# weka.classifiers.trees.J48

SYNOPSIS

Class for generating a pruned or unpruned C4.5 decision tree. For more information, see

Ross Quinlan (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA.

* + development of C4.5, which can deal with numeric attributes, missing values, and noisy data
* **Advantages**:
  + Inexpensive to construct
  + Extremely fast at classifying unknown records
  + Easy to interpret for small-sized trees
  + Accuracy is comparable to other classification techniques for many data sets

**OPTIONS**

binarySplits -- Whether to use binary splits on nominal attributes when building the trees.

confidenceFactor -- The confidence factor used for pruning (smaller values incur more pruning).

debug -- If set to true, classifier may output additional info to the console.

minNumObj -- The minimum number of instances per leaf.

numFolds -- Determines the amount of data used for reduced-error pruning. One fold is used for pruning, the rest for growing the tree.

reducedErrorPruning -- Whether reduced-error pruning is used instead of C.4.5 pruning.

saveInstanceData -- Whether to save the training data for visualization.

seed -- The seed used for randomizing the data when reduced-error pruning is used.

subtreeRaising -- Whether to consider the subtree raising operation when pruning.

unpruned -- Whether pruning is performed.

useLaplace -- Whether counts at leaves are smoothed based on Laplace.

1. *Data Mining: Practical Machine Learning Tools and Techniques (3rdedition) /* Ian Witten, Eibe Frank; Elsevier, 2011, Chapter 4
2. [Naive-Bayes and Nearest Neighbour Techniques](http://www.dbmsmag.com/9807m07.html), document available from AUT online

# weka.classifiers.bayes.NaiveBayes

* Continuous (ratio) data can be handled by *binning –* done automatically by most DM products (including Weka)
* Naïve Bayes works well in practise even though attributes may not be statistically independent
* Robust technique as it can deal with unknown/missing values
* Models developed with Naive Bayes have good explanatory power, just as with Decision Trees
* considers all attributes
* Assumes that all attributes are statistically independent of each other – not always true
* However, most efficient classification technique as it makes only *one* pass through training data
* NB is one of more simple and effective classifiers
* NB has a very strong unrealistic independence assumption:
* all the attributes are conditionally independent given the value of class

SYNOPSIS

Class for a Naive Bayes classifier using estimator classes. Numeric estimator precision values are chosen based on analysis of the training data. For this reason, the classifier is not an UpdateableClassifier (which in typical usage are initialized with zero training instances) -- if you need the UpdateableClassifier functionality, use the NaiveBayesUpdateable classifier. The NaiveBayesUpdateable classifier will use a default precision of 0.1 for numeric attributes when buildClassifier is called with zero training instances.

For more information on Naive Bayes classifiers, see

George H. John, Pat Langley: Estimating Continuous Distributions in Bayesian Classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, 338-345, 1995.

OPTIONS

debug -- If set to true, classifier may output additional info to the console.

displayModelInOldFormat -- Use old format for model output. The old format is better when there are many class values. The new format is better when there are fewer classes and many attributes.

useKernelEstimator -- Use a kernel estimator for numeric attributes rather than a normal distribution.

**useSupervisedDiscretization** -- Use supervised discretization to convert numeric attributes to nominal ones.

NAME

# weka.classifiers.bayes.BayesNet

* reducing the bias resulting from the modeling error, by relaxing the attribute independence assumption
* Bayesian Networks (BNs) graphically represent the joint probability distribution of a set **X** of random variables in a problem domain
* an **efficient** and **effective** representation of the joint probability distribution of a set of random variables

SYNOPSIS

Bayes Network learning using various search algorithms and quality measures.

Base class for a Bayes Network classifier. Provides datastructures (network structure, conditional probability distributions, etc.) and facilities common to Bayes Network learning algorithms like K2 and B.

For more information see:

http://www.cs.waikato.ac.nz/~remco/weka.pdf

OPTIONS

BIFFile -- Set the name of a file in BIF XML format. A Bayes network learned from data can be compared with the Bayes network represented by the BIF file. Statistics calculated are o.a. the number of missing and extra arcs.

debug -- If set to true, classifier may output additional info to the console.

estimator -- Select Estimator algorithm for finding the conditional probability tables of the Bayes Network.

searchAlgorithm -- Select method used for searching network structures.

useADTree -- When ADTree (the data structure for increasing speed on counts, not to be confused with the classifier under the same name) is used learning time goes down typically. However, because ADTrees are memory intensive, memory problems may occur. Switching this option off makes the structure learning algorithms slower, and run with less memory. By default, ADTrees are used.

# weka.classifiers.functions.SMO

SYNOPSIS

Implements John Platt's sequential minimal optimization algorithm for training a **support vector classifier**.

This implementation globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes by default. (In that case the coefficients in the output are based on the normalized data, not the original data --- this is important for interpreting the classifier.)

Multi-class problems are solved using pairwise classification (1-vs-1 and if logistic models are built pairwise coupling according to Hastie and Tibshirani, 1998).

To obtain proper probability estimates, use the option that fits logistic regression models to the outputs of the support vector machine. In the multi-class case the predicted probabilities are coupled using Hastie and Tibshirani's pairwise coupling method.

