III. INFORMATION-THEORETIC RECONSTRUCTION

(Putting I Basics + II Structures together)

Exercises

Midterm 2018: 2, 4, 5
Final 2018: 1ce, 2ab, 3abcd
Final 2019: 2c, 3abde, 4ab
Midterm 2021: 4
Final 2021: 2

1. Preface: information in/error of models

The task: given data, find the **simplest model which satisfactorily fits the data.**
"Satisfactorily fits" = **information high** or **error low** enough, relative to **complexity**
Implied **Reference = data** (top)

Or: find the **most complex model whose** posited **relations are justifiable**
Want information high enough relative to complexity.
Implied **Reference = independence** (bottom)

We assess information/error with **Transmission** or **Information Distance**
$T(m_i)$, transmission of model m_i: the distance of m_i to the data (m_0), the error, &
$\mathbf{I(m_i \rightarrow m_j) = T(m_j) - T(m_i)}$, information distance, a difference between transmissions of two models, one of which is a descendant of the other.

When $m_i = m_0$, the data, and m_j is some model, $\mathbf{I(m_i \rightarrow m_j)}$ is $T(\mathbf{m_j})$ = **error** of model.
When m_i is model & m_j is independence, it is **information** (captured) in model.
Note the difference in arrow convention for Δdf.
$\mathbf{\Delta df(m_i \rightarrow m_j) = df(m_i) – df(m_j)}$

Arrows in both cases **always** go from **higher to lower** models.

Note that $\mathbf{I(m_0 \rightarrow m_j) = T(\ m_j) – T(m_0\ ) = T(m_j)}$, since $T(m_0\ ) = 0$
With the p log p/q expression for transmission, the transmission of the data is
$\mathbf{T(m_0) = \sum p(m_0) \log [\ p(m_0)/p(m_0)\ ] = \sum p(m_0) \log 1 = 0}$
so **I is general**, and encompasses T. **I, T Δdf** are always **positive**.

I and **T** measures evaluate how **good** model is in terms of information or error.
Adding N & Δdf lets us say if this "goodness" is **believable**, i.e., reliable statistically.

Information & error measured by T are **non-statistical** measures, dependent only on probabilities, but not on the sample size or degrees of freedom.

The believability of the information/error is a **significance** question which depends on sample size and degrees of freedom.

2. Transmission & information distance

Transmission for independence model

With respect to T, consider the definition we've previously used:

T(X:Y:Z) = H(x) + H(y) + H(z) **–** H(x,y,z)

= H(X:Y:Z) – H(XYZ)

T(X:Y:Z) = $\sum_i \sum_j \sum_k$ p(x_i, y_j, z_k) log [p(x_i, y_j, z_k) / $q_{X:Y:Z}$(x_i, y_j, z_k)]

Where $q_{X:Y:Z}$(x,y,z) = p(x)*p(y)*p(z) = expected (**calculated**, **not observed**) probabilities UNDER ASSUMPTION (hypothesis) OF INDEPENDENCE.

= ∑ p(XYZ) log [p(XYZ) / q(X:Y:Z)]

For convenience, will only show one ∑, intended to be over all cells.

In log-linear notation, independence model is usually written {X}{Y}{Z}.

The full data, also called the saturated model, is written as {XYZ}.

In some papers, instead of a colon, there is a slash: X/Y/Z.

To generalize, where q = q(model),

T(q) = ∑ p log [p / q] **= ∑ p log p - ∑ p log q**

= – H(p) - ∑ p log q

Non-obvious lemma: ∑ p log q = ∑ q log q)

= – H(p) + H(q)

In Krippendorff notation (p.44),

$T(m_j) = \sum p(m_0) \log [p(m_0) / \rho(m_j)] = H(m_j) – H(m_0)$

LEAVE FOR LATER (a) how generate $q(m_j)$, and (b) statistical significance issue.

Krippendorff says (8.3, p.44) $T(m_j) \neq H(m_j) – H(m_0)$ for models with loops.

<u>Not correct</u>; T is difference of entropies always: for models without or with loops.

What is true is that <u>one can't algebraically simplify H(q)</u> for models with loops.

Information distance

Information distance is defined as a difference between transmissions; this is useful as a kind of bookkeeping convenience, to be able to compare any two models.

$I(m_i \rightarrow m_j)$ = amount of information modeled in (captured by) m_i, lost in m_j

Arrow always goes towards lower models

$$\begin{aligned} &= \mathbf{T(m_j)} && \mathbf{- T(m_i)} \\ &= [\, H(m_j) - H(m_0) \,] && - [\, H(m_i) - H(m_0) \,] \\ &= \mathbf{H(m_j)} && \mathbf{- H(m_i)} \end{aligned}$$

If we start from data, go to a model, and then to independence model, we have

$\mathbf{I(m_0 \rightarrow m_j)}$ = amount of information **lost** in m_j.

$= T(m_j) - T(m_0)$

$\mathbf{= T(m_j)}$

$\mathbf{I(m_j \rightarrow m_{ind})}$ = amount of information **modeled** (captured) in m_j.

