SYSC 551: Discrete Multivariate Modeling

Course Notes – Winter 2012

Professor Martin Zwick

Notes taken by Juliana Arrighi & checked by MZ

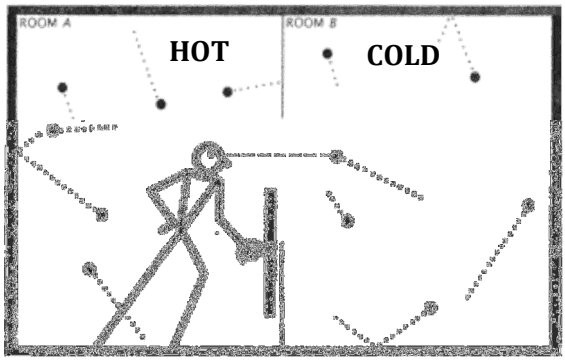
Part 1: BASIC CONCEPTS

1. Univariate Uncertainty, H; Diversity, Information

$$H(x) = -\sum_{j=1}^{n} p(x_j) \log p(x_j)$$

$$= -\sum p_i \log p_i$$

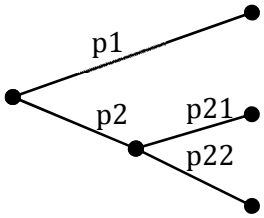
$$= \Gamma(p1, p2, \dots)$$



H increases with the number of states (n)

H increases with uniformity of probability

H = average weighted surprise



$$= \sum p_j \log(1/p_j)$$

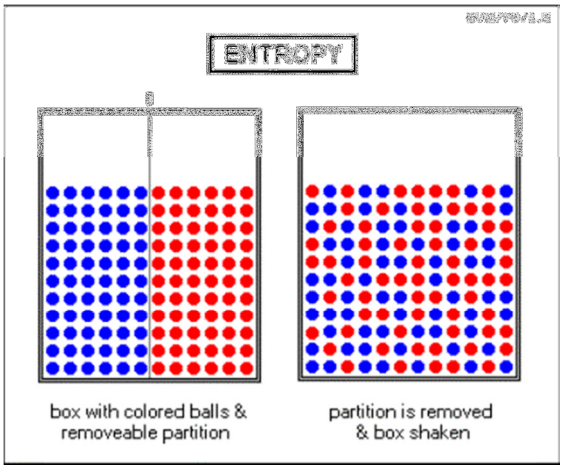
$$\{p(x_j)_t\} \rightarrow H(t)$$

$$\{p(x_j)_{t+1}\} \rightarrow H(t+1)$$

H is a measure of diversity.

Physical entropy $= -k \sum p_j \log p_j$

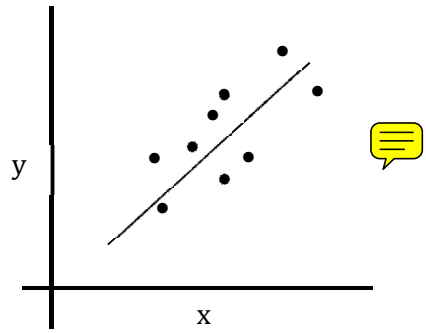
In Thermodynamics, the change in entropy (ΔS) is equal to the change in heat (ΔQ) over Temperature (T). (See image at right.)



$$\Delta S = \frac{\Delta Q}{T}$$

$$H(x) = -\sum p(x_j) \log p(x_j)$$

$$H_{initial} = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = 1$$



$$Information = -\Delta H$$

$$= -(H_{final} - H_{initial}) = H_{initial} - H_{final}$$

$$H_{final} = -1 \log 1 - 0 \log 0 = 0$$

2. Measures & Models

Notation:

x, y variables (lower case letters)
XY models (capital letters)

For two variables:
XY is the saturated model, or the data
X:Y is the independence model

For three variables, more intermediate models are possible:

XYZ

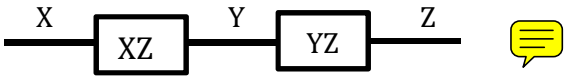
XY:YZ:XZ

XY:YZ XY:XZ XZ:YZ

XY:Z XZ:Y YZ:X

X:Y:Z

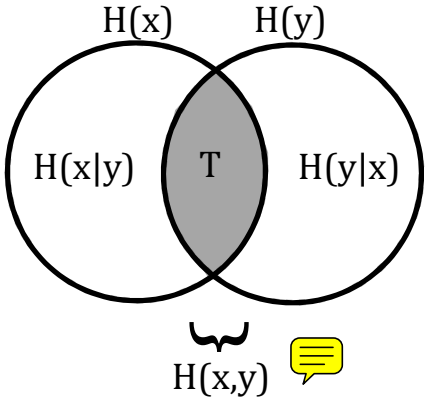
The model XZ:YZ can be depicted as this:



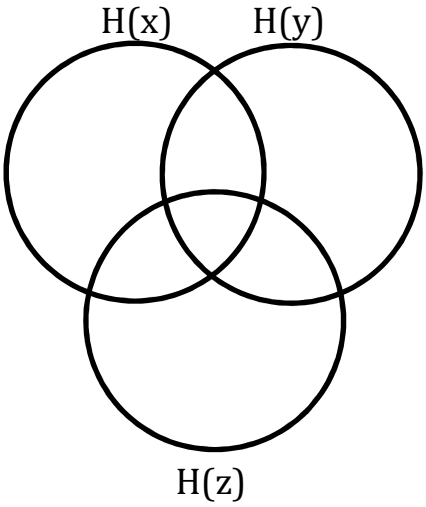
$$H(XY:YZ) = -\sum\sum\sum q_{XY:YZ}(x_j, y_k, z_l)$$

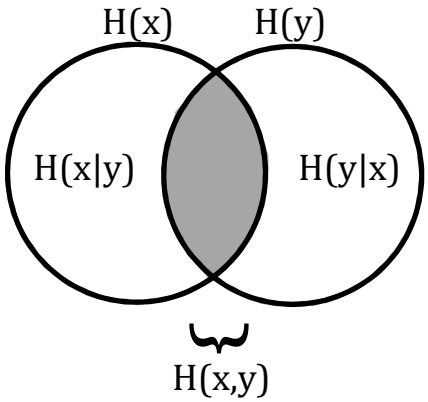
$$= \log q_{XY:YZ}(\ldots)$$

$$H_x(y) \equiv H(y|x)$$



3. Bivariate & Conditional Uncertainties





$p(x_j, y_k) = p(x_j)p(y_k|x_j)$

$= p(y_k)p(x_j|y_k)$

$\neq p(y_k)p(x_j)$

$H(\mathrm{x},\mathrm{y}) = \mathrm{H}(\mathrm{x}) + \mathrm{H}(\mathrm{y}|\mathrm{x})$

$= H(y) + H(x|y)$

$$= -\sum_j \sum_k p(x_j, y_k) \log p\,(x_j, y_k)$$

$$= -\sum_j \sum_k p(x_j)p(y_k|x_j) \log p(x_j)p(y_k|x_j)$$

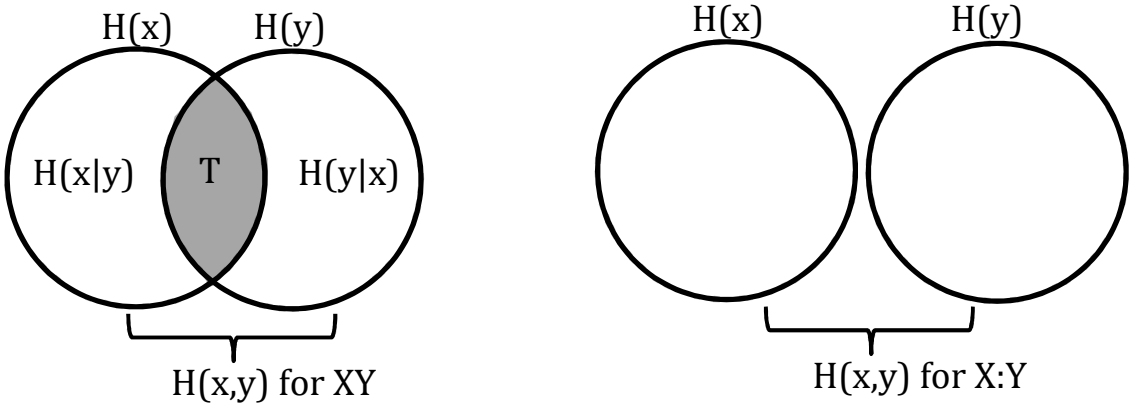
$$= -\sum_j \sum_k p(x_j)p(y_k|x_j) \log p(x_j) - \sum_j \sum_k p(x_j)p(y_k|x_j) \log p(y_k|x_j)$$

$$= -\sum_j p(x_j) \log p(x_j) \sum_k p(y_k|x_j) - \sum_j p(x_j) \sum_k p(y_k|x_j) \log p(y_k|x_j)$$

Note that $\sum_k p(y_k|x_j) \log p(y_k|x_j) = \sum p(x_j)H(y|x_j)$

$$H(y|x) = \sum_j p(x_j)H(y|x_j)$$

4. Transmission, T (Mutual Information, Constraint)



Transmission (T) = Mutual Information

This is equal to constraint in the data, or XY, due to association between variables.

Transmission is also equal to the error in X:Y, or the independence model

$T = H(x) + H(y) - H(x,y)$

$= H(X:Y) - H(XY)$

$= H(X) + H(Y) - H(XY)$

$T(model) = H(model) - H(data)$

Example:
$T(XY:XZ:YZ) = H(XY:XZ:YZ) - H(XYZ)$

Transmission of a model is equal to the entropy of the model minus the entropy of the data.

