BLOCK Data Engineering - Unit 3 on Feature Selection

Myra Spiliopoulou [1]

[1] Faculty of Computer Science, Otto-von-Guericke University Magdeburg

Literature of the unit

BOOK: Salvador Garcia, Julian Luengo, Francisco Herrera (2015) *Data Preprocessing in Data Mining*, SPRINGER International Publishing Switzerland

2 Chapter 3 'Data Preparation Basic Models'
1 Chapter 4 'Dealing with Missing Values'
3 **Chapter 7 'Feature Selection'**

and some words on

3.2.1 Data Integration –'Finding Redundant Attributes' → [Unit 3]
3.5 Data Transformation → [Unit 3]

RECALL: Running example on patient response to treatment

Id	I2	I15	I30	I22	I31	I9	I26	Response
#1	f	VH	yes	better	no	r	no	yes
#2	m	M	no	better	no	b	yes	no
#3	m	M	no	worse	no	b	no	no
#4	f	VH	yes	worse	no	b	no	yes
#5	m	L	no	no effect	no	l	no	no
#6	m	M	no	better	no	l	no	no
#7	f	??	yes	better	yes	l	yes	yes
#8	f	H	no	better	yes	r	no	no
#9	f	H	yes	better	no	l	no	yes
#10	m	M	yes	worse	no	b	no	no
#11	m	M	no	no effect	no	l	no	no
#12	f	H	no	no effect	no	r	no	no
#13	??	??	yes	??	no	l	yes	yes
#14	m	M	no	worse	no	b	no	no
#15	m	L	no	no effect	no	l	yes	no

We use the data to train models and acquire insights. Therefore we must:

- $\surd$ deal with missing values,
- $\surd$ clean away the errors in the data,
- $\surd$ normalize the data, remove duplicates and
- $\rightarrow$ eliminate redundant variables, because
  learning algorithms do not cope well with high-dimensional spaces and because the more variables we seek to record, the more likely it is that we will have missing values in them.

The real counterpart of our tiny dataset [Schleicher et al., 2024]

- *Original dataset from Charité Universitätsmedizin Berlin:* patients with tinnitus duration of at least 3 months, an age of at least 18 years and sufficient knowledge of the German language, in the period January 2011 until October 2015, treated [. . .], exclusion criteria [. . .] N=3971
- *Further exclusion criteria:* treatment finished, no intermediate visits, only one treatment sequence, no longer than 15 days N=1450
- *Constraint:* no missing values in the 9 questionnaires of the study N=1287
- *Random sample of 500 patients:* N=500 (f:240, m: 260)

Table 1: lists the questionnaires and the number of items in each one

Feature Selection: What and Why

Definition of FSel [Section 7.1]

'*Feature selection* is a process that chooses the optimal subset of features according to a certain criterion.'

Feature Selection: What and Why

Definition of FSel [Section 7.1]

'*Feature selection* is a process that chooses the optimal subset of features according to a certain criterion.'

Optimality / Goodness $\Rightarrow$ **Purpose** of the feature selection process [a]

[a] From Section 4.1 and from citation [48]:Sayes Y., Inza I, Larranaga P (2007) A review of feature selection techniques in bioinformatics. *Bioinformatics* 23(19), 2507-2517

- ⋆ discard redundant features
- ⋆ reduce the cost of data acquisition

Feature Selection: What and Why

Definition of FSel [Section 7.1]

'*Feature selection* is a process that chooses the optimal subset of features according to a certain criterion.'

