**New GAUSS programs for the revised version of Hey and Pace Mark 1**

I should start by documenting the documentation.

*Testing Theories (up to) 8.pdf* assumes CRRA but I think that this refers to an old specification of the experiment in which allocation was to all three colours.

*Background Material CARA (up to) 4.pdf* assumes CARA and has all the 5 models that we are currently considering.

*Theories Tested (up to) 9 pdf* assumes CRRA but is not consistent with the latest story.

I think it is simple: where we have for Type 1

x[ps,i]=0;

x[ps,j]=(e[pn,k]\*m[pn]+ln((w[ps,j]\*e[pn,j])/(w[ps,k]\*e[pn,k]))/r)/(e[pn,j]+e[pn,k]);

x[ps,k]=(e[pn,j]\*m[pn]+ln((w[ps,k]\*e[pn,k])/(w[ps,j]\*e[pn,j]))/r)/(e[pn,j]+e[pn,k]);

we replace it by

x[ps,i]=0;

x[ps,j]=((e[pn,j]^(r-1))\*(w[ps,j]^r)\*m[pn])/((e[pn,j]^(r-1))\*(w[ps,j]^r)+(e[pn,k]^(r-1))\*(w[ps,k]^r))

x[ps,k]=((e[pn,k]^(r-1))\*(w[ps,k]^r)\*m[pn])/((e[pn,j]^(r-1))\*(w[ps,j]^r)+(e[pn,k]^(r-1))\*(w[ps,k]^r))

and where we have for Type 2

x[ps,i]=(e[pn,j]\*m[pn]+ln((w[ps,i]\*e[pn,i])/(w[ps,j]\*e[pn,j]))/r)/(e[pn,i]+e[pn,j]);

x[ps,j]=(e[pn,i]\*m[pn]+ln((w[ps,j]\*e[pn,j])/(w[ps,i]\*e[pn,i]))/r)/(e[pn,i]+e[pn,j]);

x[ps,k]=0;

we replace it with ?????

x[ps,i]=

x[ps,j]=

x[ps,k]=0;

The new program(s) will have three *specifications* and five *preference functionals*.

**Specifications**

1. CARA plus normal (with mean 0 and standard deviation *σ*)
2. CRRA plus beta (with precision *s* so that the mean is the optimal proportion and the standard deviation decreases with *s*: see the document *Background Material.pdf*)
3. CRRA plus normal (with mean 0 and standard deviation *σ*)

**Preference Functionals**

1. SEU (was 1)
2. CEU (was 2)
3. AEU (was 4)
4. VEU (was 7)
5. CM (was 9)

These are called *esttype* in the program; I must change the numbering to be consistent with the above.

I first need to decide whether to have one program or three. I am going to start with the GAUSS program *hey and pace CARA 12.est* so let me look at that. At the moment, the structure is

*subjn=firsts;*

*do while subjn<=lasts;*

*esttype=1;*

*do while esttype<=ntypes;*

*esttype=esttype+1;*

*endo;*

*subjn=subjn+incs;*

*endo;*

so let me embed the different specifications inside the first but not the second loop. So we will get

*subjn=firsts;*

*do while subjn<=lasts;*

*specification=firstspecification;*

*do while specification<=lastspecification;*

*esttype=firsttype;*

*do while esttype<=lasttypes;*

*specification=specification+1;*

*esttype=esttype+1;*

*endo;*

*subjn=subjn+1;*

*endo;*

where *specification* takes the obvious meaning and the values 1, 2 and 3.

I have to be careful about notation:

**Likelihood functions:** *lle1, lle2, lle3, llp1, llp2, llp3* where the *e (p)* refers to the *e*stimation (*p*rediction) part and the 1, 2, 3 to the specification.

We already have *lle2* but I suspect that we will have to derive *llp2*. To get *lle3* and llp3 we need to change the bits in *lle1* and *llp1* from the CARA functions to the CRRA functions (see below).

**Preference functions:** *seua, seur, ceur, ceua, aeua, aeur, veua, veur, coma, comr* where the first three letters refer to the model *(seu, ceu, aeu, veu* and *com* – respectively SEU, Choquet, Alpha, Vector and Contraction) and the fourth letter refers to the utility function (*a* and *r* – CARA and CRRA).

We need to change *eu, ceu, aeu, veu* and *cm* to *seua, ceua, aeua, veua*  and *coma* respectively, then we need to import *eu, ceu, aeu, veu* and *cm* from *hey and pace 5.est* and rename them to *seur, ceua, aeur, veur*  and *comr* respectively.

We already have *lle2* but I suspect that we will have to derive *llp2*.

**Minor points**

The bounds on the *r* parameter for CARA and CRRA are explored in a Maple file in this directory called *utility functions.mws*. It seems that for CRRA *r* should be between 0.5 (very risk-averse and 25.0 (almost risk-neutral) – though one could go up to 100.0. For the CARA function it seems that we should have an *r* value between 0.001 (which is effectively risk-neutral, though we could go further) and 1.0 (which is very risk-averse though we could go further again). But what happens if the individual is risk-loving? I suspect it is all-or-nothing, but we should check.

The bounds on the *s* parameter are 0.005 and 1.0 for the normal distribution and 1.0 and 60.0 for the beta distribution. For the normal distribution, as these are precisions, then the variance is between 1.0 and

The allocation data is recorded to two decimal places. At the moment, I notice I use the densities for the normal and probabilities for the beta. I should be consistent. But note that the probabilities are just 0.01 times the densities – so maximising one rather than the other should make no difference. Let me work with the densities. This is OK except for specification 2 when the actual allocation is either all or nothing; in this case the density is 0. I have ‘translated’ this into the actual being withing 0.005 of the limit. But specification 2 does not do well.

The *pdf* of a beta distribution is given by *f(x) = Γ(α+β)xα-1(1-x)β-1/( Γ(α) Γ(β))* where *Γ(.)* is the gamma function. Now in GAUSS we have that *ln Γ(y)=lnfact(y-1)* where *lnfact(.)* is a built-in function. It follows therefore that *ln f(x) = lnfact(α+β-1) + (α-1)ln(x) + (β-1)ln(1-x) – lnfact(α-1) – lnfact(β-1).* I am using *θ1* and *θ2* instead of *α* and *β.* The function *lnpdfbeta(.)* finds the natural log of the *pdf* of a beta distribution.

The definitive program which now works is *hey and pace revision 2.est* which does both CARA and CRRA using proportions. But the log-likelihoods do not seem to be comparable. The program *hey and pace revision 3.est* tried to do total probabilities but failed. The program *hey and pace revision 1.est* is a dog’s dinner. To get totals from densities I could multiply the density by the width between observations – which is 0.01.