Machine Learning
Winter Semester 2019/2020

Exercise Sheet #4

Due: 2019-11-14, 16:00

Fred Hamprecht
Manuel Haußmann

General Regulations.

- You should hand in your solutions in groups of at least two people (recommended are three).
- The theoretical exercises can be either handwritten notes (scanned), or typeset using LaTeX.
- Practical exercises should be implemented in python and submitted as jupyter notebooks (`.ipynb`). Always provide the (commented) code as well as the output, and don't forget to explain/interpret the latter.
- Submit all your files in a single `.zip` archive to `mlhd1920@gmail.com` using the following standardized format: The subject line should consist of the full names of all team members as well as the exercise, and the title of the zip archive the last names. I.e. assuming your group consists of Ada Lovelace, Geoffrey Hinton and Michael Jordan, this means

  Subject: [EX04] Michael Jordan, Geoffrey Hinton, Ada Lovelace
  Zip Archive: `ex04-jordan-hinton-lovelace.zip`

Linear Regression: Preliminaries

In the lecture we considered the setup

$$y = \boldsymbol{\beta}^T \mathbf{x} + \varepsilon, \tag{1}$$

with $\mathbb{E}[\varepsilon] = 0$ and $\mathrm{var}[\varepsilon] = \sigma^2$ for some data independent variance. This does not allow for a fixed offset (i.e. the model assumes $y = 0$ for $\mathbf{x} = 0$). To be more flexible, while still keeping the structure we discussed one can proceed as follows. Consider

$$y = \beta_0 + \boldsymbol{\beta}^T \mathbf{x} + \varepsilon = \tilde{\boldsymbol{\beta}}^T \tilde{\mathbf{x}} + \varepsilon, \tag{2}$$

where we have defined $\tilde{\boldsymbol{\beta}} = (\beta_0, \boldsymbol{\beta}^T)^T$ and $\tilde{\mathbf{x}} = (1, \mathbf{x}^T)^T$. Throughout these exercises (and also most often in practice) we will assume to be in this more general case, dropping the tilde from the notation. This means that you always need to remember to include one row of ones at the beginning of your data matrix $\mathbf{X}$.

1 Plotting of the Residuals (10 pt)

One diagnostic tool for verification that the assumption of a linear model was reasonably correct is the plotting of the residuals. Given our assumptions we consider them to be spread around zero with a constant variation and no discernible pattern.

i) Without looking at the actual data, fit a linear regression model to `linear-res.npz`[1] and visualize the residuals. Discuss any patterns you observe.

ii) Extend the data matrix by including also squared features, fit a second model and visualize the residuals again.

iii) Finally, plot both the data as well as the fitted models.

[1] An `.npz` file contains multiple arrays that you can access as you would in a dictionary. You can load it as usual, e.g. `tmp = np.load('foo.npz')` and then access the keys as `list(tmp.keys())`. On this sheet, we will always have the features as `tmp['X']` and the targets as `tmp['Y']`.

2 Uncertainty in the Parameter Estimation (10 pt + 2pt)

In the lecture we discussed that given that $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and assuming the model to be correct, one can show that $\hat{\boldsymbol{\beta}} \sim \mathcal{N}\left(\boldsymbol{\beta}^*, (\mathbf{X}\mathbf{X}^T)^{-1}\sigma^2\right)$. In this exercise you will explore this uncertainty for three data sets, `linear-unc1.npz`, `linear-unc2.npz`, and `linear-unc3.npz`. Perform the following steps for each of them. For simplicity we assume the noise variance to be known with $\sigma^2 = 1$ and that we know the intercept/bias[2] term to be zero (i.e. $\beta_0 = 0$).

i) Each data set consists of a large number of features $\mathbf{x}$ and target variables y. In order to simulate the uncertainty in $\hat{\beta}$, sample 100 sets of 10 points each from them.[3]

ii) Fit a linear regression for each subset and plot the values for β_1, β_2. Discuss the patterns you observe.

iii) Fit a linear regression to each complete data set and compare your $\hat{\boldsymbol{\beta}}$ estimate it with the results you obtained in **ii)**.

iv) *(technical +2pt)* Plot the density of the distribution you obtained for $\hat{\boldsymbol{\beta}}$ in the last step.

