Feature Subset Selection

Elizaveta Pechenova

LMU Munich

December 6, 2018

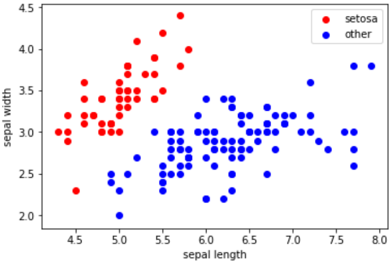
Definition

Feature subset selection - the process of selecting the relevant features for use in model construction.

Intuitively one might think, that the more features there are, the better we can perform our training...

A simple example

- illustration: have a look at iris dataset
- introduce a third random variable



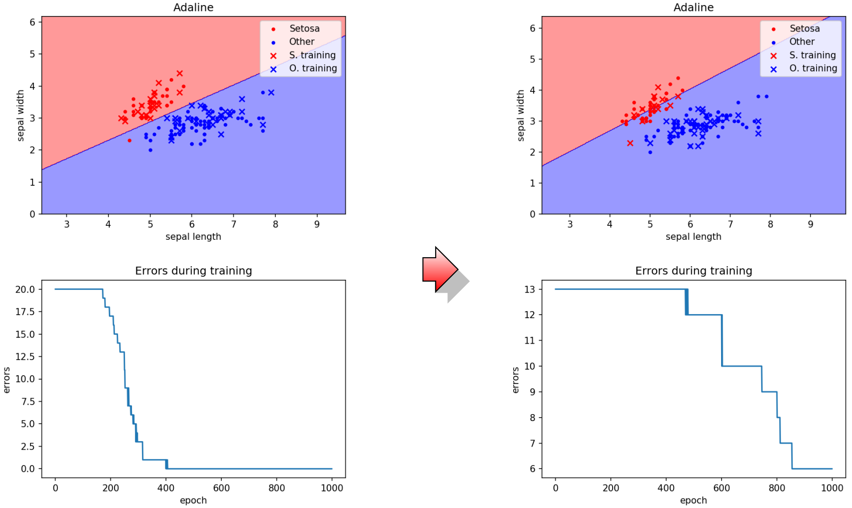
Random variable

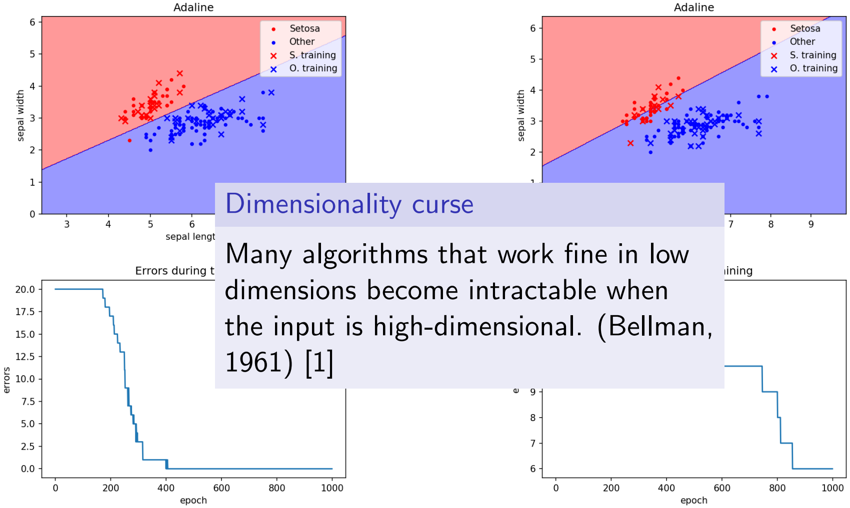
```
In [95]: X_all
Out[95]:
array([[6.4, 3.2],
       [5.5, 2.4],
       [6.5, 3. ],
       [5.5, 2.6],
       [6.1, 2.6],
       [4.8, 3.4],
       [6.7, 3.1],
       [6.5, 3. ],
       [6. , 3. ],
       [6.2, 2.2],
       [6. , 2.2],
       [5.5, 2.4],
       [6.2, 3.4],
       [6.3, 3.3],
       [5.4, 3. ],
       [4.5, 2.3],
       [7.7, 2.6],
       [6.1, 2.8],
       [5.2, 3.4],
       [5.1, 3.4],
       [4.6, 3.2],
       [5.1, 3.5],
       [6.3, 3.3],
```

