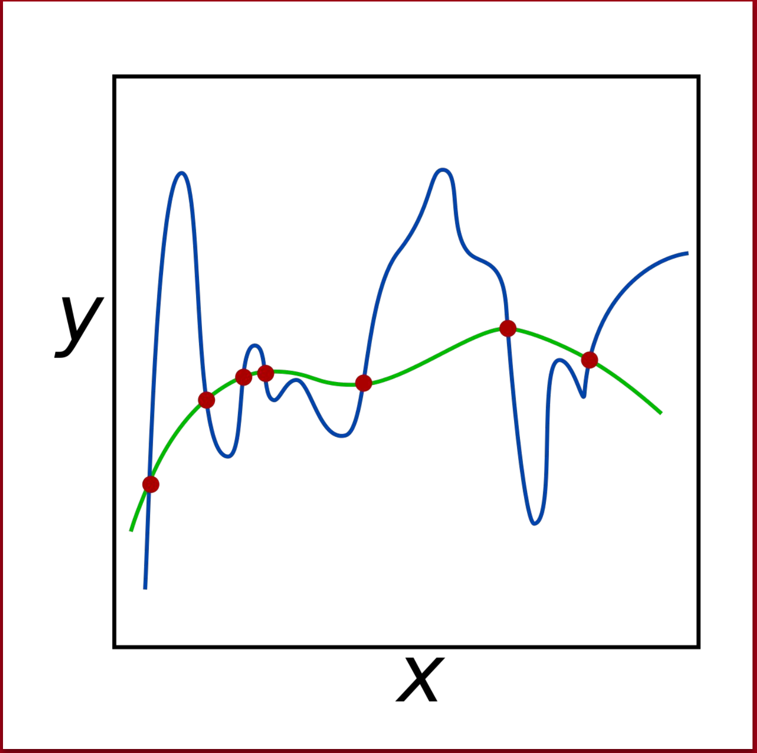
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

UNIVERSITÀ
DEGLI STUDI
DI PADOVA



Regularization and Stability

Machine Learning 2022-23
UML book chapter 13

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Regularized Loss Minimization (RLM)

Key idea: jointly minimize empirical risk and a regularization function

- Hypothesis *h*: defined by a vector $\boldsymbol{w} = (w_1, \dots, w_d)^T \in \mathbb{R}^d$
  - e.g., coefficients of a linear model, weights in a neural network, etc..
- *Regularization function* $R: \mathbb{R}^d \rightarrow \mathbb{R}$, function of $\boldsymbol{w}$
- *Regularized Loss Minimization (RLM)*: select *h* from:

$$argmin_{\boldsymbol{w}}\left(L_s(\boldsymbol{w}) + R(\boldsymbol{w})\right)$$

- $L_s(\boldsymbol{w})$: standard loss for the considered problem
- $R(\boldsymbol{w})$: regularization term (measures in some way the "*complexity*" of the found solution)

- The regularization term balances between low empirical risk and aiming at less complex hypotheses
- It is possible to view the extra term as a "*stabilizer*"

Tikhonov Regularization

Tikhonov Regularization

- Define function *R* using the `l2 norm` of the weights:

$$R(\boldsymbol{w}) = \lambda ||\boldsymbol{w}||^2 = \lambda \sum_{i=1}^{d} \boldsymbol{w}_i^2$$

- Output of function *R* is a real positive number
- Learning Rule: $A(s) = argmin_{\boldsymbol{w}}(L_s(\boldsymbol{w}) + \lambda ||\boldsymbol{w}||^2)$

- $||\boldsymbol{w}||^2$: measures the "*complexity*" of the hypothesis defined by ***w***
- λ: controls the amount of regularization
  - It controls the trade-off between empirical error and complexity
  - Low empirical error but risk of overfitting or higher empirical error but lower complexity

Ridge Regression

Ridge Regression:

Linear Regression with squared loss + *Tikhonov regularization*

Linear Regression with squared loss: find **w** that minimizes the squared loss

$$\boldsymbol{w} = argmin_{\boldsymbol{w}} \sum_{i=1}^{m} (< \boldsymbol{w}, \boldsymbol{x_i} > - y_i)^2$$

Ridge Regression : find **w** that minimizes

$$\boldsymbol{w} = argmin_{\boldsymbol{w}} \left(\lambda ||\boldsymbol{w}||^2 + \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} (< \boldsymbol{w}, \boldsymbol{x_i} > - y_i)^2 \right)$$

