

Exercise 6

Machine Learning I

6A-1.

First, rearrange through Bayes theorem:

$$p(\mathbf{t}|\mathbf{X}) = \frac{p(\mathbf{t}|\mathbf{w}, \mathbf{X})p(\mathbf{w}|\mathbf{X})}{p(\mathbf{w}|\mathbf{t}, \mathbf{X})}$$

Decompose:

$$\log p(\mathbf{t}|\mathbf{X}) = \log p(\mathbf{t}|\mathbf{w}, \mathbf{X}) + \log p(\mathbf{w}|\mathbf{X}) - \log p(\mathbf{w}|\mathbf{t}, \mathbf{X})$$

We already determined the densities for the likelihood, posterior and prior in the previous exercises. No need to recompute everything again. Because above has to be valid for any \mathbf{w} , we pick $\mathbf{w} = \boldsymbol{\mu}_N$ for convenience.

Let us first convert only the exponent

$$\begin{aligned} \log p(\mathbf{t}|\mathbf{X}) &\propto -\frac{1}{2}(\mathbf{t} - \Phi\mathbf{w})^T \tau_e \mathbf{I}(\mathbf{t} - \Phi\mathbf{w}) - \frac{1}{2}\mathbf{w}^T \tau_0 \mathbf{I}\mathbf{w} \\ &+ \frac{1}{2}(\mathbf{w} - \tau_e \boldsymbol{\Sigma}_N \boldsymbol{\Phi}^T \mathbf{t})^T (\tau_0 \mathbf{I} + \tau_e \boldsymbol{\Phi}^T \boldsymbol{\Phi})(\mathbf{w} - \tau_e \boldsymbol{\Sigma}_N \boldsymbol{\Phi}^T \mathbf{t}) \\ &= -\frac{1}{2}(\mathbf{t} - \Phi\boldsymbol{\mu}_N)^T \tau_e \mathbf{I}(\mathbf{t} - \Phi\boldsymbol{\mu}_N) - \frac{1}{2}\boldsymbol{\mu}_N^T \tau_0 \mathbf{I}\boldsymbol{\mu}_N \quad \text{set } \mathbf{w} = \boldsymbol{\mu}_N \end{aligned}$$

Now let us add the normalization constants:

$$\begin{aligned} \log p(\mathbf{t}|\mathbf{X}) &= -\frac{1}{2}(\mathbf{t} - \Phi\boldsymbol{\mu}_N)^T \tau_e \mathbf{I}(\mathbf{t} - \Phi\boldsymbol{\mu}_N) - \frac{1}{2}\boldsymbol{\mu}_N^T \tau_0 \mathbf{I}\boldsymbol{\mu}_N + \frac{1}{|r_0|} \\ \log p(\mathbf{t}|\mathbf{X}) &= -\frac{1}{2}(\mathbf{t} - \Phi\boldsymbol{\mu}_N)^T \tau_e \mathbf{I}(\mathbf{t} - \Phi\boldsymbol{\mu}_N) - \frac{1}{2}\boldsymbol{\mu}_N^T \tau_0 \mathbf{I}\boldsymbol{\mu}_N \\ &+ \frac{d}{2} \log r_0 + \frac{n}{2} \log r_e \frac{n}{2} \log 2\pi + \log |\boldsymbol{\Sigma}_n| \end{aligned}$$

TODO: Nochmal Terme überprüfen...

6A-2.

Let w_{MAX} be the (unregularized) maximum likelihood parameters. The lasso path illustrates the current selection of weights w_i given weight regularization $\|w_{current}\| = \frac{t}{\|w_{MAX}\|_1}$, $t \in [0, \|w_{MAX}\|_1]$. For a very thorough and detailed explanation, please refer to page 69 of "Elements of statistical learning".

You can find the online version here:

https://web.stanford.edu/~hastie/ElemStatLearn//printings/ESLII_print10.pdf

This means that for low values on the x axis, the model is heavily regularized. Moving along the x axis then relaxes the regularization constraint until you get to the unregularized maximum likelihood solution $\|w_{MAX}\|$.

