Training Set Construction Methods

Tomas Borovicka

Faculty of Information Technology
Czech Technical University in Prague

February 17, 2012

Outline

Motivation

- Good and reliable models,
- model parameters estimation,
- model assessment,
- accuracy prediction,
- big datasets - reduction,
- instances can not improve accuracy of the model or even can degraded the model,
- noise reduction,
- speed up learning process,
- speed up model response.
- ...

Model selection process

1. Model selection
   1.1 Model learning and parameters estimation (Training phase)
   1.2 Model validating (Validation phase)
2. Model assessment (Testing phase)

Model assessment

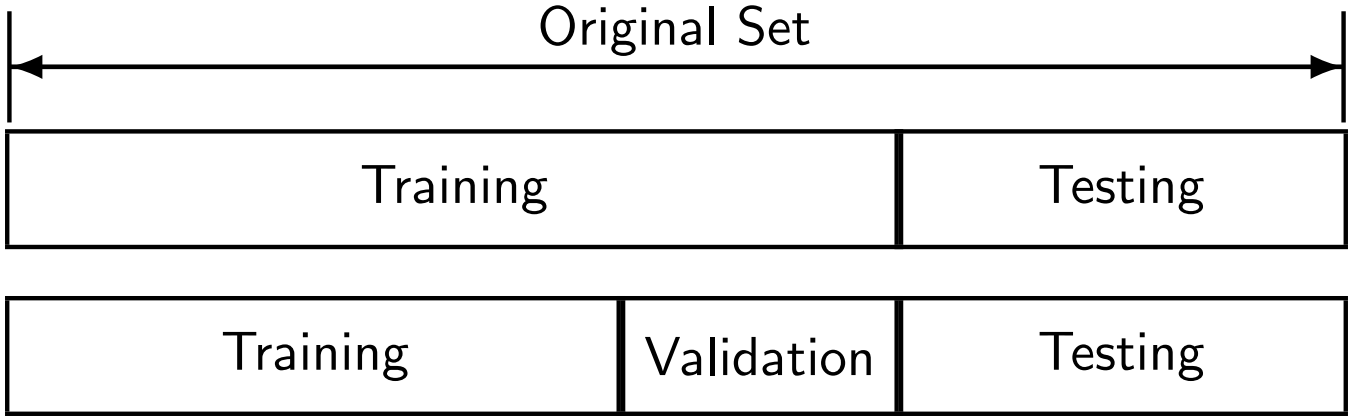
- Comparison of the model with physical theory
- Comparison of model with theoretical or empirical model
- Collect new data for assessment
  - It is the most preferred way for model evaluation. But usually we are not capable to collect new data for this purpose.
- Use the same data as for model building
  - Use the same data for assessment gives too optimistic estimation of real performance.
- Reserve part of the learning data for assessment
  - It is the most common way how to deal with absence of independent data for evaluation. Splitting the data is wished to have same effect as having two datasets. But it is not!

How to split data

Training set - set of examples used for building the model.

Validation set - set of examples used to estimate the parameters of the model.

Testing set - set of examples used to assess the performance of the model.



Commonly two thirds for training and one third for testing or half to half. But literature says that best ratio varies substantially from case to case.

Splitting strategies

1. Use one sample for training and second sample for assessment (Strategy A1), respectively one for build the model, one for validation and one for assessment (strategy A2).
2. Use one sample for build the model, second for validation and assessment.
3. Use part of data for building the model and all data for validation and assessment (strategy C).
4. Use same data for all task (Strategy N).

Strategy	**Training**	**Validating**	**Testing**
A1	Part1	Part1	Part2
A2	Part1.1	Part1.2	Part2
B	Part1	Part2	Part2
C	Part1	All	All
N	All	All	All

Hold out

- Takes original dataset and splits it randomly into two sets.

+ Simple implementation.
+ Low computational cost.

- Selected samples might not be representative and in the worst case one or more classes might be missing in the test set
- Does not prevent bias in training and testing sets.

- Advanced versions use stratification - each class is represented with approximately same frequency in both subsets.
- Repeated holdout - method is repeated and the resulting accuracy is average of all iterations.

