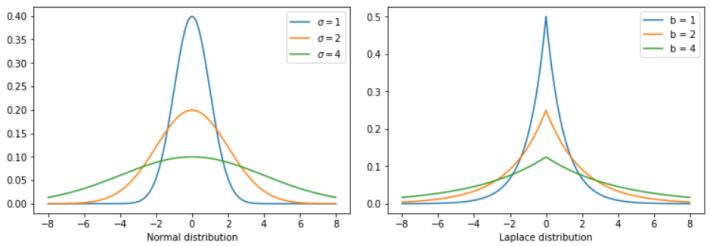
Introduction to Machine Learning

Regularization in Non-Linear Models and Bayesian Priors



Learning goals

- Understand that regularization and parameter shrinkage can be applied to non-linear models
- Know structural risk minimization
- Know how regularization risk minimization is the same as MAP in a Bayesian perspective, where the penalty corresponds to parameter prior.

SUMMARY: REGULARIZED RISK MINIMIZATION

If we should define ML in only one line, this might be it:

$$\min_{\boldsymbol{\theta}} \mathcal{R}_{\text{reg}}(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \left(\sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right) + \lambda \cdot J(\boldsymbol{\theta}) \right)$$

We can choose for a task at hand:

- the **hypothesis space** of f, which determines how features can influence the predicted y
- the **loss** function L, which measures how errors should be treated
- the **regularization** $J(\boldsymbol{\theta})$, which encodes our inductive bias and preference for certain simpler models

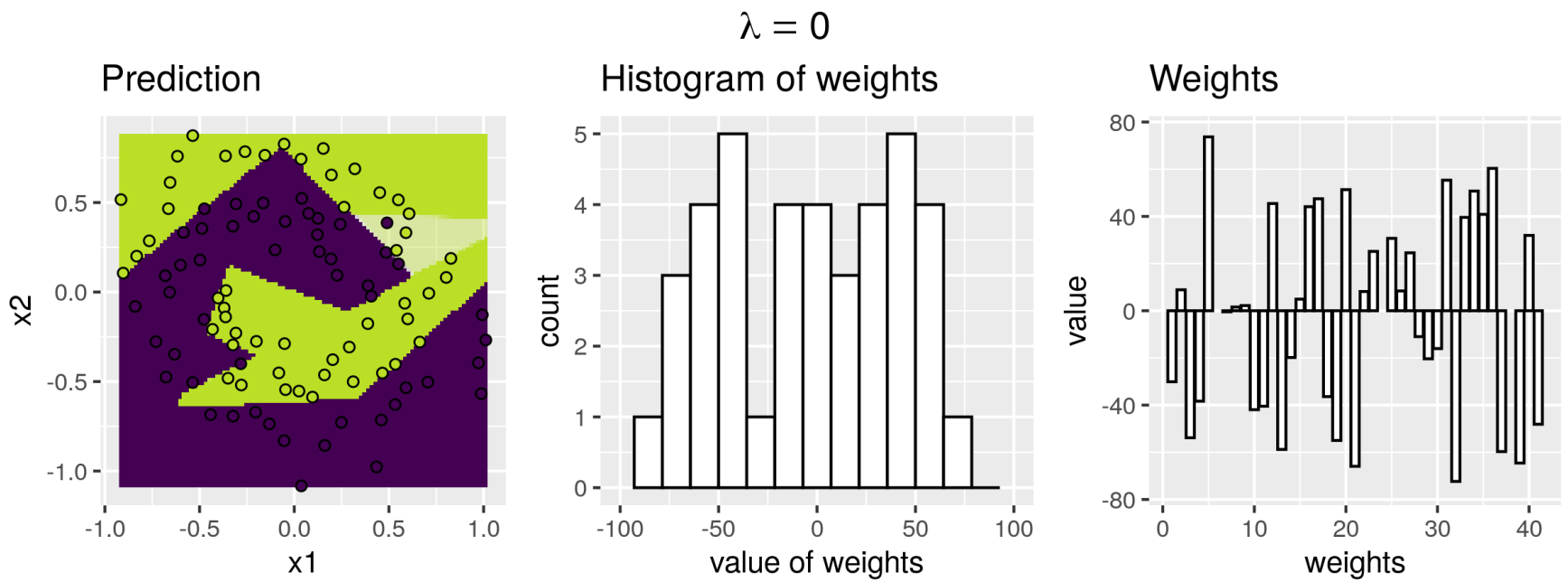
By varying these choices one can construct a huge number of different ML models. Many ML models follow this construction principle or can be interpreted through the lens of regularized risk minimization.

REGULARIZATION IN NONLINEAR MODELS

- So far we have mainly considered regularization in LMs.
- Can also be applied to non-linear models (with numeric parameters), where it is often important to prevent overfitting.
- Here, we typically use *L2* regularization, which still results in parameter shrinkage and weight decay.
- By adding regularization, prediction surfaces in regression and classification become smoother.
- Note: In the chapter on non-linear SVMs we will study the effects of regularization on a non-linear model in detail.

