Tomáš Horváth

BUSINESS ANALYTICS

Lecture 1

Machine Learning - Basics

Information Systems and Machine Learning Lab

University of Hildesheim

Germany

Overview

The goal of this lecture is to apprising us with the main concepts in Machine Learning.

More concretely, these are the following:

- Instances, labels and models
- Classification vs. regression
- Train set and test set
- k-NN: an instance-based or lazy learning technique
- Some quality indicators of a model
- Model and parameter selection

Instances and labels

- **Instances** are represented by their **attributes**

$$\mathbf{x} = (x_1, \ldots, x_k) \in \mathcal{X}, \quad \mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$$

- An instance belongs to a **class** or have a **value**. An instance a class or a value of which is known is called **labeled**

$$(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{L}$$

- Assume that labels are assigned according to some **unknown pattern** called labeling function

$$l : \mathcal{X} \to \mathcal{L}, \quad l(\mathbf{x}) = y$$

  - if $\mathcal{L} \subset \mathbb{Z}$[1] then l is a **classification** function (classifier)
  - if $\mathcal{L} \subset \mathbb{R}$ then l is a **regression** function (regressor)

[1] Important is, that we deal with discrete labels in case of classification.

Example: Instances and labeling functions

Flats $\mathbf{x} = (x_1, x_2) \in \mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ represented by two attributes:

- *area*, with domain $\mathcal{X}_1 = [25, 100]$
- *distance* from the city centre, with domain $\mathcal{X}_2 = [200, 2000]$

Assume that two labels are "hypothetically" generated by the following labeling functions:

- $price : \mathcal{X} \to \mathbb{R}$

$$price(\mathbf{x}) = \left\lfloor \left(\frac{x_1 - 25}{75} - \frac{x_2 - 200}{1800} + 1 \right) \cdot 300 \right\rfloor + 150$$

- $rentFor : \mathcal{X} \to \{students, families\}$

$$rentFor(\mathbf{x}) = \begin{cases} families, & \text{if } x_1 < 60 \ \& \ x_2 < 1000 \\ & \text{or } x_1 > 60 \ \& \ x_2 > 1000 \\ students, & \text{else} \end{cases}$$

Train set and Modeling

We have a problem: The labeling function l is unknown.

- Good news: Even if l is not known, we have observed a sample of instances with their labels. Such a set of instances is called the **training sample**

$$\mathcal{S}^{tr} = \{(\mathbf{x}, y) | \mathbf{x} \in \mathcal{X}, y \in \mathcal{L}\}$$

  which can be considered as an explicit definition of l.

The solution for the problem: Try somehow, using $\mathcal{S}^{tr}$, to **model** l by a mapping

