**Notes on Tmac**

**Inputs:**

data: vector of observed entries of tensor

known: vector of linear indices into observed tensor

Nway: size of tensor

coreNway: estimated ranks of all mode matricizations

opts.

maxit: max iterations

*tol: stopping tolerance*

maxT: max running time

alpha: weights on each mode

alpha\_adj: determine whether to dynamically adjust alpha

rank\_adj: rank adjusting strategy, can use different strategy for each mode

rank\_inc: rank increment if rank increasing strategy is used

rank\_min: min rank estimation

rank\_max: max rank estimation

X0: initial X for each mode

Y0: initial Y for each mode

**Outputs:**

X, Y: cell structures, each of length # of modes. X{i}\*Y{i} is a low-rank approximation of the tensor matricized to keep mode i in one dimension, and all other modes in the other dimension.

Out.

rank: rank at each iteration

hist\_rel: *relerr1: relative change array of total fitting*

*relerr2: array of total fitting*

iter: number of iterations

alpha: final weights alpha

**Other variables within Tmac:**

nrmb = norm(data);

estMnrm = sqrt(nrmb^2\*(prod(Nway)/length(known))); estimates Froebenius norm of underlying tensor using Froebenius norm of known elements.

alpha (weights on each mode matricization) can be adjusted adaptively, according to:

alpha = 1./(res.^2); alpha = alpha/sum(alpha);

In other words, if the residual is very low for one of the modes, this will be upweighted, and hence we will focus on getting this one right. **Does this mean that we only need to get one of the modes right in order to have an appropriate low-rank approximation of the tensor? Seems to me that each of the modes is important..**