5. Computations on Contingency Tables

Observed probability distribution, for the model XY (or the data):

	y_1	y_2	
x_1	.1	.2	.3
x_2	.3	.4	.7
	.4	.6	

If X and Y are independent, you should get this distribution, for the model X:Y.

	y_1	y_2	
x_1	.12	.18	.3
x_2	.28	.42	.7
	.4	.6	

For the data (XY), $H(x,y) = \Gamma(.1,.2,.3,.4)$

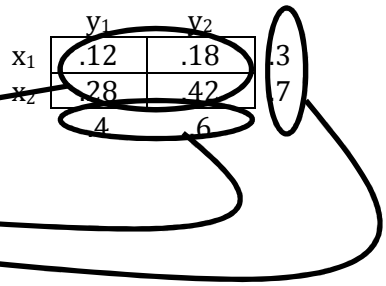
$H(x) = \Gamma(.3,.7)$

$H(y) = \Gamma(.4,.6)$

$T(model) = H(model) - H(data)$

$T = H(X:Y) - H(XY)$

$T = H(x) + H(y) - H(x,y)$



$T=\Gamma(.3,.7) + \Gamma(.4,.6) - \Gamma(.1,.2,.3,.4)$

For Three Variables:

The values in the table below indicate the observed probabilities for three variables, x,y,z.

	z_1		z_2	
	y_1	y_2	y_1	y_2
x_1	a	b	c	d
x_2	e	f	g	h

Three two-way projections can be derived from this dataset:

	y_1	y_2
x_1	a+c	b+d
x_2	e+g	f+h

	z_1	z_2
x_1	a+b	c+d
x_2	e+f	g+h

	z_1	z_2
y_1	a+e	c+g
y_2	b+f	d+h

Additionally, two-way projections can be made for individual variables:

x_1	a+b+c+d
x_2	e+f+g+h

y_1	a+e+c+g
y_2	b+f+d+h

z_1	a+b+e+f
z_2	c+d+g+h

$$H(z|x,y) = H(x,y,z) - H(x,y)$$

$$H(x,y,z) = \Gamma(a,b,c,d,e,f,g,h)$$

$$H(x,y) = \Gamma(a+c, b+d, \dots)$$

$$H(z|x,y) = \sum\sum p(x_j, y_k)\, H(z|\, x_j, y_k)$$

$$H(z|x_1,y_2) = \Gamma\left(\frac{b}{b+d}, \frac{d}{b+d}\right)$$

$$= p(x_1,y_1)\Gamma(\ \ ) + p(x_1,y_2)\Gamma\left(\frac{b}{b+d}, \frac{d}{b+d}\right) + \cdots$$

$$\uparrow$$

$$b+d$$

6. A State Decomposition of Univariate Uncertainty

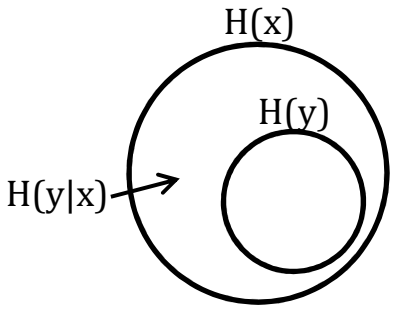
In the table below, x is a macrostate with n subsystems

x_1		x_2	
y_1	y_2	y_1	y_2
a	b	c	d

$H_{total} = H_{within\ subsystems} + H_{between\ subsystems}$

$H(y)$is within subsystems (micro), $H(x)$ is between subsystems (macro).

$H(x)$ is contained within $H(y)$



$$H_{total} = \underbrace{(a+b)\Gamma\left(\frac{a}{a+b},\frac{b}{a+b}\right)}_{p(x_1)H(y|x_1)} + \underbrace{(c+d)\Gamma\left(\frac{c}{c+d},\frac{d}{c+d}\right)}_{p(x_2)H(y|x_2)} + \Gamma(a+b,c+d)$$

For Neutral Systems
ABCD

A:B:C:D

For Directed Systems
ABCD…Z

ABC…:AZ

ABC…:Z

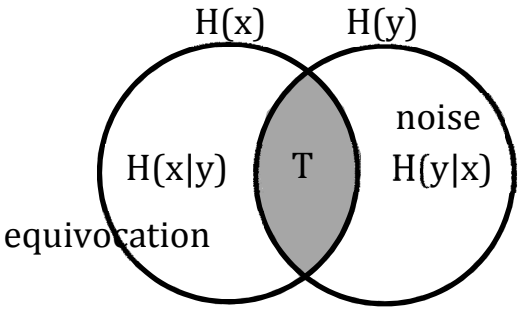
7. T in 'Transmission' & 'Sequential' Situations

Transmission (mutual information) includes

Transmission situation

Sequential Situation

For the Transmission Situation:
x = message sent
y = message received



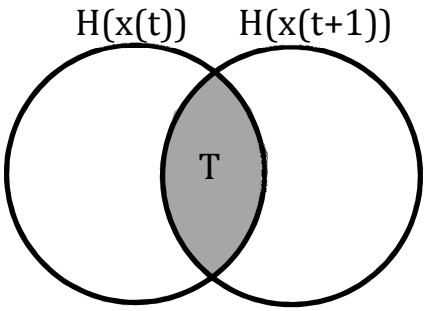
$T = H(x) + H(y) - H(x, y)$

$=$ sent and received

$T = H(y) - H(y|x)$

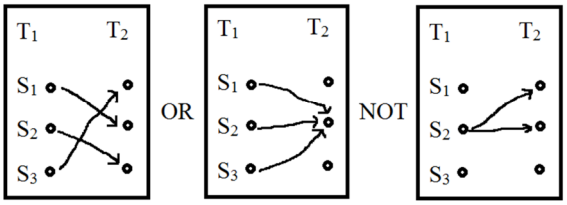
$= H(x) - H(x|y)$

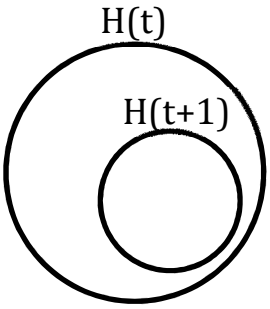
For the Sequential Situation:



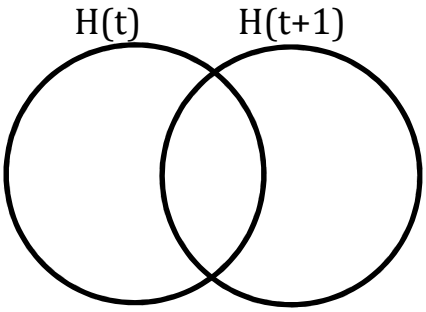
H(x(t)) might also be called H(t), and H(x(t+1)) might also be called H(t+1).

A system is deterministic if H(t+1) is contained within H(t).





A system is stochastic if H(t+1) is not contained within H(t).



Markov:

$p(x_1)$
$p(x_2)$
$p(x_3)$
…
$p(x_n)$

t+1
(nx1)

=

a	b	c	…
…			

(nxn)

$p(x_1)$
$p(x_2)$
$p(x_3)$
…
$p(x_n)$

t
(nx1)

$p(x_1)_{t+1} = ap(x_1)_t + bp(x_2)_t + \cdots$

$a = p\big(x_1(t+1)\big|x_1(t)\big)$

$b = p\big(x_1(t+1)\big|x_2(t)\big)$

8. T as Likelihood Ratio; Relation to Uncertainty

$$T(X:Y) = \sum\sum p(x,y)\log_2\left(\frac{p(x,y)}{q_{X:Y}(x,y)}\right)$$

$p(x,y) = observed$

$q_{X:Y}(x,y) = calculated$

$q_{X:Y}(x,y) = p(x)p(y)$

p

	y_1	y_2	
x_1	.1	.2	.3
x_2	.3	.4	.7
	.4	.6	

q

	y_1	y_2	
x_1	.12	.18	.3
x_2	.28	.42	.7
	.4	.6	

$T = H(x) + H(y) - H(x,y)$

$= \Gamma(.3,.7) + \Gamma(.4,.6) - \Gamma(.1,.2,.3,.4)$

$= .1\log_2\frac{.1}{.12} + .2\log_2\frac{.2}{.18} + .3\log_2\frac{.3}{.28} + .4\log_2\frac{.4}{.42}$

$L^2 = likelihood\ ratio\ chi\ square = 2N\sum\sum p\log_e\frac{p}{q}$

$= 1.3863NT$

Degrees of Freedom

For the model, df = 0

.33	.33	.33

For the data, df = 2

.4	.35	.25

$T(model) = H(model) - H(data)$

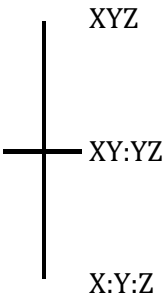
$= \Gamma(.33,.33,.33) - \Gamma(.4,.35,.25)$

$= \sum p\log\frac{p}{q_{model}}$

$= .4\log\frac{.4}{.33} + .35\log\frac{.35}{.33} + .25\log\frac{.25}{.33}$

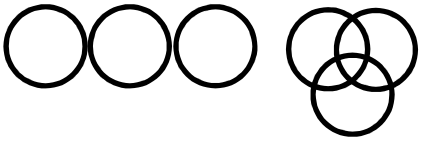
$L^2 = 1.3863NT$

$-\sum_{j=1}^{n} p\log p \quad \overrightarrow{all\ p_j\ equal} \quad \log_2 n$



$T(X:Y:Z) = H(X:Y:Z) - H(XYZ)$

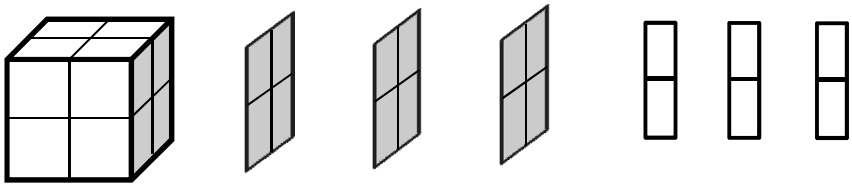
$= H(x) + H(y) + H(z) - H(x,y,z)$



$T(XY:YZ) = H(XY:YZ) - H(XYZ)$

$T(model) = H(model) - H(data)$

$H(XY:YZ) = H(XY) + H(YZ) - H(Y)$



The dataset XYZ contains three two-way relations (XY, XZ, YZ), and three one-way relations (X, Y, Z)

$H(XY:YZ:XZ) \neq H(XY) + H(YZ) + H(XZ) - H(X) - H(Y) - H(Z)$

It is not possible to calculate the entropy (or transmission) when the model has a loop!

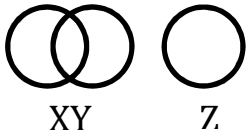
9. T, H for Trivariate (& Higher) Relations

$T(X:Y:Z) = H(X:Y:Z) - H(XYZ)$

$T(XY:Z) = H(XY:Z) - H(XYZ)$

$T(model) = H(model) - H(data)$

$H(XY:Z) = H(XY) + H(Z)$



$H(XY:YZ) = H(XY) + H(YZ) - H(Y)$

$q(XY:YZ) = \frac{p(XY)p(YZ)}{p(Y)}$

Law of Uniform Subscripting:

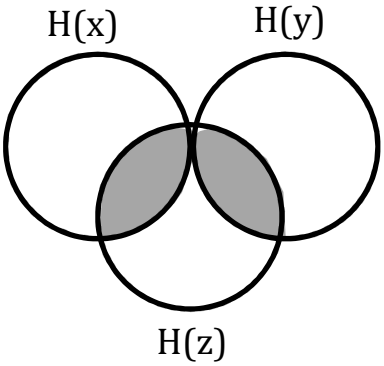
$H(X:Y) = H(X) + H(Y)$

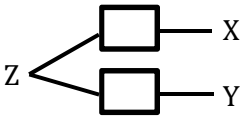
$H(X:Y|Z) = H_Z(X:Y) = H_Z(X) + H_Z(Y)$

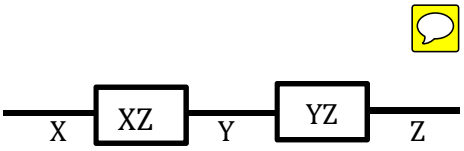
Law of distribution for conditional T

$T_Z(X:Y) = T(XZ:ZY)$

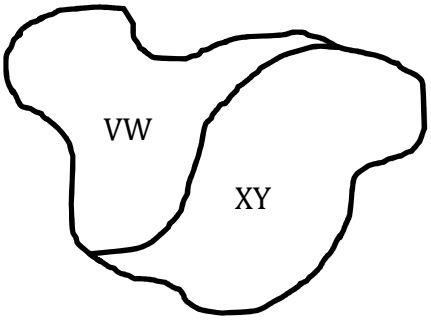
$T_Z(X:Y) \gtreqless T(XZ:ZY)$







10. A Variable Decomposition of Transmission



$T(V:W:X:Y) = T(V:W) + T(X:Y) + T(VW:XY)$

$T(V:W) + T(X:Y) =$ within subsystems

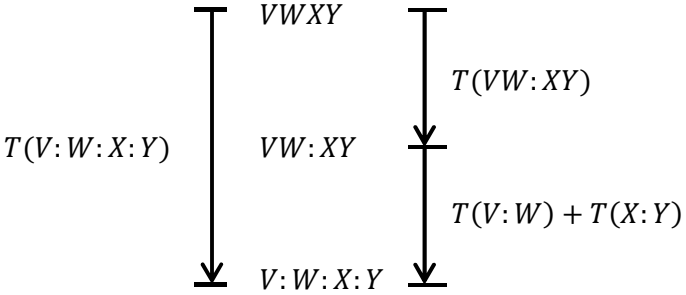
$T(VW:XY) =$ between subsystems

$T(V:W:X:Y) = H(V) + H(W) + H(X) + H(Y) - H(VWXY)$

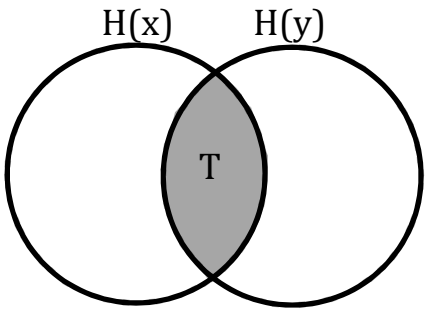
$T(X:Y) = H(X) + H(Y) - H(XY)$

$T(V:W) = H(V) + H(W) - H(VW)$

$T(VW:XY) = H(VW) + H(XY) - H(VWXY)$



11. Other Information Theoretic Functions



$$\frac{T(X:Y)}{T_{max}(X:Y)} = \frac{T(X:Y)}{\min\{H(X), H(Y)\}}$$

$$\frac{T(X:Y)}{H(Y)} = fraction\ of\ entropy\ reduced$$

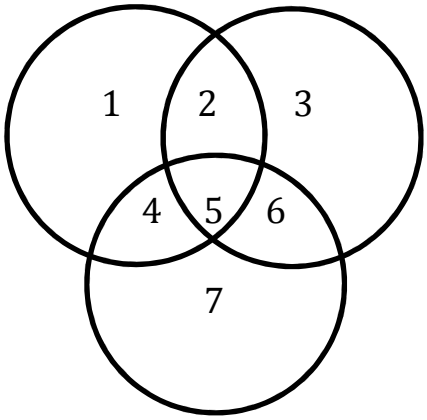
$$\frac{T(X:Y)}{H(X)} = \text{"}predictive\ efficiency\text{"}$$

$$\frac{T(X:Y)}{H_{max}} = 1 - \frac{H}{H_{max}} = redundancy$$

$$H_{max} = H(X) + H(Y)$$

Quastler's A Function

Area "5" in the diagram at right.



$$A(X,Y,Z) = -H(X) - H(Y) - H(Z) + H(XY) + H(YZ) + H(XZ) - H(XYZ)$$

XYZ

$T(XY:XZ:YZ)$

$XY:XZ:YZ$

$$T(XY:XZ:YZ) \neq A(X,Y,Z)$$

$H(XY:XZ:YZ)$ has no algebraic form!

Quastler's A Function can be positive or negative.

	R_{Happy}		$R_{Unhappy}$	
	$W_{mountain}$	$W_{seashore}$	$W_{mountain}$	$W_{seashore}$
$H_{mountain}$	10	0	0	10
$H_{seashore}$	0	10	10	0

Three two-way projections can be derived from this dataset:

	$W_{mountain}$	$W_{seashore}$
$H_{mountain}$	10	10
$H_{seashore}$	10	10

	R_{Happy}	$R_{Unhappy}$
$H_{mountain}$	10	10
$H_{seashore}$	10	10

	R_{Happy}	$R_{Unhappy}$
$W_{mountain}$	10	10
$W_{seashore}$	10	10

$$T_R(H:W) = p(R_{Happy})T_{Happy}(H:W) + p(R_{Unhappy})T_{Unhappy}(H:W)$$

When R_{Happy}

	$W_{mountain}$	$W_{seashore}$
$H_{mountain}$	10	0
$H_{seashore}$	0	10

	$W_{mountain}$	$W_{seashore}$	
$H_{mountain}$	.5	0	.5
$H_{seashore}$	0	.5	.5
	.5	.5	

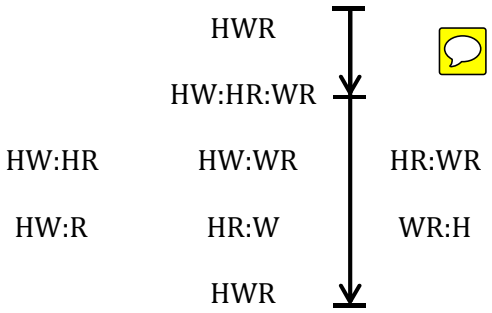
$$H(Husband) = 1$$

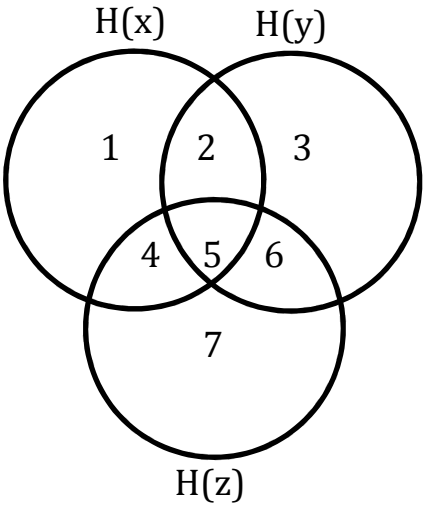
$$H(Wife) = 1$$

$$H(Husband, Wife) = 1$$

$$T_{Happy}(Husband:Wife) = 1 + 1 - 1 = 1$$

$$H(HusbandWife) = 2 = H(Husband) + H(Wife)$$





$T(X:Y:Z) = H(X:Y:Z) - H(XYZ)$

$1 + 2 + 4 + 5 + 2 + 3 + 5 + 6 + 4 + 5 + 6 + 7$

$\cancel{1} + \cancel{2} + \cancel{4} + \cancel{5} + 2 + \cancel{3} + 5 + \cancel{6} + 4 + 5 + 6 + \cancel{7}$

$= 2 + 4 + 6 + 5 + 5$

$= 2 + 4 + 6 + 5 = System\ Entropy$

$= H(XYZ) - \underbrace{H_{XY}(Z) - H_{XZ}(Y) - H_{YZ}(X)}$

noise
unique variability

Structures

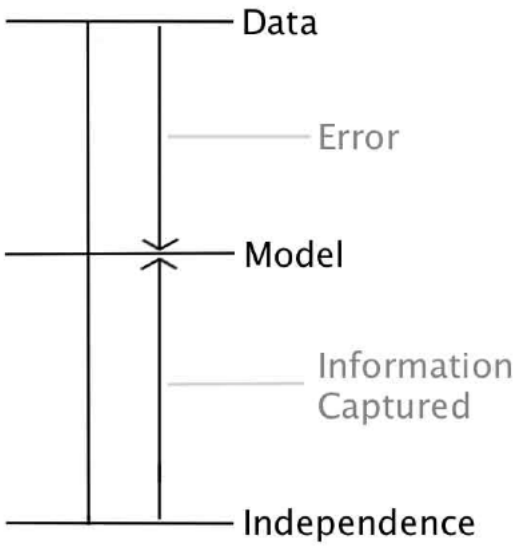
1. Introduction

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.

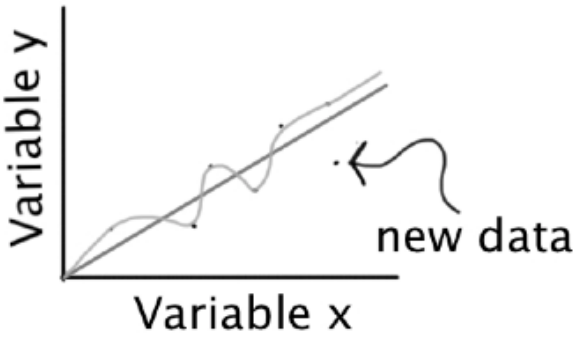


Degrees of freedom (d.f.) is the number parameters needed to specify a structure and is highest in the data.

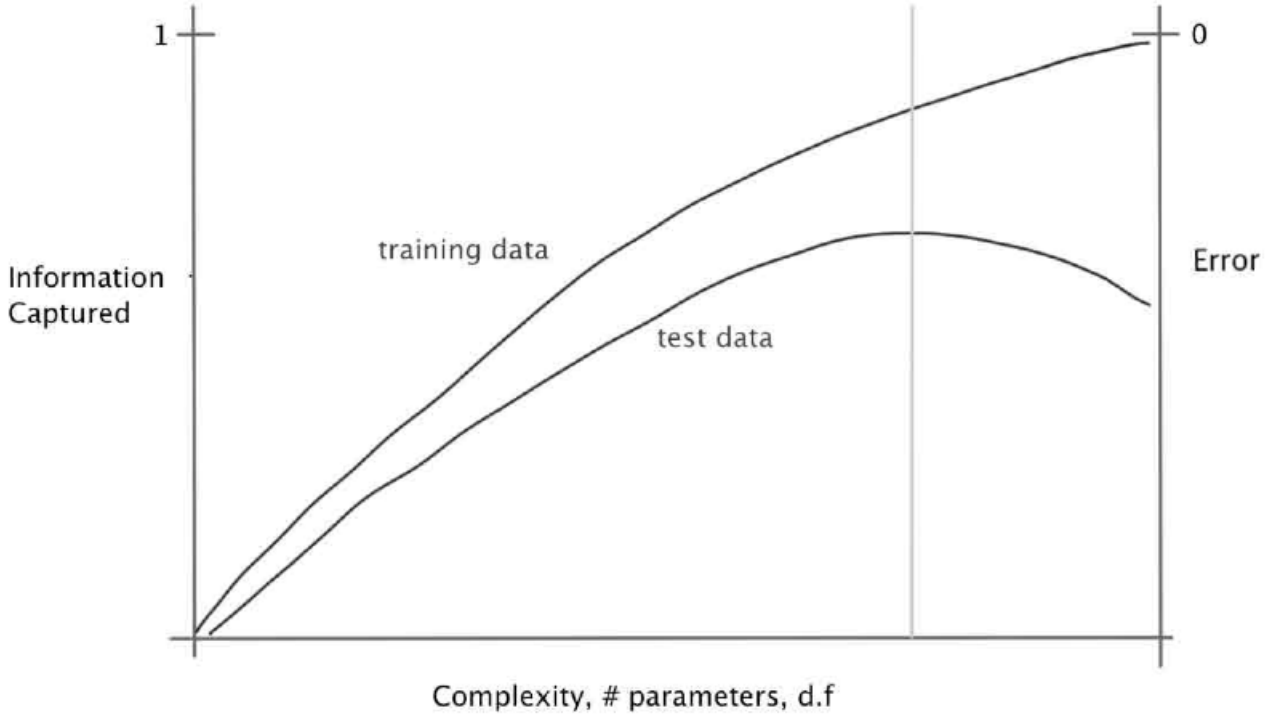
Error is the transmission between the data and the model. **Information captured** is the distance between the model of interest and the independence model (which is equal to the transmission of the independence model minus the transmission of the model). Information captured is lowest (by typical convention, 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 then to find a "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.

