Course 67564 2021/22

BAYESIAN MACHINE LEARNING

Exercise 4: Gaussian Processes

Prof. Yair Weiss *TA: Roy Friedman*

Deadline: December 26, 2021

1 Theoretical

1.1 Effects of Bandwidths

In this section we will look at the effects of the bandwidth parameter β on the prior induced by the Gaussian process. Consider the following Gaussian process priors:

- $p_1(f)$: $f \sim \mathcal{GP}\left(0, \exp\left[-\beta\|x-x'\|^2\right]\right)$ where $\beta = \frac{1}{\alpha}$
- $p_2(f)$: $f \sim \mathcal{GP}\left(0, \exp\left[-\beta\|x-x'\|^2\right]\right)$ where $\beta = \alpha$

1. Let $\{x_i\}_{i=1}^M$ be M randomly chosen 20-dimensional vectors in the unit sphere. Define $f_i = f(x_i)$ and $\boldsymbol{f} \triangleq (f_1, \cdots, f_M)^T$. What is the distribution of $\boldsymbol{f}$ under the two prior as $\alpha \to \infty$?
2. Consider the 2^M possible vectors $\boldsymbol{f}$ such that $\forall i\ f_i \in \{-1, 1\}$, under the same data set $\{x_i\}_{i=1}^M$ as before. Let $\alpha \to \infty$:
   (a) Show that under p_1 there are two vectors that are equally likely, while all others have vanishing probability
   (b) Show that under p_2 all vectors have the same probability
3. Find an analytical form for the evidence of a vector $\boldsymbol{f}$ under p_1 and p_2 at the limit $\alpha \to \infty$.
   (c) Let $\boldsymbol{f}_1$ be a vector of all ones and suppose we are using Bayesian model selection to choose between p_1 and p_2. Which model will be selected given $\boldsymbol{f}_1$?
   (d) Let $\boldsymbol{f}_2$ be a vector whose even components are 1 and odd components are -1. Which model will be selected given $\boldsymbol{f}_2$?

1.2 Constant Kernel

Consider the function $k(x, y) = 1$.

4. Show that $k(\cdot, \cdot)$ is a valid kernel and can be obtained as the dual of a regression problem of the form $f_\theta(x) = h^T(x)\theta$ where $h(\cdot)$ are some basis functions. What is the explicit form of the primal problem?
5. Consider $f_\alpha(x) = \sum_i \alpha_i k(x, x_i)$ with $\alpha = \alpha_{\text{MMSE}} = \left(K + \sigma^2 I\right)^{-1} y$ where K is the Gram matrix of $k(\cdot, \cdot)$. Give an explicit form for each α_i
6. Assume we have a data set of pairs $\{x_i,\ g(x_i) + \eta_i\}_{i=1}^M$ of size M where:

$$g(x) = \text{sign}(x) \triangleq \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases} \tag{1.1}$$

with $\eta_i \sim \mathcal{N}\left(0, \sigma^2\right)$ and x uniformly distributed in the range $x \in [-1, 1]$. What does the training error[1]of $f_\alpha(\cdot)$ converge to as the number of training examples goes to infinity and $\sigma^2 \to 0$?

[1] The training error is the average squared error between the real train samples and their predicted values

7. Consider the same data set as in the previous section. Suppose we use Gaussian process regression with an RBF kernel. What will the training error converge to as the number of training examples goes to infinity and $\sigma^2 \to 0$ in this case?

2 Practical

2.1 GP Priors

In this part of the assignment we will look at the effects different kernels have on Gaussian processes. Consider the following kernels:

- Laplacian kernel: $k_1(x, x') = \alpha \cdot e^{-\beta \|x-x'\|_1}$ with $\alpha, \beta > 0$
- RBF kernel: $k_2(x, x') = \alpha \cdot e^{-\beta \|x-x'\|^2}$ with $\alpha, \beta > 0$
- Gibbs' kernel: $k_3(x, x') = \sqrt{\frac{2\ell(x)\ell(x')}{\ell^2(x)+\ell^2(x')}} e^{-\frac{\|x-x'\|^2}{\ell^2(x)+\ell^2(x')}}$ with $\ell(x) = \alpha \cdot e^{-\beta\|x-\delta\|^2} + \gamma$ for $\alpha, \beta, \gamma > 0$ and δ is any vector
- Neural network kernel[2]: $k_4(x, x') = \alpha \cdot \frac{2}{\pi} \sin^{-1}\left(\frac{2\beta\left(x^T x'+1\right)}{\sqrt{(1+2\beta(1+x^T x))(1+2\beta(1+x'^T x'))}}\right)$ for $\alpha, \beta > 0$

1. Implement a Gaussian process model that receives as its input a user-specified kernel function and the sample noise[3]