In order to study what happens, we are using now 3 classes. Using a design matrix, we can model each degree x^i of the regression polynomial $y(x) = \sum_{i=0}^d w_i x^i$ as a separate dimension for the lasso path.

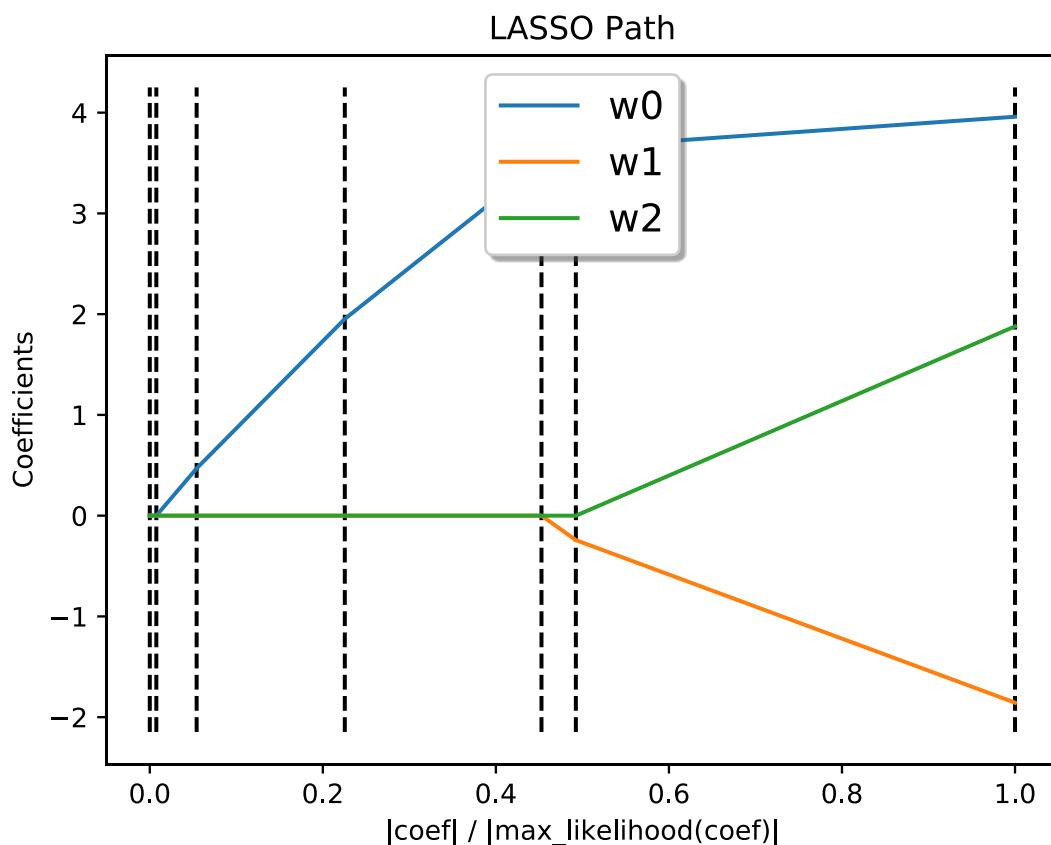
Let us have

$$f(x) = 4 - 2x + 2x^2 - x^3$$

Given enough samples, the maximum likelihood solution should set our regressor weights as such:

$$w_0 = 4, w_1 = -2, w_2 = 2, w_3 = -1$$

This can be seen, if we look at the right end of our lasso path:



The less we regularize our weights (we move towards $x=1$) the more we obtain the maximum likelihood solution for our weights. Model was created with $n=3000$ samples and sample interval $I = (0,2)$.

Now something should be noted: In the above image, weight w_0 was picked up first, then w_1 and lastly w_2 . This means, in this case, relaxing the lasso translates to gradually picking up model complexity.

