Metalearning in Multi-Agent Systems Designed for Data Mining

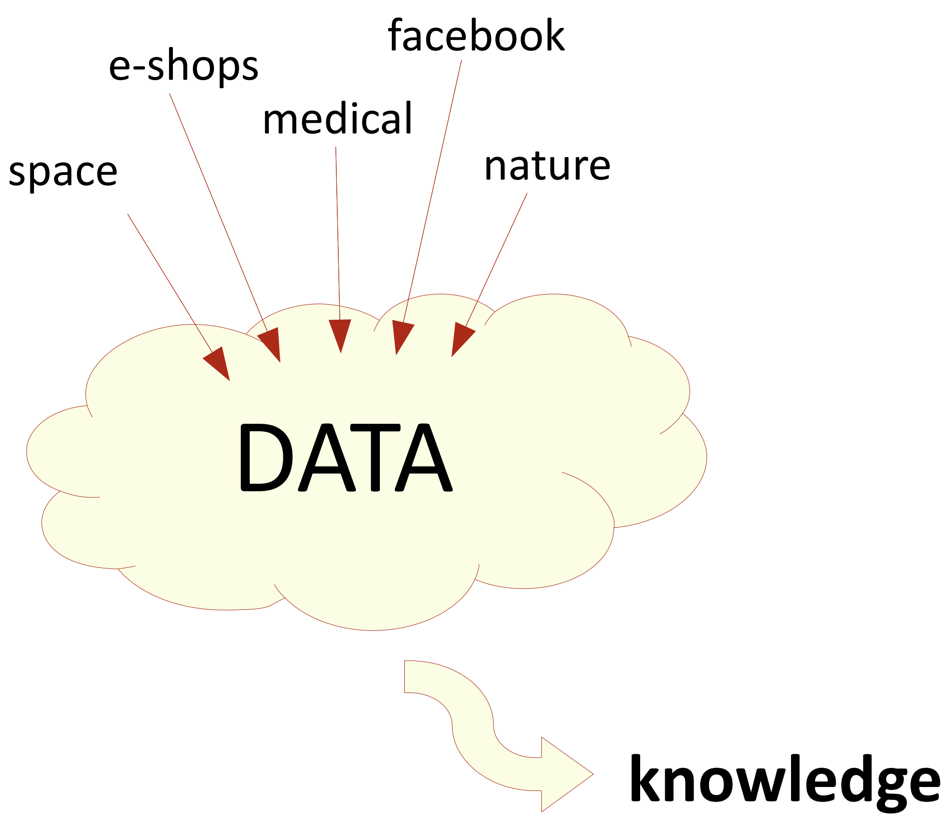
Klára Pešková

peskova@braille.mff.cuni.cz

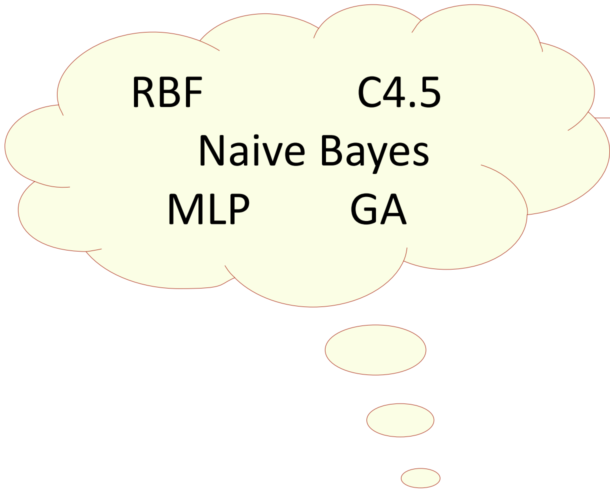
The Czech Academy of Sciences
Institute of Computer Science

Charles University in Prague, Faculty of Mathematics and Physics
Department of Theoretical Computer Science and Mathematical Logic

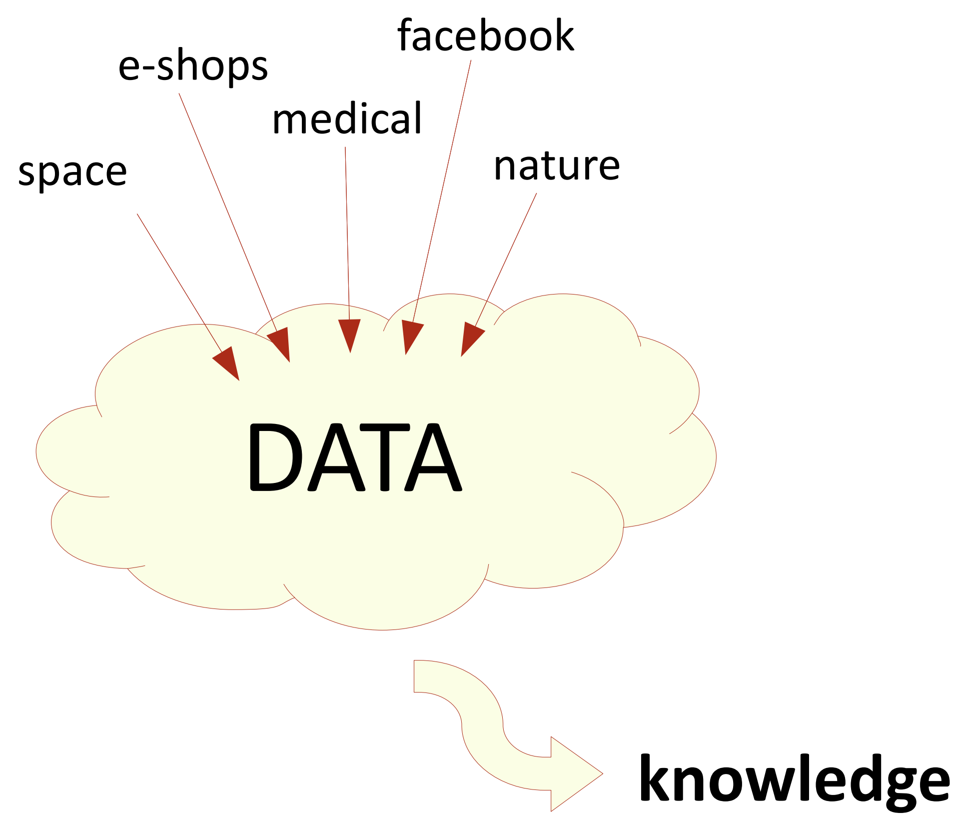
Motivation



Motivation

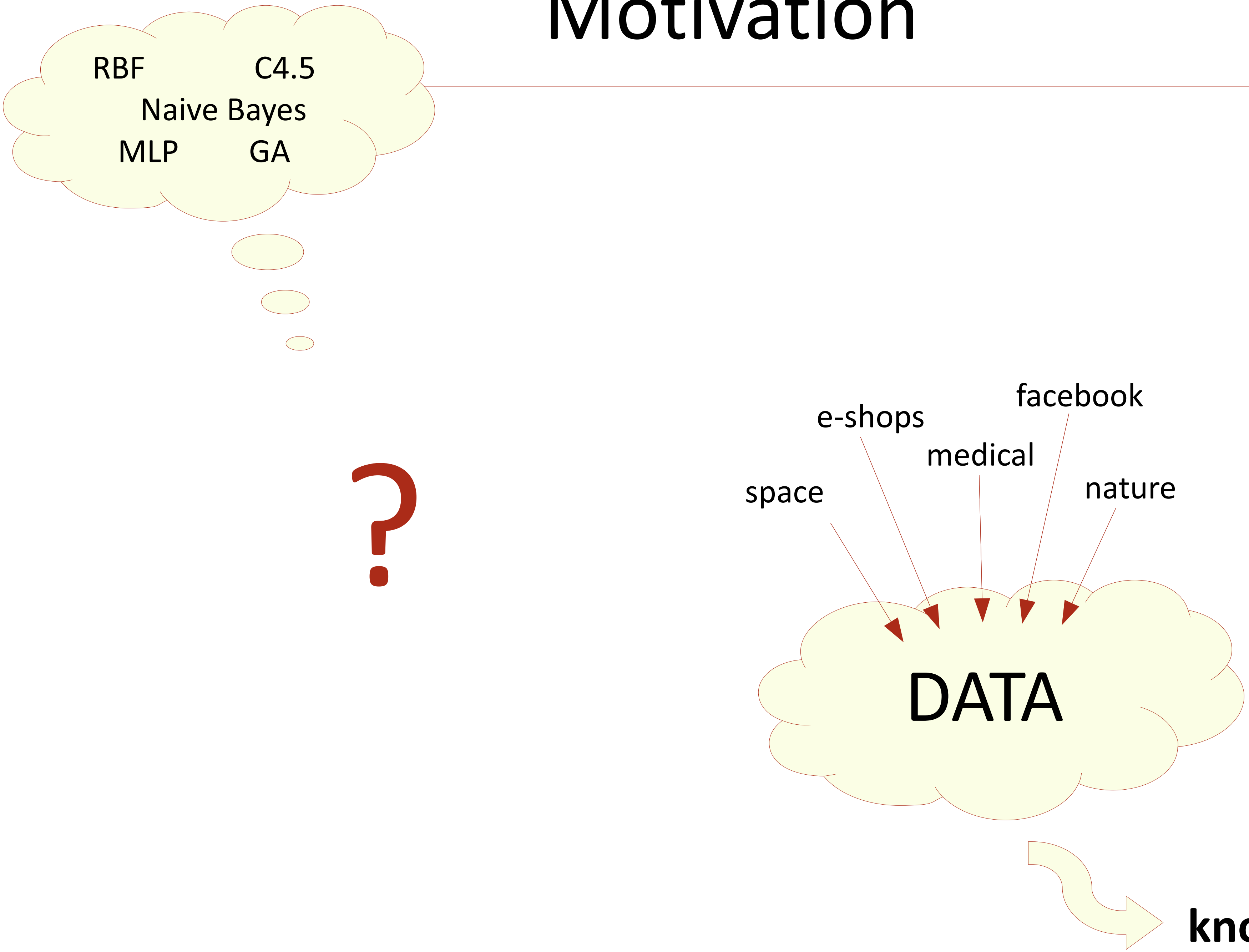


RBF
C4.5
Naive Bayes
MLP
GA



facebook
e-shops
medical
space
nature
DATA
knowledge

Motivation


RBF
C4.5
Naive Bayes
MLP
GA
?
facebook
e-shops
medical
space
nature
DATA
knowledge

Motivation

RBF C4.5
Naive Bayes
MLP GA

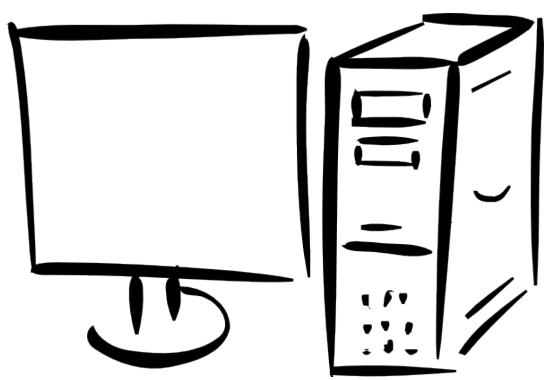
I trained ***THOUSANDS***
of neural networks!

space
e-shops
medical
facebook
nature

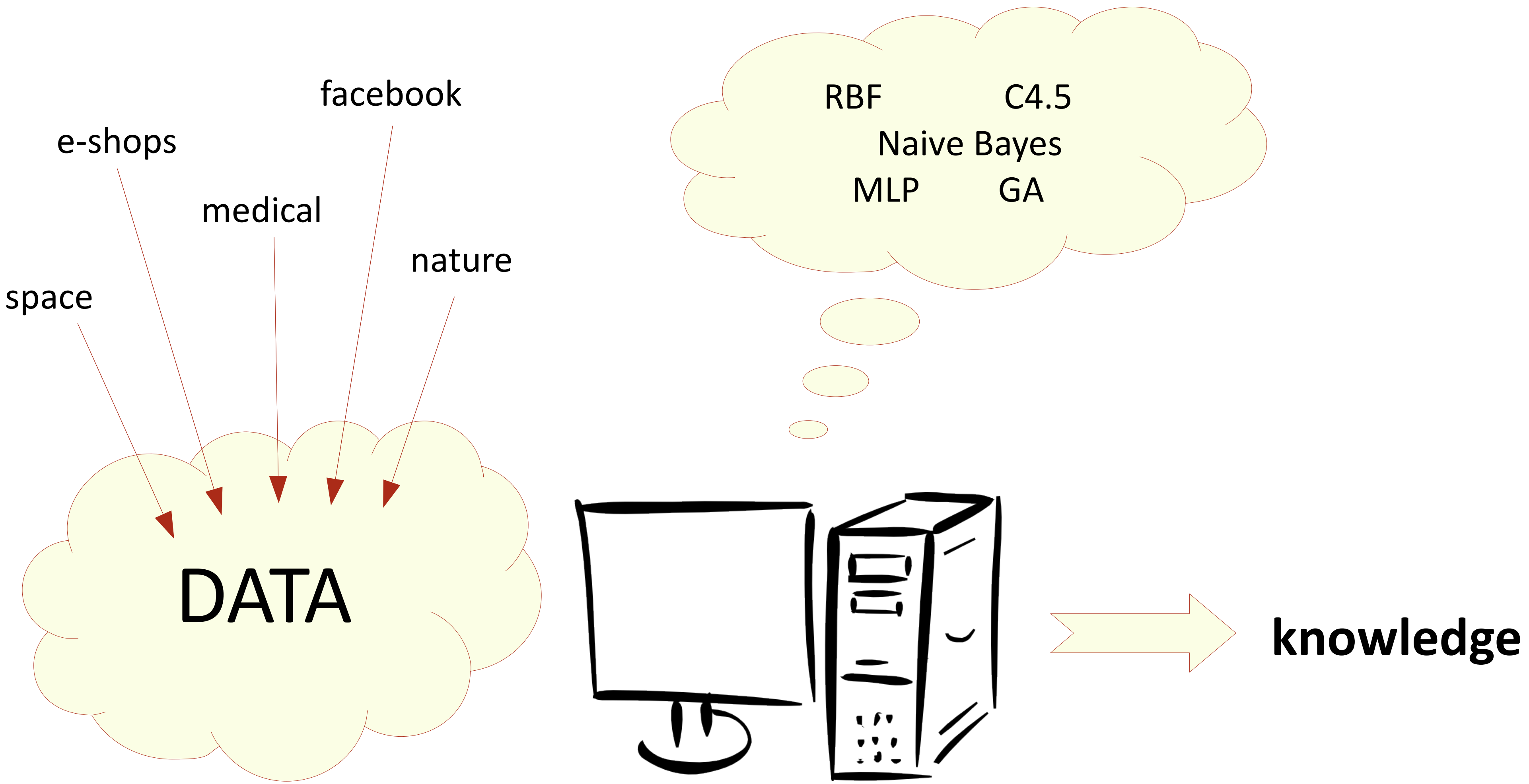
DATA

knowledge

Solution



Solution


facebook
e-shops
medical
nature
space
DATA
RBF
C4.5
Naive Bayes
MLP
GA
knowledge

Metalearning challenges

- **Recommendation of a suitable computational intelligence method for new datasets**

Metalearning challenges

- **Recommendation of a suitable computational intelligence method for new datasets**

- **Providing (the best) parameters for the chosen method**

Metalearning challenges

- **Recommendation of a suitable computational intelligence method for new datasets**

- **Providing (the best) parameters for the chosen method**

- **Generating workflow graphs**

Metalearning challenges

- **Recommendation of a suitable computational intelligence method for new datasets**
  - Metadata with suitable metrics
  - History of experiments (method, dataset, results)
  - Computational intelligence methods on metadata
- **Providing (the best) parameters for the chosen method**
- **Generating workflow graphs**

Metalearning challenges

- **Recommendation of a suitable computational intelligence method for new datasets**
  - Metadata with suitable metrics
  - History of experiments (method, dataset, results)
  - Computational intelligence methods on metadata
- **Providing (the best) parameters for the chosen method**
  - History of experiments
  - Search parameter spaces
- **Generating workflow graphs**

Metalearning challenges

- **Recommendation of a suitable computational intelligence method for new datasets**
  - Metadata with suitable metrics
  - History of experiments (method, dataset, results)
  - Computational intelligence methods on metadata
- **Providing (the best) parameters for the chosen method**
  - History of experiments
  - Search parameter spaces
- **Generating workflow graphs**
  - Standard approach: ontology based
  - Typed genetic programming

Agent-Based Approach

- requirements on system:
  - intelligent, autonomous behavior, distributed and parallel nature, extensibility
  - large number of methods, data, users...

- definition of *agent* (Wooldridge, 1995)*:
  - autonomy
  - social ability
  - reactivity
  - pro-activity

- other traits:
  - truthfull
  - mobile
  - ability to learn

* Wooldridge M., Jennings N. (1995) Intelligent agents: theory and practice. Knowledge Engineering Review 10 (2).

Communication Between Agents

- FIPA specification (Interaction Protocol, Communicative Acts*, Content Languages)
- ontology
  - formal representation of knowledge domain for automatic processing
    - concepts — types of objects
    - instances — instances of objects
    - attributes — properties of objects
    - restrictions — on attributes
    - relations — among objects
- extensibility, hybrid nature

* Austin, 1962, How to Do Things with Words

Our Data-Mining Multi-Agent System

- classification and regression tasks

- encapsulation of computational intelligence methods
- searching the parameter space
- recommendation
- metadata
- using previous experience

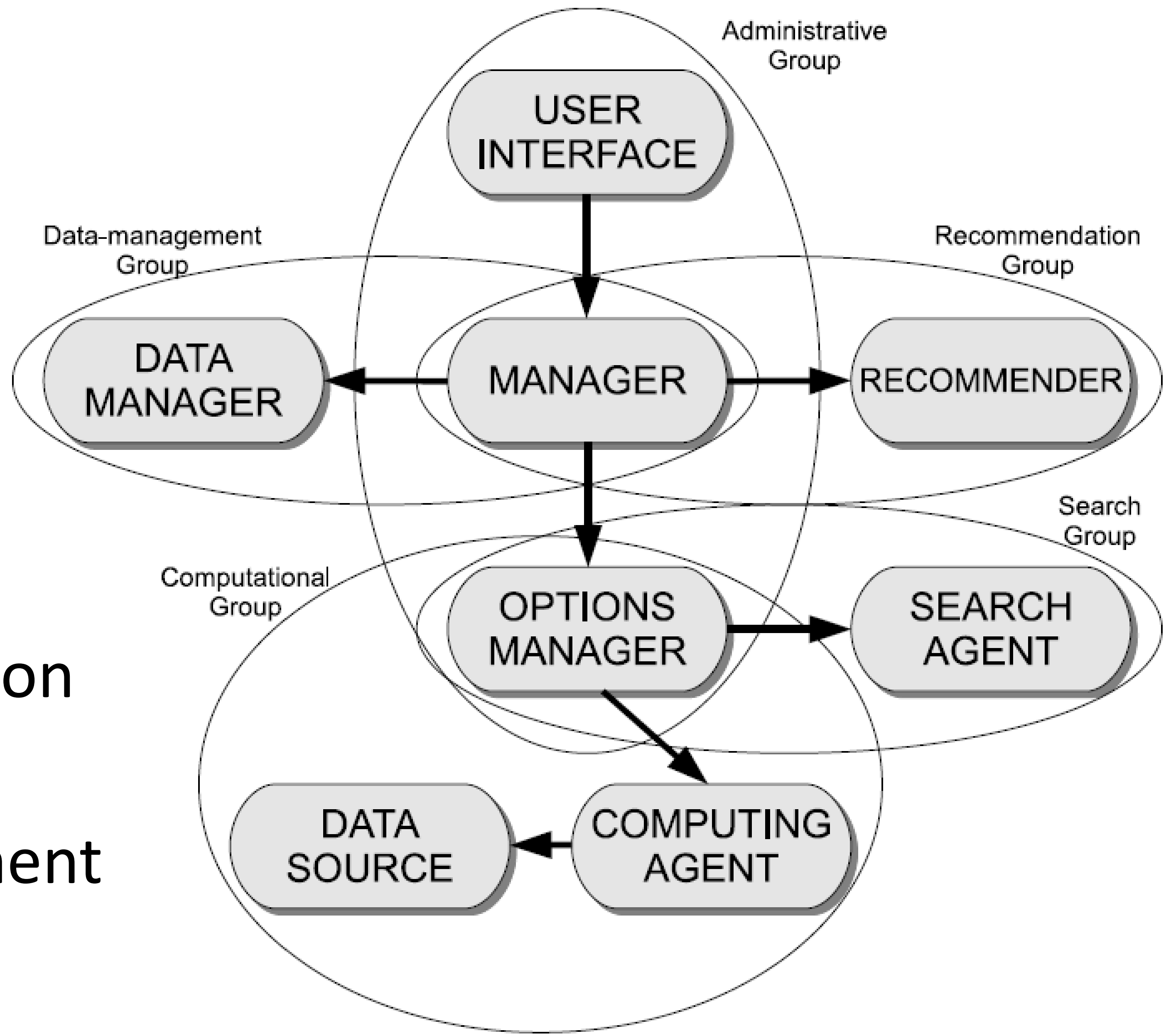
- JADE, Weka

Agent Group Role (AGR) Model

- group structures
- agent enters the group by playing a role from a group structure
- agents interact according to communication protocol defined for their roles
- agent can play more than one role

Group Structures

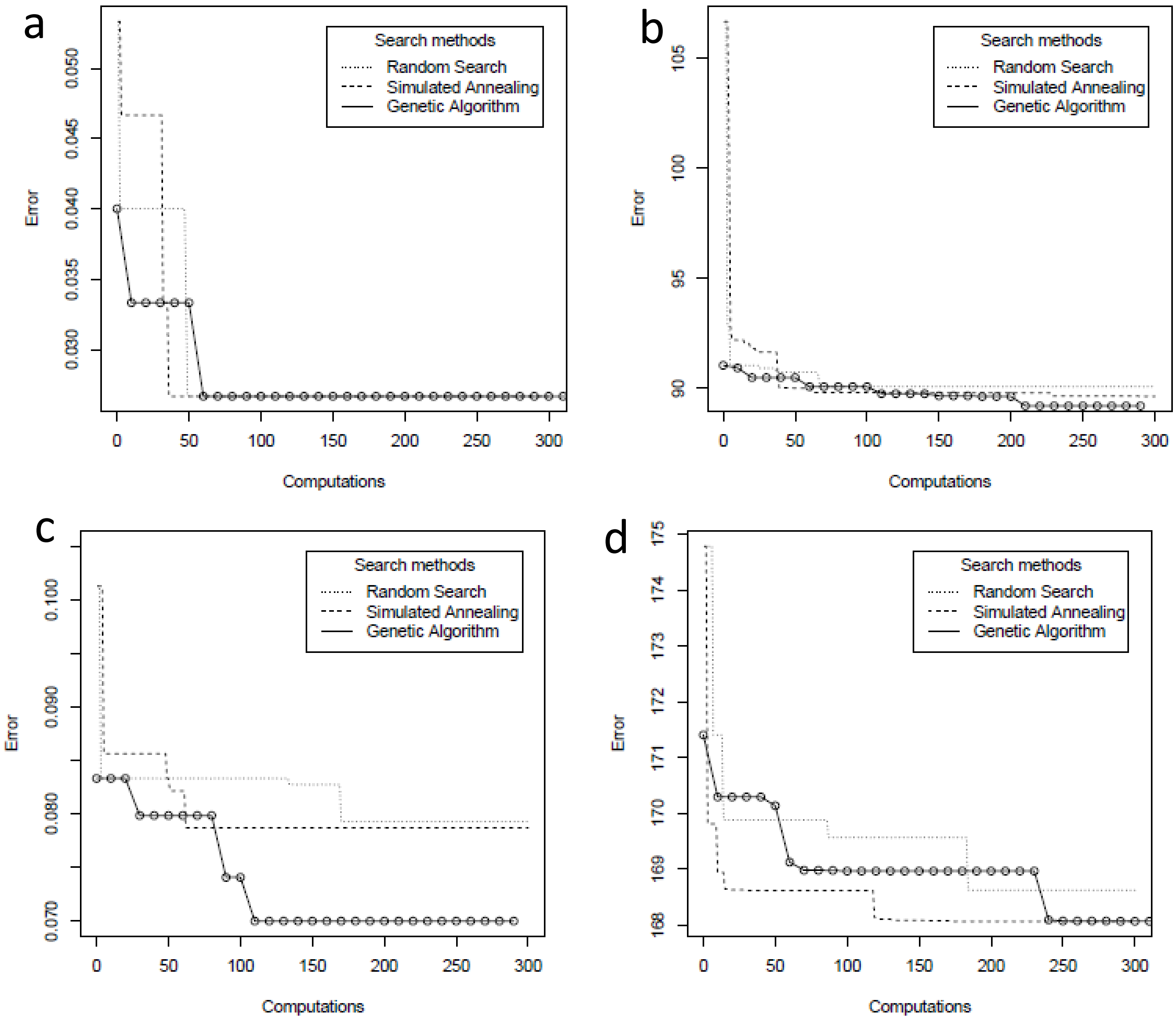
- Administrative
- Computational
- Search
- Recommendation
- Data-management



Searching the parameter space

- goal: to optimize the parameters of the method
- side-effect: database of experiments with different parameters settings -> can be used for recommendation
- methods used:
  - random search
  - grid search
  - simmulated annealing
  - genetic algorithm

Results



a) RBF network, iris.arff (4 attributes,
150 instances, classification)

b) RBF network, machine.arff dataset
(9 attributes, 209 instances, regression)

c) RBF network, car.arff (6 attributes, 1728 instances, classification

d) RBF network, wine.arff (13 attributes, 178 instances, regression)

Recommendation

- goal: to recommend a method for new dataset

- based on similarity of datasets
  - metadata
  - metrics

- single method / several methods – ranking
  - results of experiments

Metadata

Narrow

- categorical

- attributes:
  - number of attributes
  - number of instances
  - data type
  - missing values

Wide

- numerical, data complexity

- Simple measures:
  - categorical ratio
  - integer ratio
  - real ratio
  - ratio of missing values (of the two most significant attributes)
- Information theoretic measures

Wide metadata – Information Theoretic Measures

- discretization of real values

- Entropy of the two most significant attributes

$$H(X) = -\sum_{i} q_i \log_2 q_i ,$$

- Joint entropy of class and attribute → two most significant attributes

$$H(A,C) = -\sum_{v \in Val(A)} \frac{n(A(v))}{n} \sum_{t \in Val(C)} \frac{n_t(A(v))}{n(A(v))} \log_2 \frac{n_t(A(v))}{n(A(v))} ,$$

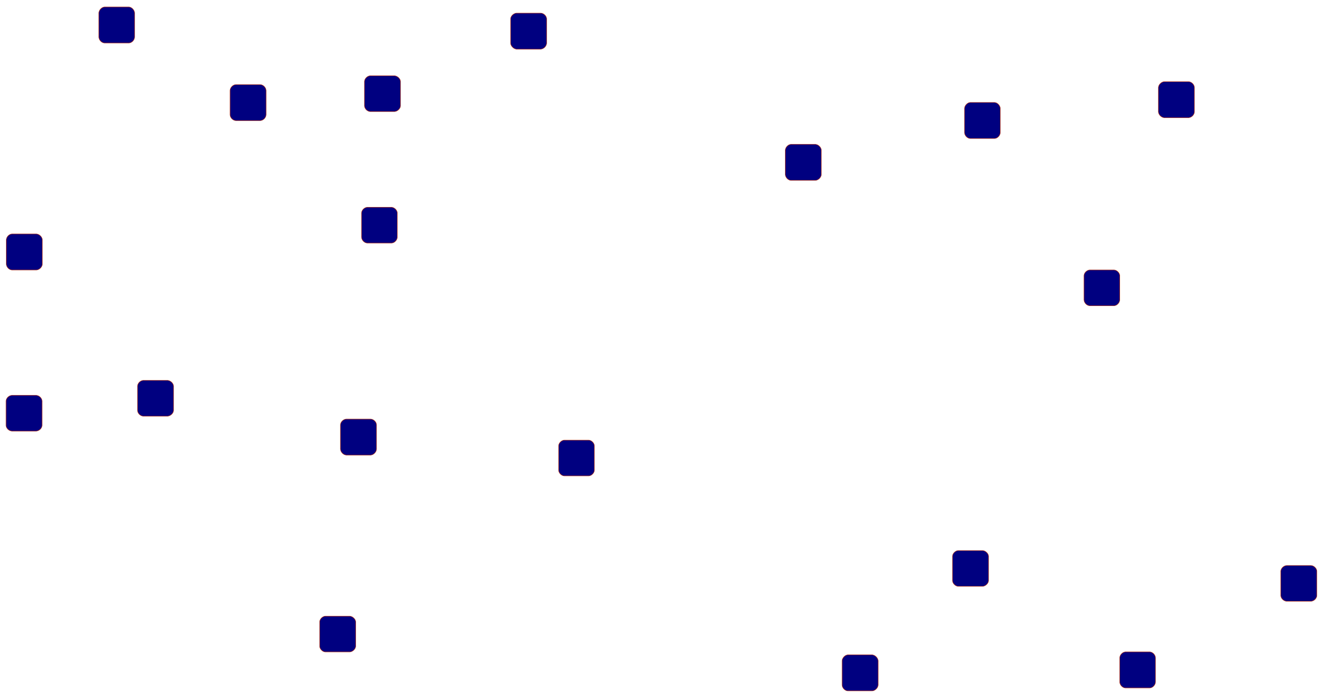
Metric

- distance between datasets - weighted sum of distances between metadata attributes

$$d(m_1, m_2) = \sum_{i=1}^{n} w_i \cdot d_i(m_1[i], m_2[i])$$

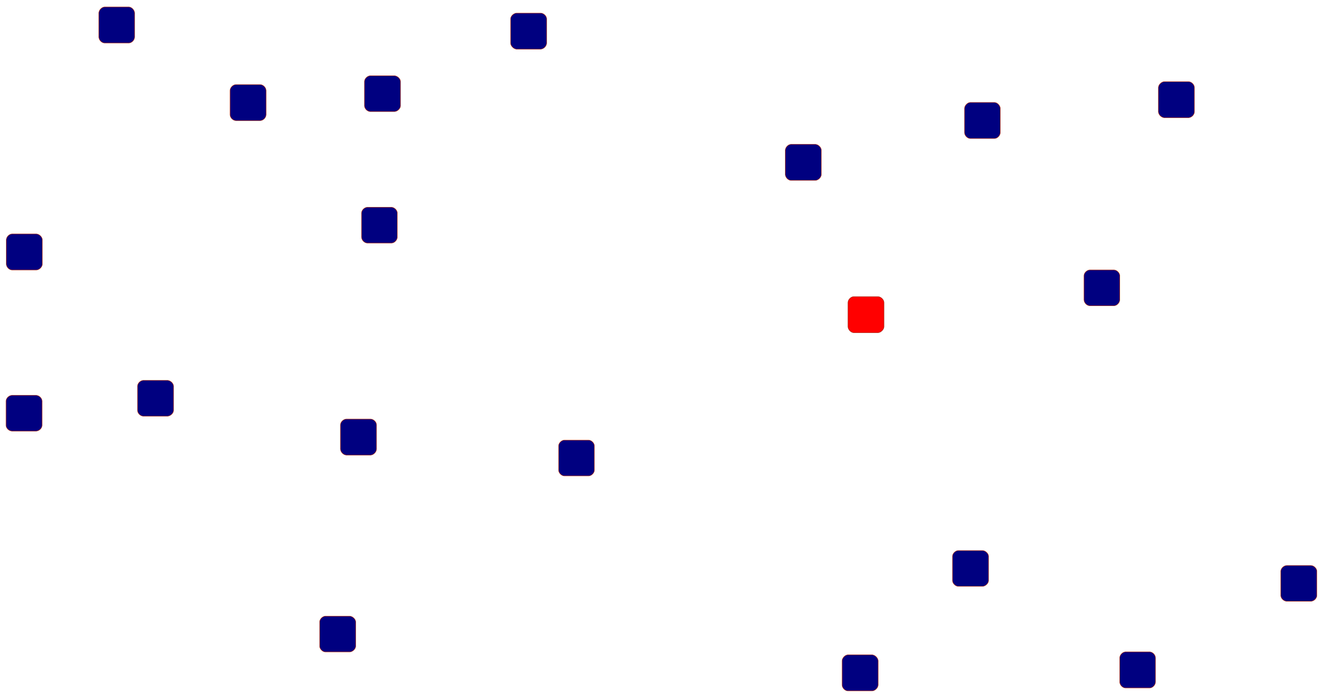
- attribute distance
  - boolean, categorical: $$d_i(v_1, v_2) = \begin{cases} 0, & \text{if } v_1 = v_2; \\ 1, & \text{otherwise.} \end{cases}$$
  - else: $$d_i(v_1, v_2) = \frac{|v_1 - v_2|}{max(v)_{v \in V[i]} - min(v)_{v \in V[i]}}$$
- weights: default value (1.0), evolutionary optimalization

Method recommendation process

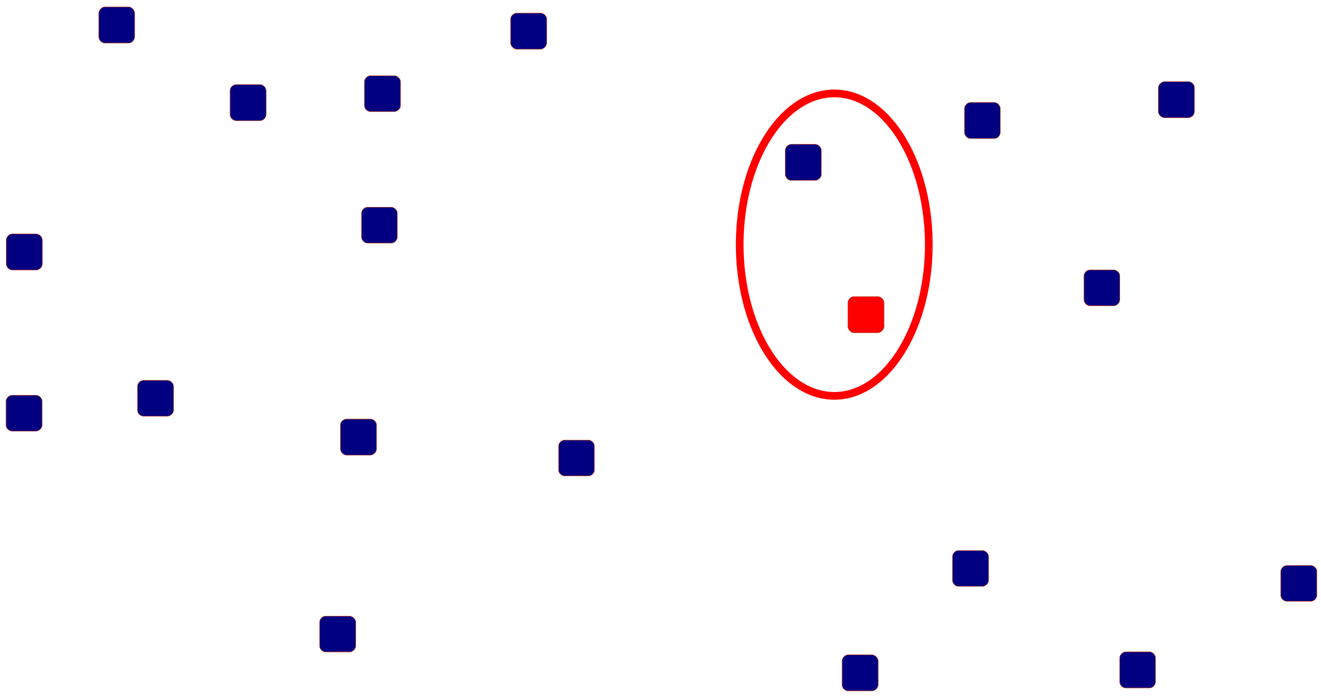


Method recommendation process

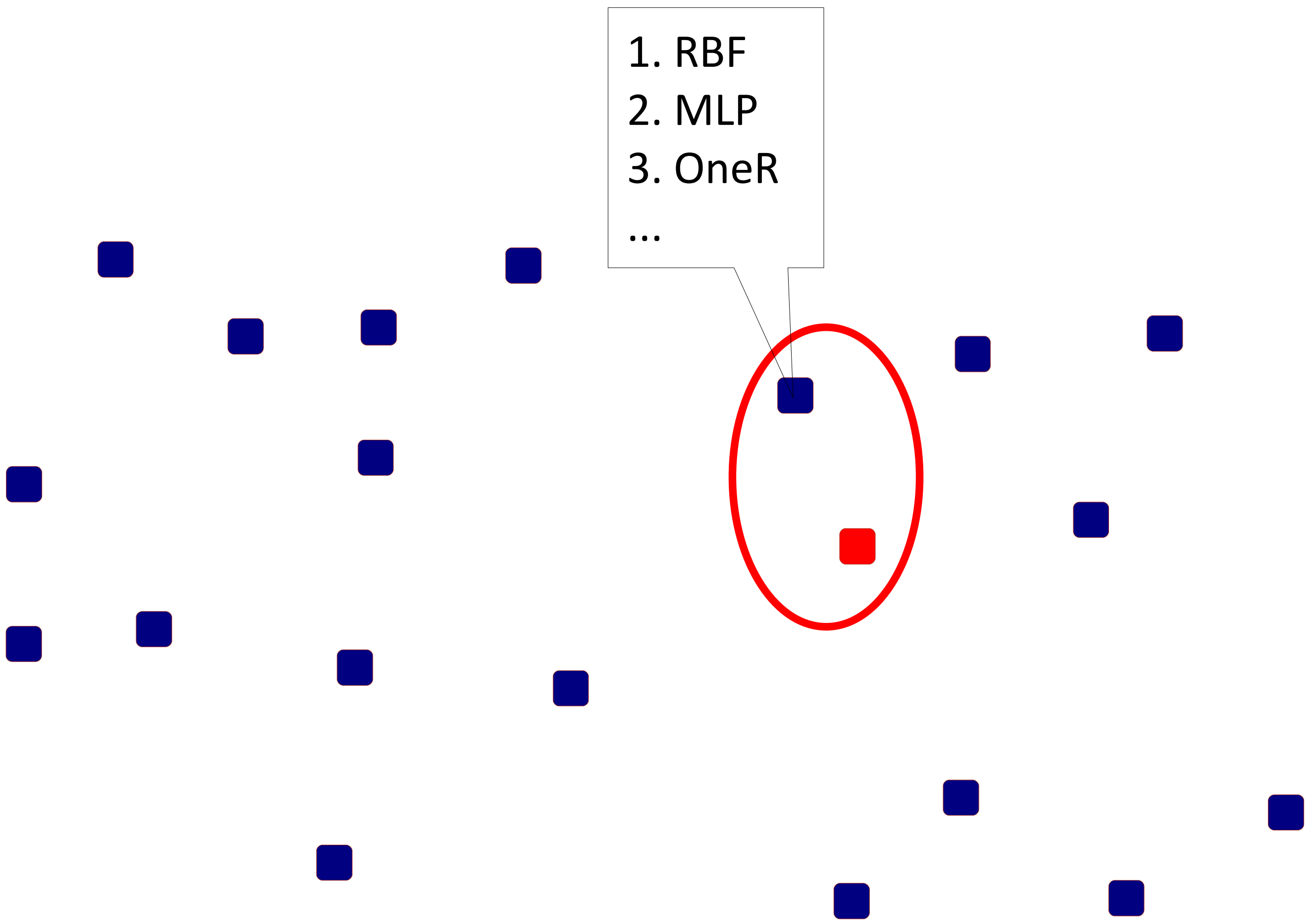
=> new dataset



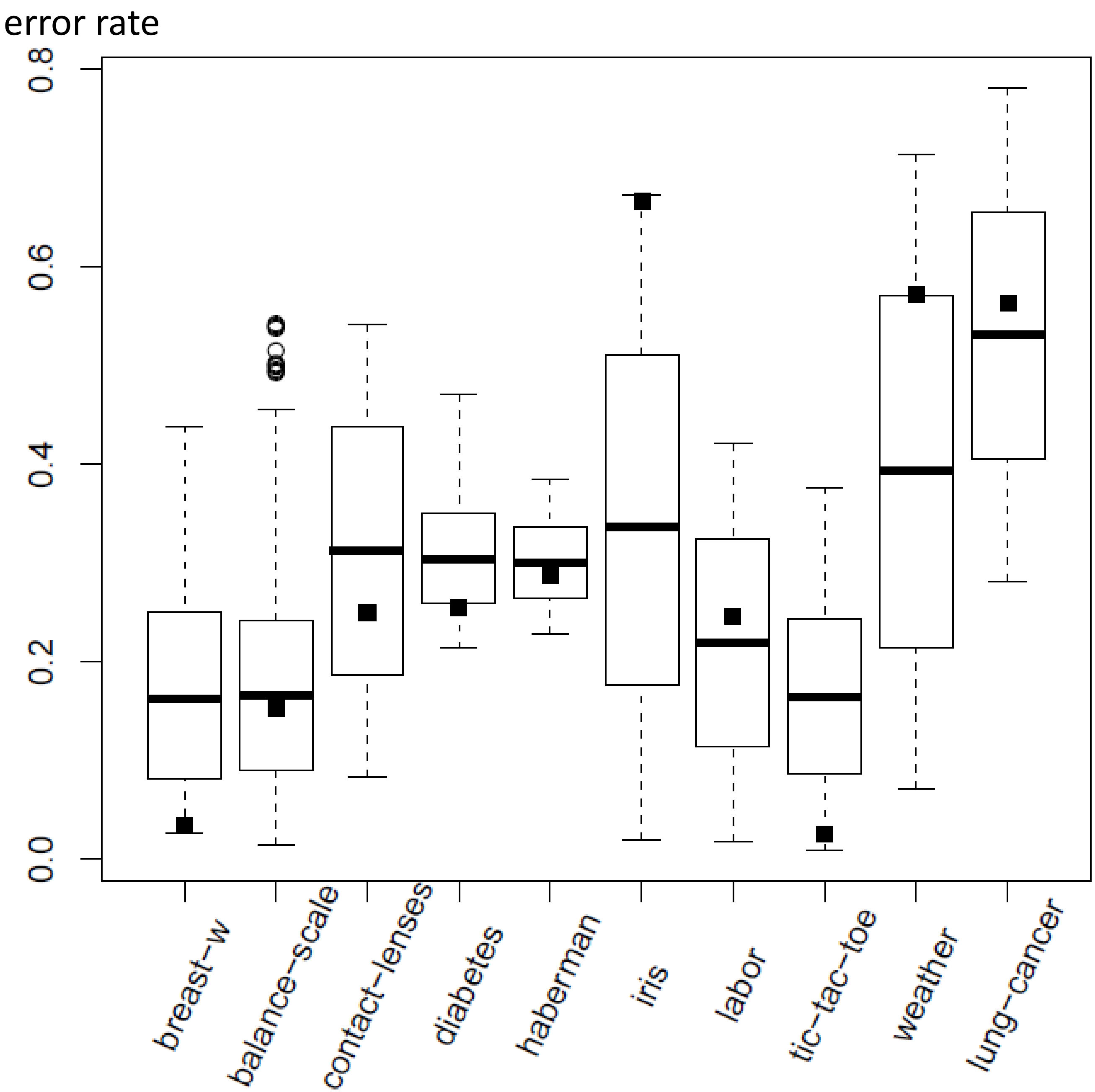
Method recommendation process



Method recommendation process



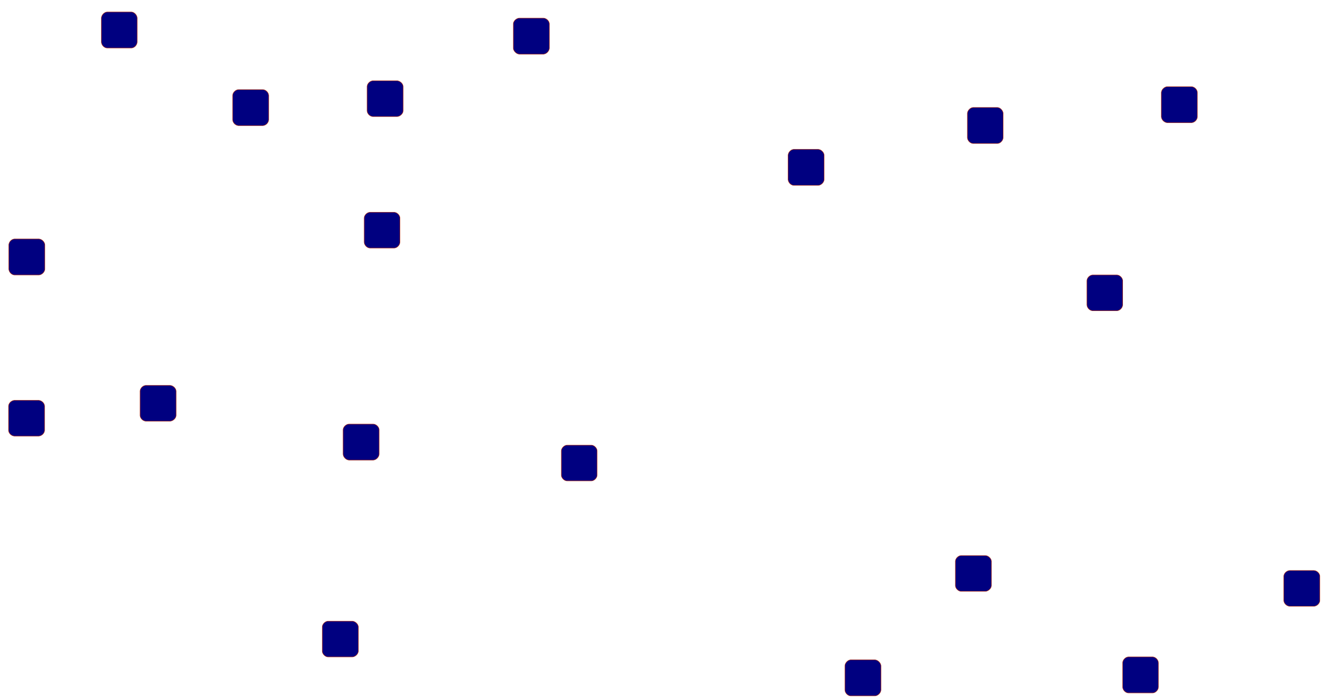
Results



- Narrow metadata, default weights (1.0)

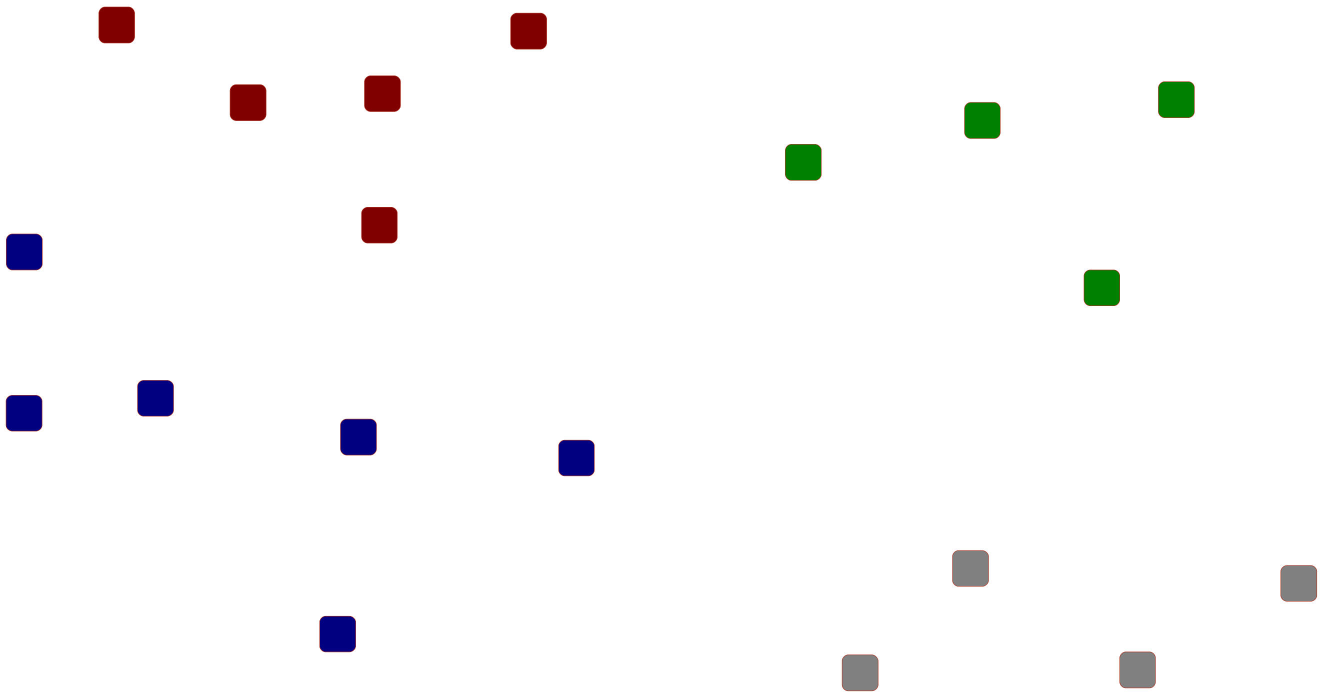
The minimum, quartiles, median, and maximum error rates for given dataset; the error rate of a recommended method is marked by a black square.

Method recommendation process II.



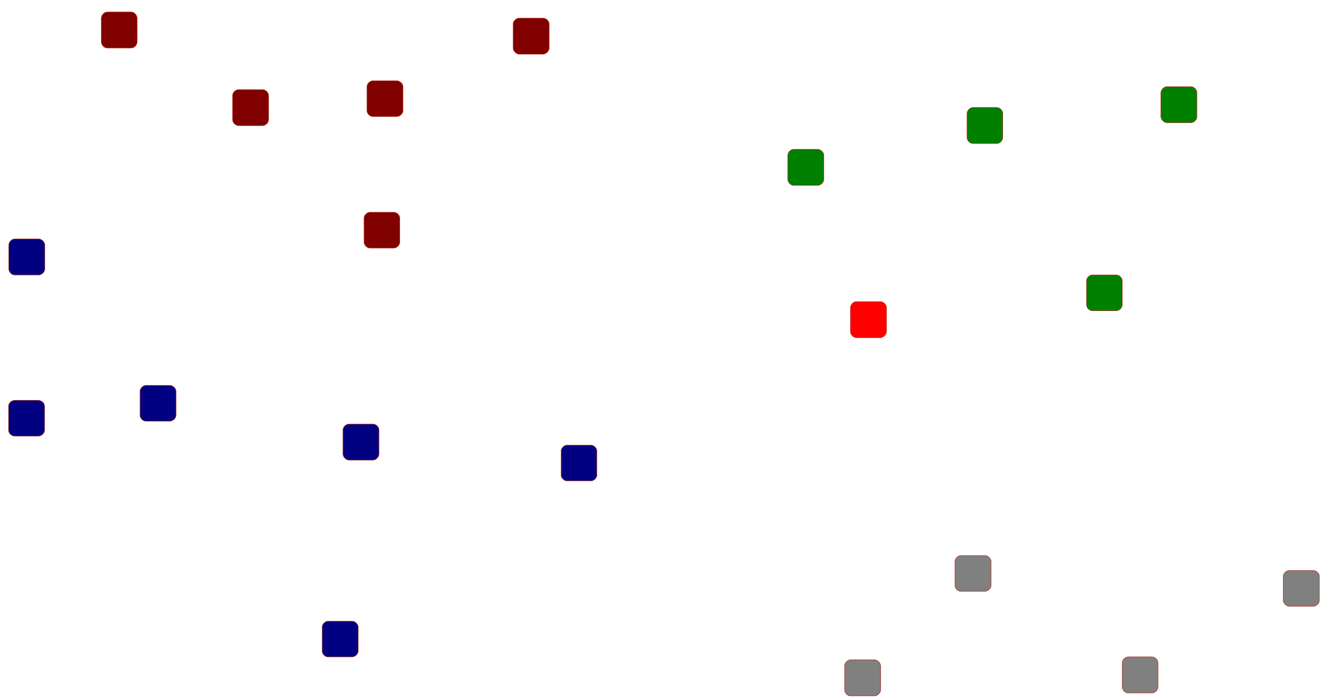
Method recommendation process II.

A. Training



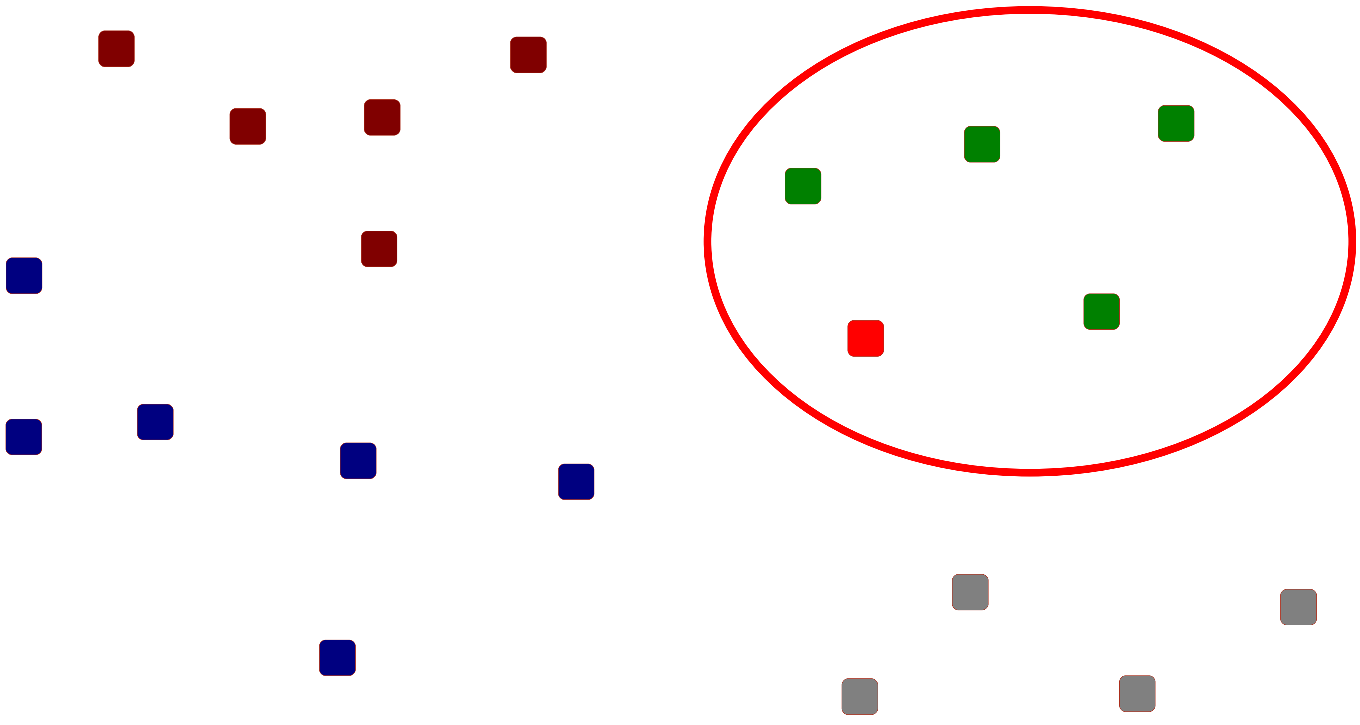
Method recommendation process II.

=> new dataset



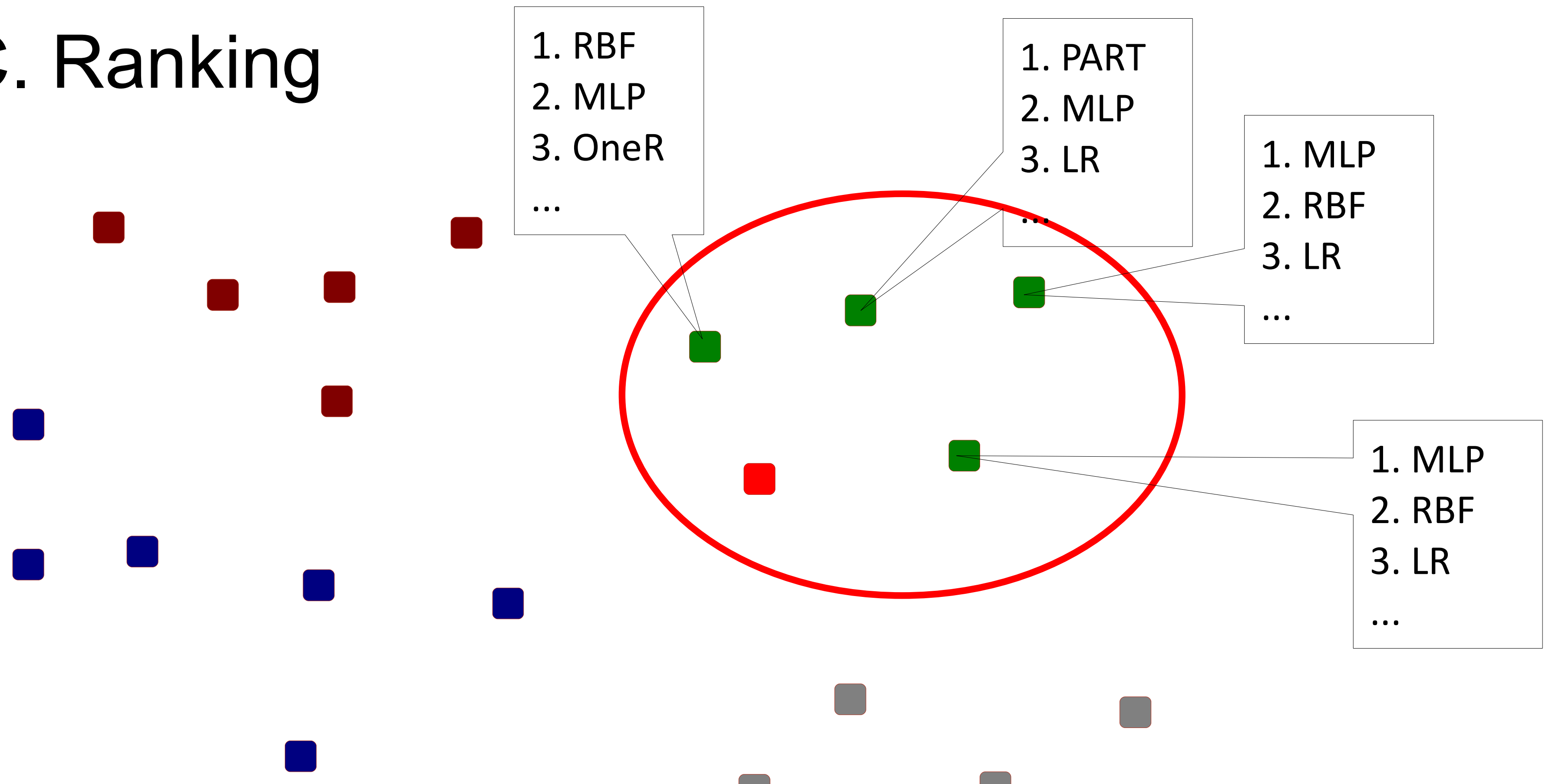
Method recommendation process II.

B. Zooming



Method recommendation process II.

C. Ranking



Choosing the Method

- comparison of methods:

normalized error rate $e_d(m)$

  - best result for each (dataset, method) pair
  - results linearly rescaled – best 0.0, worse 1.0
    - classification – error rate in [0,1]
    - regression – mean square error

- ranking:
  - compute **average of normalized errors** for each method (using datasets from the selected cluster)
  - select the method with the lowest ranking

Use Case

- 14 clusters, new dataset: vowel.arff

TABLE I. RANKING OF METHODS IN THE CLUSTER NO. 13 RECOMMENDED TO THE DATASET VOWEL.ARFF. THE RESULTS OF METHODS FOR THE DATASET ARE SHOWN.

Method	Ranking	Error Rate on *vowel*	Normalized Error Rate
RBF	0.020	0.077	0.087
MLP	0.029	0.023	0.000
PART	0.097	0.169	0.237
J48	0.106	0.125	0.166
RTree	0.115	0.124	0.164
NNGe	0.295	0.114	0.148
OneR	0.750	0.637	1.000
LReg	NaN	NaN	NaN

Results

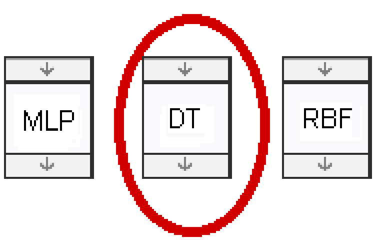
- Experiment setting:
  - 85 datasets, 8 data mining Weka methods, database of ~2 million previous results
- Variable number of clusters
- 2 different metadata types used
- 3 different metadata similarity metrics used in the training phase
- Better results achieved with smaller number of clusters (better generalization)

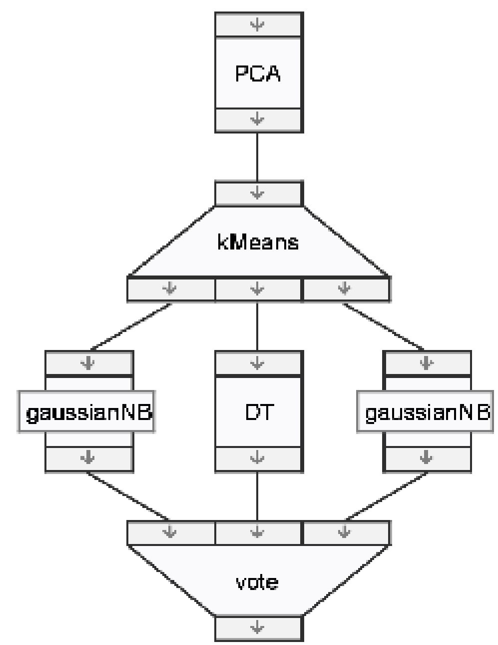
No. of Clusters	Narrow flat metric	Wide flat metric	Optimized wide metric
7	0.103	**0.076**	0.087
14	**0.074**	0.091	0.107
28	0.213	0.097	0.103
57	0.132	0.160	0.135

Average Normalized Error on testing data with the recommendation algorithm based on clusters

Generating Workflow Graphs

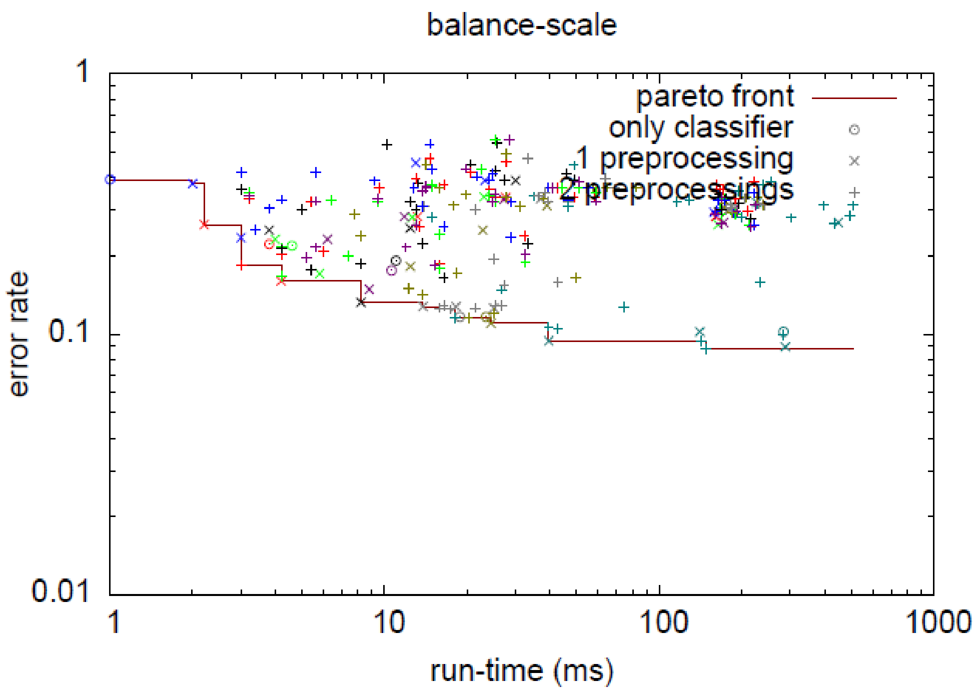
- recommendation of a single classifier
- chaining several preprocessing methods and a classifier
- generating the complete workflow schemes





Chaining Preprocessings

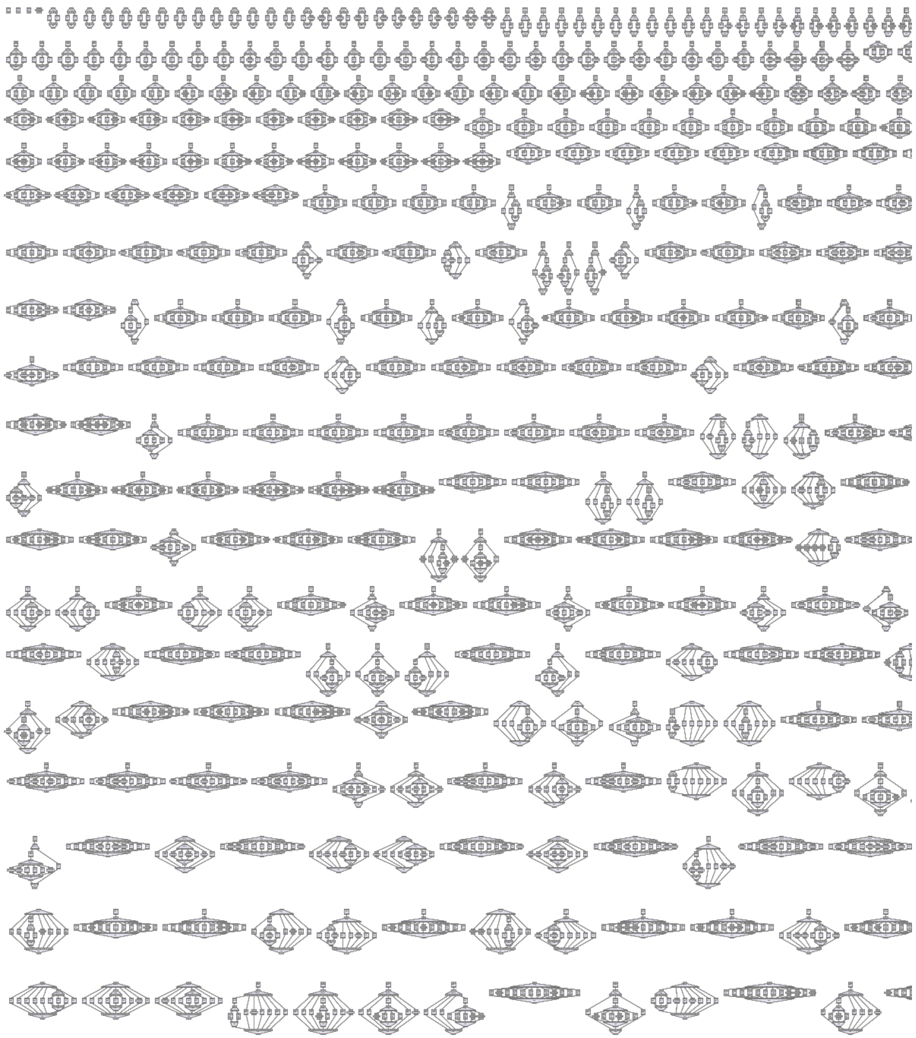
- max. 2 preprocessing methods
- full search of all possibilities
- multi-objective optimization (error-rate and time)
- example – balance-scale data set:



preprocessings	classifier	run-time	error-rate
	1R	1	0.3936
Resamp	1R	2	0.3972
Resamp	RandomTree	2.2	0.2640
Resamp-PCA	RandomTree	3	0.1840
PCA	RandomTree	4.2	0.1600
PCA	NNge	8.2	0.1328
Resamp	SMO	13.8	0.1280
Resamp-PCA	SMO	15.8	0.1264
Resamp-KMeans	SMO	17.4	0.1248
Resamp-KMeans	MLP	18	0.1152
PCA	RBF	24.4	0.1104
KMeans	MLP	39.6	0.0944
PCA-Resamp	MLP	147.8	0.0880

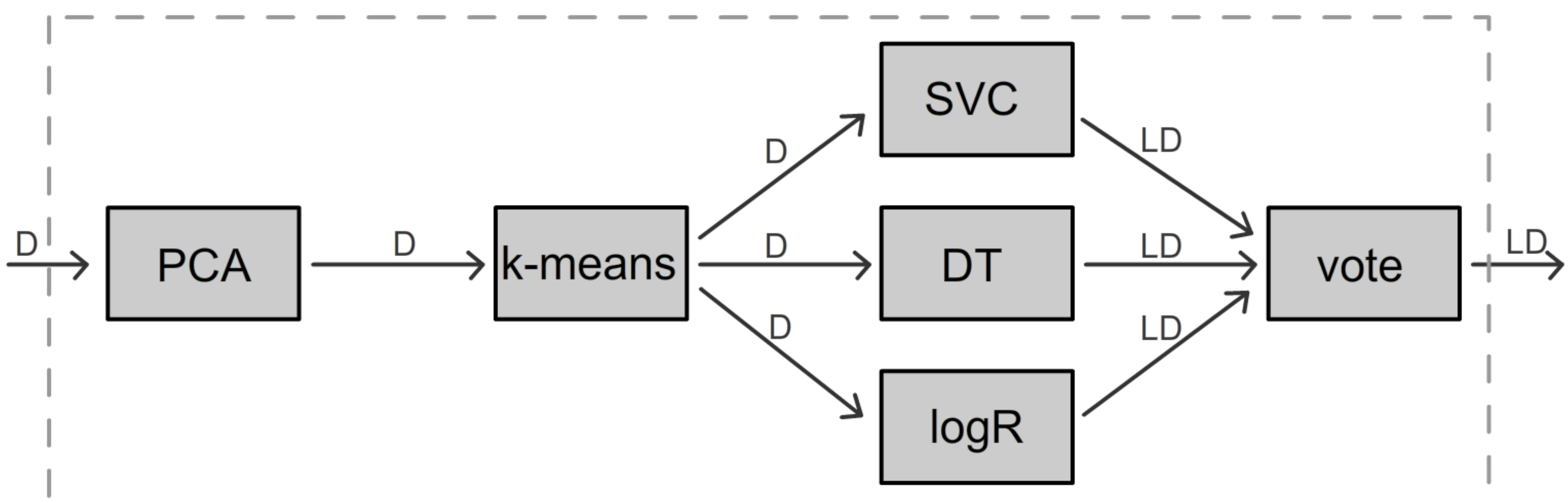
Generating Workflows

- standard approach: ontology based
- our approach: typed genetic programming
- systematically generated trees
- workflows represented as DAGs



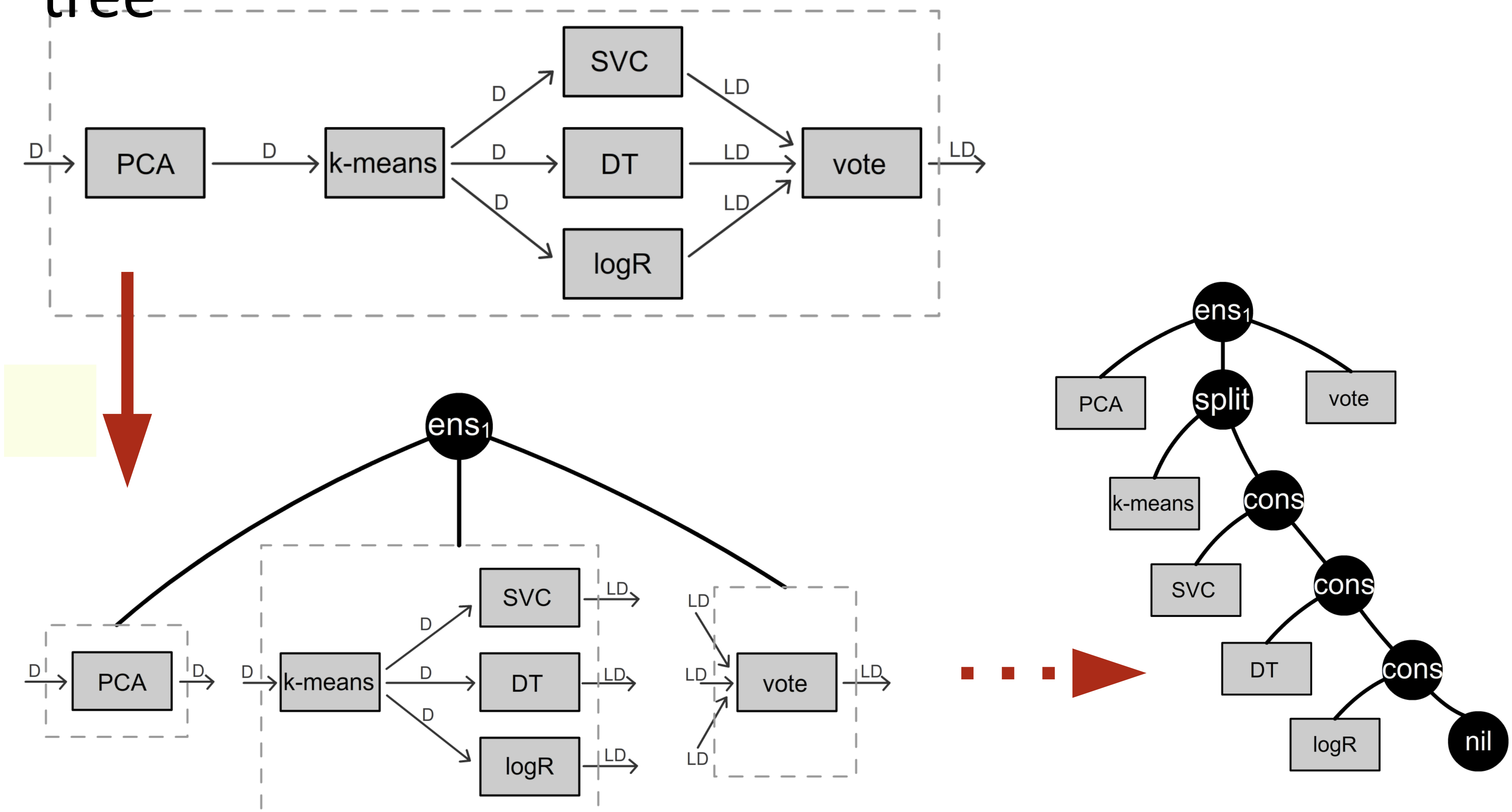
Using Typed GP

- The types ensure that the data flowing in the graph are consistent and that the whole graph makes sense from the data-mining point of view.



DAG to Tree

- example of decomposition of DAG to a syntactic tree



Testing our Workflows

- Two medium size classification problems from UCI repository: winequality-white, wilt

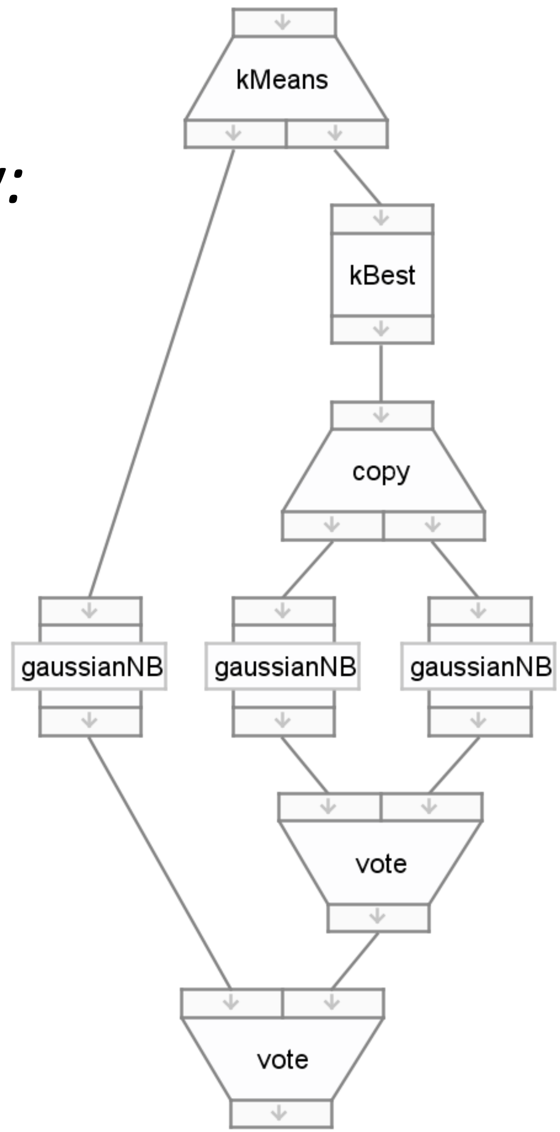
- Single classifier:
  - parameters tuned using grid search with 5-fold cross-validation