Note: for improved speed normalization should be turned off when operating on SparseInstances.

For more information on the SMO algorithm, see

J. Platt: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In B. Schoelkopf and C. Burges and A. Smola, editors, Advances in Kernel Methods - Support Vector Learning, 1998.

S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy (2001). Improvements to Platt's SMO Algorithm for SVM Classifier Design. Neural Computation. 13(3):637-649.

Trevor Hastie, Robert Tibshirani: Classification by Pairwise Coupling. In: Advances in Neural Information Processing Systems, 1998.

OPTIONS

buildLogisticModels -- Whether to fit logistic models to the outputs (for proper probability estimates).

c -- The complexity parameter C.

checksTurnedOff -- Turns time-consuming checks off - use with caution.

debug -- If set to true, classifier may output additional info to the console.

epsilon -- The epsilon for round-off error (shouldn't be changed).

filterType -- Determines how/if the data will be transformed.

kernel -- The kernel to use.

numFolds -- The number of folds for cross-validation used to generate training data for logistic models (-1 means use training data).

randomSeed -- Random number seed for the cross-validation.

toleranceParameter -- The tolerance parameter (shouldn't be changed).

# weka.classifiers.lazy.IBk

SYNOPSIS

K-nearest neighbours classifier. Can select appropriate value of K based on cross-validation. Can also do distance weighting.

For more information, see

D. Aha, D. Kibler (1991). Instance-based learning algorithms. Machine Learning. 6:37-66.

OPTIONS

KNN -- The number of neighbours to use.

crossValidate -- Whether hold-one-out cross-validation will be used to select the best k value between 1 and the value specified as the KNN parameter.

debug -- If set to true, classifier may output additional info to the console.

distanceWeighting -- Gets the distance weighting method used.

meanSquared -- Whether the mean squared error is used rather than mean absolute error when doing cross-validation for regression problems.

nearestNeighbourSearchAlgorithm -- The nearest neighbour search algorithm to use (Default: weka.core.neighboursearch.LinearNNSearch).

windowSize -- Gets the maximum number of instances allowed in the training pool. The addition of new instances above this value will result in old instances being removed. A value of 0 signifies no limit to the number of training instances.

* Chapter 4, Data Mining: Practical Machine Learning Tools and Techniques (3nd edition) / Ian Witten, Eibe Frank; Elsevier, 2011.
* Chapter 4, Introduction to Data Mining / Pang-Ning Tan, Michael Steinbach and Vipin Kumar, Pearson Education, Inc, 2006.

|  |  |  |
| --- | --- | --- |
|  | **CfsSubsetEval**  **+**  **LinearForwardSelection** | **InfoGainAttributeEval**  **+**  **Ranker** |
| **None Sampling** | Var126  Var202 | Ranked attributes:  0.3544 49 Var202  0.3469 60 Var217  0.2551 47 Var198  0.2551 63 Var220  0.2551 65 Var222  0.1922 48 Var199  0.1091 59 Var216  0.1048 42 Var192  0.0846 29 Var126  0.0704 46 Var197  0.0488 51 Var204  0.0217 8 Var28  0.0204 58 Var212  0.0198 57 Var211  0.019 43 Var193  0.0119 69 Var228  0.0114 67 Var226 |
| **Under Sampling** | Var28  Var126  Var202  Var211  Var217 | Ranked attributes:  0.94381 49 Var202  0.85052 60 Var217  0.74897 47 Var198  0.74897 63 Var220  0.74897 65 Var222  0.65649 48 Var199  0.52058 42 Var192  0.35058 59 Var216  0.30859 46 Var197  0.26949 51 Var204  0.18802 29 Var126  0.09757 8 Var28  0.08878 57 Var211  0.0713 43 Var193  0.06371 67 Var226  0.06269 69 Var228  0.05185 58 Var212  0.05072 53 Var206  0.02757 62 Var219  0.02548 44 Var195  0.02523 13 Var65  0.01361 56 Var210  0.01178 61 Var218  0.01127 50 Var203  0.01103 68 Var227 |
| **Over Sampling** | Var7  Var24  Var25  Var28  Var57  Var65  Var83  Var85  Var94  Var112  Var123  Var126  Var132  Var144  Var203  Var205  Var211  Var218  Var219  Var226  Var227 | 0.92714 49 Var202  0.88847 60 Var217  0.74634 48 Var199  0.66583 65 Var222  0.66583 63 Var220  0.66583 47 Var198  0.65352 42 Var192  0.57416 21 Var85  0.53693 27 Var123  0.52196 20 Var83  0.50494 51 Var204  0.49507 24 Var112  0.45878 59 Var216  0.45448 13 Var65  0.44677 35 Var144  0.43398 46 Var197  0.41873 7 Var25  0.40027 2 Var7  0.39884 23 Var109  0.36761 6 Var24  0.35942 67 Var226  0.30872 8 Var28  0.20689 4 Var21  0.20053 29 Var126  0.19023 30 Var132  0.17614 61 Var218  0.16074 53 Var206  0.13817 16 Var74  0.13259 52 Var205  0.12654 39 Var163  0.11342 15 Var73  0.10019 58 Var212 |