$\mathbf{= T(m_{ind}) - T(m_j)}$

Accounting is thus convenient, using Gokhale & Kullback (K, p.44) partitioning identity:

$\mathbf{I(m_0 \rightarrow m_{ind}) =}$	$\mathbf{I(m_0 \rightarrow m_j) +}$	$\mathbf{I(m_j \rightarrow m_{ind})}$
Information in data	**Information lost (error) in m_j**	**Information modeled (captured) in m_j**

WE WANT A MODEL WHERE LITTLE IS LOST FROM THE DATA, I.E., WHERE MUCH IS MODELED/PRESERVED/CAPTURED/RETAINED IN THE MODEL.

Could normalize information captured by the maximum that could be captured:

$$\begin{aligned} \textbf{Normalized information captured} &= \mathbf{I(m_j \rightarrow m_{ind}) / I(m_0 \rightarrow m_{ind})} \\ &= \mathbf{I(m_j \rightarrow m_{ind}) / T(m_{ind})} \end{aligned}$$

(This is what Occam outputs)

Information distance doesn't really add any new concept, but it introduces a slightly more complicated form of the p*log[p/q] expression (specifically, $p*\log[q_1/q_2]$), as above:

$$\begin{aligned} \mathbf{I(m_i \rightarrow m_j)} &= T(m_j) - T(m_i), \\ &= \sum p(m_0) \log [\, p(m_0) / q(m_j) \,] - \sum p(m_0) \log [\, p(m_0) / q(m_i) \,] \\ &= \sum p(m_0) \log [\, p(m_0) / q(m_j) * q(m_i) / p(m_0) \,] \\ &= \mathbf{\sum p(m_0) \log [\, q(m_i) / q(m_j) \,]} \end{aligned}$$

An example: model = AB:BC:CD:...:YZ, chain model.

$I(m_0 \rightarrow m_{ind})$ =	$I(m_0 \rightarrow m_{chain})$	+ $I(m_{chain} \rightarrow m_{ind})$
information in data	information lost in chain	information modeled in chain model

$I(m_0 \rightarrow m_{ind})$	$= T(m_{ind}) - T(m0)$ $= T(A:B:C:...:Z)$	$= T(mind)$ = information **total**
$I(m_0 \rightarrow m_{chain})$	$= T(m_{chain}) - T(m_0)$ $= T(AB:BC:CD:...:YZ)$	$= T(m_{chain})$ = information **lost**
$I(m_{chain} \rightarrow m_{ind})$	$= T(m_{ind}) - T(m_{chain})$ $= T(A:B) + T(B:C) + ... + T(Y:Z)$	= information modeled

Second example: K, p.46.
m_1 = ABCD:CDEF
m_2 = AC:BC:CD:DE:DF

show K structures

$I(m_0 \rightarrow m_{ind})$	$= I(m_0 \rightarrow m_1)$	$+ I(m_1 \rightarrow m_2)$	$+I(m_2 \rightarrow m_{ind})$
1.8301	= 0.4011	+ 1.3843	+ 0.0446
	$= I(m_0 \rightarrow m_1)$	$+ I(m_1 \rightarrow m_{ind})$	
	= 0.4011 (22%)	+ 1.4289 (78%)	
	= lost in m_1	modeled in m_1	
	$= I(m_0 \rightarrow m_2)$		$+ I(m_2 \rightarrow m_{ind})$
	= 1.7854 (98%)		0.0446 (2%)
	= lost in m_2		modeled in m_2

Simpler model (m_2) is **inadequate**.

More complex model (m_1) captures most of data and **may** be acceptable.

3. Calculating q algebraically

1. Maximum uncertainty q's

Idea of **maximum entropy:** maximizing *soaks up any extra degrees of freedom* not specified by constraints of model.

Maximize $H(q) = -\sum q \log q$ subject to constraints of model. e.g., model = AB:CD has (for dichotomous variables) df = 3 + 3 = 6 constraint equations, while df(ABCD) = 15.

Maximization soaks up extra degrees of freedom.

q(ABCD) is a 4-way distribution, just like p(ABCD), so one might think that it has the same df as p(ABCD), i.e., df=15, but it really has only 6 for this model. Uncertainty maximization fills out the specification of the distribution.

Will not prove that methods below which describe how q is actually calculated achieve this maximum uncertainty result.

2. How q's actually calculated

Four cases from simplest to most complex

(1) Independence of all variables (m_{ind})

(2) No overlap of variables in components, i.e., disjoint model (neutral systems)

(3) Overlap but no loops

(4) Overlap and loops (and structural zeros): Iterative Proportional Fitting

3. Algebraic calculation of q's (cases 1-3)

Simplest case (1): for $q(m(_{ind}))$, is product of marginals. For example, q(A:B:C) = p(A)p(B)p(C).This gives maximum entropy.

Slightly more complex case (2): variables not shared by components: AB:CD. Then q is 'essentially' the same as m_{ind}. q(AB:CD) = p(AB)p(CD)

Next most complex case (3): variables shared, but no loops. See K, p.52.

q(XY:YZ)= p(XY) p(YZ)/ p(Y) = p(x,y) p(z|y)

How get it? Consider p(XY) p(YZ). It's *dimensionally* wrong. p(Y) appears twice. So:

q(XY:YZ) = p(x,y) [p(y,z) / p(y)] = p(x,y) p(z|y) = p(y,x) [p(x,y)/p(y)] = p(y,z) p(x|y)

More complete notation: $q_{XY:YZ}(x_i, y_j, z_k) = p(x_i, y_j)\, p(z_k \mid y_j) = p(y_j, z_k)\, p(x_i \mid y_j)$.

