COMS 4771: Machine Learning
Homework 4, due April 19.

In this homework, we will study kernelized forms of ridge regression and PCA.

1 Kernel Ridge Regression

We derive kernel ridge regression in a couple steps.

Step 1. Consider solving the ridge regression problem with a training set

$$S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n) \in \mathbb{R}^p \times \mathbb{R}\}.$$

Let $\hat{\boldsymbol{w}}_\lambda$ be the ridge regression solution with regularization parameter λ, i.e.

$$\hat{\boldsymbol{w}}_\lambda \ := \ \arg\min_{\boldsymbol{w} \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} (y_i - \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle)^2 + \lambda \|\boldsymbol{w}\|_2^2.$$

In class, we derived a closed form expression for $\hat{\boldsymbol{w}}_\lambda$ in terms of the design matrix $\boldsymbol{X}$ and response vector $\boldsymbol{y}$, defined as

$$\boldsymbol{X} \ = \ \begin{bmatrix} — & \boldsymbol{x}_1^\top & — \\ — & \boldsymbol{x}_2^\top & — \\ & \vdots & \\ — & \boldsymbol{x}_n^\top & — \end{bmatrix} \quad \text{and} \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

This closed form expression turns out to be

$$\hat{\boldsymbol{w}}_\lambda = (\boldsymbol{X}^\top \boldsymbol{X} + n\lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top \boldsymbol{y}.$$

We will derive an alternative form for $\hat{\boldsymbol{w}}_\lambda$. First, note that the above closed form expression comes from solving the optimality equation for $\hat{\boldsymbol{w}}_\lambda$, i.e.

$$(\boldsymbol{X}^\top \boldsymbol{X} + n\lambda \boldsymbol{I}_p) \hat{\boldsymbol{w}}_\lambda = \boldsymbol{X}^\top \boldsymbol{y}.$$

where $\boldsymbol{I}_p$ is the identity matrix of order p. The above equation can be rewritten as

$$n\lambda \hat{\boldsymbol{w}}_\lambda = \boldsymbol{X}^\top (\boldsymbol{y} - \boldsymbol{X} \hat{\boldsymbol{w}}_\lambda). \tag{1}$$

Now, define $\boldsymbol{\alpha} := \boldsymbol{y} - \boldsymbol{X} \hat{\boldsymbol{w}}_\lambda$.

Extra credit (weightage: 2%). Using equation (1) and the definition of $\boldsymbol{\alpha}$, show that $\boldsymbol{\alpha}$ satisfies the following equation:

$$\frac{1}{n\lambda}\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{\alpha} + \boldsymbol{\alpha} = \boldsymbol{y}. \tag{2}$$

Extra credit (weightage: 1%). Solve equation (2) for $\boldsymbol{\alpha}$, and use your solution in equation (1) to get the following alternative expression for $\hat{\boldsymbol{w}}_\lambda$:

$$\hat{\boldsymbol{w}}_\lambda \;=\; \frac{1}{n\lambda}\boldsymbol{X}^\top\left(\frac{1}{n\lambda}\boldsymbol{X}\boldsymbol{X}^\top + \boldsymbol{I}_n\right)^{-1}\boldsymbol{y} \;=\; \boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{X}^\top + n\lambda\boldsymbol{I}_n)^{-1}\boldsymbol{y}. \tag{3}$$

Step 2. Now suppose labeled points $(\boldsymbol{x}, y)$ are drawn from $\mathcal{X} \times \mathbb{R}$, where $\mathcal{X}$ is the feature space. Suppose you are also given a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which computes the inner product for a feature map $\boldsymbol{\phi} : \mathcal{X} \to \mathbb{H}$ where $\mathbb{H}$ is the reproducing kernel Hilbert space[1] for k. In other words,

$$k(\boldsymbol{x}, \boldsymbol{x}') \;=\; \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}')\rangle.$$

Suppose you have collected a training set of n examples

$$S \;=\; \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n) \in \mathcal{X} \times \mathbb{R}\}.$$

Let $\hat{\boldsymbol{w}}_\lambda$ be the solution of the ridge regression problem with regularization parameter λ when data points are mapped to $\mathbb{H}$ using the feature mapping $\boldsymbol{\phi}$. In other words, consider doing ridge regression using the transformed training set

$$S' = \{(\boldsymbol{\phi}(\boldsymbol{x}_1), y_1), (\boldsymbol{\phi}(\boldsymbol{x}_2), y_2), \ldots, (\boldsymbol{\phi}(\boldsymbol{x}_n), y_n) \in \mathbb{H} \times \mathbb{R}\}.$$

Mathematically, $\hat{\boldsymbol{w}}_\lambda$ can be defined as

$$\hat{\boldsymbol{w}}_\lambda \;:=\; \arg\min_{\boldsymbol{w}\in\mathbb{H}} \frac{1}{n}\sum_{i=1}^{n}(y_i - \langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle)^2 + \lambda\|\boldsymbol{w}\|_2^2.$$

Part 1 of assignment: (weightage: 4%) Given a test point $\boldsymbol{x} \in \mathcal{X}$, give the pseudocode for computing the prediction using the ridge regression solution, i.e. $\langle \hat{\boldsymbol{w}}_\lambda, \boldsymbol{\phi}(\boldsymbol{x})\rangle$. You may use equation (3) for this even if you haven't attempted to prove the equation in the extra credit question.

Hint: construct the design matrix $\boldsymbol{X}$ for the transformed training set S', and observe that the matrix $\boldsymbol{X}\boldsymbol{X}^\top$ can be computed using the kernel function k.

2 Kernel PCA

Next, we turn to kernel PCA. We use the connection between PCA and SVD to derive a kernelized form of PCA.

Let $\boldsymbol{X}$ be the design matrix, and let $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ be the SVD of $\boldsymbol{X}$. Recall from class that the following facts:

1. The columns of $\boldsymbol{U}$ are the eigenvectors of $\boldsymbol{X}\boldsymbol{X}^\top$, with eigenvalues given by the squares of the diagonal entries in $\boldsymbol{S}$.

[1] If the jargon "reproducing kernel Hilbert space" alarms you, just think of $\mathbb{H}$ as $\mathbb{R}^D$ for some $D \gg p$.

2. The rank ℓ PCA representation of the training set is given by the columns of

$$\boldsymbol{S}_\ell^{-1} \boldsymbol{U}_\ell^\top \boldsymbol{X} \boldsymbol{X}^\top .$$

Extra credit (weightage: 3%). Now let $\boldsymbol{X}$ be the design matrix for the transformed training set S'. Explain how you can compute $\boldsymbol{S}_\ell$ and $\boldsymbol{U}_\ell$ using the kernel function k without having to explicitly compute the mapping $\boldsymbol{\phi}$ for any training points.

Part 2 of assignment: (weightage: 2%) Suppose you have computed $\boldsymbol{S}_\ell$ and $\boldsymbol{U}_\ell$. Specify the dimensions of these two matrices. Explain how you can compute the rank ℓ PCA of the transformed training set using these matrices and the kernel function k without having to explicitly compute the mapping $\boldsymbol{\phi}$ for any training points.