Contents

0 General Notes

- In general reset the kernel and rerun the notebook to avoid overloading variables and to make sure that every cell runs.
- Don't remove the given cells!! Next time we will subtract points for that as it breaks the grading. You can add your own cells, but remember to put them before the test cells.
- Make clean plots that are easy to read. This includes adding labels and description of the axis as well as a title if applicable. Make sure labels are easy to read and don't overlap with your plots and choose color schemes that are easy to understand.
- Use Markdown / Latex for your answers if it includes variables and equations
- For the open questions: If you think you mentioned all the aspects but did not get all the points it is most likely due to the fact that you only named them and the explanation was not detailed enough or was missing aspects.
- Stay within the given length for the answers. In the future we will penalize if you don't follow them.
- In the following parts we will list the most frequent mistakes, sorted by tasks.

1 Polynomial Regression

1.1 Generate periodic data

1.1.1

- Using wrong standard deviation (i.e. variance instead of std) (-2 points)

1.1.2

- Using wrong range (from 0 to n-1 instead of n) (-1 point)

1.2 Polynomial regression

1.3 Plot

1.3.1

- See Figure 1
- 1 point if there are 4 subplots
  1 point if the sines are plotted correctly
  1 point if the target data is plotted correctly
  1 point if the predictions are plotted correctly
  1 point if everything is there
- In general, if the plot is noticeably different from the plot in the book, you won't get the maximum score. Common mistakes include:
  - plotting the estimated function as a sequence of line segments (too high granularity, sample at least 100 data points to plot a function)
  - plotting points instead of lines for functions
  - plotting everything in one plot
  - y-axis exploding (and thus graph shrinking) in the last plot due to overfitting polynomial
- If the reference plot is two by two please don't plot a one by four (we did not subtract points for now but please follow that next time).
- Also, please, don't set the reference point of the plot. This leads to cropped figures.

1.4 Regularized linear regression

1.4.1

1.4.2

- Any solution that explains the behavior is fine. For example, if you consider the test case with $t = 0.3 * x + 2.5$.
- The values in $\mathbf{\Phi}$ do not change since the design matrix only depends on x and M. (2 point)
- The values in $\boldsymbol{w}$ change (1 point) in order to minimize the new loss function. In particular, without normalization the values for terms in perfectly match the coefficients of the equation for $t = f(x)$, namely $[2.5, 0.3, 0]$.When adding the regularization, we want to both solve the linear regression and minimize the norm of parameters. For this reason, the highest value (2.5, which is the most influent in the sum of squares) decreases, while the other two values (which contribute much less to the regularization term) slightly increase to maintain a good fit in the linear regression.(2 point)
- Common mistakes: describe how small/big lambdas changes the regularization term, but not what happens with $\lambda = 0$. We are asking changes between with and without regularization
- Describing the effect on the overfitting / underfitting and describing the graphs only also does not relate with the question.
- Many people didn't say that the design matrix stays the same (which should be quite easy to point out)

- Remark: a very large number of people described (all) the weights as decreasing towards zero when adding regularization. Although this is strictly not accurate (some weights very close to zero before regularization may increase after regularization to make up for higher weights shrinking, resulting in a little larger regularization error but at the same time compensating with a decrease in the regression error), we don't penalize this answer as long as the rest of it is correct. The main problem here appears to be the understanding of what happens with the regularization term $\boldsymbol{w}^T\boldsymbol{w}$. Decreasing this term does not mean that all the elements must necessarily decrease. Consider the following simple example: Let $w = [2, 0, 0]^T$. Then we get:

$$\boldsymbol{w}^T\boldsymbol{w} = [2, 0, 0] \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} = 4$$

if we now change the vector to $w' = [0, 0, 1]^T$ (decreasing the first element but increasing the last one) we get

$$\boldsymbol{w}'^T\boldsymbol{w}' = [0, 0, 1] \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 1$$

and therefore $\boldsymbol{w}^T\boldsymbol{w} > \boldsymbol{w}'^T\boldsymbol{w}'$ but $w_3 < w'_3$.

- Make sure to answer all the points in the question. Especially ' Why is this happening?'

