Introduction to Machine Learning NPFL 054

http://ufal.mff.cuni.cz/course/npfl054

Barbora Hladká
hladka@ufal.mff.cuni.cz

Martin Holub
holub@ufal.mff.cuni.cz

Charles University,
Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics

Lecture #10

Outline

- Model complexity, overfitting, bias and variance
- Regularization – Ridge regression, Lasso
  - Linear regression
  - Logistic regression
  - SVM

Model complexity

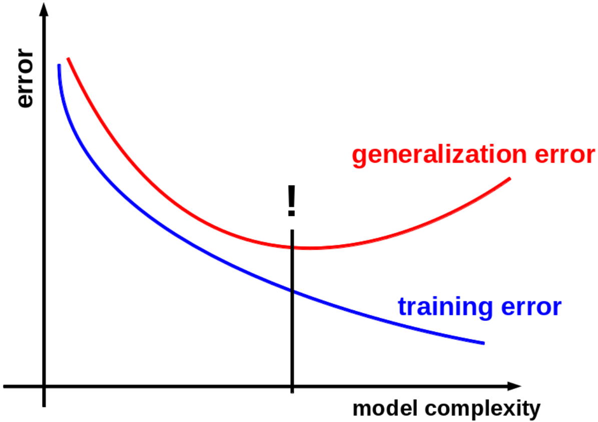
No universal definition

Here . . . **model complexity** is the number of hypothesis parameters

$$\Theta = \langle \theta_0, \ldots, \theta_m \rangle$$

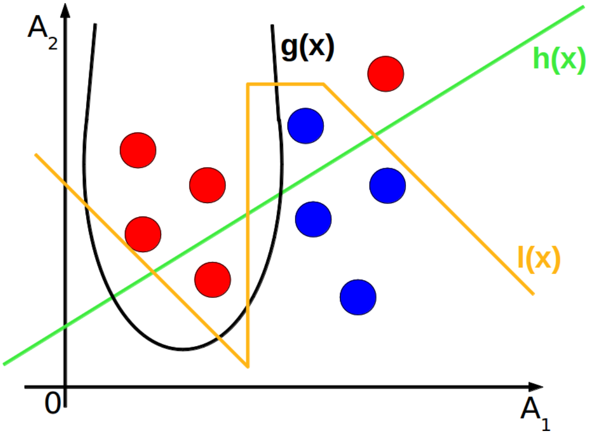
Model complexity

Finding a model that minimizes generalization error
... is one of central goals of the machine learning process



Model complexity

Complexity of decision boundary for classification

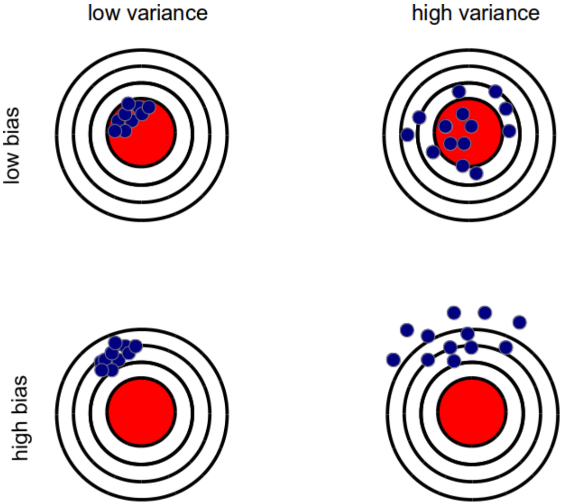


Bias and variance

1. Select a machine learning algorithm
2. Get k different training sets
3. Get k predictors

- **Bias** measures error that originates from the learning algorithm
  – how far off in general the predictions by k predictors are from the true output value
- **Variance** measures error that originates from the training data
  – how much the predictions for a test instance vary between k predictors

Bias and variance



Bias and variance

Generalization error $\mathrm{error}_{\mathcal{D}}(f)$ measures how well a hypothesis f generalizes beyond the used training data set, to unseen data with distribution $\mathcal{D}$. Usually it is defined as follows

