

Data Normalization and Scaling

EGI SAFITRI, S.MAT., M.SI



Introduction to Data Normalization

Normalization

Data normalization is a crucial process in database design that design that involves organizing data in a database to reduce reduce redundancy, improve data integrity, and enhance overall enhance overall efficiency. It involves breaking down a database database into smaller tables and defining relationships between relationships between them.

Importance of Data Normalization

Data normalization is crucial for ensuring data consistency, accuracy, and efficiency in database management and machine learning models. It helps eliminate redundancy, improve data integrity, and facilitate seamless data integration and analysis.

Normalizing data improves the performance and scalability of applications, reduces storage requirements, and enhances the overall data quality, leading to better decision-making and insights.

Introduction to Database Normalization

Database normalization is the process of organizing data in a database to reduce redundancy, improve data integrity, and optimize query performance. It involves a series of steps, each referred to as a "normal form", that progressively improve the structure of the database.

What is Unnormalized Form (UNF)?

Unnormalized Form (UNF) is the starting point of database normalization. In UNF, data is stored in a single, flat table with repeating groups of information. This structure can lead to data redundancy, update anomalies, and inefficient queries.

First Normal Form (1NF)

Definition

1NF requires that the data be data be stored in a tabular tabular format with no repeating groups. Each cell cell should contain a single single value.

Benefits

1NF eliminates data redundancy and improves improves data integrity by by ensuring that each piece piece of information is stored stored only once.

Example

Transforming a table with repeating product details into into individual rows, each with a single product per row.



DatabaseTown.com



Normalization in DBMS

Student Table

St.ID	Name	Subject	Grade
1	John Smith	Math, Science	A
2	Jane Doe	English, History	B

First Normal Form (1NF) is the most basic level of normalization. The rules for achieving 1NF are:

- 1 Each table should have a primary key, which uniquely identifies each record in the table.
- 2 Each column in the table should contain only atomic values, which means that a single cell should contain a single value and not a list of values.
- 3 There should be no repeating groups of data.

Student Table in 1NF

St.ID	Name	Subject	Grade
1	John Smith	Math	A
1	John Smith	Science	A
2	Jane Doe	English	B
2	Jane Doe	History	B



Second Normal Form (2NF)

1 Requirement

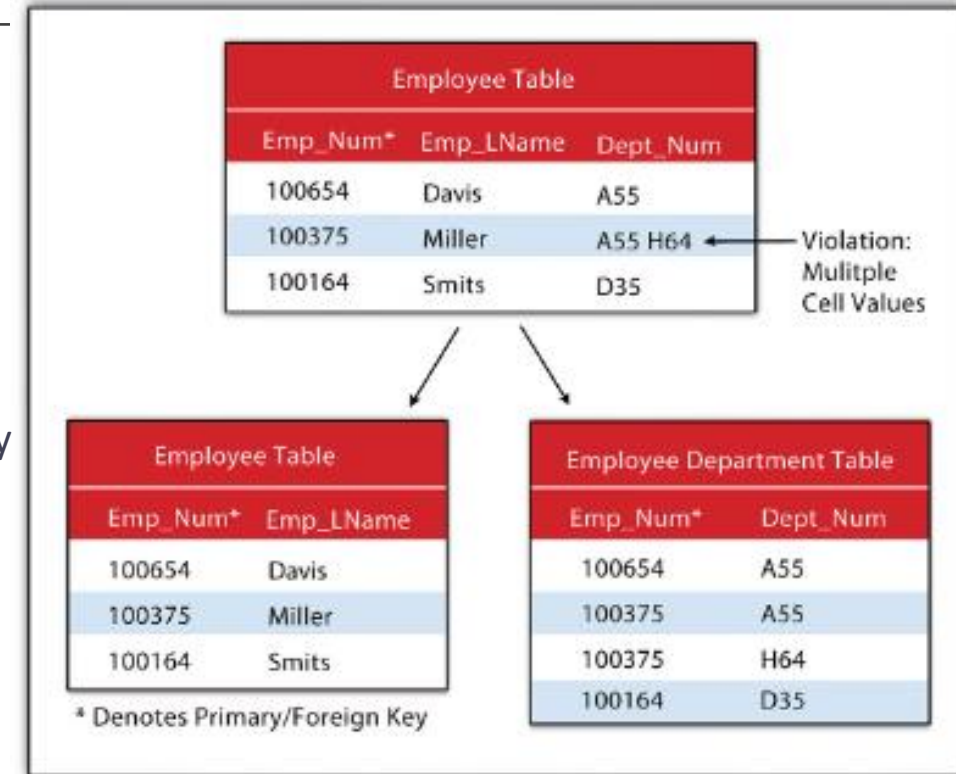
2NF requires that all non-key attributes are fully dependent on the primary key.

2 Elimination of Partial Dependencies

2NF ensures that there are no partial dependencies, where a non-key attribute depends on only a part of the primary key.

3 Benefits

2NF reduces data redundancy and improves data integrity by separating data into multiple tables based on their dependencies.



Third Normal Form (3NF)

Requirement

3NF requires that all non-key attributes are directly dependent on dependent on the primary key and and not dependent on other non-key non-key attributes.

Elimination of Transitive Dependencies

3NF ensures that there are no transitive dependencies, where a non-key attribute depends on another non-key attribute.

Benefits

3NF further reduces data redundancy and improves data integrity by eliminating any eliminating any remaining dependencies between non-key attributes.

Understanding the Concept of Data Scaling

Scaling

1

What is Data Scaling?

Data scaling is the process of transforming data into a common range, usually usually between 0 and 1, to ensure all features contribute equally to the the machine learning model.