$$m : \mathcal{X} \to \mathcal{L}, \quad m(\mathbf{x}) = \hat{y}$$

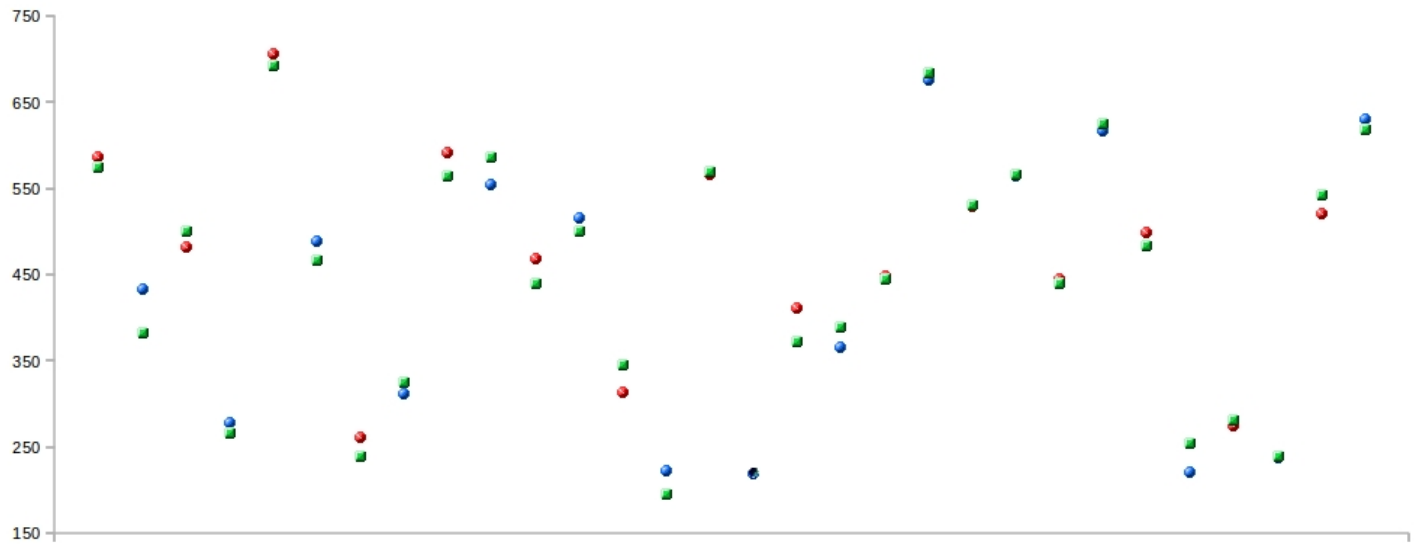
such that m is as close to l as possible.

- Bad news: usually $\mathcal{S}^{tr}$ covers just a small part of the domain of l and is noisy.

Example: Two training samples ($\mathcal{S}^{tr_1}$ and $\mathcal{S}^{tr_2}$)

Instances			Labels	
id	size	distance	price	rentFor
1	87	935	587	S
2	76	1124	482	F
3	92	349	706	S
4	41	1845	261	S
5	64	450	592	*F*
6	72	1383	469	F
7	46	1323	314	S
8	71	589	567	S
9	49	1242	411	S
10	32	400	449	F
11	87	1202	530	F
12	55	984	444	F
13	58	787	499	F
14	32	1376	275	S
15	96	1346	520	F

Instances			Labels	
id	size	distance	price	rentFor
16	37	889	433	F
17	51	1926	278	S
18	58	892	489	F
19	62	1827	312	F
20	66	361	554	*F*
21	62	787	515	*F*
22	29	1823	223	S
23	39	1915	219	S
24	64	1492	365	*S*
25	97	521	675	*F*
26	61	365	564	S
27	70	226	617	S
28	32	1533	221	*F*
29	35	1701	237	S
30	90	745	630	*F*



The k-NN model

The basic idea: **similar** instances have similar labels.

- distance function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
- k **nearest neighbors** to an instance $\mathbf{x}$ in $\mathcal{S}^{tr}$ w.r.t. d

$$\mathcal{N}_{\mathbf{x}}^{k,\mathcal{S}^{tr}} = \arg\min_{\mathcal{U}} \sum_{\substack{(\mathbf{u},y)\in\mathcal{U} \\ \mathcal{U}\subseteq\mathcal{S}^{tr},|\mathcal{U}|=k}} d(\mathbf{x},\mathbf{u})$$

The k-NN model of l (explicitly) defined by $\mathcal{S}^{tr}$

- classifier

$$m^{k,\mathcal{S}^{tr}}(\mathbf{x}) = arg\,\underset{y}{max}\,|\,\{(\mathbf{u},y') \in \mathcal{N}_{\mathbf{x}}^{k,\mathcal{S}^{tr}} \;:\; y' = y\}\,|$$

- regressor

$$m^{k,\mathcal{S}^{tr}}(\mathbf{x}) = \frac{1}{k} \sum_{(\mathbf{u},y)\in\mathcal{N}_{\mathbf{x}}^{k,\mathcal{S}^{tr}}} y$$

Example: k-NN

Settings

- training sample $\mathcal{S}^{tr_1}$
- Euclidean distance

Instances			Labels	
id	size	distance	price	rentFor
1	87	935	587	S
2	76	1124	482	F
3	92	349	706	S
4	41	1845	261	S
5	64	450	592	*F*
6	72	1383	469	F
7	46	1323	314	S
8	71	589	567	S
9	49	1242	411	S
10	32	400	449	F
11	87	1202	530	F
12	55	984	444	F
13	58	787	499	F
14	32	1376	275	S
15	96	1346	520	F

Which price would be assigned to the following instances?