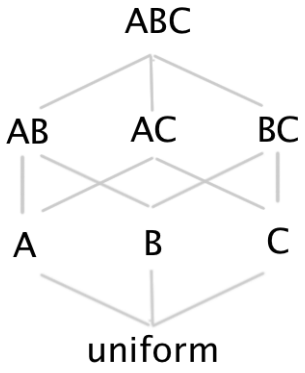
Methods of Selecting a Model.
Since we cannot use test data to select a model, these methods can be used to try to predict which model is best.

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

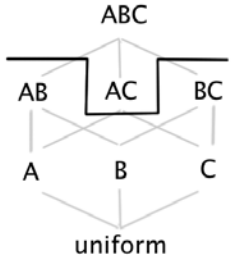


Ordinality is the number of variables in a relation. In the lattice of relations of three variables, the top level, ABC, has ordinality 3. In the second level, AB AC and BC have ordinality 2 and A, B, and C have ordinality 1.

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".

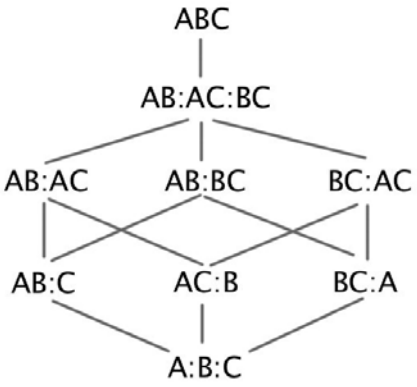
3. Lattice of Structures, Structure Types



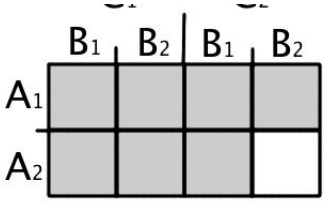
A structure is a set of relations, and it can be represented as a cut through the lattice of relations, as shown at right. It includes the relations at the top of the cut plus all lower projected relations. For

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

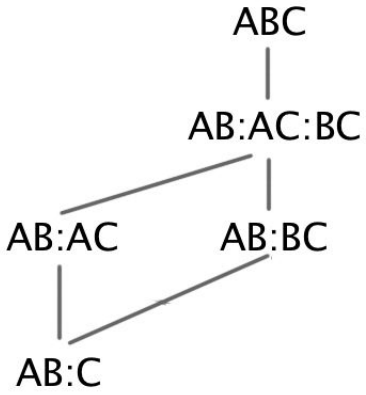


For a system of three binary variables, each level has one less degree of freedom than the one above it. For ABC with binary variables, there are 8 entries in the contingency table. The last entry in the contingency table can be inferred from the other entries. (If the contingency table has probabilities 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.

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.

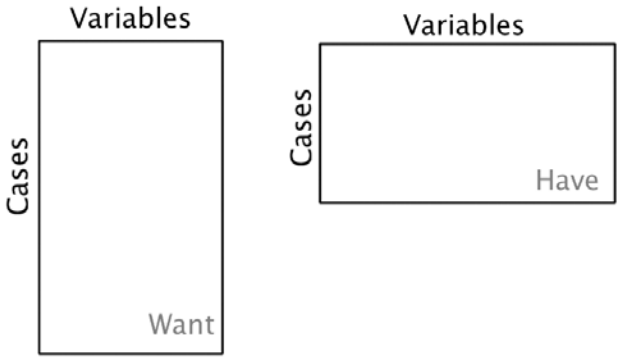
(See the example below.)

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.

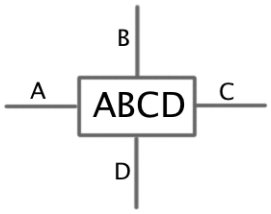
The Curse of the Lopsided Rectangle: Some models, especially those high complexity (df) require a lot of data, and in general information theory methods need much more data than, e.g., linear regression models. Ideally you would have many more cases than the number of variables, but unfortunately all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



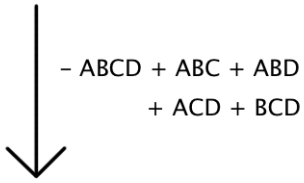
Algorithm for loop detection:

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. Some examples are given in Krippendorf, p. 42.

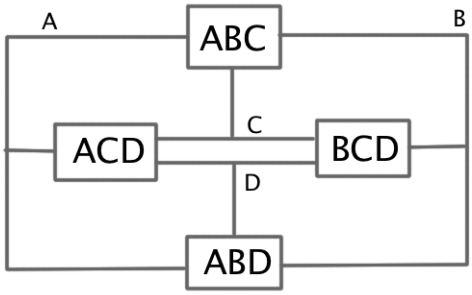
Example: Generating the first six structures for four variable neutral system.



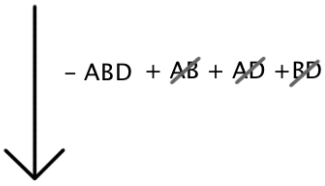
Start with the structure, ABCD, one relation among all four variables.

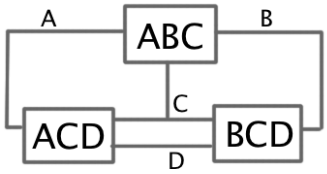


1. Remove a relation: There is only one relation to remove, ABCD
2. Restore embedded relations: All of the three variable relations need to be restored.

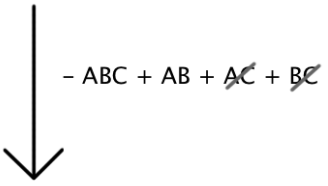


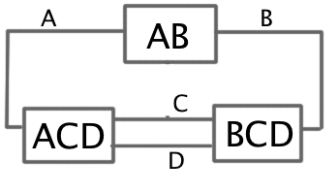
1. Remove a relation: All of the relations here are equivalent, so we can choose any one to remove. ABD is removed here.
2. Restore embedded relations: The relations AB, AD, and BD are embedded in ABD which was removed, so we need to make sure they are included in the new structure. It turns out they are all embedded in the remaining relations—AB is in ABC, AD is in ACD, and BD is in BCD.





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.

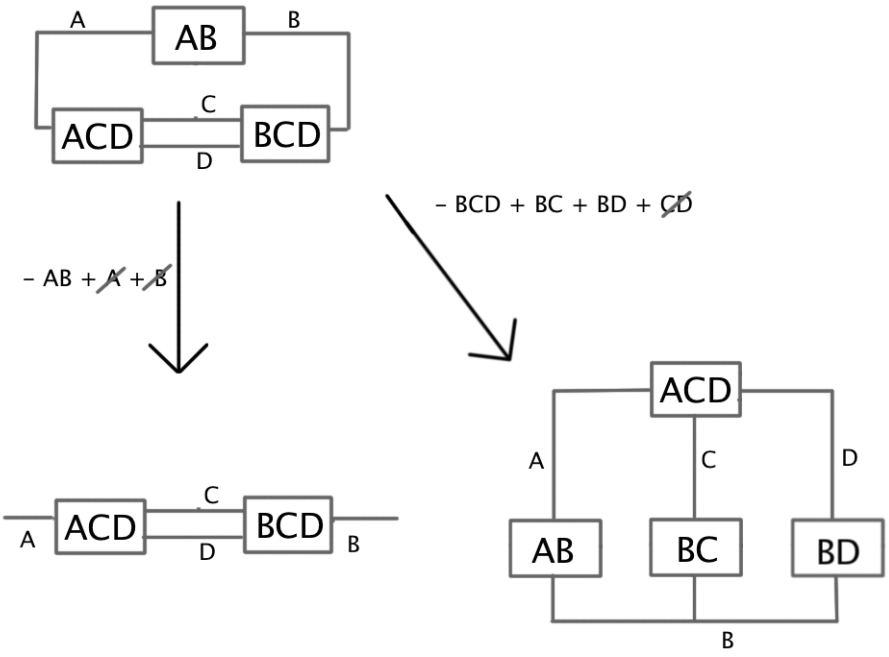




Now, for the first time we have a structure that is not symmetric with respect to all of the relations. We will need to create two structures to show all possible types of general structure descendents.

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.

3. Lattice of Structures, Structure Types

Nearest Common Ancestor, Nearest Common Descendent (Krippendorf p. 39)

If two different structural models have high goodness measures, we may look either to the (a) nearest common ancestor or the (b) nearest common descendent to (a) merge the two models, and get what's in both of them or (b) select only what they have in common that makes them good models.

To find the **nearest common ancestor** of two structural models in the lattice of structures, take the union of the relations of the two models; that is, combine all component relations of each and eliminate redundancies. For example, the nearest common ancestor of the structural models m_1 = AC:BCDE and m_2 = ABD:CD:CE could be found as follows:

$$m_1 \cup m_2 = \text{AC:BCDE} \cup \text{ABD:CD:CE} = \text{AC:BCDE:ABD:CD:CE} = \text{AC:BDCE:ABD}$$

The relations CD and CE were eliminated because they are embedded in BCDE.

To find the **nearest common descendent**, take the intersection of the two models. The intersection includes all relations that are either components or are embedded in the components of *both* models. For example

$$m_1 \cap m_2 = \text{AC:BCDE} \cap \text{ABD:CD:CE} = \text{A:BD:CD:CE}$$

A is in both models because it is embedded in both AC and ABD. BD is embedded in BCDE and ABD, and so on. A systematic method for determining the intersection of two structural models is as follows:

1. List all relations and projections of m_1 and m_2
2. Cross out any relation not present on both sides (double strike)
3. Cross out any redundant relation (single strike)

See the following table:

~~AC~~	~~BCDE~~		∩	~~ABD~~	CD	CE
A	~~BCD~~	~~BDE~~	|	~~AB~~	~~C~~	~~C~~
~~C~~	~~BCE~~	~~CDE~~	|	~~AD~~	~~D~~	~~E~~
	~~BC~~	CD	|	BD		
	BD	CE	|	A		
	~~BE~~	~~DE~~	|	~~B~~		
	~~B~~	~~D~~	|	~~D~~		
	~~C~~	~~E~~	|			

6. Models With and Without Loops, Disjoint Models

Disjoint Models

Disjoint models are those that have no overlap in their components. We will make a distinction in the criteria for directed and neutral systems.

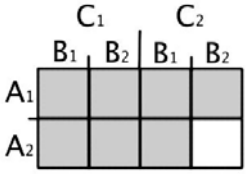
In a neutral system, a disjoint model will have no overlap in any relations.
Example: AB:CDE
In a directed system, no independent variables overlap in *predicting* relations.
Example: IV:AZ:BCZ, where IV is the relation of all independent variables

It is important to distinguish between disjoint models and loopless models. In neutral systems, disjoint models are only a subset of loopless models, but in directed systems a disjoint model may contain loops as in the example above. Also unlike with disjoint models, the criteria for looped models is the same in directed and neutral systems.

