Math Foundations of ML, Fall 2022

Homework #4

Due Wednesday Octomber 19 at 5:00pm ET

As stated in the syllabus, unauthorized use of previous semester course materials is strictly prohibited in this course.

1. Recall the bump basis $\{\phi_n(t)\}_{n=1}^N$ from Homework 2, Problem 3 (Linear approximation with "bump" functions), and its span $\mathcal{T}_N$ equipped with the standard inner product. The dual basis $\{\tilde{\phi}_n(t)\}_{n=1}^N$ can be used to find the sampling functions (reproducing kernel) for $\mathcal{T}_N$, as

$$f(\tau) = \sum_{n=1}^N \langle \boldsymbol{f}, \tilde{\boldsymbol{\phi}}_n \rangle \phi_n(\tau) = \left\langle \boldsymbol{f}, \sum_{n=1}^N \phi_n(\tau)\tilde{\boldsymbol{\phi}}_n \right\rangle = \langle \boldsymbol{f}, \boldsymbol{k}_\tau \rangle, \quad \text{where } \boldsymbol{k}_\tau = \sum_{n=1}^N \phi_n(\tau)\tilde{\boldsymbol{\phi}}_n.$$

   (a) Fix $N = 10$ and compute the dual basis vectors of the bump basis from Homework 2, Problem 3. That is, find $\tilde{\boldsymbol{\phi}}_1, \ldots, \tilde{\boldsymbol{\phi}}_{10}$ so that if

$$f(t) = \sum_{n=1}^{10} \alpha_n \phi_n(t),$$

   we can compute the $\{\alpha_n\}_{n=1}^N$ using

$$\alpha_n = \int_0^1 f(t)\tilde{\phi}_n(t)\, dt.$$

   Turn in a plot of each of the ten $\tilde{\phi}_n(t)$.

   (b) Take $N = 10$ and plot $k_\tau(t)$ as a function of t for $\tau = .371238$. Create an $\boldsymbol{f} \in \mathcal{T}_N$ by drawing the expansion coefficients $\boldsymbol{\alpha}$ at random (`alpha = randn(N,1);` in MATLAB), and verify that $\langle \boldsymbol{f}, \boldsymbol{k}_\tau \rangle = f(\tau)$.

   (c) Create an image of the kernel $k(s,t)$ for $(s,t) \in [0,1] \times [0,1]$ for the basis above — use at least a few hundred points for each of the arguments s and t. (In MATLAB you can display using `imagesc`.)

2. In this problem, we will solve a stylized regression problem using the data set `hw04p2_data.mat`. This file contains (noisy) samples of a function $f(t)$ for $t \in [0,1]$. In fact, the data points were generated by sampling the function

$$f_{\text{true}}(t) = \frac{\sin(12(t+0.2))}{t+0.2}$$

   at random locations then adding a random perturbation to the sample values. The sample locations are in the vector `T`, the sample values are in `y`. If you plot these, you will see that the samples are scattered more or less evenly across the interval. We are going to use kernel regression to form the estimate; in particular, we will use

$$k(s,t) = e^{-|t-s|^2/2\sigma^2}.$$

Last updated 0:07, October 8, 2022

(a) Compute the kernel regression estimate with $\sigma = 1/10$ and $\delta = 0.004$. Plot your estimate $\hat{f}(t)$ overlaid on the data and $f_{\text{true}}(t)$. Compute the *sample error*[1]

$$\text{sample error} = \left(\sum_{m=1}^{M} |y_m - \hat{f}(t_m)|^2 \right)^{1/2},$$

and the *generalization error*

$$\text{generalization error} = \left(\int_0^1 |\hat{f}(t) - f_{\text{true}}(t)|^2 \right)^{1/2}$$

for your estimate. Comment on why this choice of σ was a good one.

(b) Repeat part (a) with $\sigma = 1/2, 1/5, 1/20, 1/50, 1/100, 1/200$, producing plots, sample errors, and generalization errors for your estimates for each σ. Comment on how the number of data points we see would affect the right choice of σ.

3. Consider the set of bump basis vectors $\psi_1(t), \ldots, \psi_N(t)$, where

$$\psi_k(t) = g\left(t - k/N\right), \quad g(t) = e^{-200t^2} \tag{1}$$

Given a point t, define the nonlinear "feature map" as

$$\boldsymbol{\Psi}(t) = \begin{bmatrix} \psi_1(t) \\ \psi_2(t) \\ \vdots \\ \psi_N(t) \end{bmatrix}$$

Plot the feature map as a discrete set of coefficients[2] for $t = 1/3$ for $N = 10, 20, 50, 100, 200$. Compare to the radial basis kernel map

$$\boldsymbol{\Phi}_t(s) = k(s,t) = e^{-100|s-t|^2},$$

for $t = 1/3$ and $s \in [0,1]$. Discuss the relationship between kernel regression with a Gaussian radial basis function, and nonlinear regression using a basis of the form (1).