Now we are ready for eight classes. Our polynomial is

$$g(x) = 2 - 3x + 4x^2 - 5x^3 + 4x^4 - 4x^5 + 6x^6 - 5x^7$$

The order of weight pick up is very much dependent on the sample interval I . Shown are two pictures of the same function $g(x)$, one time with a sample interval of $I = (0,25)$ and one time with $I = (-1,1)$:

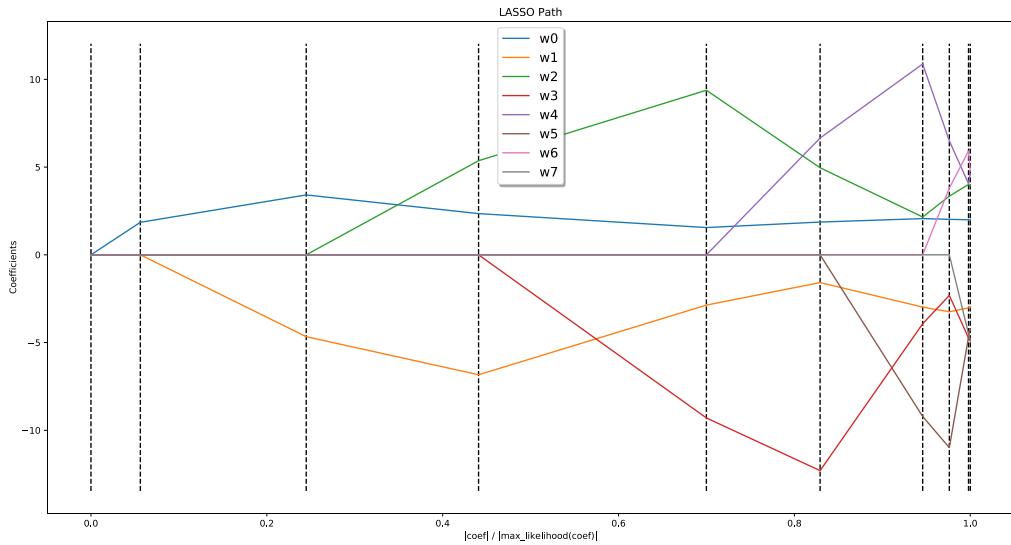


Figure 1 $n=70000, I=(-1,1)$. The weights are picked up in ascending order of degree.

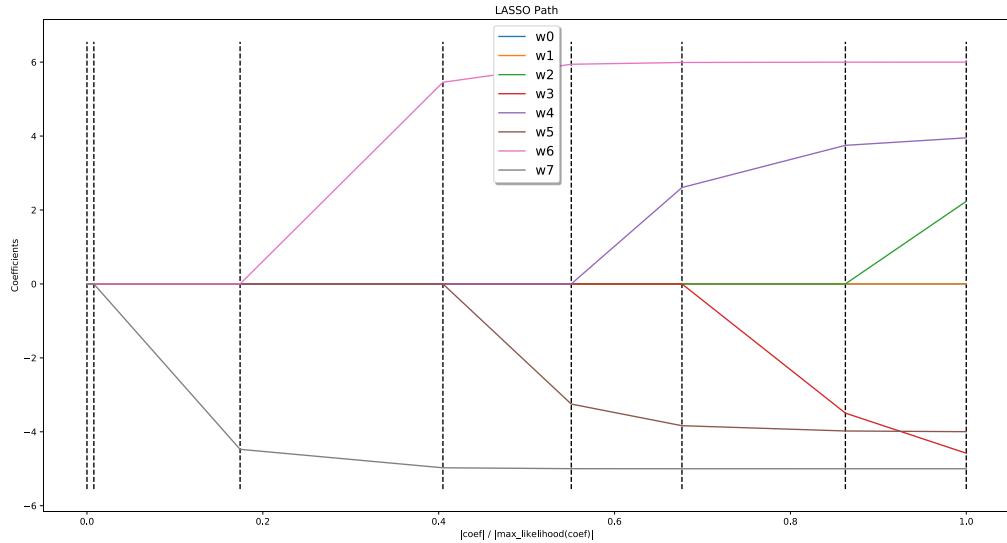


Figure 2 $n=70000, I=(0,25)$. The weights are mostly picked up in descending order.

