SS 2025
© Prof. Georgia Koppe

Course Coordinator: Andre Herz
`andre.herz@iwr.uni-heidelberg.de`

Machine Learning Essentials

Exercise Sheet 08

Due: 30.06.2025 11:15

This sheet is about **Gaussian linear models**. We will use a probabilistic perspective to highlight how Gaussian models reduce probabilistic inference to linear algebra - both in the frequentist and Bayesian setting. We explore a **fully Bayesian treatment of linear regression** which allows us to make predictions that reflect both parameter uncertainty and observation noise, leading to robust uncertainty estimates, especially when data is scarce. Applying a **basis function expansion** and the **kernel trick** on this model leads to a direct connection to Gaussian Process regression. We'll see how the **weight-space view** gives rise to Bayesian linear regression, whereas the **function-space view** induces **Gaussian Processes**.

Regulations

Please submit your solutions via Moodle in teams of **3** students. The coding tasks must be completed using the template `MLE25_sheet08.ipynb`. Each submission must include **exactly** one file:

- Upload a `.pdf` file containing both your Jupyter notebook and solutions to analytical exercises.

The Jupyter notebook can be exported to pdf by selecting **File** → **Download as** → **pdf** in JupyterLab. If this method does not work, you may print the notebook as a pdf instead. Your analytical solutions can be either scanned handwritten solutions or created using LaTeX.

Exercise 1: Linear Regression is Gaussian Inference

Consider the Gaussian linear model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}\big(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_n\big),$$

with i.i.d. observation noise $\boldsymbol{\epsilon}$ and a design matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ with full column rank, containing n samples of p features each. **Least-squares estimation (LSE)** is often introduced as purely linear algebra: minimize a squared-error objective and you get the "best" line in the least-squares sense. The goal of this exercise is to highlight that a **probabilistic view** here reveals **why** minimizing the sum of squares is sensible[1]: it is the log–likelihood of a Gaussian noise model (and inference under a Gaussian model comes down to linear algebra). The probabilistic view further let's us quantify uncertainty, change noise assumptions if needed and enables using principled tools that make use of the model likelihood to compare competing models ("**model selection**").

Tasks

1. First, to formulate a probabilistic model of the observations $\boldsymbol{y}$, show that the following two statements are equivalent:

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\Sigma}) \iff \boldsymbol{y} \,\big|\, \boldsymbol{x} \ \sim\ \mathcal{N}\big(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\Sigma}\big).$$

Hint: While this should be intuitively clear, there are (at least) two possible ways of analytically showing this:

- Using the **transformation laws of multivariate Gaussian distributions**.
- Using the general **change of (random-)variables formula**: For an invertible map $\boldsymbol{g} : \mathbb{R}^d \to \mathbb{R}^d$ and vector-valued random variables $\boldsymbol{Y}, \boldsymbol{Z}$:

$$\boldsymbol{p}_{\mathbf{Y}}(\mathrm{y}) \;=\; \boldsymbol{p}_{\mathbf{Z}}\big(g^{-1}(\mathrm{y})\big)\, \big|\det \boldsymbol{J}_{g^{-1}}(\mathrm{y})\big|,$$

where $\boldsymbol{J}_{g^{-1}}$ is the Jacobian matrix of the inverse of $\boldsymbol{g}$. The determinant factor ensures that the transformed density $\boldsymbol{p}_{\boldsymbol{Y}}$ remains properly normalized by compensating for local stretching or compression of infinitesimal volume elements induced by the change of coordinates under $\boldsymbol{g}$.

(1 pts.)

2. State the log-likelihood $\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\beta}, \sigma^2)$ of the model, and show that maximizing it w.r.t. $\boldsymbol{\beta}$ is equivalent to minimizing the residual sum of squares $\mathrm{RSS}(\boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2$ (or the MSE, since $\mathrm{MSE} = \mathrm{RSS}\,/n$).

(2 pts.)

3. Derive the **normal equations** and state the corresponding estimator $\widehat{\boldsymbol{\beta}}_{MLE}$. Also briefly state the conditions for the uniqueness of this estimator. Then, show that

$$\mathrm{Cov}(\widehat{\boldsymbol{\beta}}_{\mathrm{MLE}}) = \sigma^2(\boldsymbol{X}^\top \boldsymbol{X})^{-1}.$$

Hint: Plug in the model equation for $\boldsymbol{y}$ into the expression for $\widehat{\boldsymbol{\beta}}_{MLE}$ and simplify. Then, compute the covariance. Remember the scaling/shifting rules for this operator.

(3 pts.)

[1] In fact, Gauss developed the method of least squares precisely in the context of a normal distribution of errors.