add random variable

```
In [64]: X_all
Out[64]:
array([[ 6.4,   3.2,   4. ],
       [ 5.5,   2.4,   6. ],
       [ 6.5,   3. ,   3. ],
       [ 5.5,   2.6,   2. ],
       [ 6.1,   2.6,   8. ],
       [ 4.8,   3.4,   9. ],
       [ 6.7,   3.1,   3. ],
       [ 6.5,   3. ,   3. ],
       [ 6. ,   3. ,   6. ],
       [ 6.2,   2.2,   5. ],
       [ 6. ,   2.2,   8. ],
       [ 5.5,   2.4,   1. ],
       [ 6.2,   3.4,  10. ],
       [ 6.3,   3.3,   3. ],
       [ 5.4,   3. ,   8. ],
       [ 4.5,   2.3,   8. ],
       [ 7.7,   2.6,  10. ],
       [ 6.1,   2.8,   4. ],
       [ 5.2,   3.4,  10. ],
       [ 5.1,   3.4,   2. ],
       [ 4.6,   3.2,   7. ],
       [ 5.1,   3.5,   9. ],
       [ 6.3,   3.3,   4. ],
```

Random variable



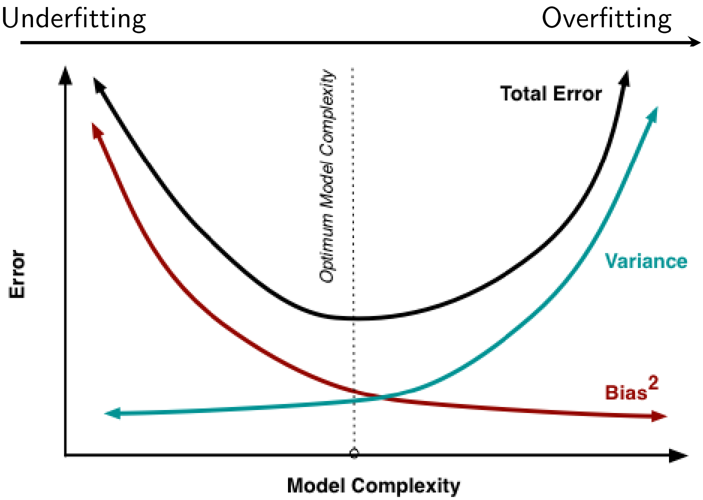


Dimensionality curse

Many algorithms that work fine in low dimensions become intractable when the input is high-dimensional. (Bellman, 1961) [1]