- for **regression**: $\mathrm{error}_{\mathcal{D}}(f) = \mathsf{E}\,(\hat{y}_i - y_i)^2$
- for **classification**: $\mathrm{error}_{\mathcal{D}}(f) = \mathsf{Pr}\,(\hat{y}_i \neq y_i)$

Decomposition of *error*$_{\mathcal{D}}(f)$

$$error_{\mathcal{D}}(f) = \mathrm{Bias}^2 + \mathrm{Variance}$$

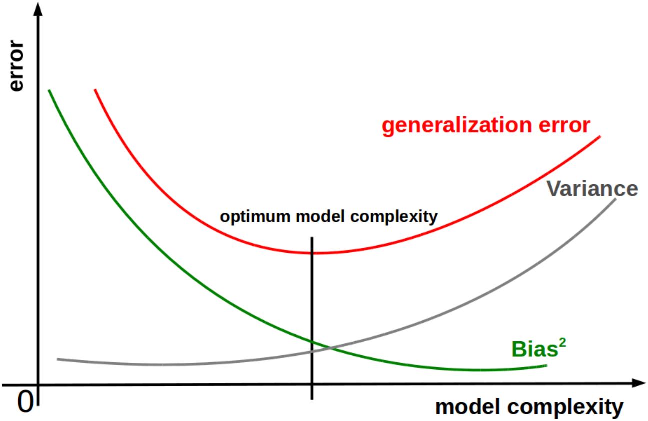
i.e.,

$$(E[\hat{f}(\mathbf{x}] - f(\mathbf{x}))^2 + E[\hat{f}(\mathbf{x}) - E[\hat{f}(\mathbf{x})]]^2$$

where $\hat{f}(\mathbf{x})$ is predicted value, $E[\hat{f}(\mathbf{x})]$ is average predicted value

Bias and variance

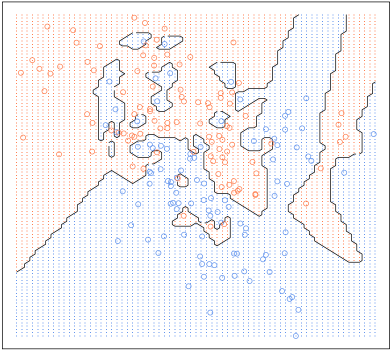
- underfitting = high bias
- overfitting = high variance



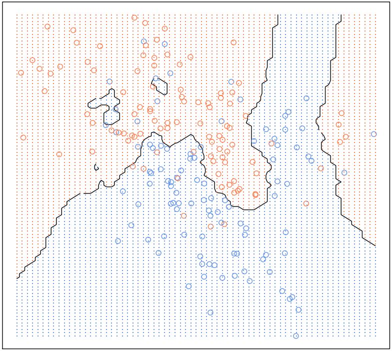
Bias and variance
k-Nearest Neighbor

- $\uparrow k \rightarrow$ smoother decision boundary $\rightarrow \downarrow$ variance and $\uparrow$ bias
- $\downarrow k \rightarrow \uparrow$ variance and $\downarrow$ bias

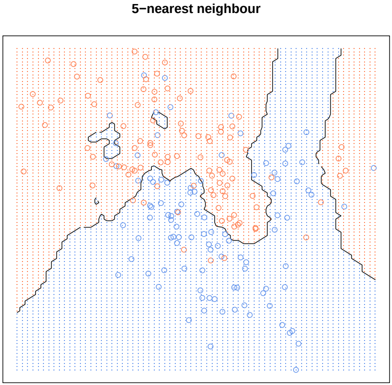
1−nearest neighbour

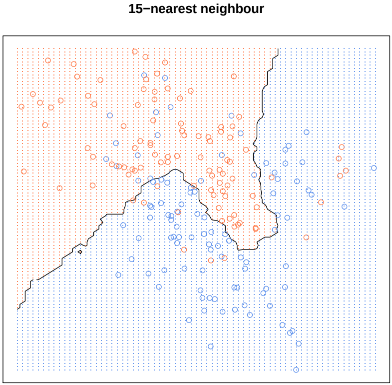


5−nearest neighbour



Bias and variance
k-Nearest Neighbor





Prevent overfitting

We want a model in between which is

- powerful enough to model the underlying structure of data
- not so powerful to model the structure of the training data

Let's prevent overfitting by **complexity regularization**,
a technique that regularizes the parameter estimates, or equivalently, shrinks the parameter estimates towards zero.

Regularization

A machine learning algorithm
estimates hypothesis parameters $\Theta = \langle \theta_0, \theta_1, \dots, \theta_m \rangle$
using $\Theta^\star$ that minimizes loss function L
for training data $Data = \{\langle \mathbf{x}_i, y_i \rangle, \mathbf{x}_i = \langle x_{1i}, \dots, x_{mi} \rangle, y_i \in Y\}$

$$\Theta^\star = \mathrm{argmin}_\Theta \mathrm{L}(\Theta)$$

Regularization

$$\Theta_R^\star = \mathrm{argmin}_\Theta \mathrm{L}(\Theta) + \lambda \cdot \textbf{penalty}(\Theta), \text{ where } \lambda \geq 0 \text{ is a tuning parameter}$$

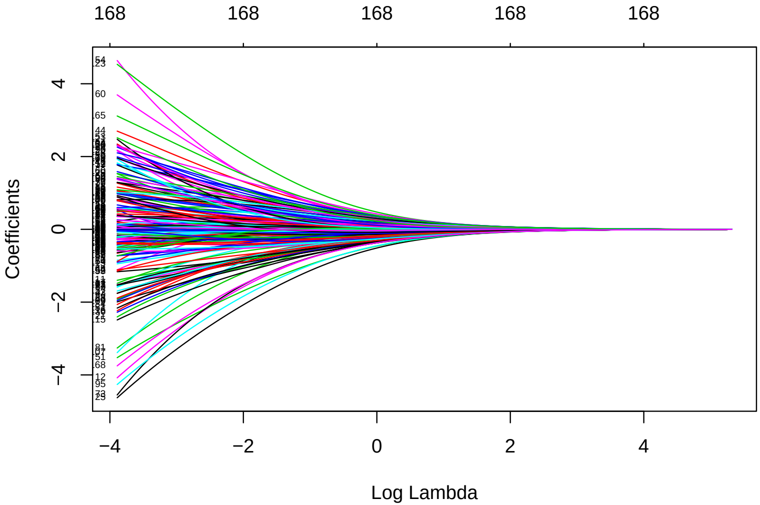
Infact, the penalty is applied to $\theta_1, \dots, \theta_m$, but not to θ_0 since the goal is to regularize the estimated association between each feature and the target value.

Regularization
Ridge regression

$$\text{penalty}(\Theta) = \theta_1^2 + \cdots + \theta_m^2 = \ell_2\text{norm}$$

- Let $\theta^\star_{\lambda_1}, \ldots, \theta^\star_{\lambda_m}$ be ridge regression parameter estimates for a particular value of λ
- Let $\theta^\star_1, \ldots, \theta^\star_m$ be unregularized parameter estimates
- $0 \leq \frac{{\theta^\star_{\lambda_1}}^2 + \cdots + {\theta^\star_{\lambda_m}}^2}{{\theta^\star_1}^2 + \cdots + {\theta^\star_m}^2} \leq 1$
- **When** $\lambda = 0$, **then** $\theta^\star_{\lambda_i} = \theta^\star_i$ for $i = 1, \ldots, m$
- **When** λ is extremely large, **then** $\theta^\star_{\lambda_i}$ is very small for $i = 1, \ldots, m$
- **When** λ between, we are fitting a model and skrinking the parameteres

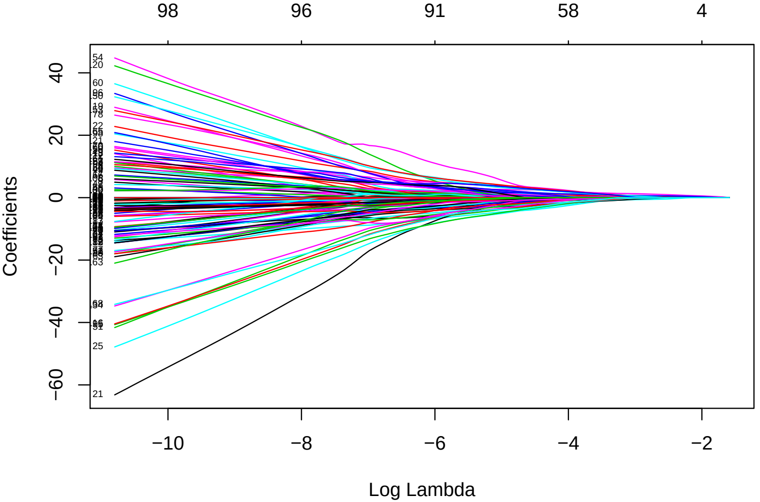
Ridge regression



Regularization
Lasso

$$\text{penalty}(\Theta) = |\theta_1| \ + \cdots + |\theta_m| = \ell_1\text{norm}$$

- Let $\theta^\star_{\lambda_1}, \ldots, \theta^\star_{\lambda_m}$ be lasso regression parameter estimates
- Let $\theta^\star_1, \ldots, \theta^\star_m$ be unregularized parameter estimates
- **When** $\lambda = 0$, **then** $\theta^\star_{\lambda_i} = \theta^\star_i$ for $i = 1, \ldots, m$
- **When** λ grows, **then** the impact of penalty grows
- **When** λ is extremely large, **then** $\theta^\star_{\lambda_i} = 0$ for $i = 1, \ldots, m$



98
96
91
58
4
Coefficients
40
20
0
−20
−40
−60
−10
−8
−6
−4
−2
Log Lambda

Ridge regression and Lasso

Ridge regression shrinks all the parameters but eliminates none, while the Lasso can shrink some parameters to zero.

Elastic net

$$\Theta^{\star}_{R} = \mathrm{argmin}_{\Theta}[\mathrm{L}(\Theta) + \lambda \cdot (|\theta_1| \ + \cdots + |\theta_m|) + (1-\lambda) \cdot (\theta_1^2 + \cdots + \theta_m^2)]$$

$0 \leq \lambda \leq 1$ is a tuning parameter

Loss function

A loss function $\mathrm{L}(\hat{y}, y)$ measures the cost of predicting $\hat{y}$ when the true value is $y \in \{-1, +1\}$. Commonly used loss functions are

- **Zero-one** $(0/1)$ $\mathrm{L}(\hat{y}, y) = \mathrm{I}(y\hat{y} \leq 0)$
  indicator variable I is 1 if $y\hat{y} \leq 0$, 0 otherwise
- **Hinge** $\mathrm{L}(\hat{y}, y) = \max(0, 1 - y\hat{y})$
- **Logistic** $\mathrm{L}(\hat{y}, y) = \max(0, \log(1 + e^{-y\hat{y}}))$
- **Exponential** $\mathrm{L}(\hat{y}, y) = e^{-y\hat{y}}$

Regularized linear regression

$$f(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \cdots + \theta_m x_m$$

$$\mathrm{L}(\Theta) = RSS = \sum_{i=1}^{n} (f(\mathbf{x}_i) - y_i)^2$$

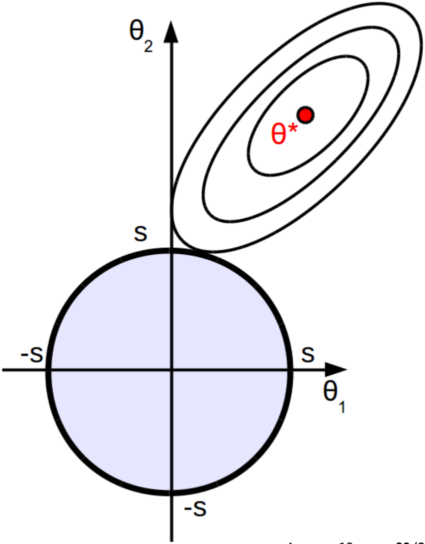
$$\Theta_R^\star = \mathrm{argmin}_\Theta [RSS + \lambda \cdot \mathrm{penalty}(\Theta)]$$

Ridge regression
Alternative formulation

$$\Theta_R^\star = \underset{\Theta}{\mathrm{argmin}} \sum_{i=1}^{n} (f(\mathbf{x}_i) - y_i)^2$$

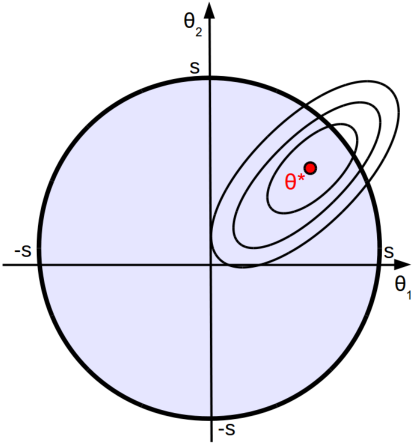
subject to $\theta_1^2 + \cdots + \theta_m^2 \leq s$

- the gray circle represents the feasible region for Ridge regression
- the contours represent different loss values for the unregularized model



Ridge regression
Alternative formulation

- If s is large enough so that the minimum loss value falls into the region of **ridge regression** parameter estimates then the alternative formulation yields the primary solution.

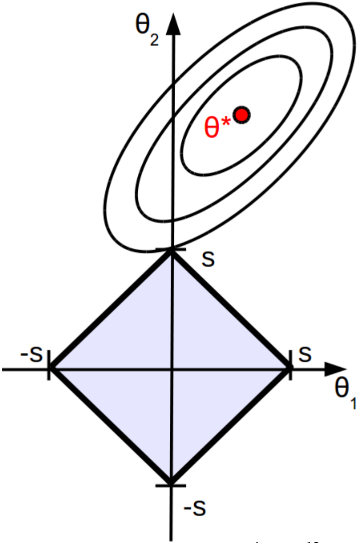


Lasso
Alternative formulation

$$\Theta_R^\star = \underset{\Theta}{\operatorname{argmin}} \sum_{i=1}^{n} (f(\mathbf{x}_i) - y_i)^2$$

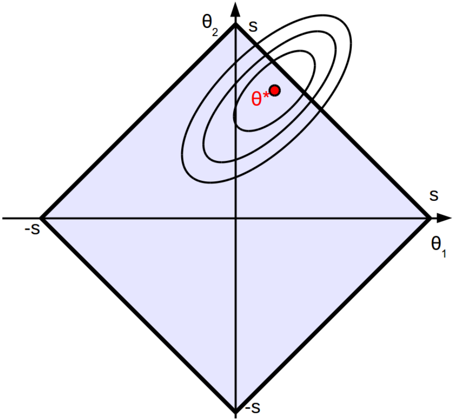
subject to $|\theta_1| + \cdots + |\theta_m| \leq s$

- the grey square represents the feasible region of the Lasso
- the contours represent different loss values for the unregularized model
- the feasible point that minimizes the loss is more likely to happen on the coordinates on the Lasso graph than on the Ridge regression graph since the Lasso graph is more angular



Lasso
Alternative formulation

- If s is large enough so that the minimum loss value falls into the region of **loss** parameter estimates then the alternative formulation yields the primary solution.