Optimality / Goodness ⇒ **Purpose** of the feature selection process [a]

[a] From Section 4.1 and from citation [48]:Sayes Y., Inza I, Larranaga P (2007) A review of feature selection techniques in bioinformatics. *Bioinformatics* 23(19), 2507-2517

- ⋆ discard redundant features
- ⋆ reduce the cost of data acquisition

and when applying FSel in preparation of an already known learning task:

- ○ discard irrelevant features, and thus also irrelevant data
- ○ increase the accuracy/quality of the learned models
- ○ reduce the complexity of the model / of the model description
- ○ improve efficiency of the learning process, e.g. by reducing storage requirements and computational costs

Feature Selection Tasks in our Example

From Table 1 in [Schleicher et al., 2024]:

Pos	Name	# Items	Outcome of interest
1	SOZK	25	-
2	TQ	52	YES
3	PSQ	30	YES
4	SF8	8	–
5	SWOP	9	–
6	TLQ	8	–
7	BSF	30	–
8	BI	56	–
9	ADSL	20	YES
		238	YES: 102

FSel tasks with known outcomes of interest:

- ⋆ remove redundant features
  ⇒ reduce cost / burden
- ○ remove irrelevant features
  ⇒ reduce cost / burden
- ○ reduce model complexity
  ⇒ make it more actionable for DS

Feature Selection Tasks in our Example

From Table 1 in [Schleicher et al., 2024]:

Pos	Name	# Items	Outcome of interest
1	SOZK	25	-
2	TQ	52	YES
3	PSQ	30	YES
4	SF8	8	–
5	SWOP	9	–
6	TLQ	8	–
7	BSF	30	–
8	BI	56	–
9	ADSL	20	YES
		238	YES: 102

FSel tasks with known outcomes of interest:

- ⋆ remove redundant features
  ⇒ reduce cost / burden
- ○ remove irrelevant features
  ⇒ reduce cost / burden
- ○ reduce model complexity
  ⇒ make it more actionable for DS

except that we cannot remove individual features, only complete questionnaires.

The Quest for the Good Features (Section 7.2.1.1, with modifications)

Given a feature space F and a goodness measure U: Build the subset of the good features S from scratch or rather by discarding features from F ?

The Quest for the Good Features (Section 7.2.1.1, with modifications)

Given a feature space F and a goodness measure U: Build the subset of the good features S from scratch or rather by discarding features from F ?

Sequential forward and backward feature set generation algorithms

SFG	Sequential forward feature set generation
function	$SFG(F,U)$
initialize	$S = \emptyset$
repeat	
	$f = FindNextBest(F,U)$
	$S = S \cup \{f\}$
	$F = F \setminus \{f\}$
until	S satisfies U or $F = \emptyset$
return	S
end function	

SBG	Sequential backward feature set generation
function	$SBG(F,U)$
initialize	$S = \emptyset$
repeat	
	$f = FindNextWorst(F,U)$
	$S = S \cup \{f\}$
	$F = F \setminus \{f\}$
until	S does not satisfy U or $F = \emptyset$
return	$F \cup \{f\}$ `// add the last feature again`
end function	

The Quest for the Good Features (Section 7.2.1.1, with modifications)

Given a feature space F and a goodness measure U: Build the subset of the good features S from scratch or rather by discarding features from F ?

Sequential forward and backward feature set generation algorithms

SFG	Sequential forward feature set generation
function	$SFG(F,U)$
initialize	$S = \emptyset$
repeat	
	$f = FindNextBest(F,U)$
	$S = S \cup \{f\}$
	$F = F \setminus \{f\}$
until	S satisfies U or $F = \emptyset$
return	S
end function	

SBG	Sequential backward feature set generation
function	$SBG(F,U)$
initialize	$S = \emptyset$
repeat	
	$f = FindNextWorst(F,U)$
	$S = S \cup \{f\}$
	$F = F \setminus \{f\}$
until	S does not satisfy U or $F = \emptyset$
return	$F \cup \{f\}$ `// add the last feature again`
end function	

where the $FindNextBest()$ and $FindNextWorst()$ functions operate on the feature set as it is modified in each iteration (rather than the original one). Hence, each newly selected feature f is best/worst given the previous selections.

