Course 67564 2022/23

Bayesian Machine Learning
**Exercise 3: Evidence and Kernels**

*Prof. Yair Weiss* *TA: Roy Friedman*

Deadline: December 22, 2022

# 1 Theoretical

## 1.1 Input-Specific Noise

So far, we have always assumed that the noise in linear regression is the same for each input, i.e.:

$$y\left(x_i\right)=\theta^T h\left(x_i\right)+\eta_i \qquad \forall\, i \quad \eta_i\sim\mathcal{N}\left(0,\sigma^2\right) \tag{1.1}$$

However, there is no reason to assume that this is the case; each sample could have a specific noise variance $\eta_i\sim\mathcal{N}\left(0,\sigma_i^2\right)$. This kind of setting makes sense if we assume that our observations are more noisy for different values of $x$.

This behavior is especially typical when the observed outputs span several scales of magnitude. As an example, consider the following setting: the $x$s are coordinates on earth and the $y$s are average house prices within a square kilometer around the coordinates. In some places, for instance the middle of the ocean, the typical house price is 0$ and we will have very very small variance (unless there are houses in the middle of some oceans which I am unaware of). On the other hand, in rural Costa Rica house prices could be in the range 15,000$-45,000$ so there might be a variance of around 5000$. Finally, in Tel Aviv the house prices could be in the range 3,000,000$-6,000,000$ with a variance of around 500,000$.

With these conditions, it doesn't seem to makes sense to model the variance in the middle of the ocean, rural Costa Rica and central Tel Aviv the same away. Notice that the variance in prices in Tel Aviv is bigger than the average house price in Costa Rica! Instead, we can think of the noise on the observations to be location specific:

$$\eta_i\sim\mathcal{N}\left(0,\sigma\left(x_i\right)^2\right) \tag{1.2}$$

which, for notational convenience, we will write as $\eta_i\sim\mathcal{N}\left(0,\sigma_i^2\right)$.

1. Given (known) input specific noise, the linear regression problem becomes:

   $$y\left(x_i\right)=\theta h\left(x_i\right)+\eta_i \qquad \eta_i\sim\mathcal{N}\left(0,\sigma_i^2\right) \tag{1.3}$$

   Let $y=\left(y\left(x_1\right),\cdots,y\left(x_N\right)\right)^T$. Write down the likelihood $p\left(y|\theta\right)$ in the case of input-specific noise as described above

2. What is the MLE solution for linear regression with input-specific noise?

3. Let $\theta\sim\mathcal{N}\left(\mu_0,\Sigma_0\right)$. What is the posterior distribution in the case of input-specific noise?

## 1.2 Product of Kernels

In this part of the exercise, we will prove one of the clauses you have seen in the recitation regarding kernels. Specifically, if $k_1\left(x,y\right)$ and $k_2\left(x,y\right)$ are valid kernels, then:

$$k_\times\left(x,y\right)=k_1\left(x,y\right)\cdot k_2\left(x,y\right) \tag{1.4}$$

is also a valid kernel. To prove this we will use the fact that valid kernels are positive semi-definite.

You may find the following identities helpful (but don't have to use them):

$$x^T Ay = \text{trace}\left[x^T Ay\right] = \text{trace}\left[yx^T A\right] \tag{1.5}$$

$$\text{trace}\left[AB\right] = \sum_i \left[AB\right]_{ii} = \sum_i \sum_j A_{ij}B_{ji} \tag{1.6}$$

where $x$ and $y$ are vectors while $A$ and $B$ are matrices.

4. Let $k\left(x, y\right)$ be a valid kernel and suppose that $K$ is the kernel's Gram matrix over some finite set of points $\{x_i\}_{i=1}^N$, such that $K_{ij} = k\left(x_i, x_j\right)$. Show that for any finite set of $N$ points, there exists some function $f : \mathcal{X} \to \mathbb{R}^N$ such that:

$$k\left(x_i, x_j\right) = f^T\left(x_i\right) f\left(x_j\right) \tag{1.7}$$

   where $\mathcal{X}$ is the space of the points $x_i$. Using this fact, show that:

$$k_1\left(x, y\right) \cdot k_2\left(x, y\right) = \sum_i \sum_j g_i\left(x\right) f_j\left(x\right) f_j\left(y\right) g_i\left(y\right) \tag{1.8}$$

   where $k_1\left(\cdot, \cdot\right)$ and $k_2\left(\cdot, \cdot\right)$ are valid kernels, and some functions $f, g : \mathcal{X} \to \mathbb{R}^N$, where $f_i\left(x\right)$ denotes the $i$th index of the output of $f\left(x\right)$