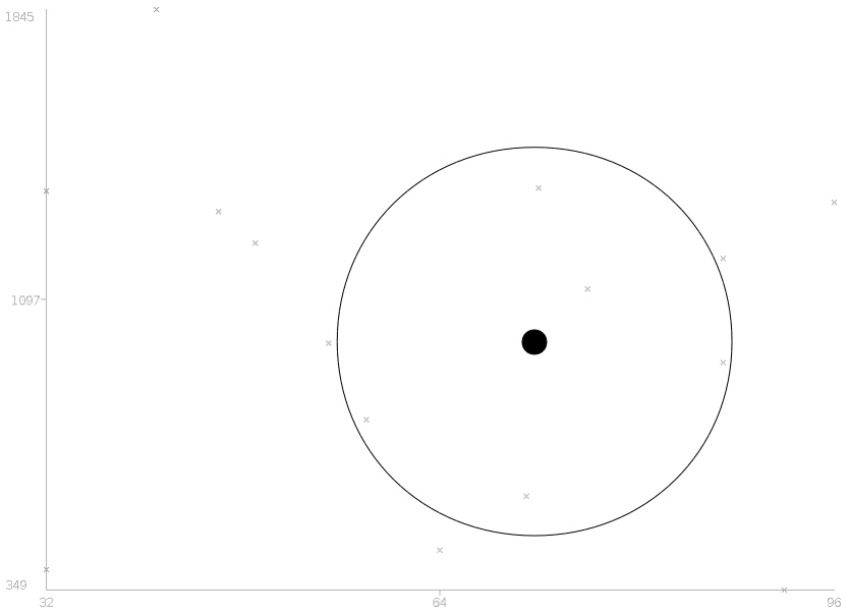
- for $k = \{1, 2, \ldots, 15\}$

Which label of rentFor (F or S) would be assigned to the following instances?

- for $k = \{1, 3, 5, 7, 9, 11, 13, 15\}$

Instances			Labels	
id	size	distance	price	rentFor
100	72	1770	?	?
200	51	490	?	?
300	96	960	?	?

Example: k-NN



$$m^{3,\mathcal{S}^{tr_1}}_{rentFor}(\mathbf{x}_{100}) = S$$

$$m^{5,\mathcal{S}^{tr_1}}_{price}(\mathbf{x}_{100}) = 452$$

$$\ldots$$

k	rentFor for instance		
	100	200	300
1	*S*	F	*F*
3	*S*	*S*	*F*
5	*S*	F	*F*
7	*S*	F	*F*
9	F	F	*F*
11	F	F	*F*
13	F	F	*F*
15	F	F	*F*

k	price for instance		
	100	200	300
1	469	592	587
2	476	546	559
3	494	513	546
4	500	527	530
5	452	510	565
6	446	494	565
7	427	468	552
8	429	470	545
9	446	483	550
10	429	505	540
11	436	484	528
12	447	483	510
13	458	487	492
14	476	489	476
15	474	474	474

The quality of a model

We have many (e.g. 15 in case of $m_{price}^{k,\mathcal{S}^{tr_1}}$)[2] **different models**.

- Which model is the best one?
- Which properties a good model should have?
  - We need some quality indicators for a model...

One model could be trained using many **different training samples**.

- What would the results be in case of using $\mathcal{S}^{tr_2}$ or any other training sample instead of $\mathcal{S}^{tr_1}$?

[2] We will mostly deal only with the *price* labels from now on.