2

Normalization vs. Standardization

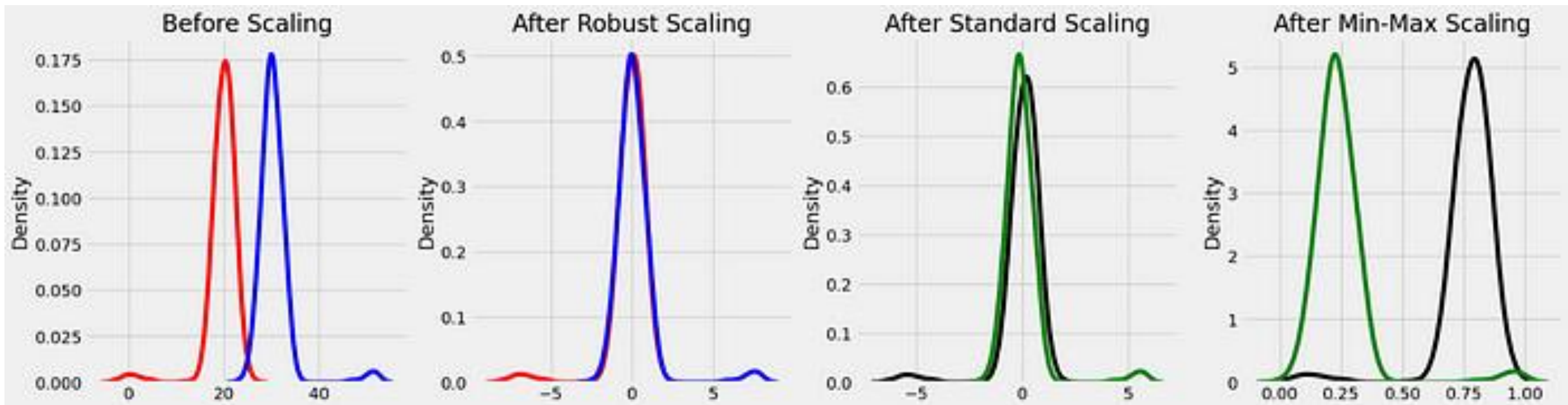
Normalization scales the values to a common range, while standardization standardization transforms the data to have a mean of 0 and standard deviation deviation of 1.

3

Common Scaling Techniques

Min-Max Scaling, Z-Score Normalization, and Robust Scaling are popular popular techniques for data scaling, each with their own advantages and use and use cases.

Data scaling is a technique utilized during the pre-processing phase of various Machine Learning projects to mitigate the impact of data bias on predictions. This method finds widespread application across prediction models, encompassing neural networks and regularized linear models alike.



Techniques for Data Normalization

Min-Max Normalization

Also known as feature scaling, scaling, this technique linearly linearly transforms the data to a to a common range, usually 0 to to 1, to make all features contribute equally to the model. model.

Z-Score Normalization

This method standardizes the the data by subtracting the mean mean and dividing by the standard deviation, resulting in a in a distribution with a mean of 0 of 0 and standard deviation of 1. of 1.

Decimal Scaling Normalization

This technique scales the data by data by moving the decimal point point of the values. The number number of decimal places moved moved is based on the maximum maximum absolute value of the the feature.

Min-Max Normalization (Feature Scaling)

1

Rescaling

Converts features to a common scale

2

Bounded Range

Transforms to a fixed range, typically 0 to 1

3

Preserves Shape

Maintains the original distribution and relationships

$$X'_i = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}}$$

Min-Max normalization, also known as feature scaling, is a key data normalization technique. It rescales features to a common bounded range, typically 0 to 1, while preserving the original shape and relationships in the data. This ensures all features are treated equally during modeling, avoiding bias towards features with larger magnitudes.

Z-Score Normalization (Standardization)

1 Centering Data
Z-score normalization, also known as standardization, centers the data by subtracting the mean and scales it by the standard deviation. This ensures the data has a mean of 0 and a standard deviation of 1.

2 Removing Scale
Standardization removes the original scale of the features, allowing algorithms to treat all features equally. This is important for machine learning models that assume features are on a similar scale.

3 Handling Outliers
Z-score normalization is robust to outliers, as it scales the data based on the overall distribution. This can be beneficial when dealing with data containing extreme values.

$$X'_i = \frac{X_i - \mu}{\sigma} = \frac{X_i - X_{\text{mean}}}{X_{\text{std}}}$$

StandardScaler

Decimal Scaling Normalization

1

Divide by Powers of 10

Decimal scaling normalization divides each feature value by the maximum absolute value of that feature, raised to the power of 10.

2

Preserve Relative Distances

This technique preserves the relative distance between data points while scaling the values to a common range, often between -1 and 1.

3

Ideal for Large Ranges

Decimal scaling is particularly useful when working with features that have vastly different scales, such as measurements in meters vs. kilometers.

Robust Scaling Normalization

Trim Outliers

Remove extreme values that can skew the data

Compute Median

Use the median instead of the mean

$$X'_i = \frac{X_i - X_{med}}{IQR} = \frac{X_i - X_{med}}{X_{75} - X_{25}}$$

Scale to Interquartile Range

Scale the data to the interquartile range

Robust scaling normalization is a technique that provides more resilient normalization in the presence of outliers or extreme values in the data. It involves trimming outliers, computing the median instead of the mean, and scaling the data to the interquartile range. This approach helps to ensure that the normalization is not unduly influenced by a small number of extreme data points.

Quantile Transformation Normalization

1

Rank Data

Quantile transformation ranks the data from smallest to largest, assigning each value a percentile.

2

Normalize Ranks

The ranked data is then transformed to a normal distribution with a mean of 0 and standard deviation of 1.

3

Preserve Relationships

This method preserves the relative relationships between data points, making it useful for skewed or non-normal distributions.

Normalization for Different Data Types

Numerical Data

For numerical features like age or income, techniques like min-max scaling, z-score, and decimal scaling can be effectively applied to normalize the values.

Categorical Data

Categorical variables, such as gender or employment status, can be normalized using one-hot encoding to create binary indicator features.

Text Data

Text data, like reviews or descriptions, can be normalized using techniques like tf-idf or word embeddings to convert text into numerical vectors.

Date/Time Data

Date and time features can be normalized by converting them to numerical representations, such as the number of days since a reference point or the hour of the day.

Advantages of Scaling Data



Improved Model Performance

Scaling data can enhance the performance of machine learning models by ensuring all features are on a similar numerical range.