5. Conclude that $k_\times\left(x, y\right) = k_1\left(x, y\right) \cdot k_2\left(x, y\right) = h^T\left(x\right) h\left(y\right)$ for some function $h\left(\cdot\right)$, thereby proving that $k_\times\left(\cdot, \cdot\right)$ is a valid kernel

## 1.3 Kernel Functions

For each of the following functions, prove whether it is a valid kernel or not:

6. $k\left(x, y\right) = e^{\beta\|g(x)-g(y)\|^2}$ for any function $g\left(\cdot\right)$ and $\beta > 0$
7. $k\left(x, y\right) = k_1\left(x, y\right) - k_2\left(x, y\right)$ for any two valid kernels $k_1\left(\cdot, \cdot\right)$ and $k_2\left(\cdot, \cdot\right)$
8. $k\left(x, y\right) = k_a\left(x_a, y_a\right) + k_b\left(x_b, y_b\right)$ for any two valid kernels $k_a\left(\cdot, \cdot\right)$ and $k_b\left(\cdot, \cdot\right)$, where $x = \begin{pmatrix} x_a \\ x_b \end{pmatrix}$ and $y = \begin{pmatrix} y_a \\ y_b \end{pmatrix}$
9. $k\left(x, y\right) = \sqrt{\ell\left(x\right)^T \ell\left(y\right)}$ for any function such that $\forall i, x \quad \left[\ell\left(x\right)\right]_i \geq 0$

## 1.4 Evidence in the Dual Space

Recall that, given a kernel function $k\left(x, x'\right)$, the dual problem has the equivalent form:

$$f_\alpha\left(x\right) = \boldsymbol{k}^T\left(x\right)\alpha + \eta \tag{1.9}$$

where $\eta \sim \mathcal{N}\left(0, \sigma^2\right)$, $\boldsymbol{k}\left(x\right) = \begin{pmatrix} k\left(x, x_1\right) \\ \vdots \\ k\left(x, x_N\right) \end{pmatrix} \in \mathbb{R}^N$ and the points $\{x_i\}_{i=1}^N$ are the training points. Additionally:

$$\alpha \sim \mathcal{N}\left(0, K^{-1}\right) \tag{1.10}$$

where $K$ is the Gram matrix of the kernel $k\left(\cdot, \cdot\right)$, defined as:

$$K_{ij} = k\left(x_i, x_j\right) \tag{1.11}$$

10. Show that the evidence of the dual problem is given by:

$$p\left(y|k\left(\cdot, \cdot\right), \sigma^2\right) = \mathcal{N}\left(y \,|\, 0,\, K + I\sigma^2\right) \tag{1.12}$$

# 2 Practical

In this exercise, we will try to choose which basis function and sample noise to use in our models by calculating the evidence function. Because of numerical stability, what we will actually want to do is to calculate the log-evidence score, which is just the log of the evidence score.

For this exercise, you will also need to fit Bayesian linear regression models to data. You can either use your implementation from the previous exercise, or use the implementation supplied in `ex3_utils.py`. The utils script also contains an example of how to use the supplied implementation to predict and to find the standard deviation of the MMSE prediction.

1. Implement a function that receives the prior and sample noise of a Bayesian linear regression model and returns the log-evidence score for the given model, defined as[1]:

$$\log p\left(y \mid \mu_\theta, \Sigma_\theta, \sigma^2\right) = \frac{1}{2}\log\frac{\left|\Sigma_{\theta|\mathcal{D}}\right|}{\left|\Sigma_\theta\right|} - \frac{1}{2}\left[\left(\mu_{\theta|\mathcal{D}} - \mu_\theta\right)^T \Sigma_\theta^{-1}\left(\mu_{\theta|\mathcal{D}} - \mu_\theta\right) + \frac{1}{\sigma^2}\|y - H\mu_{\theta|\mathcal{D}}\|^2 + N\log\sigma^2\right] - \frac{p}{2}\log 2\pi \tag{2.1}$$

   where $N$ is the number of points and $p$ is the number of basis functions

## 2.1 Evidence for Artificial Functions

In this section we will look at the behavior of the evidence function on artificial data. The data we will use will be functions with some added noise. We will then evaluate various possible basis functions using the evidence function.

