UNIVERSITY OF MANNHEIM

Data Mining

Classification

- Part 3 -

Outline

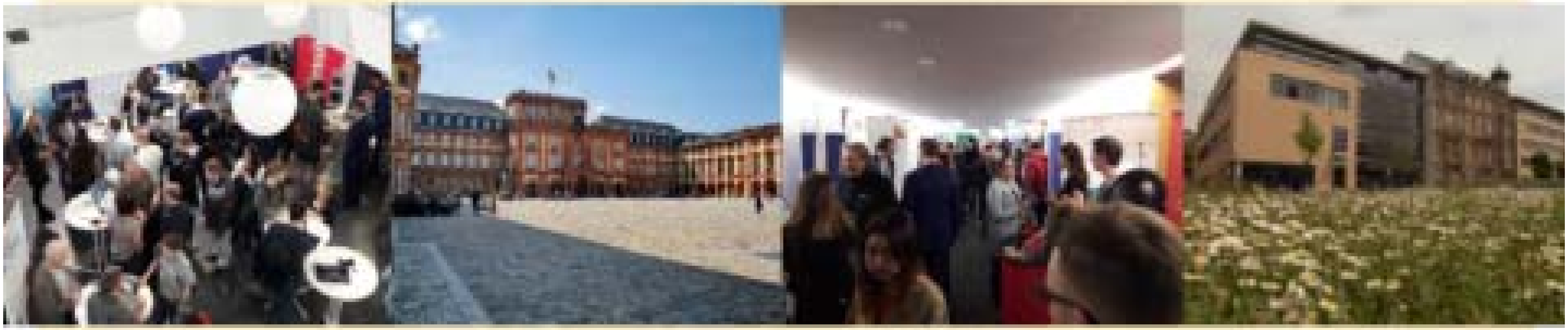
1. What is Classification?
2. K-Nearest-Neighbors
3. Decision Trees
4. Rule Learning
5. Decision Boundaries
6. Model Evaluation
7. Naïve Bayes
8. Artificial Neural Networks
9. Support Vector Machines
10. Parameter Tuning

20.03.2019 – 16:00-18:00 Uhr

MINT-MARKTPLATZ

Fakultät für Wirtschaftsinformatik & Wirtschaftsmathematik

B6, 30-32, Bauteil E-F (Neubau) im 1.OG



Accenture
AppSphere
BASF
Brandt & Partner
BridgingIT
Camelot
d-fine
EXA Deutschland
Inter-Versicherung
KPMG
Materna
mayato
MSW & Partner
Porsche
Procter & Gamble
Roche
SAP
Scheer
Stadt Mannheim
Stocard
Xenium
zeb

Data Fest 2019 in Mannheim

– Visualize and mine from 3rd to 5th May 2019, Registration: 18th April
– https://hiwissml.github.io/datafest2019/

6. Naïve Bayes

- Probabilistic classification technique based on Bayes theorem.
  - Widely used and especially successful at classifying texts
- Goal: Estimate the most probable class label for a given record.
- Probabilistic formulation of the classification task:
  - consider each attribute and class label as random variables
  - Given a record with attributes $(A_1, A_2, \ldots, A_n)$, the goal is to find the class C that maximizes the conditional probability

$$P(C \mid A_1, A_2, \ldots, A_n)$$

- Example: Should we play golf?
  - P(Play=yes | Outlook=rainy, Temperature=cool)
  - P(Play=no | Outlook=rainy, Temperature=cool)
- Question: How to estimate these probabilities given training data?

Bayes Theorem

- Thomas Bayes (1701-1761)
  - British mathematician and priest
  - tried to formally prove the existence of God
- Bayes Theorem

$$P(C|A) = \frac{P(A|C)P(C)}{P(A)}$$

- useful in situations where P(C|A) is unknown while P(A|C), P(A) and P(C) are known or easy to estimate

Bayes Theorem: Evidence Formulation

– Prior probability of event H:
  - Probability of event <u>before</u> evidence is seen.
  - We play golf in 70% of all cases ➔ P(H) = 0.7
– Posterior probability of event H:
  - Probability of event *after* evidence is seen.
  - Evidence: It is windy and raining ➔ P(H | E) = 0.2
– Probability of event *H* given evidence *E*:

$$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)}$$

Applying Bayes Theorem to the Classification Task

Evidence = record

Class-conditional probability of evidence

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

Prior probability of class

Class

Prior probability of evidence

1. Compute the probability P(C | A) for all values of C using Bayes theorem.
   - P(A) is the same for all classes. Thus, we need to estimate P(C) and P(A|C)
2. Choose value of C that maximizes P(C | A).

Example:

$$P(\text{Play}=\text{yes} \mid \text{Outlook}=\text{rainy}, \text{Temp}=\text{cool}) = \frac{P(\text{Outlook}=\text{rainy}, \text{Temp}=\text{cool} \mid \text{Play}=\text{yes})P(\text{Play}=\text{yes})}{P(\text{Outlook}=\text{rainy}, \text{Temp}=\text{cool})}$$

$$P(\text{Play}=\text{no} \mid \text{Outlook}=\text{rainy}, \text{Temp}=\text{cool}) = \frac{P(\text{Outlook}=\text{rainy}, \text{Temp}=\text{cool} \mid \text{Play}=\text{no})P(\text{Play}=\text{no})}{P(\text{Outlook}=\text{rainy}, \text{Temp}=\text{cool})}$$

Estimating the Prior Probability P(C)

– The prior probability $P(C_j)$ for each class is estimated by
  1. counting the records in the training set that are labeled with class C_j
  2. dividing the count by the overall number of records

– Example:
  - P(Play=no) = 5/14
  - P(Play=yes) = 9/14

Training Data

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Estimating the Class-Conditional Probability P(A | C)

- Naïve Bayes assumes that all attributes are *statistically independent*.
  - knowing the value of one attribute says nothing about the value of another
  - this independence assumption is almost never correct!
  - but … this scheme works well in practice
- The independence assumption allows the joint probability P(A | C) to be reformulated as the product of the individual probabilities $P(A_i | C_j)$:

$$P(A_1, A_2, \ldots, A_n | C_j) = P(A_1 | C_j) \times P(A_2 | C_j) \times \ldots \times P(A_n | C_j)$$

P(Outlook=rainy, Temperature=cool | Play=yes) = P(Outlook=rainy | Play=yes) × P(Temperature=cool | Play=yes)

- Result: The probabilities $P(A_i | C_j)$ for all A_i and C_j can be estimated directly from the training data

Estimating the Probabilities $P(A_i \mid C_j)$

Outlook			Temperature			Humidity			Windy			Play	
	Yes	*No*		*Yes*	*No*		*Yes*	*No*		*Yes*	*No*	*Yes*	*No*
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	**2/9**	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

The probabilities $P(A_i \mid C_j)$ are estimated by

1. counting how often an attribute value appears together with class C_j
2. dividing the count by the overall number of records belonging to class C_j

Example:
2 times “Yes” together with “Outlook=sunny” out of altogether 9 “Yes” examples
➔ p(Outlook=sunny|Yes) = 2/9

Classifying a New Day

Unseen Record

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

Class-conditional probability of the evidence

$$\Pr[yes \mid E] = \Pr[Outlook = Sunny \mid yes]$$
$$\times \Pr[Temperature = Cool \mid yes]$$
$$\times \Pr[Humidity = High \mid yes]$$
$$\times \Pr[Windy = True \mid yes]$$
$$\times \frac{\Pr[yes]}{\Pr[E]}$$

Probability of class "yes" given the evidence

Prior probability of class "yes"

Prior probability of evidence

$$= \frac{\frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14}}{\Pr[E]}$$

Classifying a New Day: Weigh the Evidence!

Outlook			Temperature			Humidity			Windy			Play	
	Yes	*No*		*Yes*	*No*		*Yes*	*No*		*Yes*	*No*	*Yes*	*No*
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

– A new day:

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

Prior probability

Evidence

Likelihood of the two classes

For "yes" = 2/9 × 3/9 × 3/9 × 3/9 × 9/14 = 0.0053

For "no" = 3/5 × 1/5 × 4/5 × 3/5 × 5/14 = 0.0206

Conversion into a probability by normalization:

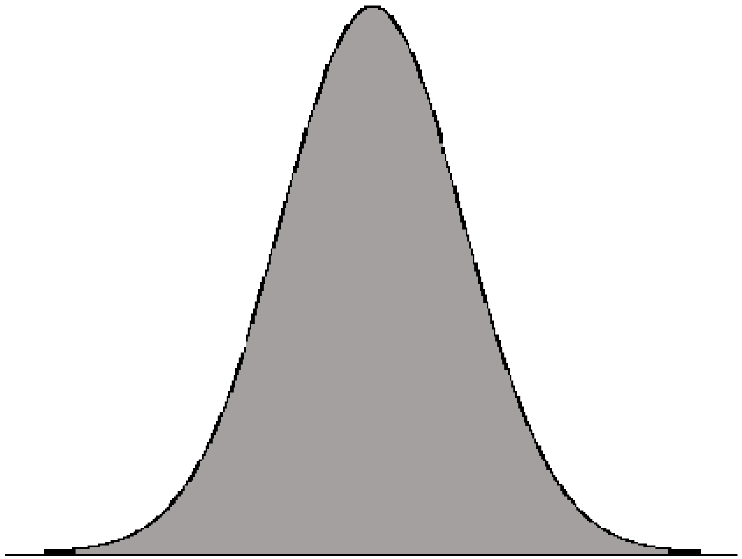
P("yes") = 0.0053 / (0.0053 + 0.0206) = 0.205

Choose Maximum

P("no") = 0.0206 / (0.0053 + 0.0206) = 0.795

Handling Numerical Attributes

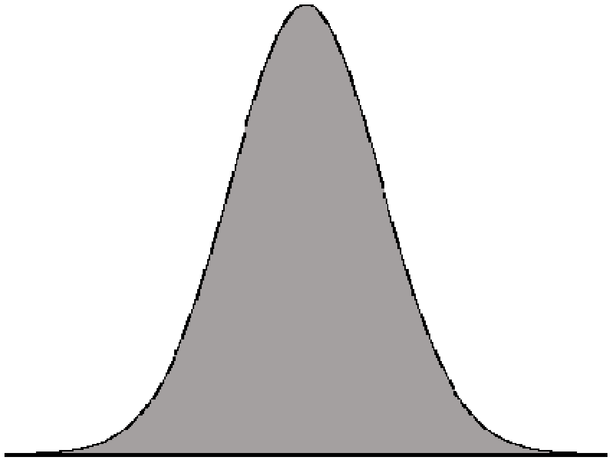
- Option 1:
  Discretize numerical attributes before learning classifier.
  - Temp= 37°C ➔ "Hot"
  - Temp= 21°C ➔ "Mild"
- Option 2:
  Make assumption that numerical attributes have a normal distribution given the class.
  - Use training data to estimate parameters of the distribution
    (e.g., mean and standard deviation)
  - Once the probability distribution is known, it can be used to estimate the conditional probability $P(A_i|C_j)$



Handling Numerical Attributes

– The probability density function for the normal distribution is

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



– It is defined by two parameters:

- *Sample mean μ*

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- *Standard deviation σ*

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2}$$

– Both parameters can be estimated from the training data.

Statistics for the Weather Data

Outlook			Temperature		Humidity		Windy			Play	
	Yes	*No*	*Yes*	*No*	*Yes*	*No*		*Yes*	*No*	*Yes*	*No*
Sunny	2	3	64, 68,	65, 71,	65, 70,	70, 85,	False	6	2	9	5
Overcast	4	0	69, 70,	72, 80,	70, 75,	90, 91,	True	3	3		
Rainy	3	2	72, …	85, …	80, …	95, …					
Sunny	2/9	3/5	μ =73	μ =75	μ =79	μ =86	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	σ =6.2	σ =7.9	σ =10.2	σ =9.7	True	3/9	3/5		
Rainy	3/9	2/5									

Example calculation:

$$f(temp = 66 \mid yes) = \frac{1}{\sqrt{2\pi}\,6.2} e^{-\frac{(66-73)^2}{2*6.2^2}} = 0.0340$$

Classifying a New Day

Unseen Record

Outlook	Temp.	Humidity	Windy	Play
Sunny	66	90	true	?

Likelihood of "yes" = $2/9 \times 0.0340 \times 0.0221 \times 3/9 \times 9/14 = 0.000036$

Likelihood of "no" = $3/5 \times 0.0291 \times 0.0380 \times 3/5 \times 5/14 = 0.000136$

P("yes") = 0.000036 / (0.000036 + 0. 000136) = 20.9%

P("no") = 0.000136 / (0.000036 + 0. 000136) = 79.1%

But note: Some numeric attributes are not normally distributed and you may thus need to choose a different probability density function or use discretization.

Handling Missing Values

– Missing values may occur in training and classification examples.

– Training: Instance is not included in frequency count for attribute value-class combination.

– Classification: Attribute will be omitted from calculation.

– Example:

Outlook	Temp.	Humidity	Windy	Play
?	Cool	High	True	?

Likelihood of "yes" = $3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0238$

Likelihood of "no" = $1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0343$

P("yes") = $0.0238 / (0.0238 + 0.0343) = 41\%$

P("no") = $0.0343 / (0.0238 + 0.0343) = 59\%$

The Zero-Frequency Problem

– What if an attribute value doesn't occur with every class value? (e.g. no "Outlook = overcast" for class "no")
  - Class-conditional probability will be zero! $P[Outlook = overcast \mid no] = 0$
  - *Problem: Posterior* probability will also be zero! (No matter how likely the other values are!) $P[no \mid E] = 0$

– Remedy: Add 1 to the count for every attribute value-class combination (*Laplace Estimator*)

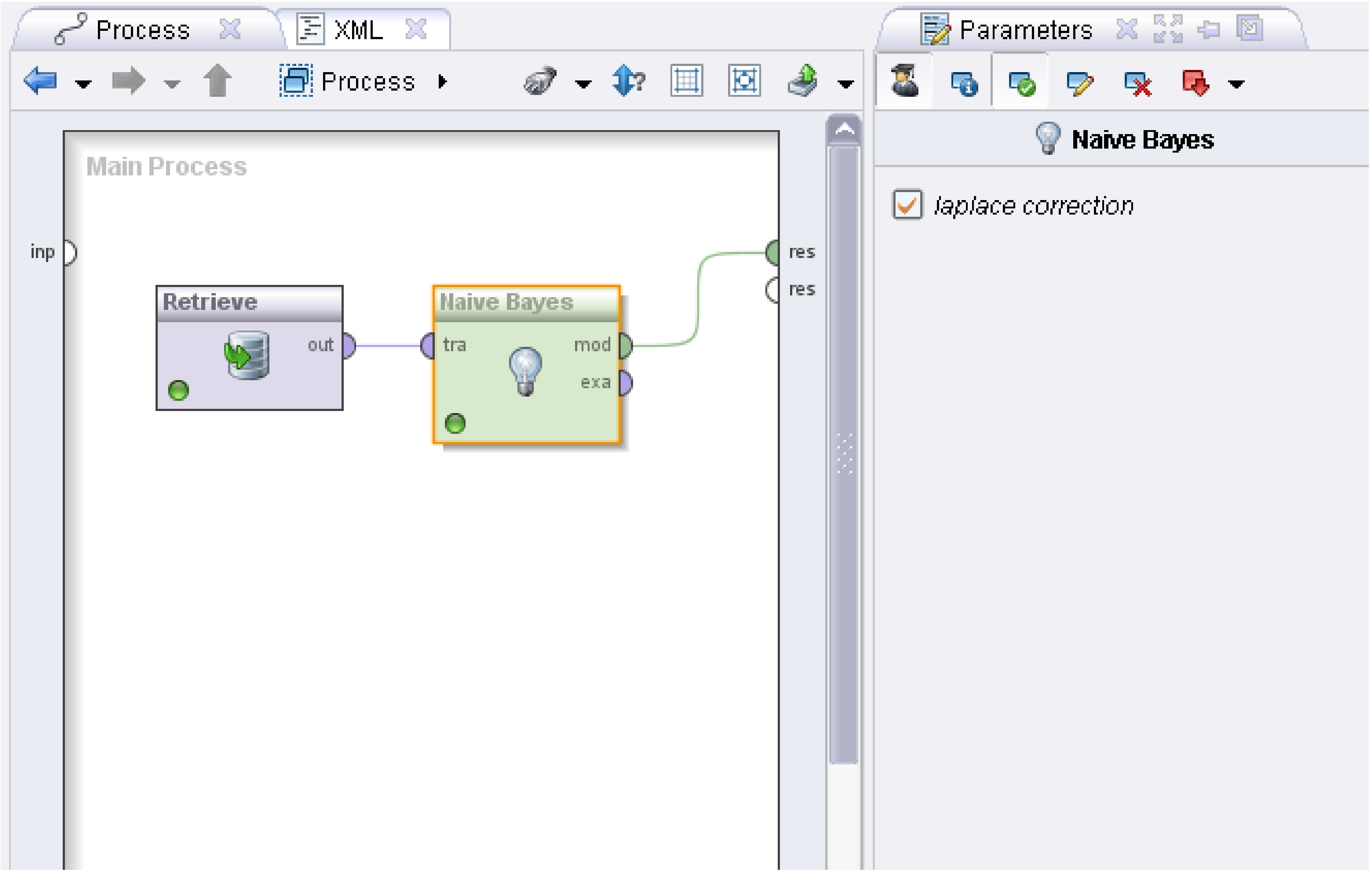
– Result: Probabilities will never be zero! (also: stabilizes probability estimates)

$$\text{Original}: P(A_i \mid C) = \frac{N_{ic}}{N_c}$$

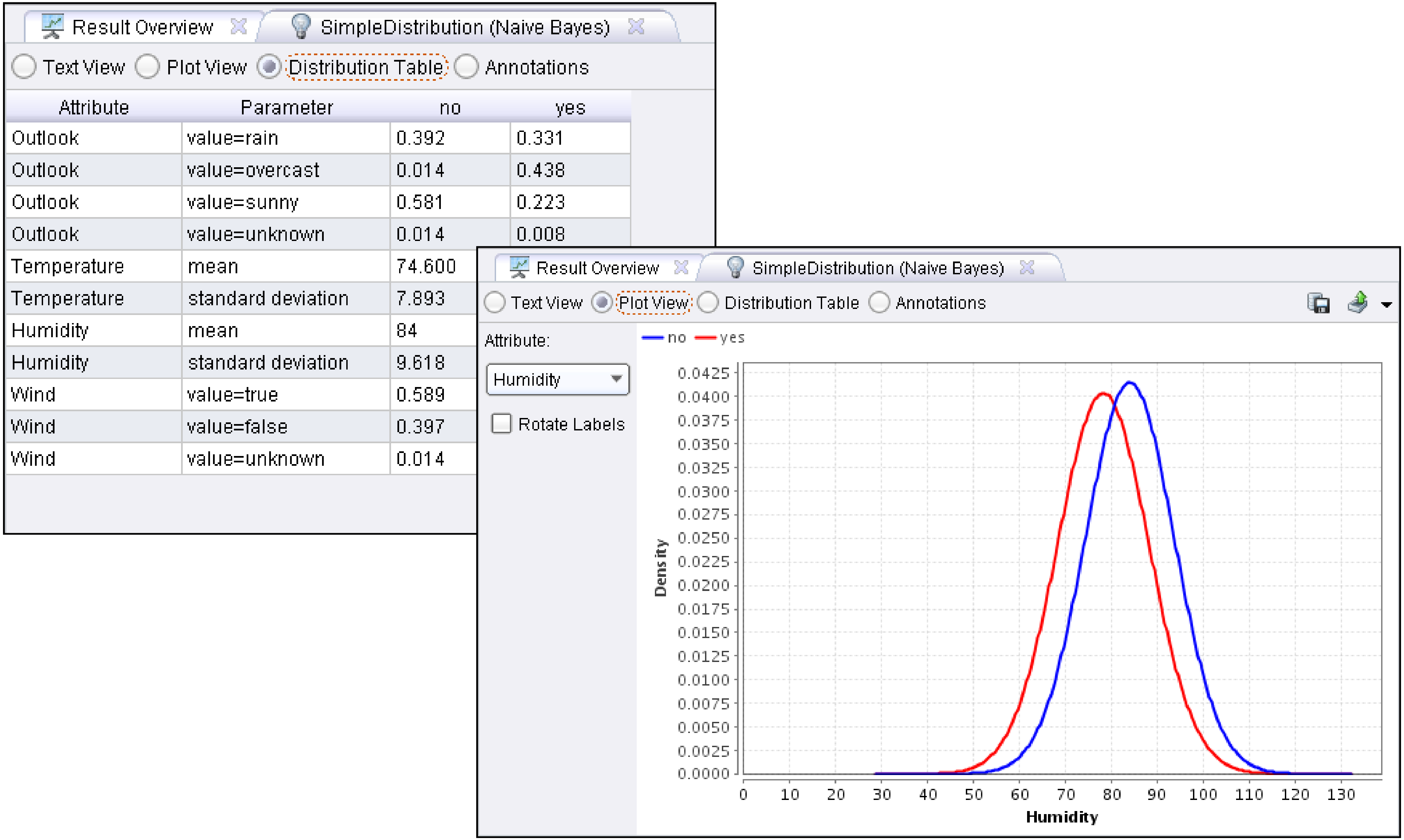
$$\text{Laplace}: P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + c}$$

c: number of classes

Naïve Bayes in RapidMiner



Naïve Bayes in RapidMiner: The Distribution Table



Attribute	Parameter	no	yes
Outlook	value=rain	0.392	0.331
Outlook	value=overcast	0.014	0.438
Outlook	value=sunny	0.581	0.223
Outlook	value=unknown	0.014	0.008
Temperature	mean	74.600	
Temperature	standard deviation	7.893	
Humidity	mean	84	
Humidity	standard deviation	9.618	
Wind	value=true	0.589	
Wind	value=false	0.397	
Wind	value=unknown	0.014	

Naïve Bayes in RapidMiner: Confidence Scores

Result Overview | ExampleSet (Retrieve Golf-Testset)

Data View | Meta Data View | Plot View | Advanced Charts | Annotations

ExampleSet (14 examples, 4 special attributes, 4 regular attributes) View Filter (14 / 14):

Row No.	Play	confidence(no)	confidence(yes)	prediction(Play)	Outlook	Temperature	Humidity	Wind
1	yes	0.711	0.289	no	sunny	85	85	false
2	no	0.058	0.942	yes	overcast	80	90	true
3	yes	0.014	0.986	yes	overcast	83	78	false
4	yes	0.412	0.588	yes	rain	70	96	false
5	yes	0.460	0.540	yes				true
6	no	0.336	0.664					true
7	yes	0.010	0.990	yes	overcast	64	65	true
8	no	0.596	0.404	no	sunny	72	95	false
9	yes	0.248	0.752	yes	sunny	69	70	false
10	no	0.407	0.593	yes	sunny	75	80	false
11	yes	0.496	0.504					true
12	yes	0.038	0.962	yes				true
13	no	0.027	0.973	yes	overcast	81	75	true
14	yes	0.453	0.547	yes	rain	71	80	true

Classifier is quite sure

Classifier is not sure

Characteristics of Naïve Bayes

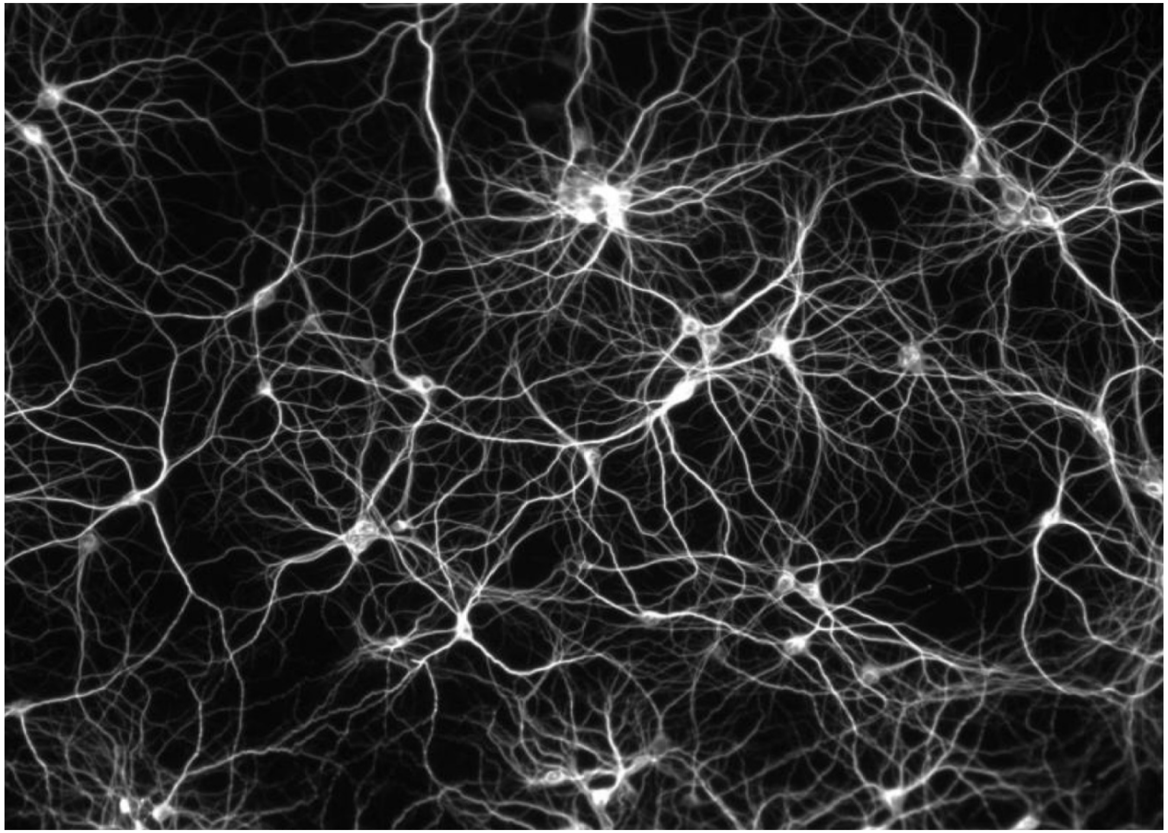
- Naïve Bayes works surprisingly well for many classification tasks.
  - even if independence assumption is clearly violated
  - Why? Because classification doesn't require accurate probability estimates *as long as* maximum probability *is assigned to correct class*
- Robust to isolated noise points as they will be averaged out
- Robust to irrelevant attributes as $P(A_i| C)$ distributed uniformly for A_i
- Adding too many redundant attributes can cause problems.
  - Solution: Select attribute subset as Naïve Bayes often works better with just a fraction of all attributes.
- Technical advantages
  - Learning Naïve Bayes classifiers is computationally cheap as probabilities can be estimated doing one pass over the training data.
  - Storing the probabilities does not require a lot on memory.

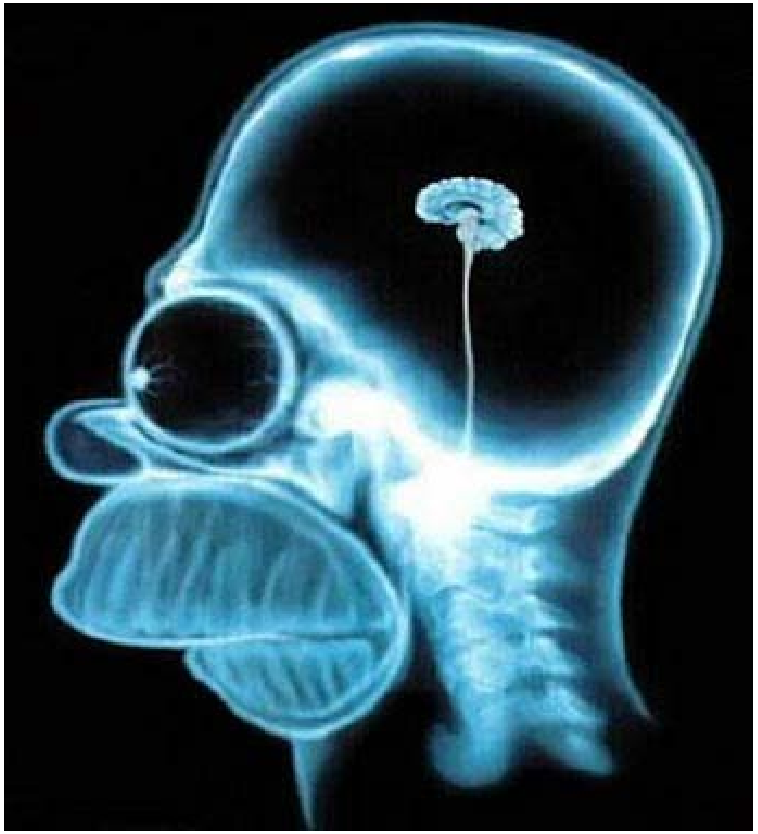
Further Classification Methods

- There are various methods of classification
  - e.g., RapidMiner implements 53 different methods
- So far, we have seen
  1. k-NN
  2. Decision Trees
  3. C4.5 and Ripper
  4. Naive Bayes
- Now: Brief introduction to
  1. Artificial Neural Networks
  2. Support Vector Machines

7. Artificial Neural Networks (ANN)

- Inspiration
  - one of the most powerful super computers in the world

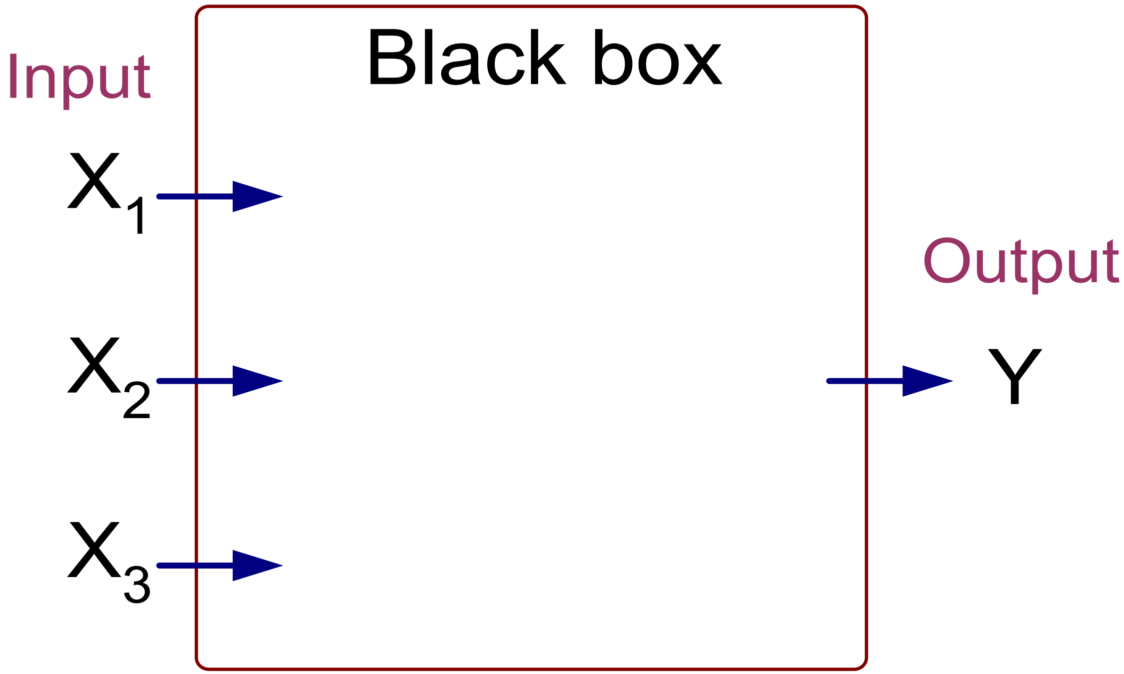




Artificial Neural Networks (ANN)

Training Data

X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0

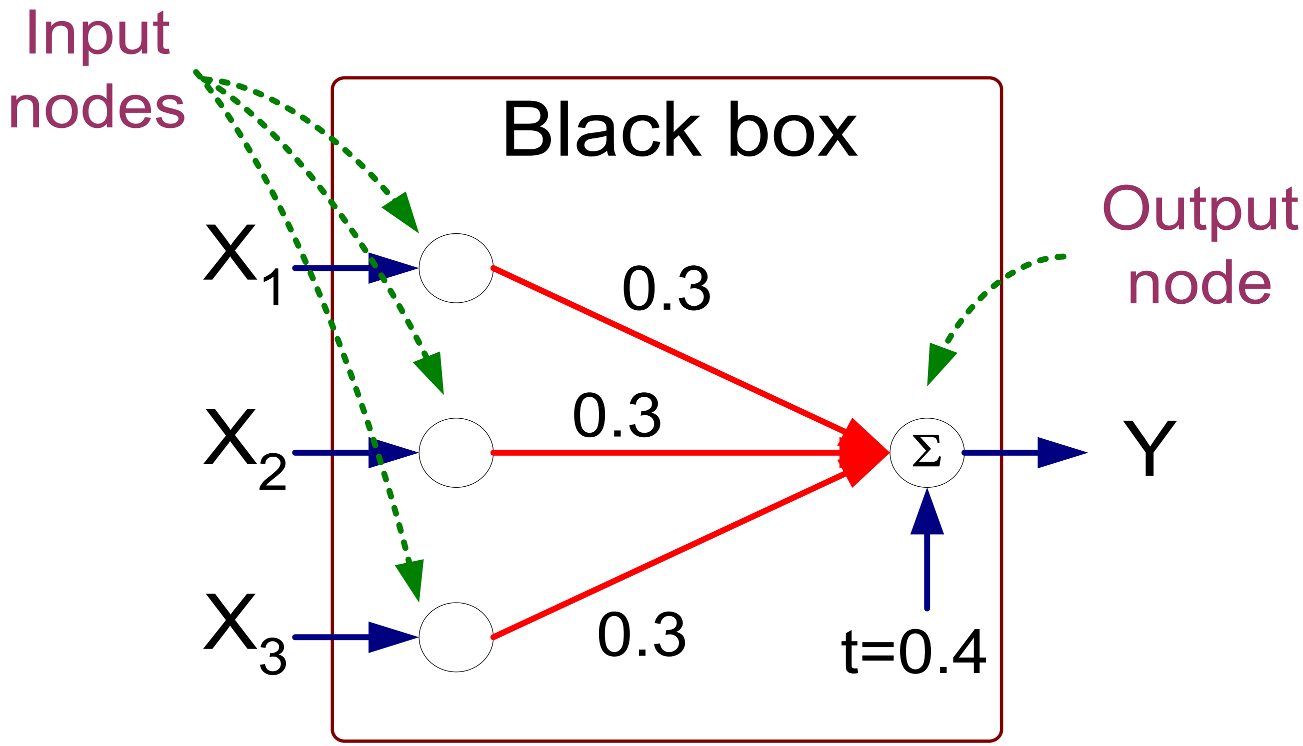


Example:
Output Y is 1 if at least two of the three inputs are equal to 1.

Artificial Neural Networks (ANN)

Training Data

X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0



$$Y = I(0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4 > 0)$$

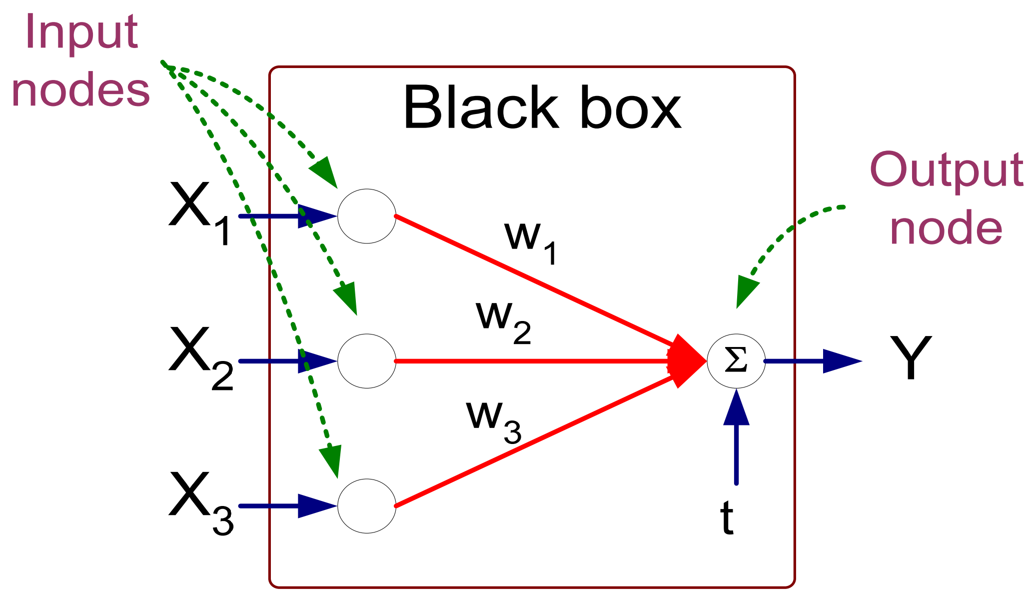
$$\text{where} \quad I(z) = \begin{cases} 1 & \text{if } z \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

Artificial Neural Networks (ANN)

- Model is an assembly of inter-connected nodes (called neurons) and weighted links

- Output node sums up each of its input values according to the weights of its links

- Classification decision: Compare output node against some threshold t

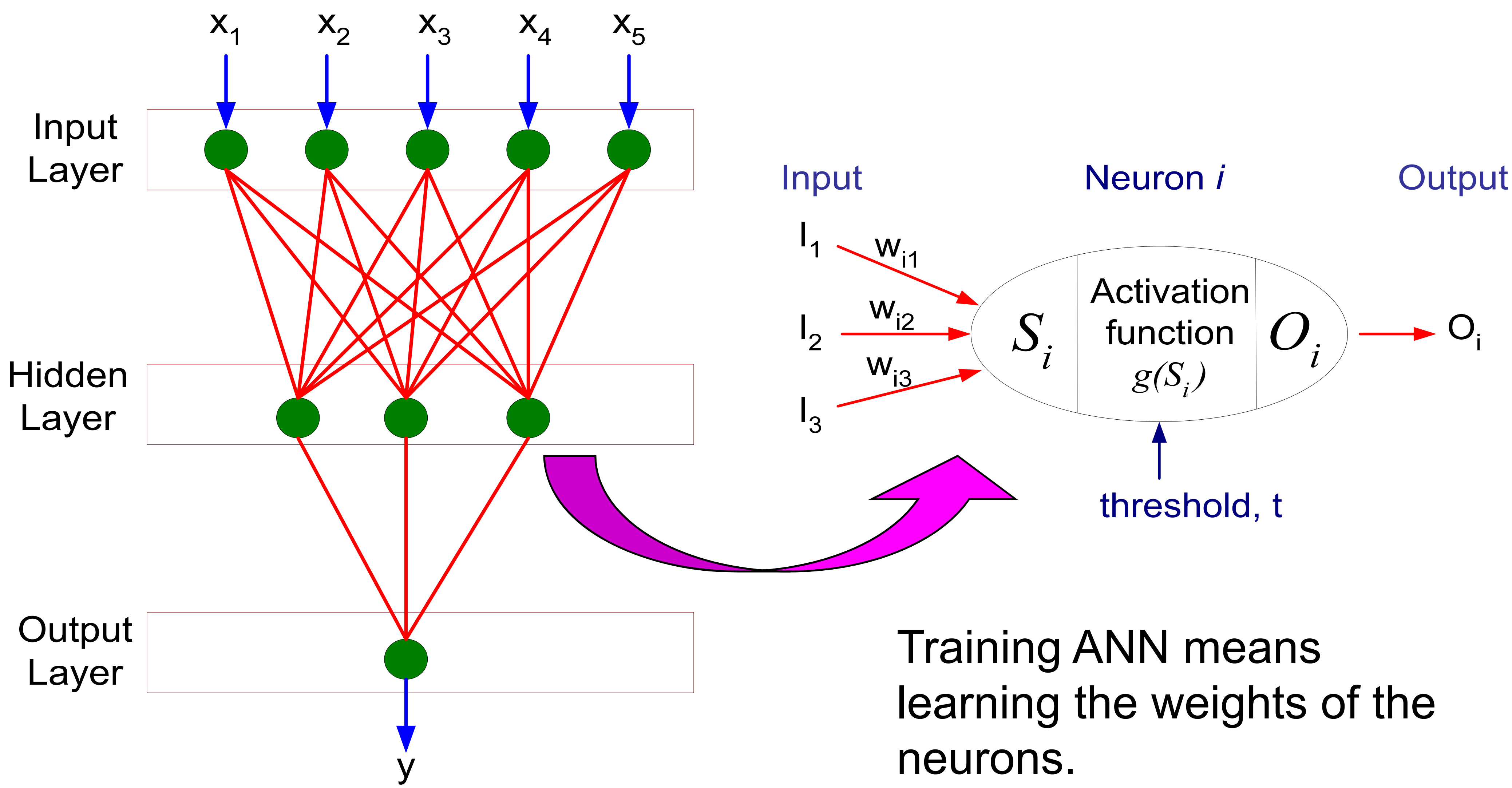


Perceptron Model

$$Y = I\left(\sum_i w_i X_i - t > 0\right) \quad \text{or}$$

$$Y = sign\left(\sum_i w_i X_i - t\right)$$

Multi-Layer Artificial Neural Networks



Training ANN means learning the weights of the neurons.

Algorithm for Training ANNs

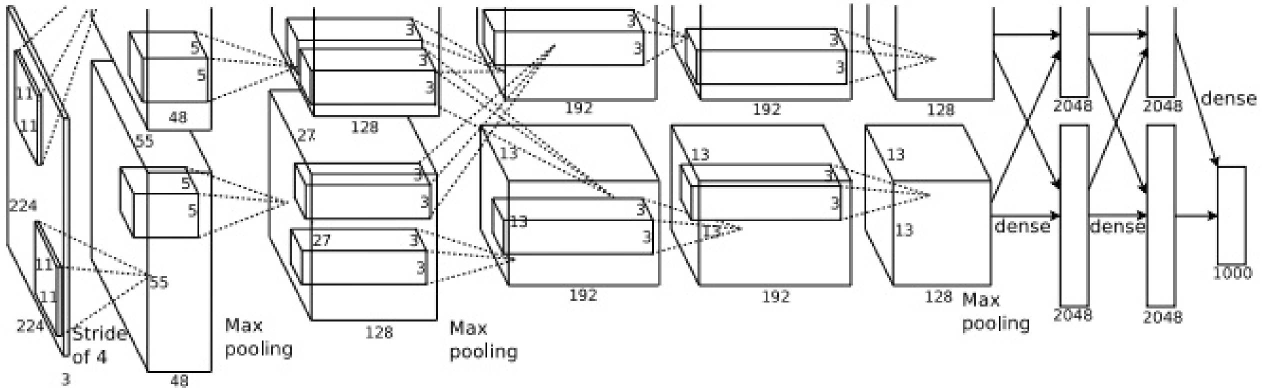
1. Initialize the weights (w_0, w_1, …, w_k), e.g., all with 1 or random

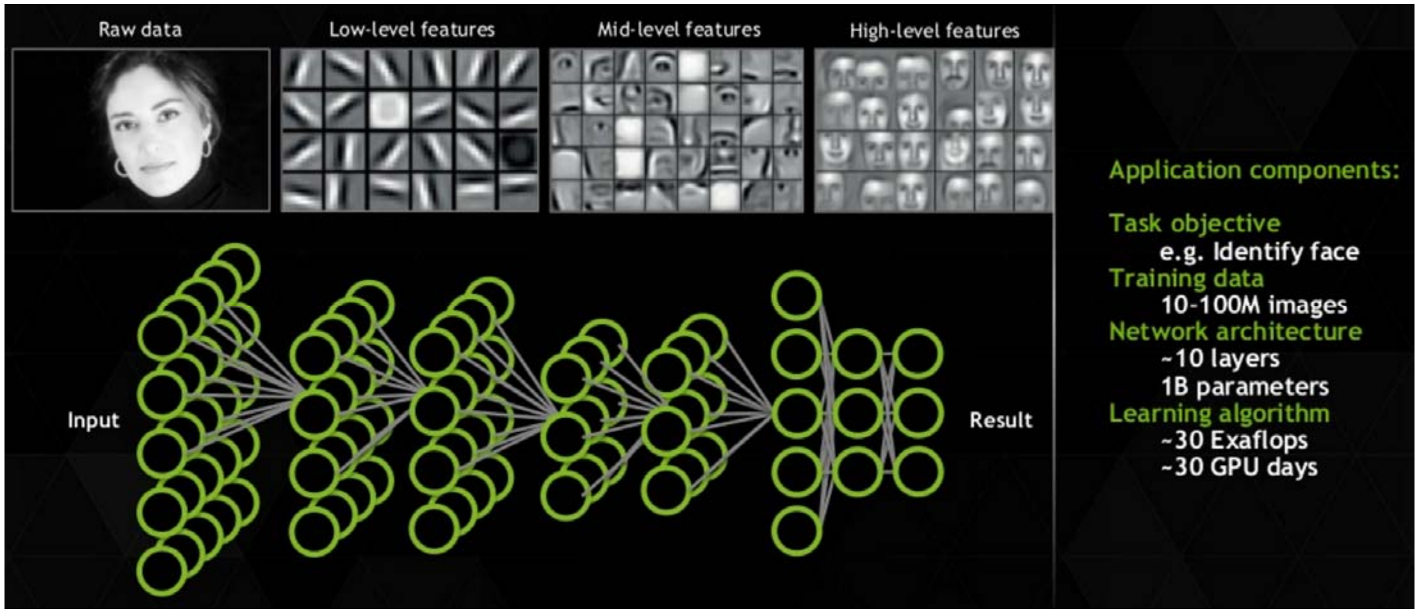
2. Adjust the weights in such a way that the output of ANN is as consistent as possible with class labels of the training examples

   - Objective function: $$E = \sum_i \left[Y_i - f(w_i, X_i) \right]^2$$
   - Find the weights w_i's that minimize the error E
   - using for example the **back propagation algorithm** (see Tan Steinbach, Chapter 5.4)

Deep Neural Networks (DNN)

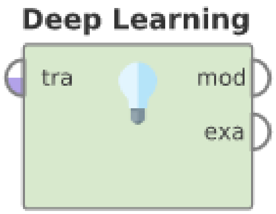
– Hype topic as Google successfully uses DNN for
  - computer vision
  - speech recognition
  - NLP
– Require
  - lots of training data
  - GPUs to calculate weights
– Details
  - Data Mining II
  - Lecture on 18th of March





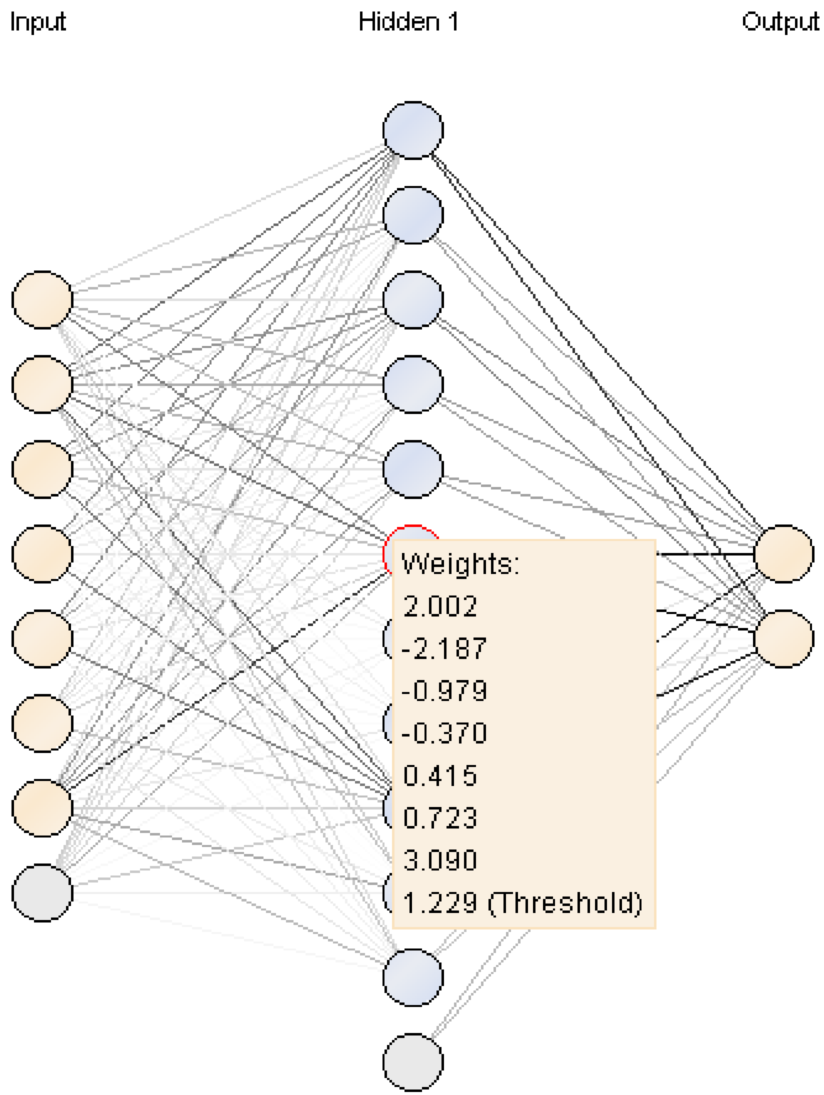
Source: NVIDIA

Artificial Neural Networks in RapidMiner



RapidMiner offers three alternatives:

1. RapidMiner's own **Neural Net** operator
2. wrapper for **H2O** Deep Learning
3. wrapper for **Keras** (Python) as separate extension



Characteristics of Artificial Neural Networks

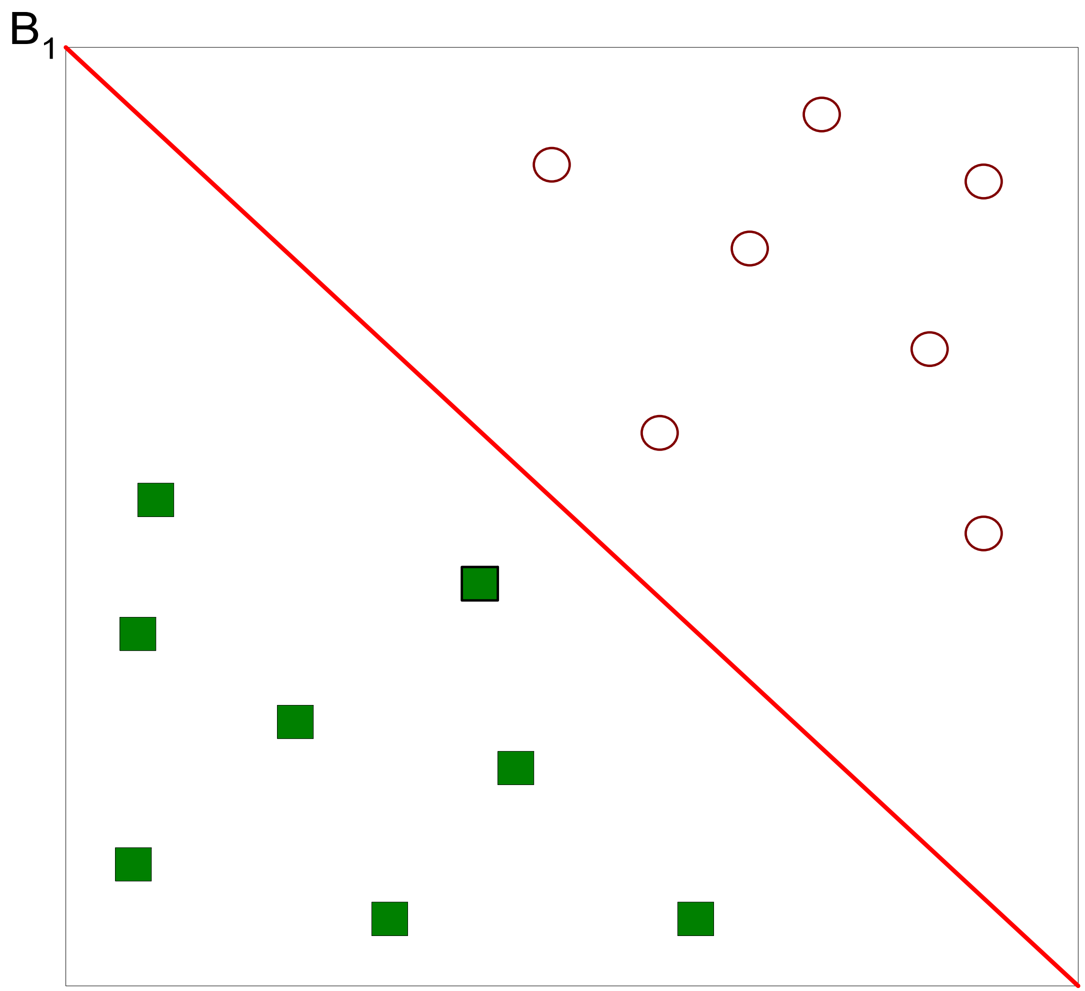
- ANNs can be used for classification as well as numerical regression tasks (more on this next week)
- Multi-layer neural networks are universal approximators
  - meaning that the can approximate any target function
- Very important but difficult to choose the right network topology
  - Expressive hypothesis space often leads to overfitting
  - Possible approaches to deal with overfitting:
    - Use more training data (a lot more might be necessary)
    - Step-by-step simplify the topology (also called regularization)
      1. Start with several hidden layers and larger number of nodes
      2. Estimate generalization error using validation dataset
      3. Step by step remove nodes as long as generalization error improves
- Model building can be time consuming, model application is fast

8. Support Vector Machines

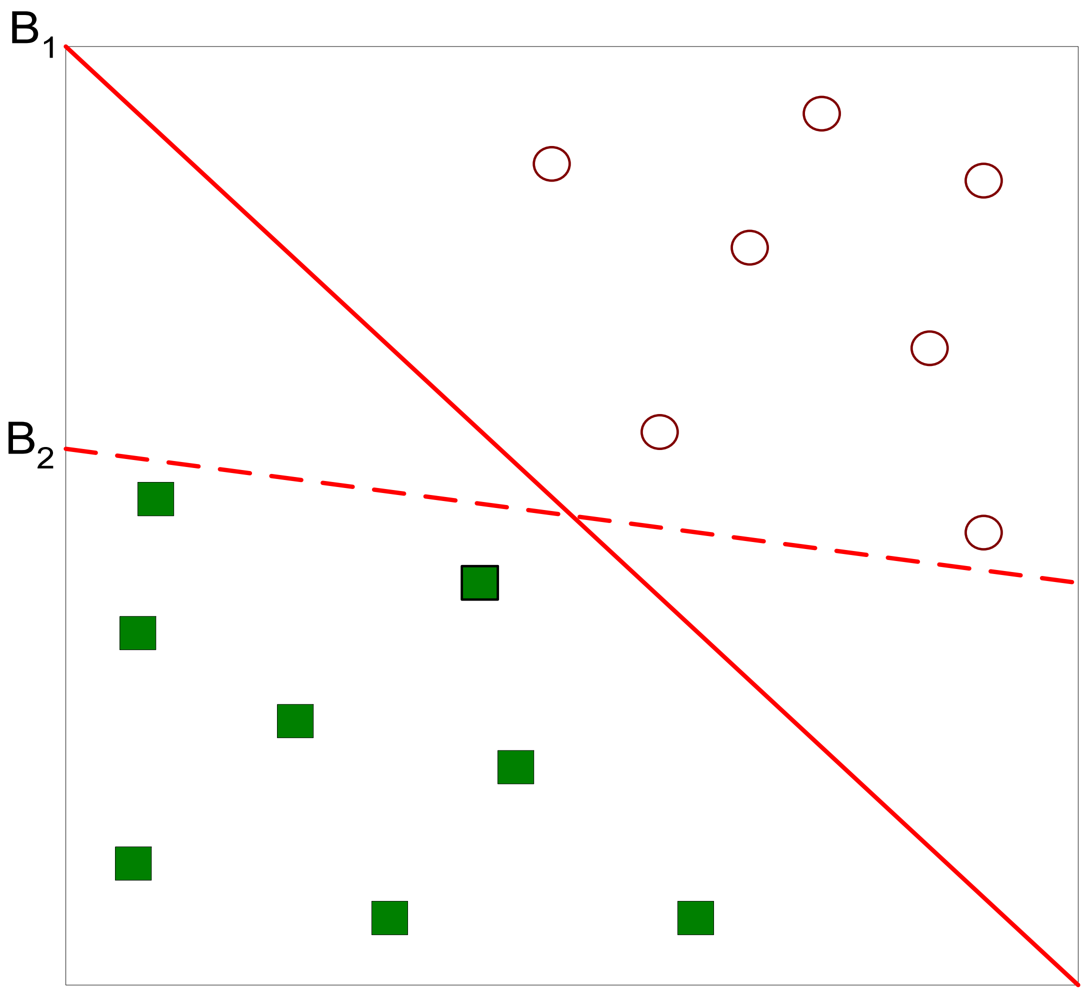
- Support vector machines (SVMs) are algorithms for learning linear classifiers for
  - two class problems (a positive and a negative class)
  - from examples described by continuous attributes.
- SVMs achieve very good results especially for high dimensional data.
- SVMs were invented by V. Vapnik and his co-workers in 1970s in Russia and became known to the West in 1992.

Support Vector Machines

– SVMs find a linear hyperplane (decision boundary) that will separate the data.



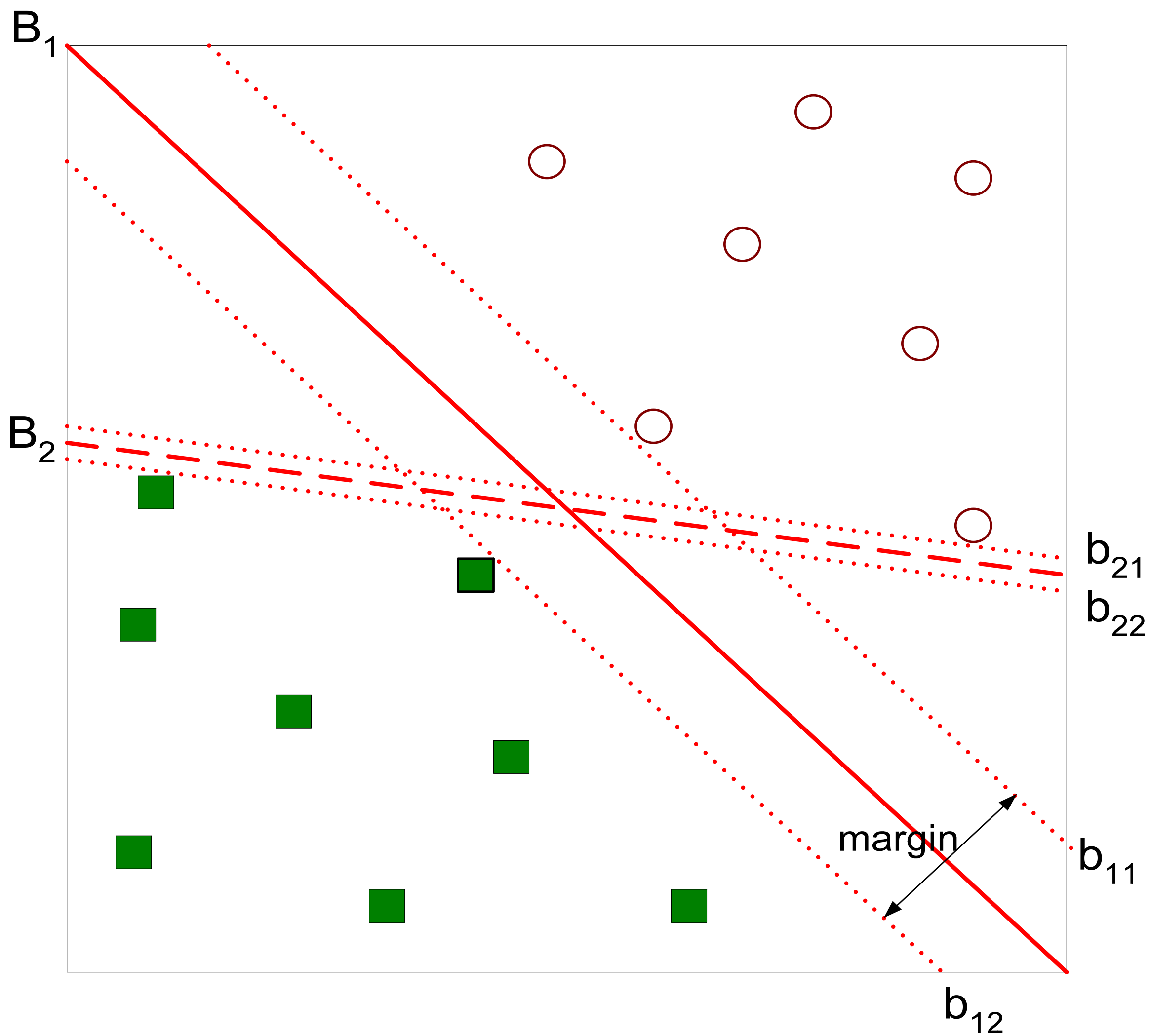
Support Vector Machines



– Which one is better? B1 or B2?
– How do you define “better”?

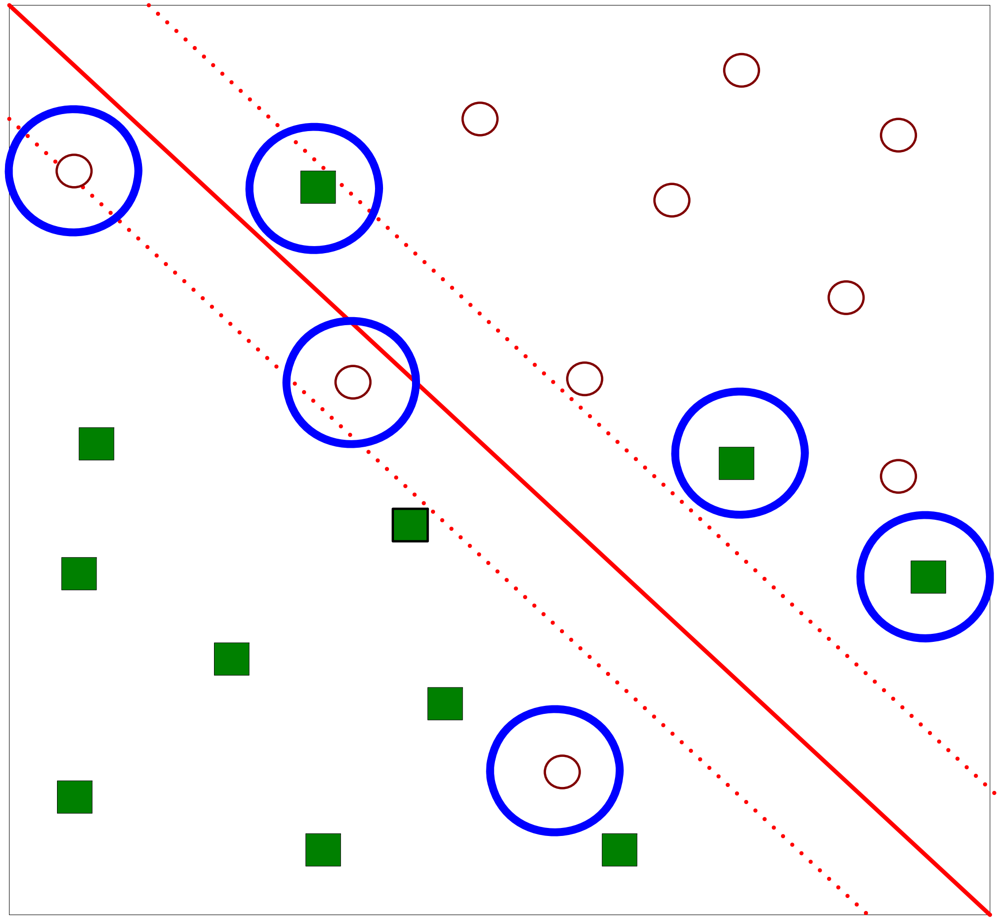
Which Hyperplane is better?

- In order to avoid overfitting and to generalize for unseen data, SVMs find the hyperplane that maximizes the margin to the closest points (support vectors).
- Visual solution:
  - B1 is better than B2
- Mathematical solution:
  - *Constrained optimization* that can be solved using quadratic programming.
  - See Tan/Steinbach/Kumar, Chapter 5.5



Dealing with Not Linearly Separable Data

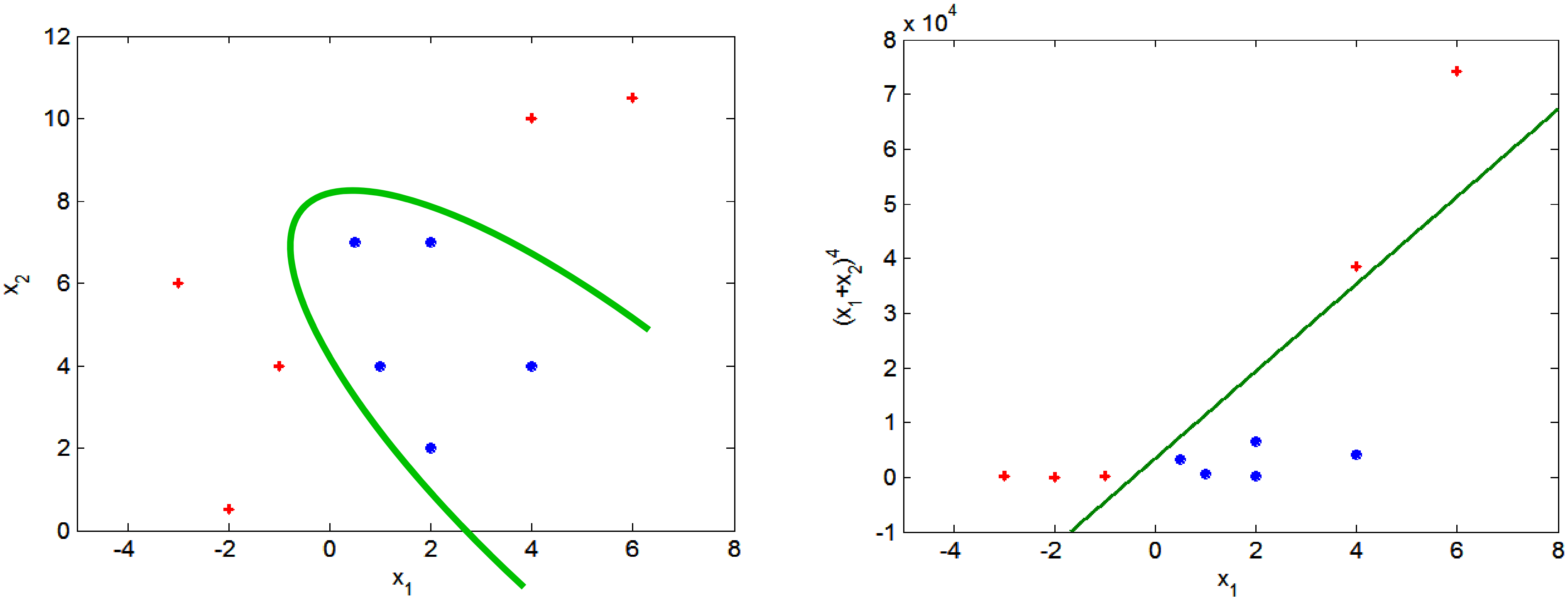
– What if the problem is not linearly separable due to noise points?



– Solution: Introduce slack variables in margin computation which result in a penalty for each data point that violates decision boundary.

Dealing with Non-Linear Decision Boundaries

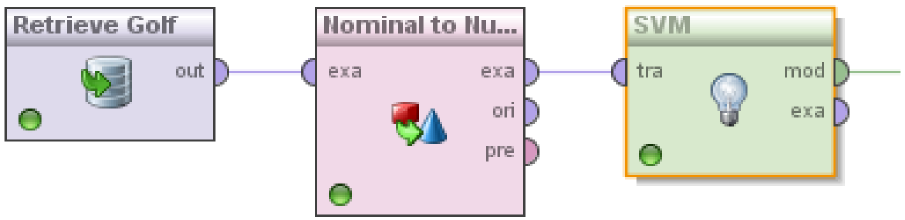
– Problem: What if decision boundary is not linear?
– Solution: Transform data into higher dimensional space where there is a linear separation
    - Details: Higher mathematics (see Tan/Steinbach, Chapter 5.5)
    - Different types of kernel functions are used for this transformation



Characteristics of Support Vector Machines

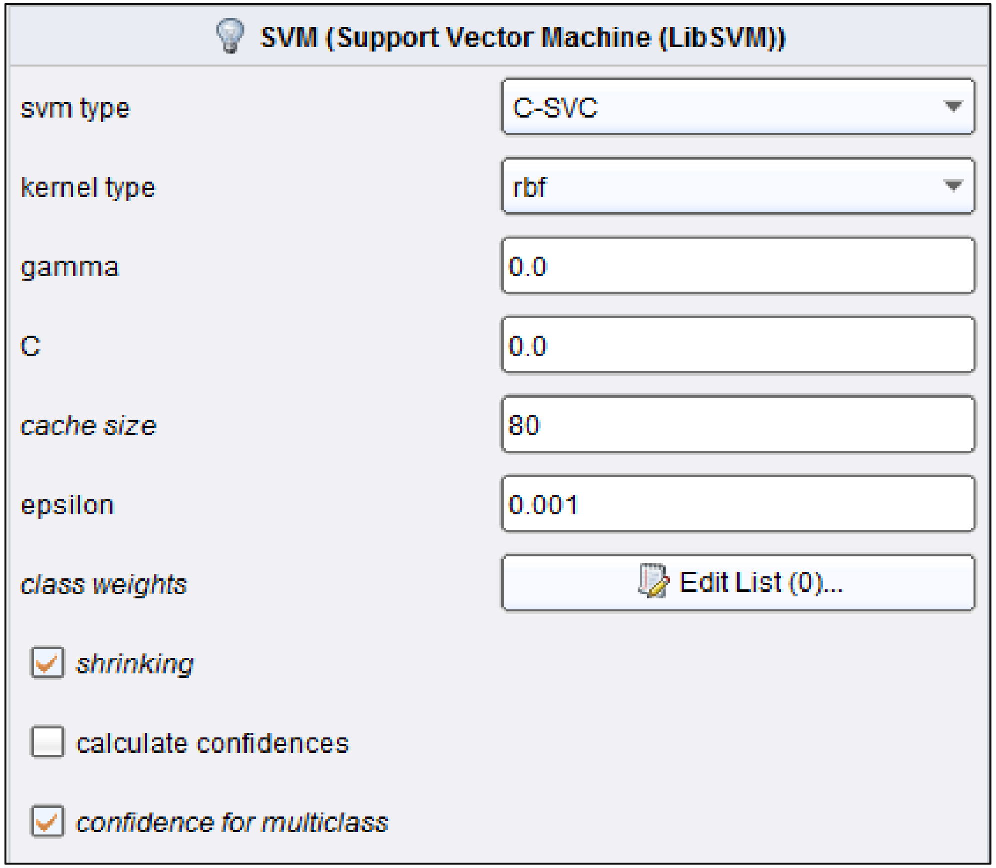
- SVMs were often the most successful classification technique for high dimensional data before DNNs appeared
- Application areas of SVMs include
  - Text classification
  - Machine vision, e.g face identification
  - Handwritten digit recognition
  - SPAM detection
  - Bioinformatics
- Parameter tuning often has a high impact on the performance of SVNs
  - see next slide

SVMs in RapidMiner



Tuning a SVM

1. Transform all attributes to numeric scale (Operator: Nominal to Numeric)
2. Normalize all value ranges to [0,1] (Operator: Normalize)
3. Use the RBF kernel function
4. Use nested cross-validation to find the best values for the parameters
    1. C = weight of slack variables (Range: 0.03 to 30000)
    2. gamma = kernel parameter (Range: 0.00003 to 8)



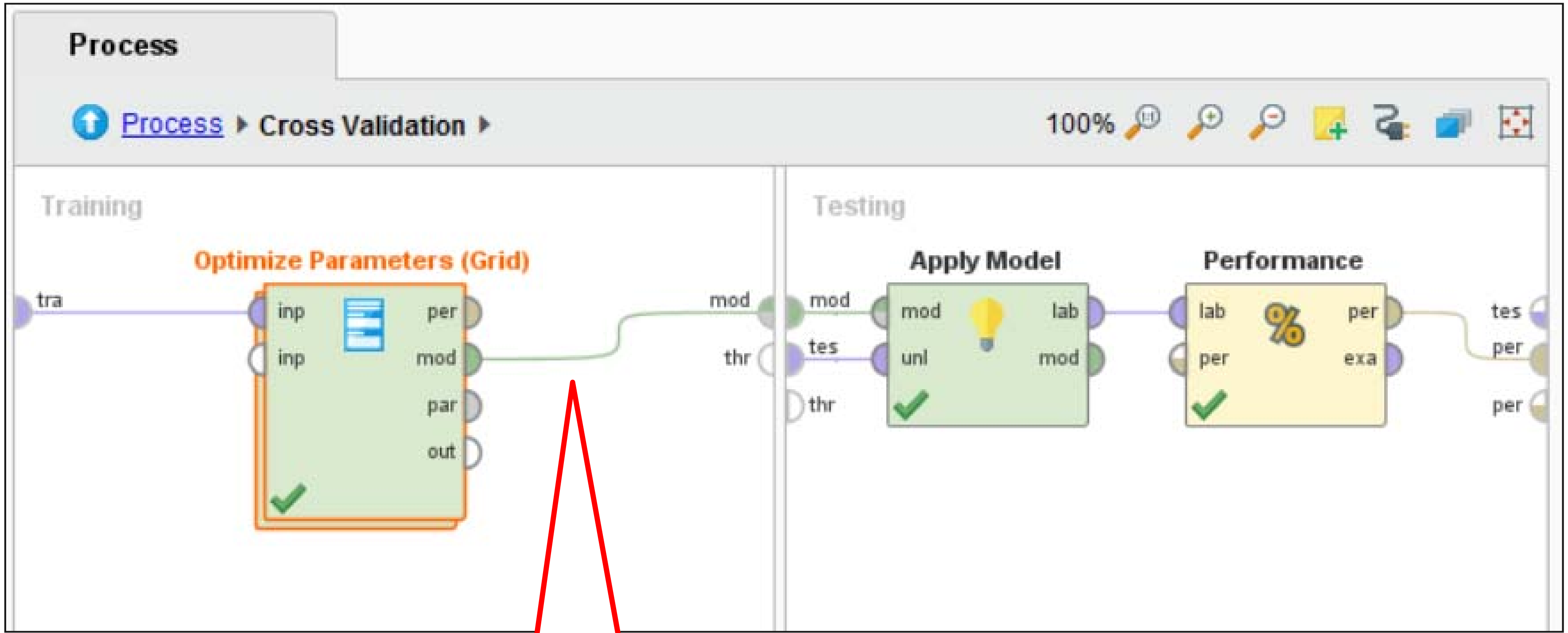
More details on the tuning procedure, see: Hsu, et al: A Practical Guide to Support Vector Classification. 2010.

9. Parameter Tuning

- Many learning methods require parameters
  - k for k-nearest-neighbors
  - pruning thresholds for trees and rules
  - hidden layers configuration for ANN
  - gamma and C for SVM
- Some methods often work rather poorly with default parameters
- How to determine the optimal parameters?
  - Play around with different parameters yourself
  - Alternative: Let your data mining tool test different parameter settings for you

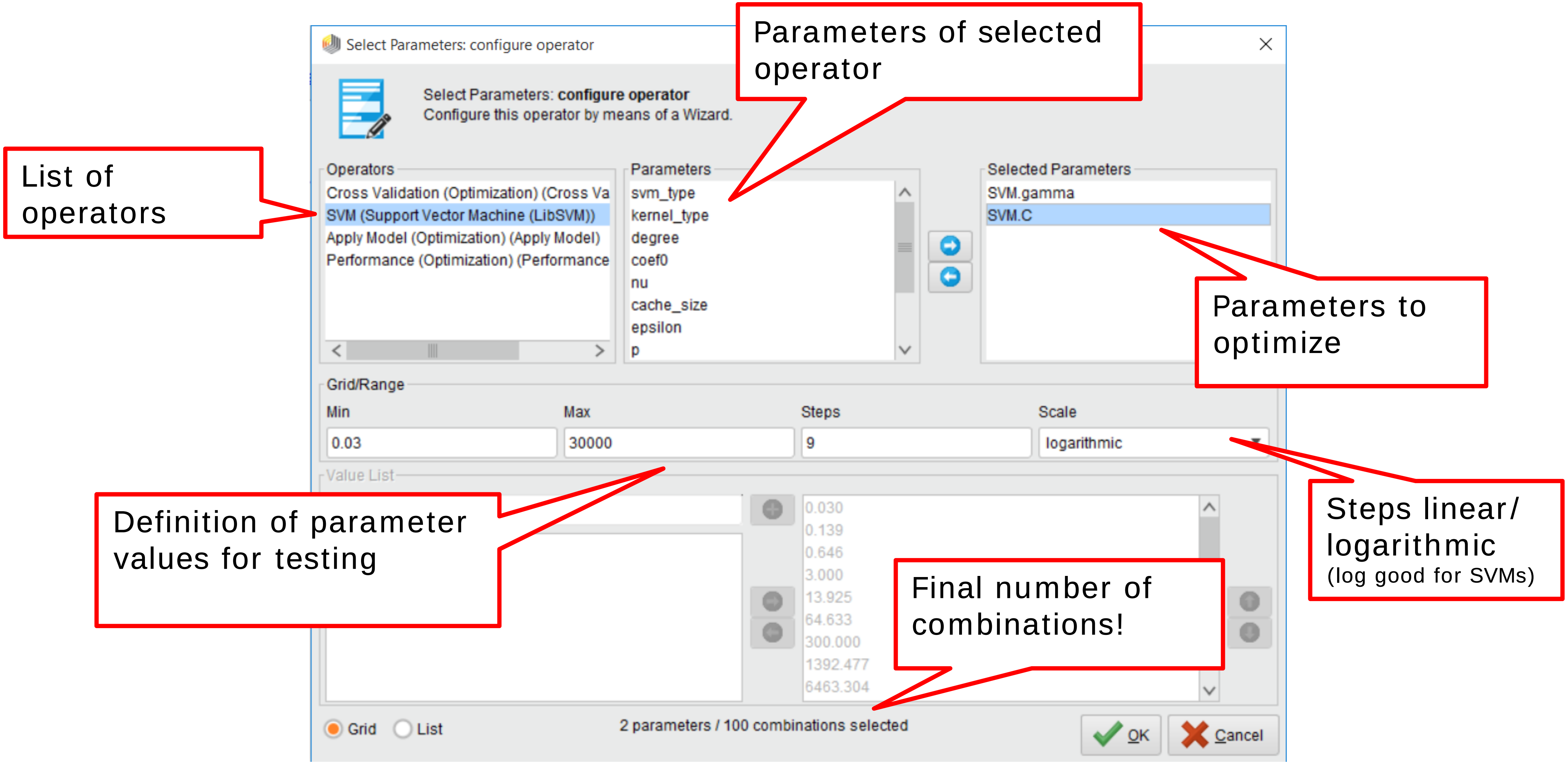
Parameter Optimization

The Optimize Parameters operator allows you to automatically test various parameter combinations.



Model learned using the best parameter setting

Parameter Optimization



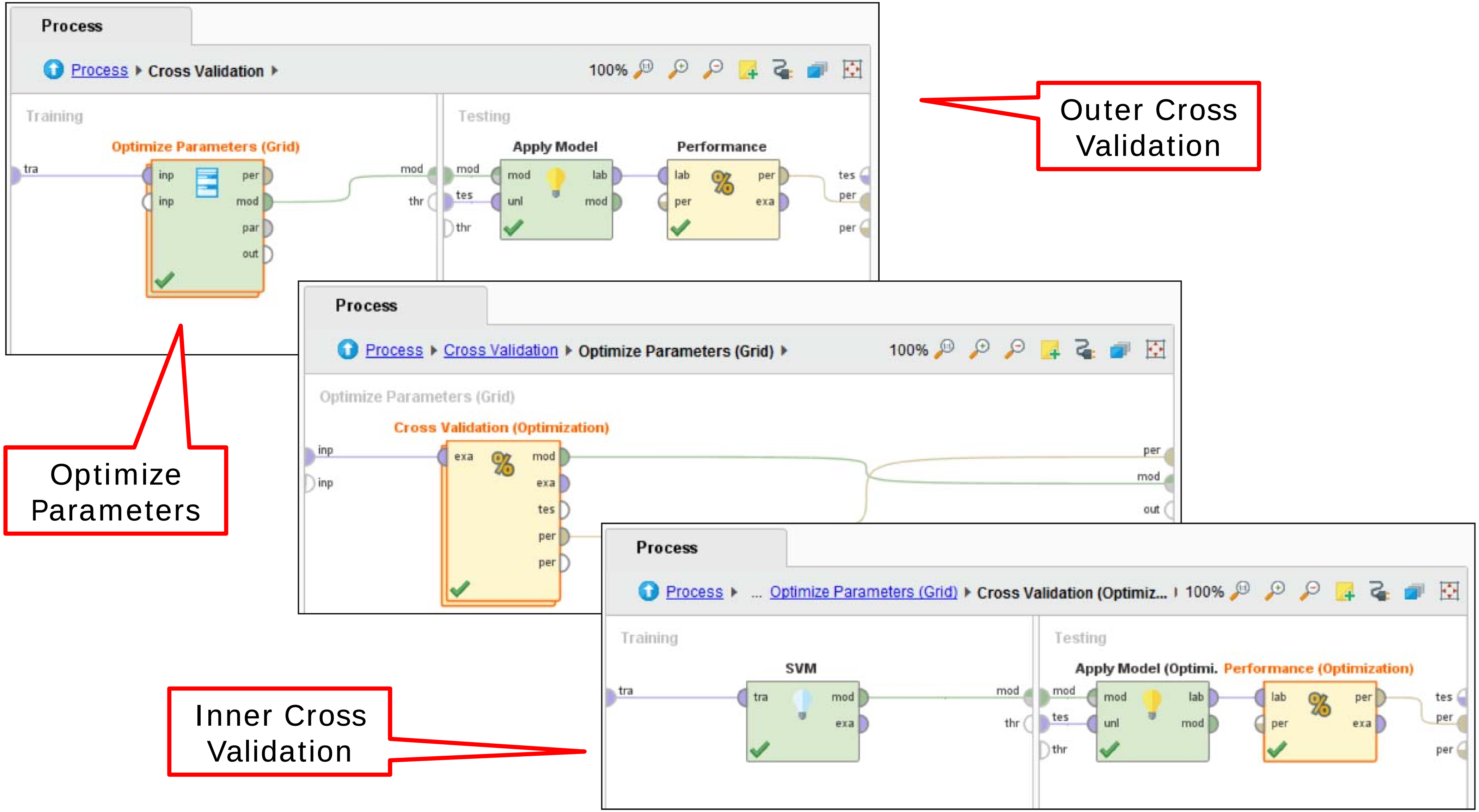
Alternative optimization approaches that are more efficient than brute force grid search are beam search and evolutionary algorithms.

Parameter Optimization and Generalization Error

- keeping training and test set strictly separate is crucial in order to not estimate the generalization error too optimistic
- Wrong: Parameter optimization using inner cross-validation
  - parameter value “leaks” information about test set into the model
  - model overfits training and test data ➔ higher error on unseen data
- Right: **Nested Cross-Validation**
  - Outer Cross-Validation estimates generalization error using test set
  - Inner Cross-Validation
    - estimates optimal parameter setting using cross-validation to split training set from outer cross-validation into training and validation set.
    - Once the optimal parameters are found, a model is learned with these parameters using the complete outer training data.
  - Result: Good estimate of generalization error as no information from test set is leaked into parameter setting.

https://scikit-learn.org/stable/auto_examples/model_selection/plot_nested_cross_validation_iris.html
https://rapidminer.com/resource/correct-model-validation/

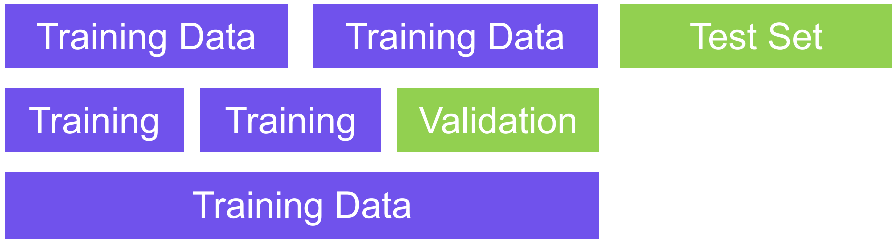
Nested Cross-Validation for Parameter Optimization



What If Nested Cross-Validation Gets Too Slow?

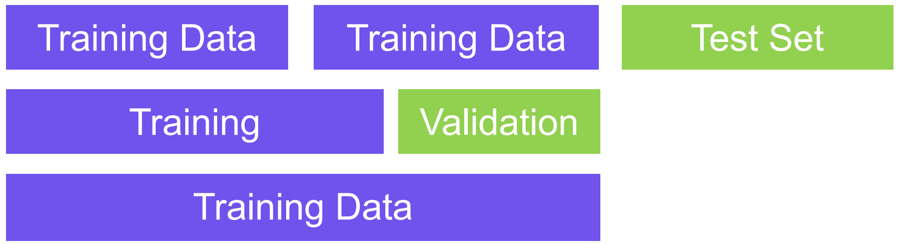
– Nested Cross-Validation

- Outer X-Val: 3 times
- X-Val parameters: 3 * 30 times
- Learn using best parameters: 1
- 3*(3*30+1) = **273 models learned**



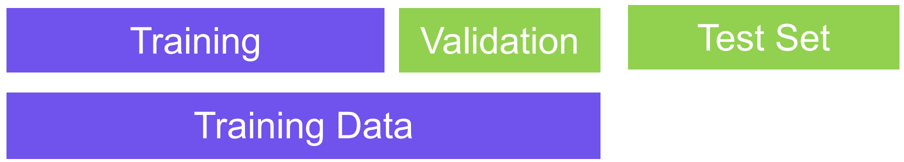
– X-Val with Inner Hold-Out Validation

- Outer X-Val: 3 times
- Parameter optimization: 30 times
- Learn using best parameters: 1
- 3*(30+1) = **93 models learned**



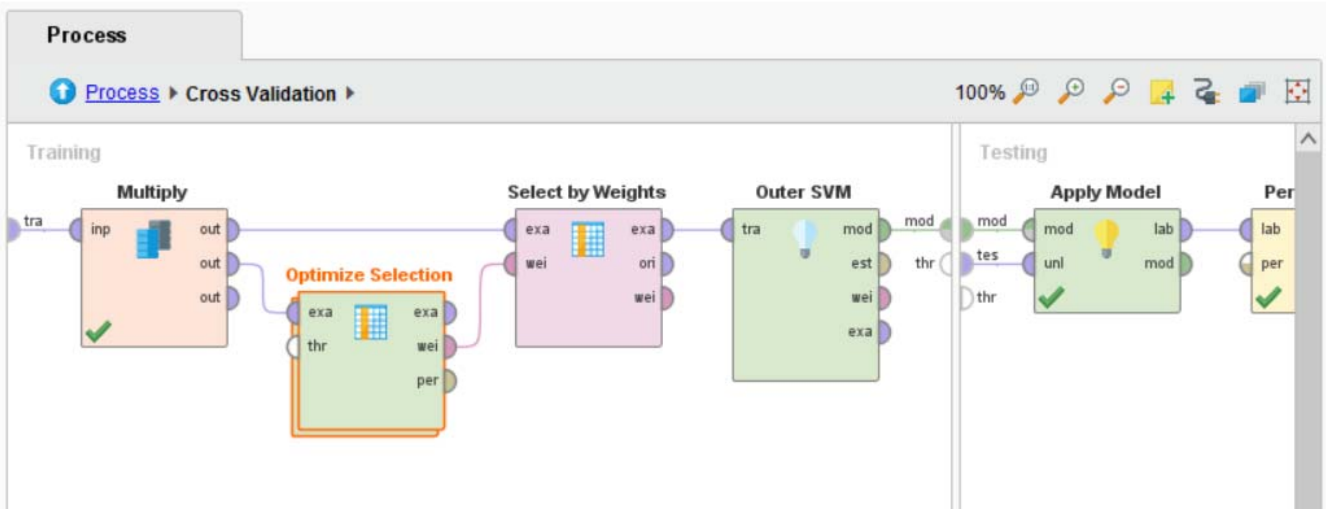
– Parameter Optimization using Hold-Out Validation

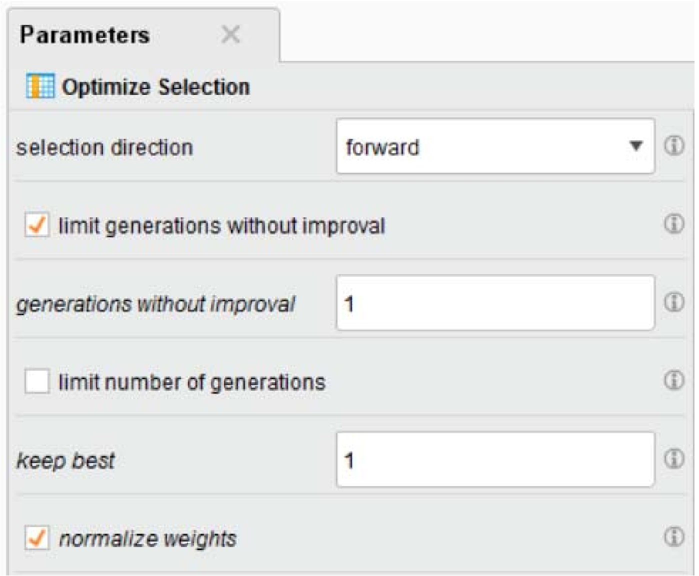
- Parameter Search: 30
- Learning using parameters: 1
- *30+1 = **31 models learned**



Attribute Subset Selection

The Optimize Selection operator allows you to find the optimal subset of the available attributes.

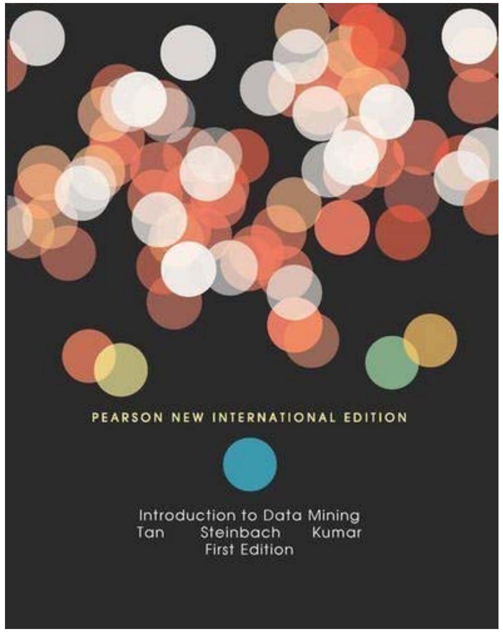




Forward selection: Find best single attribute, add further attributes, test again
Backward selection: Start with all attribute, remove attributes, test again

Literature and Video References for this Slideset

Pang-Ning Tan, Michael Steinbach, Vipin Kumar:
Introduction to Data Mining.
Pearson / Addison Wesley.



Chapter 5.3: Naïve Bayes

Chapter 5.4: Artificial Neural Networks

Chapter 5.5: Support Vector Machines

Videos

- Parameter Optimization
  http://www.youtube.com/watch?v=R5vPrTLMzng
- Attribute Subset Selection
  - Part 1: http://www.youtube.com/watch?v=7IC3IQEdWxA
  - Part 2: http://www.youtube.com/watch?v=j5vhwbLIZWg