1.5 Model selection by cross-validation

1.6 Why grid search

1.6.1

- Anything that touches the following basic idea was accepted (2 points): if we would optimize one parameter at a time, we would only find the best value for p_1 given p_2, and subsequently the best value for p_2 given the best p_1 for the initial p_2. (1 more point if explained well enough, for example discussing the concept of correlation/independence of variables or explicitly saying that the optimization of the second parameter is biased because of conditioning on the first parameters)
- Independence arguments should technically distinguish between independence of the two variables and independence of two variables given a latent variable (we did not subtract points for missing the distinction but keep it in mind).
- If you mention 'interaction between the parameters' without explaining what exactly they mean and why this interaction can't be understood with sequential search we subtract. (-1 point)
- The argument that sequential search would be more optimal for high-dimensional data is not necessarily true.

1.6.2

- 1 point for naming the algorithm
  1 for a brief explanation
- Examples of possible answers: Random Search, Bayesian Optimization, Gradient-Based Optimization, Evolutionary Optimization, ...

1.7 Plot best cross-validated fit

- See Figure 2
- 1 input is shown correctly (N datapoints must be = 10!)
  2 sine is shown correctly
  2 prediction is shown correctly (including the optimal values for lambda and M)
  -1 point if the functions are plotted correctly but using segments of line instead of smooth curves
- We don't penalize if approximated curves are off as this was already penalized in the cross-validation section.

2 Bayesian Linear (Polynomial) Regression

2.1 Sine 2

2.1.1

- If you sort x before applying normal distribution but everything else is correct, you get full points as this doesn't break any logic afterwards.

2.1.2

- Using variance instead of std. (-1 point)

2.1.3

- Using random variates (np.rvs) instead of normal np.random.normal. Please, don't complicate tasks which can be kept simple. (-1 point)
- If you specifically wrote a function don't use built-in functions (i.e. by using np.poly1d). (-1 point)

2.1.4

- If function is correct but results are different it might be due to the iterative filling of the array (in a for-loop). We do not penalize that.

2.2 Compute Posterior

2.2.1

- If any of the formulas are correct, add half of the points even if the tests are not passed.

2.3 Prediction

2.3.1

- A lot of people return std instead of variance (-2 points instead of -4)

2.4 Plot Predictive Distribution

2.4.1

- See Figure 3
- 1 point if sine is plotted correctly
  1 point if dataset is plotted correctly
  1 point if the predictive distribution is correct
  2 points if the variance is plotted correctly
  -1 point if everything's there but there is no legend/specification of what function is what
- See Figure 4
- 2 points if 100 polynomials are plotted
  3 points if the polynomials are 4th order and seem to fit a sine
  -1 point if the functions are plotted correctly but using segments of line instead of smooth curves
  -1 point if using np.poly1d

2.5 Additional questions

2.5.1

- 3 points if you identify that it depends on the noise in the dataset,
  2 points if you actually write down/explain the mathematical relationship between precision and variance
  -1 point if precision is given without sufficiently explicit connection to variance (i.e. beta corresponds to variance)
  -1 point if the reason to use usch beta is generally wrong (i.e. it prevents overfitting) even if formula is correct

- In the present case, the noise in the data is known and was generated using $\sigma = 0.25$. Moreover, the precision β for gaussian noise is defined as $\frac{1}{\sigma^2}$, and thus, in order to represent the noise in the data the best choice for β is $\beta = \frac{1}{\sigma^2} = \frac{1}{0.25^2}$.

2.5.2

- Basis functions are fixed before the training dataset is observed. The number of basis functions needs to grow rapidly, often exponentially, with the dimensionality D of the input space.
- 3 points if one of the two issues is pointed out, 5 for both/ a second issues
  -1 point if any of the problems is not well explained and just named