- Workflows:
  - more than 65,000 different workflows generated

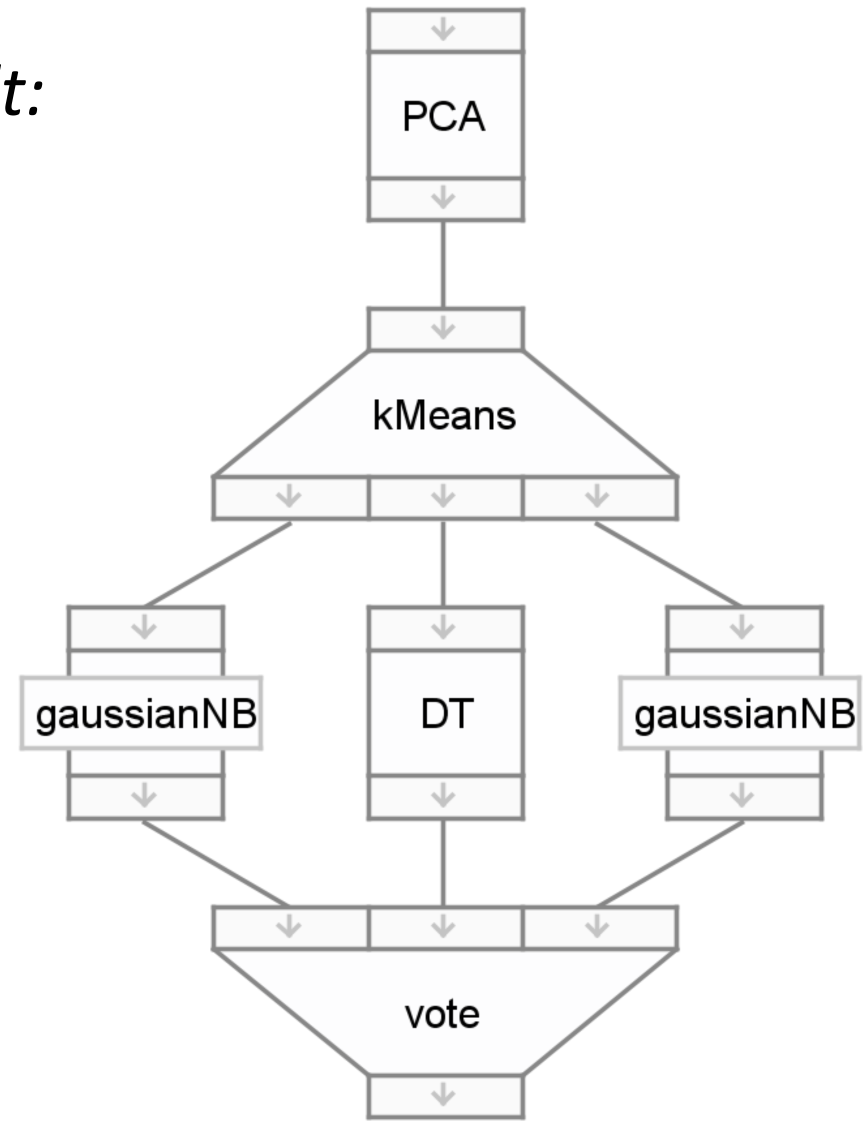
- Two experiment settings:
  - default parameters
  - parameters tuned for single classifiers

Example of Generated Workflows

wine quality:



wilt:

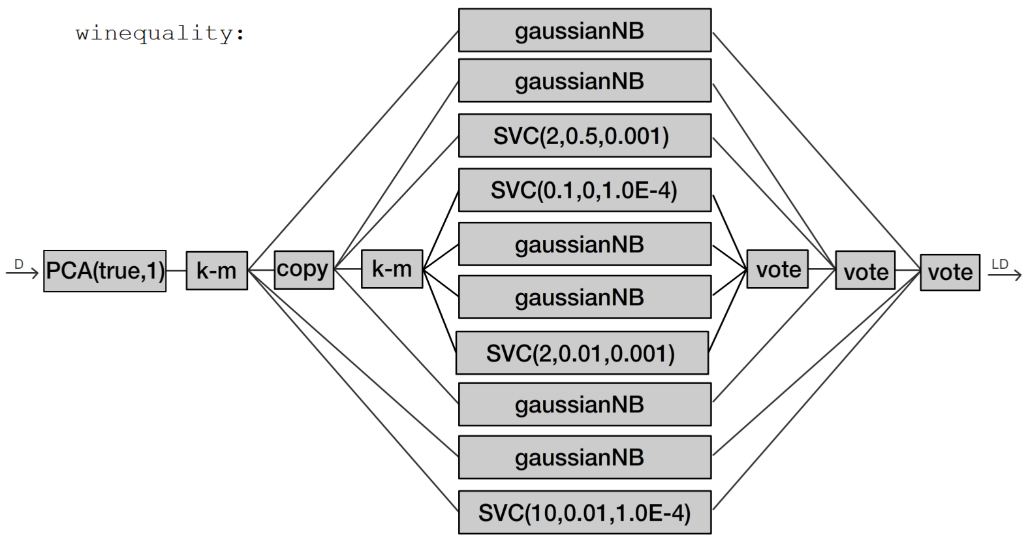


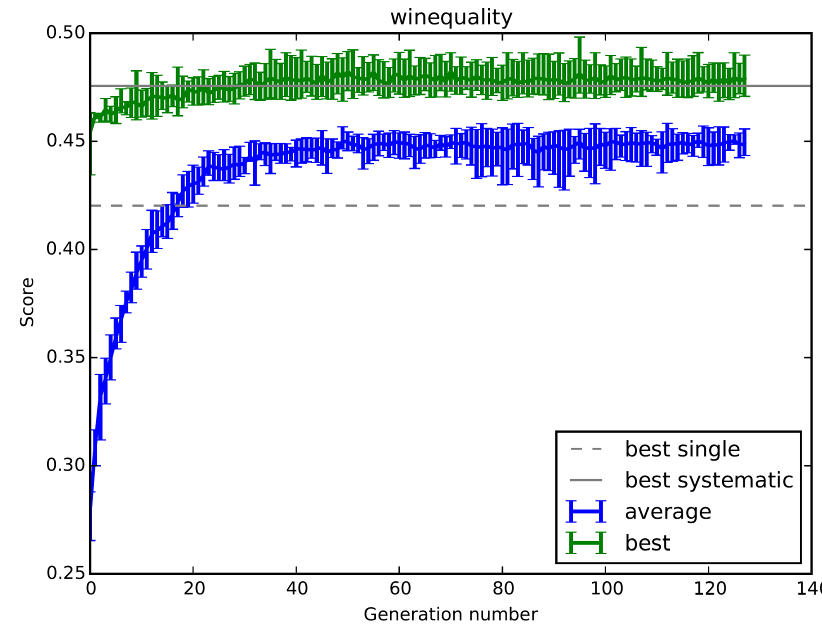
Evolving Workflow Graphs

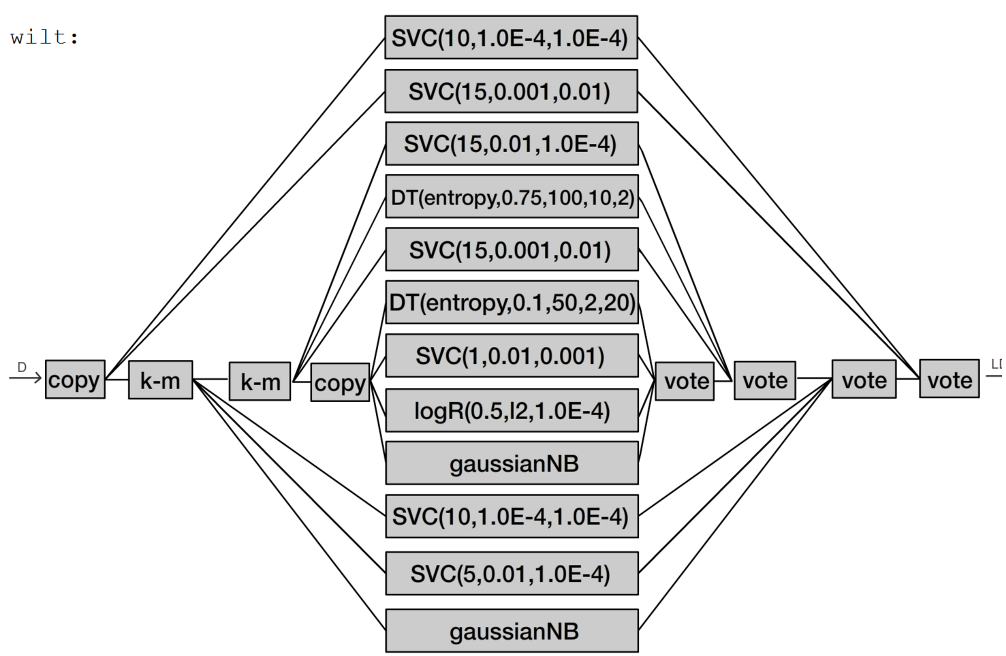
- 128 generations, 256 individuals
- mutation – replacing subtree with new subtree of the same size (prob. 0.3, max. 10 nodes)
- simple typed crossover – swaping a pair of subtrees with the same type (prob. 0.3, max. 50 nodes)
- fitness: quadratic weighted kappa
- tournament selection (prob. 0.8)

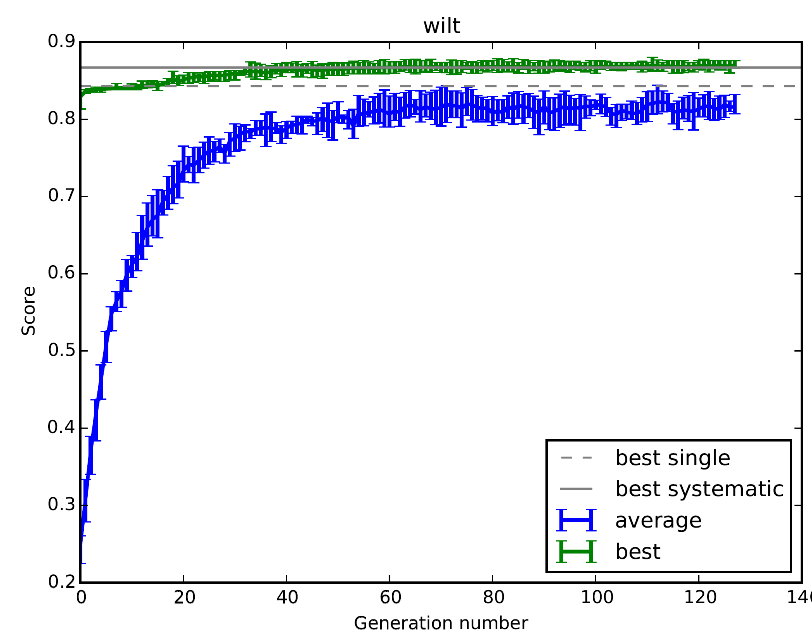
- parameter tuning is part of the genetic programming

Results


winequality:
D
PCA(true,1)
k-m
copy
k-m
gaussianNB
gaussianNB
SVC(2,0.5,0.001)
SVC(0.1,0,1.0E-4)
gaussianNB
gaussianNB
SVC(2,0.01,0.001)
gaussianNB
gaussianNB
SVC(10,0.01,1.0E-4)
vote
vote
vote
LD


winequality
Score
Generation number
best single
best systematic
average
best


wilt:
D
copy
k-m
k-m
copy
SVC(10,1.0E-4,1.0E-4)
SVC(15,0.001,0.01)
SVC(15,0.01,1.0E-4)
DT(entropy,0.75,100,10,2)
SVC(15,0.001,0.01)
DT(entropy,0.1,50,2,20)
SVC(1,0.01,0.001)
logR(0.5,l2,1.0E-4)
gaussianNB
SVC(10,1.0E-4,1.0E-4)
SVC(5,0.01,1.0E-4)
gaussianNB
vote
vote
vote
vote


wilt
Score
Generation number
best single
best systematic
average
best

Results

dataset	winequality		wilt	
params	default	tuned	default	tuned
SVC	0.1783	0.3359	0.0143	0.8427
LR	0.3526	0.3812	0.3158	0.6341
GNB	0.4202	0.4202	0.2916	0.2917
DT	0.3465	0.4283	0.7740	0.8229
GP	**0.4792**		**0.8702**	
systematic	**0.4731**	**0.4756**	**0.8471**	**0.8668**

Comparison of the classifiers and the workflows. Results were compared using quadratic weighted kappa metric with cross-validation.

Thank you for your attention

Questions...