The Quest cntd

(Algorithms 3 & 4, with modifications)

Combining SFG and SBG into: Bidirectional feature set generation algorithm

BG	Bidirectional feature set generation
function	$BG(F, U)$
initialize	$S_g = \emptyset$; $S_b = \emptyset$; $F_g = F$; $F_b = F$
repeat	
	$f_g = FindNextBest(F_g, U)$; $S_g = S_g \cup \{f_g\}$; $F_g = F_g \setminus \{f_g\}$
	$f_b = FindNextWorst(F_b, U)$; $S_b = S_b \cup \{f_b\}$; $F_b = F_b \setminus \{f_b\}$
until	$(a) S_f$ satisfies U or $F_f = \emptyset$ or (b) S_b does not satisfy U or $F_b = \emptyset$
if	(a) holds *then return* S_f *else return* $F_b \cup \{f_b\}$ *endif*
end function	

The Quest cntd

(Algorithms 3 & 4, with modifications)

Combining SFG and SBG into: Bidirectional feature set generation algorithm

BG	Bidirectional feature set generation
function	$BG(F, U)$
initialize	$S_g = \emptyset$; $S_b = \emptyset$; $F_g = F$; $F_b = F$
repeat	
	$f_g = FindNextBest(F_g, U)$; $S_g = S_g \cup \{f_g\}$; $F_g = F_g \setminus \{f_g\}$
	$f_b = FindNextWorst(F_b, U)$; $S_b = S_b \cup \{f_b\}$; $F_b = F_b \setminus \{f_b\}$
until	$(a) S_f$ satisfies U or $F_f = \emptyset$ or (b) S_b does not satisfy U or $F_b = \emptyset$
if	(a) holds *then return* S_f *else return* $F_b \cup \{f_b\}$ *endif*
end function	

Generating *subsets* of F at random:

RG	Random feature set generation
function	$RG(F, U)$
initialize	$S = S_{best} \emptyset$; $c_{best} = \|F\|$
repeat	
	$S =$ *RandGen*(F) `// pick features from` F `at random`
	if $\|S\| \leq c_{best}$ and S satisfies U *then* $S_{best} = S$; $c_{best} = \|S\|$ *endif*
until	some stopping criterion is satisfied
return	S_{best}
end function	

How long to search?

(Section 7.2.1.2, with modifications)

- *Exhaustive search:*
  Generate all non-empty subsets of F, i.e. $\mathscr{P}(F) \setminus \{\emptyset\}$, and choose the best one
  - ok if F is small
  - nok if F is large $\Rightarrow$ set a threshold m as the minimum number of features to be selected or removed
- *Heuristic search:*
  Traverse the search space of solutions by adhering to some heuristic, which also incorporates a termination criterion
- *Non-deterministic search:*
  Generate subsets of F at random (cf. Algorithm RG) without an explicit termination criterion $\Rightarrow$ at each time point, RG() returns the best subset found thus far [1]

[1] In 'stream mining', best effort algorithms of this kind are called *anytime algorithms*.

How long to search?

(Section 7.2.1.2, with modifications)

- *Exhaustive search:*
  Generate all non-empty subsets of F, i.e. $\mathscr{P}(F) \setminus \{\emptyset\}$, and choose the best one
  - ok if F is small
  - nok if F is large $\Rightarrow$ set a threshold m as the minimum number of features to be selected or removed
- *Heuristic search:*
  Traverse the search space of solutions by adhering to some heuristic, which also incorporates a termination criterion
- *Non-deterministic search:*
  Generate subsets of F at random (cf. Algorithm RG) without an explicit termination criterion $\Rightarrow$ at each time point, RG() returns the best subset found thus far [1]

To which category does each of SFG, SBG and BG belong? Heuristic search

[1] In 'stream mining', best effort algorithms of this kind are called *anytime algorithms*.

Thus far: We have seen what feature selection means, and what reasons are there to perform a feature selection process.
We have seen some ways of building a 'good' subset of the original set of features, assuming a criterion of 'goodness', such as redundancy (negative criterion) or good predictive power towards a target (positive criterion)

Thus far: We have seen what feature selection means, and what reasons are there to perform a feature selection process.
We have seen some ways of building a 'good' subset of the original set of features, assuming a criterion of 'goodness', such as redundancy (negative criterion) or good predictive power towards a target (positive criterion)

Your turn: You must be able to explain the four feature set construction algorithms, name their termination criteria (if any), and say whether each of them is exhaustive, heuristic or non-deterministic – and explain why.

