

OCCAM3

A Reconstructability Analysis Program (Organizational Complexity Computation and Modeling)

Kenneth Willett and Martin Zwick
Systems Science Ph.D. Program, Portland State University
Portland OR 97207
last modified: Feb. 28, 2002

OCCAM INPUT

ACTION

When one brings Occam up, one first must choose between two Occam “actions”: “Do Fit” and “Do Search”. Occam then returns a window specific to the choice made. Search assesses *many models* either from the full set of all possible models or from various partial subsets of models. Fit examines *one model (or a few models)* in greater detail. In an exploratory mode, one would do Search first, and then Fit, but in a confirmatory mode, one would simply do Fit.

DATA FILE

The user must specify a data file on the user’s computer, which is then uploaded to the Occam server. This is actually all that is needed to submit an Occam job, if the user is satisfied with the default setting of all the parameters.

Data files are ascii files, generated by Notepad, or Word (where file is saved in .txt format), or Excel (where file is saved in .prn format, i.e., with space-separated columns). A minimal data file looks like this (this is the data from the Wholes & Parts paper):

```
:action
search

:nominal
alpha, 2,1,a
beta, 2,1,b
gamma, 2,2,c

:data
0 0 0 143
0 0 1 253
0 1 0 77
0 1 1 182
1 0 0 227
1 0 1 411
1 1 0 46
1 1 1 139
```

This simple file has 3 parts: (1) the action specification, (2) specification of the variables, (3) the data to be analyzed. Each part in this example begins with a line of the form “:parameter”, where “parameter” is “action”, “nominal”, or “data”.

Variable specification

The action specification is straightforward. Variable specification begins with “:nominal” which reminds the user that nominal (categorical, qualitative) variables must be used. In the future, Occam will be able to bin (discretize) values of quantitative variables and thus convert the variables to nominal, but at present, the user must do this conversion. (Unless there are compelling reasons to choose a different number of bins, 3 bins is a good default; this allows detection of nonlinearities.) After “:nominal”, the variables are specified, one per line. In the above example, the first line is (spaces in these lines are ignored):

```
alpha, 2,1,a
```

“alpha” is the name of the first variable. The second field indicates that it has 2 possible states (a “cardinality” of 2). The third field (shown above as 1) is either 1 or 2. 1 means the variable is an “independent variable” (IV) or input. 2 means it is a “dependent variable” (DV) or output.

In the future but *not yet*, a value of 0 will also be allowed in this third field. 0 will mean that the variable (and the corresponding column in the data) should be ignored. This will allow the user to have data for more variables than will be analysed at any one time; the user could then easily alter which variables are to be included in the analysis and which are to be omitted.

The fourth field is a variable abbreviation, usually one letter, specified in lower case, but used in Occam output in upper case. In the above example, alpha will be referred to in Occam output as A. If there are more than 26 variables, one can use double (or triple, etc.) letters as abbreviations, for example “aa” or “ab”. Such a variable will appear in a model name with only its first letter capitalized, e.g., AaB:AbC:.... Numbers may not be used to identify variables.

If all variables are designated as IVs (1) or as DVs (2), the system is “neutral.” If some variables are IVs, and at least one is a DV, the system is “directed.” The above data file is for a directed system.

Data specification

The third part of this file is the data, which follows the “:data” line. In the data, variables are columns, separated by one or more spaces. The columns from left to right correspond to the sequence of variables specified above, i.e., the first column is alpha, the second beta, and the third gamma. Following the variable columns is an additional column which gives the frequency of occurrence of the particular state specified by the variable values.

Since variables are nominal, their values (states) are *names*. Normally, these will be 0,1,2,... or 1,2,3,... To avoid possible confusion, it is best to start the labelling of all variables either with 0 or with 1, i.e., avoid starting one variable with 0 and another with 1 (though Occam can handle such inconsistencies of convention). The user should know the number of different states which occur for each variable and indicate the cardinality of the variable correctly in the variable specification.