Go on to equations on K, p.55. In general, multiply probabilities of all relations, divide by probabilities of pair overlaps, multiply by triplet overlaps, etc.

Show an XYZ table, with parameters a…h, and calculate $q(x_1,y_2,z_2)$ in terms of parameters, for model XY:YZ.

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

	z_1		z_2	
	y_1	y_2	y_1	y_2
x_1	q_1	q_2	q_3	q_4
x_2	q_5	q_6	q_7	q_8

q(XY:YZ) = p(XY) p(YZ)/ p(Y) model notation

$q_{XY:YZ}(x,y,z)$ = p(x,y) p(y,z) / p(y) variable notation

Specifically, for some particular values of x,y,z (an example):

$q_{XY:YZ}(x_1,y_2,z_2)$ = $p(x_1,y_2)$ $p(y_2,z_2)$ / $p(y_2)$

= (b+f) (f+h) / (b+d+f+h)

4. Calculating q maximizes entropy subject to constraints

Discuss how calculating q's for **X:Y** maximizes entropy subject to constraints.

Do linear algebra, showing matrix-vector equation, where |X|=|Y|=2, so |X||Y|=4, with data as $p_1 \ldots p_{|X||Y|}$ vector & model probabilities as $q_1 \ldots q_{|X||Y|}$ vector.

Only need df(X:Y) = 2 rows filled in for matrices, as long as independent

q	y_1	y_2	
x_1	q_1	q_2	q_1+q_2
x_2	q_3	q_4	
	q_2+q_4		

p	y_1	y_2	
x_1	a	b	a+b
x_2	c	d	
	b+d		

$$\begin{bmatrix} 1 & 1 & & \\ & & & \\ & 1 & & 1 \\ & & & \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & & \\ & & & \\ & 1 & & 1 \\ & & & \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

Note that the number of constraint, not counting the last row that imposes the sum of probabilities to be 1, is df(X:Y). This is also the "rank" of the above matrix.

Imposing the constraints of the margins defined by the model is an **underdetermined** problem. Have 2 equations here to get 3 numbers. **One gets a unique solution by maximizing $-\sum q \log q$ subject to these constraints**. This maximization "soaks up" the extra degrees of freedom.

(Reversing left and right tables,) now consider matrix-vector equation also for XY:YZ

df(XY:YZ) = df(XY) + df(YZ) – df(Y) = 3 + 3 – 1 = 5

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

	z_1		z_2	
	y_1	y_2	y_1	y_2
x_1	q_1	q_2	q_3	q_4
x_2	q_5	q_6	q_7	q_8

XY constraints:

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

	y_1	y_2
x_1	q_1+q_3	q_2+q_4
x_2	q_5+q_7	q_6+q_8

$q_1 + q_3 = a + c$

$q_2 + q_4 = b + d$

$q_5 + q_7 = e + g$

$q_6 + q_8 = f + h$ Don't need since probabilities add up to 1

YZ constraints:

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

	z_1	z_2
y_1	q_1+q_5	q_3+q_7
y_2	q_2+q_6	q_4+q_8

$q_1 + q_5 = a + e$

$q_2 + q_6 = b + f$ Don't need since know that $q_2 + q_4 + q_6 + q_8 = q(y_2) = b + d + f + h$

$q_3 + q_7 = c + g$ Don't need since know that $q_1 + q_3 + q_5 + q_7 = q(y_1) = a + c + e + g$

$q_4 + q_8 = d + h$

q1	q2	q3	q4	q5	q6	q7	q8
1		**1**					
	1		**1**				
				1		**1**	
1				**1**			
			1				**1**
1	1	1	1	1	1	1	1

q_1		a
q_2		b
q_3		c
q_4	= M	d
q_5		e
q_6		f
q_7		g
q_8		h

5. Calculating q with IPF

General description of IPF: Deming-Stephan algorithm (K, p.58-9)

Simplest example of IPF to get q(X:Y)

Original XY distribution with its X and Y marginals

p(XY)	Y_1	Y_2	
X_1	.4	.3	.7
X_2	.2	.1	.3
	.6	.4	

q(X:Y)	Y_1	Y_2	
X_1	.42	.28	.7
X_2	.18	.12	.3
	.6	.4	

1. Want to find q(X:Y) by IPF. Start with uniform distribution

	Y_1	Y_2	
X_1	.25	.25	.5
X_2	.25	.25	.5
	.5	.5	

2. **Impose X margin, which are {.7, .3}**: fit to X by dividing each row by calculated margin, then multiply by correct margins:

	Y_1	Y_2	
X_1	.25 ***.7**/.5 = .35	.25 ***.7**/.5 = .35	.7
X_2	.25 ***.3**/.5 = .15	.25 ***.3**/.5 = .15	.3
	.5	.5	

It now agrees with X margins.

3. **Impose Y margin of {.6, .4}**: fit to Y by dividing each column by calculated margins, then multiply by correct margins:

	Y_1	Y_2	
X_1	.35 *.6/.5 = .42	.35 *.4/.5 = .28	.7
X_2	.15 *.6/.5 = .18	.15 *.4/.5 = .12	.3
	.6	.4	

It now agrees with Y marginals.

It agrees now with **both** margins, and thus with q(X:Y), so we're done.

In a model with loops, e.g., XY:YZ:XZ, we'd impose XY, then impose YZ, then impose XZ, **which would mess up the XY agreement**, so we'd repeat this until it converged, which it is guaranteed to do.

