NAME

pboost − PrefixSpan Boosting for sequence data.

SYNOPSIS

pboost *[options] input.txt*

DESCRIPTION

Train a classifier for sequences, given training data. The classifier consists of a linear combination of simple decision stumps, each checking for the existence of a particular subsequence within the sample. The program **ptest** can be used to apply the trained classifier to new sequences.

The classifier can handle 2-class problems as well as multiclass classification problems using a decision dag (DDAG) decomposition. Additionally, a "1.5-class" variant is implemented, which models a single positive class given knowledge of negative training examples. The difference is here, that only those stumps sequences are selected which appear in the positive class.

GENERIC OPTIONS

--help Print a short usage summary.

PARAMETERS

--output <file.txt>

Set the output file name the trained classifier will be written to. By default this is set to "output.txt".

--classifier-type <value>

Select the LPBoost classifier type. Possible values are 1 (1.5-class LPBoost classifier) and 2 (2-class standard LPBoost classifier). For most problems, the default value of 2 is recommended.

--nu <value>

Set the LPBoost nu parameter value. The value must satisfy: 0.0 < value < 1.0. A large value will cause a stronger regularization and possibly lower performance on the training set, whereas a small value enforces correct classification of the training set, with less regularization. Common values are 0.01, 0.05, 0.1, 0.25. By default this is set to 0.1.

--conv-eps <value>

Set the convergence tolerance of each LPBoost training run. If this is near-zero, the training is exact but might take many iterations to complete. A higher value allows faster training times but possible reduces the accuracy. Values such as 0.01 are recommended. By default, a conservative value of 0.001 is used.

--pricing-count <value>

Number of classifiers to add in each mining iteration. A larger value can increase the convergence speed and corresponds to "multiple pricing column generation" in the linear programming literature. The default value is 1, but this might be the easiest parameter to change to reduce the time of a training run.

--search-strategy <value>

Select the search strategy in the subsequence tree. Possible values are 1 (A-Star optimal search frontier), 2 (Depth-First-Search) and 3 (iterative-deepening A-Star). The default and recommended setting is 3, to use the iterative deepening A-Star variant.

--exploration-max <value>

The maximum search depth per subsequence finding iteration. If set to zero (0), no maximum search depth is enforced and the search is exact. The default value is 10e6.

REGULARIZATION OPTIONS

--support-min <value>

The number of times a subsequence needs to appear in the training set such that it is considered as stump sequence. A value of at least two is recommended but a higher value can be used to regularize the classifier to sequences that are likely to appear in testing data. By default, at least two

training samples need to contain the sequence.

--length-min <value>
The minimum required sequence length (total number of items) for stump sequences. By default, no minimum length is enforced.

--length-max <value>
The maximum allowed sequence length (total number of items) for stump sequences. By default, no maximum length is enforced.

--maxgap <value>
The maximum allowed gap when matching a subsequence to a larger sequence. This can influence the training time, where the most efficient training is achieved by arbitrary long gaps (value -1) and longer allowed gaps reduce efficiency. By default arbitrary long gaps are allowed (-1).

2-CLASS LPBOOST OPTIONS

--directional <0|1>
Experimental: use directional decision stumps. This makes sense only for the 2-class LPBoost classifier and should only be used with knowledge of the implementation details.

1.5-CLASS LPBOOST OPTIONS

--alpha-bound <value>
This enforces an upper bound on each classifiers alpha coefficient. This is experimental and only implemented for the 1.5-class LPBoost classifier. This should not be used, except you know the implementation details.

--pos-support-min <value>
The same as the previous option, where the minimum-support condition is enforced only in the positive class. This option is only implemented for the 1.5-class LPBoost classifier.

SEQUENCE DATA

A sequence is an ordered list of one or more elements. An element is an set of zero or more items. An item is a positive integer number.

The subsequence relationship is defined between two sequences s1 and s2 as: s1 is a subsequence of s2, if and only if a monotonically-increasing index mapping I for each element in s1 can be established, such that each element s1(k) is a subset of s2(I(k)). That is, a sequence is a subsequence of another, if it can be matched with arbitrary long gaps but preserving the order, such that the matched elements satisfy a subset relationship.

If a maximum gap constraint is used, the above holds but additionally the difference of each pair of adjacent indices in the mapping I must be no greater then maxgap+1.

INPUT FILE FORMAT

The input file consists of n lines, each containing a filename and a class label, separated by any whitespace character. The filename corresponds to a sequence file. If only two classes are considered, the class labels are "1" and "-1", for the positive and negative class, respectively. If more classes are considered, the first class is labeled "0", and the following classes "1", "2", etc.

Each sequence file consists of p lines, each line representing one sequence element. Each element contains a number of items (positive integer numbers), separated by whitespace characters. The items within each element must be ordered and contain no duplicate item.

BUGS

No bugs known, if you find any, please send a bug report to me. I will try to fix it.

AUTHOR

Sebastian Nowozin <sebastian.nowozin@tuebingen.mpg.de>

SEE ALSO

pspan(1), **ptest**(1)