Data can alternatively be provided to Occam without frequencies, where each line (row) represents a single *case*. The rows do not have to be ordered in any particular way. Occam will generate the frequencies itself, but it needs to be told that the data do not include frequencies, as follows:

```
:no-frequency
```

```
:data
```

```
0 0 0
1 1 1
0 1 1
1 0 1
0 0 1
0 1 0
0 1 1
0 0 1
1 0 0
1 0 1
1 1 0
1 1 1
```

Uploading data will be faster if the data provides frequencies, so if response time is a problem, the user might consider doing this operation before calling Occam.

Missing values

In the data, a row (case) and column (variable) cannot have a missing value. In preparing data for Occam, a missing value can be handled in one of three ways: (a) the row can be deleted from the data, (b) an additional value for the variable can be defined, which will mean “missing” (for example, if the variable is binary with states 0 and 1, a missing value could be assigned a new value of 2 and the cardinality of the variable would now be 3), or (c) the value can be assigned randomly according to the observed probabilities of the different values in the rest of the data. If only a few lines have missing values, (a) is the best choice.

Additional parameter specifications

In addition to action, variables, and data, the data file may include additional parameter specifications. A parameter specification is either just a single line when the parameter is a “switch,” such as the “no-frequency” parameter shown above, or it involves two lines, the first giving the parameter name and the second its value. For example, to specify the number of levels to be searched to be 10, one would add, above the data portion the following two lines:

```
:search-levels
10
```

Some parameters, such as “search-levels”, cannot presently be set on the browser input page; if their default values are not satisfactory, the desired values *must* be specified in the data file. All Occam parameters relevant to Search, and their default values, are listed in Table 1.

Table 1. Data file parameters and their defaults

parameter	default
search-levels	1000000
optimize-search-width	10
ipf-maxit	266
ipf-maxdev	.25

“search-levels” controls the depth/height of the search; “optimize-search-width” controls the width of the search, i.e., the number of models retained at every level. “ipf-maxit” and “ipf-maxdev” pertain to the Iterative Proportional Algorithm which generates the calculated probabilities (q’s) for a model. “ipf-maxit” is the maximum number of iterations allowed for IPF; “ipf-maxdev” is the maximum difference of frequencies (not probabilities) allowed between a state in the distribution for a calculated projection included in the model and the corresponding state in the observed projection. If Chi-square errors are reported in a run, consider increasing “ipf-maxit” and decreasing “ipf-maxdev.”

For parameters specified on the browser input page, this browser specification overrides any contrary specification that might be found in the data file. Parameter specifications can be echoed in Occam’s output by checking the “Print Options Settings” (see below) so that one can have a record of them.

STARTING MODEL

Occam searches from a starting model. This can be specified on the browser page as “top” (the data or “saturated model”), “bottom” (the independence model), or some structure other than the top or bottom, e.g., “AB:BC”. This field can also be omitted, in which case Occam uses the starting model specified in the data file (after the variable specification and before the data), as follows,

```
:short-model
AB:BC
```

(“Short” refers to the variable abbreviations.) If the data file also does not specify a starting model, Occam uses the default starting model, which for neutral systems is “top”, and for directed systems is “bottom”.

REFERENCE MODEL

Assessing the quality of a model involves comparing it to a reference model, usually either “top” or “bottom”. If the reference model specified in the browser page is left as default, it will be “top” for neutral systems and “bottom” for directed systems (like the convention for the starting model). If the reference model is “top,” one is asking if it is reasonable to represent the data by a simpler model. If the reference model is “bottom,” one is asking whether the data justifies a model more complex than the independence model.

The reference model can be the starting model. When the starting model is neither the top or the bottom, this can be used to determine whether “incremental” changes from the starting model are acceptable, as opposed to whether “cumulative” changes from the top or bottom are acceptable. The starting model may be a good model obtained in a prior search, and one may now be investigating whether it can be improved upon. At present, if the reference model is chosen to be the starting model, the starting model must be entered explicitly on the browser input page; Occam will not pick it up from the data file.