Cross-validation

k-fold cross-validation

- The original data set is split into k disjoint folds of same size.
- In each from k turns it uses one fold for testing and the remaining $k - 1$ folds for training.
- Experiments has shown that the best number for k is ten.
- All the instances in the original data set are used for training and for testing.

Leave-one-out cross-validation

- Special case of k-fold cross-validation in which $k = n$, where n is the size of the original dataset.
- All test sets have only one instance.
- Extremely computationally expensive.

Kennard-Stone

- For splitting data sets into two subsets.
- Algorithm starts with selecting two most distant instances.
- For each instance remaining in the data set find smallest distance to already selected objects.
- Select instance with the maximal distance among these smallest distances.
- Repeat until enough objects are selected.

Main idea

- The goal of this methods is to split original data set into two similar disjoint sets.
- Based on assumption that two sets P and Q formed by splitting original dataset T are as similar as possible when sum of distances of all pairs (two instances each from one set) are minimized over whole datasets.

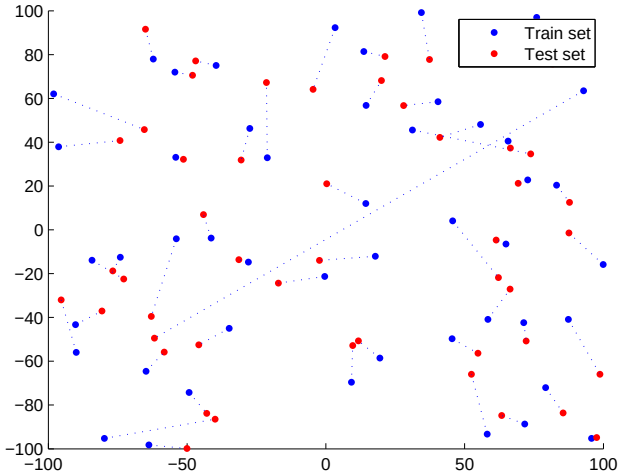
$$d^* = \arg\min_d \sum_{\{p,q\} \in T} dist(p, q).$$

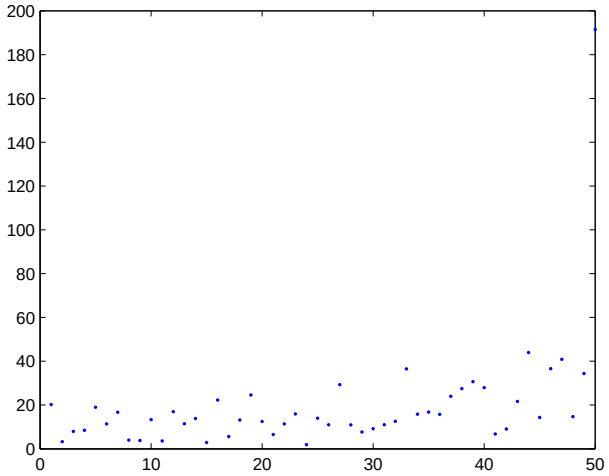
Nearest neighbours method

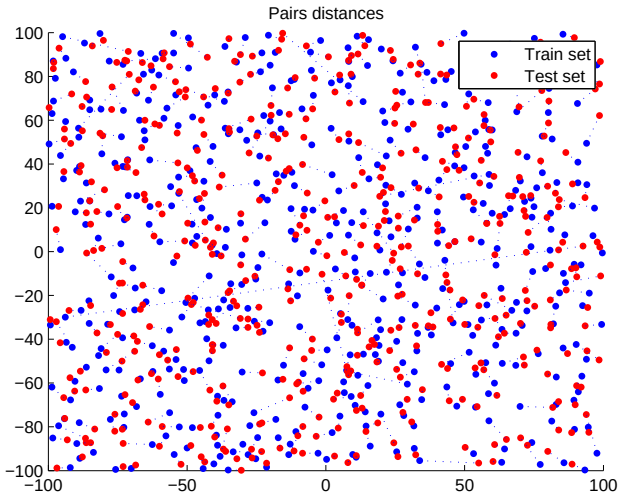
- Algorithm splits original datasets into two datasets by finding nearest neighbours of instance and putting each into the different subset.