IPF for state-based models.

Structural constants: K, p.48: Structural zeros (or structural constants that are not zero) complicate: (1) maximum entropy calculation of q's; (2) calculation of df.

For (1), could add it as another IPF step, OR if it is in data table (but not projections), just remove it from list of q's to be adjusted and fix it in the q list.

For (2), just subtract the number of these known values from df.

Original XY distribution

p(XY)	Y_1	Y_2	
X_1	.4	.3	.7
X_2	.2	.1	.3
	.6	.4	

$q(X_2Y_2)$	Y_1	Y_2	
X_1	.3	.3	.6
X_2	.3	.1	.4
	.6	.4	

1. Want to find $q(X_2Y_2)$ by IPF. Start with uniform distribution

	Y_1	Y_2	
X_1	.25	.25	.5
X_2	.25	**.25**	.5
	.5	.5	

2. **Impose X_2Y_2 by dividing by current calculated value and multiplying by known value that model specifies, where [1-p(X_2Y_2)] / 3 = .3 is specified by model for the 3 remaining cells**

	Y_1	Y_2	
X_1	.25 ***.3**/**.25** = .3	.25 ***.3**/**.25** = .3	.6
X_2	.25 ***.3**/**.25** = .3	.25 ***.1**/**.25** = .1	.4
	.6	.4	

We're done.

6. Choosing models based on information content

Two criteria for good model: (1) high information (low error), (2) low complexity.

When one has two criteria in an optimization problem, one can:

(a) Maximize / minimize one criterion subject to the other as a constraint

(a.1) Minimize complexity subject to information constraint:

Find simplest model that has information greater than some %

This is **non-statistical**. This is this **Topic 6**.

(a.2) Find best model by information/error subject to p-value constraint

This is **statistical**. **This is Topic 8**.

(b) Maximize some weighted sum of the two criteria

Using BIC or AIC to weight information/error and complexity.

This is Topic 7.

Thinking ahead to (a.2) to justify (a.1):

Information/error is assessed by some information distance.

This gets **multiplied by 1.3863 N** (Krip, p.87) to get **likelihood ratio Chi-square, L^2**.

L^2 = LR

From L^2 one assesses statistical significance (a.2).

When **N** is very **large,** L^2 will be large.

Then, if one goes down lattice, immediately we get a **rejection of the null hypothesis**, i.e., immediately we will find that our **model differs significantly from the data**, i.e., has differences with the data not attributable to chance.

We **couldn't accept any simplifying model** if we insist that it not differ from the data.

So we can adopt a different perspective: **we accept a model which accounts for some specified minimum % of information in the data**.

For **large samples** and for **reference=top** searches, % information captured is only possible criterion if simplified models are to be considered.

7. Choosing models with AIC & BIC (criterion (b))

Models are selected from the one of the measures that OCCAM outputs for different models applied to the training set data, namely the **Bayesian Information Criterion (BIC)** also known as the Schwartz Criterion (Schwartz 1978).

BIC is a way of linearly integrating the error of a model and its complexity (df) which differs from the **Akaike Information Criterion (AIC)** (Akaike 1994) by its inclusion of a factor which depends on the sample size, N:

$$AIC = -2N \sum \mathbf{p\ ln\ q} + 2\,dF.$$

$$BIC = -2N \sum \mathbf{p\ ln\ q} + \ln(N)\,dF$$

These measure are unaffected by adding the constant $N \sum \mathbf{p\ ln\ p}$, which gives

$$AIC' = 2N \sum \mathbf{p\ ln\ (p/q)} + 2\,dF.$$

$$BIC' = 2N \sum \mathbf{p\ ln\ (p/q)} + \ln(N)\,dF$$

The equation **inherently** takes the **reference** to be the **top**.

The first term of AIC ′and BIC ′ is L^2(model), scaled model error; **we want it small.**

Good models also have **low** values of dF, model **complexity**.

So, good models have **low** (if negative, maximally negative) values of AIC ′ and BIC ′.

We thus want these **minimized**.

In OCCAM, , AIC and BIC are given relative to a reference model, usually the bottom (independence) model:

$$\Delta AIC = AIC(ref) - AIC(model) = AIC'(ref) - AIC'(model)$$
$$= [\,LR(ref) - LR(model)\,] + 2\,[\,df(ref) - df(model)\,]$$
$$= \mathbf{\Delta LR} + \mathbf{2 * \Delta df} \qquad \text{Note that } \mathbf{\Delta df \le 0}$$

$$\Delta BIC = BIC(ref) - BIC(model)$$
$$= \mathbf{\Delta LR} + \mathbf{\ln(N) * \Delta df}$$

For **reference=bottom**, ΔAIC and ΔBIC have *high* (positive) values for good models, since **ΔLR is always positive** & is the information *captured* in the model, and since **Δdf is always negative**, and thus **it diminishes the measure the more complex the model is** (for more complex models, Δdf is more negative), and we don't want complex models.

The **ln(N) factor in ΔBIC penalizes more complex models**, as long as N is equal to or greater than 7.4 (see below table). BIC is more conservative than AIC in recommending departures from the reference independence model. In our experience, models picked by ΔBIC do better on generalization (test or recall data) than the more complex models picked by ΔAIC.

