A comparison of algorithms for fitting the parafac model
Tomasi and Bro

1 introduction

There are papers regarding the 3 main groups of fitting algoritms: alternating algorithms, derivative based methods, and direct procedures. on page 3.
Look at these references

Fitting models present numerous problems. One aspect affecting speed of convergence and retrieval of the underlying solution is the condition number of the loading matrices, which reflects both collinearity and relative magnitude of the factors.
compression has been reported to be beneficial in this respect and has the advantage of reducing the computational expense
Bro and Andersson, 1998; Kiers, 1998

Another problem is the so-called two factor degeneracy (2FD): the presence in teh solution of two factors are almost perfectly colinear but have opposite signs and almost cancel out each others' contribution. The problem is that the gegenrate factors can grow arbitrarirly large while loss function decreases very slowly.
2FDs may appear in the final solution as a consequence of a wrong estimate of the rank of the array or some aspects specific to the data set at hand.
A connection between the possible causes of 2FSs and the so-called swamps are sequences of iterations where the interim solutions contain features similar to 2FDs of increasing severity and the loss function decreases very slowly.
When an iterative algorithm encounters a swamp it either emerges from it after an unpredictibly large number of iterations or it reaches an earlier stop because of one of the convergence criteria is suddenly met.
Attention has been given to the development of new methods capable of dealing with swamps and more in general with the more frequent high collinearity case

Two examples are regularisation (Paatero, 1997; Rayens and Mitchell, 1997) and line search (Bro, 1998; Harshman, 1970)

2 algorithms

Rank Annihilation Factor Analysis
RAFA for extimating the concentration of chemical analyte in unknown sample matrix. Property was coined the second-order advantage as it is obtained by using second order or two0way structure of the indicidual measurements, instead of vectorising the matrix. 2nd order adv equivalent to the uniqueness of the trilinear structure. Method was originially intuitively appealing but unsatisfactory and slow.
Later it was shown possible to automate this search for rank-reduction.
Automated method called Generalized Rank Annihilation Method
GRAM works with two samples. Using only 2 samples there is a direct solution to the PARAFAC model based on solving a generalized eigenvalue problem. Though this solution is not least squares solution it works well for datat that are well approximated by a parafac model.
For more than 2 samples, the two loading matrices obtain from any two samples can be used for calculating the score values for all samples in a simple regression step.
Idea behind DTLD
Alternating strategies for selecting the two samples used in the GRAM step. DTLD is a direct method using a generalized eigenvalue problem of fixed known complexity. The required storage is represented in essence by the sole data array X, the 3 loading matrices and 3 matrices used to compress the array and to select the two samples used for gram step.

2.1 Alternating Least Squares

REduing optimisation problem to smaller sub-problems solved iteratively. Parameters to de det separated in diffeent groups and by fixing all but one a new loss function depending only on the set left free to vary is minimized. The solution is relatively simple to calculate. Next stages are applying principle to other parameters. The alg iterates from one set to the next until the variation of the loss function or of the parameters is less than a convergence criterion. Since all the steps are optimized the loss function bound to not increase at any step and tends to asymptocially to a min.

2.1.1 PARAFAC-ALS

The general approach of PARAFAC-ALS is the same presented by Kolda-Bader. The actual implementation of algorithm certain properties of Khatri-Rao Product and line searches are employed in order to accelerate the calculations. Look to those papers for speed up of my algorithm.
The shortcomings are the occasional slowness of convergence in presence of swamps or high collinearity. The loss function decreases almost linearly with the iterations while other methods can provide superlinear or quadratic convergence.
The memory consumption for PARAFAC-ALS is limited in essence: two arrays of size IxJxK, a matrix of size (IJ, JK, IK)xF for KRP and three loading matrices.