Think about why this happens (Tip: With polynomial basis functions, we have $x^i \leq x^j, i < j, x \in [-1,1]$. What does this mean for our regularized error?).

If we put the length of our interval between the ascending ($I=(-1,1)$) and descending ($I=(0,25)$) cases, then the following more chaotic weight pick up happens:

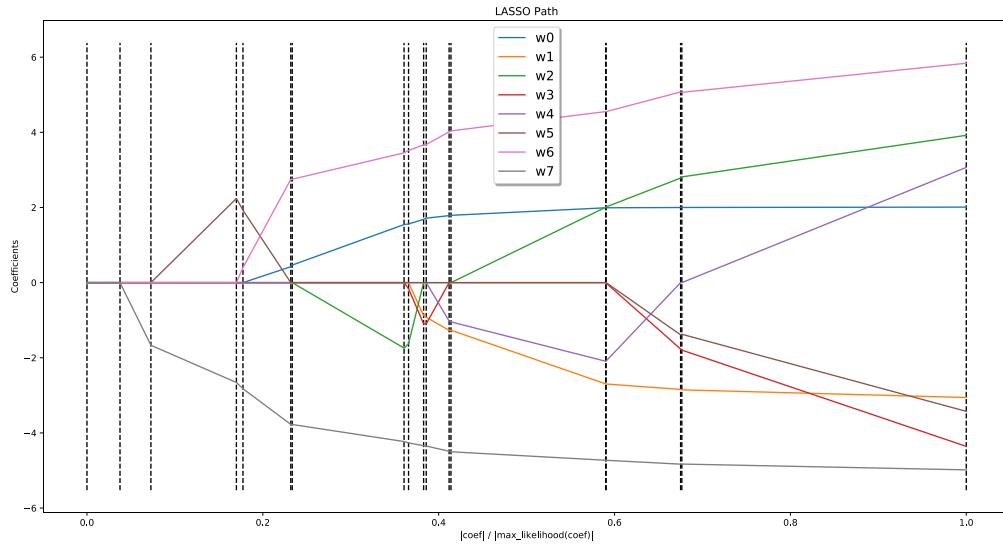


Figure 3 $n=70000, I=(0,2)$ Chaotic pick up of weights

6A-3.

The noise ϵ is symmetric around t . So by the same way we derived $t \sim N(t; \mathbf{w}^T \mathbf{x}, \epsilon)$ when the noise was normal, we can now say

$$t \sim \text{Stud}_t(t; \nu, \mathbf{w}^T \mathbf{x}, \sigma_\epsilon^2)$$

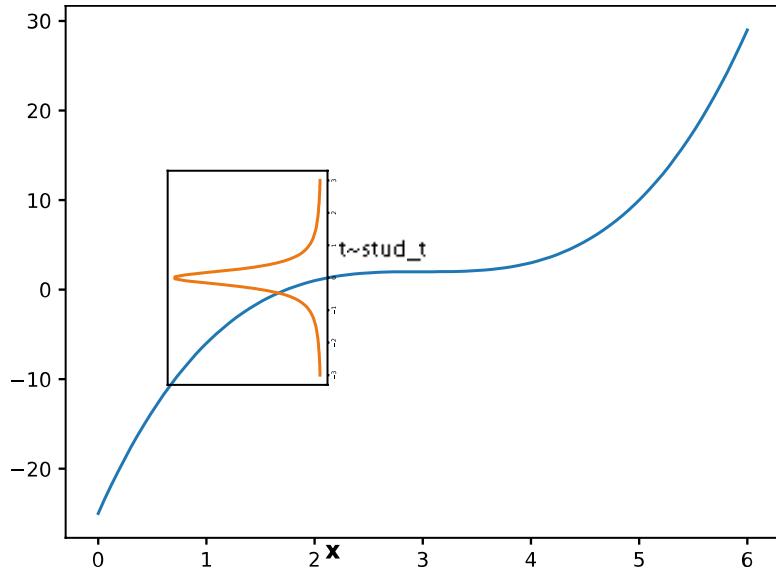


Figure 4 Because the noise is symmetrically distributed around t and the hidden function $f(x)$ is non-random ($p(t|x)=1$ for one value of t), t is identically distributed to its noise with adjusted mean.