1. Randomly choose one instance.
2. Find for the instance k-nearest neighbours (where k is the number of desired subsets - 1) and put each neighbour into the different subset.
3. Repeat until any instance remain.

Visualization - 2D data





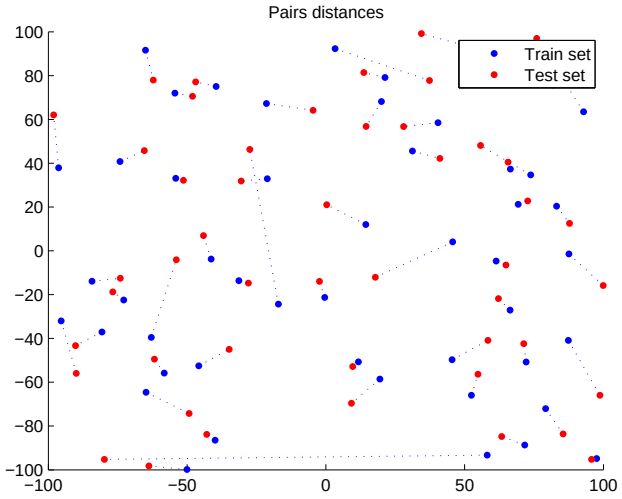


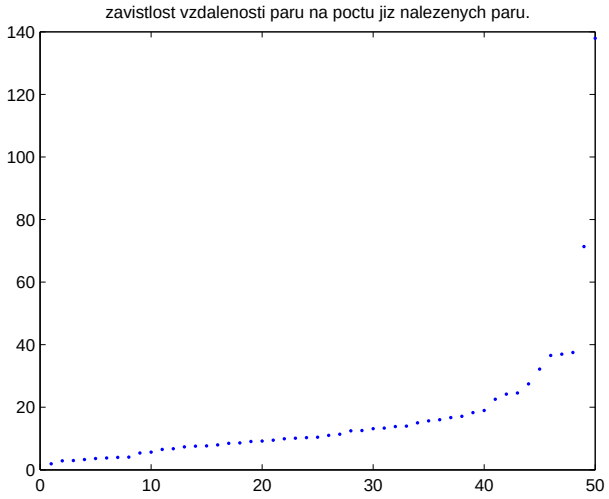
Closests pairs method

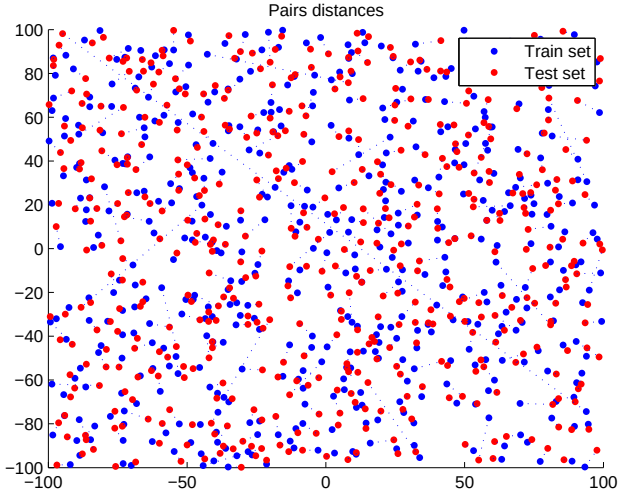
- Heuristic approach based on repeatedly finding the closest pairs of instances and putting each into the different subset.

1. Find the closest pair in the original dataset.
2. Put each instance into the different subset.
3. Repeat until any instance remain.

Visualization - 2D data







Goals of experiment

- Analyse the evolved algorithms.
- Evaluate performance of the evolved algorithms and make comparison with most common methods.
- Assess error prediction reliability.