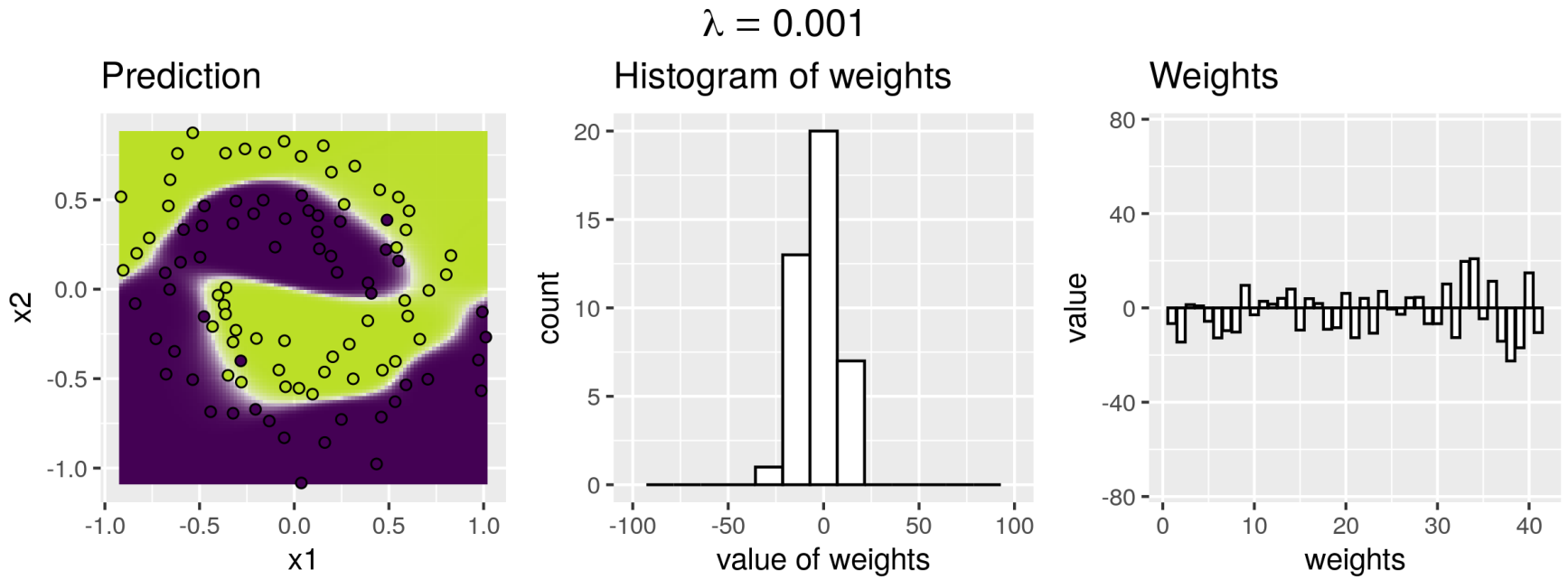
REGULARIZATION IN NONLINEAR MODELS

Setting: Classification for the `spirals` data. Neural network with single hidden layer containing 10 neurons and logistic output activation, regularized with *L2* penalty term for $\lambda > 0$. Varying λ affects smoothness of the decision boundary and magnitude of network weights:



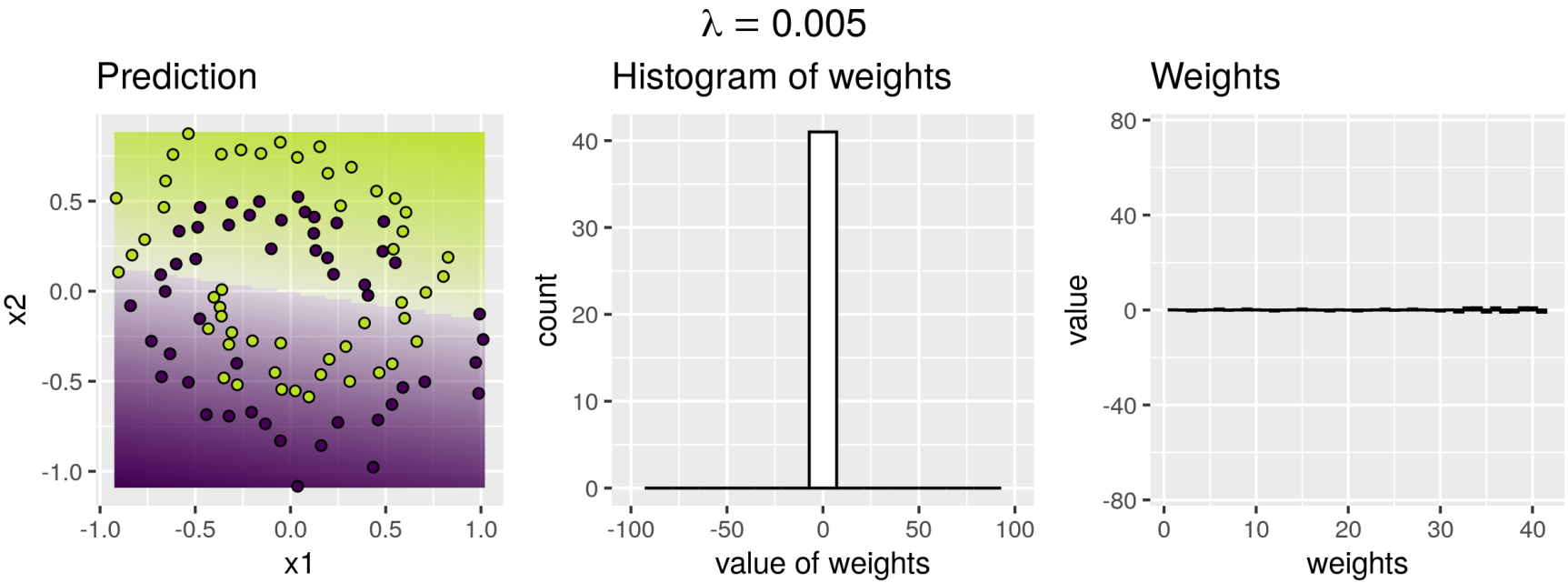
REGULARIZATION IN NONLINEAR MODELS

Setting: Classification for the `spirals` data. Neural network with single hidden layer containing 10 neurons and logistic output activation, regularized with *L2* penalty term for $\lambda > 0$. Varying λ affects smoothness of the decision boundary and magnitude of network weights:



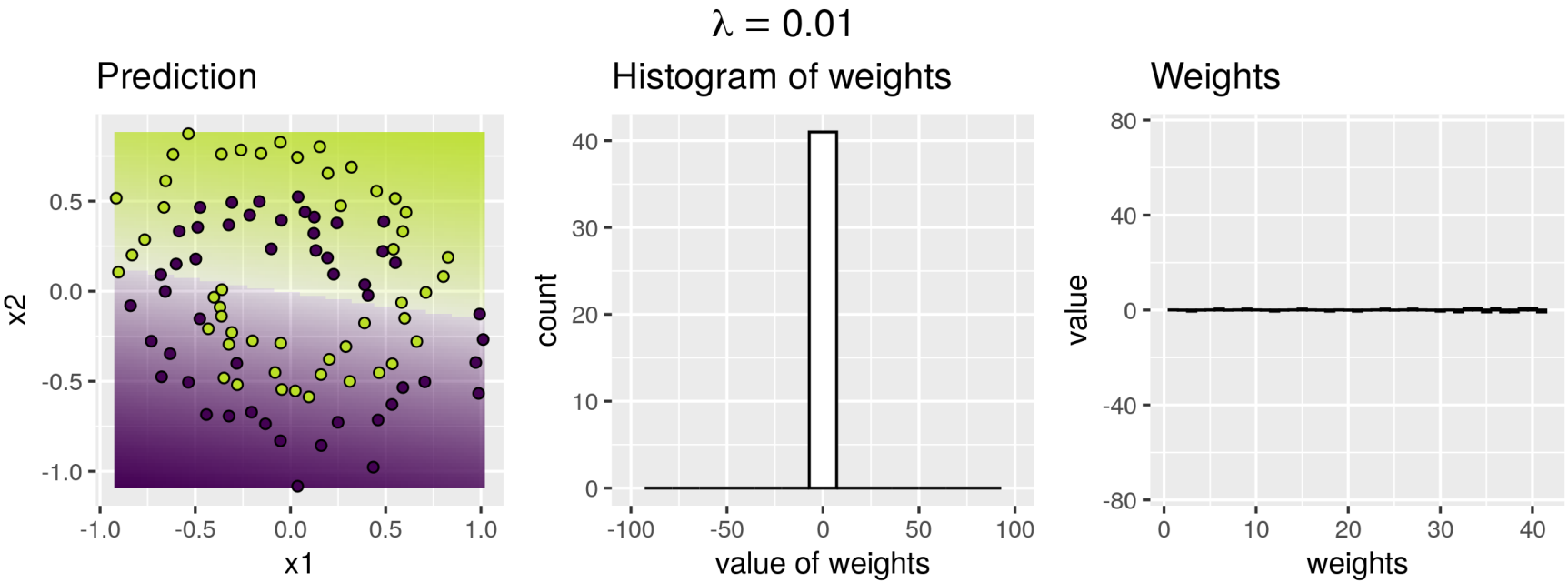
REGULARIZATION IN NONLINEAR MODELS

Setting: Classification for the `spirals` data. Neural network with single hidden layer containing 10 neurons and logistic output activation, regularized with *L2* penalty term for $\lambda > 0$. Varying λ affects smoothness of the decision boundary and magnitude of network weights:



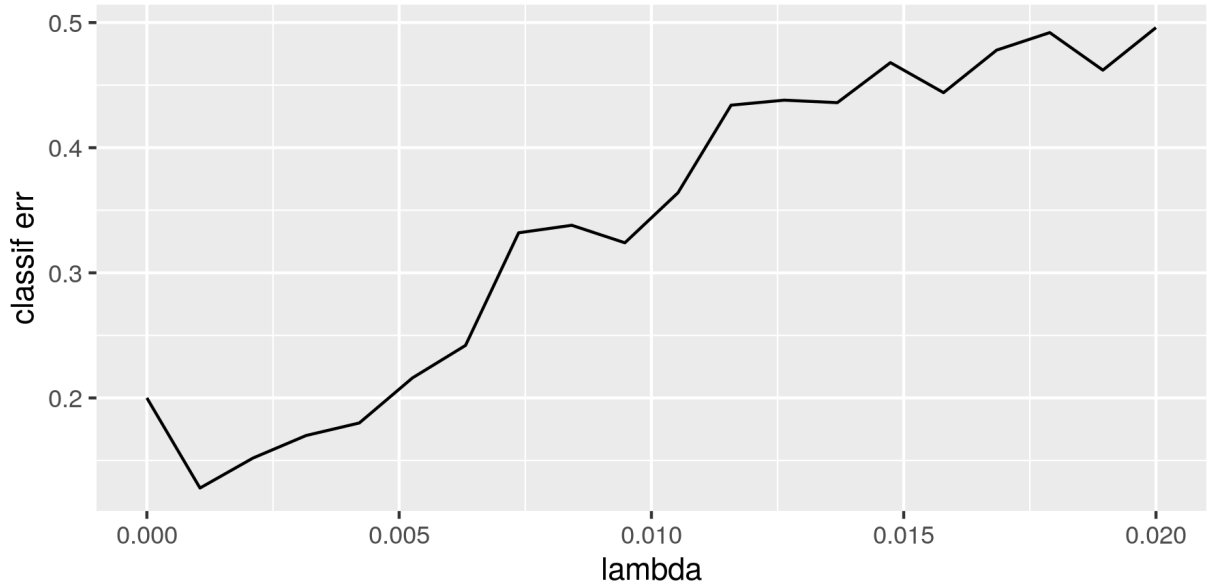
REGULARIZATION IN NONLINEAR MODELS

Setting: Classification for the `spirals` data. Neural network with single hidden layer containing 10 neurons and logistic output activation, regularized with *L2* penalty term for $\lambda > 0$. Varying λ affects smoothness of the decision boundary and magnitude of network weights:



REGULARIZATION IN NONLINEAR MODELS

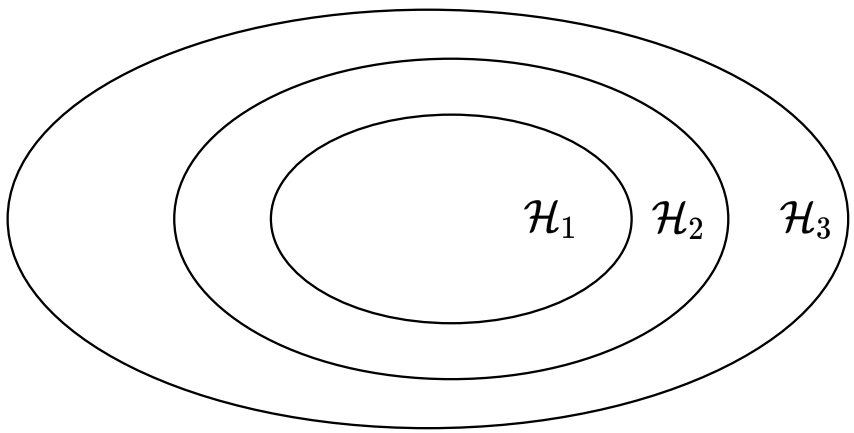
The prevention of overfitting can also be seen in CV. Same settings as before, but each λ is evaluated with repeated CV (10 folds, 5 reps).