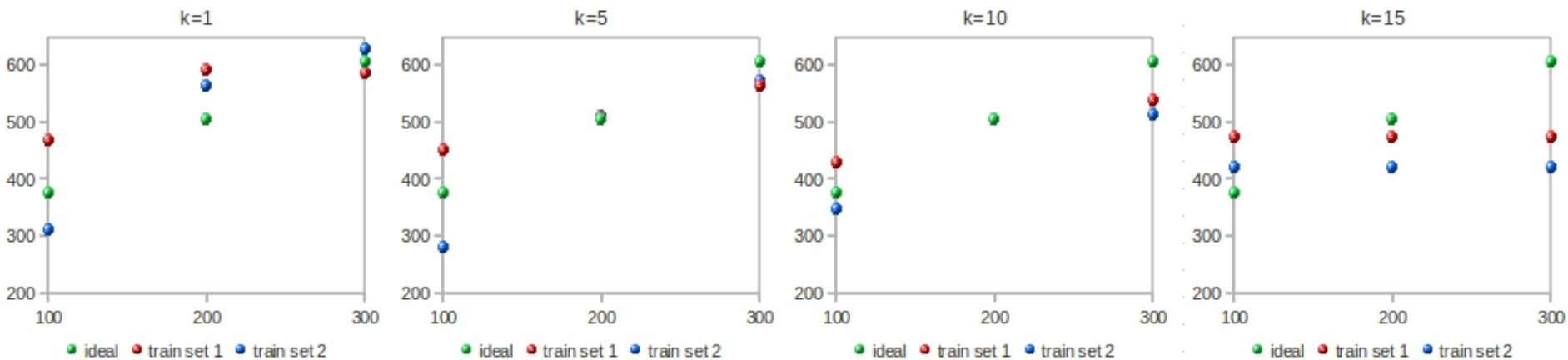
Bias and Variance

Bias

- measures, how $m^{\theta,\mathcal{S}^{tr_1}}, m^{\theta,\mathcal{S}^{tr_2}}, \ldots, m^{\theta,\mathcal{S}^{tr_m}}$ differs from l
- determines, how generic the model[1] m^{θ} is

Variance

- measures, how $m^{\theta,\mathcal{S}^{tr_1}}, m^{\theta,\mathcal{S}^{tr_2}}, \ldots, m^{\theta,\mathcal{S}^{tr_m}}$ differs frome each other
- determines, how stable the model m^{θ} is



[1] θ stands for the parameters of the model (e.g. k for k-NN)

Underfitting vs. Overfitting

Bias

$$bias^2_{m^\theta}(\mathbf{x}) = (\, l(\mathbf{x}) - \mathrm{E}_{\mathcal{S}^{tr}}\{m^{\theta,\mathcal{S}^{tr}}(\mathbf{x})\}\,)^2$$

Variance

$$variance_{m^\theta}(\mathbf{x}) = \mathrm{E}_{\mathcal{S}^{tr}}\{\,(\, m^{\theta,\mathcal{S}^{tr}}(\mathbf{x}) - \mathrm{E}_{\mathcal{S}^{tr}}\{m^{\theta,\mathcal{S}^{tr}}(\mathbf{x})\}\,)^2\}$$

$\mathrm{E}_{\mathcal{S}^{tr}}\{\, X \,\}$ is an **expected value** of X over all training samples.