Experiment implementation

1. Resample original dataset randomly
   - 70% Learning set
   - 30% Testing set
2. Split learning set into training and validation sets (50/50) using different methods
   - Random - The random resampling method,
   - NN - The nearest neighbour method,
   - Closest pairs - The closest pairs method,
   - Kennard-Stone.
3. Estimate the classification accuracy on the validation set and on the test sets using several common classifiers.
   - Bayes - The Naive Bayes classifier,
   - kNN - The k-Nearest Neighbours classifier,
   - ANN - The Artificial Neural Network classifier (multi-layer perceptron).
4. Repeat process 20 times and average the results.

Datasets

Dataset 1: Mammographic Mass Data

Number of Instances: 961
Number of Attributes: 6
Class Distribution: benign: 516; malignant: 445
Missing Attribute Values: Yes

Dataset 2: Pima Indians Diabetes Database

Number of Instances: 768
Number of Attributes: 8
Class Distribution: 0:500 1:268
Missing Attribute Values: Yes

Dataset 3: Blood Transfusion Service Center Data Set

Number of Instances: 748
Class Distribution: 0:76% 1:24%
Number of Attributes: 5
Missing Attribute Values: No

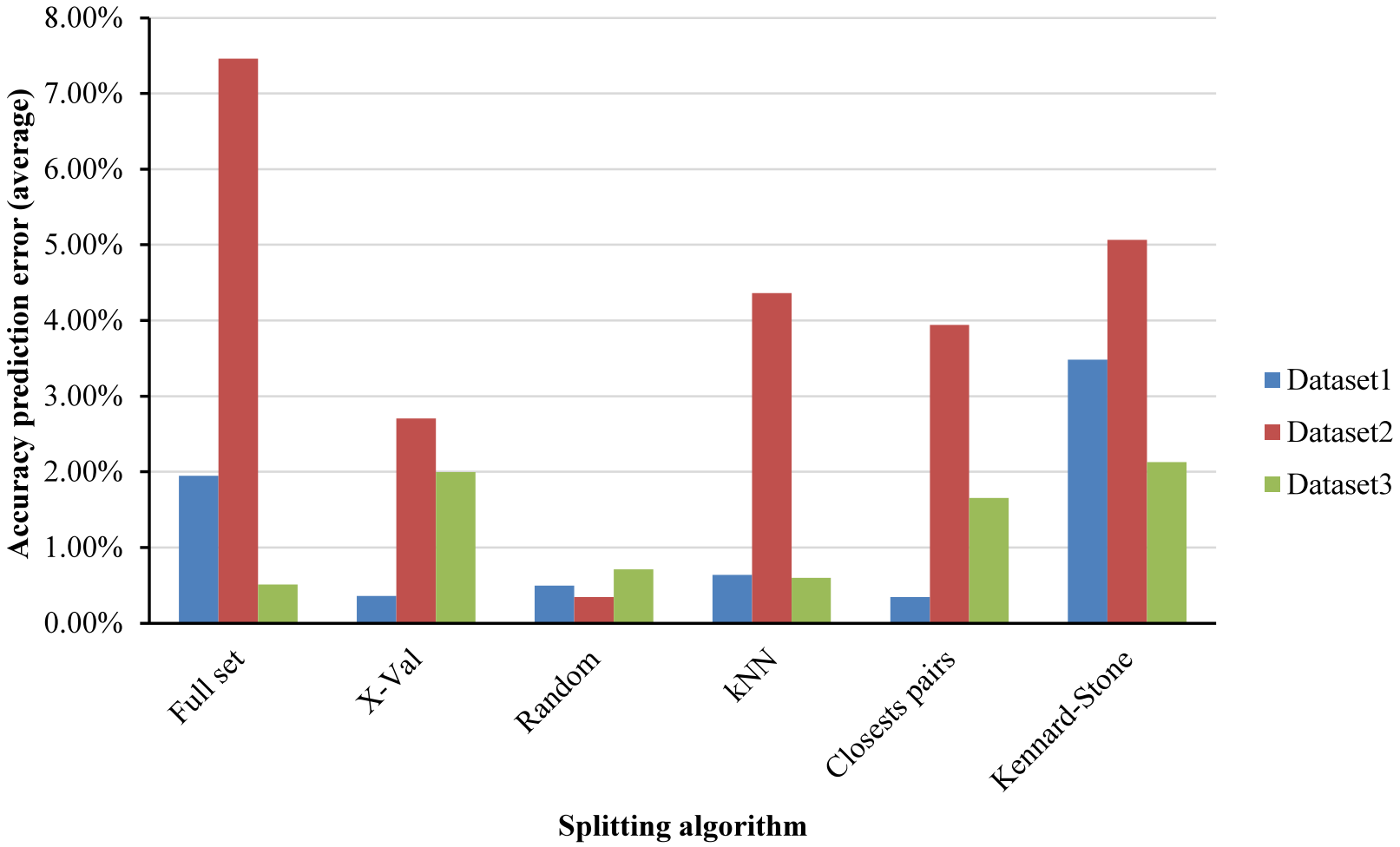
Experiment - results

Dataset1	Predicted accuracy			Real accuracy			Accuracy prediction error			
	Bayes	NeuralN	kNN	Bayes	NeuralN	kNN	Bayes	NeuralN	kNN	AVG
Full set	0.7994	0.8366	0.8148	0.7977	0.8056	0.8005	0.21%	3.85%	1.79%	1.95%
X-Val	0.7963	0.8030	0.7985	0.7977	0.8083	0.8005	0.18%	0.66%	0.25%	0.36%
Random	0.7915	0.8060	0.7914	0.7958	0.8092	0.7957	0.54%	0.40%	0.54%	0.49%
kNN	0.8024	0.8162	0.8080	0.7993	0.8056	0.8064	0.38%	1.32%	0.20%	0.64%