N	ln(N)
5	1.609438
6	1.791759
7	1.94591
7.4	2.00148
7.5	**2.014903**
8	**2.079442**
9	**2.197225**
10	**2.302585**
15	**2.70805**

If **reference=top**, then

$$\Delta AIC = [\ LR(ref) - LR(model)\ ] + 2\ [\ df(ref) - df(model)\ ]$$
$$= [\ \ 0 \quad\quad - LR(model)] + 2\ \ \Delta df$$

Want LR(model) to be as *small* as possible, since it's error of the model, so **1st bracketed term should be as large as possible**.

We also want df(model) to be as *small* as possible, because we want a simple model, so the **2nd bracketed terms should be as large as possible**.

So, we want ΔAIC to be as **large** as possible.

So, **for reference being the bottom <u>or</u> the top**, we want ΔAIC and ΔBIC to be as **large** (positive) as possible.

Akaike, H. (1994). "Implications of Informational Point of View on the Development of Statistical Science." In *Proceedings of the First US/Japan Conference on the Frontiers of Statistical Modeling: An Informational Approach*, H. Bozdogan, ed., pp. 27-38, Kluwer Academic Publishers, the Netherlands.

Schwartz, G. (1978). *Ann. Stat.* 6, pp. 461-464.

8. Choosing models statistically

Choosing models

Confirmatory vs **exploratory** data modeling (a.k.a. data mining, knowledge discovery, machine learning)

Confirmatory modeling:

Could **fit** model to data and **evaluate (validate)** it with **same data**

(e.g., %correct for directed system)

But then don't know how well model would do on new data that it wasn't fit on.

So could do a **2-way data split: training / test**

Fit model on **training** data

Validate model on **test** data

Or, more elaborately, could do **N-fold validation** (N = 5 or 10 typically)

Divide data into training/test **N ways**, e.g.,

(i) **N blocks** with N-1 train & Nth test, or

(ii) **Randomly** with replacement

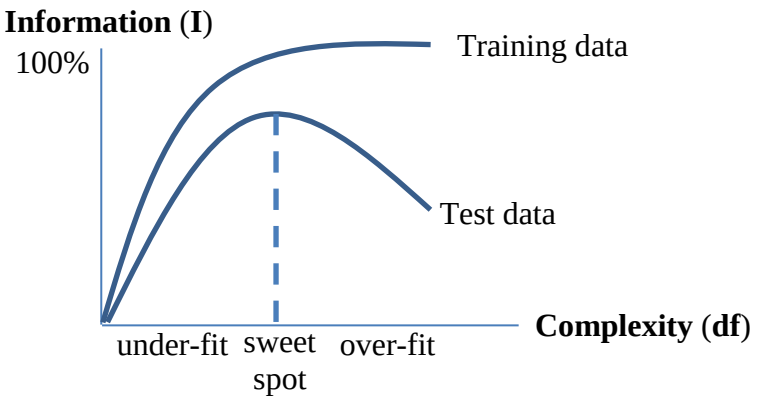
Exploratory modeling:

Find a good model **& fit** model on **training data**

Validate model on **test data**

If one plots **information vs complexity (df)**, on training & test data, one **typically** gets

(Not always; sometimes test performance tracks with training performance)

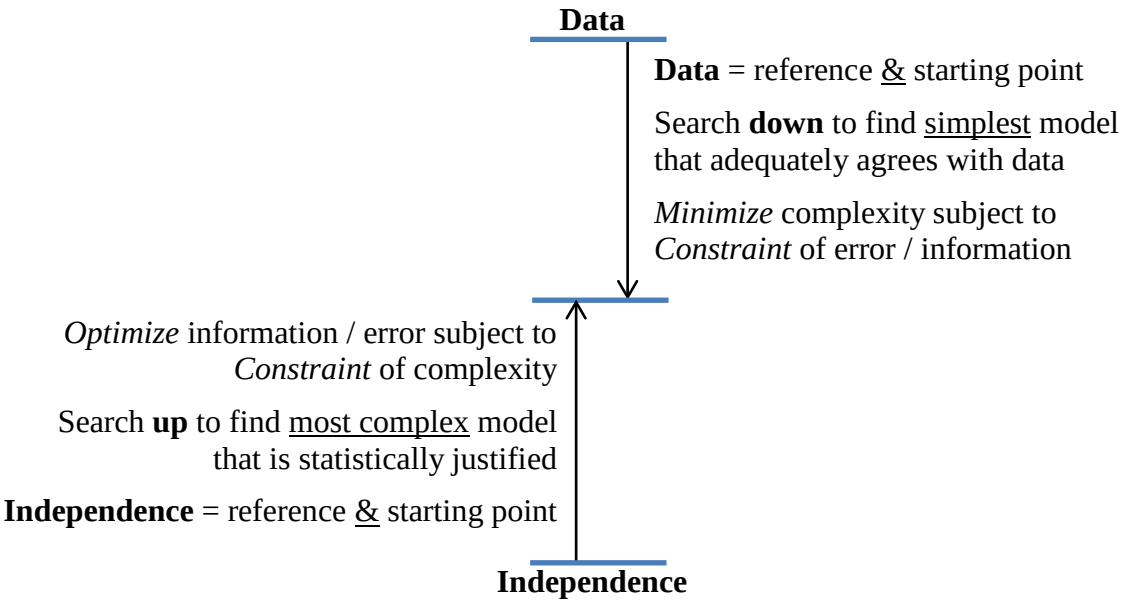


Can always get **better performance** (high I) in training data with models **more complex**

But when apply a **too complex model** to test data, it **doesn't generalize well**.