Thus far: We have seen what feature selection means, and what reasons are there to perform a feature selection process.
We have seen some ways of building a 'good' subset of the original set of features, assuming a criterion of 'goodness', such as redundancy (negative criterion) or good predictive power towards a target (positive criterion)

Your turn: You must be able to explain the four feature set construction algorithms, name their termination criteria (if any), and say whether each of them is exhaustive, heuristic or non-deterministic – and explain why.

What comes next: 'Goodness' quantified

Goodness as 'redundancy': Correlations between features (1a)

how often the feature are independent, and how often its is observed together
if both are dependent on each other we might discard any one of them.

χ^2 for two categorical attributes (Section 3.2.1.1)

Let A, B be two nominal (categorical) attributes with c, respectively r distinct values, in a dataset with m instances. We compute

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

where o_{ij} is the observed frequency of the joint event (A_i, B_j), and $e_{ij} = \frac{count(A{=}a_i) \times count(B{=}b_j)}{m}$ is the expected frequency.
We test against the hypothesis that A, B are independent (H_0) with $(c-1) \times (r-1)$ degrees of freedom. [a]

[a] To perform a statistical test like χ^2 we use a table or a software that can provide the values for the likelihood of the independence hypothesis.

Goodness as 'redundancy': Correlations between features (1b)

Pearson product moment coeff for two numerical attributes (Section 3.2.1.2)

A, B with means $\overline{A}, \overline{B}$, standard deviations σ_A, σ_B:

$$r_{A,B} = \frac{\sum_{i=1}^{m}(a_i - \overline{A})(b_i - \overline{B})}{m\sigma_A\sigma_B} \in [-1, +1]$$

where (a_i, b_i) the values of A, B at the i^{th} instance in the dataset, $i = 1 \ldots m$.

Goodness as 'redundancy': Correlations between features (1b)

Pearson product moment coeff for two numerical attributes (Section 3.2.1.2)

A, B with means $\overline{A}, \overline{B}$, standard deviations σ_A, σ_B:

$$r_{A,B} = \frac{\sum_{i=1}^{m}(a_i - \overline{A})(b_i - \overline{B})}{m\sigma_A \sigma_B} \in [-1, +1]$$

where (a_i, b_i) the values of A, B at the i^{th} instance in the dataset, $i = 1 \ldots m$.

Pearson correlation coeff (Section 7.2.2.3)

measures the degree of linear correlation between two variables X, Y with measurements $\{x_i\}$ and $\{y_i\}$ and means $\overline{x}, \overline{y}$:

$$\rho(X, Y) = \frac{\sum_i ()x_i - \overline{x})(y_i - \overline{y})}{\sqrt{(\sum_i (x_i - \overline{x})^2 \sum_i (y_i - \overline{y})^2)}}$$

value closer to one will decribe it the two feature values are strongly corellated or no

Goodness as 'redundancy': Correlations between features (1c)

Covariance between two numerical attributes (Section 3.2.1.2)

$$Cov(A,B) = E((A-\overline{A})(B-\overline{B})) = \frac{1}{m}\sum_{i=1}^{m}(a_i-\overline{A})(b_i-\overline{B})$$

- If A, B are independent, then $E(A \cdot B) = E(A) \cdot E(B)$, and thus $Cov(A,B) = 0$
- If A, B vary similarly, then whenever $A > \overline{A}$ we also expect that $B > \overline{B}$, and thus the covariance will be positive.
- If A, B vary in opposite directions, then whenever $A > \overline{A}$ we expect that $B < \overline{B}$, and thus the covariance will be negative.