7. Degrees of Freedom

Krippendorff Method for calculating df, p.48-53

For ABC, df= |ABC| - 1, where |structure| = number of states in the structure
Let cardinality of A be N_A



$df_{ABC} = N_A N_B N_C - 1$
For $N_A = N_B = N_C = 2$, $df = 2 \cdot 2 \cdot 2 - 1 = 7$
For models lower on the lattice of structure, e.g. AB:AC:BC, add the df of the components and subtract the overlap between components.

$df(AB:AC:BC) = df(AB) + df(AC) + df(BC) - df(A) - df(C) - df(B)$

For $N_A = N_B = N_C = 2$, $df(AB:AC:BC) = 3 + 3 + 3 - 1 - 1 - 1 = 6$

For ABC:ABD:ACD:BCD, add the df of the components, subtract the df of the overlap between each pair (double overlap) and add the df of the overlap among each set of three components (triple overlap).

Double overlap:
ABC ∩ ABD = AB
ABC ∩ ACD = AC
ABC ∩ BCD = BC
ABD ∩ BCD = BD
ABD ∩ ACD = AD
ACD ∩ BCD = CD

Triple overlap:
ABC ∩ ABD ∩ ACD = A
ABC ∩ ABD ∩ BCD = B
ABC ∩ ACD ∩ BCD = C
ABD ∩ ACD ∩ BCD = D

$$df(ABC:ABD:ACD:BCD) = df(ABC) + df(ABD) + df(ACD) + df(BCD) - df(AB) - df(AC) - df(BC) - df(BD) - df(AD) - df(CD) + df(A) + df(B) + df(C) + df(D)$$

For $N_A = N_B = N_C = N_D = 2$,

$$df(ABC:ABD:ACD:BCD) = 4 \times 7 - 6 \times 3 + 4 \times 1 = 14$$

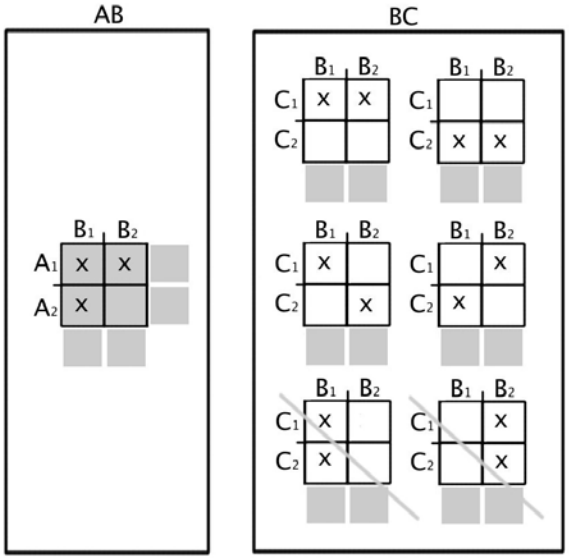
For AB:AC

$$df(AB:AC) = df(AB) + df(AC) - df(A)$$

Note that you can replace df by H to get an entropy equation, **except** when there are **loops** in the structure. Remember that the algebra doesn't work for entropy in these structures with loops, but it does work for df. For example the df of ABC:ABD:ACD:BCD was determined algebraically above, but since the structure has loops, H could not be calculated this way.

Contingency table examples for df.

The table example for ABC was given above. The data table for ABC where |A| = |B| = |C| = 2 has 8 values. However only 7 need to be specified. The eighth can be determined by subtracting the other probability values from 1 (or the frequency values from the total sample size).
For AB:BC, there are two tables, one for AB and one for BC. Three values need to be specified in AB and only two need to be specified in BC. In the figure, an x represents a specified value. And

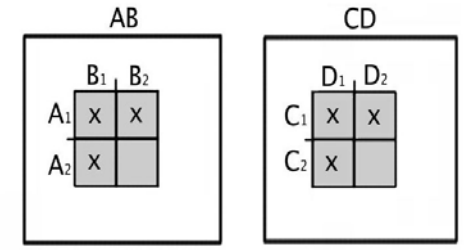


gray boxes represent values that can be determined from the information specified in the AB table. Both of the B margins are known in the BC table because they can be determined from the AB table. Now, only two more values need to be specified in BC, one in the B_1 column and one in the B_2 column. The remaining values can be obtained by subtracting the specified values from the appropriate B margin value. Specifying both of the values in either column would not be enough since the two values in either column are not independent.

Compare this with the Krippendorff method:

df(AB:BC) = df(AB) + df(BC) – df(B) = 3 + 3 -1 = 5

For AB:CD there are two tables, one for each relation, but for this structure there is no overlap (this is a disjoint structure). Three values need to be specified in each table.



df(AB:CD) = df(AB) + df(CD) = 3 + 3 = 6

Log-Linear method for calculating df Knoke and Burke p 36-37

Write down all relations and their projections but do not duplicate projections.
For each relation, multiply one less than the cardinalities of each variable. Add the values for each relation to get df of the structure.

Example: MER:MV:EV, where |M| = |R| = |V| = 2, |E| = 3
Log-linear method:

Relations	Product of cardinalities minus one	
MEV	(2-1)(3-1)(2-1)	= 2
ME	(2-1)(3-1)	= 2
MR	(2-1)(2-1)	= 1
ER	(3-1)(2-1)	= 2
M	(2-1)	= 1
E	(3-1)	= 2
R	(2-1)	= 1
MV	(2-1)(2-1)	= 1
V	(2-1)	= 1
EV	(3-1)(2-1)	= 2
	Total	= 15

Krippendorff Method:

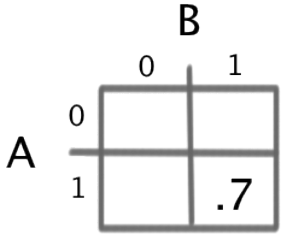
df(MER:MV:EV) = df(MER) + df(MV) + df(EV) – df(M) – df(E) – df(V)
= (2·3·2 – 1) + (2·2 – 1) + (3·2 – 1) – (2 – 1) – (3 – 1) – (2 – 1)
= 11 + 3 + 5 – 1 – 2 – 1 = 15

Log-Linear method is very good for calculating Δdf between two models, since the relations in common can be ignored. The Krippendorff and log-linear methods for calculating df do not apply to models with structural zeros. (e.g. pregnant males)

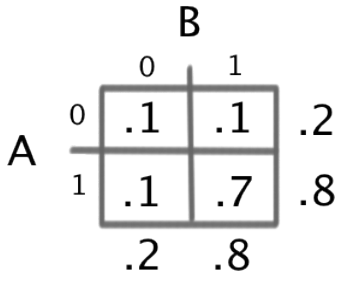
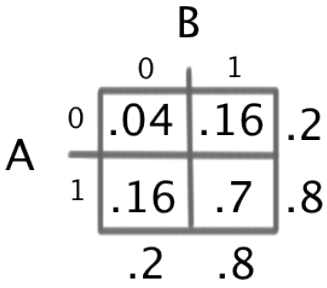
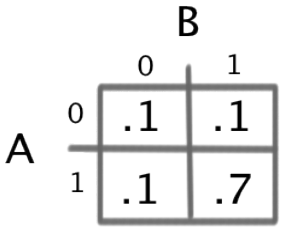
8. State Based and Latent Variables

State-Based Models

State-based models specify particular values in the table.
For example A_1B_1 is a state based model. It specifies the value of A_1B_1 in the AB table. In this table, $p(A_1B_1) = .7$



A summary the independence model and a state-based model for AB is given below:

AB (p table)	q(A:B)	$q(A_1B_1)$
		
df = 3 (any three table values)	df = 2 (one A margin, one B margin)	df = 1 (A_1B_1)
	$T \neq 0$	$T = 0$

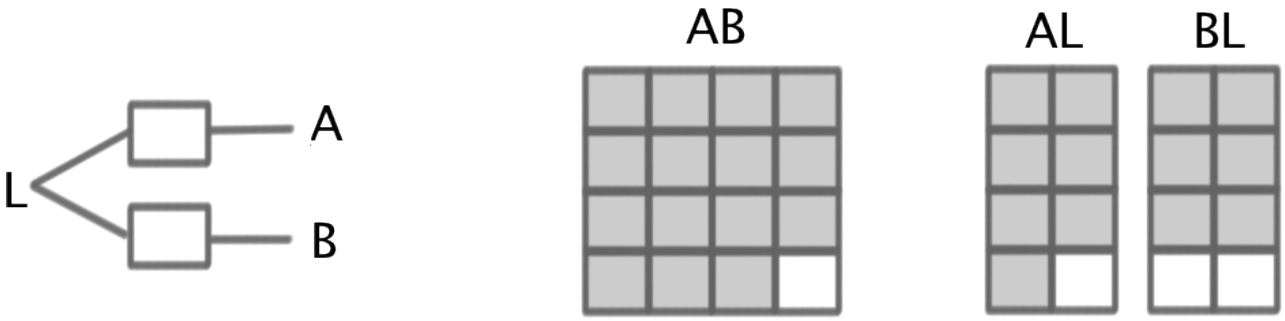
The state-based model, A_1B_1 has only one degree of freedom, because the only constraint is that $p(A_1B_1) = .7$. Entropy is maximized for the set of other probability values, i.e. probabilities or frequencies are uniformly distributed, so margins are irrelevant. Here, A_1B_1 is a simpler model than the independence model, but has no error.

Latent Variable Models

If you have data AB, find ABL such that AL:LB is a good model of ABL. This is a good idea if AL:LB is simpler, i.e., has smaller df, than AB. Latent class analysis is the nominal version of factor analysis

e.g. $|A| = |B| = 4$
$|L| = 2$

$df(AB) = 15$
$df(AL:LB) = (4 \cdot 2 - 1) + (4 \cdot 2 - 1) - (2 - 1) = 13$



9. Discussion: Complexity and Decomposability

In reconstructability analysis, complexity is the same as degrees of freedom. However there is more than one way to quantify complexity. For example, consider the equations:

$$z = ax + by$$

$$z = \left(\sqrt[\text{int}(a)]{\tanh(by)}\right)^{\text{int}(ax)!}$$

The second equation seems more complex, although each equation has the same number of variables. Function form could, in principal, enter into a complexity calculation.

Another complexity measures --: minimum description length –makes use of functional form in calculating complexity

vonBertalanffy's progressive segregation, systematization
(For 'complexify' in the diagram below, 'compose' might be a better word, since it's the opposite of decompose.)

10. Grouping Structure Types (R, C, P Structures)

The lattice of all possible structures can be broken up into ρ, C and P structures

ρ groupings are determined as follows. In ρ_1 all variables are directly connected to all other variables; that is, they are separated in the structure graph by only one box. In ρ_2, one pair of variable is not directly connected, i.e. those two variables are separated in the structure graph by 2 boxes.

C structures are the most complex of each ρ group. For example in ρ_1 group, the saturated model is the most complex, because the variables are the most interrelated.
P structures are the simplest in each ρ group. In the ρ_1 group for four variables, AB:AC:AD:BC:BD:CD is the simplest way for all variables share a relation with all other variables because this is the only ρ_1 structure with only dyadic relationships.

Search types:

Hierarchical search using ρ, C and P structures: First search representatives of ρ groups by searching among only C or P structures; then, for some given C or P structure, search within its ρ group

Beam search (what OCCAM does now): Find the best 'width' number of parent models, going up (or child models, going down); from these best models, then consider the best 'width' of their parents (or children), etc., as one goes up (or down) from level to level.
Could do a beam search 'breadth first' by having a large 'width' parameter going up (or down) hopefully only a modest number of levels, or 'depth first' by having a small 'width' parameter but going up (or down) many levels.