Underfitting

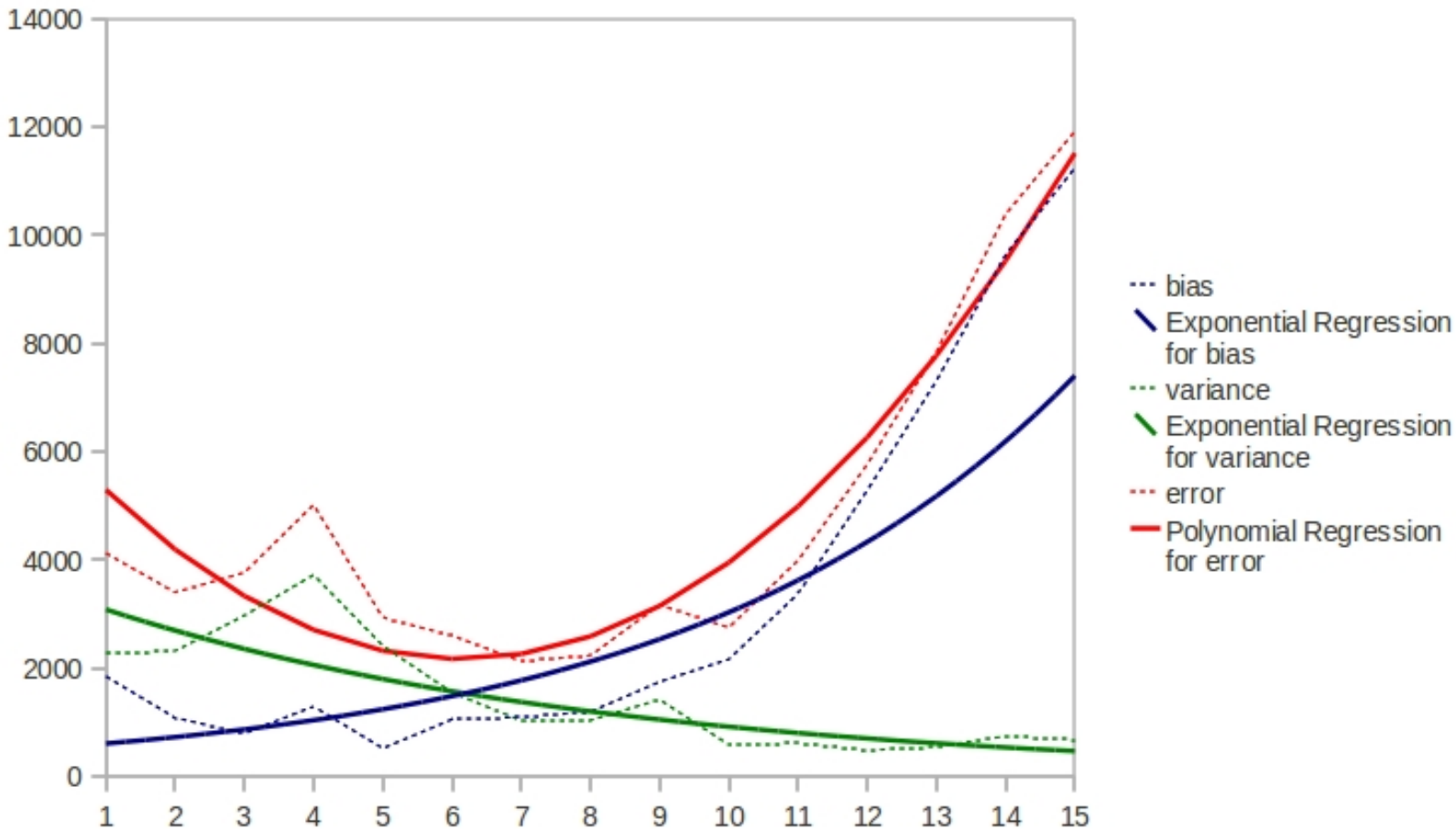
- when the model has high bias and low variance, i.e. is too general

Overfitting

- when the model has low bias and high variance, i.e. is too specific

The bias-variance tradeoff

Usually, the bias decreases with the **complexity** of the model, while variance increases with the complexity of the model. Thus, we need to find a tradeoff model, which is not too general nor too specific.



Error

What happens if we sum up the bias and the variance?[1]

$$
\begin{aligned}
& bias^2_{m^\theta}(\mathbf{x}) + variance_{m^\theta}(\mathbf{x}) = \\
&= (l(\mathbf{x}) - \mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\})^2 + \mathrm{E}_{\mathcal{S}^{tr}}\{(\hat{y} - \mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\})^2\} \\
&= (l(\mathbf{x}) - \mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\})^2 + \mathrm{E}_{\mathcal{S}^{tr}}\{(\hat{y} - \mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\})^2\} \\
&\quad + 2 \cdot (l(\mathbf{x}) - \mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\})(\mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\} - \mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\}) \\
&= (l(\mathbf{x}) - \mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\})^2 + \mathrm{E}_{\mathcal{S}^{tr}}\{(\hat{y} - \mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\})^2\} \\
&\quad + 2 \cdot (l(\mathbf{x}) - \mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\})\mathrm{E}_{\mathcal{S}^{tr}}\{(\mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\} - \hat{y})\} \\
&= \mathrm{E}_{\mathcal{S}^{tr}}\{(l(\mathbf{x}) - \mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\})^2\} + \mathrm{E}_{\mathcal{S}^{tr}}\{(\mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\} - \hat{y})^2\}\} \\
&\quad + \mathrm{E}_{\mathcal{S}^{tr}}\{2 \cdot (l(\mathbf{x}) - \mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\})(\mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\} - \hat{y})\} \\
&= \mathrm{E}_{\mathcal{S}^{tr}}\{(l(\mathbf{x}) - \mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\} + \mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\} - \hat{y})^2\}\} \\
&= \mathrm{E}_{\mathcal{S}^{tr}}\{(l(\mathbf{x}) - \hat{y})^2\}
\end{aligned}
$$

