SYSC 551 Discrete Multivariate Modeling

Prof. Martin Zwick

Portland State University

Notes from 1/19/2012 and 1/26/2012

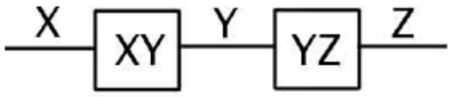
As taken by Juliana Arrighi

Structures

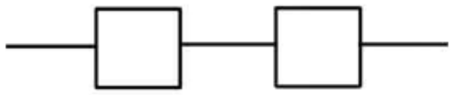
1. Introduction

Models and Structures.

A **structure** is a composition of relations, specified by listing component relations, e.g. AC:BC, or by a diagram. A structure is data-free (except for the cardinality of its variables). It does not have error, but it does have complexity, measured by degrees of freedom. Specific structures include information about particular variables, but structures can be represented more generally. For example, the structures XY:YZ, XY:XZ, and XZ:YZ all have the same general structure.

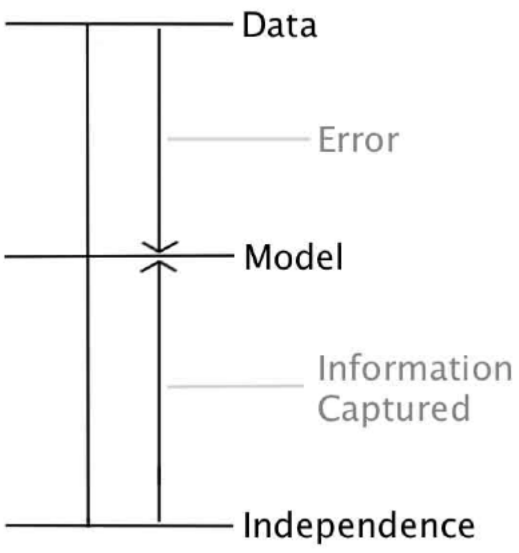


Specific Structure



General Structure

A **model** is a structure applied to some data. Models have both error and complexity (degrees of freedom). The saturated model, the relation that includes all of the variables, is the data and thus has no error. The goodness of a model depends on its error (or, conversely, information captured) and its complexity, i.e., degrees of freedom (or, conversely, simplicity). The best model is the one that has the best trade-off between these two. We want to minimize both error and complexity, and need to trade these off; or, conversely, we want to maximize both information and simplicity, and need to trade these off.

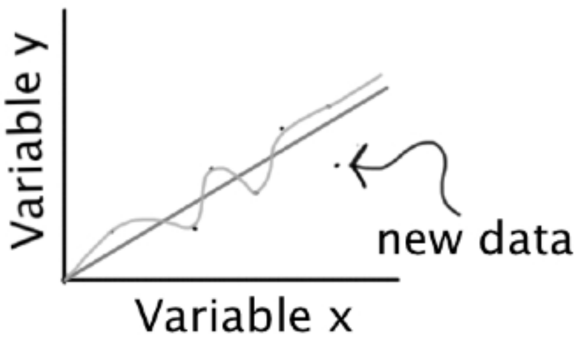


Degrees of freedom (d.f.) is the number parameters needed to specify a structure and is highest in the data.

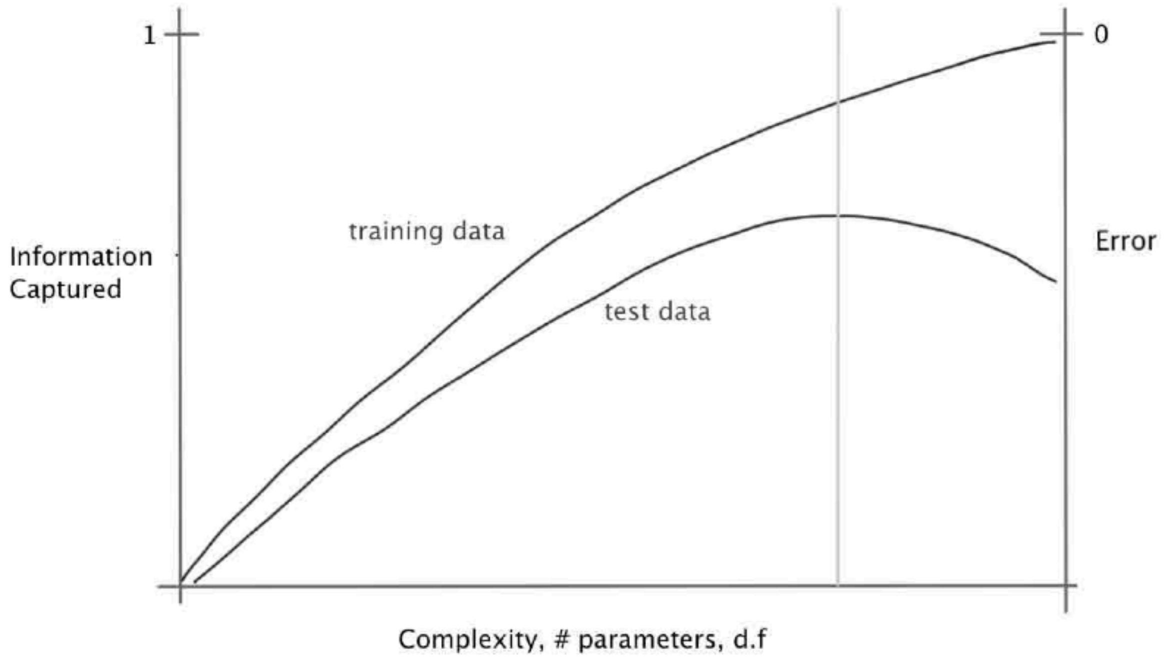
Error is the transmission between the data and the model. **Information captured** is the distance between the model of interest and the independence model (which is equal to the transmission of the independence model minus the transmission of the model). Information captured is lowest (by typical convention, 0%) in the independence model and highest (100%) in the data. (But one could use lower reference models than independence, e.g., the uniform distribution; in this case this distribution would be said to have 0% information captured.)

