NAME

pspan − PrefixSpan frequent subsequence mining.

SYNOPSIS

pspan *[options] input.txt*

DESCRIPTION

Mine a set of sequences for frequently appearing subsequences. Each returned sequence appears as subsequence in the training set "frequently", its' support is above a given threshold.

There are two modes of operation: either the minimum required support is given as absolute value or fraction of the number of training samples (--minsup and --minsuppct options, respectively), or a constant K is provided and the top K frequent subsequences are returned (--topk option).

Additionally, a maximum gap constraint can be specified using the --maxgap option (see the documentation of the **pboost** program for the meaning of the maximum gap constraint).

The resulting set of sequences can be written to an output file and projection of the training set onto these sequences can be carried out (--project option).

GENERIC OPTIONS

--help Print a short usage summary.

PARAMETERS

--output <file.txt>
: Set the output file name where the found subsequences will be written to. By default this is set to "output.txt".

--project <file.txt>
: After mining the frequent subsequences, project the training data onto the found sequences, producing for n training sequences and k frequent subsequences an n-times-k binary matrix encoding which subsequence appears in which training sequence.

--verbose
: Enable verbose mode.

--verify Debug option: explicit subsequence occurence verification. Because PrefixSpan implicitly enumerates the sequences, the meta-information about each found sequence is the result of a long chain of computations. This option enables an explicit verification function that verifies the meta-information for each found sequence and abort the program if an inconsitent state is found. This should be used only for debugging purposes as it slows down the program considerably.

--topk <value>
: Mine the top-K frequent subsequences. The mining is carried out using the A-Star algorithm and is very efficient.

  At least K subsequences are returned, usually more. The reason is that given a certain threshold K there might be more sequences equally frequent as the K'th found sequence. Such sequences are equally frequent and hence equally important.

  This option is the default and set to 20.

--minsup <value>
: Explicitly give the minimum required support as absolute sequence count.

--minsuppct <value>
: Give the minimum required support as fraction of the training sample count. Thus, 0.0 < value <= 1.0 must hold.

--length-min <value>
Minimum required subsequence length, measured in total item count. The default is 1.

--length-max <value>
Maximum allowed subsequence length, measured in total item count. If zero (0) is given, no maximum length is enforced. The default is zero.

--maxgap <value>
Maximum allowed element gap when matching, or -1 for infinite gaps. The default is -1.

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.

PERFORMANCE

On a medium sized test problem (383 sequences, with 15 elements and an average item count of circa 400) the following timings have been obtained, extracting the top-500 frequent subsequences: 1m10s for infinite gap length (-G -1), for maximum gap constrained problems: 1m54s (-G 0), 3m00s (-G 1), 4m20s (-G 2), 5m55s (-G 3), 7m35s (-G 4) and 9m13s (-G 5).

This highlights the relative speeds of mining frequent subsequences with arbitrary long gaps (fastest) compared to maximum-gap constrained subsequences with high gap lengths (slowest). The performance is empirically linear in the maximum allowed gap length.

When using the iterative deepening A-Star search strategy, memory usage should never be a problem, even for a larger database of sequences.

INPUT FILE FORMAT

The input file consists of n lines, each containing a filename. The filename corresponds to a sequence file.

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

pboost(1), **ptest**(1)