Balanced Contributions

Scaling helps ensure that all all features contribute equally equally to the model, preventing features with larger larger numerical ranges from dominating the optimization process.



Faster Convergence

Scaling can accelerate the convergence of optimization optimization algorithms like like gradient descent, leading leading to more efficient model training.

Best Practices and Considerations

Understand Data Types

Determine the appropriate normalization technique based on the data type, such as numerical, categorical, or text data. Different techniques work better for different data structures.

Maintain Data Integrity

Ensure that normalization does not introduce errors or distort the underlying data. Carefully monitor the process and validate the results.

Consider Business Context

Understand the business problem and the intended use of the data. Tailor the normalization approach to align with the specific requirements and constraints of the application.

Iterative Refinement

Experiment with different normalization techniques and evaluate their impact. Refine the approach based on performance metrics and feedback from stakeholders.

Feature Selection

EGI SAFITRI, S.MAT., M.SI



Feature Selection

1. Overview
2. Perspectives
3. Aspects
4. Most Representative Methods
5. Related and Advanced Topics
6. Experimental Comparative Analyses

Feature Selection

1. Overview
2. Perspectives
3. Aspects
4. Most Representative Methods
5. Related and Advanced Topics
6. Experimental Comparative Analyses

Overview

- Why we need FS:
 1. to improve performance (in terms of speed, predictive power, simplicity of the model).
 2. to visualize the data for model selection.
 3. To reduce dimensionality and remove noise.
- *Feature Selection* is a process that chooses an optimal subset of features according to a certain criterion.

Overview

Reasons for performing FS may include:

- removing irrelevant data.
- increasing predictive accuracy of learned models.
- reducing the cost of the data.
- improving learning efficiency, such as reducing storage requirements and computational cost.
- reducing the complexity of the resulting model description, improving the understanding of the data and the model.

Feature Selection

1. Overview
2. Perspectives
3. Aspects
4. Most Representative Methods
5. Related and Advanced Topics
6. Experimental Comparative Analyses

Perspectives

1. searching for the best subset of features.
2. criteria for evaluating different subsets.
3. principle for selecting, adding, removing or changing new features during the search.

Perspectives: Search of a Subset of Features

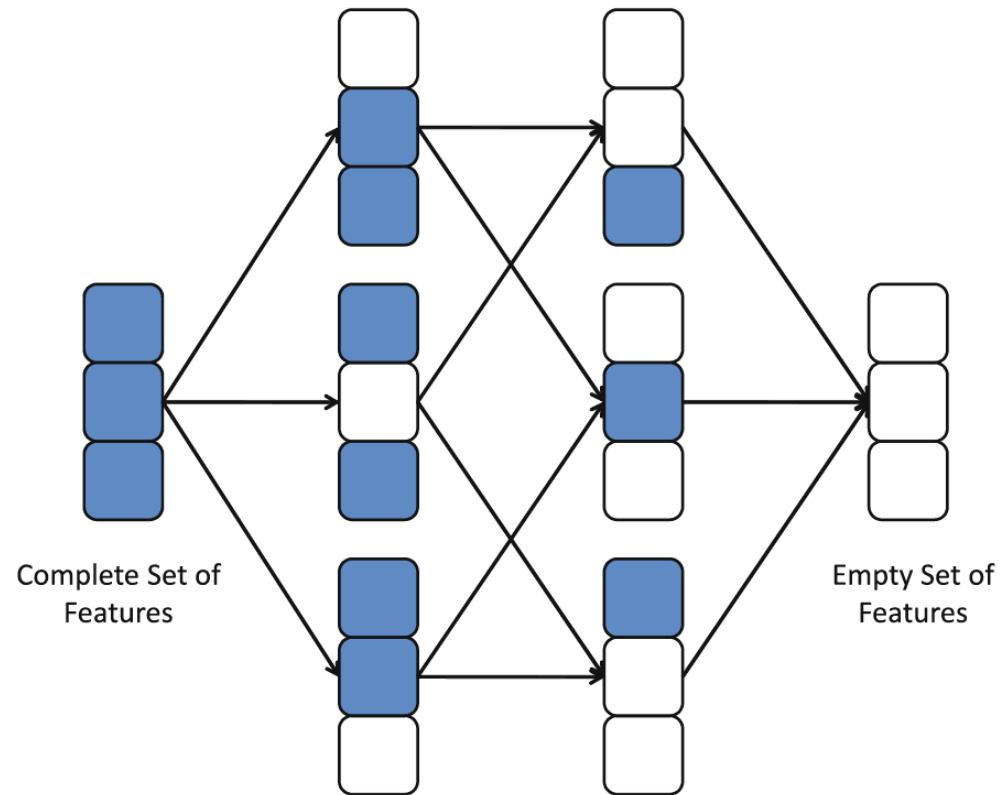
FS can be considered as a search problem, where each state of the search space corresponds to a concrete subset of features selected.

The selection can be represented as a binary array, with each element corresponding to the value 1, if the feature is currently selected by the algorithm and 0, if it does not occur.

There should be a total of 2^M subsets where M is the number of features of a data set.

Perspectives: Search of a Subset of Features

Search
Space:



Perspectives: Search of a Subset of Features

Search Directions:

- **Sequential Forward Generation (SFG):** It starts with an empty set of features S . As the search starts, features are added into S according to some criterion that distinguish the best feature from the others. S grows until it reaches a full set of original features. The stopping criteria can be a threshold for the number of relevant features m or simply the generation of all possible subsets in brute force mode.
- **Sequential Backward Generation (SBG):** It starts with a full set of features and, iteratively, they are removed one at a time. Here, the criterion must point out the worst or least important feature. By the end, the subset is only composed of a unique feature, which is considered to be the most informative of the whole set. As in the previous case, different stopping criteria can be used.