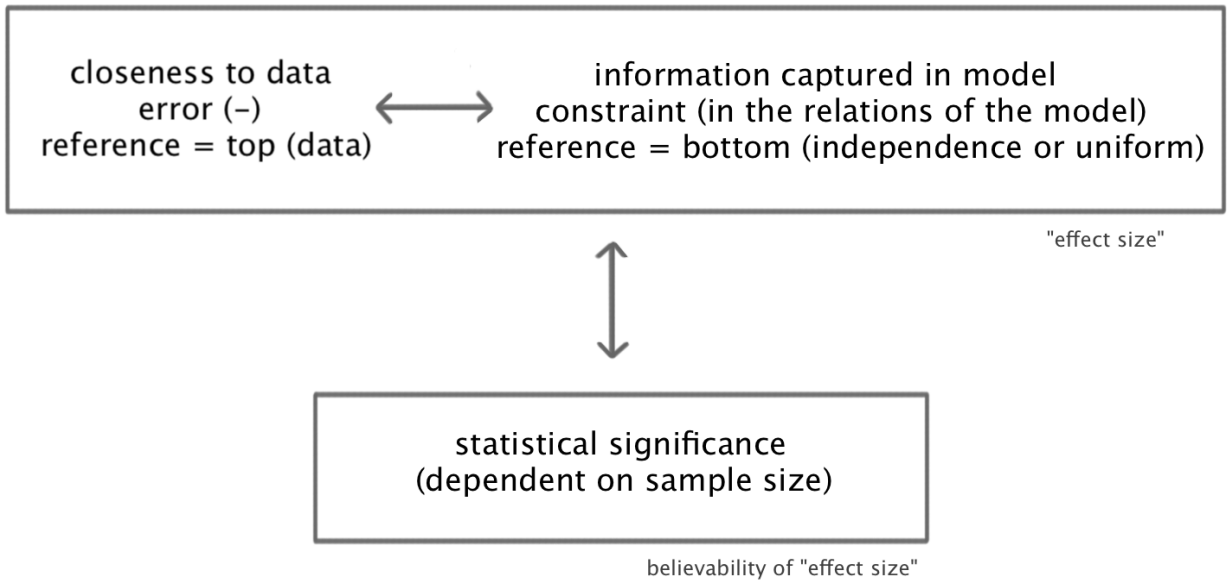
Information-Theoretic Reconstructability Analysis

(Putting it all together)

1. Preface: Goodness of a Model

We have talked about the goodness of a model being a trade-off between error (or its 'opposite,' information captured or effect size) and simplicity. We should also distinguish between effect size and statistical significance.

We then also have to consider statistical significance, which is the believability of these quantities, in other words, how certain we are that the effect size is not just due to chance. It is useful to examine both the effect size and its statistical significance. It is possible to have a small effect size that is highly significant, but that would not likely be much use to us.



2. Transmission and Information Distance

Reminder: the following are equivalent:

$$T(x,y,z) = H(x) + H(y) + H(z) - H(x,y,z)$$ variable notation

$$T(X:Y:Z) = H(X) + H(Y) + H(Z) - H(XYZ)$$ model notation

$$T(m_j) = H(m_j) - H(m_o)$$ Krippendorff notation where m_j = model, m_0 = data

The information distance between two models is the difference of their transmissions.

$$I(m_j \to m_k) = T(m_k) - T(m_j)$$
$$= H(m_k) - H(m_o) - \left(H(m_j) - H(m_o)\right)$$
$$= H(m_k) - H(m_j)$$
$$T(m_j) = \sum p(m_o) \log\left(\frac{p(m_o)}{q(m_j)}\right)$$

e.g. q(AB:BC) is calculated distribution for AB:BC
= p(AB)p(C | B)
= p(AB)p(BC)/p(B)

$$I(m_j \to m_k) = \sum p(m_o) \log\left(\frac{p(m_o)}{q(m_k)}\right) - \sum p(m_o) \log\left(\frac{p(m_o)}{q(m_j)}\right)$$
$$= \sum p(m_o) \log\left(\frac{q(m_j)}{q(m_k)}\right)$$

which is the weighted difference (weighted by the observed probabilities) of the difference between the logs of the two calculated probabilities

T and I are effect sizes, always positive, and I is more general than T

$$I(m_o \to m_j) = T(m_j) - T(m_o) = T(m_j)$$

$I(m_o \to m_j)$ is information lost in the model. (reference = top)
$I(m_j \to m_{ind})$ is information captured in a model. (reference = bottom)

We can only compare Transmission of models that are nested in the lattice of structure, i.e. they must be ancestors or descendents of each other.

OCCAM prints out information normalized by $T(m_{ind})$ so that information is between 0 and 1.

e.g. $\frac{I(m_j \to m_k)}{T(m_{ind})}$

where $\frac{I(m_o \rightarrow m_{ind})}{T(m_{ind})} = 1$

L^2 is likelihood ratio, a measure of statistical significance of the effect size, (Kripendorff p. 87)

$L^2 = 1.3863\ n\ I$ For n = sample size

Krippendorff p. 44-45 information is additive for chain models

A chain model has a general structure that looks like a chain, with pairs of variables.

$I(m_0 \rightarrow m_{chain})$ = error in the chain model
$I(m_{chain} \rightarrow m_{ind})$ = information captured in the chain model

$I(m_o \rightarrow m_{ind}) = I(m_0 \rightarrow m_{chain}) + I(m_{chain} \rightarrow m_{ind})$
$T(A:B:C:\ldots) = T(AB:BC:\ldots) + T(A:B) + T(B:C) + \ldots$

OCCAM will let you search only for chain models.

3 – 4. Calculating q and IPF; Maximizing H Subject to Constraint

Algebraic Calculations

$$T(m_j) = \sum p(m_o) \log\left(\frac{p(m_o)}{p(m_j)}\right)$$

There are 4 cases from simplest to most complex

1. $q(m_{ind})$: $q(A:B:C:\ldots) = p(A)\ p(B)\ p(C)\ldots$

2. disjoint; $q(AB:CDE:\ldots) = p(AB)\ p(CDE)\ldots$

3. overlap, no loops
   $q(AB:BC) = p(AB)\ p(C|B) = p(AB)\ p(BC) / p(B)$
   $q(AB:BC\ldots) = p(AB)\ p(C|B) = p(AB)\ p(BC) / p(B) \ldots$
   $q(AB:BC:CDE) = p(AB)\ p(C|B)\ p(DE|C)$

4. loops By IPF (no algebraic solutions)

OCCAM always does IPF, which converges in one iteration when there are no loops.
Iterative Proportional Fitting

Consider the probability table for data AB

AB p table **df = 3**

	B_1	B_2	
A_1	**.1**	**.2**	.3
A_2	**.3**	.4	.7
	.4	.6	

Calculating the q table for A:B, **df = 2**

	B_1	B_2	
A_1	q_1	q_2	**.3**
A_2	q_3	q_4	.7
	.4	**.6**	

	B_1	B_2	
A_1	.12	.18	**.3**
A_2	.28	.42	.7
	.4	**.6**	

This method maximizes entropy subject to the constraints the margins, i.e. we want to maximize $H(q) = -q_1\log q_1 - q_2\log q_2 - q_3\log q_3 - q_4\log q_4$ [or $\Gamma(q_1, q_2, q_3, q_4)$], such that

$q_1 + q_2 = .3$
$q_3 + q_4 = .7$
$q_1 + q_3 = .4$
$q_2 + q_4 = .6$
$q_1 + q_2 + q_3 + q_4 = 1$

The last constraint is assumed, and the second and fourth are redundant. Therefore there are two constraints, and df = 2

To satisfy the two constraints,

$q_1 = .12$
$q_2 = .18$
$q_3 = .28$
$q_4 = .42$

Constraints could also be written as matrix-vector equation. For three constraints, α, β, γ

α: $q_1 + q_2 = .3$
β: $q_2 + q_4 = .6$
γ: $q_1 + q_2 + q_3 + q_4 = 1$

The matrix, M, is given by the following, where df = rank of M (# of rows) - 1

$$\begin{array}{c} \\ \alpha \\ \beta \\ \gamma \end{array}\begin{array}{c} \begin{array}{cccc} q_1 & q_2 & q_3 & q_4 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

In IPF, entropy is maximized subject to the following equation:

$$M\check{q} = M\check{p}$$

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} .1 \\ .2 \\ .3 \\ .4 \end{bmatrix} = \begin{bmatrix} .3 \\ .6 \\ 1 \end{bmatrix}$$

To do IPF:
Start with uniform model.
Impose constraints one at a time.
If after posing all constraints, each constraint is still satisfied, IPF is done and the model does not contain loops. If some constraints are not satisfied, impose them again.

State-based model example where df = 2:

$q_1 + q_2 = .3$
$q_3 = .3$
$q_1 + q_2 + q_3 + q_4 = 1$

.1	.2	**.3**
.3	.4	.7
.4	.6	

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Topic 3: Calculating q and IPF

Given the formula for Transmission of the model m_j below, the challenge covered here is how to calculate $q(m_j)$.

$$T(m_j) = \sum p(m_0) \log\left(\frac{p(m_0)}{q(m_j)}\right)$$

There are 4 cases for covering how to calculate $q(m_j)$, and they are listed below, from the simplest case to the most complex.

1. The independence model: $q(m_{ind})$

$$q(A:B:C:\ldots) = p(A)p(B)p(C)\ldots$$

Ex 2: (showing p(AB) as a 2x2 matrix to expose the underlying linear algebra)

Given p(AB) = $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$

Then by projection p(A) = $\begin{bmatrix} a+b \\ c+d \end{bmatrix}$ and p(B) = $\begin{bmatrix} a+c & b+d \end{bmatrix}$

Then q(A:B) = p(A)p(B) = $\begin{bmatrix} a+b \\ c+d \end{bmatrix}\begin{bmatrix} a+c & b+d \end{bmatrix} = \begin{bmatrix} (a+b)(a+c) & (a+b)(b+d) \\ (c+d)(a+c) & (c+d)(b+d) \end{bmatrix}$

or q(AB) = $\begin{bmatrix} q_1 & q_2 \\ q_3 & q_4 \end{bmatrix}$

where:
$q_1 = p(A_1)p(B_1) = (a+b)(a+c)$
$q_2 = p(A_1)p(B_2) = (a+b)(b+d)$
$q_3 = p(A_2)p(B_1) = (c+d)(a+c)$
$q_4 = p(A_2)p(B_2) = (c+d)(b+d)$

Ex 2: Given p(ABC) =

	C_1		C_2	
	B_1	B_2	B_1	B_2
A_1	a	b	c	d
A_2	e	f	g	h

Because we are considering the independence model A:B:C, we project ABC down to the independent variables (skipping any intermediate projections such as AB or BC).