The functions we will use are:

- $f_1(x) = x^2 - 1$
- $f_2(x) = -x^4 + 3x^2 + 50\sin\frac{x}{6}$
- $f_3(x) = \frac{1}{2}x^6 - 0.75x^4 + 2.75x^2$
- $f_4(x) = \frac{5}{1+e^{-4x}} - \begin{cases} x & x-2>0 \\ 0 & \text{otherwise} \end{cases}$
- $f_5(x) = \cos 4x + 4\left|x-2\right|$

The noise we will add to these functions will be $\eta \sim \mathcal{N}\left(0, I\sigma^2\right)$ with a variance of $\sigma^2 = 0.25$. For simplicity, the prior we will use in each of the models will be $\theta_R \sim \mathcal{N}\left(0, I\alpha\right)$ with $\alpha = 5$. For each of the functions $f_1(\cdot)$, $f_2(\cdot)$, $f_3(\cdot)$, $f_4(\cdot)$ and $f_5(\cdot)$:

2. Sample 500 points equally spaced in the range $x \in [-3, 3]$, calculate the function on these points and add random noise sampled from $\eta$

3. Calculate the log-evidence for 10 models with polynomial basis functions with degrees $d = [2, \cdots, 10]$ and the matching prior $\theta_R$. Plot the log-evidence for each of these models as a function of the degree of the polynomial basis functions. Which model has the highest evidence?

4. Fit Bayesian linear regression to the best and worst model, according to the evidence function, and plot them (together with the confidence intervals). Does the evidence match your choice?

[1] This is simply the log of the definition we saw in class and in recitation 5

### 2.2 Estimating the Sample Noise for Temperature Prediction

In this part of the exercise, we will try to find the sample noise of the temperatures data set from the previous exercise. To do this, we will calculate the evidence of out model under different sample noises. In the file `temp_prior.npy` you are supplied with the prior $\theta_7$ for a 7th order polynomial fitted to the temperatures data set from the previous exercise. An example of how to load this can be found in the supplied utilities script `ex3_utils.py`.

5. Calculate and plot the log-evidence score for 100 different models using sample noises equally spaced in the range $\sigma^2 \in [0.05, 2]$ on the temperatures of the first half of November 16 2020, which can be found in the file `nov162020.npy`
6. Plot the log-evidence score for each of the models as a function of the sample noise. Which sample noise has the highest evidence?
7. Does this mean that this is the true sample noise of the original measurements? Explain your answer

## 3 Submission Guidelines

Submit a single zip file named "`ex3_<YOUR ID>.zip`". This file should contain your code, along with an "`ex3.pdf`" file in which you should write your answers to the theoretical part and add the figures/text for the practical part. Please write readable code, as the code will also be checked manually (and you may find it useful in the following exercises). In the submitted code, please make sure that you write a basic main function in a file named "`ex3.py`" that will run (without errors) and produce all of the results that you showed in the pdf of answers that you submitted. The only packages you should use are `numpy, scipy` and `matplotlib`. You may also reuse code from your previous exercise in order to answer the questions in this exercise, if needed.

In general, it is better if you type your homework, but if you prefer handwriting your answers, please make sure that the text is readable when you scan it.

Part of your assignment will be graded by submitting your answers through Moodle, at this link. In each of the questions, write the answer to the corresponding question for grading. These answers will be graded automatically, so write only numeric values where needed.

## 4 Supplementary Code

In the file `ex3utils.py` you can find an example of how to load the supplied data as well as a few helper functions. You can use this code as you see fit, and change any part of it that you want, just be sure to submit it as well if you change it. Finally, we have also supplied an outline code which you can use to get started in `ex3.py`. You don't have to use the format we outlined, but your code must run without errors and you must submit the plots required in the exercise description.

Good luck!