We see the typical U-shape with the sweet spot between overfitting (LHS, low λ) and underfitting (RHS, high λ) in the middle.

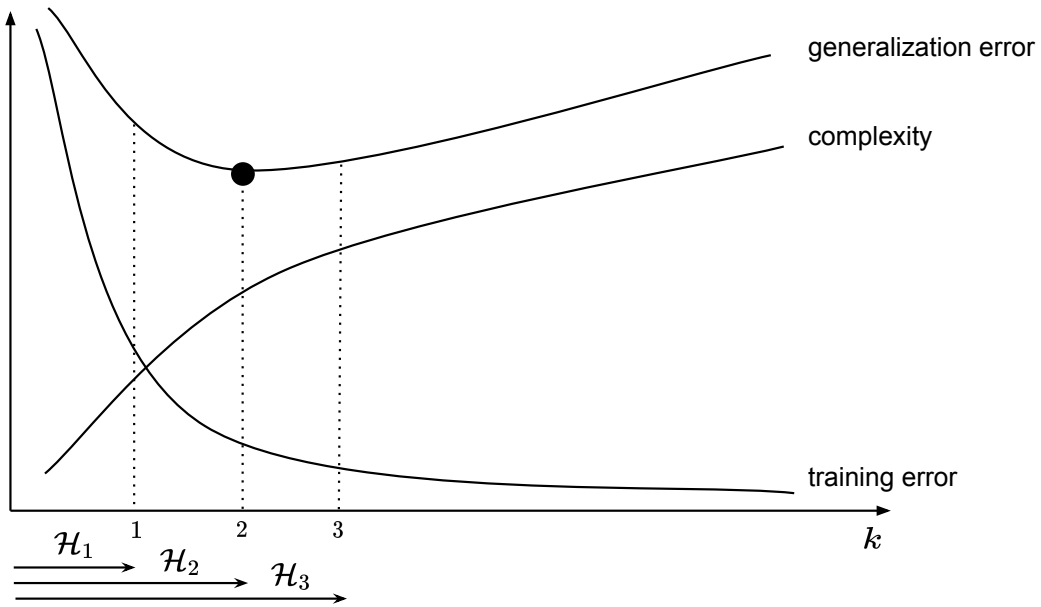
STRUCTURAL RISK MINIMIZATION

- Thus far, we only considered adding a complexity penalty to empirical risk minimization.
- Instead, structural risk minimization (SRM) assumes that the hypothesis space $\mathcal{H}$ can be decomposed into increasingly complex hypotheses (size or capacity): $\mathcal{H} = \cup_{k \geq 1} \mathcal{H}_k$.
- Complexity parameters can be the, e.g. the degree of polynomials in linear models or the size of hidden layers in neural networks.



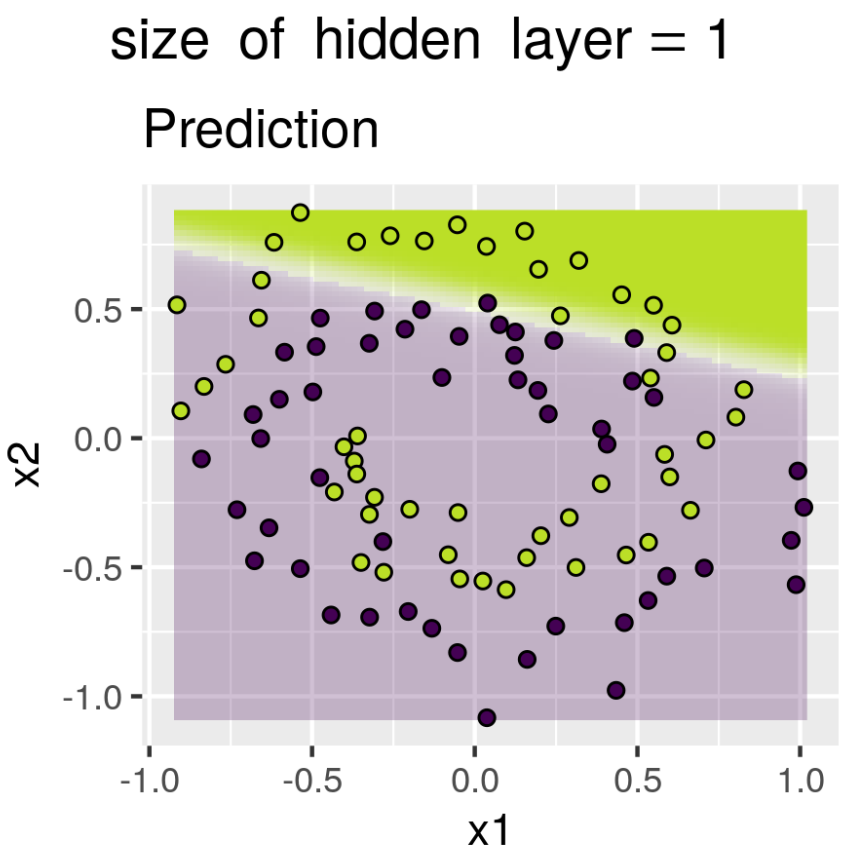
STRUCTURAL RISK MINIMIZATION

- SRM chooses the smallest k such that the optimal model from $\mathcal{H}_k$ found by ERM or RRM cannot significantly be outperformed by a model from a $\mathcal{H}_m$ with $m > k$.
- By this, the simplest model can be chosen, which minimizes the generalization bound.
- One challenge might be choosing an adequate complexity measure, as for some models, multiple complexity measures exist.



