EECS 545 – Machine Learning - Homework #4

Due: 5:00PM 11/01/2018

Homework submission via Gradescope as usual.

1) **Kernels (20 pts).**

(a) (4 pts) To what feature map Φ does the kernel

$$k(u,v) = (\langle u, v\rangle + 1)^3$$

correspond? Assume the inputs have an arbitrary dimension d and the inner product is the dot product.

(b) (12 pts) Let k_1, k_2 be symmetric, positive-definite kernels over $\mathbb{R}^D \times \mathbb{R}^D$, let $a \in \mathbb{R}^+$ be a positive real number, let $f : \mathbb{R}^D \to \mathbb{R}$ be a real-valued function and let $p : \mathbb{R} \to \mathbb{R}$ be a polynomial with *positive* coefficients. For each of the functions k below, state whether it is necessarily a positive-definite kernel. If you think it is, prove it; if you think it is not, give a counterexample.

(i) $k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z}) + k_2(\mathbf{x}, \mathbf{z})$

(ii) $k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z}) - k_2(\mathbf{x}, \mathbf{z})$

(iii) $k(\mathbf{x}, \mathbf{z}) = ak_1(\mathbf{x}, \mathbf{z})$

(iv) $k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z})k_2(\mathbf{x}, \mathbf{z})$

(v) $k(\mathbf{x}, \mathbf{z}) = f(\mathbf{x})f(\mathbf{z})$

(vi) $k(\mathbf{x}, \mathbf{z}) = p(k_1(\mathbf{x}, \mathbf{z}))$

(c) (4 pts) Prove one direction of an equivalence stated in the notes, namely, that if k is an inner product kernel, then k is a symmetric, positive definite kernel.

2) **Kernel Ridge Regression (20 pts).**

This problem examines the similarities and differences between KRR with and without offset.

(a) (3 pts) In KRR with offset, give a formula for the offset b using the kernel.

(b) (6 pts) This problem shows you how to compute the KRR with offset solution without a bunch of loops. The train-train kernel matrix is the $n \times n$ matrix $K = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]$. The "centered" train-train kernel matrix $\tilde{K}$ is the $n \times n$ matrix whose entries are $\langle \tilde{\Phi}(\boldsymbol{x}_i), \tilde{\Phi}(\boldsymbol{x}_j)\rangle$, where $\tilde{\Phi}(\boldsymbol{x}) := \Phi(\boldsymbol{x}) - \frac{1}{n}\sum_{\ell=1}^{n} \Phi(\boldsymbol{x}_\ell)$ and Φ is a feature map corresponding to the kernel k. This is the matrix that arises in KRR with offset. Calculation of $\tilde{K}$ is facilitated by the formula

$$\tilde{K} = K - KO - OK + OKO, \qquad (1)$$

where O is a square matrix with all entries equal to $1/n$. You should verify this fact but you don't need to turn your work in.

Now consider a test data set $\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_m$, and let K' be the $n \times m$ train-test matrix with entries $k(\boldsymbol{x}_i, \boldsymbol{x}'_j)$, and let $\tilde{K}'$ be the $n \times m$ centered train-test matrix, whose entries are $\langle \tilde{\Phi}(\boldsymbol{x}_i), \tilde{\Phi}(\boldsymbol{x}'_j) \rangle$. Determine a formula analogous to (1) for relating $\tilde{K}'$ to K'. Also, determine a formula for computing the predicted outputs on all the test points. Your formula should yield a column vector of length m.

Let's revisit the body fat data which we have seen earlier. We saw that a linear fit was reasonable, but now let's try a nonlinear fit.

Use the first 150 examples for training, and the remainder for estimating the mean squared error.

You will be asked to implement two variants of kernel ridge regression. For the next two problems please turn in

- the mean squared error on the training data
- the mean squared error on the test inputs
- the offset b in part **(c)** using your formula from **(a)**

(c) (3 pts) Implement kernel ridge regression *with* offset. Remember to report the offset in addition to the other requested items.

(d) (4 pts) Implement kernel ridge regression *without* offset. Comment on any differences in performance with respect to part **(c)**.

(e) (4 pts) Please submit your code.

Additional comments:

- Use the Gaussian kernel with parameters $\sigma = 15$ and $\lambda = 0.003$.
- The functions `dist2.m` and `dist2.py` have been uploaded to Canvas. These are useful for computing squared distances between a bunch of points.
- Part **(b)** will save you from having to write a bunch of loops. The commands `ones` and `repmat` in Matlab, or `np.ones` and `np.tile` in Python, may also be useful here.

3) **Support Vector Regression (20 pts).**

Support vector regression (SVR) is a method for regression analogous to the support vector classifier. Let $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, $i = 1, \ldots, n$ be training data for a regression problem.

In the case of *linear regression*, SVR solves

$$\begin{aligned}
\min_{\mathbf{w}, b, \boldsymbol{\xi}^+, \boldsymbol{\xi}^-} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{C}{n}\sum_{i=1}^{n}(\xi_i^+ + \xi_i^-) \\
\text{s.t.} \quad & y_i - \mathbf{w}^T\boldsymbol{x}_i - b \le \epsilon + \xi_i^+ \quad \forall i \\
& \mathbf{w}^T\boldsymbol{x}_i + b - y_i \le \epsilon + \xi_i^- \quad \forall i \\
& \xi_i^+ \ge 0 \quad \forall i \\
& \xi_i^- \ge 0 \quad \forall i
\end{aligned}$$

where $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$, $\xi^+ = (\xi_i^+, \ldots, \xi_n^+)^T$, and $\xi^- = (\xi_i^-, \ldots, \xi_n^-)^T$.

Here $\epsilon > 0$ is fixed.

a. (5pts) Show that for an appropriate choice of λ, SVR solves

$$\min_{\mathbf{w},b} \ \frac{1}{n}\sum_{i=1}^{n} \ell_\epsilon(y_i, w^T\boldsymbol{x}_i + b) + \lambda\|w\|_2^2$$

where $\ell_\epsilon(y,t) = \max\{0, |y-t| - \epsilon\}$ is the so-called *ϵ-insensitive loss*, which does not penalize prediction errors below a level of ϵ.

b. (8 pts) The optimization problem is convex with affine constraints, and therefore strong duality holds. Use the KKT conditions to derive the dual optimization problem in a manner analogous to the support vector classifier. As in the SVC, you should eliminate the dual variables corresponding to the constraints $\xi_i^+ \geq 0$, $\xi_i^- \geq 0$.

c. (4 pts) Explain how to kernelize SVR. Be sure to explain how to determine b^*.

d. (3 pts) Argue that the final predictor will only depend on a subset of training examples, and characterize those training examples.