4. Using the provided notebook, simulate a simple 1-D linear-Gaussian dataset and visualize the correspondence of OLS to MLE under a Gaussian noise model:

   (a) Generate $n = 60$ points (x_i, y_i) with $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i,\ \varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ (choose any β_0, β_1, σ you like).

   (b) Fit an OLS using sklearn's `LinearRegression` and show on one plot:

   - the data,
   - the fitted line, and
   - vertical "residual bars" from each point to the line.

   (c) Also do a histogram of the residuals, and

   (d) Plot the log likelihood and the RSS as a function of β_1 into one figure to visually show that the maximum of the likelihood coincides with the minimum of the RSS at the MLE.

   (4 pts.)

Exercise 2: Bayesian Linear Regression and Basis Function Expansion

Now we're going to extend our probabilistic view of OLS by putting on our Bayesian hat. This leads to **Bayesian Linear Regression.** In the classical model, the parameters $\boldsymbol{\beta}$ are treated as fixed but unknown. This view led us to an MLE point estimator, while in a Bayesian framework we instead place a prior distribution over the parameters and infer their **posterior distribution** after observing data. Lastly, we'll explore how using a **basis function expansion** for this model leads us straight to Gaussian process regression via the kernel trick.

Tasks

1. As in the "Statistical Darts" exercise of Sheet 2, we place a multivariate Gaussian prior on the unknown parameter vector (there: the aim point $\boldsymbol{\mu}$, here: the regression coefficients $\boldsymbol{\beta}$). Recall that since the Gaussian prior is conjugate to the Gaussian likelihood, the posterior is also a Gaussian. Derive the posterior distribution

$$p(\boldsymbol{\beta} \mid \boldsymbol{y}, \boldsymbol{X}) = \mathcal{N}(\boldsymbol{\mu}_{\text{post}}, \boldsymbol{\Sigma}_{\text{post}})$$

   for the Bayesian linear regression model with prior

$$\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{0}, \tau^2 \boldsymbol{I}_p), \quad \text{and likelihood} \quad \boldsymbol{y} \mid \boldsymbol{\beta}, \boldsymbol{X} \sim \mathcal{N}(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}_n).$$

   Hint(!): The standard derivation is tedious. One has to sum the exponents of the log likelihood and log prior, and complete the square with respect to $\boldsymbol{\beta}$ to identify the mean and covariance of the resulting Gaussian posterior. Here, we'll take a much simpler path by reframing the problem so that you can directly map the formulas from the "Statistical Darts" exercise to the problem. The key is to realize that the likelihood, viewed as a function of $\boldsymbol{\beta}$, is also proportional to a Gaussian. One can show that $p(\boldsymbol{y}|\boldsymbol{\beta}, \boldsymbol{X})$ is proportional, as a function of $\boldsymbol{\beta}$, to the following Gaussian distribution:

$$p(\boldsymbol{y}|\boldsymbol{\beta}, \boldsymbol{X}) \propto \mathcal{N}\left(\boldsymbol{\beta}; \widehat{\boldsymbol{\beta}}_{\text{MLE}},\ \sigma^2(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\right).$$

Now, the posterior is the product of two Gaussians in $\boldsymbol{\beta}$:

$$p(\boldsymbol{\beta} \mid \boldsymbol{y}) \propto \mathcal{N}(\boldsymbol{\beta}; \boldsymbol{0}, \tau^2 \boldsymbol{I}_p) \times \mathcal{N}(\boldsymbol{\beta}; \widehat{\boldsymbol{\beta}}_{\text{MLE}}, \sigma^2(\boldsymbol{X}^\top \boldsymbol{X})^{-1})$$

Use this to find the posterior's parameters by mapping the formulas from the Statistical Darts exercise onto the setting here.

(2 pts.)

2. In a fully Bayesian model, we want to make predictions using the **posterior predictive distribution**, which captures a full distribution over possible outcomes rather than a single point estimate. For a new observation $\boldsymbol{x}^\star$, the predictive distribution is given by

$$p(y^\star \mid \boldsymbol{x}^\star, \boldsymbol{X}, \boldsymbol{y}) = \int p(y^\star \mid \boldsymbol{x}^\star, \boldsymbol{\beta})\, p(\boldsymbol{\beta} \mid \boldsymbol{X}, \boldsymbol{y})\, d\boldsymbol{\beta}.$$

This expression integrates over the posterior of parameters $\boldsymbol{\beta}$, accounting for both **aleatoric uncertainty** (inherent noise in the observations) and **epistemic uncertainty** (uncertainty about the model parameters). Unlike point estimate predictions based on MLE or MAP estimates of $\boldsymbol{\beta}$, this leads to more realistic uncertainty estimates, especially in situations where data is scarce. Derive the predictive distribution for a new observation. **Hint:** Explain why this is also a Gaussian distribution. Then, compute the predictive mean by $m(\boldsymbol{x}^\star) = \mathbb{E}_{p(\boldsymbol{\beta}|\boldsymbol{y})}[(\boldsymbol{x}^\star)^\top \boldsymbol{\beta}]$, and the predictive variance $v(\boldsymbol{x}^\star) = \text{Var}[\boldsymbol{y}^\star|\boldsymbol{x}^\star]$ by using the law of total variance.