Bias-Variance Dilemma

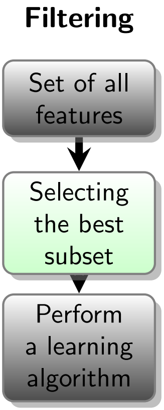
Bias-Variance Dilemma

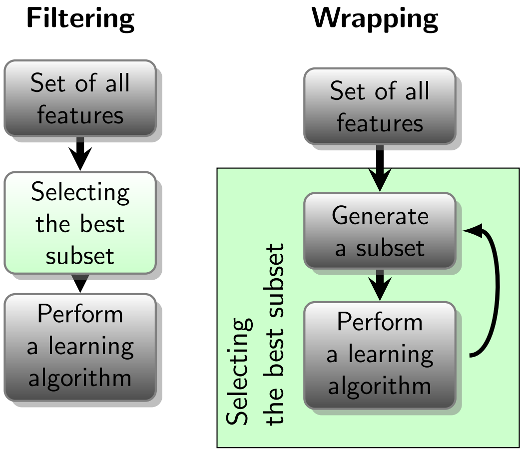


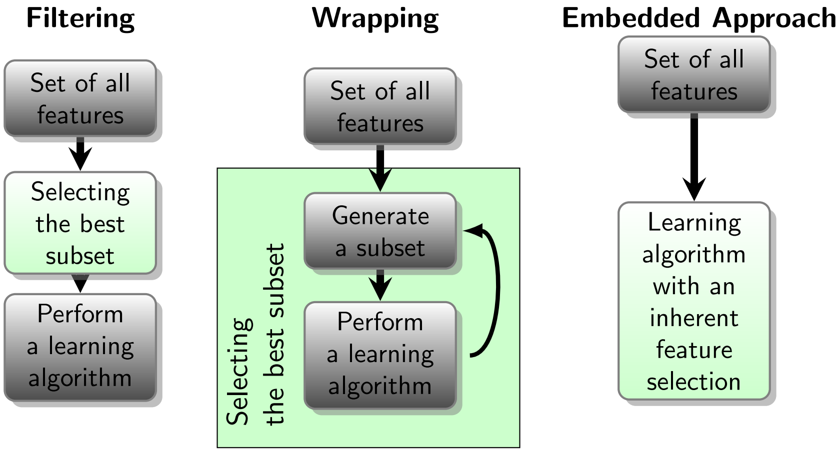
Reasons for using dimensionality reduction

- to improve prediction performance
- to improve learning efficiency
- to provide faster predictors requiring less information
- to reduce complexity of the learned results and enable better understanding of the underlying process
- to prevent over-fitting

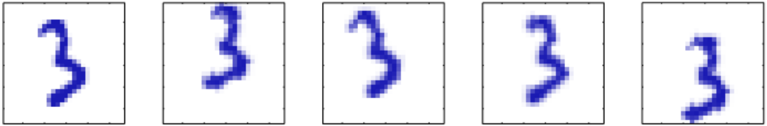
"Here's a list of 100,000 warehouses full of data. I'd like you to condense them down to one meaningful warehouse."


Filtering
Set of all features
Selecting the best subset
Perform a learning algorithm


Filtering
Set of all features
Selecting the best subset
Perform a learning algorithm
Wrapping
Set of all features
Selecting the best subset
Generate a subset
Perform a learning algorithm


Filtering
Set of all features
Selecting the best subset
Perform a learning algorithm
Wrapping
Set of all features
Selecting the best subset
Generate a subset
Perform a learning algorithm
Embedded Approach
Set of all features
Learning algorithm with an inherent feature selection

Principal Component Analysis: Motivation



A synthetic data set obtained by taking one of the off-line digit images and creating multiple copies in each of which the digit has undergone a random displacement and rotation within some larger image field. The resulting images each have $100 \times 100 = 10,000$ pixels.

- simply three degrees of freedom
- vertical and horizontal translations and the rotations
- each image represented by 10000 pixels

Filtering: Principal Component Analysis

Main idea

PCA ... [is] defined as the orthogonal projection of the data onto a lower dimensional linear space, known as the principal subspace, such that the variance of the projected data is maximized [2, 561]

In other words we want to perform dimensionality reduction and keep as much information as possible.

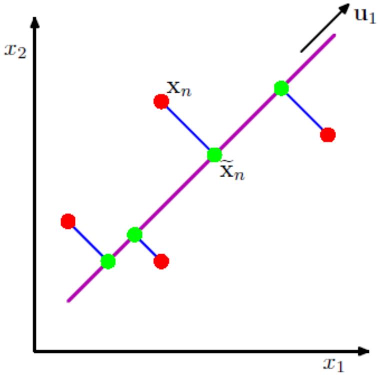
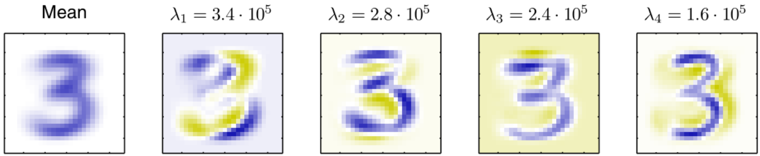


Figure: [2, 561]

Blackboard

Coming back to the example:



The mean vector $\overline{\mathbf{x}}$ along with the first four PCA eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_4$ for the off-line digits data set, together with the corresponding eigenvalues.

Python implementation:

https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html

3D example:

http://setosa.io/ev/principal-component-analysis/

PCA: summary

- Calculate the covariance matrix
- Find the eigenvalues and eigenvectors of the covariance matrix
- Transform the data into the new coordinate system

Pros

- can be applied for data compression and dimensionality reduction
- first insight into the domain at hand – visualization of high dimensional data
- easy method for understanding the data especially in high dimensions
- helps to reduce noise

Cons

- assumes linearity relations between the features
- variance is used as a measure of the importance of the particular dimension
- assumes that principle components are orthogonal

Variable ranking: classical statistics

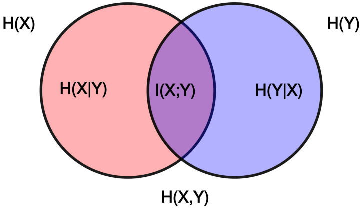
- mutual information
- T-test
- χ^2-test

Mutual Information between X and Y

Definition

Mutual information is a measure of mutual dependence between the chosen variable and the classification variable.

$$I(X;Y) = H(X) - H(X|Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right)$$



Mutual information only zero if X and Y are independent random variables.

Hypothesis test

H_0: feature X_i is irrelevant to Y
H_1: X_i is dependent to Y

χ^2-Test

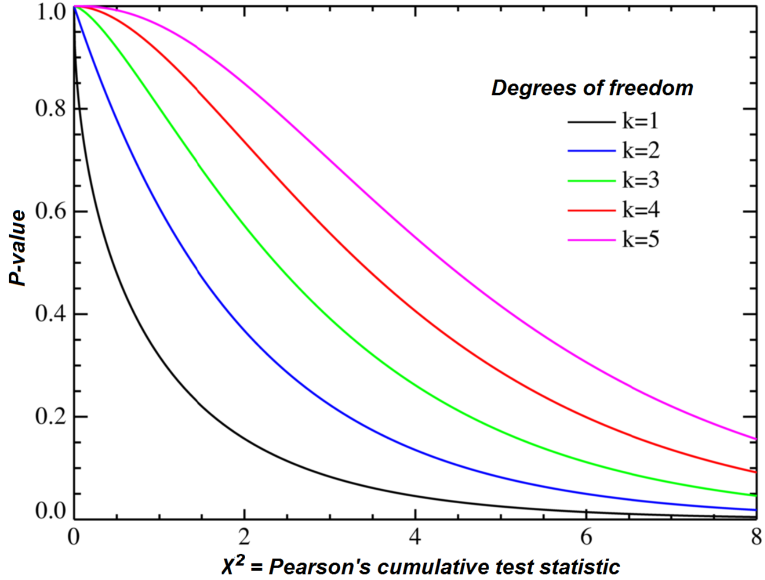
χ^2-Test is based on the assumption, that the two events are independent:

$$P(A \wedge B) = P(A)P(B) \quad (1)$$

Definition

Observed number: O_k
Under H_0 expected number: E_k

$$\chi^2 = \sum_{k=1}^{n} \frac{(O_k - E_k)^2}{E_k}$$


Degrees of freedom
k=1
k=2
k=3
k=4
k=5
P-value
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
X² = Pearson's cumulative test statistic

T−Test: Slope of the regression line

Have a look at the classification variable and one other feature

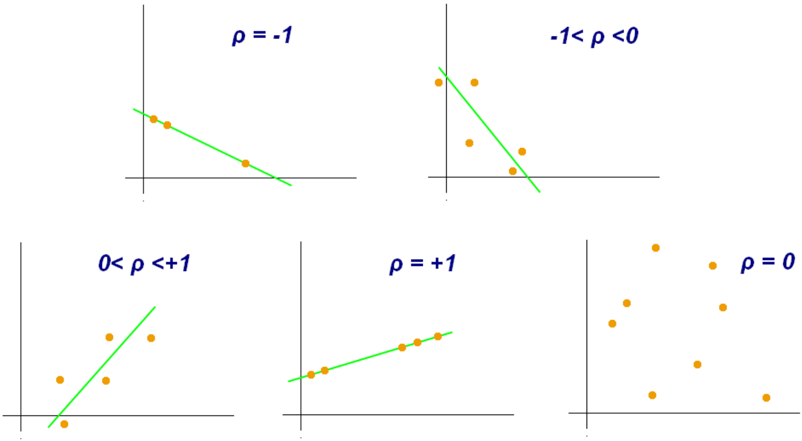
Perform a hypothesis test:

- H_0: the model created by just a constant
- H_1: the model created by a constant and the feature

1. calculate the Pearson correlation $r = \frac{cov(x,y)}{\sqrt{Var(x)Var(y)}}$

$$Cov(x, y) = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

T−Test: Slope of the regression line



$T-$Test: Slope of the regression line

Have a look at the classification variable and one other feature

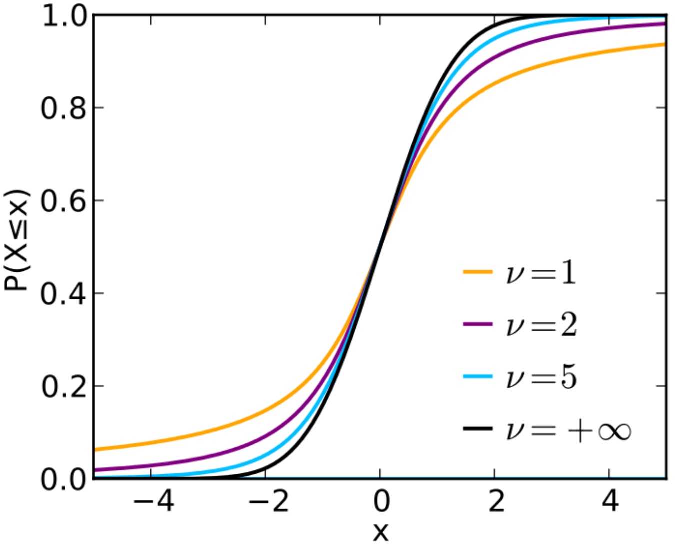
Perform a hypothesis test:

- H_0: the model created by just a constant
- H_1: the model created by a constant and the feature

1. calculate the Pearson correlation $r = \frac{cov(x,y)}{\sqrt{Var(x)Var(y)}}$

$$Cov(x, y) = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

2. compute the t-statistics: $t_{score} = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}}$, n is the number of degrees of freedom
3. calculate the p-value and compare to the significance level
4. sort by variables with the smallest p-values


1.0
0.8
0.6
0.4
0.2
0.0
P(X≤x)
−4
−2
0
2
4
x
$\nu = 1$
$\nu = 2$
$\nu = 5$
$\nu = +\infty$

But what is better? A study on the feature selection algorithms

Table 1
LOOCV classification accuracies with NBC of six gene expression datasets for different gene selection methods using 10–100 selected genes.

Dataset	Method	NBC					
		10	20	40	60	80	100
ALL_AML	ERCS	98.61	97.22	97.22	97.22	97.22	97.22
	Relief-F	93.06	91.67	94.44	91.67	91.67	93.06
	MRMR-FDM	58.33	68.06	61.11	70.83	65.28	65.28
	MRMR-FSQ	48.61	65.28	62.50	58.33	66.67	65.28
	t-Statistic	94.44	95.83	97.22	97.22	97.22	97.22
	Info. Gain	94.44	97.22	95.83	95.83	95.83	95.83
	χ^2-Statistic	97.22	97.22	95.83	95.83	95.83	95.83
COLON	ERCS	82.26	82.26	79.03	80.65	79.03	83.87
	Relief-F	70.97	75.81	75.81	74.19	75.81	79.03
	MRMR-FDM	46.77	46.77	53.23	56.45	61.29	66.13
	MRMR-FSQ	51.61	48.39	58.06	59.68	64.52	64.52
	t-Statistic	82.26	77.42	79.03	80.65	79.03	79.03
	Info. Gain	79.03	79.03	77.42	80.65	79.03	82.26
	χ^2-Statistic	80.65	79.03	79.03	77.42	79.03	79.03
DLBCL	ERCS	94.79	92.71	94.79	94.79	93.75	93.75
	Relief-F	93.75	90.63	90.63	92.71	91.67	90.63
	MRMR-FDM	90.63	89.58	88.54	90.63	91.67	91.67
	MRMR-FSQ	82.29	90.63	90.63	90.63	90.63	91.67
	t-Statistic	93.75	91.67	93.75	94.79	93.75	93.75
	Info. Gain	92.71	92.71	92.71	92.71	92.71	92.71
	χ^2-Statistic	94.79	91.67	93.75	93.75	93.75	93.75
LUNG	ERCS	95.03	96.13	98.90	98.90	98.34	100.00
	Relief-F	92.82	95.03	92.27	97.79	97.24	98.34
	MRMR-FDM	83.43	88.40	91.71	92.82	92.27	92.82
	MRMR-FSQ	82.87	83.43	90.06	90.06	90.06	91.71
	t-Statistic	92.82	92.82	97.24	97.24	97.79	97.79
	Info. Gain	93.37	93.37	93.37	95.03	95.03	95.03
	χ^2-Statistic	92.82	93.37	93.37	95.03	95.03	95.03
MLL	ERCS	94.44	94.44	94.44	95.83	95.83	97.22
	Relief-F	93.06	90.28	90.28	88.89	88.89	90.28
	MRMR-FDM	40.28	41.67	47.22	50.00	47.22	50.00
	MRMR-FSQ	43.06	34.72	54.17	50.00	50.00	48.61
	Info. Gain	93.06	94.44	95.83	94.44	95.83	94.44
	χ^2-Statistic	90.28	93.06	94.44	95.83	94.44	94.44

Table 2
LOOCV classification accuracies with SVM of six gene expression datasets for different gene selection methods using 10 to 100 selected genes.

Dataset	Method	SVM					
		10	20	40	60	80	100
ALL_AML	ERCS	93.06	97.22	97.22	98.61	100.00	98.61
	Relief-F	81.94	90.28	84.72	86.11	87.50	93.06
	MRMR-FDM	58.33	61.11	70.83	80.56	84.72	81.94
	MRMR-FSQ	48.61	59.72	77.78	84.72	87.50	80.56
	t-Statistic	91.67	97.22	95.83	98.61	98.61	97.22
	Info. Gain	91.67	94.44	95.83	98.61	98.61	97.22
	χ^2-Statistic	91.67	95.83	95.83	98.61	97.22	97.22
COLON	ERCS	82.26	80.65	79.03	82.26	80.65	83.87
	Relief-F	69.35	75.81	66.13	75.81	77.42	75.81
	MRMR-FDM	66.13	70.97	70.97	66.13	62.90	67.74
	MRMR-FSQ	62.90	70.97	66.13	69.35	67.74	67.74
	t-Statistic	79.03	77.42	74.19	72.58	74.19	80.65
	Info. Gain	77.42	79.03	75.81	79.03	77.42	77.42
	χ^2-Statistic	79.03	79.03	77.42	74.19	75.81	79.03
DLBCL	ERCS	92.71	93.75	95.83	96.88	96.88	95.83
	Relief-F	91.67	89.58	89.58	85.42	92.71	92.71
	MRMR-FDM	91.67	90.63	91.67	94.79	94.79	93.75
	MRMR-FSQ	82.29	89.58	90.63	89.58	93.75	94.79
	t-Statistic	96.88	95.83	95.83	95.83	96.88	95.83
	Info. Gain	96.88	96.88	96.88	96.88	96.88	97.92
	χ^2-Statistic	96.88	95.83	97.92	96.88	97.92	95.83
LUNG	ERCS	98.34	98.34	99.45	99.45	99.45	99.45
	Relief-F	97.24	97.24	98.90	98.90	98.90	98.90
	MRMR-FDM	82.87	86.74	87.29	91.16	95.58	95.58
	MRMR-FSQ	83.43	87.29	82.87	87.29	91.71	93.37
	t-Statistic	97.79	97.24	97.79	98.34	99.45	99.45
	Info. Gain	98.34	97.24	99.45	99.45	98.90	98.90
	χ^2-Statistic	98.34	95.03	99.45	99.45	98.90	99.45
MLL	ERCS	88.89	93.06	97.22	95.83	95.83	97.22
	Relief-F	80.56	87.50	87.50	88.89	93.06	91.67
	MRMR-FDM	59.72	54.17	59.72	72.22	73.61	79.17
	MRMR-FSQ	44.44	56.94	65.28	69.44	69.44	66.67
	Info. Gain	87.50	91.67	91.67	95.83	97.22	97.22
	χ^2-Statistic	87.50	93.06	93.06	90.28	91.67	95.83

Nevertheless there is a difference...