[2] Covariance_trends.svg.png from Cmglee, CC BY-SA 4.0 https://creativecommons.org/licenses/by-sa/4.0, via Wikimedia Commons

Goodness as 'redundancy': Correlations between features (1c)

Covariance between two numerical attributes (Section 3.2.1.2)

$$Cov(A,B) = E((A-\overline{A})(B-\overline{B})) = \frac{1}{m}\sum_{i=1}^{m}(a_i-\overline{A})(b_i-\overline{B})$$

- If A, B are independent, then $E(A \cdot B) = E(A) \cdot E(B)$, and thus $Cov(A,B) = 0$
- If A, B vary similarly, then whenever $A > \overline{A}$ we also expect that $B > \overline{B}$, and thus the covariance will be positive.
- If A, B vary in opposite directions, then whenever $A > \overline{A}$ we expect that $B < \overline{B}$, and thus the covariance will be negative.

Pearson product moment coeff and the covariance matrix

$$r_{A,B} = \frac{\sum_{i=1}^{m}(a_i-\overline{A})(b_i-\overline{B})}{m} \cdot \frac{1}{\sigma_A \sigma_B} = \frac{Cov(A,B)}{\sigma_A \sigma_B} \quad \text{(Eq. 3.5)}$$

[2] Covariance_trends.svg.png from Cmglee, CC BY-SA 4.0 https://creativecommons.org/licenses/by-sa/4.0, via Wikimedia Commons

Goodness as 'redundancy': Correlations between features (1c)

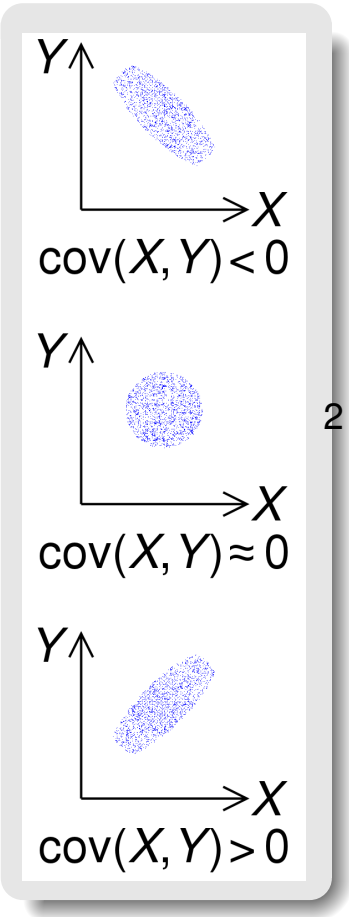
Covariance between two numerical attributes (Section 3.2.1.2)

$$Cov(A,B) = E((A-\overline{A})(B-\overline{B})) = \frac{1}{m}\sum_{i=1}^{m}(a_i-\overline{A})(b_i-\overline{B})$$

- If A, B are independent, then $E(A \cdot B) = E(A) \cdot E(B)$, and thus $Cov(A,B) = 0$
- If A, B vary similarly, then whenever $A > \overline{A}$ we also expect that $B > \overline{B}$, and thus the covariance will be positive.
- If A, B vary in opposite directions, then whenever $A > \overline{A}$ we expect that $B < \overline{B}$, and thus the covariance will be negative.

Pearson product moment coeff and the covariance matrix

$$r_{A,B} = \frac{\sum_{i=1}^{m}(a_i-\overline{A})(b_i-\overline{B})}{m} \cdot \frac{1}{\sigma_A \sigma_B} = \frac{Cov(A,B)}{\sigma_A \sigma_B} \quad \text{(Eq. 3.5)}$$



[2]

[2] Covariance_trends.svg.png from Cmglee, CC BY-SA 4.0 https://creativecommons.org/licenses/by-sa/4.0, via Wikimedia Commons

Goodness as 'redundancy': Correlations between features (1d)

for categorical data- usage of chi square is preffered
below one depends on numerical data