Fitting and Overfitting.

The goal in selecting a model selecting is to find the right balance between error and df so that the model most likely to be generalizable to other data of interest. It is possible to find a model that fits the data extremely well by increasing the complexity, or the number of parameters of a model. However, if the model fits particular data too well, the likelihood of the model fitting new data is low and it is not a very useful model.



Ideally one would find a level of complexity for which the model is most likely to fit new data. The goal is to find the "sweet spot" of complexity in which the model fits the data well but also generalizes well (indicated by the gray line in the figure below).



The test data should not be used to choose a model, but should be used only to verify the model selected with training data, so one must try to guess which models have an ideal balance between error and complexity. Data may be split into training and test data or test data and training data may be different data sets.

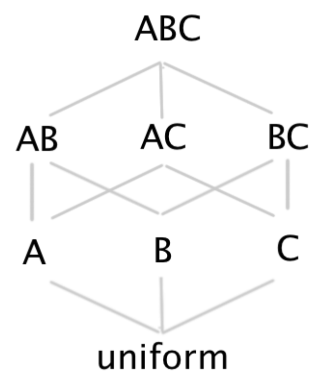
Methods of Selecting a Model.
Since we cannot use test data to select a model, these methods can be used to try to predict which model is best.

1. Use training data and statistical significance (p-value)
2. Use training data and an integrated measure (e.g., AIC, BIC)
3. Do 3-way splits of the data into training, pseudo-test, test: pick a model fit on training data based on how generalizable it is with pseudo test data

Then subject the model to a real – and final!! -- test by applying it to test data.

OCCAM gives percent correct, the percent of cases in which the outputs were correctly predicted by the model, as one of the measures of the goodness of the model. This is not an information theoretic measure so it can be used to compare RA to other techniques.

2. Lattice of Relations, Ordinality



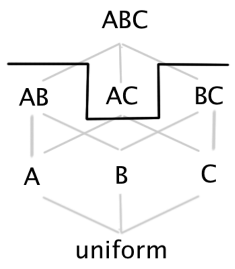
Ordinality is the number of variables in a relation. In the lattice of relations of three variables, the top level, ABC, has ordinality 3. In the second level, AB AC and BC have ordinality 2 and A, B, and C have ordinality 1.

Systemic relations are not just compositions of pair-wise relations. For example ABC is a three-way relation. This relation is not equivalent to three two-way relations, i.e., to the structure AB:AC:BC. On page 34, Krippendorf gives some examples of methods that assume pair-wise relationships, but in general, higher order relationships are possible. For, example network models usually only look at pair-wise relations, two nodes connected by one edge. However, three-way or higher relations can be represented by hypergraphs.

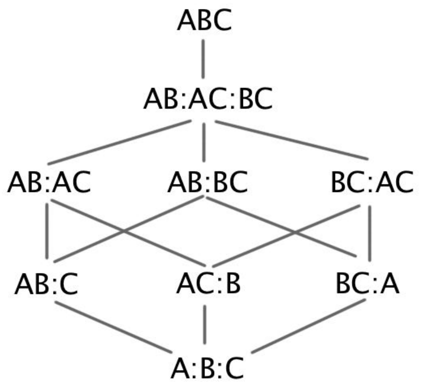
Constraint in the whole (ABC for a three variable system) is greater than or equal to the sum of the constraint in parts (e.g. AB). Another way of saying this is that decomposition generally decreases the constraint. This is a more specific and completely rigorous way of describing holism or "a whole is greater than the sum of its parts".