We get the expected squared error of the model over all training samples w.r.t. the labeling.

[1] We will denote $m^{\theta,\mathcal{S}^{tr}}(\mathbf{x})$ as $\hat{y}$ for better readability on the next slides.

Noise in sampling

The error introduced on the previous slide deals with the labeling l.

- However, the precise values of l are unknown.
  - We should consider to use the observed labels from the training sample.

As we have seen, observations are usually noisy, i.e. $y = l(\mathbf{x}) + \epsilon$ for all $(\mathbf{x}, y) \in \mathcal{S}^{tr}$, where $\mathcal{S}^{tr}$ is an arbitrary sample of instances.

- there can be more instances with same attribute values but different labels
- note, that we don't care about where the noise came from
  - non-perfect measuring devices, human factor, etc.

$$noise(\mathbf{x}) = \mathrm{E}_{(\mathbf{x},y)}\{ (y - l(\mathbf{x}))^2 \}$$

Noise in sampling

Usually, we assume a normally distributed sampling error $\epsilon \sim \mathcal{N}(0,1)$

- thus, $\mathrm{E}_{(\mathbf{x},y)}\{y\} = l(\mathbf{x})$

Let's rewrite the equations introduces before as

$$bias^2_{m^\theta}(\mathbf{x}) = (\ \mathrm{E}_{(\mathbf{x},y)}\{y\} - \mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\}\ )^2$$

$$variance_{m^\theta}(\mathbf{x}) = \mathrm{E}_{\mathcal{S}^{tr}}\{\ (\ \hat{y} - \mathrm{E}_{\mathcal{S}^{tr}}\{\hat{y}\}\ )^2\}$$

$$noise(\mathbf{x}) = \mathrm{E}_{(\mathbf{x},y)}\{\ (y - \mathrm{E}_{(\mathbf{x},y)}\{y\})^2\ \}$$

and sum them up

$$\underbrace{bias^2_{m^\theta}(\mathbf{x}) + variance_{m^\theta}(\mathbf{x})}_{\mathrm{E}_{\mathcal{S}^{tr}}\{(\mathrm{E}_{(\mathbf{x},y)}\{y\} - \hat{y})^2\}} + noise(\mathbf{x})$$