4. Let

$$\boldsymbol{A} = \begin{bmatrix} 1.01 & 0.99 \\ 0.99 & 0.98 \end{bmatrix}$$

(a) Find the eigenvalue decomposition of $\boldsymbol{A}$ by hand. Recall that λ is an eigenvalue of $\boldsymbol{A}$ if for some $u[1], u[2]$ (entries of the corresponding eigenvector) we have

$$\begin{aligned} (1.01 - \lambda)u[1] + 0.99u[2] &= 0 \\ .99u[1] + (0.98 - \lambda)u[2] &= 0. \end{aligned}$$

Another way of saying this is that we want the values of λ such that $\boldsymbol{A} - \lambda\mathbf{I}$ (where $\mathbf{I}$ is the 2×2 identity matrix) has a non-trivial null space — there is a

[1] Also called the "training error".

[2] In MATLAB, use `plot(1:N,Psit(1:N),'o')`.

nonzero vector $\boldsymbol{u}$ such that $(\boldsymbol{A} - \lambda\mathbf{I})\boldsymbol{u} = 0$. Yet another way of saying this is that we want the values of λ such that $\det(\boldsymbol{A} - \lambda\mathbf{I}) = 0$. Once you have found the two eigenvalues, you can solve the 2×2 systems of equations $\boldsymbol{A}\boldsymbol{u}_1 = \lambda_1\boldsymbol{u}_1$ and $\boldsymbol{A}\boldsymbol{u}_2 = \lambda_2\boldsymbol{u}_2$ for $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$.

Show your work above, but feel free to check you answer using MATLAB/numpy.

(b) If $\boldsymbol{y} = \begin{bmatrix}1 & 1\end{bmatrix}^{\mathrm{T}}$, determine the solution to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$.

(c) Now let $y = \begin{bmatrix}1.1 & 1\end{bmatrix}^{\mathrm{T}}$ and solve $Ax = y$. Comment on how the solution changed.

(d) Suppose we observe

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$$

with $\|\boldsymbol{e}\|_2 = 1$. We form an estimate $\tilde{\boldsymbol{x}} = \boldsymbol{A}^{-1}\boldsymbol{y}$. Which vector $\boldsymbol{e}$ (over all error vectors with $\|\boldsymbol{e}\|_2 = 1$) yields the maximum error $\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_2^2$?

(e) Which (unit) vector $\boldsymbol{e}$ yields the minimum error?

(f) Suppose the components of $\boldsymbol{e}$ are independent and identically distributed (i.i.d.) Gaussian random variables:

$$\boldsymbol{e} \sim \mathrm{Normal}(\mathbf{0}, \boldsymbol{I}).$$

What is the mean-square error $\mathbb{E}[\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_2^2]$?

(g) Verify your answer to part (f) in MATLAB/Python by taking $\boldsymbol{A}\boldsymbol{x} = \begin{bmatrix}1 & 1\end{bmatrix}^{\mathrm{T}}$, and then generating 10,000 different realizations of $\boldsymbol{e}$ using the `randn` command, and then averaging the results. Turn in your code and the results of your computation.

5. (a) Let $\boldsymbol{A}$ be a $N \times N$ symmetric matrix. Show that[3]

$$\mathrm{trace}(\boldsymbol{A}) = \sum_{n=1}^{N} \lambda_n,$$

where the $\{\lambda_n\}$ are the eigenvalues of $\boldsymbol{A}$.

(b) Now let $\boldsymbol{A}$ be an arbitrary $M \times N$ matrix. Recall the definition of the Frobenius norm:

$$\|\boldsymbol{A}\|_F = \left(\sum_{m=1}^{M}\sum_{n=1}^{N} |A[m,n]|^2\right)^{1/2}.$$

Show that

$$\|\boldsymbol{A}\|_F^2 = \mathrm{trace}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}) = \sum_{r=1}^{R} \sigma_r^2,$$

where R is the rank of $\boldsymbol{A}$ and the $\{\sigma_r\}$ are the singular values of $\boldsymbol{A}$.

(c) The *operator norm* (sometimes called the *spectral norm*) of an $M \times N$ matrix is

$$\|\boldsymbol{A}\| = \max_{\boldsymbol{x}\in\mathbb{R}^N,\ \|\boldsymbol{x}\|_2=1} \|\boldsymbol{A}\boldsymbol{x}\|_2.$$

[3] The trace of a (square) matrix is the sum of the elements on the diagonal: $\mathrm{trace}(\boldsymbol{A}) = \sum_{n=1}^{N} A[n,n]$.

(This matrix norm is so important, it doesn't even require a designation in its notation — if somebody says "matrix norm" and doesn't elaborate, this is what they mean.) Show that

$$\|\boldsymbol{A}\| = \sigma_1,$$

where σ_1 is the largest singular value of $\boldsymbol{A}$. For which $\boldsymbol{x}$ does

$$\|\boldsymbol{A}\boldsymbol{x}\|_2 = \|\boldsymbol{A}\| \cdot \|\boldsymbol{x}\|_2 \ \ ?$$

(d) Prove that $\|\boldsymbol{A}\| \leq \|\boldsymbol{A}\|_F$. Give an example of an $\boldsymbol{A}$ with $\|\boldsymbol{A}\| = \|\boldsymbol{A}\|_F$.