Closests pairs	0.7924	0.8143	0.8063	0.7948	0.8120	0.8026	0.30%	0.28%	0.45%	0.35%
Kennard-Stone	0.8330	0.8368	0.8270	0.8012	0.8106	0.8010	3.97%	3.23%	3.25%	3.48%

Dataset2	Predicted accuracy			Real accuracy			Accuracy prediction error			
	Bayes	NeuralN	kNN	Bayes	NeuralN	kNN	Bayes	NeuralN	kNN	AVG
Full set	0.7850	0.8344	0.7804	0.7493	0.7441	0.7398	4.76%	12.14%	5.49%	7.46%
X-Val	0.7769	0.7222	0.7296	0.7493	0.7450	0.7398	3.68%	3.06%	1.38%	2.71%
Random	0.7578	0.7314	0.7359	0.7530	0.7337	0.7352	0.63%	0.31%	0.09%	0.34%
kNN	0.7810	0.7829	0.7667	0.7517	0.7422	0.7393	3.90%	5.49%	3.70%	4.36%
closests pairs	0.7749	0.7697	0.7667	0.7526	0.7298	0.7415	2.96%	5.47%	3.40%	3.94%
Kennard-Stone	0.7842	0.7791	0.7777	0.7550	0.7341	0.7393	3.87%	6.13%	5.19%	5.06%

Dataset3	Predicted accuracy			Real accuracy			Accuracy prediction error			
	Bayes	NeuralN	kNN	Bayes	NeuralN	kNN	Bayes	NeuralN	kNN	AVG
Full set	0.7705	0.7893	0.7990	0.7746	0.7929	0.7946	0.53%	0.45%	0.55%	0.51%
X-Val	0.7625	0.7712	0.7817	0.7746	0.7935	0.7946	1.56%	2.81%	1.62%	2.00%
Random	0.7624	0.7742	0.7714	0.7700	0.7800	0.7745	0.99%	0.74%	0.40%	0.71%
kNN	0.7689	0.7901	0.7914	0.7710	0.7830	0.7865	0.27%	0.90%	0.62%	0.60%
closests pairs	0.7812	0.7986	0.8060	0.7680	0.7880	0.7910	1.72%	1.34%	1.89%	1.65%
Kennard-Stone	0.7644	0.7739	0.7716	0.7783	0.7908	0.7911	1.79%	2.14%	2.46%	2.13%

Experiment - results



Summarized

- Train a test on the same data leads to optimistic prediction, not surprisingly.
- Random selection outperforms all methods in terms of accuracy prediction in this case.
- Can not recommend the best way for splitting dataset.