Expected squared error

$$
\begin{aligned}
&\mathrm{E}_{\mathcal{S}^{tr}}\{(\mathrm{E}_{(\mathbf{x},y)}\{y\} - \hat{y})^2\} + \mathrm{E}_{(\mathbf{x},y)}\{(y - \mathrm{E}_{(\mathbf{x},y)}\{y\})^2\} \\
&= \mathrm{E}_{(\mathbf{x},y)}\{(y - \mathrm{E}_{(\mathbf{x},y)}\{y\})^2\} + \mathrm{E}_{\mathcal{S}^{tr}}\{(\mathrm{E}_{(\mathbf{x},y)}\{y\} - \hat{y})^2\} \\
&\quad + \mathrm{E}_{\mathcal{S}^{tr}}\{2 \cdot (\mathrm{E}_{(\mathbf{x},y)}\{y\} - \mathrm{E}_{(\mathbf{x},y)}\{y\})(\mathrm{E}_{(\mathbf{x},y)}\{y\} - \hat{y})\} \\
&= \mathrm{E}_{\mathcal{S}^{tr}}\{\mathrm{E}_{(\mathbf{x},y)}\{(y - \mathrm{E}_{(\mathbf{x},y)}\{y\})^2\}\} + \mathrm{E}_{\mathcal{S}^{tr}}\{\mathrm{E}_{(\mathbf{x},y)}\{(\mathrm{E}_{(\mathbf{x},y)}\{y\} - \hat{y})^2\} \\
&\quad + \mathrm{E}_{\mathcal{S}^{tr}}\{\mathrm{E}_{(\mathbf{x},y)}\{2 \cdot (y - \mathrm{E}_{(\mathbf{x},y)}\{y\})(\mathrm{E}_{(\mathbf{x},y)}\{y\} - \hat{y})\}\} \\
&= \mathrm{E}_{\mathcal{S}^{tr}}\{\mathrm{E}_{(\mathbf{x},y)}\{(y - \mathrm{E}_{(\mathbf{x},y)}\{y\} + \mathrm{E}_{(\mathbf{x},y)}\{y\} - \hat{y})^2\}\} \\
&= \mathrm{E}_{\mathcal{S}^{tr}}\{\mathrm{E}_{(\mathbf{x},y)}\{(y - \hat{y})^2\}\}
\end{aligned}
$$

We get the expected squared error of the model over all training samples and all instances w.r.t. the observed labeling.

- known labels for observed instances

Test set, RMSE and MAE

In practice, we train a model m^θ on a train set $\mathcal{S}^{tr}$ and test its error on a so-called **test sample** $\mathcal{S}^{te}$ defined as

$$\mathcal{S}^{te} \subset \mathcal{X} \times \mathcal{L} \setminus \mathcal{S}^{tr}$$

Root mean squared error (regression)

$$rmse(m^{\theta,\mathcal{S}^{tr}}(\mathbf{x}), \mathcal{S}^{te}) = \sqrt{\frac{\sum_{(\mathbf{x},y)\in\mathcal{S}^{te}} (m^{\theta,\mathcal{S}^{tr}}(\mathbf{x}) - y)^2}{|\mathcal{S}^{te}|}}$$

Mean absolute error (classification)

$$mae(m^{\theta,\mathcal{S}^{tr}}(\mathbf{x}), \mathcal{S}^{te}) = \frac{\sum_{(\mathbf{x},y)\in\mathcal{S}^{te}} I(m^{\theta,\mathcal{S}^{tr}}(\mathbf{x}) \neq y)}{|\mathcal{S}^{te}|}$$

where $I(\cdot) = 1$ if the condition $(\cdot)$ holds, otherwise $I(\cdot) = 0$.

Cross-validation

A small complication: As usual, we have only one training and one test set on the input! Moreover, the labels of instances in the test set are "hidden"[1] to the model.

- Question: How can we get the model with the least expected error?
  - which means evaluating over all training samples and all instances...
- Answer: Try to simulate learning over "more" training sets and "more" instances.
  - which means creating more training sets (with lower sizes) from the original one...
  - this process is called cross-validation

[1] The test set should be usually used for the final evaluation of the model but not for tuning it (selection of a best technique or good parameters, etc.)

k-fold Cross-validation

One possible alternative:

1. Split the training sample $\mathcal{S}^{tr}$ to k parts of similar size

$$\mathcal{S}^{tr} = \bigcup_k \mathcal{S}^{tr}_k, \qquad \forall i \neq j: \quad -1 \leq |\mathcal{S}^{tr}_i| - |\mathcal{S}^{tr}_j| \leq 1$$

   - split can be made systematically or randomly (better)

2. choose the model m^{θ}_{best} such that

$$m^{\theta}_{best} = \underset{m^{\theta}}{arg\,min} \left\{ \frac{1}{k} \sum_{i=1}^{k} err(m^{\theta, \bigcup_{1 \leq j \leq k, j \neq i} \mathcal{S}^{tr}_j}, \mathcal{S}^{tr}_i) \right\}$$

   where $err(\cdot, \cdot)$ is an $rmse(\cdot, \cdot)$ or $mae(\cdot, \cdot)$ error measure.