2. For each of the kernels described above, choose 3 different parameter settings while keeping the sample noise at $\sigma^2 = 0.05$. For each of these settings, plot the mean and confidence interval of the prior in the interval $x \in [-5, 5]$. Sample 5 functions from the prior and plot them together with the confidence interval of the prior[4]

3. Consider the following 5 data points:

x	-2	-1	0	1	2
y	-2.1	-4.3	0.7	1.2	3.9

   Calculate the posterior $\boldsymbol{f}|\left\{(x_1, y_1), \cdots, (x_5, y_5)\right\}$ and plot the posterior mean and confidence intervals in the range $x \in [-5, 5]$ for each parameter setting chosen in the previous question. Sample 5 functions from the posterior and plot them together with the posterior mean

The purpose of the above two questions is to make you play around with the kernels a bit. As you probably noticed, the last two kernels are a bit weird - try to play around with them and get them to act like you want them to. You'll see that they are pretty expressive and can form rather weird functions. The Gibbs kernel is a type of generalized RBF kernel (if you set α to be pretty low and γ to be pretty high, then γ will act like the typical length-scale of the RBF) while the NN kernel is what you would get if you try to make predictions using a (fully connected) neural network with infinitely-wide layers and an error-function (sigmoid) activation function (if you're interested, read more about it in Rasmussen and Williams). Notice that in the supplied skeleton for your solution, we already did (most) of the heavy lifting for the plotting, so all you have to do is set the kernel parameters you want to see and run the code. We *really* recommend that you play around with the kernels, this will give you more of an intuition than anything we can show in class.

4. For the data points defined above, plot the evidence function of the RBF kernel with 101 βs evenly spaced in the range $\beta \in [.1, 15]$ and noise $\sigma^2 = 0.15$ as a function of β. Which value of β has the highest evidence score? Plot the points, posterior mean and confidence interval of the best, worst and median βs according to the evidence

[2] See section 4.2.3 of Rasmussen and Williams for more information

[3] See Algorithm 2.1 on page 37 of Rasmussen and Williams for pseudo-code of a numerically stable version

[4] You can use the visualization tool from Moodle as a basic guideline for how these plots should look, although most of the plotting work is already there in the skeleton

2.2 Random Fourier Features

In class we saw that the RBF kernel can be approximated by random features given by:

$$k\left(x,y\right) \propto e^{-\frac{\beta}{2}\|x-y\|^2} \approx \frac{1}{M}\sum_{i=1}^{M}\cos\left(\omega_i^T x + b_i\right)\cdot\cos\left(\omega_i^T y + b_i\right) \tag{2.1}$$

$$= \begin{pmatrix} \frac{\cos\left(\omega_1^T x + b_1\right)}{\sqrt{M}} \\ \vdots \\ \frac{\cos\left(\omega_M^T x + b_M\right)}{\sqrt{M}} \end{pmatrix}^T \begin{pmatrix} \frac{\cos\left(\omega_1^T y + b_1\right)}{\sqrt{M}} \\ \vdots \\ \frac{\cos\left(\omega_M^T y + b_M\right)}{\sqrt{M}} \end{pmatrix} \triangleq h_M^T\left(x\right)h_M\left(y\right) \tag{2.2}$$

where $\forall i \quad \omega_i \sim \mathcal{N}\left(0, I\beta\right)$ and $b_i \sim \mathcal{U}\left(0, 2\pi\right)$. In this exercise you will implement the random Fourier features and see the effects of a small number of random features versus using many features.

For this exercise, you will also need to fit Bayesian linear regression models to data. You can either use your implementation from exercise 2, or use the implementation supplied in ex4_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 (which was simply copied from exercise 3).

Consider the function:

$$f\left(x\right) = \frac{1}{2}\sin\left(3x\right) - \left|0.75\cdot x\right| + 1 \tag{2.3}$$

5. Fit a GP regression model on 100 points of the function $f\left(\cdot\right)$ in the range $x \in [-3, 3]$ using an RBF kernel where $\beta = 2$, $\alpha = 1$ and the noise is $\sigma^2 = 0.25$. What is the average squared error of this fit? Plot a scatter of the points as well the posterior mean and confidence intervals

6. Implement a function that returns M random Fourier features as defined in equation 2.2

7. For $M = [1, 2, 5, 15, 25, 50, 75, 100]$, fit a Bayesian linear regression with with M random Fourier features to the function $f\left(\cdot\right)$ on the same range as above, with $\beta = 2$ and $\sigma^2 = 0.25$. The prior you should use when fitting the linear regression is $\theta \sim \mathcal{N}\left(0, IM\right)$. Plot the average squared error of these fits as a function of M. Plot the MMSE and confidence interval of $M = [1, 5, 15, 100]$ on top of the data points

3 Submission Guidelines

Submit a single zip file named "ex4_<YOUR ID>.zip". This file should contain your code, along with an ”ex4.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 "ex4.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 ex4utils.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 `ex4.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!