary classification problems. A set of distinct points S is *schattered* by a hypothesis class $\mathcal{H}$ if every binary label assignment on the set can be realized by a certain $h \in \mathcal{H}$, in other words if $\mathcal{H}$ realizes all possible dichotomies of S. The VC dimension of $\mathcal{H}$ is the size of the largest set that can be shattered by $\mathcal{H}$. Hence, to prove $\mathcal{H}$ has a VC dimension of d, one needs to show that

- there exists a set of cardinality d that can be shattered by $\mathcal{H}$, *and*
- no set of cardinality $d+1$ can be shattered by $\mathcal{H}$.

Please answer the following questions.

(i) Let $\mathcal{B}$ be the set of all ℓ_2 ball functions on $\mathbb{R}^2$, i.e.,

$$\begin{aligned}\mathcal{B} \doteq & \left\{\boldsymbol{x} \mapsto \mathbb{1}\left\{\|\boldsymbol{x} - \boldsymbol{c}\|_2 \le r\right\} - \mathbb{1}\left\{\|\boldsymbol{x} - \boldsymbol{c}\|_2 > r\right\} : \boldsymbol{c} \in \mathbb{R}^2, r \in \mathbb{R}\right\} \\ & \bigcup \left\{\boldsymbol{x} \mapsto -\mathbb{1}\left\{\|\boldsymbol{x} - \boldsymbol{c}\|_2 \le r\right\} + \mathbb{1}\left\{\|\boldsymbol{x} - \boldsymbol{c}\|_2 > r\right\} : \boldsymbol{c} \in \mathbb{R}^2, r \in \mathbb{R}\right\}.\end{aligned} \tag{20}$$

What's the VC dimension of $\mathcal{B}$? Show your argument. (1/12)

(ii) Show that the VC dimension of the set of Boolean conjunction of at most n Boolean literals as discussed in (a)(i) is upper bounded by n. (1/12)

(c) For a set of training points $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ and a function $h \in \mathcal{H}$, we write $h(S) \doteq [h(\boldsymbol{x}_1), \dots, h(\boldsymbol{x}_N)]$. The empirical Rademacher complexity is defined as

$$\widehat{\mathcal{R}}_S(\mathcal{H}) = \frac{1}{N} \mathbb{E}_{\boldsymbol{r} \sim_{iid} \mathrm{Rad}} \left[\sup_{h \in \mathcal{H}} \langle \boldsymbol{r}, h(S)\rangle\right], \tag{21}$$

[2] https://en.wikipedia.org/wiki/Logical_conjunction

where Rad denotes the Rademacher distribution. Can you provide an upper bound for the empirical Rademacher complexity of the class of hyperplanes classifiers, i.e.,

$$\left\{\boldsymbol{x} \mapsto \operatorname{sign}\left(\langle\boldsymbol{w}, \boldsymbol{x}\rangle + b\right) : \boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}\right\}? \tag{22}$$

For your information, the VC dimension of this class is $d+1$, and so the contribution to the generalization error is $C\sqrt{\frac{d}{N}}$. The corresponding contribution as estimated by the empirical Rademacher complexity is $2\widehat{\mathcal{R}}_S\left(\mathcal{H}\right)$. (1/12)

References

[SS04] Alex J. Smola and Bernhard Schölkopf, *A tutorial on support vector regression*, Statistics and Computing **14** (2004), no. 3, 199–222.