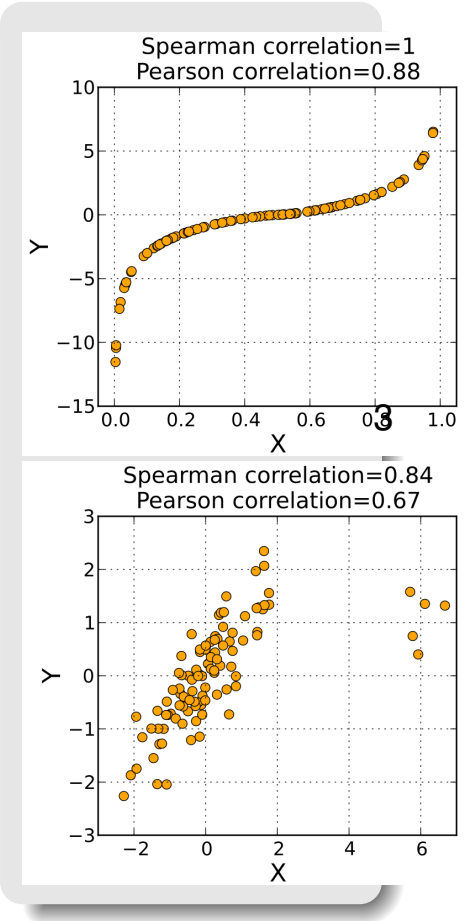
Spearman correlation coeff

is the Pearson correlation coeff for the *ranks of the variables*, i.e.

$$r_s = \rho(R(X), R(Y)) = \frac{Cov(R(X), R(Y))}{\sigma_{R(X)} \sigma_{R(Y)}}$$

where X, Y are numerical variables and $R(X), R(Y)$ are their ranks.

Spearman correlation coeff can be used for ordinal variables, and, iff all ranks are distinct integers, then it holds that [Fieller et al., 1957] : $r_s = 1 - \frac{6 \sum_{i=1}^{m} (x_i - y_i)^2}{m^3 - m}$



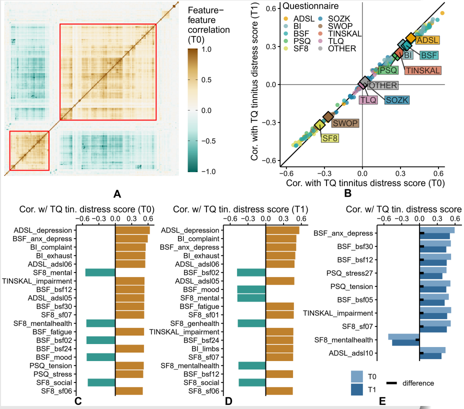
[3] Upper fig: upload.wikimedia.org/wikipedia/commons/4/4e/Spearman_fig1.svg
Lower fig: upload.wikimedia.org/wikipedia/commons/6/67/Spearman_fig3.svg
Skbkekas, CC BY-SA 3.0 https://creativecommons.org/licenses/by-sa/3.0, via Wikimedia Commons

RECALL [Unit 2] Example of a correlation analysis [Niemann et al., 2020] [4]

Correlation analysis (with Spearman correlation coefficient) on a dataset of 4,117 tinnitus patients who had been treated at the Tinnitus Center of Charité Universitaetsmedizin Berlin between January 2011 and October 2015.

Figure 3

Feature-feature correlation & feature correlation with respect to TQ tinnitus-related distress score in T0 and T1.
(A) Correlation heatmap for all pairs of features (T0). Features are ordered by agglomerative hierarchical clustering with complete linkage. (B) Correlation of each feature with TQ tinnitus-related distress score, in T0 (x-axis) and in T1 (y-axis). The diamond symbol represents a questionnaire's median. (C) Top-20 features with highest correlation to TQ tinnitus-related distress score (T0). (D) Top-20 features with highest correlation to TQ tinnitus-related distress score (T1). (E) Top-10 features whose correlational effects with TQ tinnitus-related distress score differ in T0 vs. T1. Correlation values before and after treatment are shown as light blue and dark blue bars, respectively. Differences in correlation are represented as black bars centered in between.



