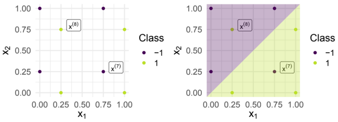
Introduction to Machine Learning

Regularization for Underdetermined Problems



Learning goals

- Understand that regularization is used to make ill-posed problems well defined
- Know that regularization can guarantee convergence for logistic regression on a linearly separable dataset

UNDERDETERMINED PROBLEMS

Regularization can also be motivated from a numerical perspective:

- Regularization can sometimes be necessary to make certain ill-posed problems well defined. Linear models such as (linear) regression and PCA depend "inverting" / solving a linear system, which not always works.
- When we solve linear systems like $\mathbf{X}\boldsymbol{\theta} = \boldsymbol{y}$, there are 3 cases:
  1. $\mathbf{X}$ is of square form and has full rank. This is normal linear system solving and irrelevant for us here, now.
  2. $\mathbf{X}$ has more rows than columns. The system is "overdetermined". We now try to solve $\mathbf{X}\boldsymbol{\theta} \approx \boldsymbol{y}$, by minimizing $||\mathbf{X}\boldsymbol{\theta} - \boldsymbol{y}||$. Ideally, this difference would be zero, but due to the too many rows this is often not possible. This is equivalent to linear regression!
  3. $\mathbf{X}$ has more columns than rows / linear dependence between columns exists. Now there are usually an infinite number of solutions. We have to define a "preference" for them to make the problem well-defined (sounds familiar?). Such problems are called ''underdetermined''.

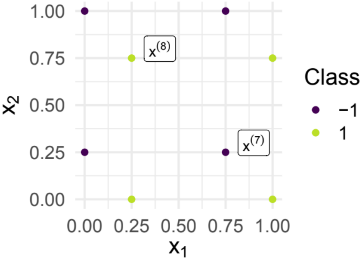
UNDERDETERMINED PROBLEMS

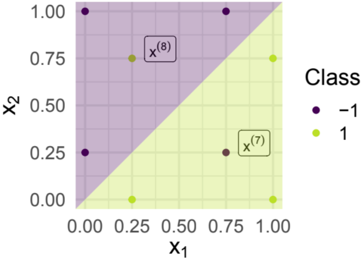
- A very old and well-known approach in underdetermined cases is to still reduce the problem to optimization by minimizing $||\mathbf{X}\boldsymbol{\theta} - \boldsymbol{y}||$, but adding a small positiv constant to the diagonal of $\mathbf{X}^T\mathbf{X}$.
- In optimization / numerical analysis this is known as **Tikhonov** regularization.
- But as you should be able to see now: This is completely equivalent to Ridge regression!

UNDERDETERMINED PROBLEMS

We now study not the normal LM (which we could), but logistic regression applied to a linearly separable dataset for a more subtle example:

First, we take a look at logistic regression for an almost linearly separable dataset consisting of the observations $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(8)}$.





Note: WLOG we estimate the model without intercept, s.t. we can visualize the regression coefficient $\boldsymbol{\theta}$ in 2D. Also, the symmetry of the data does not influence the generality of our conclusions.

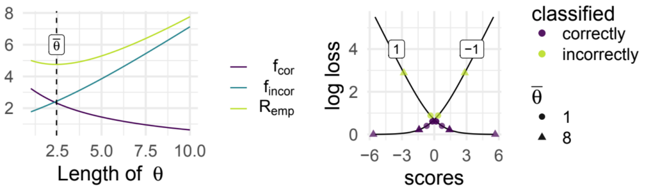
UNDERDETERMINED PROBLEMS

Because of the symmetry of the data, the direction[1] of $\boldsymbol{\theta}$ is $\tilde{\boldsymbol{\theta}} := (\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})^\top$.

To find $\overline{\theta} := ||\boldsymbol{\theta}||_2$, we consider the empirical risk $\mathcal{R}_{\text{emp}}$ along $\tilde{\boldsymbol{\theta}}$:

$$\begin{aligned}\mathcal{R}_{\text{emp}} &= \sum_{i=1}^{8} \log\left[1 + \exp\left(-y^{(i)}\boldsymbol{\theta}^\top \mathbf{x}^{(i)}\right)\right] \\ &= \underbrace{\sum_{i=1}^{6} \log\left[1 + \exp\left(-\overline{\theta}\left|\tilde{\boldsymbol{\theta}}^\top \mathbf{x}^{(i)}\right|\right)\right]}_{=:\, f_{\text{cor}}(\overline{\theta}) \text{ (correctly classified)}} + \underbrace{\sum_{i=7}^{8} \log\left[1 + \exp\left(\overline{\theta}\left|\tilde{\boldsymbol{\theta}}^\top \mathbf{x}^{(i)}\right|\right)\right]}_{=:\, f_{\text{incor}}(\overline{\theta}) \text{ (incorrectly classified)}}.\end{aligned}$$

Clearly, f_{cor} / f_{incor} are monotonically decreasing/increasing with rising length of $\boldsymbol{\theta}$:



---

[1] $\boldsymbol{\theta}$ is perpendicular to the decision boundary and points to the "1"-space.

UNDERCONSTRAINED PROBLEMS

- By removing the 7th and 8th observation, we get a linearly separable dataset.
- This also means that we lose our "counterweight", i.e., if a parameter vector $\boldsymbol{\theta}$ is able to classify the samples perfectly, the vector $2\boldsymbol{\theta}$ also classifies the samples perfectly, with decreased risk.
- Therefore, an iterative optimizer such as stochastic gradient descent (SGD) will continually increase $\boldsymbol{\theta}$ and never halt (in theory).
- In such cases, regularization can guarantee convergence: