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 #9

Outline

- Evaluation of binary classification (cntnd) – ROC curve
- Model complexity, overfitting, bias and variance
- Regularization – Ridge regression, Lasso
    - Linear regression
    - Logistic regression

Evaluation of binary classifiers
Sensitivity vs. specificity

Confusion matrix

		Predicted class		
		Positive	**Negative**	
True class	**Positive**	True Positive (TP)	False Negative (FN)	P
	Negative	False Positive (FP)	True Negative (TN)	N

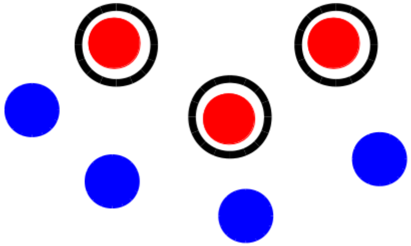
Measure	**Formula**
Precision	TP/(TP+FP)
Recall/Sensitivity/TPR	TP/(TP+FN) = TP/P
Specificity	TN/(TN+FP)
1-Specificity/FPR	FP/(TN+FP) =FP/N
Accuracy	(TP+TN)/(TP+FP+TN+FN)

Evaluation of binary classifiers
Sensitivity vs. specificity

Seven training examples

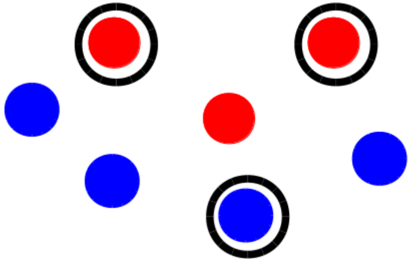
Classifier's output – examples in black circle are positives, other examples are negatives

Perfect classifier – no error



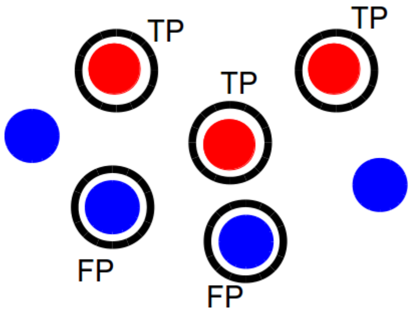
Evaluation of binary classifiers
Sensitivity vs. specificity

Reality – e.g. 2 miclassified examples
sensitivity $= 2/3$, specificity $= 3/4$



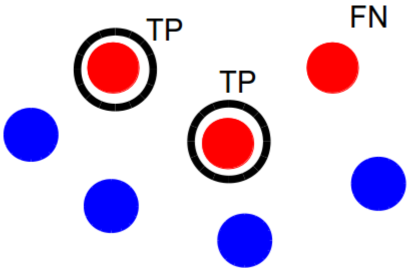
Evaluation of binary classifiers
Sensitivity vs. specificity

Reality – e.g. 2 miclassified examples
sensitivity $= 1$, specificity $= 1/2$



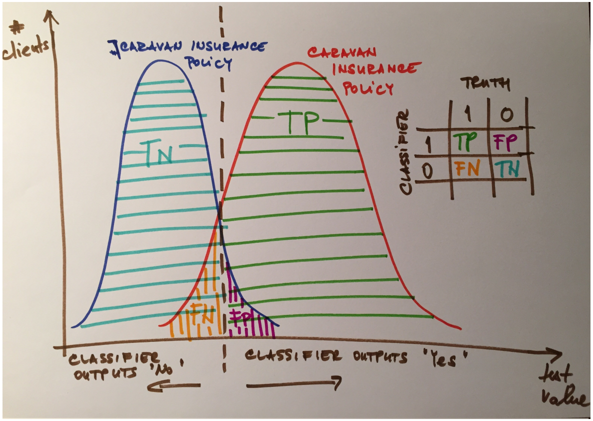
Evaluation of binary classifiers Sensitivity vs. specificity

Reality – e.g. 1 miclassified example
sensitivity $= 2/3$, specificity $= 1$



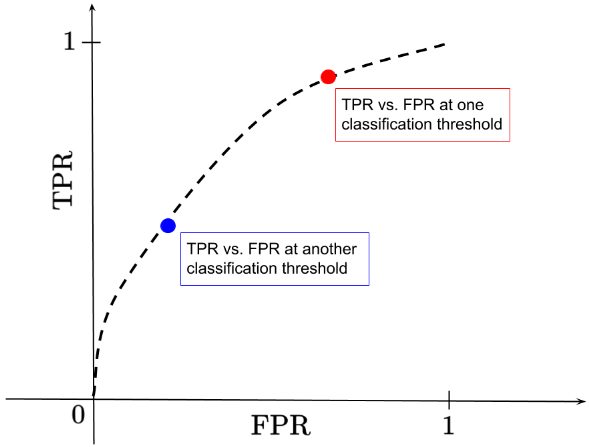
Evaluation of binary classifiers
Sensitivity vs. specificity

Sensitivity (TPR) vs. specificity (TNR)
– as the sensitivity increases, the specificity decreases and vice versa



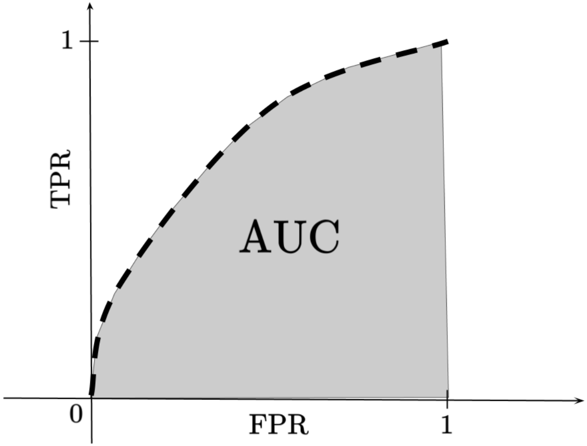
Evaluation of binary classifiers
ROC curve

An **ROC curve** plots True Positive Rate vs. False Positive Rate at different classification thresholds where FPR = 1 - TNR = FP/N = FP/(FP+TN)



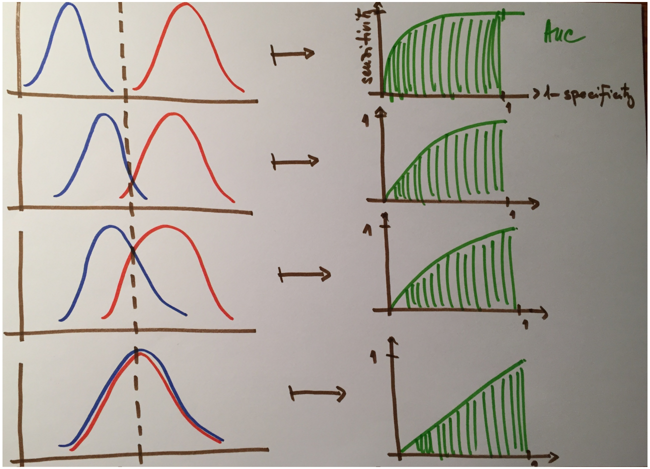
Evaluation of binary classifiers
AUC measure

Area Under ROC (= AUC)
is a measure of how good is a distinguishing property of classifier



Evaluation of binary classifiers
ROC & AUC

Curves closer to the top-left corner indicate a better performance.



Model complexity

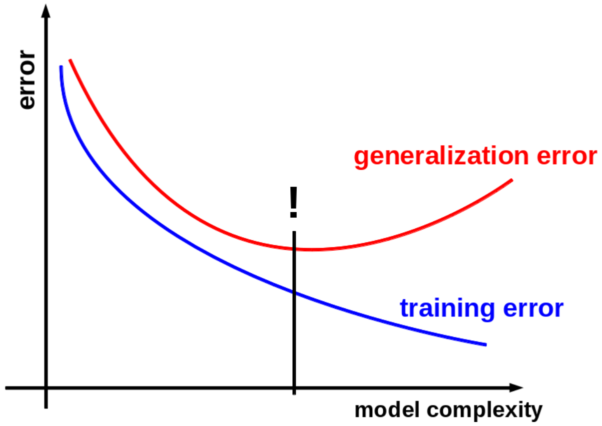
No universal definition

Heading for the regularization . . . **model complexity** is the number of hypothesis parameters