(3 pts.)

3. Imagine that we now want to be able to also capture nonlinear relationships between our inputs and our observations. One way of doing so is by performing a **basis function expansion**: Consider a nonlinear transformation of the inputs by a **feature map** $\boldsymbol{\phi}(\boldsymbol{x})$:

$$\boldsymbol{\phi} : \mathbb{R}^n \to \mathbb{R}^D, \quad \boldsymbol{x} \mapsto \left[\boldsymbol{\phi}_1(\boldsymbol{x}), \boldsymbol{\phi}_2(\boldsymbol{x}), \ldots, \boldsymbol{\phi}_D(\boldsymbol{x})\right]^\top.$$

The model then reads

$$\boldsymbol{y} = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{w} + \boldsymbol{\epsilon}, \quad \boldsymbol{w} \sim \mathcal{N}(0, \tau^2 \boldsymbol{I}_D), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\Sigma}).$$

(a) Explain why it is important that the expanded model is still **linear** as a function of the parameters.

(b) Let's formulate the Bayesian linear model in terms of the expanded feature space. State the likelihood $p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w})$, using the expanded feature matrix $\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{x}_1)^\top \\ \vdots \\ \boldsymbol{\phi}(\boldsymbol{x}_n)^\top \end{bmatrix} \in \mathbb{R}^{n \times D}$ to express it. Then, using your results from Tasks 2.1 and 2.2, also state the posterior distribution in terms of the $\boldsymbol{\Phi}$ and the predictive mean $m(\boldsymbol{x}^\star)$ in terms of $\boldsymbol{\phi}(\boldsymbol{x}^\star)$.

(c) By writing the predictive mean more explicitly, show that the predictions only depend on inner products $\boldsymbol{\phi}(x_i)^\top \boldsymbol{\phi}(x_j)$.

(d) Recall the kernel trick from the SVM. To apply it to our model, use the relation

$$\boldsymbol{\Sigma}_{\text{post}} \frac{1}{\sigma^2} \boldsymbol{\Phi}^\top = \boldsymbol{\Phi}^\top \big(\boldsymbol{K} + \tfrac{\sigma^2}{\tau^2} \boldsymbol{I}_n\big)^{-1},$$

where the **kernel matrix** is defined by $\boldsymbol{K} = \Phi\Phi^\top$ with entries $\boldsymbol{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}(\boldsymbol{x}_j)$. Show that

$$m(\boldsymbol{x}^\star) = \boldsymbol{k}_\star^\top \big(\boldsymbol{K} + \tfrac{\sigma^2}{\tau^2} \boldsymbol{I}_n\big)^{-1} \boldsymbol{y}, \qquad \text{where } \; k_\star(i) = k(\boldsymbol{x}_i, \boldsymbol{x}^\star).$$

(The steps from (c) to (e) can analogously be performed for the predictive variance which then also depends only on the kernel and no longer on $\boldsymbol{\phi}$ explicitly).

(e) Explain the conceptual step from the "kernelized" model to a Gaussian process. **Hint:** Interpret $k(\boldsymbol{x}, \boldsymbol{x}')$ as a covariance function, and consider what happens for large D.

(5 pts.)

4. Starting from the predictive variance (c.f. Task 2.2):

$$v(\boldsymbol{x}^\star) \;=\; \sigma^2 + \boldsymbol{\phi}_\star^\top \Sigma_{\text{post}} \, \boldsymbol{\phi}_\star,$$

briefly explain which summand corresponds to **aleatoric** and which to **epistemic** uncertainty, and why the latter shrinks as $n \to \infty$ while the former does not.

(1 pts.)

5. The previous tasks have shown that Bayesian linear regression can be "kernelized", allowing us to work in potentially very high-dimensional feature spaces implicitly. This final coding task will build on this idea to highlight a more profound insight: **a prior on the weights $\boldsymbol{w}$ induces a prior distribution over functions f.** By choosing a kernel, we are effectively choosing the properties of the functions we

expect to see, the kernel therefore takes the role of (the covariance function of) a prior. Conditioning on data then updates this distribution over functions to a **posterior distribution over functions**. This function-space perspective is the core idea behind Gaussian Processes. Here, you'll explore this concept by comparing two different kernels and visualizing the final uncertainty, as well as the distributions over functions themselves.