λ balances between the 2 targets

Balancing should not depend on the size of training set

Closed Form Solution

- Find optimal ***w*:** minimize loss ($\lambda||\mathbf{w}||^2 + \frac{1}{m}\sum_i \frac{1}{2}(<\mathbf{w},\mathbf{x}_i> - y_i)^2$)

- Compute gradient w.r.t. ***w*** and set to 0

$$\frac{\partial L}{\partial \mathbf{w}} = 2\lambda\mathbf{w} + \frac{1}{m}\sum_{i=1}^{m}(<\mathbf{w},\mathbf{x}_i> - y_i)\mathbf{x}_i = 0 \rightarrow 2\lambda m\mathbf{w} + \sum_{i=1}^{m}\langle\mathbf{w},\mathbf{x}_i\rangle\mathbf{x}_i = \sum_{i=1}^{m} y_i\mathbf{x}_i$$

- Set (as for standard least squares)

$$A = \left(\sum_{i=1}^{m}\mathbf{x}_i\mathbf{x}_i^T\right) = \begin{bmatrix} \vdots & & \vdots \\ \mathbf{x}_1 & \cdots & \mathbf{x}_m \\ \vdots & & \vdots \end{bmatrix}\begin{bmatrix} \cdots & \mathbf{x}_1 & \cdots \\ & \vdots & \\ \cdots & \mathbf{x}_m & \cdots \end{bmatrix} \quad \mathbf{b} = \sum_{i=1}^{m} y_i\mathbf{x}_i = \begin{bmatrix} \vdots & & \vdots \\ \mathbf{x}_1 & \cdots & \mathbf{x}_m \\ \vdots & & \vdots \end{bmatrix}\begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

- The solution can be rewritten as*:

$$2\lambda mI\,\mathbf{w} + A\mathbf{w} = \mathbf{b} \rightarrow \mathbf{w} = (2\lambda mI + A)^{-1}\,\mathbf{b}$$

*differently from standard least square in this case the matrix is always invertible

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Tikhonov Regularization and Stability

- Tikhonov regularization makes the learner stable w.r.t. small perturbations of the training set
  - this in turn leads to small bounds on generalization error

- Informally: an algorithm A is stable if a small change of the training data *S* (i.e., its input) will lead to a small change of its output hypothesis
  - what is a "small change of the training data"?
  - what is a "small change of its output hypothesis"?

Stability

- "*small change of the training data*" = replace one sample!
  - Given $S = (z_1, \dots, z_m)$ and an additional example z' (i.e., pair instance label/target) let $S^{(i)} = (z_1, \dots, z_{i-1}, z', z_{i+1}, \dots, z_m)$
- "small change of its output hypothesis" = small change in the loss
  - *On-Average-Replace-One-Stable* (OAROS) algorithms

Definition:
Let be $\epsilon: \mathbb{N} \to \mathbb{R}$ a monotonically decreasing function. We say that a learning algorithm *A* is *on-average-replace-one-stable* (OAROS) with rate $\epsilon(m)$ if for every distribution *D*:

$$\mathbb{E}_{(S,z') \sim D^{m+1}, i \sim U(m)}\left[l\left(A\left(S^{(i)}\right), z_i\right) - l\left(A(S), z_i\right)\right] \le \epsilon(m)$$

Draw IID from D

Select at random which to replace

With z' in place of z_i

Depends on training set size

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Stable Rules do not Overfit

Theorem:
If algorithm A is OAROS with rate $\epsilon(m)$ then:

$$\mathbb{E}_{S\sim D^m}[L_D(A(S)) - L_S(A(S))] \leq \epsilon(m)$$

Demonstration

1. True error: expected loss on one IID sample (from *D*):

$$\forall i:\ \mathbb{E}_S\big[L_D\big(A(S)\big)\big] = \mathbb{E}_{S,z'}\,[l(A(S),z')] = \mathbb{E}_{S,z'}\,\big[l\big(A\big(S^{(i)}\big),z_i\big)\big]$$