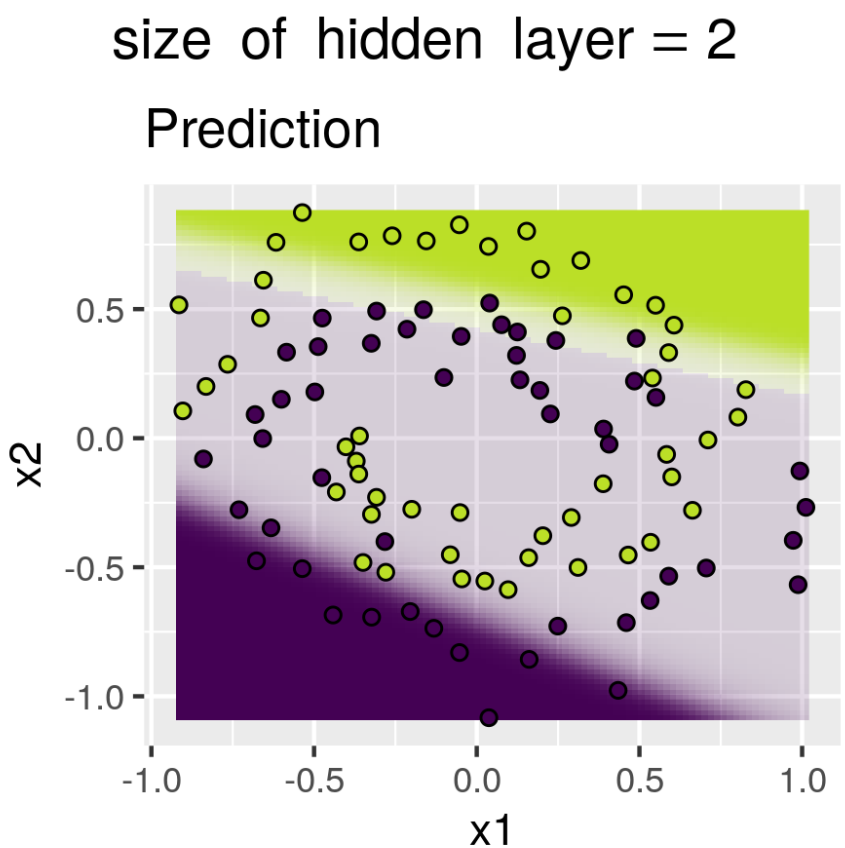
STRUCTURAL RISK MINIMIZATION

Setting: Classification for the `spirals` data. Neural network with single hidden layer containing k neurons and logistic output activation, L2 regularized with $\lambda = 0.001$. So here SRM and RRM are both used. Varying the size of the hidden layer affects smoothness of the decision boundary:



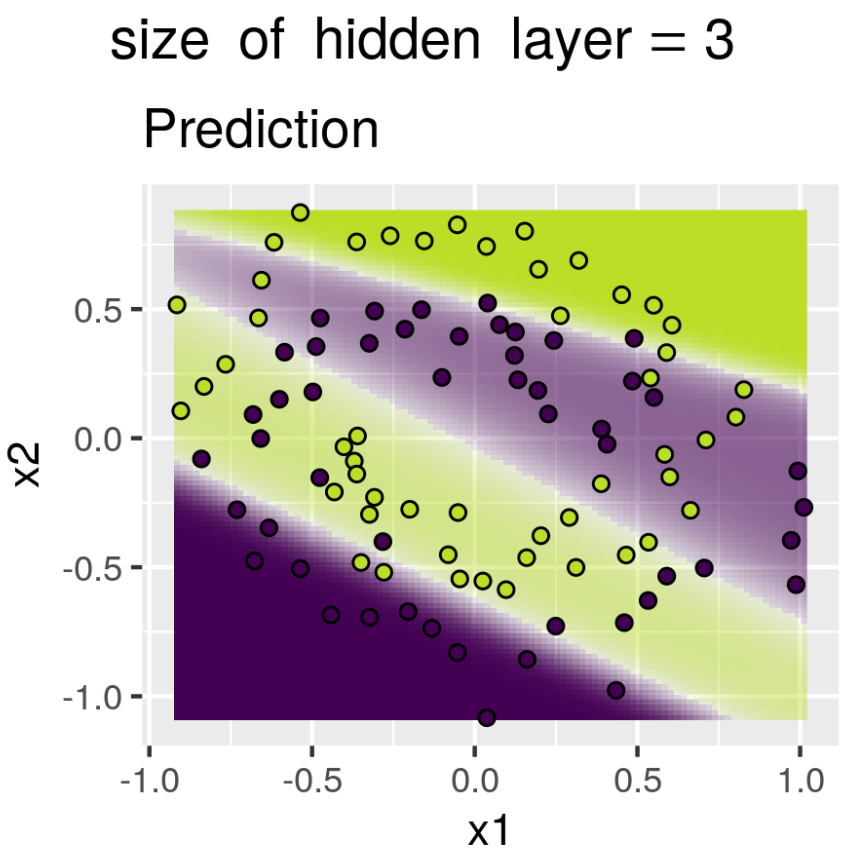
STRUCTURAL RISK MINIMIZATION

Setting: Classification for the `spirals` data. Neural network with single hidden layer containing k neurons and logistic output activation, L2 regularized with $\lambda = 0.001$. So here SRM and RRM are both used. Varying the size of the hidden layer affects smoothness of the decision boundary:



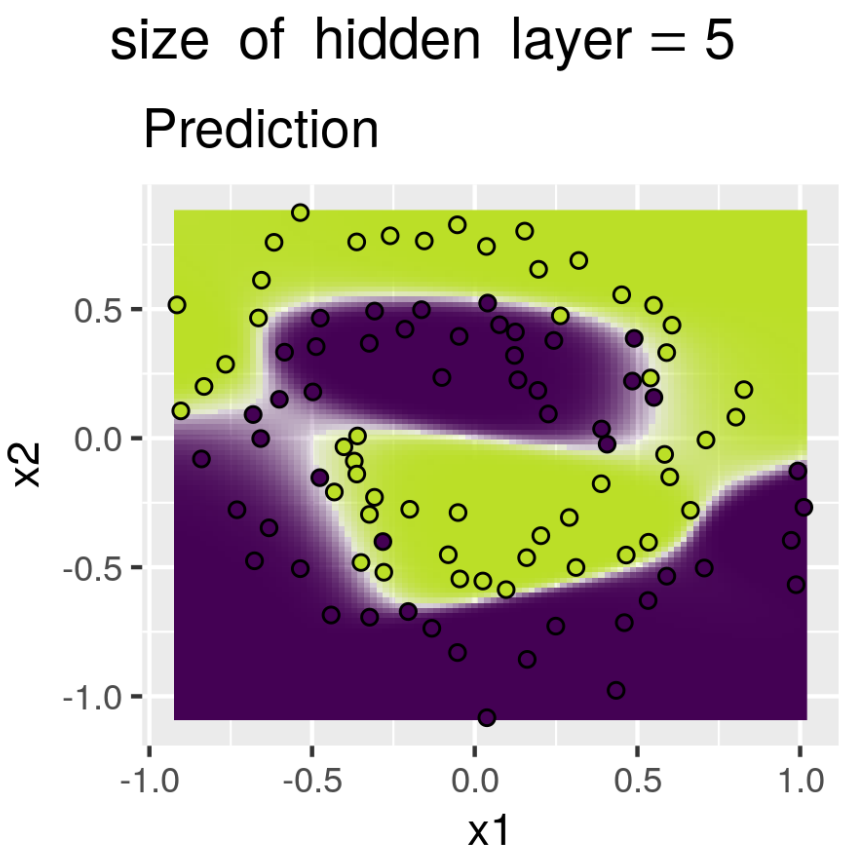
STRUCTURAL RISK MINIMIZATION

Setting: Classification for the `spirals` data. Neural network with single hidden layer containing k neurons and logistic output activation, L2 regularized with $\lambda = 0.001$. So here SRM and RRM are both used. Varying the size of the hidden layer affects smoothness of the decision boundary:



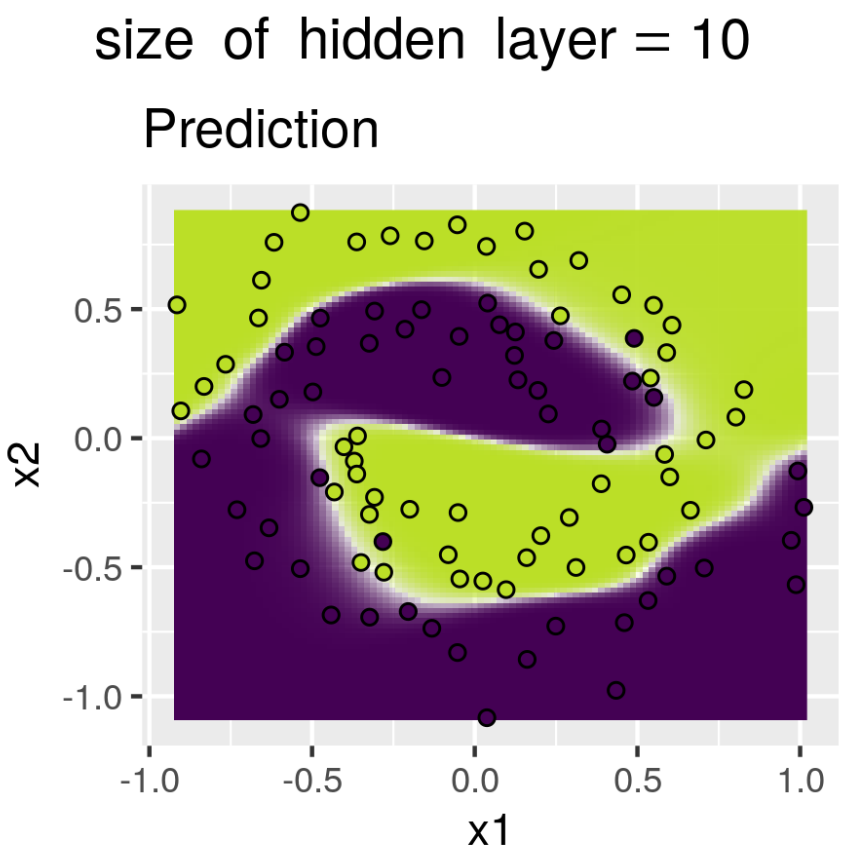
STRUCTURAL RISK MINIMIZATION

Setting: Classification for the `spirals` data. Neural network with single hidden layer containing k neurons and logistic output activation, L2 regularized with $\lambda = 0.001$. So here SRM and RRM are both used. Varying the size of the hidden layer affects smoothness of the decision boundary:



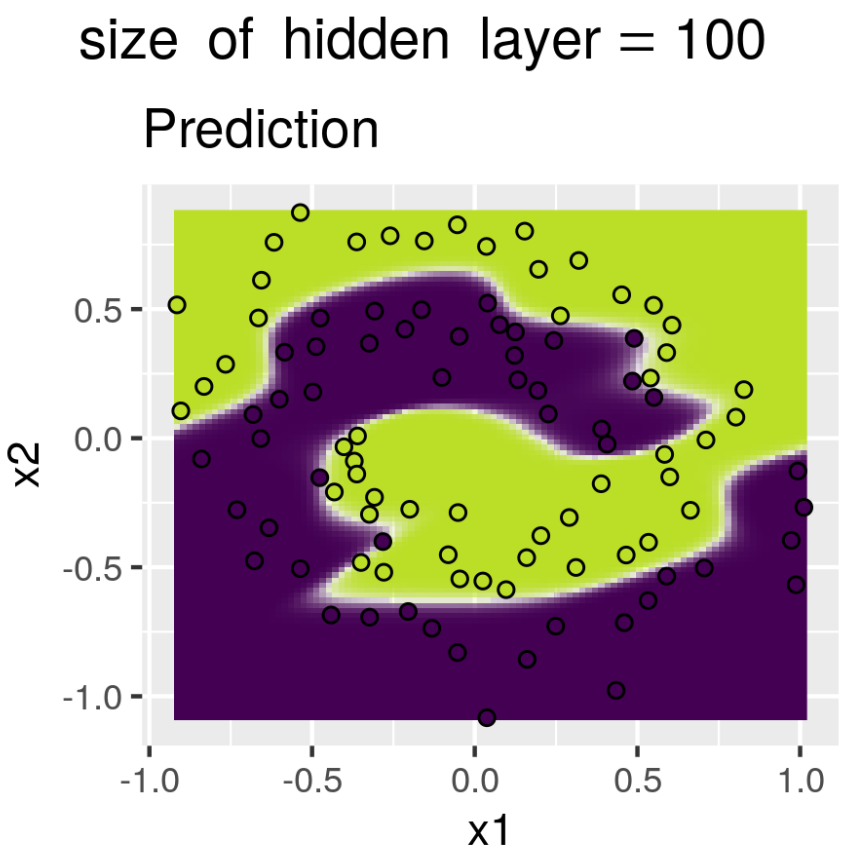
STRUCTURAL RISK MINIMIZATION

Setting: Classification for the `spirals` data. Neural network with single hidden layer containing k neurons and logistic output activation, L2 regularized with $\lambda = 0.001$. So here SRM and RRM are both used. Varying the size of the hidden layer affects smoothness of the decision boundary:



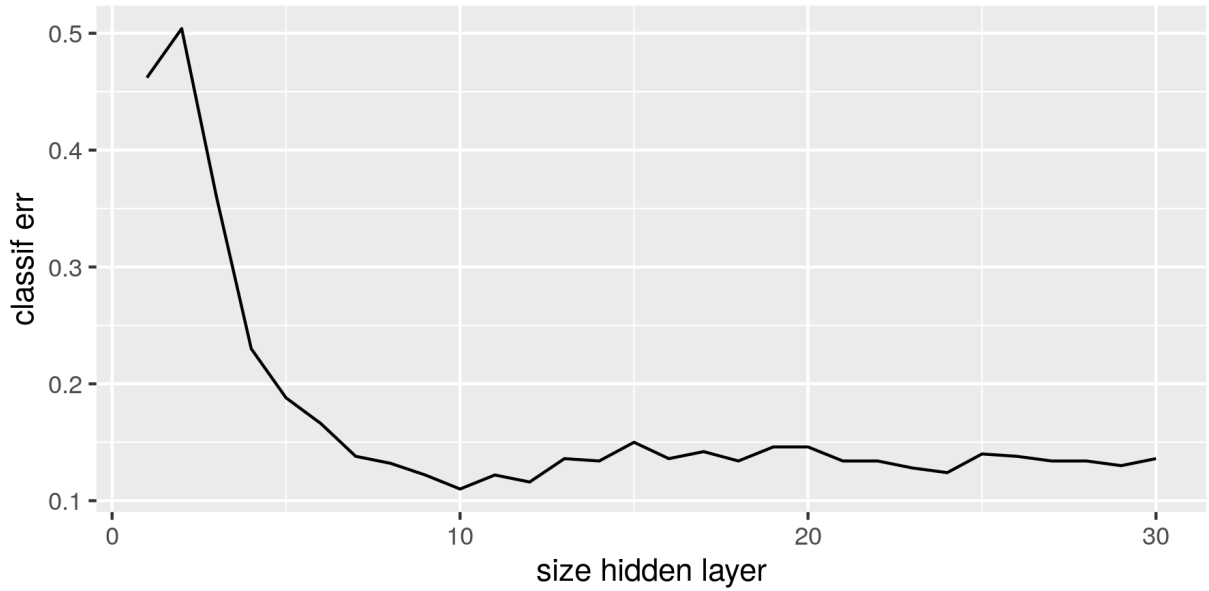
STRUCTURAL RISK MINIMIZATION

Setting: Classification for the `spirals` data. Neural network with single hidden layer containing k neurons and logistic output activation, L2 regularized with $\lambda = 0.001$. So here SRM and RRM are both used. Varying the size of the hidden layer affects smoothness of the decision boundary:



STRUCTURAL RISK MINIMIZATION

Again, complexity vs CV score.

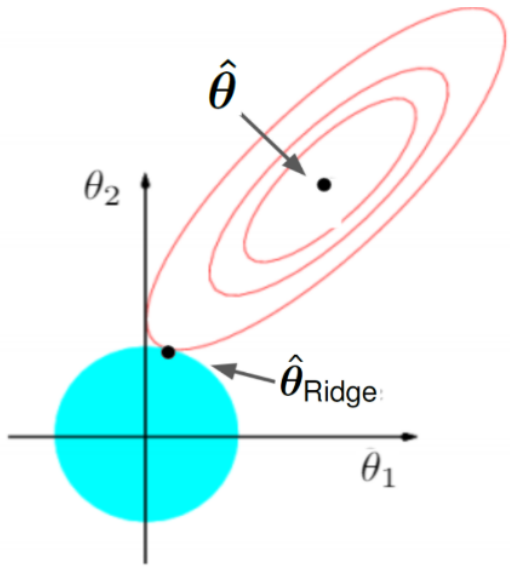


A minimal model with good generalization seems to have ca. 6-8 hidden neurons.

STRUCTURAL RISK MINIMIZATION AND RRM

Note that normal RRM can also be interpreted through SRM, if we rewrite the penalized ERM as constrained ERM.

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right) \\ \text{s.t.} \quad & \|\boldsymbol{\theta}\|_2^2 \leq t \end{aligned}$$



We can interpret going through λ from large to small as through t from small to large. This constructs a series of ERM problems with hypothesis spaces $\mathcal{H}_\lambda$, where we constrain the norm of $\boldsymbol{\theta}$ to unit balls of growing size.

RRM VS. BAYES

We already created a link between max. likelihood estimation and ERM.

