Machine Learning
Winter Semester 2019/2020

SHEET #5 – SOLUTION

Due:

Fred Hamprecht
Manuel Haußmann

1 Visualize Regularization Contours (10 pt)

For two dimensional parameter vectors β we can visualize the error/loss surface of linear regression using contour plots. In this exercise you will create a set of such plots in order to familiarize yourself further with the influence of regularization. You can visualize the contours for example via `plt.contour` or `plt.contourf`.[1]

i) Plot the Ridge regression regularization term as well as the Lasso[2] regularization term for $\beta_1, \beta_2 \in [-1, 3]$.

ii) For the data set `sheet5-linreg1.npz` plot the sum of squares (SSQ) of a linear regression as a function of $\boldsymbol{\beta}$ over the same range as in **i)**, i.e. over the grid $[-1, 3] \times [-1, 3]$.

iii) Plot the ridge loss function, i.e. $\text{SSQ}(\boldsymbol{\beta}) + \lambda||\boldsymbol{\beta}||_2^2$ for $\lambda \in \{0, 10, 50, 100, 200, 300\}$ in the same $\boldsymbol{\beta}$ grid as before and *discuss your observations!*

iv) Plot the Lasso loss function, i.e. $\text{SSQ}(\boldsymbol{\beta}) + \lambda||\boldsymbol{\beta}||_1$ for $\lambda \in \{0, 10, 50, 100, 200, 300\}$ in the same $\boldsymbol{\beta}$ grid as before and *discuss your observations!*

v) Repeat steps **ii) - iv)** for the data set `sheet5-linreg2.npz`. Which qualitative differences do you observe?

Solution: See jupyter-notebook.

2 Elastic Net: Combine both L1 and L2 (10 pt)

Lasso (L1) and Ridge (L2) regression both have different strengths and weaknesses. One extension is to combine them into a combined model, known as the *Elastic net.*[3] The objective can then be written as

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} ||\mathbf{y} - \boldsymbol{\beta}^T\mathbf{X}||^2 + \lambda_2||\boldsymbol{\beta}||_2^2 + \lambda_1||\boldsymbol{\beta}||_1, \tag{1}$$

where λ_1, λ_2 are the regularization strength hyper-parameters. Show that this can be equivalently reformulated as a Lasso problem by creating a modified data set consisting of $\tilde{\mathbf{X}} = c \cdot (\mathbf{X}, \sqrt{\lambda_2}\mathbb{1}_p) \in \mathbb{R}^{p \times N \cdot p}$, $\tilde{\mathbf{y}} = (\mathbf{y}, \mathbf{0}_{1\times p}) \in \mathbb{R}^{1\times N\cdot p}$ and optimizing[4]

$$\arg\min_{\tilde{\boldsymbol{\beta}}} ||\tilde{\mathbf{y}} - \tilde{\boldsymbol{\beta}}^T\tilde{\mathbf{X}}||^2 + c\lambda_1||\tilde{\boldsymbol{\beta}}||_1 \tag{2}$$

with $c = 1/\sqrt{1+\lambda_2}$ and $\boldsymbol{\beta} = c\tilde{\boldsymbol{\beta}}$.

[1] See https://matplotlib.org/3.1.1/gallery/images_contours_and_fields/contour_demo.html for an example.
[2] The abbreviation comes from *least absolute shrinkage and selection operator*.
[3] See *Regularization and variable selection via the elastic net* (Zou & Hastie, 2005) for details on the model and its motivation.
[4] $\mathbb{1}_p$ is the p-dimensional identity matrix and $\mathbf{0}_{1\times p}$ is a p-dimensional vector of 0's.

Solution: Using the notation $\mathbf{x}_j = \mathbf{X}_{:,j}$ we can write the Lasso objective as

$$\begin{aligned}
||\tilde{\mathbf{y}} - \tilde{\boldsymbol{\beta}}^T\tilde{\mathbf{X}}||^2 + c\lambda_1||\tilde{\boldsymbol{\beta}}||_1 &= \sum_{j=1}^{Np}(\tilde{y}_j - \boldsymbol{\beta}^T\tilde{\mathbf{x}}_j)^2 + \lambda_1||\underbrace{c\tilde{\boldsymbol{\beta}}}_{=\boldsymbol{\beta}}||_1 \\
&= \sum_{j=1}^{N}(\tilde{y}_j - \tilde{\boldsymbol{\beta}}^T\tilde{\mathbf{x}}_j)^2 + \sum_{j=N+1}^{Np}(\tilde{y}_j - \tilde{\boldsymbol{\beta}}^T\tilde{\mathbf{x}}_j)^2 + \lambda_1||\boldsymbol{\beta}||_1 \\
&= \sum_{j=1}^{N}(y_j - c\tilde{\boldsymbol{\beta}}^T\mathbf{x}_j)^2 + \underbrace{\sum_{j=1}^{p}(0 - c\sqrt{\lambda_2}\tilde{\boldsymbol{\beta}}^T\mathbf{1}_j)^2}_{=\lambda_2||\boldsymbol{\beta}||_2^2} + \lambda_1||\boldsymbol{\beta}||_1
\end{aligned}$$

