Seminars on Overparametrized Machine Learning: Hand-in assignment 2

Antônio H. Ribeiro, Dave Zachariah, Per Mattsson

Due: 14th of October 2021, 23:59

The items in the assignment can be implemented in the programming language of choice. Nonetheless, we recommend the usage of Python as a programming language, since we might include suggestions of functions in the exercise description.

Double-descent in Linear Regression with Random Covariates

You are going to reproduce the experiment in Figure 2 from (Hastie et al., 2019). The instructions give some extra details on how to do it. You should obtain both empirical and asymptotic results and plot them together. You will use covariates that are random and i.i.d and estimate the parameters using the minimum-norm solution.

Data. You will generate simulated data similar to the one described in (Hastie et al., 2019) in Section 3. The data will come from a linear model with isotropic features.

- **Isotropic inputs.** You will generate random inputs to be used to train and test your model. The i-th input will be a vector with size p:
$$\mathbf{x}_i = (x_{i1}, \cdots, x_{ip})$$
where the entries x_{ij} are i.i.d. with zero mean and variance 1 (for instance, values sampled from $\mathcal{N}(0, 1)$)).
- **Parameter vector with fixed ℓ_2 norm.** Choose a parameter vector $\boldsymbol{\beta} \in \mathbb{R}^p$ to be used to generate the data. You can use any arbitrary value that satisfy the restriction $\|\boldsymbol{\beta}\|_2 = r$ for the values of r specified latter. As an example you could sample the entries of β_j from the normal distribution $\mathcal{N}(0, \frac{r^2}{p})$. Or you could just make it constant $\boldsymbol{\beta} = \left(\frac{r}{\sqrt{p}}, \cdots, \frac{r}{\sqrt{p}}\right)$.
- **Linear model.** You should generate the dataset $(\mathbf{x}_i, y_i) \in \mathbb{R}^p \times \mathbb{R}$ for $i = 1, \ldots, n$ according to the linear equations
$$y_i = \mathbf{x}_i^\mathsf{T} \boldsymbol{\beta} + \epsilon_i, \tag{1}$$
for which the random draws across $i = 1, \ldots, n$ are independent. The error ϵ_i is generated independently from $\mathbf{x}_i$ and comes from a distribution with zero mean and variance σ^2.
- **Train and test datasets.** Using this procedure generate both train and test datasets. We denote the training dataset using matrix notation $(\mathbf{X}, \mathbf{y})$ where $\mathbf{X} \in \mathbb{R}^{n \times p}$ is the matrix containing the j-th entry of the i-th input x_{ij} at position (i, j) and $y \in \mathbb{R}^n$ is the vector of outputs. The training dataset should contain $n = 100$ samples. And, let the test set be denoted by $(\mathbf{X}_{\text{test}}, \mathbf{y}_{\text{test}})$ consist of $n_{\text{test}} = 100$ samples.

Model. Estimated parameters by minimizing the sum of square errors
$$\frac{1}{n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2, \tag{2}$$
In the overparametrized regime, there are multiple solutions, and you should use the minimum ℓ_2-norm solution to the problem (as used by Hastie et al. (2019); Belkin et al. (2019)), i.e.:
$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_2 \quad \text{subject to} \quad \mathbf{y} = \mathbf{X}\boldsymbol{\beta}, \tag{3}$$
This behaviour can be expressed by the analytical solution
$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\mathsf{T}\mathbf{X})^+ \mathbf{X}^\mathsf{T} \mathbf{y}, \tag{4}$$
where $(\mathbf{X}^\mathsf{T}\mathbf{X})^+$ denotes the Moore-Penrose pseudo-inverse of $\mathbf{X}^\mathsf{T}\mathbf{X}$ which does yield the desired behaviour in both the underparametrized and overparametrized regions.

Note: scipy.linalg.lstsq does yield the desired behaviour both in the underparametrized and overparametrized region.