[4] Article at DOI: 10.1371/journal.pone.0228037, figure at DOI: 10.1371/journal.pone.0228037

Goodness towards a target variable (2)

Quantifying goodness as uncertainty U

Given a set of k classes/labels $\mathscr{C}$ with prior class probability $P(c_i)$ for each $i = 1, \ldots, k$ and a feature space F:

- *Information measures:* What is the information gain achieved when we consider a feature $A \in F$ (i) in comparison to $P(c_k)$ or (ii) in comparison to considering another feature $B \in F$? (Section 7.2.2.1)
- *Accuracy-like measures:* How does accuracy change when (i) we consider $A \in F$ additionally to previous selected features, (ii) we skip $A \in F$? (Section 7.2.2.5)

Goodness towards a target variable (2)

Quantifying goodness as uncertainty U

Given a set of k classes/labels $\mathscr{C}$ with prior class probability $P(c_i)$ for each $i = 1, \ldots, k$ and a feature space F:

- *Information measures:* What is the information gain achieved when we consider a feature $A \in F$ (i) in comparison to $P(c_k)$ or (ii) in comparison to considering another feature $B \in F$? (Section 7.2.2.1)
- *Accuracy-like measures:* How does accuracy change when (i) we consider $A \in F$ additionally to previous selected features, (ii) we skip $A \in F$? (Section 7.2.2.5)

Example: information gain as uncertainty reduction (Section 7.2.2.1)

The 'Information Gain' achieved by considering a feature A is the difference between the prior uncertainty $\sum_{i=1}^{k} U(P(c_i))$ and the expected posterior uncertainty using A:

$$IG(A) = \sum_{i=1}^{k} U(P(c_i)) - E\left(\sum_{i=1}^{k} U(P(c_i|A))\right)$$

and as $U()$ we use Shannon's entropy (or a derived function), so that:
$\sum_{i=1}^{k} U(P(c_i)) = -\sum_{i=1}^{k} P(c_i)\log_2 P(c_i)$
and accordingly for the expected posterior uncertainty given A.

Thus far: We have seen ways of quantifying goodness of features.
We focused on correlations between two features, and we have seen some of the many many functions that can be used to compute correlations. We have also re-visited a visualization instrument for showing all pairs of correlation coefficient values in a heatmap matrix.
We have also seen examples of functions that compute the goodness of a feature for class separation.

Thus far: We have seen ways of quantifying goodness of features.
We focused on correlations between two features, and we have seen some of the many many functions that can be used to compute correlations. We have also re-visited a visualization instrument for showing all pairs of correlation coefficient values in a heatmap matrix.
We have also seen examples of functions that compute the goodness of a feature for class separation.

Your turn: You must be able to compute correlation coefficients for example cases, and to explain correlation results. You must be also able to compute information gain with Shannon's entropy.

Thus far: We have seen ways of quantifying goodness of features.
We focused on correlations between two features, and we have seen some of the many many functions that can be used to compute correlations. We have also re-visited a visualization instrument for showing all pairs of correlation coefficient values in a heatmap matrix.
We have also seen examples of functions that compute the goodness of a feature for class separation.

Your turn: You must be able to compute correlation coefficients for example cases, and to explain correlation results. You must be also able to compute information gain with Shannon's entropy.

What comes next: Closing the workflow of feature selection

What is missing from the feature selection workflow?

We already have

- √ functions that help us decide whether a feature is good to keep or rather to discard
- √ procedures to traverse the search space, i.e. all combinations of features, and create feature subsets
- √ stopping criteria

What is missing from the feature selection workflow?

We already have

- √ functions that help us decide whether a feature is good to keep or rather to discard
- √ procedures to traverse the search space, i.e. all combinations of features, and create feature subsets
- √ stopping criteria

but how good is our constructed subset of features ? for a learning algorithm ?

What is missing from the feature selection workflow?

We already have

- √ functions that help us decide whether a feature is good to keep or rather to discard
- √ procedures to traverse the search space, i.e. all combinations of features, and create feature subsets
- √ stopping criteria

but how good is our constructed subset of features ? for a learning algorithm ?