So by projection p(A) = $\begin{bmatrix} a+b+c+d \\ e+f+g+h \end{bmatrix}$

p(B) = $[a+e+c+g \quad b+f+d+h]$

p(C) = $[a+b+e+f \quad c+d+g+h]$

And q(A:B:C) = p(A)p(B)p(C)

=

	C_1		C_2	
	B_1	B_2	B_1	B_2
A_1	q_1	q_2	q_3	q_4
A_2	q_5	q_6	q_7	q_8

=

	C_1		C_2	
	B_1	B_2	B_1	B_2
A_1	(a+b+c+d) (a+e+c+g) (a+b+e+f)	(a+b+c+d) (b+f+d+h) (a+b+e+f)	(a+b+c+d) (a+e+c+g) (c+d+g+h)	(a+b+c+d) (b+f+d+h) (c+d+g+h)
A_2	(e+f+g+h) (a+e+c+g) (a+b+e+f)	(e+f+g+h) (b+f+d+h) (a+b+e+f)	(e+f+g+h) (a+e+c+g) (c+d+g+h)	(e+f+g+h) (b+f+d+h) (c+d+g+h)

or:

$q_1 = p(A_1)p(B_1)\ p(C_1)$

$q_2 = p(A_1)p(B_2)\ p(C_1)$

$q_3 = p(A_1)p(B_1)\ p(C_2)$

…

2. Model with no overlap: $q(AB:CD)$

$$q(AB:CD) = p(AB)p(CD)$$

Ex 2: Given p(ABCD) =

	D_1				D_2			
	C_1		C_2		C_1		C_1	
	B_1	B_2	B_1	B_2	B_1	B_2	B_1	B_2
A_1	a	b	c	d	e	f	g	h
A_2	i	j	k	l	m	n	o	p

Here, we only project to the level of the relationships AB and CD to get q(AB:CD)

P(AB) =

	B_1	B_2
A_1	a+c+e+g	b+d+f+h
A_2	i+k+m+o	j+l+n+p

P(CD) =

	D_1	D_2
C_1	a+b+i+j	e+f+m+n
C_2	c+d+k+l	g+h+o+p

q(AB:CD) =

	D_1				D_2			
	C_1		C_2		C_1		C_1	
	B_1	B_2	B_1	B_2	B_1	B_2	B_1	B_2
A_1	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8
A_2	q_9	q_{10}	q_{11}	q_{12}	q_{13}	q_{14}	q_{15}	q_{16}

=

	D_1				D_2			
	C_1		C_2		C_1		C_2	
	B_1	B_2	B_1	B_2	B_1	B_2	B_1	B_2
A_1	(a+c+e+g) (a+b+i+j)	(b+d+f+h) (a+b+i+j)	(a+c+e+g) (c+d+k+l)	(b+d+f+h) (c+d+k+l)	(a+c+e+g) (e+f+m+n)	(b+d+f+h) (e+f+m+n)	(a+c+e+g) (g+h+o+p)	(b+d+f+h) (g+h+o+p)
A_2	(i+k+m+o) (a+b+i+j)	(j+l+n+p) (a+b+i+j)	(i+k+m+o) (c+d+k+l)	(j+l+n+p) (c+d+k+l)	(i+k+m+o) (e+f+m+n)	(j+l+n+p) (e+f+m+n)	(i+k+m+o) (g+h+o+p)	(j+l+n+p) (g+h+o+p)

3. Model with overlap, but no loops: $q(AB:BC)$

Ex 1:

$$q(AB:BC) = p(AB)p(C|B) = \frac{p(AB)p(BC)}{p(B)}$$

Given p(ABC) =

	C_1		C_2	
	B_1	B_2	B_1	B_2
A_1	a	b	c	d
A_2	e	f	g	h

Then by projection:

P(AB) =

	B_1	B_2
A_1	a+c	b+d
A_2	e+g	f+h

P(BC) =

	C_1	C_2
B_1	a+e	c+g
B_2	b+f	d+h

P(B) =

B_1	a+c+e+g
B_2	b+d+f+h

and q(AB:BC) =

		C_1		C_2	
		B_1	B_2	B_1	B_2
A_1		$\frac{(a+c)(a+e)}{(a+c+e+g)}$	$\frac{(b+d)(b+f)}{(b+d+f+h)}$	$\frac{(a+c)(c+g)}{(a+c+e+g)}$	$\frac{(b+d)(d+h)}{(b+d+f+h)}$
A_2		$\frac{(e+g)(a+e)}{(a+c+e+g)}$	$\frac{(f+h)(b+f)}{(b+d+f+h)}$	$\frac{(e+g)(c+g)}{(a+c+e+g)}$	$\frac{(f+h)(d+h)}{(b+d+f+h)}$

Ex 2: $q(\boldsymbol{AB:BC:CDE}) = p(AB)p(C|B)p(DE|C)$

$$= p(AB) * \frac{p(BC)}{p(B)} * \frac{p(CDE)}{p(C)}$$

$$= \frac{p(AB)p(BC)p(CDE)}{p(B)p(C)}$$

Ex 3: $q(\boldsymbol{AB:BC:CDE:DEF}) = p(AB)p(C|B)p(DE|C)p(F|DE)$

$$= p(AB) * \frac{p(BC)}{p(B)} * \frac{p(CDE)}{p(C)} * \frac{p(DEF)}{p(DE)}$$

$$= \frac{p(AB)p(BC)p(CDE)p(DEF)}{p(B)p(C)p(DE)}$$

Ex 4: $q(\boldsymbol{ABC:BCD:CDE}) = p(ABC)p(D|BC)p(E|CD)$

Here there are overlapping relations where the overlap portions also overlap each other, so continue a nested dividing by the residual overlap, which in effect gives you an alternating multiply-divide-multiply. (*Dr. Zwick: If the overlaps had overlaps, one would have to multiply them, i.e., just like the Krip. method of alternating signs, one would have here alternating multiplication and division*)

$$= \frac{p(ABC)p(BCD)p(CDE)}{\frac{p(BC)p(CD)}{p(C)}} = \frac{p(ABC)p(BCD)p(CDE)p(C)}{p(BC)p(CD)}$$

5. Choosing Models Statistically

Let's start out with some basic definitions…

- **Type I error:** This is when I reject a null hypothesis and I shouldn't have.
  - Let's say two things are not different in reality (e.g., typing speed for men vs. women), but they *happen* to look different in my sample, just by chance, because of who I happened to sample. If I reject the null hypothesis, and say that there *is* a difference in typing speed between genders, I have done so incorrectly. This is a Type I error.
- **Type II error:** This is when I fail to reject a null hypothesis and should have.
  - Let's say two things really *are* different in reality (e.g., height for men vs. women), but it just so happens they *don't* look very different in my sample, just by chance, because of who I happened to sample. If I don't reject the null, and say "we didn't find evidence of a height difference between genders," this is a Type II error.
- **P-Value, or α:** This is the probability of making a Type I error.
  - Usually you want this to be small, because you don't want to go spouting off "I found a significant difference!" when it was just due to chance variations in your sample. You want to be confident that there's only a very small likelihood that this difference could have been caused by chance variations. When $p < .05$, it means there's less than a 5% chance that the difference you observed was due to chance alone.

OK, now let's talk about some DMM definitions…

- **"Good" Model:** Qualitatively speaking, a model is good if it captures a lot of the information in your data. Technically speaking, a model is good when its probability distribution (q) is really similar to the probability distribution (p) in your original dataset.
  - Let's say that you can exactly reproduce the values in the observed (p) probability distribution for AB by just knowing the probabilities of A being A_1 or A_2 (50/50 split) and probabilities of B being B_1 or B_2 (25/75 split). If a calculated distribution (q), using only those numbers is exactly the same as the probability distribution you observed, then the model of A:B is perfect. It captures all the information present in your data.

p

	B_1	B_2	
A_1	.125	.375	.5
A_2	.125	.375	.5
	.25	.75	

q

	B_1	B_2	
A_1	.125	.375	.5
A_2	.125	.375	.5
	.25	.75	

- **"Good Enough" Model:** You can measure how good a model is by calculating T (i.e., error) or I (information distance), but that doesn't tell you whether or not the amount of error (or information captured) is significant. It doesn't tell you whether your model is "good enough," statistically speaking. Instead we look for the model that is better than all the others.
- **"Better" and "Worse" Model:** When we use statistical approaches to determine which models are better and worse, we want to know two things:
  - **Significance**, or is this model *significantly* different? (or is the difference pretty likely to be due to chance), and

- **Relative to What?** What are we comparing this model to, anyway?
  This will depend on whether you are using the independence model (bottom) as your reference, or the data (top) as your reference…

Now we are ready to start building ideas on top of the definitions. Here are some rules of thumb…

When the Independence Model is your Reference, Test if Models are Significantly Better

Models may or may not be significantly better (at replicating the p distribution) when the independence model is your reference. Here you generally start at the bottom and work upward.

Why go from the Bottom Up?

- Sometimes we want to see how complex of a model is justified by our data. I have a bunch of variables measured, which I suspect are related. I want to know,
  - Which associations actually exist among these variables in the real world?
  - Are there simply 2-way associations among these variables?
  - Or more complex relations, such as 3-way and higher-way?
  - How confident am I that this complex model is significantly better than a simpler model?

When the Data is your Reference, Test if Models are Significantly Worse

Models may or may not be significantly worse (at replicating the p distribution) when the data is your reference. Here you generally start at the top and work downward.

Why go from the Top Down?

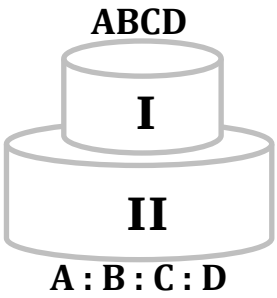
- Sometimes we want to see how simple of a model will still decently capture the important patterns we observed in our data. I might want to know,
  - Do we really need a 4-way relation, ABCD, to capture the probability distribution we observed?
  - Would information about the nature of four 3-way relations, ABC:ABD:ACD:BCD, capture the observed patterns just as well?
  - How confident am I that this simpler model is not significantly worse than a more complex model?

Using Cake to Understand Type I and Type II Errors

OK, look at the picture below. Imagine a 'Type I Error Zone' at the top of the lattice of structures, and a 'Type II Error Zone' at the bottom of the lattice of structures. (This is just symbolic, to help you remember.) The 'I' in "Type I" is smaller than the 'II' in "Type II," so you might imagine stacking the 'I' on top of the 'II' to keep it straight. (And who doesn't love cake?)

OK, so let's say the data is my reference. I'm starting from the top and working my way downward. I am going to see how far down I can go (how simple of a model I can get), but I want to make sure to stop before I get into the *Type II Error Zone*. (Cue the scary music.)