Perspectives: Search of a Subset of Features

Search Directions:

Algorithm 1 Sequential forward feature set generation - SFG.

function SFG(F - full set, U - measure)

initialize: $S = \{\}$

▷ S stores the selected features

repeat

$f = \text{FINDNEXT}(F)$

$S = S \cup \{f\}$

$F = F - \{f\}$

until S satisfies U or $F = \{\}$

return S

end function

Perspectives: Search of a Subset of Features

Search Directions:

Algorithm 2 Sequential backward feature set generation - SBG.

function SBG(F - full set, U - measure)

initialize: $S = \{\}$

▷ S holds the removed features

repeat

$f = \text{GETNEXT}(F)$

$F = F - \{f\}$

$S = S \cup \{f\}$

until S does not satisfy U or $F = \{\}$

return $F \cup \{f\}$

end function

Perspectives: Search of a Subset of Features

Search Directions:

- **Bidirectional Generation (BG):** Begins the search in both directions, performing SFG and SBG concurrently. They stop in two cases: (1) when one search finds the best subset comprised of m features before it reaches the exact middle, or (2) both searches achieve the middle of the search space. It takes advantage of both SFG and SBG.
- **Random Generation (RG):** It starts the search in a random direction. The choice of adding or removing a features is a random decision. RG tries to avoid the stagnation into a local optima by not following a fixed way for subset generation. Unlike SFG or SBG, the size of the subset of features cannot be stipulated.

Perspectives: Search of a Subset of Features

Search Directions:

Algorithm 3 Bidirectional feature set generation - BG.

function BG(F_f , F_b - full set, U - measure)

initialize: $S_f = \{\}$

▷ S_f holds the selected features

initialize: $S_b = \{\}$

▷ S_b holds the removed features

repeat

$f_f = \text{FINDNEXT}(F_f)$

$f_b = \text{GETNEXT}(F_b)$

$S_f = S_f \cup \{f_f\}$

$F_b = F_b - \{f_b\}$

$F_f = F_f - \{f_f\}$

$S_b = S_b \cup \{f_b\}$

until (a) S_f satisfies U or $F_f = \{\}$ or (b) S_b does not satisfy U or $F_b = \{\}$

return S_f if (a) or $F_b \cup \{f_b\}$ if (b)

end function

Perspectives: Search of a Subset of Features

Search Directions:

Algorithm 4 Random feature set generation - RG.

function RG(F - full set, U - measure)

initialize: $S = S_{best} = \{\}$

▷ S - subset set

initialize: $C_{best} = \#(F)$

▷ $\#$ - cardinality of a set

repeat

$S = \text{RANDGEN}(F)$

$C = \#(S)$

if $C \leq C_{best}$ and S satisfies U **then**

$S_{best} = S$

$C_{best} = C$

end if

until some stopping criterion is satisfied

return S_{best}

▷ Best set found so far

end function

Perspectives: Search of a Subset of Features

Search Strategies:

- **Exhaustive Search:** It corresponds to explore all possible subsets to find the optimal ones. As we said before, the space complexity is $O(2^M)$. If we establish a threshold m of minimum features to be selected and the direction of search, the search space is, independent of the forward or backward generation. Only exhaustive search can guarantee the optimality. Nevertheless, they are also impractical in real data sets with a high M .
- **Heuristic Search:** It employs heuristics to carry out the search. Thus, it prevents brute force search, but it will surely find a non-optimal subset of features. It draws a path connecting the beginning and the end of the previous Figure, such in a way of a depth-first search. The maximum length of this path is M and the number of subsets generated is $O(M)$. The choice of the heuristic is crucial to find a closer optimal subset of features in a faster operation.

Perspectives: Search of a Subset of Features

Search Strategies:

- **Nondeterministic Search:** Complementary combination of the previous two. It is also known as random search strategy and can generate best subsets constantly and keep improving the quality of selected features as time goes by. In each step, the next subset is obtained at random.
- it is unnecessary to wait until the search ends.
- we do not know when the optimal set is obtained, although we know which one is better than the previous one and which one is the best at the moment.

Perspectives: Selection Criteria

- Information Measures.

- Information serves to measure the uncertainty of the receiver when she/he receives a message.
- Shannon's Entropy:

$$-\sum_i P(c_i) \log_2 P(c_i).$$

- Information gain:

$$IG(A) = I(D) - \sum_{j=1}^p \frac{|D_j|}{|D|} I(D_j^A)$$

Perspectives: Selection Criteria

- Distance Measures.
 - Measures of separability, discrimination or divergence measures . The most typical is derived from distance between the class conditional density functions.

	Mathematical form
Euclidean distance	$D_e = \left\{ \sum_{i=1}^m (x_i - y_i)^2 \right\}^{\frac{1}{2}}$
City-block distance	$D_{cb} = \sum_{i=1}^m x_i - y_i $
Cebyshev distance	$D_{ch} = \max_i x_i - y_i $
Minkowski distance of order m	$D_M = \left\{ \sum_{i=1}^m (x_i - y_i)^m \right\}^{\frac{1}{m}}$
Quadratic distance Q , positive definite	$D_q = \sum_{i=1}^m \sum_{j=1}^m (x_i - y_i) Q_{ij} (x_j - y_j)$
Canberra distance	$D_{ca} = \sum_{i=1}^m \frac{ x_i - y_i }{x_i + y_i}$
Angular separation	$D_{as} = \frac{\sum_{i=1}^m x_i \cdot y_i}{\left[\sum_{i=1}^m x_i^2 \sum_{i=1}^m y_i^2 \right]^{\frac{1}{2}}}$

Perspectives: Selection Criteria

- Dependence Measures.
 - known as measures of association or correlation.
 - Its main goal is to quantify how strongly two variables are correlated or present some association with each other, in such way that knowing the value of one of them, we can derive the value for the other.
 - *Pearson correlation coefficient*:

$$\rho(X, Y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\left[\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2 \right]^{\frac{1}{2}}}$$

Perspectives: Selection Criteria

- Consistency Measures.
 - They attempt to find a minimum number of features that separate classes as the full set of features can.
 - They aim to achieve $P(C | \text{FullSet}) = P(C | \text{SubSet})$.
 - An inconsistency is defined as the case of two examples with the same inputs (same feature values) but with different output feature values (classes in classification).