MODELS TO CONSIDER

Occam offers a choice between (a) all, (b) loopless, (c) disjoint, and (d) chain models.

All models

“All” means there are no restrictions on the type of model to be considered. One controls the extent of this search with parameters “search-levels” and “optimize-search-width”, both of which must be set in the data file. Their current default values are 1000000 and 20, respectively. A width of 20 will generate the full lattice (114 models) for a 4-variable neutral system. Occam generates all “parents” of a model if search direction is “up” or all “children” if search direction is “down”. It then retains the best “optimize-search-width” number of models, where best is determined by the parameter “During Search, Sort By”, which is either Information or Alpha. (At the starting level, there is only one model, but subsequently there will always be “optimize-search-width” models.)

Loopless models

Loopless models are a subset of the full lattice of structures. For example, AB:BC is loopless, but AB:BC:AC has a loop, and would not be included in a loopless search. Doing a loopless search will be faster than an “all” search for two reasons: (1) the iterative procedure (Iterative Proportional Fitting, or IPF) used to generate model probabilities converges in a single cycle for loopless models, but requires several and possibly many cycles for models with loops, and (2) the lattice of loopless models is smaller than the full lattice.

An important use of a loopless search is for variable screening in directed systems. In a directed system, all models have one component which includes all the IVs, and all other components include at least one DV. Call a component that includes a DV a “predicting component”; these are shown in bold in this paragraph and the next. A *single-predicting-*

component (SPC) model, e.g., AB:AC, will never have a loop, but *multiple-predicting-component* (MPC) models, e.g., AB:AC:BC, will always have loops. So a loopless search looks only at SPC models. This is valuable for screening IVs, i.e., for eliminating IVs which don't impact the DV(s) very much. Suppose one had 100 IVs and 1 DV, and wanted to find out which of the 100 IVs has predictive value for the DV. A loopless search will provide this information.

For a loopless search, "search-levels" determines how many IVs will be in the SPC, and "optimize-search-width" determines whether all such models are considered. To illustrate: suppose, one has four IVs, A,B,C,D, and one DV, Z, and one starts the search at the bottom. If "optimize-search-width" is 2 and "search-levels" is 3, then at the first search level Occam generates all parents of ABCD:Z, i.e., all one-IV SPC models: ABCD:AZ, ABCD:BZ, ABCD:CZ, ABCD:DZ. On the basis of the Sort parameter specified in the browser input page, Occam then picks the best 2 of these, say ABCD:BZ and ABCD:DZ. Then, at the second search level, all parents of these 2 models are considered. These will include predicting components of ABZ, CBZ, DBZ, and ADZ, BDZ, CDZ. The best 2 of these 5 models will be retained. Say these are ABCD:ABZ and ABCD:BDZ. Occam then examines, at the third search level, all parents of these models, and again keeps the best 2.

If one wants to do an *exhaustive* search of *all* SPC models with a certain number of IVs in the predicting component, one needs to set the width parameter high enough. For problems with many variables, if the number of IV predictors one wants to consider is high, this may be impractical. A *heuristic* selection of good SPC models may then have to be done, using reasonable values of "optimize-search-width" and "search-levels".

Disjoint models

"Disjoint" means non-overlapping, that is, any two components of a model do not overlap in their variables. For neutral systems, the idea of a disjoint model is straightforward. A disjoint model search would reveal what are the best "cuts" of a system into non-overlapping subsystems, e.g., for a 4-variable system, AB:CD or AC:B:D. Such a search could also be used as a rough search, after which one might do a downwards search relaxing the constraint of disjointness.