3 Fitting a 1D Gaussian Process (10 pt)

Throughout this exercise, you will be fitting a Gaussian Process (GP)[5] to the data set `gp-data.npz`. Throughout we assume the GP to have a zero mean function and covariance function is given by an exponentiated quadratic[6] covariance function

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\ell}\right) \tag{3}$$

with lengthscale ℓ.

i) Plot 10 samples from the prior on the range $[0, 12]$.
Hint: How to sample from a multivariate Normal distribution? As in the 1d case we can transform samples from a standard Normal distribution $\mathcal{N}(0, \mathbb{1})$ into a distribution with an arbitrary mean $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, by using that if $\mathbf{z} \sim \mathcal{N}(0, \mathbb{1})$, then $\mathbf{x} = \boldsymbol{\mu} + \mathbf{L}\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\mathbf{L}$ such that $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^T$, e.g. via the Cholesky decomposition (use `np.linalg.cholesky`).

ii) Pick $N \in \{1, 2, 5, 10, 20\}$ points from the data set and visualize how the posterior mean and variance changes as you add more and more observations. You can visualize this posterior uncertainty either by drawing and plotting multiple samples from the posterior, or by using `plt.fill_between` to plot the posterior variance. Assume an observation noise with a variance of $\sigma^2 = 0.0001$.

iii) Compute the posterior given all $N = 20$ data points. Visualize and discuss how it changes as you change the length-scale for $\ell \in \{0.01, 0.1, 0.5, 1, 5, 100\}$.

Solution: See jupyter-notebook.

4 Constructing new Kernels (technical +6pt)

One major source of the power of Gaussian Processes is the strength and flexibility of kernels. A nice feature of kernels is that given a set of existing kernels, you can easily create new ones. Show that for two valid kernels $k_1(\mathbf{x}, \mathbf{x}')$ and $k_2(\mathbf{x}_,\mathbf{x}')$

$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}') \qquad \text{and} \qquad k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') \cdot k_2(\mathbf{x}, \mathbf{x}') \tag{4}$$

are again valid kernels.[7]

[5] If you want more material on GPs, a great resource is http://gpss.cc/gpss19/ with many lectures and exercises.

[6] Also known as radial basis function (RBF), Gaussian kernel, squared exponential,... it has many names.

[7] These results can be extended for a large set of further combinations and modifications.

Hint: Make use of the following two results for kernels. i) for a kernel function to be valid it is necessary and sufficient for the so called Gram matrix $\mathbf{K}$[8] *to be positive semi-definite*[9]*; ii) If* $k(\mathbf{x}, \mathbf{x}')$ *is a valid kernel, there exists a mapping* $\phi(\cdot)$*, such that* $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x})$*.*

Solution:

- *Adding two kernels gives a new kernel*: Assuming we have two valid covariance functions $k_1(\cdot,\cdot), k_2(\cdot,\cdot)$ and define:

  $$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}'). \tag{5}$$

  Let $\mathbf{K}$ be given with $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and analogously $,\mathbf{K}_1, \mathbf{K}_2$. then $\mathbf{K} = \mathbf{K}_1 + \mathbf{K}_2$ and we have for an arbitrary vector $\mathbf{a}$ that

  $$\mathbf{a}^T \mathbf{K} \mathbf{a} = \mathbf{a}^T (\mathbf{K}_1 + \mathbf{K}_2) \mathbf{a} = \mathbf{a}^T \mathbf{K}_1 \mathbf{a} + \mathbf{a}^T \mathbf{K}_2 \mathbf{a} \geq 0, \tag{6}$$

  and thus $k(\cdot,\cdot)$ is again a valid covariance kernel function.

- *Multiplying two kernels gives a new kernel*: Given two valid kernels k_1, k_2 there exist mappings $\phi(\cdot), \psi(\cdot)$ such that

  $$k_1(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}') \qquad \text{and} \qquad k_2(\mathbf{x}, \mathbf{x}') = \psi(\mathbf{x})^T \psi(\mathbf{x}'). \tag{7}$$

  We then have that

  $$\begin{aligned} k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') k_2(\mathbf{x}, \mathbf{x}') &= \phi(\mathbf{x})^T \phi(\mathbf{x}') \psi(\mathbf{x})^T \psi(\mathbf{x}') \\ &= \sum_n^N \phi_n(\mathbf{x}) \phi_n(\mathbf{x}') \sum_m^M \psi_m(\mathbf{x}) \psi_m(\mathbf{x}') \\ &= \sum_n^N \sum_m^M \phi_n(\mathbf{x}) \phi_n(\mathbf{x}') \psi_m(\mathbf{x}) \psi_m(\mathbf{x}') \\ &= \sum_k^K \zeta_k(\mathbf{x}) \zeta_k(\mathbf{x}') = \zeta(\mathbf{x})^T \zeta(\mathbf{x}') \end{aligned}$$

  for suitably defined $\zeta_k(\cdot)$.

[8] $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$

[9] $\mathbf{a}^T \mathbf{K} \mathbf{a} \geq 0$ for arbitrary $\mathbf{a}$.