In literature...

- Success of training set construction method is strongly problem dependent.
- Moreover, closely related to classification and regression method.

Instance selection methods

Representative set

We can define **representative set** as a special subset of the original dataset which satisfies three main characteristics:

1. It is significantly smaller in size compared to the original dataset.
2. It captures the most of information from the original dataset compared to any subset of the same size.
3. It has low redundancy among the representatives it contains.

In order to defining representative we can define **minimal consistent subset** of training set:

- Given an original training set T, $S \subset T$,
- such that S is the smallest set of instances and
- $Acc(S) \cong Acc(T)$.

Dividing

Wrapper methods

The selection criterion is based on the accuracy obtained by a classifier (commonly, those instances that do not contribute to the classification accuracy are discarded from the training set).

Filter methods

The selection criterion uses a selection function that is not based upon a classifier but rather on the features of the instance vector.

Condensed Nearest Neighbour (CNN)

- Probably first published instance selection algorithm,
- Incremental method starting with new set S includes one instance per class chosen randomly from T.
- Next step classifies T using S as a training set.
- Each wrongly classified instance from T is added to S.
  - Select noise instances.
  - Bad performance.

Generalized Condensed Nearest Neighbour (GCNN)

- Based on CNN.
- Define absorption criterion:
  instance x is absorbed if $\|x - q\| - \|x - p\| > \delta$, where p is the nearest neighbour of the same class as x and q is the nearest neighbour belonging to a different class than x.

+ Good performance in accuracy and reduction ratio.
+ Good for medium/large datasets.

Edited Nearest Neighbour (ENN)

- Decremental algorithm starting with $S = T$.
- Removes given instance from S if its class does not agree with the majority class of its neighbourhoods.

\+ Preferred as noise filter.

Methods based on ENN:

All k-nn

Runs ENN repeatedly for all $k(k = 1, 2, \ldots, l)$. In each iteration misclassified instances are discarded.

Multiedit

Divides T randomly into r blocks and $B_1 \ldots B_r$ and applies ENN over each block B_i using B_{i+j} for finding neighbours.

Instance Based (IB1-5)

- IB1 is 1-NN algorithm.
- IB2 selects the instances misclassified by IB1.
- IB3 uses a significance test for retaining instances. Confidence intervals are used for determine the impact of the instance.

\+ Good performance in accuracy and reduction ratio.

IB4 and IB5 extends IB3 in order to handle irrelevant attributes ...

Decremental Reduction Optimization Procedure (DROP1-5)

- Uses associate, defined as $Associates(x)$ contains all instances that have x as one of its neighbours.
- DROP1 removes from S instances that do not change classification of its associates.
- DROP2 is same as DROP1 but the associates are taken from the original training set T.
- DROP3 and DROP4 run noise filter first and then apply DROP2.

\+ Good performance in accuracy and reduction ratio.

\- Non usable on larger datasets.

Pair Opposite Class-Nearest Neighbour (POC-NN)

- Calculates the mean of all instances in each class (m_i).
- Finds a border instance p_{b1} belonging to the class C_1 as instance that is the nearest to m_2, the mean of class C_2.

\+ Good performance in reduction and acceptable accuracy.

And many others

- Object Selection by Clustering (OSC)
  Based on clustering, selects border instances in heterogeneous clusters and some interior instances in homogeneous clusters.
- Iterative Case Filtering (ICF)
- Reduced Nearest Neighbour (RNN)
- Selective Nearest Neighbour (SNN)
- Pattern by Ordered Projections (POP)
- Generalized-Modified Chang algorithm (GCM)
- Weighting prototype (WS)
- Selection by Relevance (PSR)
- Methods based on evolutionary algorithms, TS
- ...

Summarized

- DROP3, GCNN, Explore, IB3 are the most effective as presented in the literature,
- from filter methods POC-NN and OSC.
- Instance selection method is problem dependent and none of them is superior over many problems then other.

Thanks for your attention