For directed systems, the notion of a disjoint model is not as straightforward. Only the independence model and the saturated model are disjoint in a strict sense. For example, for three variables where A and B are IVs and C is the DV, since every directed system model must have an AB component, only AB:C and ABC are disjoint. What one is really interested in here is the disjointness of the *predicting components*, and more specifically, the disjointness of the *IVs in the predicting components*. A disjoint model, for a directed system, will thus be defined to mean that there is no overlap in the IVs of any two predicting components. That is, the influence of subsets of the IVs on the DV are separable, and have no interaction effects. For example, directed system ABC:AZ:BX is disjoint, but directed system ABC:ABZ:BCZ is not. Note that if ABC:AZ:BX were a neutral system, it would be considered *not* disjoint.

In summary, for neutral systems, disjoint models partition all the variables into non-overlapping subsets. For directed systems (with one DV), disjoint models partition all the IVs which affect the DV into non-overlapping subsets.

Chain models

AB:BC:CD:DE illustrates the idea of a chain model. All components have two variables, and every component, except for the ends, overlaps the component to the left with one variable and the component to the right with the other. Chain models searches are not searches in the sense of starting with a model and going either up or down the lattice. Occam simply generates and evaluates all chain models. Chain are currently being used for studies on the use of RA to prestructure genetic algorithm genomes. One could compare all possible lineal causal chains, of the form $A \rightarrow B \rightarrow C \rightarrow D$, by using the chain model option.

SEARCH DIRECTION

The default direction is up for directed systems and down for neutral systems, but for some purposes one might wish to do a downwards search for a directed system or an upwards search for a neutral system. The Search Direction should not be confused with the Reference Model. Model assessments depend on the Reference Model but not on the Search Direction.

DURING SEARCH, SORT BY

The browser page offers a choice of sorting by Information, or Alpha. The best “optimize-search-width” models at every level which are retained for going to the next level are determined by this sort criterion.

Information is constraint captured in a model, normalized to a range of 0 to 1. It is linear with uncertainty (Shannon entropy), likelihood-ratio Chi-square, and %-reduction of uncertainty (for directed systems with one DV), so sorting on information simultaneously sorts on these parameters.

Alpha is obtained from Chi-square tables using the likelihood-ratio Chi-square and dDF (delta-degrees of freedom) as inputs. It is the probability of a Type I error, namely the probability of being in error if one rejects the null hypothesis that a model is really the same as the reference model. Note that if the reference model is “bottom”, a model is good, in the sense of being statistically different from the independence model, if Alpha is *low*, so the “standard” cut-off of 0.05 could be used. If, the reference model is “top”, a model is good, in the sense of being statistically the same as the data, if Alpha is *high*, so the standard 0.05 makes no sense. However, we don’t want Alpha to be too high, or the model will be too complex. In one log-linear book, an Alpha of .1 to .35 is recommended, but the choice of Alpha really depends on the user’s purposes.

WHEN SEARCHING, PREFER

At every level Occam chooses the best “width” out of a set of candidate models by using the sorting criterion (Information or Alpha). When this criterion is Information, one

obviously prefers Larger Values, but when the sort criterion is Alpha, one might prefer *either* “Larger Values” (if the reference model is the top and one cares a great deal about fidelity to the data) or “Smaller Values” (if the reference model is the bottom and one cares a great deal about the statistical justifiability of complex models).

IN REPORT, SORT BY

Output can be sorted by (a) Information, (b) Alpha, (c) dDF, and (d) levels. (NB: the measure used to sort the Occam output report need not be the same as the measure used in sorting done in the search process.) dDF is the change of degrees of freedom relative to the reference model. Sort by levels allows the user to have output which truly follows the order of the Lattice of Structures; this is not actually accomplished by sorting on dDF, because different variable cardinalities can result in a model at a lower level still having a higher DF than a model at a higher level.

IN REPORT, SORT

Occam output can be sorted in either (a) descending or (b) ascending order of the magnitudes of the sorting measure. For example, if the report is sorted on Information in a descending order, then the most complex, high information, models will appear in the output at the top of the page.