3 Estimating Parameter Relevance (10 pt)

Given the importance of linear regression, there exist many approaches to estimate the importance of the individual features after learning the parameters (e.g. hypothesis tests, close inspections of the posteriors in a Bayesian setup, ...). Here you will be implementing an approach that follows from a direct intuitive motivation. In the permutation (排列) test to test whether the i-th feature is relevant, you randomly permute it among the data points. Starting from $\mathbf{X} \in \mathbb{R}^{p \times N}$, you then get $\tilde{\mathbf{X}} \in \mathbb{R}^{p \times N}$ with $\tilde{\mathbf{X}}_{i,:} = \pi(\mathbf{X}_{i,:})$, i.e. the i-th row of the data matrix is permuted (where $\pi(\cdot)$ indicates the permutation operation).

You will apply this to `linear-rel.npz`, a variation of the classical boston housing data set. In it, one attempts to estimate the median value of a property given features like the number of rooms, the local crime rate, age of the building, ... I added one additional random feature that is completely useless prediction-wise and your goal is to find it.

For each of the p features, create a data matrix $\tilde{\mathbf{X}}$ with one of the rows permutated. Fit a new linear regression and compare it to the sum-of-squared residuals of the original $\mathbf{X}$. If they remain roughly equal then this suggests that the feature is irrelevant. Which feature is the random additional feature?

4 σ^2 Estimation and Heteroscedastic Noise (technical +12pt)

i) Maximum Likelihood. Focusing on a single point $(y_n, \mathbf{x}_n)$ our linear regression model simplifies to

$$y_n = \boldsymbol{\beta}^T \mathbf{x}_n + \varepsilon_n. \tag{3}$$

If we assume that $\varepsilon_n \sim \mathcal{N}(0, \sigma^2)$, this is equivalent to the assumption that $y_n \sim \mathcal{N}(\boldsymbol{\beta}^T \mathbf{x}_n, \sigma^2)$. The logarithm $p(y_n | \boldsymbol{\beta}, \sigma^2)$ is known as the log-likelihood. Having observed N data points this formulation generalizes to a sum of log-likelihoods and we can learn $\boldsymbol{\beta}$ by maximizing the logarithm of

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \sum_{n=1}^{N} \log \mathcal{N}\left(y_n | \boldsymbol{\beta}^T \mathbf{x}, \sigma^2\right). \tag{4}$$

Show that we are solving the same objective as in the SSQ formulation (just with a different scaling factor) and get the same solution for $\boldsymbol{\beta}$. This formulation is known as the *Maximum Likelihood* approach, as we learn the parameters that maximize the likelihood of the data.

[2] Not the statistical bias, just a naming collision.

[3] See the last exercise from the first sheet if you need inspiration on how to implement this.

ii) Estimation of σ^2. Estimating σ^2 then analogously consists of finding the $\hat{\sigma}^2$ that maximizes this log-likelihood given the estimates $\hat{\boldsymbol{\beta}}$, i.e.

$$\hat{\sigma}^2 = \arg\max_{\sigma^2} \sum_{n=1}^{N} \log \mathcal{N}\left(y_n | \hat{\boldsymbol{\beta}}^T \mathbf{x}, \sigma^2\right). \tag{5}$$

Solve this and relate the result to the SSQ residual formulation from the lecture.

Hint: It might be easier to work with the precision $\alpha = \frac{1}{\sigma^2}$ for most of the derivation.

iii) Heteroscedastic Noise. The standard formulation of linear regression is of homoscedastic noise, i.e. the variances of the observation noise is independent of $\mathbf{x}$. A generalization is to have a data point dependent variance on the observation noise, i.e. we have

$$y_n = \boldsymbol{\beta}^T \mathbf{x}_n + \varepsilon_n, \tag{6}$$

with $\mathbb{E}[\varepsilon_n] = 0$ and $\operatorname{var}[\varepsilon_n] = \sigma_n$, which is known as *heteroscedastic noise*. Give the sum-of-squares problem in that case and derive mean and covariance structure of the $\hat{\boldsymbol{\beta}}$ in that case.