Contents

ATTRIBUTE EVALUATORS

NAME - weka.attributeSelection.CfsSubsetEval

SYNOPSIS
CfsSubsetEval :

Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them.

Subsets of features that are highly correlated with the class while having low intercorrelation are preferred.

OPTIONS
locallyPredictive -- Identify locally predictive attributes. Iteratively adds attributes with the highest correlation with the class as long as there is not already an attribute in the subset that has a higher correlation with the attribute in question

missingSeperate -- Treat missing as a separate value. Otherwise, counts for missing values are distributed across other values in proportion to their frequency.

NAME - weka.attributeSelection.ChiSquaredAttributeEval

SYNOPSIS
ChiSquaredAttributeEval :

Evaluates the worth of an attribute by computing the value of the chi-squared statistic with respect to the class.

OPTIONS
binarizeNumericAttributes -- Just binarize numeric attributes instead of properly discretizing them.

missingMerge -- Distribute counts for missing values. Counts are distributed across other values in proportion to their frequency. Otherwise, missing is treated as a separate value.

NAME - weka.attributeSelection.ClassifierSubsetEval

SYNOPSIS
Evaluates attribute subsets on training data or a seperate hold out testing set

OPTIONS
classifier -- Classifier to use for estimating the accuracy of subsets

holdOutFile -- File containing hold out/test instances.

useTraining -- Use training data instead of hold out/test instances.

NAME - weka.attributeSelection.ConsistencySubsetEval

SYNOPSIS
ConsistencySubsetEval :

Evaluates the worth of a subset of attributes by the level of consistency in the class values when the training instances are projected onto the subset of attributes.

Consistency of any subset can never be lower than that of the full set of attributes, hence the usual practice is to use this subset evaluator in conjunction with a Random or Exhaustive search which looks for the smallest subset with consistency equal to that of the full set of attributes.

NAME - weka.attributeSelection.GainRatioAttributeEval

SYNOPSIS
GainRatioAttributeEval :

Evaluates the worth of an attribute by measuring the gain ratio with respect to the class.

GainR(Class, Attribute) = (H(Class) - H(Class | Attribute)) / H(Attribute).

OPTIONS

missingMerge -- Distribute counts for missing values. Counts are distributed across other values in proportion to their frequency. Otherwise, missing is treated as a separate value.

NAME - weka.attributeSelection.InfoGainAttributeEval

SYNOPSIS
InfoGainAttributeEval :

Evaluates the worth of an attribute by measuring the information gain with respect to the class.

InfoGain(Class,Attribute) = H(Class) - H(Class | Attribute).

OPTIONS
binarizeNumericAttributes -- Just binarize numeric attributes instead of properly discretizing them.

missingMerge -- Distribute counts for missing values. Counts are distributed across other values in proportion to their frequency. Otherwise, missing is treated as a separate value.

NAME - weka.attributeSelection.OneRAttributeEval

SYNOPSIS
OneRAttributeEval :

Evaluates the worth of an attribute by using the OneR classifier.

OPTIONS
evalUsingTrainingData -- Use the training data to evaluate attributes rather than cross validation.

folds -- Set the number of folds for cross validation.

minimumBucketSize -- The minimum number of objects in a bucket (passed to OneR).

seed -- Set the seed for use in cross validation.

NAME - weka.attributeSelection.PrincipalComponents

SYNOPSIS
Performs a principal components analysis and transformation of the data. Use in conjunction with a Ranker search. Dimensionality reduction is accomplished by choosing enough eigenvectors to account for some percentage of the variance in the original data---default 0.95 (95%). Attribute noise can be filtered by transforming to the PC space, eliminating some of the worst eigenvectors, and then transforming back to the original space.

OPTIONS
maximumAttributeNames -- The maximum number of attributes to include in transformed attribute names.

normalize -- Normalize input data.

transformBackToOriginal -- Transform through the PC space and back to the original space. If only the best n PCs are retained (by setting varianceCovered < 1) then this option will give a dataset in the original space but with less attribute noise.

varianceCovered -- Retain enough PC attributes to account for this proportion of variance.

NAME - weka.attributeSelection.ReliefFAttributeEval

SYNOPSIS
ReliefFAttributeEval :

Evaluates the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class. Can operate on both discrete and continuous class data.

OPTIONS
numNeighbours -- Number of nearest neighbours for attribute estimation.

sampleSize -- Number of instances to sample. Default (-1) indicates that all instances will be used for attribute estimation.

seed -- Random seed for sampling instances.

sigma -- Set influence of nearest neighbours. Used in an exp function to control how quickly weights decrease for more distant instances. Use in conjunction with weightByDistance. Sensible values = 1/5 to 1/10 the number of nearest neighbours.

weightByDistance -- Weight nearest neighbours by their distance.