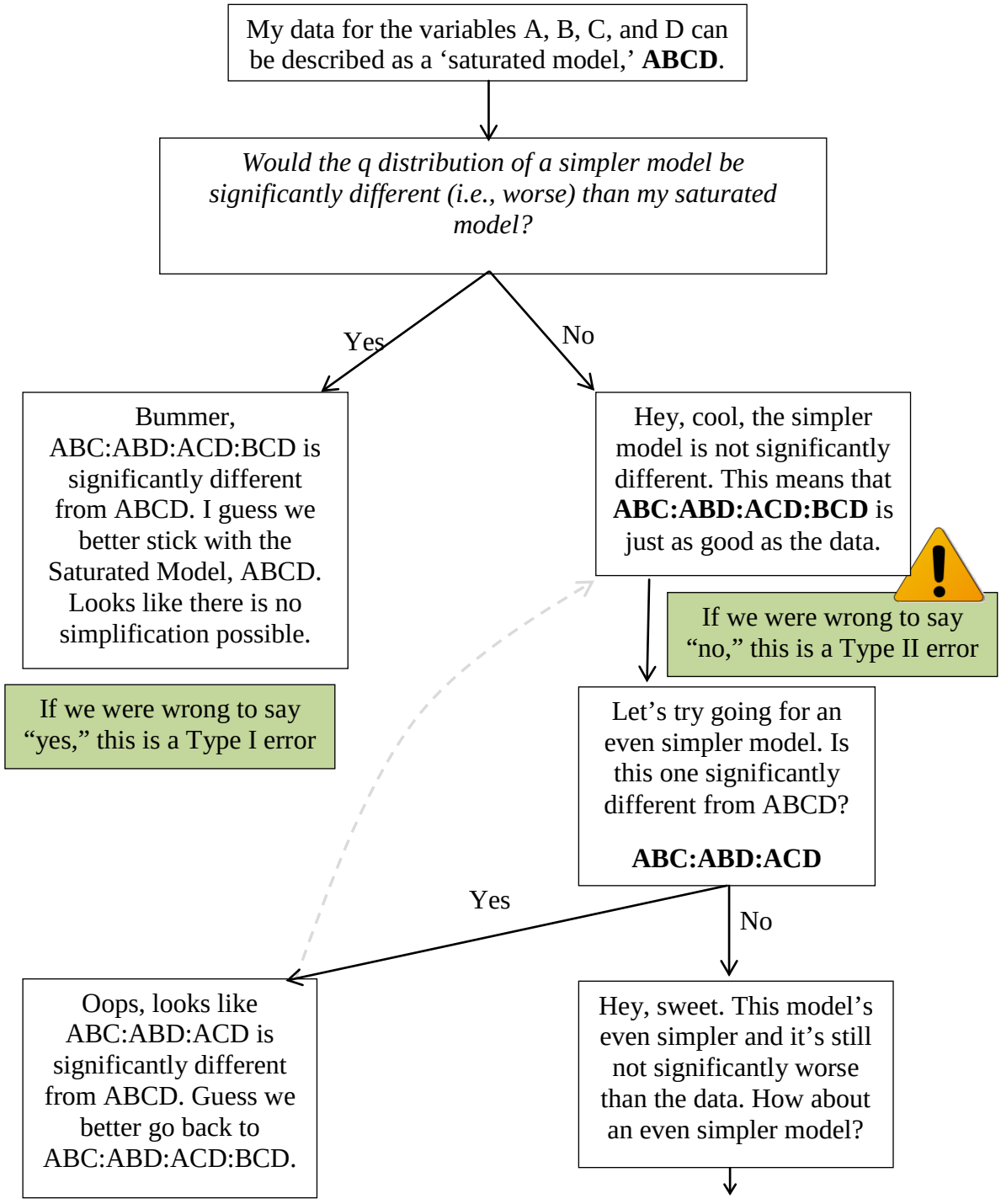
Alternatively, let's say the independence model is my reference. I'm starting from the bottom and working my way upward. I am going to see how far up I can go (seeing how complex of a model I can justify), but I want to make sure to stop before I get into the *Type I Error Zone*. (Again, cue scary music.)



The moral of the story is this: Whichever way you're going, you want to go as far as you can, but not too far. Going too far is like overstating findings that are not warranted. It's worse to overstate your findings than to understate. When the bottom is your reference, don't go too far up (you'll get a Type I error, and be over fitting).
When the top is your reference, don't go too far down (you'll get a Type II error, and be over simplifying).

A Hypothetical Example when the Reference is the Top

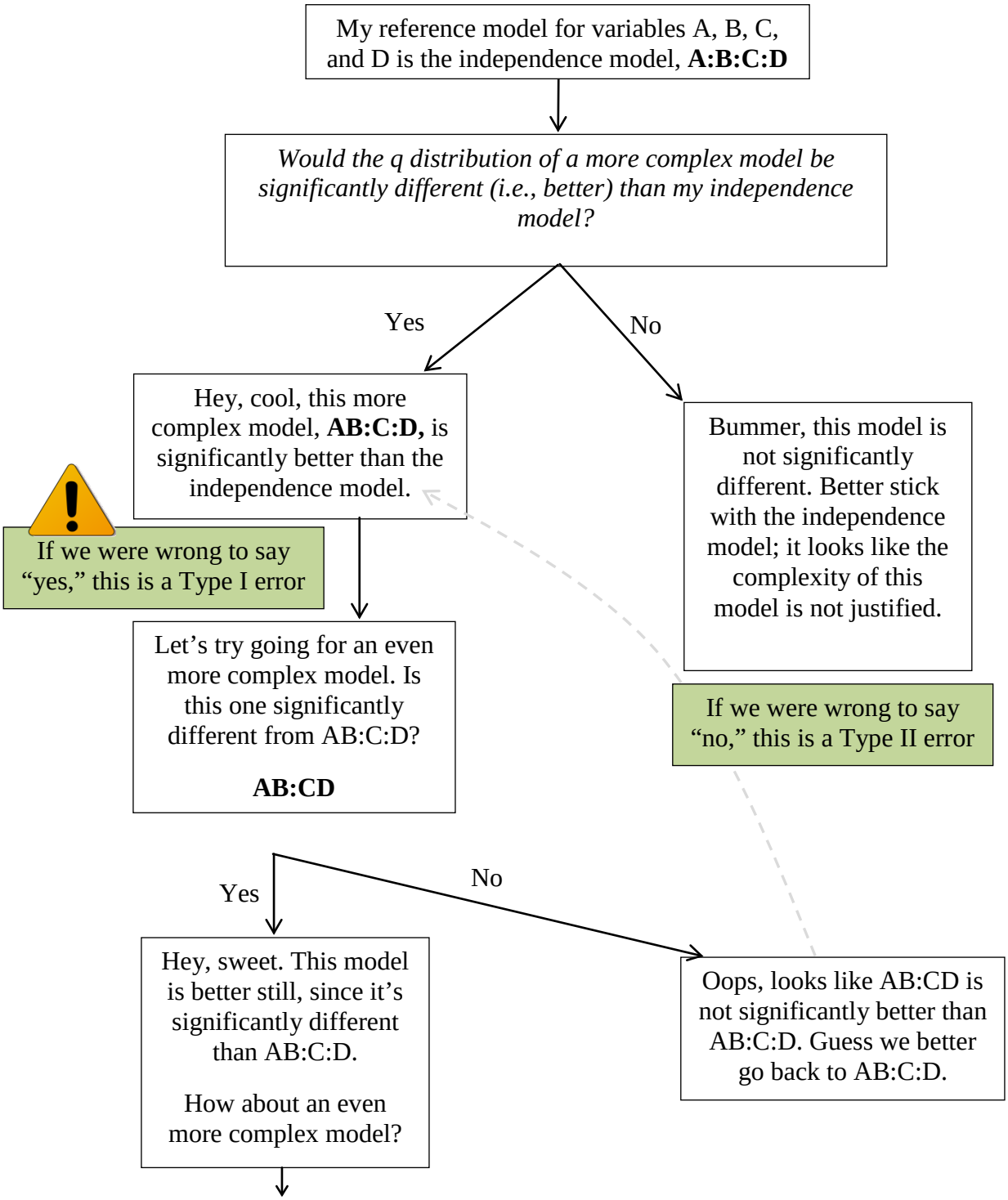
Here's a more fine-grained look at what Type I and Type II errors mean for evaluating models when the reference is the top. Usually when the reference is the top, you work from the top down.



**Note that here, if our Type I error rate is really small ($p < .05$), we have to be really confident a model is significantly worse before we'll stop going down. We'll probably have Type II errors, which are troublesome: We may be over confident that a simpler model is ''just as good.'*

A Hypothetical Example when the Reference is the Bottom

Here's a more fine-grained look at what Type I and Type II errors mean for evaluating models when the reference is the bottom. Usually when the reference is the bottom, you work from the bottom up.



Note that here, the Type I error rate is more intuitive, because we do want to be really confident that a model is significantly better before we keep going up. A small p value, such as $p < .05$, will keep us from being over confident that a complex model is justified.

Overall Patterns

Note that regardless of your reference, rejection of the null always results in an upward focus.

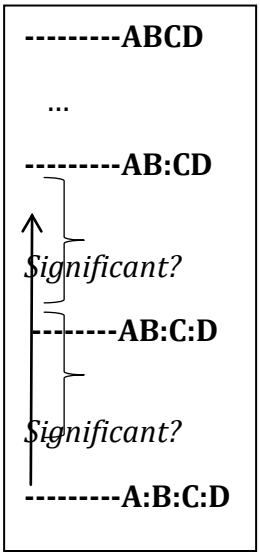
- If your reference is the top, rejecting the null means you will go back up to the previous level.
- If your reference is the bottom, rejecting the null means you will at least stay there, and maybe even try to move up another level.

Also note that regardless of your reference, failure to reject the null results in a downward focus.

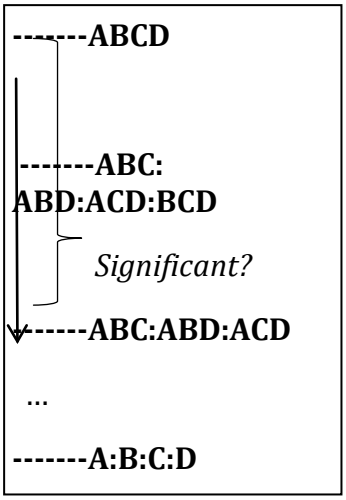
- If your reference is the top, failing to reject the null means that you will at least stay there, and maybe even try to move down another level.
- If your reference is the bottom, failing to reject the null means you will go back down to the previous level.

So remember: Rejection is upward (think of flipping the bird?), and non-rejection is downward.

Incremental Alpha, but not Beta



When you are going up the lattice, each additional model ought to be significantly different (i.e., significantly better) than the model below it. That is, if I go up from A:B:C:D to AB:C:D, and want to go up even further to AB:CD, I need to make sure that AB:CD is significantly better than AB:C:D (not only better than A:B:C:D). Why? Well, think of it this way: If the difference between A:B:C:D and AB:C:D is significant, then that significant difference will also be present in your test of whether A:B:C:D and AB:CD are significantly different. Finding a significant difference between A:B:C:D and AB:CD will be influenced (or "contaminated") by the significant difference between AB:C:D and A:B:C:D. Testing incrementally helps to "purify" your tests of significance, so you can be sure that each step up the lattice is incrementally significant (not just cumulatively significant). It helps protect you from passing into the *Type I Error Zone*.



When going down the lattice, things are a bit different. In this case you actually want to compare each model with the data (rather than comparing it with the model directly above). The reason is this: We are more worried about Type II errors here, and they are less likely if we compare models that are further away from each other. Imagine you are climbing onto your roof. The step ladder is not significantly far from the ground, and your roof is not significantly higher from the

stepladder. But falling off the roof onto the ground will be significant. In the same way, if I find that ABC:ABD:ACD:BCD is not significantly worse than ABCD, and that ABC:ABD:ACD is not significantly worse than ABC:ABD:ACD:BCD, it could still be the case that this lower model, ABC:ABD:ACD, *is* significantly worse than my data. I want to make sure I reject the null in this case, so that I won't head into the *Type II Error Zone*.