

Assignment 1 - NAÏVE BAYES

Authors:

Mateusz Magda 179346@student.pwr.wroc.pl BTH ID: 900902P115

Paweł Łabuda 179211@student.pwr.wroc.pl BTH ID: 900328P418

Class NaiveBayes – a class extending Weka's Classifier abstract class. It implements buildClassifier and classifyInstance method. Its class properties are list of AttributeConditionalProbability objects of each nominal non-class attribute, reference to the class attribute and reference to the ClassAttributeProbability class. Method buildClassifier **initializes** all required probabilities. Method classifyInstance calculates probabilities of affiliation to each class attribute value and chooses the highest one.

Class ClassAttributeValue – a class storing class attribute probability needed to calculate probability of affiliation of an instance to a particular class. Its class properties are set of instances and reference to the class attribute. For each class attribute value probability is calculated based on set of instances. You can get particular probability by call getProbability(String classAttributeValue) method.

Class AttributeConditionalProbability – a class storing conditional probability of affiliation instance to the class attribute value for each attribute value. Its class properties are set of instances, reference to attribute, reference to class attribute and reference to ConditionalProbabilityTable. It has method calculateConditionalProbabilities to calculate all probabilities basing on set of instances. You can get particular probability by calling a method getConditionalProbability(String attributeValue, String classAttributeValue). ConditionalProbabilityTable class is used to particular conditional probabilities.

Class ConditionalProbabilityTable – a class used as a structure to store particular conditional probabilities. It is helpful when you want to put particular probability after calculating it and use it further to calculate probability of affiliation of an instance to the particular class attribute value.

Class KMeans – class used to discretize numeric attribute. It gets a set of instances and an attribute to discretize. It returns a new set of instances with discretized attribute. Algorithm used to discretize look as follows:

1. Sort the instances by discretizing attribute value.
2. Distribute instances to $(\text{int}) \sqrt{(\text{numInstances})}$ folds. The earlier fold the lower discretizing attribute value.
3. Run k-means algorithm to make folds more appropriate – it means distribute instances to fold with closer values of an attribute that is discretized.
4. Merge similar folds.
5. Discretize values of the attribute by setting its name as the average discretized attribute value in fold.

cc – correctly classified, ic – incorrectly classified

	weather.nominal.arff		weather.numeric.arff	
	Our classifier	Weka	Our classifier	Weka
Learning set validation	13/14(92.85%) – cc 1/14(7.14%) – ic	13/14(92.85%) – cc 1/14(7.14%) – ic	11/14(78.57%) – cc 3/14(21.42%) – ic	11/14(78.57%) – cc 3/14(21.42%) – ic
Cross validation	8/14(57.14%) – cc 6/14(42.85%) – ic	8/14(57.14%) – cc 6/14(42.85%) – ic	7/14(50%) – cc 7/14(50%) – ic	8/14(57.14%) – cc 6/14(42.85%) – ic