RETURN DATA IN SPREADSHEET FORMAT

If this is selected, Occam returns its output as a .csv (comma separated columns) file, where the first name of the file is the first name of the input file. The .csv format is one of the standard input formats for Excel, so if one clicks on the .csv file, one will go directly into Excel, and see the Occam output in an Excel spreadsheet for further processing. While Occam seems to allow the user to either open or save the .csv file, in actual fact, the user must save the file and open it later.

PRINT OPTION SETTINGS

When selected, Occam echoes the parameter setting which have been specified in both the browser input page and the data file before it displays the actual output of the Occam run. This allows the user to document what data file and parameter settings produced the Occam output.

SEND

This sends the browser page to the Occam server. Occam will return its output in a new window. This makes it easy for the user to change parameter settings on the browser input page, and resubmit.

OCCAM OUTPUT

If Print Options Settings has been selected, the Occam output will begin by echoing the parameter settings from the web input page and from the data file. At present Occam also outputs the values of “search-levels” and “search-width” even if these have not been explicitly specified in the data file (the latter is specified as “optimize-search-width”); this tells the user what the default values currently are.

Occam will always print out as it proceeds from level to level how many models are generated at each level and how many of these are kept. This lets the user track the progress of Occam. It also shows whether an exhaustive search is being done (all models generated are kept) or only a partial (heuristic) search is being done (only some generated models are kept, i.e., the lattice is being pruned).

Output file for directed system

Below are the output for the data given as an example in the DATA FILE section. This is a directed system with the DV being C and the IVs being A and B. The output has been sorted on Information.

MODEL	Level	H	dD F	LR	Alpha	Information	%dH(DV)
ABC	3	2.76118	3	10.6121	0.01397	1	0.563878
IV:AC:BC	2	2.76156	2	9.82045	0.00737	0.925395	0.52181
IV:BC	1	2.76182	1	9.29782	0.00207	0.876147	0.49404
IV:AC	1	2.76634	1	0.02839	0.86604	0.002676	0.001509
IV:C	0	2.76635	0	0	1	0	0

In the Model field, “IV” means a component with all the IVs in it, here AB. H is information-theoretic uncertainty (Shannon entropy). dDF is delta-degrees of freedom, the difference in df between a model and the reference model. The model for which dDF is 0 is the reference model. LR is the Likelihood-Ratio Chi-square (L^2 in Krippendorff), which is the error between a model and the reference model. Alpha is the probability of a Type I error, namely the probability of being in error if one asserts that the model is really the same as the reference model. Information is a measure of the constraint captured in a model, normalized to [0,1] range, namely $[T(\text{bottom}) - T(\text{model})] / T(\text{bottom})$. %dH(DV) is the % reduction in uncertainty of the DV (if there is only one DV) given the IVs in the predicting components (note that for this data, the reduction of uncertainty is very small, less than 1% even if one predicts with both IVs and these IVs interacting).

Note that only dDF, LR, and Alpha depend on the choice of reference model. Values of H, Information, and %dH(DV) are “absolute” and do not depend on reference model.

Output file for neutral system

If C is regarded, along with A and B, as an IV, then the system is neutral. Below are the measures for the larger lattice of neutral systems. Note that the column for uncertainty reduction is omitted because there are no DVs.

MODEL	Level	H	dDF	LR	Alpha	Information
ABC	0	2.76118	0	0	1	1
AB:AC:BC	1	2.76156	1	0.79171	0.37351	0.987028
		7		8	1	
AB:BC	2	2.76182	2	1.31435	0.51831	0.978465
		2		5	2	
AB:AC	2	2.76634	2	10.5837	0.00503	0.826589
		6		8	2	
AB:C	3	2.76635	3	10.6121	0.01397	0.826123
		9		8	5	
AC:BC	2	2.78641	2	51.7065	0	0.152807
		6		5		
A:BC	3	2.78643	3	51.7349	0	0.152341
				5		
AC:B	3	2.79095	3	61.0043	0	0.000465
		4		8		
A:B:C	4	2.79096	4	61.0327	0	0
		8		7		