   - $\mathcal{S}^{tr}_i$ is called **validation fold**.

Example: Cross-validation

$\mathcal{S}_1^{tr} = \{1, 6, 9, 12, 13\}, \quad \mathcal{S}_2^{tr} = \{4, 5, 7, 11, 15\}, \quad \mathcal{S}_3^{tr} = \{2, 3, 8, 10, 14\}$

- 3 randomly created folds from the first training sample introduced at the beginning (denoted as $\mathcal{S}^{tr_1}$)

m^θ	Folds $\mathcal{S}_1^{tr}$	$\mathcal{S}_2^{tr}$	$\mathcal{S}_2^{tr}$	$rmse(m^{\theta,train}, validation$	average $rmse$
m^1	validation	train		82.887	143.386
	train	validation	train	142.833	
	train		validation	204.439	
m^2	...				135.624
m^3	...				146.726
m^4	...				127.200
m^5	...				110.454
m^6	...				112.489
m^7	...				148.286
m^8	...				156.629
m^9	...				169.326
m^{10}	...				202.006

$m^\theta_{best} = m^5$

Example: The whole story

1. We have the training set $\mathcal{S}^{tr_1}$ and a following test set $\mathcal{S}^{te}$ on the input

Instances			Price	
id	size	distance	observed	l
100	72	1770	362	376
200	51	490	496	505
300	96	960	609	607

2. We decide to use a k-NN regression technique
3. Using 3-fold cross-validation on $\mathcal{S}^{tr_1}$ we find the best parameter $k = 5$
4. We test our regressor $m^{5,\mathcal{S}^{tr_1}}$ on the test as follows
   - regres the values of the test instances as: $m^{5,\mathcal{S}^{tr_1}}(100) = 452.4$, $m^{5,\mathcal{S}^{tr_1}}(200) = 510.2$, $m^{5,\mathcal{S}^{tr_1}}(300) = 565$
   - compute

$$rmse(m^{5,\mathcal{S}^{tr_1}}, \mathcal{S}^{te}) = \sqrt{\frac{(362-452.4)^2+(496-510.2)^2+(609-565)^2}{3}} = 58.623$$

Summary

- Instances, labels and models
  - Assuming that the labels of instances are generated according to some hidden pattern, the goal is a modeling of that pattern as accurately as possible.
- Classification vs. regression
  - Nominal labels vs. continuos labels... The choice of a good technique depends on the task.
- Train set and test set
  - The model is learned using the train set and finally tested using the test set. The labels of test instances are "hidden" during the learning.
- k-NN: an instance-based or lazy learning technique
  - A quite popular ML technique with many variations and heuristics used. The larger the training sample the more precise the model is, however, with increasing time consumption. Also, some problems occur in case of high-dimensional spaces.

Summary

- Some quality indicators of a model
  - The bias/variance determines how generic/stable a model is. Usually, the bias decreases while the variance increases with the complexity of the model and we need to find a trade-off.
  - The error of a model depends on the bias and the variance, and, on the noise in sampling. the most popular error measures for regression and classification are RMSE and MAE, respectively. However, other error measures can be found in the literature.
- Model and parameter selection
  - Cross-validation is a popular model and parameter selection procedure. Depending on the characteristics of the training sample, some modifications of this approach, as the "leave-one-out" or the "hold-out", can be used, too. Also, there are other techniques for this issue.

Recommended reading

In ascending order of difficulty:

- Pang-Ning Tan, Michael Steinbach and Vipin Kumar: Introduction to Data Mining. Addison-Wesley, 2006, ISBN-13: 978-0-321-32136-7, 769pp.
- Christopher M. Bishop: Pattern Recognition and Machine Learning. Springer-Verlag, 2006, ISBN: 978-0-387-31073-2, 740 pp.
- Trevor Hastie, Robert Tibshirani and Jerome Friedman: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer-Verlag, 2009, ISBN 978-0-387-84857-0, 746pp.

And also many other, good textbooks and web sites...

Thanks for Your attention!

Questions?

horvath@ismll.de