NAME - weka.attributeSelection.SVMAttributeEval

SYNOPSIS
SVMAttributeEval :

Evaluates the worth of an attribute by using an SVM classifier.

For more information see:
Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. Machine Learning, 46, 389-422

OPTIONS
attsToEliminatePerIteration -- Constant rate of attribute elimination.

complexityParameter -- C complexity parameter to pass to the SVM

epsilonParameter -- P epsilon parameter to pass to the SVM

filterType -- filtering used by the SVM

percentThreshold -- Threshold below which percent elimination reverts to constant elimination.

percentToEliminatePerIteration -- Percent rate of attribute elimination.

toleranceParameter -- T tolerance parameter to pass to the SVM

NAME - weka.attributeSelection.SymmetricalUncertAttributeEval

SYNOPSIS
SymmetricalUncertAttributeEval :

Evaluates the worth of an attribute by measuring the symmetrical uncertainty with respect to the class.

SymmU(Class, Attribute) = 2 * (H(Class) - H(Class | Attribute)) / H(Class) + H(Attribute).

OPTIONS
missingMerge -- Distribute counts for missing values. Counts are distributed across other values in proportion to their frequency. Otherwise, missing is treated as a separate value.

NAME - weka.attributeSelection.WrapperSubsetEval

SYNOPSIS
WrapperSubsetEval:

Evaluates attribute sets by using a learning scheme. Cross validation is used to estimate the accuracy of the learning scheme for a set of attributes.

OPTIONS
classifier -- Classifier to use for estimating the accuracy of subsets

folds -- Number of xval folds to use when estimating subset accuracy.

seed -- Seed to use for randomly generating xval splits.

threshold -- Repeat xval if stdev of mean exceeds this value.

SEARCH METHODS

NAME - weka.attributeSelection.BestFirst

SYNOPSIS
BestFirst:

Searches the space of attribute subsets by greedy hillclimbing augmented with a backtracking facility. Setting the number of consecutive non-improving nodes allowed controls the level of backtracking done. Best first may start with the empty set of attributes and search forward, or start with the full set of attributes and search backward, or start at any point and search in both directions (by considering all possible single attribute additions and deletions at a given point).

OPTIONS
direction -- Set the direction of the search.

lookupCacheSize -- Set the maximum size of the lookup cache of evaluated subsets. This is expressed as a multiplier of the number of attributes in the data set. (default = 1).

searchTermination -- Set the amount of backtracking. Specify the number of

startSet -- Set the start point for the search. This is specified as a comma seperated list off attribute indexes starting at 1. It can include ranges. Eg. 1,2,5-9,17.

NAME - weka.attributeSelection.ExhaustiveSearch

SYNOPSIS
ExhaustiveSearch :
Performs an exhaustive search through the space of attribute subsets starting from the empty set of attrubutes. Reports the best subset found.

OPTIONS
verbose -- Print progress information. Sends progress info to the terminal as the search progresses.

NAME - weka.attributeSelection.GeneticSearch

SYNOPSIS
GeneticSearch :

Performs a search using the simple genetic algorithm described in Goldberg (1989).

OPTIONS
crossoverProb -- Set the probability of crossover. This is the probability that two population members will exchange genetic material.

maxGenerations -- Set the number of generations to evaluate.

mutationProb -- Set the probability of mutation occuring.

populationSize -- Set the population size. This is the number of individuals (attribute sets) in the population.

reportFrequency -- Set how frequently reports are generated. Default is equal to the number of generations meaning that a report will be printed for initial and final generations. Setting the value to 5 will result in a report being printed every 5 generations.

seed -- Set the random seed.

startSet -- Set a start point for the search. This is specified as a comma seperated list off attribute indexes starting at 1. It can include ranges. Eg. 1,2,5-9,17. The start set becomes one of the population members of the initial population.

NAME - weka.attributeSelection.GreedyStepwise

SYNOPSIS
GreedyStepwise :

Performs a greedy forward or backward search through the space of attribute subsets. May start with no/all attributes or from an arbitrary point in the space. Stops when the addition/deletion of any remaining attributes results in a decrease in evaluation. Can also produce a ranked list of attributes by traversing the space from one side to the other and recording the order that attributes are selected.