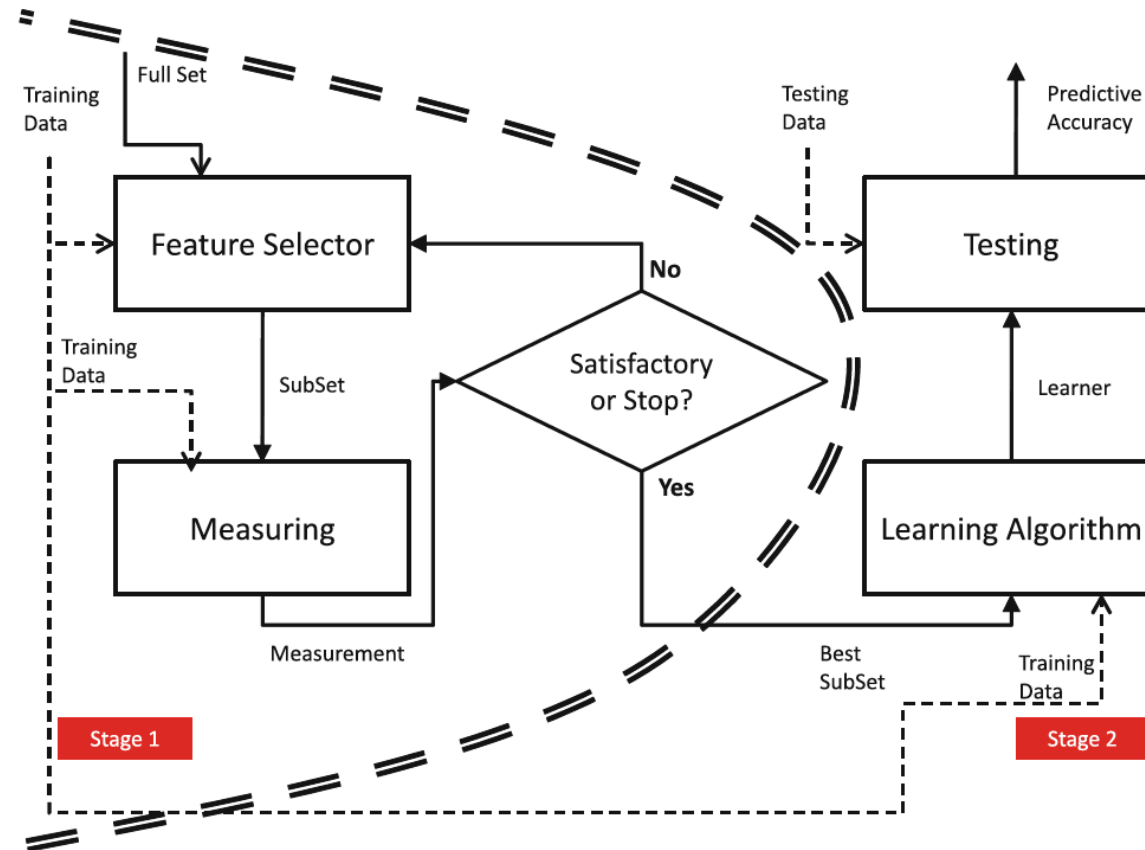
Perspectives: Selection Criteria

- Accuracy Measures.
 - This form of evaluation relies on the classifier or learner. Among various possible subsets of features, the subset which yields the best predictive accuracy is chosen

	Mathematical form
Accuracy	$\frac{tp+fp}{tp+tn+fp+fn}$
Error rate	1 - Accuracy
Chi-squared	$\frac{n(fp \times fn - tp \times tn)^2}{(tp+fp)(tp+fn)(fp+tn)(tn+fn)}$
Information gain	$e(tp + fn, fp + tn) - \frac{(tp+fp)e(tp,fp)+(tn+fn)e(fn,tn)}{tp+fp+tn+fn}$ where $e(x, y) = -\frac{x}{x+y} \log_2 \frac{x}{x+y} - \frac{y}{x+y} \log_2 \frac{y}{x+y}$
Odds ratio	$\frac{tpr}{1-tpr} / \frac{fpr}{1-fpr} = \frac{tp \times tn}{fp \times fn}$
Probability ratio	$\frac{tpr}{fpr}$

Perspectives

Filters:



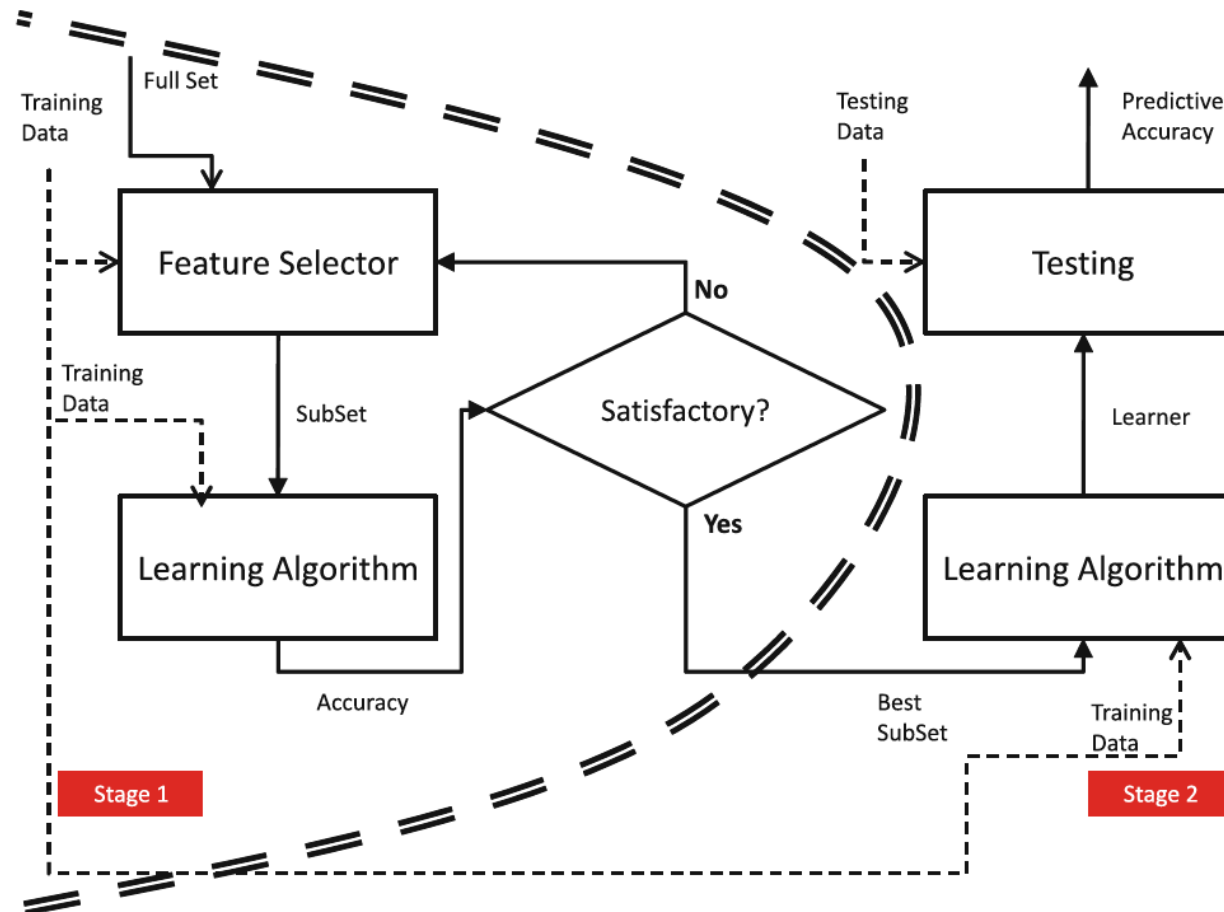
Perspectives

Filters:

- measuring uncertainty, distances, dependence or consistency is usually cheaper than measuring the accuracy of a learning process. Thus, filter methods are usually faster.
- it does not rely on a particular learning bias, in such a way that the selected features can be used to learn different models from different DM techniques.
- it can handle larger sized data, due to the simplicity and low time complexity of the evaluation measures.

Perspectives

Wrappers:



Perspectives

Wrappers:

- can achieve the purpose of improving the particular learner's predictive performance.
- usage of internal statistical validation to control the overfitting, ensembles of learners and hybridizations with heuristic learning like Bayesian classifiers or Decision Tree induction.
- filter models cannot allow a learning algorithm to fully exploit its bias, whereas wrapper methods do.

Perspectives

Embedded FS:

- similar to the wrapper approach in the sense that the features are specifically selected for a certain learning algorithm, but in this approach, the features are selected during the learning process.
- they could take advantage of the available data by not requiring to split the training data into a training and validation set; they could achieve a faster solution by avoiding the re-training of a predictor for each feature subset explored.

Feature Selection

1. Overview
2. Perspectives
3. Aspects
4. Most Representative Methods
5. Related and Advanced Topics
6. Experimental Comparative Analyses

Aspects: Output of Feature Selection

Feature Ranking Techniques:

- we expect as the output a ranked list of features which are ordered according to evaluation measures.
- they return the relevance of the features.
- For performing actual FS, the simplest way is to choose the first m features for the task at hand, whenever we know the most appropriate m value.

Aspects: Output of Feature Selection

Feature Ranking Techniques:

Algorithm 5 A univariate feature ranking algorithm.

```
function RANKING_ALGORITHM( $x$  - features,  $U$  - measure)
  initialize: list  $L = \{\}$                                      ▷  $L$  stores ordered features
  for each feature  $x_i, i \in \{1, \dots, M\}$  do
     $v_i = \text{COMPUTE}(x_i, U)$ 
    position  $x_i$  into  $L$  according to  $v_i$ 
  end for
  return  $L$  in decreasing order of feature relevance.
end function
```

Aspects: Output of Feature Selection

Minimum Subset Techniques:

- The number of relevant features is a parameter that is often not known by the practitioner.
- There must be a second category of techniques focused on obtaining the minimum possible subset without ordering the features.
- whatever is relevant within the subset, is otherwise irrelevant.

Aspects: Output of Feature Selection

Minimum Subset Techniques:

Algorithm 6 A minimum subset algorithm.

```
function MIN- SET ALGORITHM(x - features, U - measure)
  initialize:  $L = \{\}$ , stop = false
  repeat
     $S_k = \text{SUBSETGENERATE}(x)$ 
    if LEGITIMACY( $S_k$ , U) is true and  $\#(S_k) < \#(S)$  then
       $S = S_k$ 
    end if
  until stop = true
  return S - the minimum subset of features
end function
```

▷ *S* holds the minimum set

▷ **stop** can be set here

▷ *S* is replaced by S_k

Aspects: Evaluation

Goals:

- **Inferability:** For predictive tasks, considered as an improvement of the prediction of unseen examples with respect to the direct usage of the raw training data.
- **Interpretability:** Given the incomprehension of raw data by humans, DM is also used for generating more understandable structure representation that can explain the behavior of the data.
- **Data Reduction:** It is better and simpler to handle data with lower dimensions in terms of efficiency and interpretability.

Aspects: Evaluation

We can derive three assessment measures from these three goals:

- **Accuracy**
- **Complexity**
- **Number of Features Selected**
- **Speed of the FS method**
- **Generality of the features selected**

Aspects: Drawbacks

The resulted subsets of many models of FS are strongly dependent on the training set size.

It is not true that a large dimensionality input can always be reduced to a small subset of features because the objective feature is actually related with many input features and the removal of any of them will seriously effect the learning performance.

A backward removal strategy is very slow when working with large-scale data sets. This is because in the firsts stages of the algorithm, it has to make decisions funded on huge quantities of data.

In some cases, the FS outcome will still be left with a relatively large number of relevant features which even inhibit the use of complex learning methods.

Aspects: Using Decision Trees for FS

Decision trees can be used to implement a trade-off between the performance of the selected features and the computation time which is required to find a subset.

Decision tree inducers can be considered as anytime algorithms for FS, due to the fact that they gradually improve the performance and can be stopped at any time, providing sub-optimal feature subsets.