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

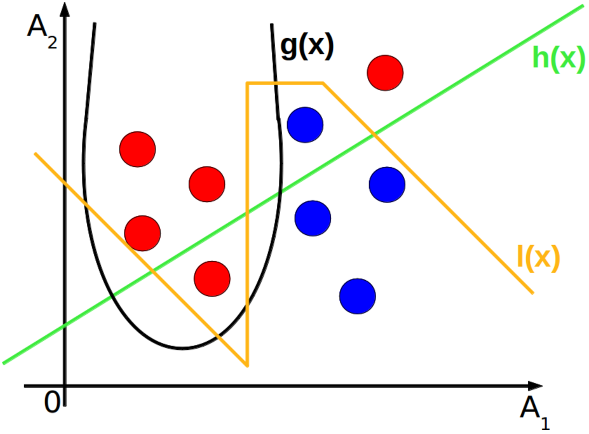
Model complexity

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



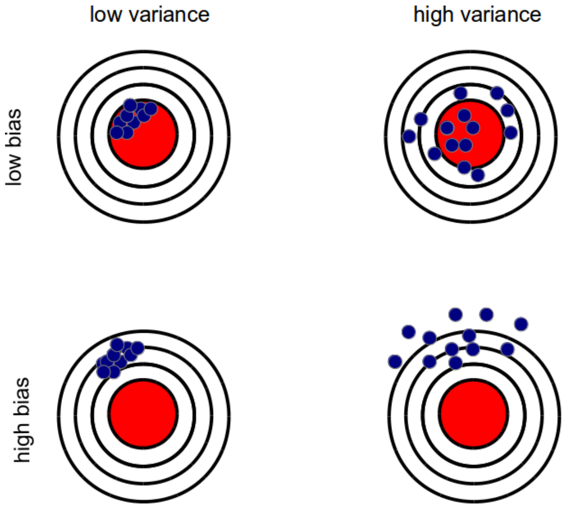
Bias and variance

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

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

Decomposition of $error_{\mathcal{D}}(\hat{f})$

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

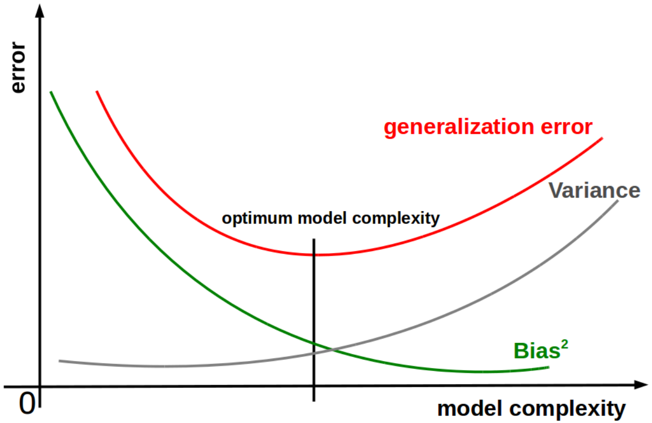
i.e.,

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

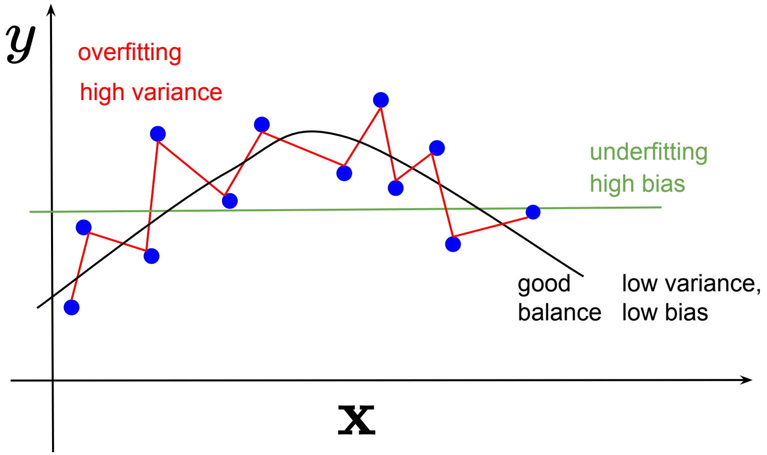
where $\hat{f}(\mathbf{x})$ is a predicted value, $E[\hat{f}(\mathbf{x})]$ is average predicted value