3. Lattice of Structures, Structure Types

A structure is a set of relations, and it can be represented as a cut through the lattice of relations, as shown at right. It includes the relations at the top of the cut plus all lower projected relations. For example, the structure AB:AC includes only the two-way relations AB and AC (and their embedded projections, A, B, and C) and excludes the relations ABC and BC.

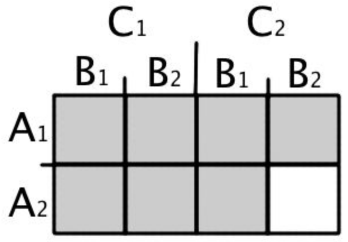


A structure can also be represented by a graph as described in the introduction. Since relations are of more importance than variables, relations are represented as boxes and the lines connecting them represent variables.



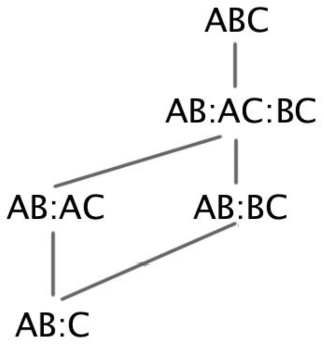
The lattice of structures gives the ways in which a number of things can relate. Krippendorf gives several different lattices of structure, both general and specific, on p. 40. Specific structures are what need to be considered when fitting data

Consider the lattice of structures for three binary variables A, B, C. For ABC there are 8 entries in the contingency table. The last entry in the contingency table can be inferred from the other 7 entries. (If the contingency table has probabilities in it, these have to sum to 1; if it has frequencies in it, these have the sum to the sample size, which is assumed to be known.) So degrees of freedom of ABC is 7. df(AB:AC:BC) is 6, and so on down the lattice, decreasing by 1 at every level. (All this only for binary variables.)



4. Directed Vs. Neutral Systems

In a **neutral system**, any variable could be considered an input or output, for example in AB, A could affect (or predict) B and B could affect (or predict) A. In a **directed system**, the inputs and outputs are specified and the relations are one way.



The lattice of structures for a directed system contains fewer structures than the neutral system with the same number of variables. The independence model for directed systems is the relation containing all of the inputs and each output as a separate relation (e.g. AB:C if A and B are input variables and C is an output variable). Directed system structures always have the relation containing all of the inputs to allow for interactions among the inputs. This also makes all the models hierarchically nested to allow for statistical tests.

5. Generating the Lattice of Neutral Structures

The algorithm for generating a descendent structure in the lattice of structures is as follows. (Example given below)

1. Remove a relation:
   There will be a unique descendent for each different relation that can be removed so the algorithm will be performed for each. When there are multiple symmetric relations, only one need be removed (if one is just interested in the general structure that results).

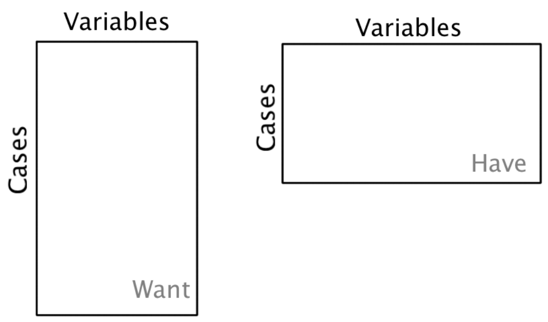
2. Restore embedded relations not already present:
   When restoring relations, consider all of the relations that are embedded in the one removed, but restore only those that are not embedded in remaining relations.

This algorithm will generate all possible general structures. If one wants to search for only models without loops a different algorithm would be needed.

6. Models With and Without Loops, Disjoint Models

For three variables there are five general structures and only one has loops. For four variables there are twenty general structures and ten have loops. As the number of variables increases, there is a higher proportion of general structures that have loops.

The Curse of the Lopsided Rectangle: Some models, especially those high complexity (df) require a lot of data, and in general information theory methods need much more data than, e.g., linear regression models. Ideally you would have many more cases than the number of variables, but often you have many variables and not enough cases to test some of the most complex models.

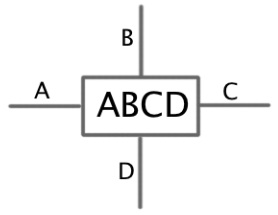


Algorithm for loop detection:

1. Remove any variable that appears in only one relation.
2. Remove relations imbedded in other relations
3. Repeat 1 & 2. If you get to a null structure, there are no loops in the original structure; otherwise, there are loops in the structure.