Feature Selection

1. Overview
2. Perspectives
3. Aspects
4. Most Representative Methods
5. Related and Advanced Topics
6. Experimental Comparative Analyses

Most Representative Methods

Three major components to categorize combinations:

- **Search Direction**
- **Search Strategy**
- **Evaluation Measure**

Search direction	Evaluation measure	Search strategy		
		Exhaustive	Heuristic	Nondeterministic
Forward	Probability	C1	C7	–
	Consistency	C2	C8	–
	Accuracy	C3	C9	–
Backward	Probability	C4	C10	–
	Consistency	C5	C11	–
	Accuracy	C6	C12	–
Random	Probability	–	C13	C16
	Consistency	–	C14	C17
	Accuracy	–	C15	C18

Most Representative Methods

Exhaustive Methods

Cover the whole search space.

Six Combinations (C1-C6).

- Focus method: C2.
- Automatic Branch and Bound (ABB): C5.
- Best First Search (BFS): C1.
- Beam Search: C3.
- Branch and Bound (BB): C4.

Most Representative Methods

Exhaustive Methods

Algorithm 7 Focus algorithm.

```
function FOCUS( $F$  - all features in data  $D$ ,  $U$  - inconsistency rate as evaluation measure)
  initialize:  $S = \{\}$ 
  for  $i = 1$  to  $M$  do
    for each subset  $S$  of size  $i$  do
      if  $\text{CALU}(S, D) = 0$  then                                      $\triangleright$   $\text{CALU}(S, D)$  returns inconsistency
        return  $S$  - a minimum subset that satisfies  $U$ 
      end if
    end for
  end for
end function
```

Most Representative Methods

Heuristic Methods

They do not have any expectations of finding an optimal subset with a rapid solution.

Nine Combinations (C7-C15).

- Use a DM algorithm for FS: C12.
- Wrapper Sequential Forward Selection: C9.
- SetCover: C8.
- Heuristic search algorithm and in each sub-search space: C13-C15.
- **MIFS: C10.**

Most Representative Methods

Heuristic Methods

Algorithm 8 MIFS algorithm.

```
function MIFS( $F$  - all features in data,  $S$  - set of selected features,  $k$  - desired size of  $S$ ,  $\beta$  -  
  regulator parameter)  
  initialize:  $S = \{\}$   
  for each feature  $f_i$  in  $F$  do  
    Compute  $I(C, f_i)$   
  end for  
  Find  $f_{max}$  that maximizes  $I(C, f)$   
   $F = F - \{f_{max}\}$   
   $S = S \cup f_{max}$   
  repeat  
    for all couples of features ( $f_i \in F, s_j \in S$ ) do  
      Compute  $I(f_i, s_j)$   
    end for  
    Find  $f_{max}$  that maximizes  $I(C, f) - \beta \sum_{s \in S} I(f_i, s_j)$   
     $F = F - \{f_{max}\}$   
     $S = S \cup f_{max}$   
  until  $|S| = k$   
  return  $S$   
end function
```

Most Representative Methods

Nondeterministic Methods

They add or remove features to and from a subset without a sequential order.

Three Combinations (C16-C18).

- Simulated Annealing / Genetic Algorithms are the most common techniques.
- **LVF: C17.**
- **LVW: C18.**

Most Representative Methods

Nondeterministic Methods

Algorithm 9 LVF algorithm.

```
function LVF( $D$  - a data set with  $M$  features,  $U$  - the inconsistency rate,  $maxTries$  - stopping
criterion,  $\gamma$  - an allowed inconsistency rate)
  initialize: list  $L = \{\}$  ▷  $L$  stores equally good sets
   $C_{best} = M$ 
  for  $maxTries$  iterations do
     $S = \text{RANDOMSET}(\text{seed})$ 
     $C = \#(S)$  ▷  $\#$  - the cardinality of  $S$ 
    if  $C < C_{best}$  and  $\text{CALU}(S, D) < \gamma$  then
       $S_{best} = S$ 
       $C_{best} = C$ 
       $L = \{S\}$  ▷  $L$  is reinitialized
    else if  $C = C_{best}$  and  $\text{CALU}(S, D) < \gamma$  then
       $L = \text{APPEND}(S, L)$ 
    end if
  end for
  return  $L$  ▷ all equivalently good subsets found by LVF
end function
```

Most Representative Methods

Nondeterministic Methods

Algorithm 10 LVW algorithm.

```
function LVW( $D$  - a data set with  $M$  features,  $LA$  - a learning algorithm,  $maxTries$  - stopping
criterion,  $F$  - a full set of features)
  initialize: list  $L = \{\}$  ▷  $L$  stores sets with equal accuracy
   $A_{best} = ESTIMATE(D, F, LA)$ 
  for  $maxTries$  iterations do
     $S = RANDOMSET(seed)$ 
     $A = ESTIMATE(D, S, LA)$  ▷ # - the cardinality of  $S$ 
    if  $A > A_{best}$  then
       $S_{best} = S$ 
       $A_{best} = A$ 
       $L = \{S\}$  ▷  $L$  is reinitialized
    else if  $A = A_{best}$  then
       $L = APPEND(S, L)$ 
    end if
  end for
  return  $L$  ▷ all equivalently good subsets found by LVW
end function
```

Most Representative Methods

Feature Weighting Methods

Provide weights to features, also can be used for FS.

Relief (binary) and ReliefF (multiple classes).

Algorithm 11 Relief algorithm.

```
function RELIEF( $x$  - features,  $m$  - number of instances sampled,  $\tau$  - relevance threshold)
  initialize:  $w = 0$ 
  for  $i = 1$  to  $m$  do
    randomly select an instance  $I$ 
    find nearest-hit  $H$  and nearest-miss  $J$ 
    for  $j = 1$  to  $M$  do
       $w(j) = w(j) - dist(j, I, H)^2/m + dist(j, I, J)^2/m$   $\triangleright dist$  is a distance function
    end for
  end for
  return  $w$  greater than  $\tau$ 
end function
```

Feature Selection

1. Overview
2. Perspectives
3. Aspects
4. Most Representative Methods
5. Related and Advanced Topics
6. Experimental Comparative Analyses

Related and Advanced Topics

Leading and Recent FS Techniques

The related literature is huge, quite chaotic and difficult to understand or categorize the differences among the hundreds of algorithms published.