Empirical evaluation. Evaluate your model on the test set by computing the mean square error

$$\text{MSE}_{\text{test}} = \frac{1}{n_{\text{test}}}\|\mathbf{y}_{\text{test}} - \mathbf{X}_{\text{test}}\hat{\boldsymbol{\beta}}\|^2. \tag{5}$$

Asymptotics Use the asymptotics obtained by Hastie et al. (2019) in Theorem 1 and Coroloary 1, as $n, p \to \gamma$ and $p/n \to \gamma$,

$$\text{MSE}_{\text{test}} \quad \to \quad \begin{cases} \sigma^2 \frac{\gamma}{1-\gamma} + \sigma^2 & \gamma < 1, \\ r^2(1 - \frac{1}{\gamma}) + \sigma^2 \frac{1}{\gamma - 1} + \sigma^2 & \gamma > 1, \end{cases} \tag{6}$$

$$\|\hat{\boldsymbol{\beta}}\|_2^2 \quad \to \quad \begin{cases} r^2 + \sigma^2 \frac{\gamma}{1-\gamma} & \gamma < 1, \\ r^2 \frac{1}{\gamma} + \sigma^2 \frac{1}{\gamma - 1} & \gamma > 1. \end{cases} \tag{7}$$

- **Note 1:** More precisely, Hastie et al. (2019) establishes that the expectation (conditioned on $\mathbf{X}$) of the values on the left hand side converge, almost surely, to the values on the right hand side under the appropriate conditions.

- **Note 2:** You might notice some subtle differences in the formulation above from the one in Hastie et al. (2019). They give the asymptotics for the risk $R = E[\mathbf{x}_0^\mathsf{T}\hat{\boldsymbol{\beta}} - \mathbf{x}_0^\mathsf{T}\boldsymbol{\beta}]$, where $\mathbf{x}_0$ is a test point not seem during training. Here, we work directly with the mean square error on the test error instead. Eq. 6 above follows from the fact that $\text{MSE}_{\text{test}} \to R + \sigma^2$ as $n_{\text{test}} \to \infty$.

Exercise

Choose σ^2 and r^2, and (for the values you choose) plot the **asymptotics** *and, also,* the **empirical values** of

1. The **test** mean square error;
2. The **parameter** ℓ_2 **norm** $\|\beta\|_2$.

For the **empirical evaluation**: fix the number of training points $n = 200$, and perform the empirical evaluation described above. Do it for the number of parameters p ranging from $0.1n$ to $10n$ (generate at least 100 configurations in this range using logspace). Plot as a function of the ratio $\gamma = p/n$.

For the **asymptotics** evaluation: Compute the asymptotic values for γ ranging from 0.1 to 10 (use a finer grid so it looks continuous). Plot the asymptotics superimposed to the empirical obtained values.

Some tips for the plots to look nice:

- Close to the interpolation point the test error takes very large values. We suggest to manually set the y-limits in the plot, so the region of interest is highlighted. The same also applies to the parameter norm.

- Using logscale in the x-axis might also make the plot more clear.

- Add a vertical line in the threshold $\gamma = 1$ to show where the interpolation threshold is.

- Use markers to plot the empirical results and lines to plot the asymptotics.

The Submission

Your submission should have a single page of content (a4paper, fontsize=10pt, margin=2cm, both single and double column are acceptable...). Include your name, the plots, a short description of the experiment parameters that you used and a paragraph of discussion/conclusion. You can assume that whoever will read your report has both read paper from (Hastie et al., 2019) and the entire description above, so there is no need for repeating it...

All requested plots should have proper figure captions, legends, and axis labels. You should submit two files, one pdf-file with the report and a standalone script (or jupyter notebook) that can be used to run the code and generate the plots (Write as comments the packages/libraries versions and additinal requirements as comments in the top of the script). Compress the two files as a single zip (containing pdf + script) and mail it to antonio.horta.ribeiro@it.uu.se. You will receive a confirmation mail back.

References

M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, Aug. 2019. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1903070116.

T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani. Surprises in High-Dimensional Ridgeless Least Squares Interpolation. *arXiv:1903.08560*, Nov. 2019. URL http://arxiv.org/abs/1903.08560.