Bias and variance

- underfitting = high bias
- overfitting = high variance

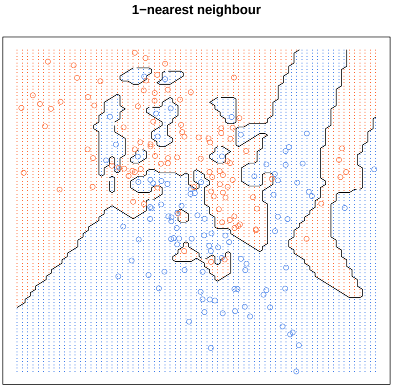


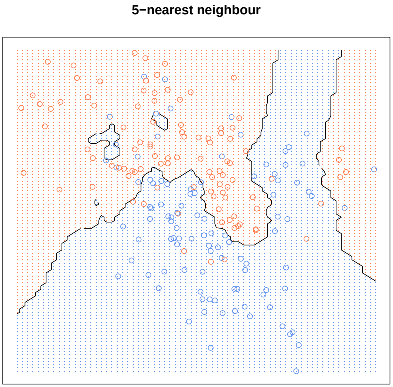
Bias and variance



Bias and variance
k-Nearest Neighbor

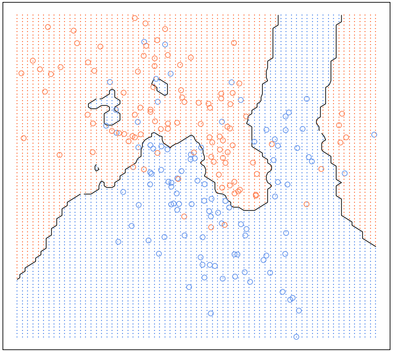
- $\uparrow k \rightarrow$ smoother decision boundary $\rightarrow \downarrow$ variance and $\uparrow$ bias
- $\downarrow k \rightarrow \uparrow$ variance and $\downarrow$ bias



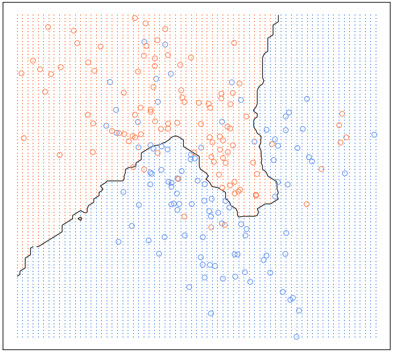


Bias and variance
k-Nearest Neighbor

5−nearest neighbour



15−nearest neighbour



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 $Data = \{\langle \mathbf{x}_i, y_i \rangle, \mathbf{x}_i = \langle x_{1i}, \dots, x_{mi} \rangle, y_i \in Y\}$

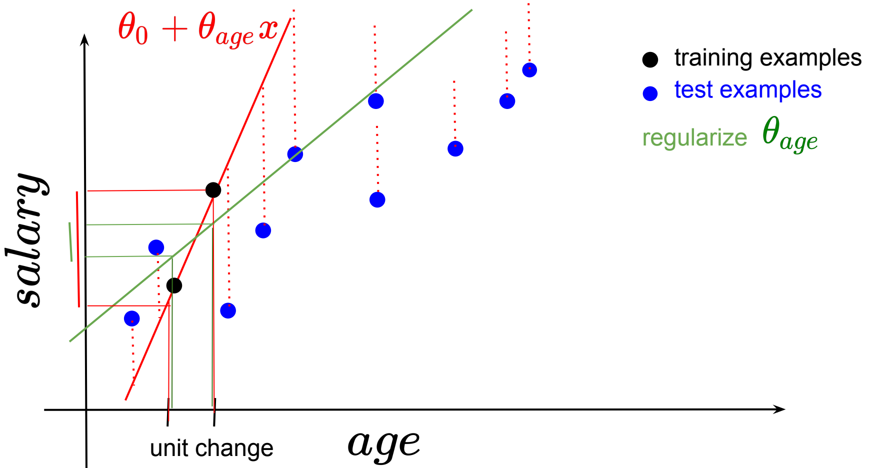
$$\Theta^\star = \mathrm{argmin}_\Theta \mathrm{L}(\Theta)$$