(Examples in Krippendorf, p. 42.)

Example: Generating the first six structures for four variable neutral system.

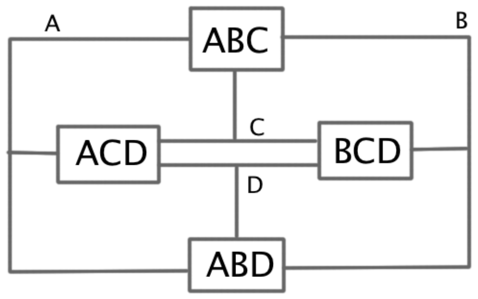


Start with the structure, ABCD, one relation among all four variables.

1. Remove a relation: There is only one relation to remove, ABCD

2. Restore embedded relations: All of the three variable relations need to be restored.

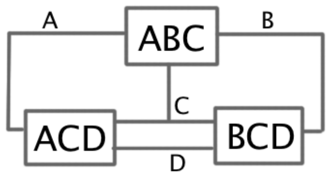
$$\Big\downarrow \quad - ABCD + ABC + ABD + ACD + BCD$$



1. Remove a relation: All of the relations here are equivalent, so we can choose any one to remove. ABD is removed here.

2. Restore embedded relations: The relations AB, AD, and BD are embedded in ABD which was removed, so we need to make sure they are included in the new structure. It turns out they are all embedded in the remaining relations—AB is in ABC, AD is in ACD, and BD is in BCD.

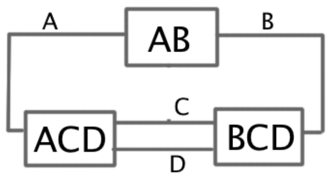
$$\Big\downarrow \quad - ABD + \cancel{AB} + \cancel{AD} + \cancel{BD}$$



1. Remove a relation: Again we have a symmetric model, so we can remove any relation. We will remove ABC.

2. Restore embedded relations: AB, AC, and BC are candidates for relations we need to restore, but AC is in ACD and BC is in BCD, so we only have to restore AB.

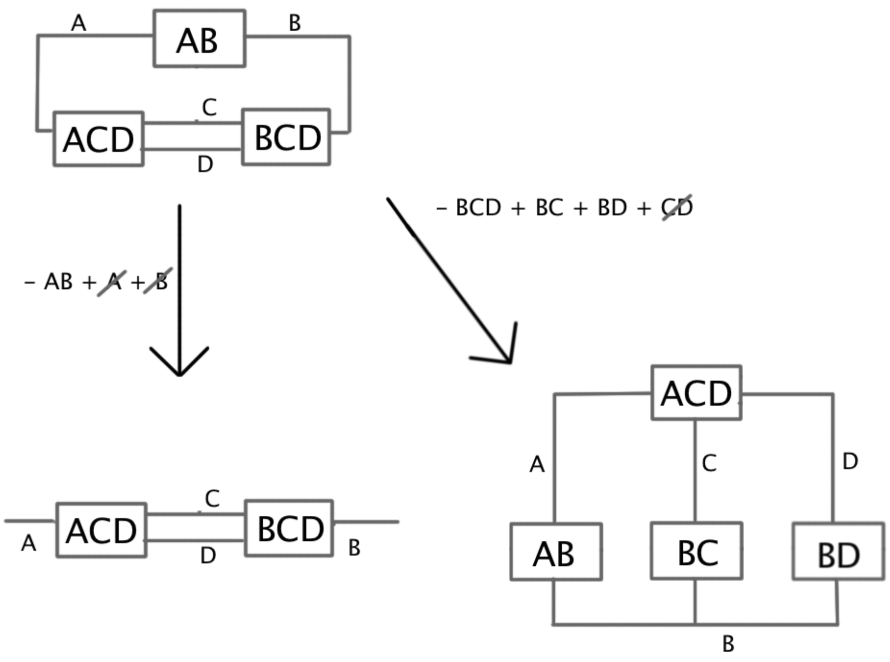
$$\Big\downarrow \quad - ABC + AB + \cancel{AC} + \cancel{BC}$$



Now, for the first time we have a structure that is not symmetric with respect to all of the relations. We will need to create two structures to show all possible types of general structure descendents.

1. Remove a relation: One relation we could remove is AB.

2. Restore embedded relations: Only A and B are embedded in AB, and we do not need to restore them because they are already included in the remaining relations.



1. Remove a relation: ACD and BCD are symmetric, so we only need to show the descendent from removing one of them. BCD is removed.

2. Restore embedded relations: BC, BD, and CD are embedded in BCD. Since CD is embedded in ACD, we do not restore it.