Question: **how to find** the **sweet spot** with **training** data?

Try to find sweet spot by searching down or up



But **reference & starting model** for search **don't have to be the same**.

Could have reference = top & search up or reference = bottom & search down,

But taking reference & starting models as the same is more typical.

Searches up / down might use one of **approaches** mentioned earlier **to find sweet spot**:

(a.1) Minimize complexity subject to information constraint:

(a.2) Find best model by **information/error subject to p-value constraint**

I.e., find best model considering statistical significance of information / error

(b) Maximize some weighted sum of the two criteria

I discussed (a.1) and (a.2) earlier; now discussing (a.2)

But there is **yet another way of picking a best model**:

3-way training/test/validation (really, training/pseudo-test/test) splits

Pick the **model fit on training data** that **generalizes best to test data**.

Validate it with hold-out sample, 3rd part of data.

Could do this N-fold.

FOR BOTH CONFIRMATORY & EXPLORATORY MODELING, one needs to get statistical significance of information captured or lost (error) in a model, generate **Likelihood-Ratio Chi-square** from **Information distance**, as follows:

Reference = top

$$\mathbf{L^2(m_0 \rightarrow m_j)} = \mathbf{1.3863\ N\ I(m_0 \rightarrow m_j)} \quad \textbf{ERROR}$$

$$= \mathbf{1.3863\ N\ T(m_j)}$$

$$= \mathbf{2\ N \Sigma\ p\ ln\ [\ p\ /\ q(m_j)\ ]}$$

K, p.87; N is sample size, K uses n; also K uses π instead of q for expected probabilities.

Note this is **different from ordinary Chi-square** = $N \Sigma\ (p - q)^2 / q$

Reference = bottom

$$\mathbf{L^2(m_j \rightarrow m_{ind})} = \mathbf{1.3863\ N\ I(m_j \rightarrow m_{ind})} \quad \textbf{INFORMATION CAPTURED}$$

$$= \mathbf{1.3863\ N\ [\ T(m_{ind}) - T(m_j)\ ]}$$

$$= \mathbf{2\ N \Sigma\ p\ ln\ [\ p\ /\ q(m_{ind})\ ] - 2\ N \Sigma\ p\ ln\ [\ p\ /\ q(m_j)\ ]}$$

$$= \mathbf{2\ N \Sigma\ p\ ln\ [\ q(m_j)\ /\ q(m_{ind})\ ]}$$

Note different forms in sum:

For Reference = top	p log **[p/q]**	form
		Difference between p & q, weighted by p
For Reference = bottom	p log **[q_2/q_1]**	form
		Difference between q_1 & q_2, weighted by p

This difference in forms will cause these two situations to not be fully symmetric.

L^2 is used here to test hypotheses where the reference is top or bottom, but actually, the reference can be **any model**.

Moreover, a model might be tested not only against a **fixed reference** of top or bottom, a "**cumulative"** test; we could also insist that it be satisfactory in every **"incremental"** test for **every step** down or up from the top or bottom reference. This would be a **more stringent** requirement of satisfactoriness.

To understand **how to use L^2 to assess statistical significance** of error or information captured, now discuss **reference models, null hypotheses, and Types I, II errors**.

REFERENCE MODELS, NULL HYPOTHESES, & TYPE I, II ERRORS

Reference = top; exploratory search usually (not necessarily) going top-down

NULL HYPOTHESIS (H_0): the model is indistinguishable from the data.

Note (K & B, p.30) that for some model, we **DO NOT WANT TO REJECT** this hypothesis since we want model to agree with data. More exactly, **we want the simplest structure, lowest on lattice, where null hypothesis is not rejected**.

In wanting to not reject the null hypothesis, this contrasts with common applications where we want to reject a null hypothesis and hence want L^2 to be large.

We REJECT the null hypothesis if the L^2 is large (for particular df, to be discussed later), i.e., if a lot of information is lost.

We DO NOT REJECT (speaking loosely, not rigorously, ACCEPT) the null hypothesis if L^2 is small, i.e., if very little information is lost.

Since we have a lattice of models, we have a lattice of hypotheses. We take a series of steps that go down the lattice until null hypothesis is rejected, and then back up one step.

Reference = bottom (exploratory search usually (not necessarily) bottom-up)

Or choose a different null hypothesis which **we DO want to reject:** independence model.

NULL HYPOTHESIS (H_0): the model is indistinguishable from independence.

(For some model,) we **DO** WANT TO REJECT this hypothesis.

We can continue to go up the lattice searching for the **most complex structure that is statistically justified**. It is statistically justified if L^2 is large, so we can reject the null.

For each model in this ascent, we'll ask if the model is both **cumulatively** significant relative to the **fixed reference** of independence and **incrementally** significant relative to the **immediately lower** model that we're going up from.

Type I and II errors

Possible errors in rejecting or not rejecting a model hypothesis:

TYPE I ERROR: **rejecting** the null hypothesis when it **should be not rejected**

TYPE II ERROR: **not rejecting** the null hypothesis when it **should be rejected**

TO DECIDE TO REJECT/NOT REJECT HYPOTHESIS, ONE USES L^2 & df & CHI-SQUARE TABLE. Table gives statistical significance of L^2:

K, p.62: consider calculating some L^2 value for df = 7. This **df** is really **Δdf**.