Focus attention on the main ideas that lead to updates and improvements.

Related and Advanced Topics

Leading and Recent FS Techniques

Modifications of classical FS:

- MIFS-U, MIFS based on Parzen window, mRMR, NMIFS.
- Several Relief modifications: Iterative-RELIEF.

Separability and Similarity:

- Kernel class separability.
- Common subspace measure and Fisher subspace measure combination.
- The redundancy-constrained FS (RCFS).
- Many more...

Related and Advanced Topics

Leading and Recent FS Techniques

Use of meta-heuristics:

- Genetic algorithms.
- Tabu search.
- Hybridizations between genetic algorithms and local searches.

Rough Sets theory:

- evaluation criteria based on reducts and approximations.
- Fuzzy Rough FS (FRFS).

Related and Advanced Topics

Leading and Recent FS Techniques

Fusion of filters and wrappers:

- Evaluation criteria merging dependency, coefficients of correlations and error estimation by KNN.
- GAMIFS: genetic algorithm to form an hybrid filter/wrapper.

Extremely high-dimensional data:

- Reduction of the FS task to a quadratic optimization problem: QPFS.
- Big Data solutions.

Related and Advanced Topics

Feature Extraction

Find new features that are calculated as a function of the original features.

Dimensionality Reduction is done by mapping a multidimensional space into a space of fewer dimensions.

It is another name given to Space Transformations.

Related and Advanced Topics

Feature Construction

Attach to the algorithms some mechanism to compound new features from the original ones endeavouring to improve accuracy and the decrease in model complexity.

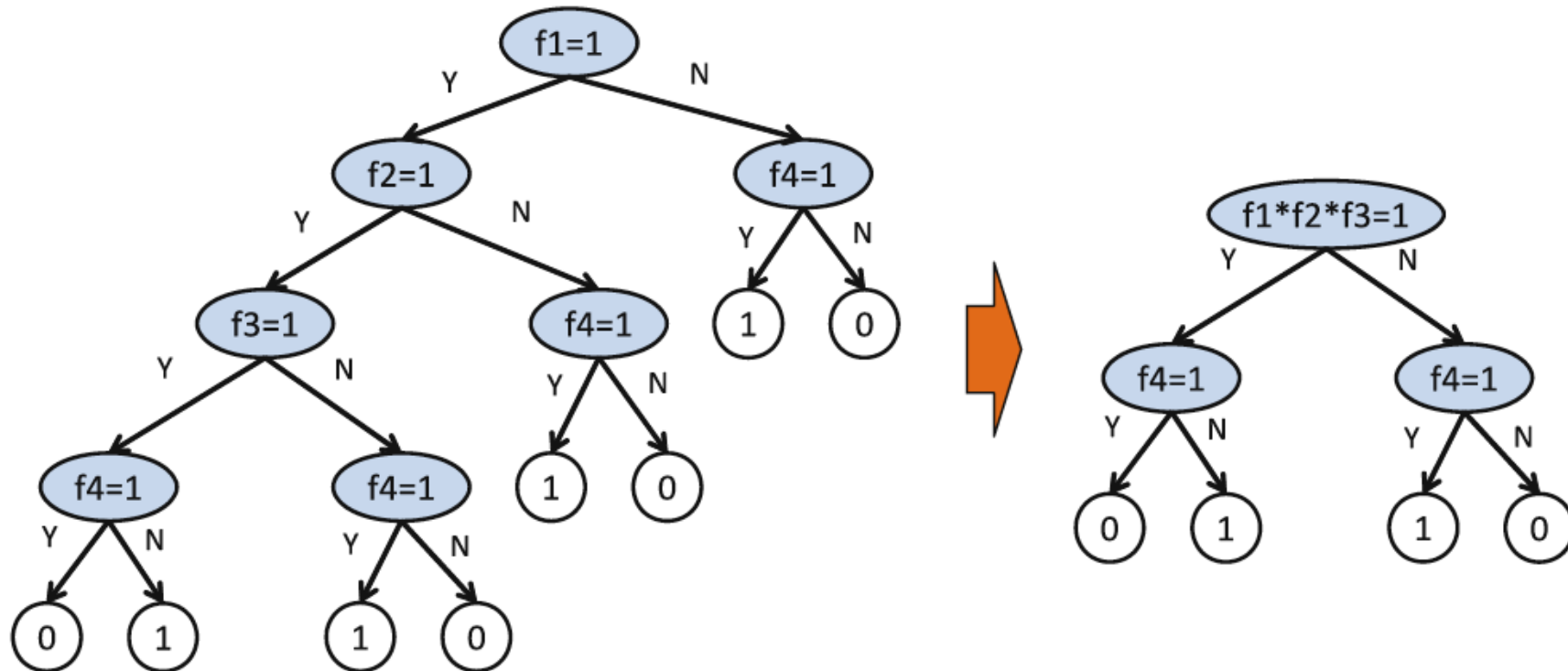
They have been extensively applied on separate-and-conquer predictive learning approaches.

Related and Advanced Topics

Feature Construction

Constructive operators:

- Product, inequality, maximum, minimum, average, addition, subtraction, division, count, ...



Feature Selection

1. Overview
2. Perspectives
3. Aspects
4. Most Representative Methods
5. Related and Advanced Topics
6. Experimental Comparative Analyses

Experimental Comparative Analyses

Summary of some major studies:

- With 1-NN, the main conclusions point to the use of the bidirectional approaches for small and medium scale data sets.
- Regarding evaluation measures, the inconsistency criterion was emphasized in several aspects.
- Evaluation measure + stopping criterion: information theory based functions obtain better accuracy results; no cutting criterion can be generally recommended.
- On synthetic data, ReliefF turned out to be the best option independent of the particulars of the data.