(a) We'll compare two models based on two different kernels. The first is a degree-9 Polynomial kernel: $k_9(\boldsymbol{x}, \boldsymbol{x}') = (1 + \boldsymbol{x}^\top \boldsymbol{x}')^9$. The second is the widely used **Radial basis function (RBF) kernel**, also known as the Gaussian or squared-exponential kernel:

$$k_{\text{RBF}}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{x}'||_2^2}{2\ell^2}\right),$$

where ℓ is a hyperparameter called the **lengthscale**, which controls the "smoothness" or "wiggliness" of the functions. The RBF kernel corresponds to a basis function expansion into an infinite-dimensional feature space, making the kernel trick essential.

i. Implement the polynomial kernel and the RBF kernel function.

ii. Before looking at any data, we can visualize our prior beliefs by drawing samples from the prior distribution over functions. A function sample $f(x)$ can be generated by drawing a weight vector from its prior, $\boldsymbol{w}_{\text{sample}} \sim \mathcal{N}(\boldsymbol{0}, \tau^2 \boldsymbol{I})$, and then plotting $f(x) = \boldsymbol{\phi}(x)^\top \boldsymbol{w}_{\text{sample}}$. For a grid of input points $\boldsymbol{X}_{\text{grid}}$, create a figure with 1×2 subplots, showing 5 function samples from the polynomial kernel prior and the RBF kernel prior, respectively use $\ell = 0.1, \tau^2 = 1.0$. Briefly comment on the qualitative differences between the functions generated by each prior.
Hint: For the RBF kernel, you cannot construct the feature map $\boldsymbol{\phi}(x)$ explicitly. However, you can draw samples from the equivalent Gaussian Process prior, which is a multivariate normal distribution with mean zero and covariance matrix given by the kernel matrix $\boldsymbol{K} = k(\boldsymbol{X}_{\text{grid}}, \boldsymbol{X}_{\text{grid}})$. Sample from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{K})$ using `np.random.multivariate_normal`.

(3 pts.)

(b) Now, let's see how these prior beliefs are updated by data. Since we're in a Bayesian framework, this means we condition on the data to get the posterior.

i. Use the provided functions to generate a small training set of $n = 20$ points. To better observe the model's behavior, we'll generate data from the slightly more complex function, $y_i = \sin(2\pi x_i) + 0.5 \sin(4\pi x_i) + \varepsilon_i$, and importantly, we'll create a **gap** in the data by only sampling x_i from the ranges $[0, 0.4]$ and $[0.6, 1.0]$. Fit two Bayesian regression models to this data, one with the polynomial kernel and one with the RBF kernel ($\ell = 0.1$). For both models, compute the posterior predictive distribution (mean and variance) over a dense grid from $x = 0$ to $x = 1$.

ii. Create another 1×2 plot to show the results. For each model:
   - Plot the training data, the true function, and the predictive mean.
   - Plot the 95% credible interval for the total uncertainty (aleatoric + epistemic) as a shaded band.
   - Plot the 95% credible interval for the epistemic uncertainty **only** as a differently colored shaded band.

(3 pts.)

(c) The uncertainty bands in the previous plot summarize the posterior distribution at each point. We can also visualize it by drawing function samples from the posterior. On the same two plots from part (b), draw and overlay 5 faint function samples from each model's posterior distribution.
Hint: To draw posterior samples, sample from the multivariate Gaussian whose mean you computed in (b) and whose covariance matrix is

$$\mathrm{Cov}(\boldsymbol{f}_\star) = \boldsymbol{K}_{\star\star} - \boldsymbol{K}_\star(\boldsymbol{K} + \sigma^2 \boldsymbol{I}_n)^{-1}\boldsymbol{K}_\star^\top,$$

where $\boldsymbol{K} = k(\boldsymbol{X}_{\text{train}}, \boldsymbol{X}_{\text{train}})$, $\boldsymbol{K}_\star = k(\boldsymbol{X}_{\text{grid}}, \boldsymbol{X}_{\text{train}})$ and $\boldsymbol{K}_{\star\star} = k(\boldsymbol{X}_{\text{grid}}, \boldsymbol{X}_{\text{grid}})$.

(1 pts.)

(d) Based on your plots from parts (b) and (c), briefly answer the following:

i. Which kernel provides a more reasonable fit to the data and why?

ii. Compare the epistemic uncertainty for both models. Where is it largest? How does it behave inside the data gap you created?

iii. How do the posterior function samples (part c) relate to the uncertainty bands (part b)? Explain what the spread of these samples represents.

(2 pts.)