Top row is p-value, probability(type I error). Assume that want significance level of **.005.** That means we are **willing to accept a probability of .005 of making a type I error**.

				p-values					
Δdf	**0.200**	**0.100**	**0.075**	**0.050**	**0.025**	**0.010**	**0.005**	**0.001**	**0.0005**
5	7.289	9.236	10.008	11.070	12.833	15.086	16.750	20.516	22.106
6	8.558	10.645	11.466	12.592	14.449	16.812	18.548	22.458	24.104
7	9.803	12.017	12.883	14.067	16.013	18.475	20.278	24.322	26.019

Table shows that for Δdf=7, p=.005, *critical value* of Chi-square = 20.3. This means that one will get $L^2 > 20.3$ with probability = .005 *if the null hypothesis is true* (or $L^2 < 20.3$ with probability = .995 if the null hypothesis is true).

We will **reject null** whenever **L^2 exceeds L^2_c (critical value** of Chi-square).

input to table		output	decision re null hypothesis	
Δdf	**p_c**	**L^2_c**	**$L^2 < L^2_c$**	**$L^2 > L^2_c$**
7	.005	20.3	don't reject	reject *(difference is real)*

If we reject the null hypothesis whenever $L^2 > 20.3$, then .005 of the time Chi-square will be bigger than 20.3 **even** under the null hypothesis, so we will be in **error** .005 of the time in rejecting the null hypothesis, i.e., we have prob. of .005 of making a type I error.

If we were willing to tolerate a higher probability of a type I error, then we could reject null whenever $L^2 > 18.5$. This would produce a probability of type I error of .01, i.e., a higher probability of error, because we are rejecting null more readily.

What if our acceptable type I error was between .1 – .35, which Log-linear book (K&B) recommends for ref=top, say at .20. **We will <u>reject identity</u> with a <u>smaller difference</u>.**

input		output	decision	
Δdf	**p_c**	**L^2_c**	**$L^2 < L^2_c$**	**$L^2 > L^2_c$**
7	.20	9.80	don't reject	reject *(difference is real)*

Alternative way of using table: **go into table with df & L^2, get p**, then make decision by **comparing p to p_c**. Say **p_c = .01**:

input		output	decision re null hypothesis	
Δdf	**L^2**	**p**	**$p < p_c$**	**$p > p_c$**
7	20.3	.005	reject	
7	9.80	.20		do not reject

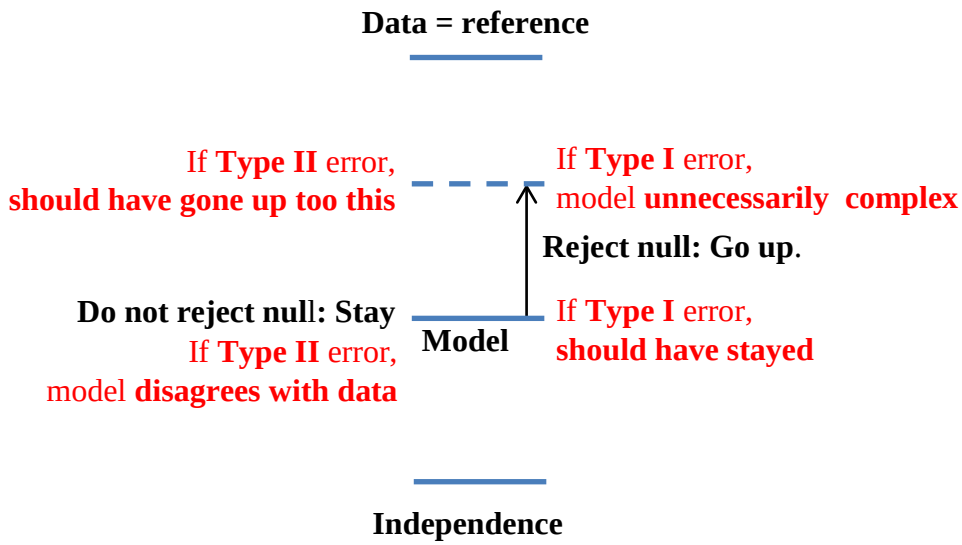
When $p>p_c$, then probability of error in rejecting null is greater than what I'm willing to tolerate, so I don't reject. if $p\leq p_c$, then my probability of error in rejecting null is less than what I tolerate, so I can go ahead and reject null confidently, without fear.

CONSEQUENCES IN EXPLORATORY SEARCH OF ERRORS WHEN USING DIFFERENT REFERENCE MODELS

Reference = top

If Chi-square test resulted in **rejecting the null hypothesis**, i.e., rejecting identity of model and data, saying that the **error is real, i.e., statistically significant**, **one would go up** the lattice to a more complex structure.

One wants to go as low as possible (compress maximally) without rejecting the null.



If this rejection is in error, it is a Type I error

If one made a **type I error**, one would be adopting a model which was UNNECESSARILY COMPLEX, i.e., one which includes unnecessary relations, **is not as simple as one can get**.

IF ONE WANTS TO AVOID A TYPE I ERROR, i.e., **require that p-value = probability of Type I error is very low**, ONE CAN CHOOSE A VERY SIMPLE STRUCTURE. We'd then be **unlikely to be wrong in rejecting** the null.

Thus, if we want the p-value (OCCAM calls this *alpha*) to be less than .05, i.e., very low probability of Type I error, we should go very far down.