`https://scikit-learn.org/stable/auto_examples/feature_selection/plot_f_test_vs_mi.html`

- F-statistics is better in capturing linear relationships
- χ^2 and MI almost the same for big sample sizes
- MI is easy to compute
- use filters to get rid of about the half of the features and use multiple of them

Wrapper Approach

Main idea

Use the learning algorithm itself to evaluate the goodness of the feature subset. At each step remove different features from the subset. The subset with the highest evaluation is chosen as the final set on which to run the induction algorithm. [3]

The search space for n features has the dimensionality $O(2^n)$

Wrapper Approach

- forward selection
- backward elimination
- random choice: e.g. generic algorithms - algorithms using mutation, crossover and selection
- Problem: risk of over-fitting, computationally expensive
- not used in the era of big data

Embedded Approach: Regularization

Small reminder: Regularization

Introduce an additional constraint, a **regularizer**, to the loss function, which penalties complexity to avoid over-fitting.

L2/Ridge Regularization

$$\text{minimize } \sum_{i=1}^{n}(y_i - w_i^T x_i)^2 \text{ s.t. } \|w\|^2 \leq t$$

$$L_{l2} = \sum_{i=1}^{n}(y_i - w_i^T x_i)^2 + \lambda \sum_{j=1}^{n} w_j^2$$

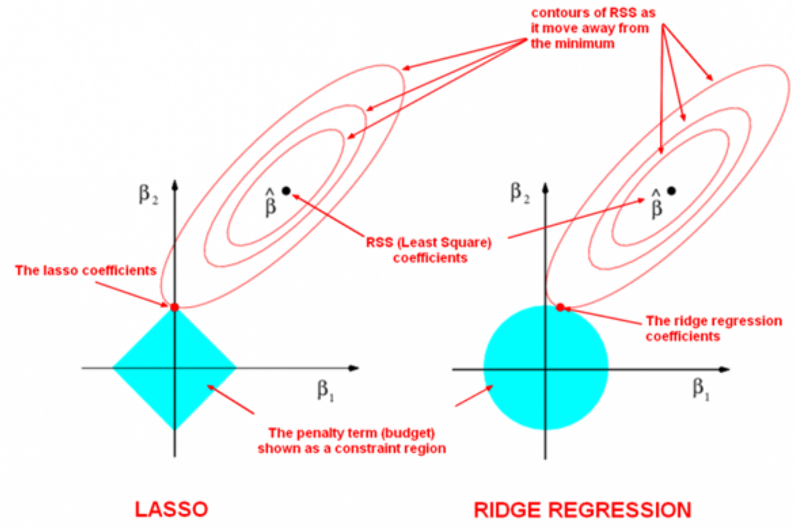
Lasso regression

Main idea: use l_1-norm of the weight vector

$$L_{lasso} = \sum_{i=1}^{n} (y_i - w_i^T x_i)^2 + \lambda \|w\|_1 \quad (2)$$

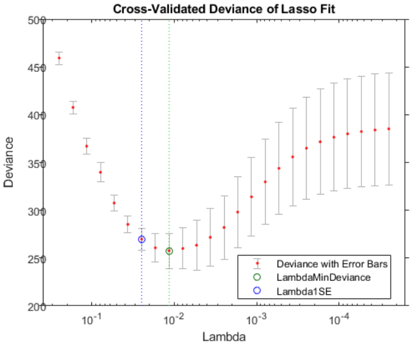
[4]

Lasso regression

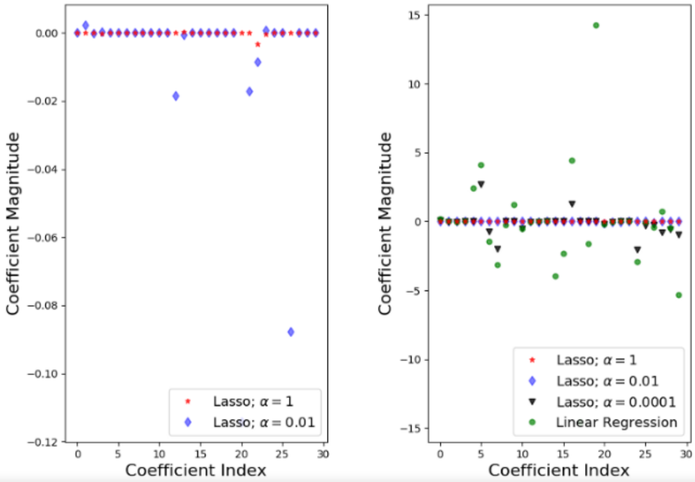


[5]