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
Motivation

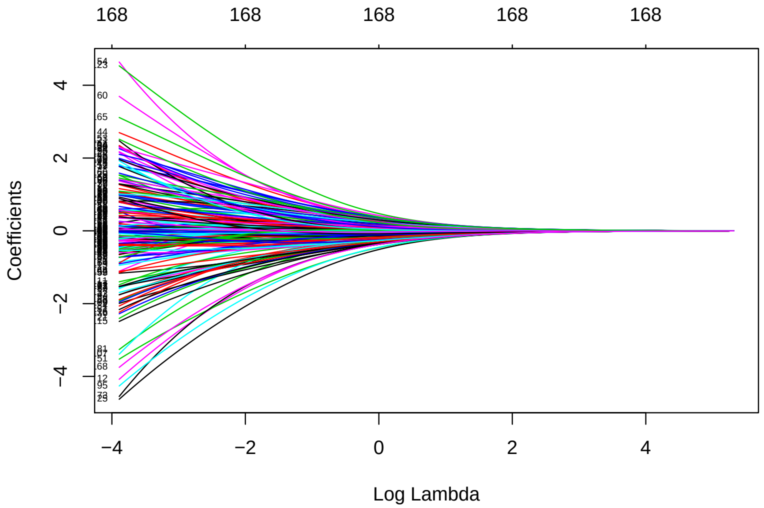


Regularization
Ridge regression

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

- 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

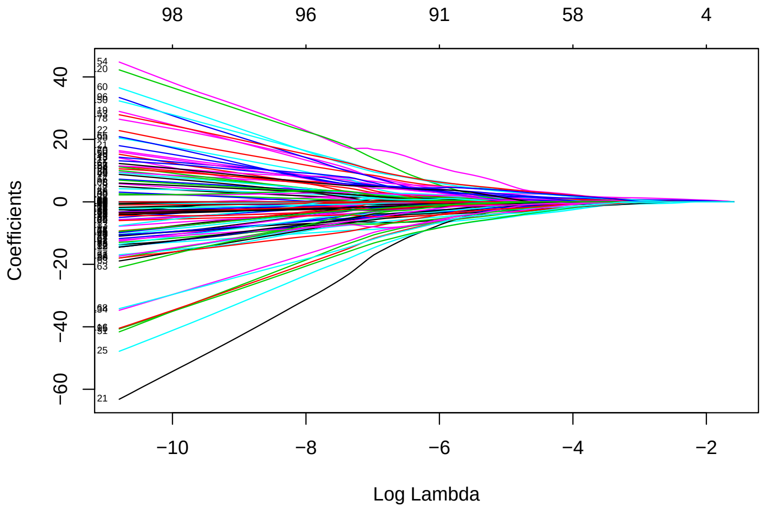


Regularization
Lasso

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

- 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 $i = 1, \ldots, m$

Lasso



98
96
91
58
4
Coefficients
Log Lambda
40
20
0
−20
−40
−60
−10
−8
−6
−4
−2
54
20
60
96
30
19
78
22
21
63
25
21

Ridge regression and Lasso

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

Elastic net

$$\Theta_R^\star = \mathrm{argmin}_\Theta[\mathrm{L}(\Theta) + \lambda_1 \cdot (|\theta_1| \ + \cdots + |\theta_m|) + \lambda_2 \cdot (\theta_1^2 + \cdots + \theta_m^2)]$$

$0 \leq \lambda_1, \lambda_2$ are tuning parameters

!!! In `glmnet` package

$$\Theta_R^\star = \mathrm{argmin}_\Theta \mathrm{L}(\Theta) + \lambda(\alpha(|\theta_1| \ + \cdots + |\theta_m|) + (1-\alpha)(\theta_1^2 + \cdots + \theta_m^2))$$

$$0 \leq \alpha \leq 1$$

Regularized linear regression

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

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

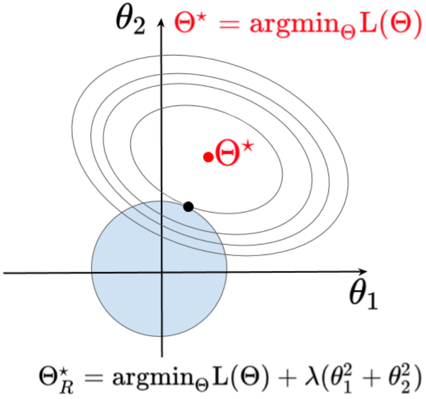
$$\Theta_R^\star = \mathrm{argmin}_\Theta [RSS + \lambda \cdot \text{penalty}(\Theta)]$$