OPTIONS
generateRanking -- Set to true if a ranked list is required.

numToSelect -- Specify the number of attributes to retain. The default value (-1) indicates that all attributes are to be retained. Use either this option or a threshold to reduce the attribute set.

searchBackwards -- Search backwards rather than forwards.

startSet -- Set the start point for the search. This is specified as a comma seperated list off attribute indexes starting at 1. It can include ranges. Eg. 1,2,5-9,17.

threshold -- Set threshold by which attributes can be discarded. Default value results in no attributes being discarded. Use in conjunction with generateRanking

NAME - weka.attributeSelection.RaceSearch

SYNOPSIS
Races the cross validation error of competing attribute subsets. Use in conjuction with a ClassifierSubsetEval. RaceSearch has four modes:

forward selection races all single attribute additions to a base set (initially no attributes), selects the winner to become the new base set and then iterates until there is no improvement over the base set.

Backward elimination is similar but the initial base set has all attributes included and races all single attribute deletions.

Schemata search is a bit different. Each iteration a series of races are run in parallel. Each race in a set determines whether a particular attribute should be included or not---ie the race is between the attribute being "in" or "out". The other attributes for this race are included or excluded randomly at each point in the evaluation. As soon as one race has a clear winner (ie it has been decided whether a particular attribute should be inor not) then the next set of races begins, using the result of the winning race from the previous iteration as new base set.

Rank race first ranks the attributes using an attribute evaluator and then races the ranking. The race includes no attributes, the top ranked attribute, the top two attributes, the top three attributes, etc.

It is also possible to generate a raked list of attributes through the forward racing process. If generateRanking is set to true then a complete forward race will be run---that is, racing continues until all attributes have been selected. The order that they are added in determines a complete ranking of all the attributes.

Racing uses paired and unpaired t-tests on cross-validation errors of competing subsets. When there is a significant difference between the means of the errors of two competing subsets then the poorer of the two can be eliminated from the race. Similarly, if there is no significant difference between the mean errors of two competing subsets and they are within some threshold of each other, then one can be eliminated from the race.

OPTIONS
attributeEvaluator -- Attribute evaluator to use for generating an initial ranking. Use in conjunction with a rank race

debug -- Turn on verbose output for monitoring the search's progress.

generateRanking -- Use the racing process to generate a ranked list of attributes. Using this mode forces the race to be a forward type and then races until all attributes have been added, thus giving a ranked list

numToSelect -- Specify the number of attributes to retain. Use in conjunction with generateRanking. The default value (-1) indicates that all attributes are to be retained. Use either this option or a threshold to reduce the attribute set.

raceType -- Set the type of search.

selectionThreshold -- Set threshold by which attributes can be discarded. Default value results in no attributes being discarded. Use in conjunction with generateRanking

significanceLevel -- Set the significance level to use for t-test comparisons.

threshold -- Set the error threshold by which to consider two subsets equivalent.

NAME - weka.attributeSelection.RandomSearch

SYNOPSIS
RandomSearch :
Performs a Random search in the space of attribute subsets. If no start set is supplied, Random search starts from a random point and reports the best subset found. If a start set is supplied, Random searches randomly for subsets that are as good or better than the start point with the same or or fewer attributes. Using RandomSearch in conjunction with a start set containing all attributes equates to the LVF algorithm of Liu and Setiono (ICML-96).

OPTIONS
searchPercent -- Percentage of the search space to explore.

startSet -- Set the start point for the search. This is specified as a comma seperated list off attribute indexes starting at 1. It can include ranges. Eg. 1,2,5-9,17. If specified, Random searches for subsets of attributes that are as good as or better than the start set with the same or lower cardinality.

verbose -- Print progress information. Sends progress info to the terminal as the search progresses.

NAME - weka.attributeSelection.Ranker

SYNOPSIS
Ranker :
Ranks attributes by their individual evaluations. Use in conjunction with attribute evaluators (ReliefF, GainRatio, Entropy etc).

OPTIONS
generateRanking -- A constant option. Ranker is only capable of generating attribute rankings.

numToSelect -- Specify the number of attributes to retain. The default value (-1) indicates that all attributes are to be retained. Use either this option or a threshold to reduce the attribute set.

startSet -- Specify a set of attributes to ignore. When generating the ranking, Ranker will not evaluate the attributes in this list. This is specified as a comma seperated list off attribute indexes starting at 1. It can include ranges. Eg. 1,2,5-9,17.

threshold -- Set threshold by which attributes can be discarded. Default value results in no attributes being discarded. Use either this option or numToSelect to reduce the attribute set.

NAME - weka.attributeSelection.RankSearch

SYNOPSIS
RankSearch :

Uses an attribute/subset evaluator to rank all attributes. If a subset evaluator is specified, then a forward selection search is used to generate a ranked list. From the ranked list of attributes, subsets of increasing size are evaluated, ie. The best attribute, the best attribute plus the next best attribute, etc.... The best attribute set is reported. RankSearch is linear in the number of attributes if a simple attribute evaluator is used such as GainRatioAttributeEval.

OPTIONS
attributeEvaluator -- Attribute evaluator to use for generating a ranking.