A Reference Figures

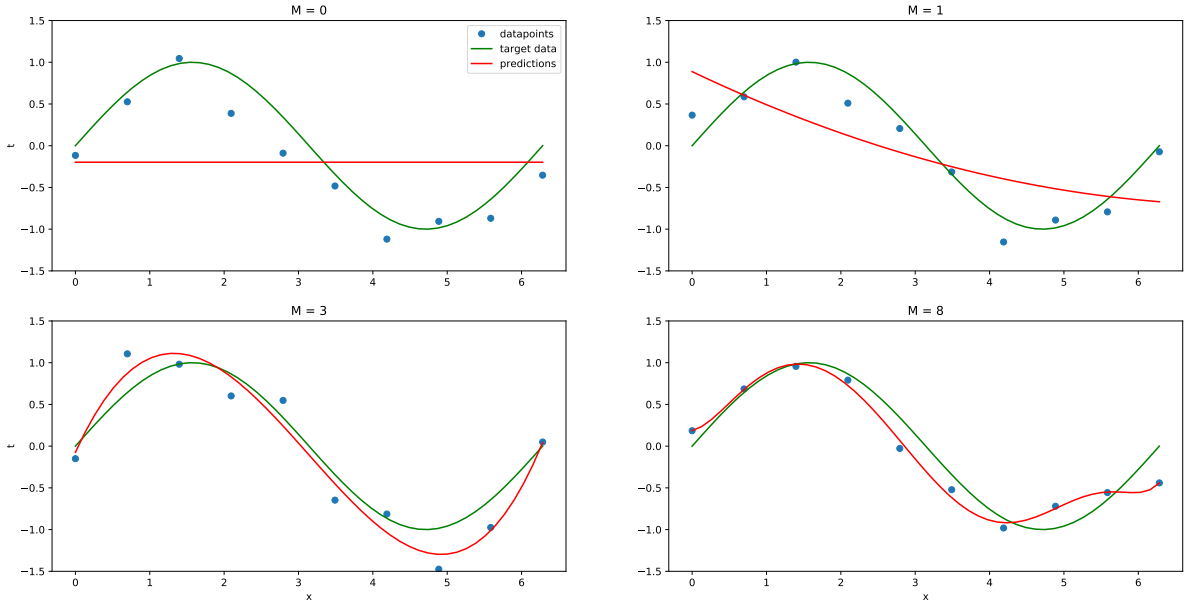

Fig. 1: Question 1.3.: Polynomial regression

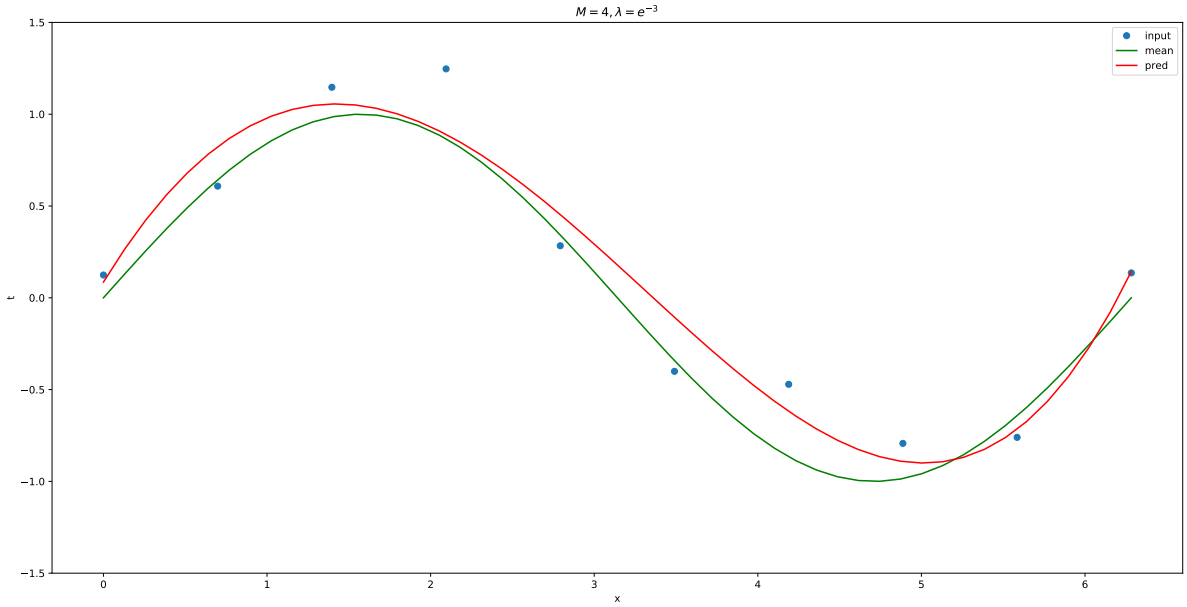

Fig. 2: Question 1.7.: Plot best cross-validated fit