Ridge regression
Alternative formulation

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

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

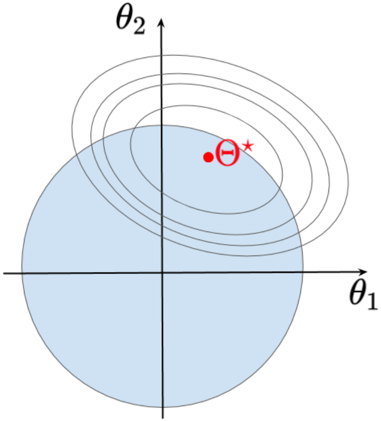
- the gray circle represents the feasible region for Ridge regression
- the contours represent different RSS values for the unregularized model



θ_2
$\Theta^\star = \text{argmin}_\Theta \text{L}(\Theta)$
$\Theta^\star$
θ_1
$\Theta^\star_R = \text{argmin}_\Theta \text{L}(\Theta) + \lambda(\theta_1^2 + \theta_2^2)$

Ridge regression
Alternative formulation

- If s is large enough, i.e. $\lambda = 0$, so that the minimum RSS value falls into the region of **ridge regression** parameter estimates then the alternative formulation yields the least square estimates.

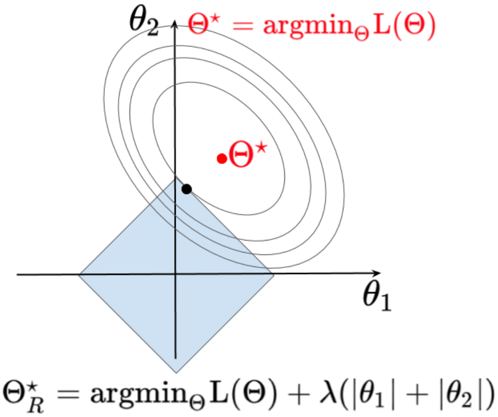


Lasso
Alternative formulation

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

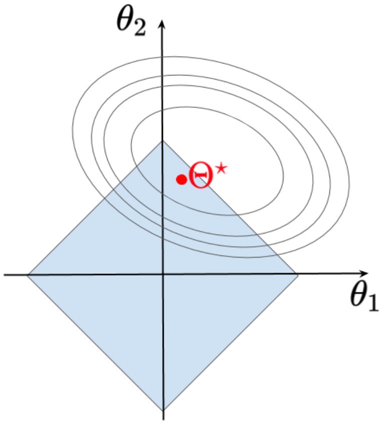
subject to $|\theta_1| + \cdots + |\theta_m| \leq s$

- the grey square represents the feasible region of the Lasso
- the contours represent different RSS values for the unregularized model

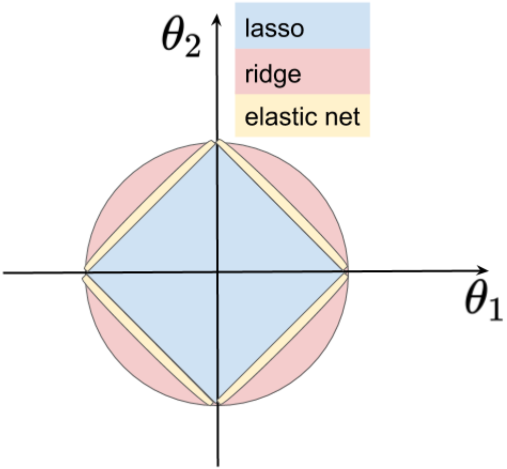

θ_2
$\Theta^\star = \operatorname{argmin}_\Theta \mathrm{L}(\Theta)$
$\Theta^\star$
θ_1
$\Theta^\star_R = \operatorname{argmin}_\Theta \mathrm{L}(\Theta) + \lambda(|\theta_1| + |\theta_2|)$

Lasso
Alternative formulation

- If *s* is large enough, i.e. $\lambda = 0$, so that the minimum RSS value falls into the region of **loss** parameter estimates then the alternative formulation yields the primary solution.



Elastic net



Regularized logistic regression

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

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

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

Summary of Examination Requirements

- Binary classifier using ROC curve (True Positive Rate vs. False Positive Rate)
- Model complexity, generalization error, Bias and variance
- Lasso and Ridge regularization for linear and logistic regression