Data Preparation

Nikola Milikić
nikola.milikic@fon.bg.ac.rs

Jelena Jovanović
jeljov@fon.bg.ac.rs

Normalization

Normalization is the process of rescaling the values of an attribute to a specific value range, typically [0 ,1]

In Weka:

- it is implemented as a type of Filter
- relevant class: *weka.filters.unsupervised.attribute.Normalize*

Standardization

Standardization is the process of rescaling the values of an attribute so that:

- the mean value of the attribute is 0
- standard deviation is 1

In Weka:

- it is implemented as a type of Filter
- relevant class: *weka.filters.unsupervised.attribute.Standardize*

Attribute discretization

Discretization is the process of transforming a numeric attribute into a nominal one, by splitting the attribute's range of values into several distinct groups (bins)

Common approaches:

- Unsupervised:
  - Equal-width binning
  - Equal-frequency binning
- Supervised – takes into account the class attribute

Equal- Width Binning

Equal-width binning divides the range of values into N intervals (bins) of the same width

$$\boxed{\text{width} = (\text{max\_value} - \text{min\_value}) \;/\; N}$$

Example: if the range of values is [0, 100], and we want to create 5 intervals (bins), then:

$$\text{width} = (100 - 0) \;/\; 5 = 20$$

Intervals are: [0-20], (20-40], (40-60], (60-80], (80-100]

Usually, the first and the final interval (bin) are expanded in order to include possible values outside the original range

Equal-frequency binning

Equal-frequency binning (or equal-length binning) divides the attribute's range of values into N intervals where each interval (bin) has the same number of instances

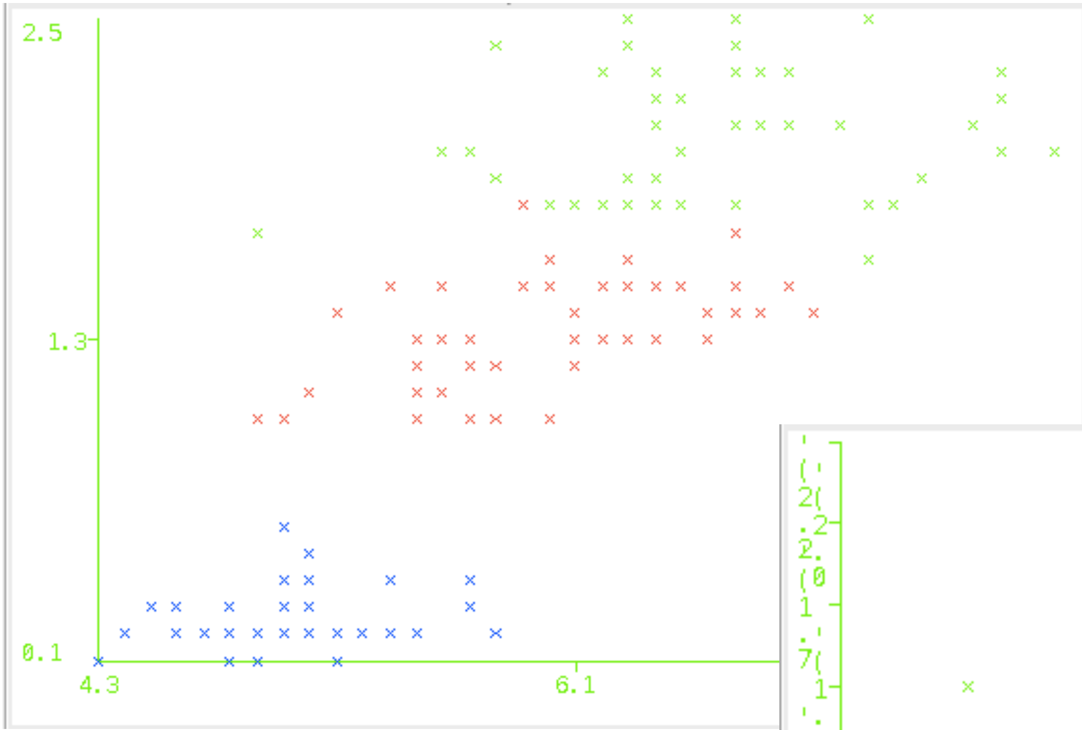
Example: Suppose we want to apply equal-frequency binning to an attribute with the following values:

5, 7, 12, 35, 65, 82, 84, 88, 90, 95

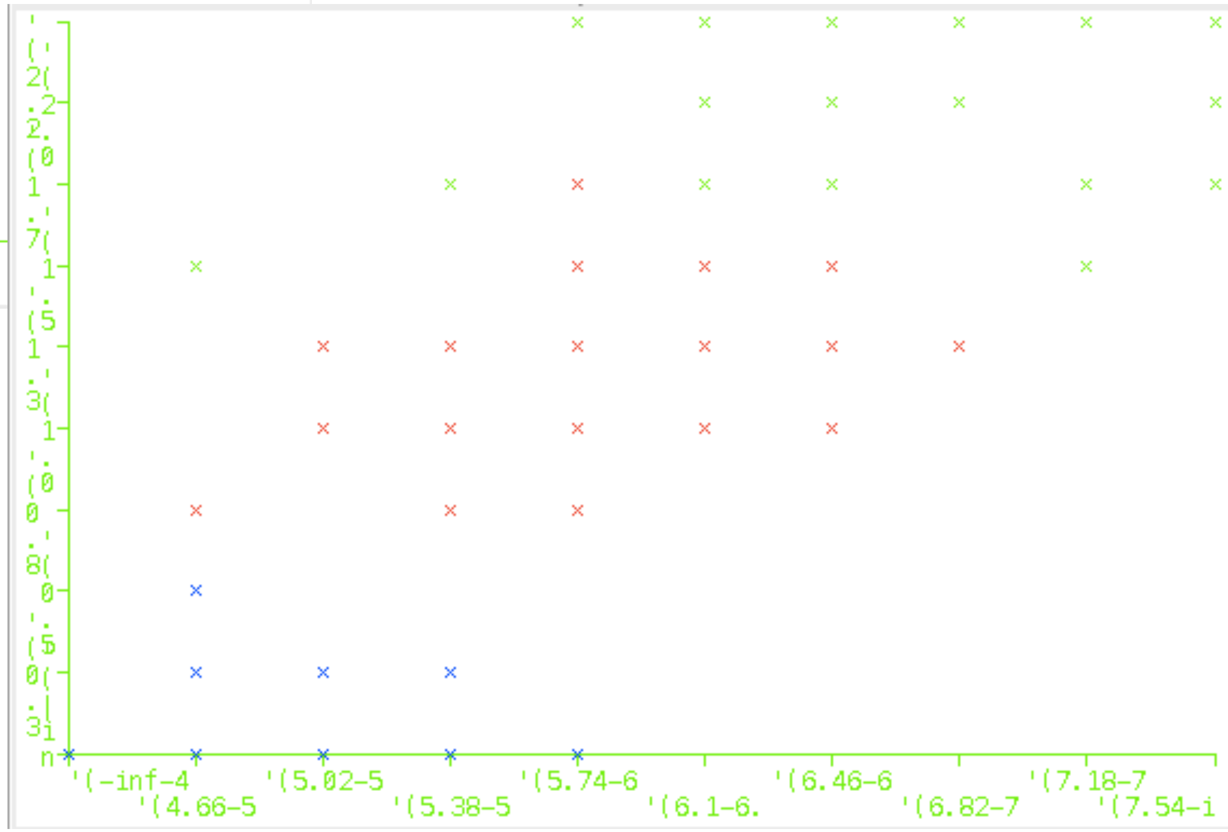
and we want to create 5 bins; then, each bin will have 2 instances:

5, 7, | 12, 35, | 65, 82, | 84, 88, | 90, 95

Data, before and after discretization



Before



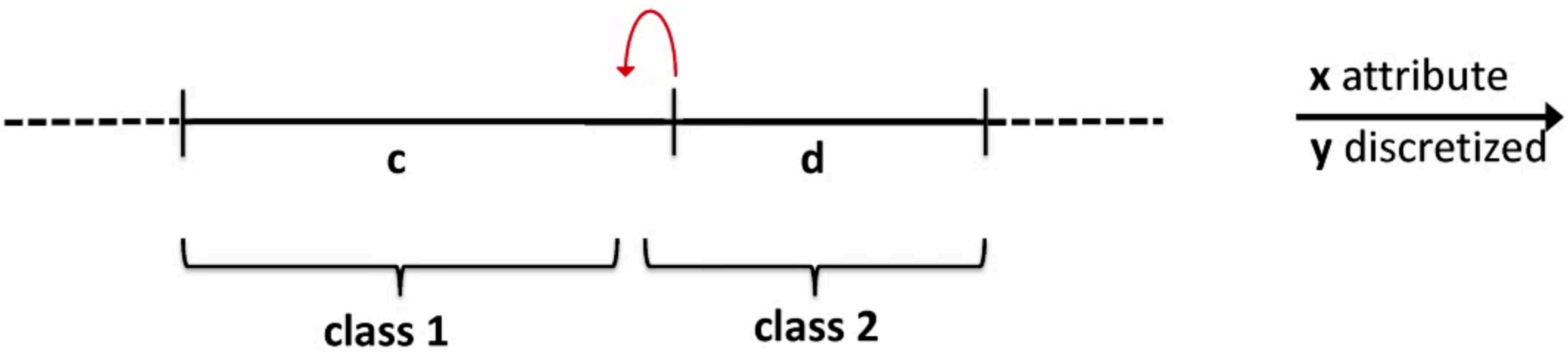
After

Discretization in Weka

- Weka defines specific *Filter* for unsupervised discretization : *weka.filters.unsupervised.attribute.Discretize*
- Equal-width binning is the default option
- Options for customizing the discretization:
  - *attributeIndices* (or *attributeIndicesArray*) – we can choose the attributes to be discretized, by providing their indexes; by specifying “first-last”, we state that we want to discretize all the attributes
  - *bins* – set the desired number of intervals (bins)
  - *useEqualFrequency* - *false* by default; *true* if we want to use Equal Frequency binning

Supervised discretization

- What if all instances in one bin (*c*) belong to Class 1, and all instances in the adjacent bin (*d*) belong to Class 2 except for the first instance, which is of the Class 1?



- Supervised discretization takes the class values in account; therefore, the first instance from bin *d* will be moved to bin *c*

Supervised discretization

- Uses the entropy heuristic
- Consider the *temperature* attribute in the example *weather.numeric.arff*

64	65	68	69	70	71	72	75	80	81	83	85
yes	no	yes	yes	yes	no	no	yes	no	yes	yes	no
						yes	yes				

4 yes, 1 no | 5 yes, 4 no

entropy = 0.934 bits

- Choose split points that with lead to the smallest entropy (largest information gain)

64	65	68	69	70	71	72	75	80	81	83	85
yes	no	yes	yes	yes	no	no	yes	no	yes	yes	no
						yes	yes				

Supervised discretization in Weka

Weka's class for supervised discretization:
weka.filters.supervised.attribute.Discretize

Note: the name of the class (Discretize) is the same as for unsupervised discretization; the difference is in the packages where the two classes are defined:

*weka.filters.**supervised**.attribute.Discretize*

*weka.filters.**unsupervised**.attribute.Discretize*

Attribute Selection

Attribute Selection (or Feature Selection) is the process of choosing a subset of relevant attributes that will be used for building a machine learning model

It is applied when we suspect that our dataset contains attributes that are either redundant or irrelevant

- Redundant attributes are the ones that do not provide more information than some other attributes in our dataset
- Irrelevant attributes are the ones that are useless in the context of the task at hand

Attribute Selection

Excessive attributes can degrade the performance of the model; thus their removal leads to the following advantages:

- the resulting model is 'leaner', and thus less prone to overfitting => has higher generalization power
- shorter training time
- increased readability and comprehensibility of the model

Attribute Selection

If the problem is known and well-understood, the best way to select attributes is to do it through manual inspection

In other cases, automated approaches are required; they also tend to give good results

Approaches to Attribute Selection in Weka

Filter methods

Perform selection based on the general estimation of attributes ability to distinguish instances in the training set

Examples:

- look for the set of attributes that are highly correlated with the class attribute but not strongly correlated with one another
- use one ML method – e.g., decision tree or linear regression – to select relevant attributes, and then the selected attribute set is used for building another kind on ML model (e.g, NB)

Approaches to Attribute Selection in Weka

Wrapper methods

- attribute subsets are evaluated by using the ML algorithm that will be used for building the model
- the name Wrapper comes from the fact that the process of selection is 'wrapped' around the ML algorithm
- the chosen subset of attributes is the one for which the selected algorithm gives the best results

Using Wrapper method in Weka

When doing *wrapped* attribute selection in Weka, we need to:

- use the *WrapperSubsetEval* Weka class
- specify 2 key elements:
    - the classifier we want to use (e.g., Naïve Bayes)
    - the method to be used for searching the space of attributes (e.g., BestFirst)

Search Methods in Attribute Selection

- Exhaustive search – considers all possible combinations of attributes from the datasets
  - even for a small number of attributes, there will be a large number of possible combinations; e.g., for 6 attributes, there will be 62 combinations
  - therefore, this method is often inefficient

Search Methods in Attribute Selection

Best First search:

- most frequently used search option
- 3 possible search directions:
    - starting with an empty set of attributes and searching *forward*, or
    - starting with the full set of attributes and search *backward*, or
    - starting at any point and search *in both directions* (considers all possible single attribute additions and deletions at a given point)
- *searchTermination* parameter determines how many non-improving attribute subsets to allow before terminating the search

Recommendations and credits

"Data Mining with Weka" and "More Data Mining with Weka": MOOCs from the University of Waikato. A self-paced session of "Data Mining with Weka" runs until 23 October June 2015.

- Link: https://www.youtube.com/user/WekaMOOC/

(Anonymous) survey for your comments and suggestions:

http://goo.gl/cqdp3I