2. Training error: average error on one sample **in training set**:

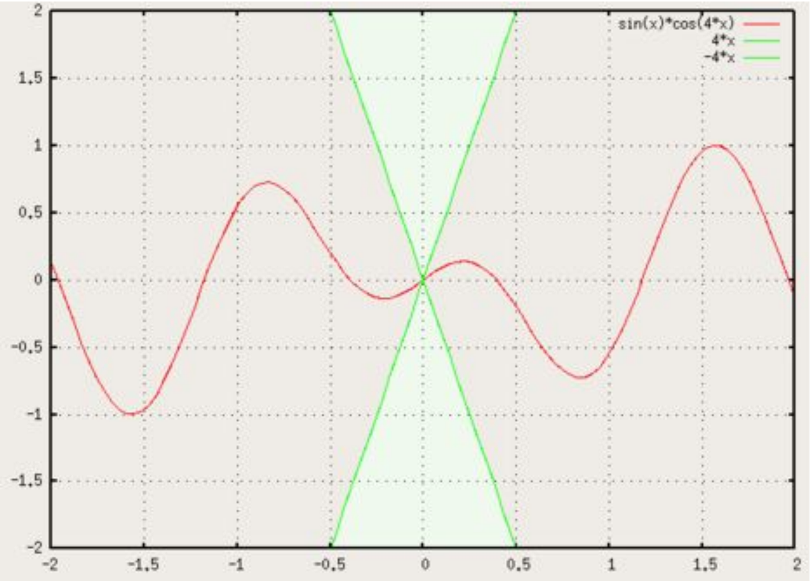
$$\mathbb{E}_S\big[L_S\big(A(S)\big)\big] = \mathbb{E}_{S,i}\,[l(A(S),z_i)]$$

3. Combine (1)+(2) and exploit linearity of expectation and OAROS def.

$$\mathbb{E}_S\big[L_D\big(A(S)\big) - L_S\big(A(S)\big)\big] = \mathbb{E}_{S,z',i}\,\big[l\big(A\big(S^{(i)}\big),z_i\big) - l(A(S),z_i)\big] \leq \epsilon(m)$$

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Lipschitzness



Definition (Lipschitzness):

- Let $C \subset \mathbb{R}^d$. A function $f: \mathbb{R}^d$ -> $\mathbb{R}^k$ is ρ-Lipschitz over C if $\forall \boldsymbol{w_1}, \boldsymbol{w_2} \in C$, we have that $||f(\boldsymbol{w_1}) - f(\boldsymbol{w_2})|| \leq \rho ||\boldsymbol{w_1} - \boldsymbol{w_2}||$

- Intuitively: the function cannot change too fast
- For derivable functions corresponds to bound on derivative:
  - If derivative bounded by ρ at any point ⇒ function is ρ-Lipschitz

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Tikhonov Regularization is a Stabilizer

Theorem:
Assume the loss function is convex and ρ-Lipschitz continuous.
Then, the RLM rule with regularizer $\lambda||\boldsymbol{w}||^2$ is OAROS with rate $\frac{2\rho^2}{\lambda m}$.
It follows that for the RLM rule:

$$\mathbb{E}_{S\sim D^m}[L_D(A(S)) - L_S(A(S))] \leq \frac{2\rho^2}{\lambda m}$$

- Tikhonov Regularization is a Stabilizer
- Larger λ leads a more stable solution (→ less overfitting)
- Larger training set also leads to more stable solution
- *First step*: demonstration not part of the course
- *Second step*: consequence of previous theorem

Fitting-Stability Trade-off (1)

$$E_S[L_D(A(S))] = E_S[L_S(A(S))] + E_S[L_D(A(S)) - L_S(A(S))]$$

- $E_S[L_S(A(S))]$: how well A fits the training set S
- $E_S[L_D(A(S)) - L_S(A(S))]$: measures overfitting, bounded by stability of A

In Tikhonov regularization, λ controls tradeoff between the 2 terms

- how do $L_S(A(S))$ and $||\boldsymbol{w}||^2$ vary as a function of λ ?
  - Larger λ leads to higher empirical risk $L_S(A(S))$
- how may $E_S[L_D(A(s)) - L_S(A(S))]$ change as a function of λ ?
  - On the other side increasing λ the stability term $E_S[L_D(A(s)) - L_S(A(S))]$ decreases
- How to set λ ?
  - Theoretical bound in the book

Fitting-Stability Trade-off (2)

$$E_S[L_D(A(S))] = E_S[L_S(A(S))] + E_S[L_D(A(S)) - L_S(A(S))]$$

- $E_S[L_S(A(S))]$: how well A fits the training set S
- $E_S[L_D(A(S)) - L_S(A(S))]$: measures overfitting, bounded by stability of A

Small λ: focus on training error
Training error L_S : small
Difference $L_D - L_S$: large
Overfitting the training data

Large λ: focus on regularization
Training error L_S : large
Difference $L_D - L_S$: small
Underfitting the training data