This leads to the log likelihood:

$$\begin{aligned} \log p(\mathbf{t}|\mathbf{x}, \nu, \sigma_\epsilon) &= \log \prod_{i=1}^n \text{Stud_t}(t_i; \nu, \mathbf{w}^T \mathbf{x}_i, \sigma_\epsilon^2) \\ &= n \left[\log \Gamma\left(\frac{\nu+1}{2}\right) - \log \left(\sqrt{\nu \pi \sigma_\epsilon^2} \Gamma\left(\frac{\nu}{2}\right) \right) \right] \\ &\quad - \frac{(\nu+1)}{2} \sum_{i=1}^n \left(1 + \frac{1}{\nu} \frac{(t_i - \mathbf{w}^T \mathbf{x}_i)^2}{\sigma_\epsilon^2} \right) \end{aligned}$$

Let us set $d = (t_1 - \mathbf{w}^T \mathbf{x}_1)$. We now plot the error of one datapoint in comparison to a log normal pdf. The following picture was attained with the parameters σ

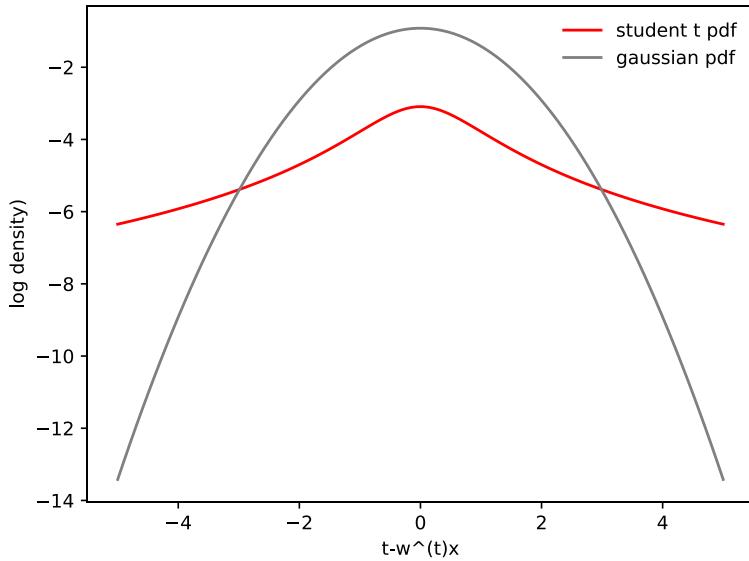


Figure 5 Comparison of the test error between a log normal pdf and a student t pdf. Image was attained with the parameters $\sigma_\epsilon = 1, v = 1$.

For small errors $\mathbf{w}^T \mathbf{x} \approx t$, the student t distribution appears to be less forgiving, as less density is concentrated around small errors ($\log \text{Stud_t}(t_1; 1, d, 1) < \log N(t_1; d, 1)$). Larger errors $|\mathbf{w}^T \mathbf{x}| \gg |t|$ are less penalized however, as $\log N(t_1; d, 1) < \log \text{Stud_t}(t_1; 1, d, 1)$.

Generally speaking, this means if we try to maximize our weights \mathbf{w} with respect to $e \sim N(0, \sigma^2)$, then our fit is trying to mitigate large errors while being mostly indifferent to smaller ones.

If we maximize \mathbf{w} with respect to $t \sim \text{Stud_t}(t; v, \mathbf{w}^T \mathbf{x}, \sigma_\epsilon^2)$, we get a more balanced fit, where small deviations are considered.