IN THE LIMIT, if one chooses independence (!!), one is extremely unlikely to be wrong in rejecting the hypothesis that it fits the data.

BUT THIS IS ABSURD!!

A low p-value model is one that we're virtually certain (low probability of error) disagrees with the data. WE DON'T WANT A MODEL WE'RE *CERTAIN* DOESN'T FIT THE DATA!

THUS, WE **DON'T WANT** A MODEL WITH A SMALL P-VALUES (e.g.,< 0.05)

Type II error

But if one had made a type II error, one **should have rejected** this hypothesis and **gone up** the lattice to get closer to the data.

But **since one is in error**, one stays put and is using a **model that is TOO SIMPLE** to represent the data adequately, i.e., which omits some necessary relations, & thus is **in error with respect to the data**.

TO AVOID A TYPE II ERROR, CHOOSE A MORE COMPLEX STRUCTURE.

IN THE LIMIT, if one chooses a model which is the data itself, the possibility of incorrectly accepting (not rejecting) it is nil.

BUT WE **DON'T WANT** A VERY LOW P(TYPE II ERROR) THAT FORCES US TO CHOOSE THE DATA, WHICH IS **OVERFITTING**.

Which error is worse? Type I or II?

There's a **tradeoff** between type I and II errors.

If one allows only a very low probability of a type I error, e.g., .05 or .01, then one will have chosen so simple a structure that it is unlikely that we'll reject it wrongly.

But probability of type II error is then high since model is too simple.

Up to user, but **most users would say that Type II error here is worse, more serious.**

It's worse to choose a model that doesn't fit the data than a model that does and is just more complex than necessary.

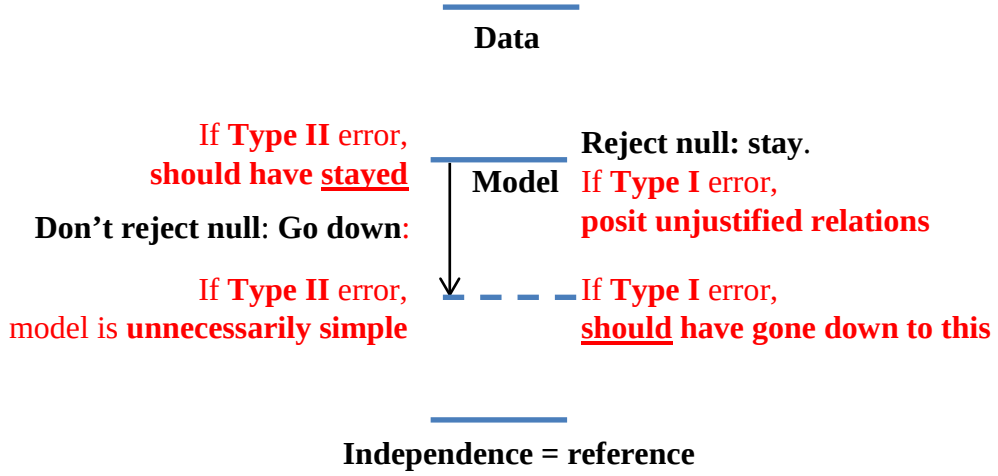
This values information/error criterion over complexity criterion.

What to do? Log-linear book, probably psychologically wedded to old p = 0.05 criterion says to relax this and maybe allow p to be between .1 and .35. See K&B, p.64.

To me, this isn't satisfactory. To be honest, we want p-value to be high!

Reference = bottom

If Chi-square test resulted in **rejecting the null hypothesis**, i.e., rejecting identity of model and independence, **one would stay**, since one is happy that our model is different from independence (OR one *could* try going up further).



Type I error

If one made a **type I error**, one would be adopting a model which was TOO COMPLEX, i.e., NOT STATISTICALLY JUSTIFIED, one that includes unjustified relations.
Its big difference from independence is not believable, given the data.

IF ONE WANT TO AVOID UNJUSTIFIED RELATIONS, ONE WOULD CHOOSE A SIMPLER STRUCTURE. We'd then be **unlikely to be wrong in rejecting** the null. The **smaller difference** from independence **is believable**, given the data.

WE DO WANT A MODEL WITH A SMALL P-VALUES (e.g.,< 0.05) BECAUSE WE WANT TO AVOID ASSERTING UNJUSTIFIED RELATIONS.

TYPE I ERROR OVER-FITS

If chi-squared test resulted in **not rejecting** the null hypothesis, **one would go down** since difference from independence is not believable.

Type II error

But if one had made a type II error, one **should have rejected** this hypothesis and **stayed**.

But **since one is in error**, one has gone down and is using a **model that is UNNECESSARILY SIMPLE**, that omits real (statistically justified) relations.

TYPE II ERROR UNDER-FITS.

Which error is worse? Type I or II?

Overfitting usually considered worse than underfitting. Type I is more serious here.

Implications

For directed systems, I strongly favor reference = bottom for three reasons:

1. Usual p-value expectations apply to reference = bottom but not reference = top.

2. Calculation of p(Type I error) more straightforward than p(Type II error); fewer assumptions needed

3. Computationally, for directed systems, computations are faster at the bottom of the lattice.

But for neutral systems, one sometimes encounters computational difficulties precisely at the bottom of the lattice, so I have no general recommendation for neutral systems. (I haven't done neutral system analysis that often.)