Training and testing

Given a sample of the data D that is representative of the population under study:

- We split D into a training set D_{train} and a test set D_{test} so that $D = D_{train} \cup D_{test}$ and $D_{train} \cap D_{test} = \emptyset$.
- We use D_{train} to train the learning algorithm and induce a model M
- We use D_{test} to test the behaviour of M on previously unseen data

What is missing from the feature selection workflow?

We already have

√ functions that help us decide whether a feature is good to keep or rather to discard

√ procedures to traverse the search space, i.e. all combinations of features, and create feature subsets

√ stopping criteria

but how good is our constructed subset of features ? for a learning algorithm ?

Training and testing

Given a sample of the data D that is representative of the population under study:

- We split D into a training set D_{train} and a test set D_{test} so that $D = D_{train} \cup D_{test}$ and $D_{train} \cap D_{test} = \emptyset$.
- We use D_{train} to train the learning algorithm and induce a model M
- We use D_{test} to test the behaviour of M on previously unseen data

→

? How to test the impact of the feature selection process on the model M

? And WHEN?

Filters (Section 7.2.3.1) & Wrappers (Section 7.2.3.2)

Filters

filter the good from the no-good features *before learning*.

Stage 1: Feature subset selection with a goodness criterion and a feature set generation algorithm – it delivers its best subset of features

Stage 2: The learning algorithm uses the data of this subset of features only, first for training, then for testing

Subcategory **Rankers**: Stage 1 delivers all features, each one ranked on goodness; at stage 2, the algorithm applies a threshold to pick the top-ranked features

Filters (Section 7.2.3.1) & Wrappers (Section 7.2.3.2)

Filters

filter the good from the no-good features *before learning*.

Stage 1: Feature subset selection with a goodness criterion and a feature set generation algorithm – it delivers its best subset of features

Stage 2: The learning algorithm uses the data of this subset of features only, first for training, then for testing

Subcategory **Rankers**: Stage 1 delivers all features, each one ranked on goodness; at stage 2, the algorithm applies a threshold to pick the top-ranked features

Wrappers

engage a classifier to decide whether a feature should be kept or discarded, depending on its impact on classification quality.

Stage 1: Feature subset selection, where the goodness function is in the blackbox of the classification algorithm – it delivers the best subset of features, from the viewpoint of the classifier

Stage 2: as for Filters, but

- ! a model must be induced on the best subset of features
- ! and tested on data that have not been seen before

Summary and Outlook

We have seen

a workflow for the feature selection process:

- $\surd$ Ways of building up a feature subset by adding or discarding features
- $\surd$ Functions that quantify the goodness of features
- $\surd$ Two types of link to the learning algorithm: filter the feature space before learning, or wrap the learning algorithm into the feature selection workflow

Instruments that we skipped:

- Methods for feature *construction* in a projected space, e.g. 'Principal Component Analysis'
- Methods quantifying the discriminative power of a feature towards a target, e.g. SHAP

What comes next: **Shifting from static data to time series**

- 4 What is a time series
- 4 What to learn on one time series, and what to learn on many
- 4 Imputing missing values $\longrightarrow$ filling gaps

Thank you very much!

Questions?

Bibliography I

- Fieller, E. C., Hartley, H. O., and Pearson, E. S. (1957). Tests for rank correlation coefficients. i. *Biometrika*, 44(3/4):470–481.
- Niemann, U., Boecking, B., Brueggemann, P., Mebus, W., Mazurek, B., and Spiliopoulou, M. (2020). Tinnitus-related distress after multimodal treatment can be characterized using a key subset of baseline variables. *PLOS ONE*, 15(1):1–18.
- Schleicher, M., Brüggemann, P., Böcking, B., Niemann, U., Mazurek, B., and Spiliopoulou, M. (2024). Parsimonious predictors for medical decision support: Minimizing the set of questionnaires used for tinnitus outcome prediction. *Expert Systems with Applications*, 239:122336.