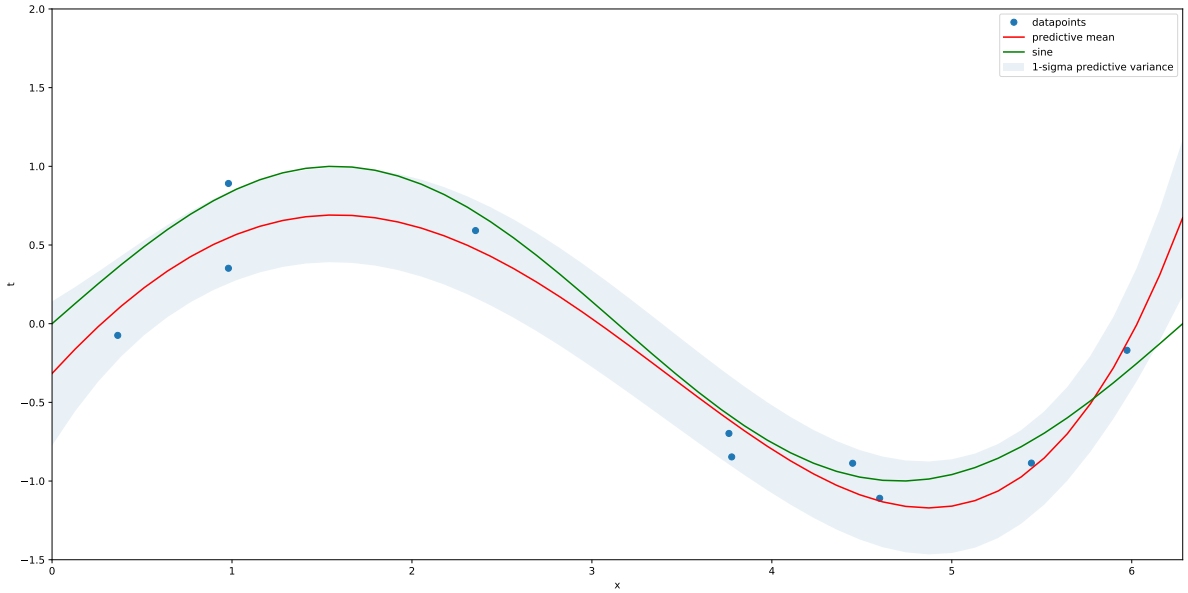

Fig. 3: Question 2.4.a): Plot predictive distribution

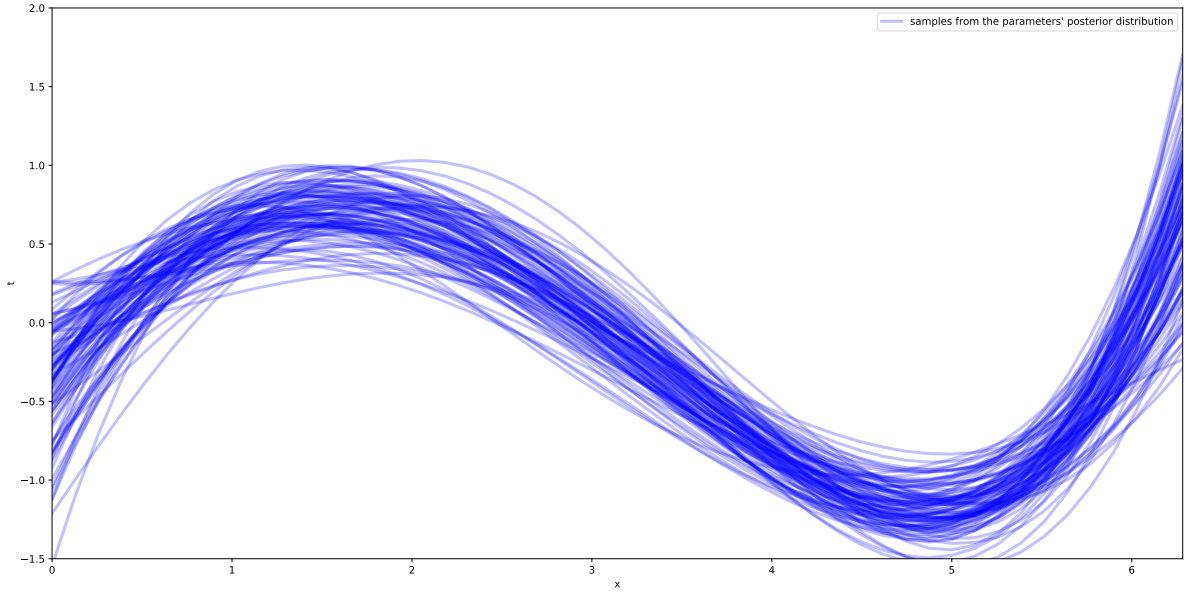

Fig. 4: Question 2.4.b): draw 100 samples from the parameters' posterior distribution