Now we will generalize this for RRM.

Assume we have a parameterized distribution $p(y|\boldsymbol{\theta}, \mathbf{x})$ for our data and a prior $q(\boldsymbol{\theta})$ over our parameter space, all in the Bayesian framework.

From the Bayes theorem we know:

$$p(\boldsymbol{\theta}|\mathbf{x}, y) = \frac{p(y|\boldsymbol{\theta}, \mathbf{x})q(\boldsymbol{\theta})}{p(y|\mathbf{x})} \propto p(y|\boldsymbol{\theta}, \mathbf{x})q(\boldsymbol{\theta})$$

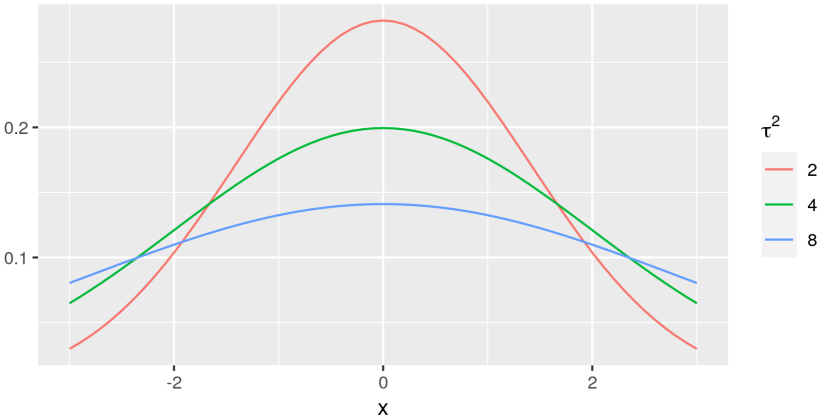
RRM VS. BAYES

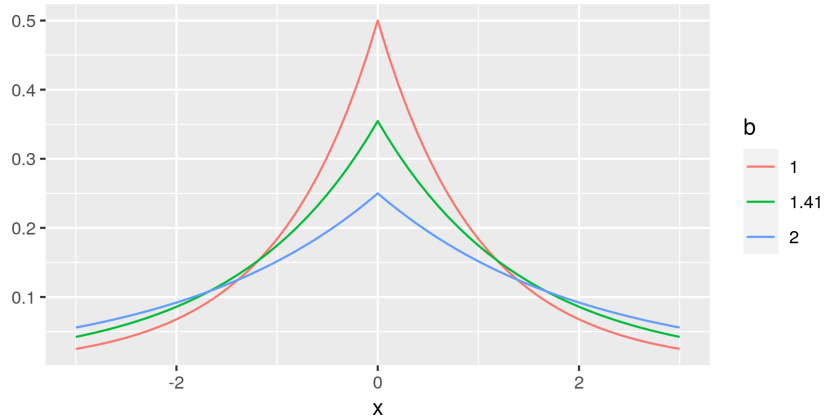
The maximum a posteriori (MAP) estimator of $\boldsymbol{\theta}$ is now the minimizer of

$$-\log p\left(y \mid \boldsymbol{\theta}, \mathbf{x}\right) - \log q(\boldsymbol{\theta}).$$

- Again, we identify the loss $L\left(y, f(\mathbf{x} \mid \boldsymbol{\theta})\right)$ with $-\log(p(y|\boldsymbol{\theta}, \mathbf{x}))$.
- If $q(\boldsymbol{\theta})$ is constant (i.e., we used a uniform, non-informative prior), the second term is irrelevant and we arrive at ERM.
- If not, we can identify $J(\boldsymbol{\theta}) \propto -\log(q(\boldsymbol{\theta}))$, i.e., the log-prior corresponds to the regularizer, and the additional λ, which controls the strength of our penalty, usually influences the peakedness / inverse variance / strength of our prior.

RRM VS. BAYES





- *L2* regularization corresponds to a zero-mean Gaussian prior with constant variance on our parameters: $\theta_j \sim \mathcal{N}(0, \tau^2)$
- *L*1 corresponds to a zero-mean Laplace prior: $\theta_j \sim \mathit{Laplace}(0, b)$. $\mathit{Laplace}(\mu, b)$ has density $\frac{1}{2b} \exp(-\frac{|\mu - x|}{b})$, with scale parameter b, mean μ and variance $2b^2$.
- In both cases, regularization strength increases as the variance of the prior decreases: a prior probability mass more narrowly concentrated around 0 encourages shrinkage.

EXAMPLE: BAYESIAN L2 REGULARIZATION

We can easily see the equivalence of *L2* regularization and a Gaussian prior:

- We define a Gaussian prior with uncorrelated components for $\boldsymbol{\theta}$:

$$q(\boldsymbol{\theta}) = \mathcal{N}_d(\mathbf{0}, diag(\tau^2)) = \prod_{j=1}^{d} \mathcal{N}(0, \tau^2) = (2\pi\tau^2)^{-\frac{d}{2}} \exp\left(-\frac{1}{2\tau^2}\sum_{j=1}^{d}\theta_j^2\right).$$

- With this, the MAP estimator becomes

$$\begin{aligned}
\hat{\theta}^{\text{MAP}} &= \arg\min_{\boldsymbol{\theta}} \left(-\log p\left(y \mid \boldsymbol{\theta}, \mathbf{x}\right) - \log q(\boldsymbol{\theta})\right) \\
&= \arg\min_{\boldsymbol{\theta}} \left(-\log p\left(y \mid \boldsymbol{\theta}, \mathbf{x}\right) + \tfrac{d}{2}\log(2\pi\tau^2) + \frac{1}{2\tau^2}\sum_{j=1}^{d}\theta_j^2\right) \\
&= \arg\min_{\boldsymbol{\theta}} \left(-\log p\left(y \mid \boldsymbol{\theta}, \mathbf{x}\right) + \frac{1}{2\tau^2}\|\boldsymbol{\theta}\|_2^2\right).
\end{aligned}$$

- We see how the inverse variance (precision) $1/\tau^2$ controls shrinkage.