Regularized logistic regression

$$f(\mathbf{x}) = \frac{1}{1 + e^{-\Theta^\top \mathbf{x}}}$$

$$\mathrm{L}(\Theta) = -\sum_{i=1}^{n} y_i \log \mathrm{P}(y_i|\mathbf{x_i}; \Theta) + (1 - y_i) \log(1 - \mathrm{P}(y_i|\mathbf{x_i}; \Theta))$$

$$\Theta_R^\star = \mathrm{argmin}_\Theta [\mathrm{L}(\Theta) + \lambda \cdot \mathrm{penalty}(\Theta)]$$

Regularized logistic regression
Ridge regression

$$\Theta_R^\star = \mathrm{argmin}_\Theta - [\sum_{i=1}^{n} y_i \log(f(\mathbf{x}_i)) + (1 - y_i)\log(1 - f(\mathbf{x}_i))] + \lambda \sum_{j=1}^{m} \theta_j^2] =$$

$$= \mathrm{argmin}_\Theta [\sum_{i=1}^{n} y_i(-\log(f(\mathbf{x}_i))) + (1 - y_i)(-\log(1 - f(\mathbf{x}_i))) + \lambda \sum_{j=1}^{m} \theta_j^2] =$$

$$= \mathrm{argmin}_\Theta [\sum_{i=1}^{n} y_i L_1(\mathbf{\Theta}) + (1 - y_i) L_0(\mathbf{\Theta}) + \lambda \sum_{j=1}^{m} \Theta_j^2]$$

Regularized logistic regression
Ridge regression

Since

$$\mathbf{A} + \lambda\mathbf{B} \equiv C\mathbf{A} + \mathbf{B},\, C = \frac{1}{\lambda}$$

then

$$\Theta_R^\star = \mathrm{argmin}_\Theta[\sum_{j=1}^{m} \theta_j^2 + C[\sum_{i=1}^{n} y_i L_1(\Theta) + (1 - y_i)L_0(\Theta)]]$$

where

$L_1(\Theta) = -\log \frac{1}{1+e^{-\Theta^\top \mathbf{x}}}$

$L_0(\Theta) = -\log(1 - \frac{1}{1+e^{-\Theta^\top \mathbf{x}}})$

Regularized logistic regression
Ridge regression

$$\Theta_R^\star = \mathrm{argmin}_\Theta[\sum_{j=1}^{m} \theta_j^2 + C \sum_{i=1}^{n} \log(1 + e^{-\overline{y_i}\Theta^\top \mathbf{x}_i})]$$

where

$$\overline{y}_i = \begin{cases} -1 \ \mathit{if} & y_i = 0 \\ +1 \ \mathit{if} & y_i = 1 \end{cases}$$

SVM

$$\Theta^\star = \text{argmin}_\Theta \sum_{j=1}^{m} \theta_j^2 + C \sum_{i=1}^{n} \xi_i$$

$\xi_i \geq 0$ is equivalent to $\xi_i = \max(0, 1 - y_i\Theta^\top \mathbf{x}_i)$, i.e.

$$\Theta^\star = \text{argmin}_\Theta [\sum_{j=1}^{m} \theta_j^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i\Theta^\top \mathbf{x}_i)]$$

$$\text{s.t. } \Theta^\top \mathbf{x}_i \geq 1 - \xi_i \text{ if } y_i = +1 \text{ and } \Theta^\top \mathbf{x}_i \leq -1 + \xi_i \text{ if } y_i = -1$$

Hinge loss $= \max(0, 1 - y_i\Theta^\top \mathbf{x})$

1. $y_i\Theta^\top \mathbf{x}_i > 1$: no contribution to loss
2. $y_i\Theta^\top \mathbf{x}_i = 1$: no contribution to loss
3. $y_i\Theta^\top \mathbf{x}_i < 1$: contribution to loss

SVM

Soft-margin is equivalent to the regularization problem.

Summary of Examination Requirements

- Model complexity, generalization error, Bias and variance
- Lasso and Ridge regularization for linear and logistic regression
- Soft margin classifier and regularization