



Evaluation file format

We use the simple text file format specified below to describe problem instances. The format is a generalization of the Ergo file format initially developed by  Noetic Systems Inc. for their  Ergo software.

Evaluation benchmark files typically have the ending *.uai*.

Structure

A file in the UAI format consists of the following two parts, in that order:

```
<Preamble>
<Function tables>
```

The contents of each section (denoted `< . . . >` above) are described in the following:

Preamble

The preamble starts with one line denoting the type of network. This will be either **BAYES** (if the network is a Bayesian network) or **MARKOV** (in case of a Markov network). This is followed by a line containing the number of variables. The next line specifies each variable's domain size, one at a time, separated by whitespace (note that this implies an order on the variables which will be used throughout the file).

The fourth line contains only one integer, denoting the number of functions in the problem (conditional probability tables for Bayesian networks, general factors for Markov networks). Then, one function per line, the scope of each function is given as follows: The first integer in each line specifies the size of the function's scope, followed by the actual indexes of the variables in the scope. The order of this list is not restricted, except when specifying a conditional probability table (CPT) in a Bayesian network, where the child variable has to come last. Also note that variables are indexed starting with 0.

For instance, a general function over variables 0, 5 and 11 would have this entry:

```
3 0 5 11
```

A simple Markov network preamble with three variables and two functions might for instance look like this:

```
MARKOV
3
2 2 3
2
2 0 1
3 0 1 2
```

The first line denotes the Markov network, the second line tells us the problem consists of three variables, let's refer to them as X , Y , and Z . Their domain size is 2, 2, and 3 respectively (from the third line). Line four specifies that there are 2 functions. The scope of the first function is X,Y , while the second function is defined over X,Y,Z .

An example preamble for a Belief network over three variables (and therefore with three functions) might be:

```
BAYES
3
2 2 3
3
1 0
2 0 1
2 1 2
```

The first line signals a Bayesian network. This example has three variables, let's call them X , Y , and Z , with domain size 2, 2, and 3, respectively (from lines two and three). Line four says that there are 3 functions (CPTs in this case). The scope of the first function is given in line five as just X (the probability $P(X)$), the second one is defined over X and Y (this is $P(Y|X)$). The third function, from line seven, is the CPT $P(Z|Y)$. We can therefore deduce that the joint probability for this problem factors as $P(X,Y,Z) = P(X).P(Y|X).P(Z|Y)$.

Function tables

In this section each function is specified by giving its full table (i.e, specifying the function value for each tuple). The order of the functions is identical to the one in which they were introduced in the preamble.

For each function table, first the number of entries is given (this should be equal to the product of the domain sizes of the variables in the scope). Then, one by one, separated by whitespace, the values for each assignment to the variables in the function's scope are enumerated. Tuples are implicitly assumed in ascending order, with the last variable in the scope as the 'least significant'.

To illustrate, we continue with our Bayesian network example from above, let's assume the following conditional probability tables:

X	$P(X)$
0	0.436
1	0.564

X	Y	$P(Y X)$
0	0	0.128
0	1	0.872
1	0	0.920
1	1	0.080

Y	Z	$P(Z Y)$
0	0	0.210
0	1	0.333

0	2	0.457
1	0	0.811
1	1	0.000
1	2	0.189

The corresponding function tables in the file would then look like this:

```

2
0.436 0.564

4
0.128 0.872
0.920 0.080

6
0.210 0.333 0.457
0.811 0.000 0.189

```

(Note that line breaks and empty lines are effectively just whitespace, exactly like plain spaces " ". They are used here to improve readability.)

Summary

To sum up, a problem file consists of 2 sections: the preamble and the full the function tables, the names and the labels

For our Markov network example above, the full file could be:

```

MARKOV
3
2 2 3
2
2 0 1
3 0 1 2

4
4.000 2.400
1.000 0.000

12
2.2500 3.2500 3.7500
0.0000 0.0000 10.0000
1.8750 4.0000 3.3330
2.0000 2.0000 3.4000

```

Here is the full Bayesian network example from above:

```

BAYES
3
2 2 3
3
1 0
2 0 1
2 1 2

2

```

```
0.436 0.564
4
0.128 0.872
0.920 0.080
6
0.210 0.333 0.457
0.811 0.000 0.189
```

Expressing evidence

Evidence is specified in a separate file. This file has the same name as the original problems file but an added *.evid* extension at the end. For instance, `problem.uai` will have evidence in `problem.uai.evid`.

The file simply starts with a line specifying the number of evidence variables. This is followed by the pairs of variable and value indexes for each observed variable, one pair per line. The indexes correspond to the ones implied by the original problem file.

If, for our above example, we want to specify that variable *Y* has been observed as having its first value and *Z* with its second value, the file `example.uai.evid` would contain the following:

```
2
1 0
2 1
```

Examples

Some example problem instances are available [here](#).