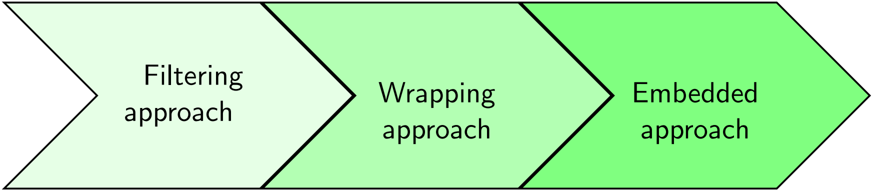
- Lasso regression forces some weights to zero.
- implemented feature selection in the model
- lambda determines the size of the feature set: determined by the cross-validation risk estimate
- breaks down for non-linear methods, as no natural mapping between weights and data
- other approaches exist like feature vector machine: modification of Lasso regression, applies a kernel function K to the feature vectors



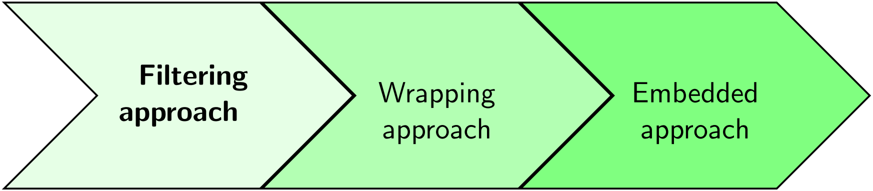
Example for λ-Choice



Overview: Feature Selection methods

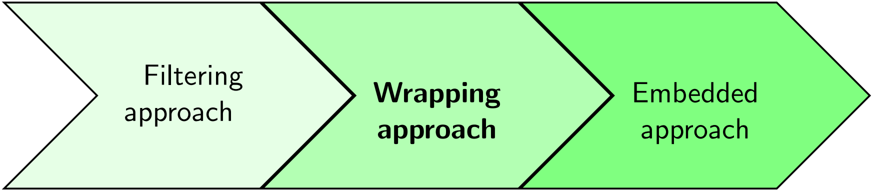


Overview: Feature Selection methods



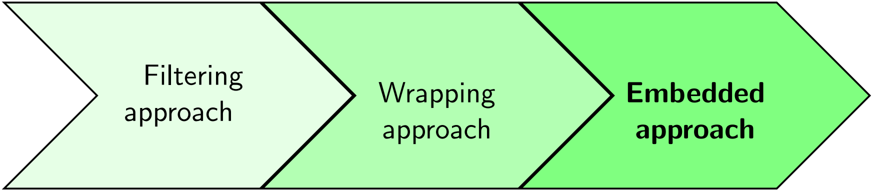
- PCA
- Variable ranking: Mutual information, χ^2-Test, T-Test

Overview: Feature Selection methods



- due to the age of big data rather unpopular as computationally expensive

Overview: Feature Selection methods



- Lasso regression


TIME WELL SPENT™
by Tom Fishburne
SEE, I TOLD YOU THAT BIG DATA WAS TOO SCARY
BIG DATA
KRONOS
Workforce Innovation That Works™
© 2012
KRONOS.COM/TIMEWELLSPENT

[1] Richard E. Bellman.
Adaptive Control Processes: A Guided Tour.
MIT Press, 1961.

[2] Christopher M. Bishop.
Pattern Recognition and Machine Learning (Information Science and Statistics).
Springer-Verlag, Berlin, Heidelberg, 2006.

[3] Dunja Mladenic.
Feature selection for dimensionality reduction.
In *Subspace, Latent Structure and Feature Selection, Statistical and Optimization, Perspectives Workshop, SLSFS 2005, Bohinj, Slovenia, February 23-25, 2005, Revised Selected Papers*, pages 84–102, 2005.

[4] Roland Nilsson.
Statistical Feature Selection.
LiU-Tryck, 2007.

[5] Quora.
How would you describe the difference between linear regression, lasso regression, and ridge regression?
`https://www.quora.com/How-would-you-describe-the-difference-between-linear-regression-lasso-regression-and-ridge-regressio`
Accessed: 2018-12-05.