2.1.2 SWATLD and ASD

Recently several algorithms based on PARAFAC-ALS have been proposed. A comparison of the methods exits in the literature
Faber at al, 2003
only SWATLD and ASD have been chosen to rep this group of algorithms based on prelim test.
SWATLD not attempt to find the min of normal als instead alternates between min 3 different loss functions one per each of the loading matrices. structured as follows

$$L_C(C) = \sum_{k=1}^{K} \|(A^+ X_k - D_k B^T)^T D_B^{-1}\|_F^2 + \sum_{k=1}^{K} \|(X_k (B^+)^T - A D_k) D_A^{-1}\|_F^2$$

where DA and DB are FxF diagonal matrices holding the norm of the loading vectors of A and B,Xk is the kth frontal slab of X and Dk is an FxF diagonal matrix with the elements of the kth row of C on the diagonal. Thus given A and B a new extimate of C is found as equation 9.
SWATLD yields solutions less affected by deviations from exact trilinearity than the least squares one. ADV in terms of convergence speed and resistance to over-factoring. (opt criterion not well defined so alg must be used w/care)
A,B and C must have full column rank in order for the algoirthm to resolve uniquely the components of interest.
SWATLD needs about as much memory as ALS.
ASD
based on hyp of A and B having full column rank. A and B have inveses. The slab flattening can then be expressed as equation 10.
The loss function is minimized by ASD using an alternating algorithm of the form equation 11. one thing to note is that the final solution is not providing a least squares fit and must be assessed with care.
ASD employs a compression procedure based on singular value decomposition that allows a sig reduction in the number of operations per iteration. Compression step further reduces memory consumption of the algorithm.

2.2 Gauss-Newton Method

Minimizing the original loss function is a relatively difficult non-linear least squares problem several have proposed the use of algorithms employed for such optimization problems.

Check out the Gauss-Newton method Bjorck 1996

some mods from basic to deal with Parafac can be found in equation 12 and 13.
The method works under the assumption that the residuals in the neighnourhood of point p^0 can be approximated by a taylor expansion truncated after the linear term. This method requires laplacian and updates a jacobian. The min of a function is a stationary point, the algorithm should stop if $\|g\|_\infty$ is smaller than a sufficiently small number.

The iterative method for fitting the parafac model proposed by Hayashi and Hayashi 1982
Line search procedures and trust region methods can be used to deal with the problem of convergence. The sparse structure of the jacobi in the parafac case creates observations regarding whether or not it is advantageous to explot it in solving without using the system of normal equations. A reduction in the number of operations could be attained by using a QR algorithm designed for this type of problem. Greater savings can be achieved considering the I blocks in Ja are identical. The loss function is solved by means of system of normal equations using Cholesky decomposition and back substitution. No weights are required, the Jacobian J is not formed and both JTJ and JTr can be computed directly without recurring to sparse matrices.

2.2.1 dGN

the previous algorithm can't yield a globally convergant algorithm for fitting the parafac model. This leads to singularity in the Hessian which can be handled by damped Gauss Newton algorithm (dGN) modified and constrained normal equations are used to solve.
With specific reference to memory consumption, dGN is the most expensive method together with PMF3.

2.2.2 PMF3

modification to dGN: the presence of a regularization factor and a very specific non-linear update. Also emplys a line search procedure that is called when the algorithm diverges. It includes a weighted least squares loss function and possible non-negative constraints on the parameters.

2.3 line search

applicable for all iterative methods with a well-defined loss function. In PARAFAC ALS line search is employed to speed up the algorithm.
After the s-th iteration, the variation of the three loading matrices wrt iteration S-1 is calculated ($\Delta A = A^{(s)} - A(s-1)$) and is used to linearly predict the corresponding matrix d iteration ahead ($A^{(s+d)} = A^{(s)} + d\Delta A$) where d is determined empirically as a function of the number of iterations
Bro 1998
There is a separate method for line search in derivative based methods.

2.4 Compression

Main problem with der based methods is, though they require fewer iterations than ALS, the memory req are higher. Possible solution is reduce dim of the array on which Gauss-Newton method is used. Based on the Tucker3 model fulfills this task. Though it adds complexity, provides a large reduction in number of operations. It only needs to be done once. Once the parafac model has been calculated on the core extracted by the tucker3 algorithm the solution can be expanded to its original dim providing good starting values for the ALS standard algorithm which is used only to refine it. Using this method its not necessary to calculate an exact Tucker3 model because the compression array is only approx , the final model need to be refined anyhow.

3 experimental section

Nine algorithms are studied using simulated and real data: DTLD-GRAM, PARAFAC-ALS, ASD, SWATLD, dGN, PMF3, ALS with compression, dGN with compression and PMF3 with compression.
using matlab

3.0.1 simulated data

A total of 1440 models were fitted using each algorithm, randomly noise to create some coliniearity was generated in this set of data

3.0.2 Real data sets

Flourescents measurements were tested

3.1 initialization and conventions

All algorithms but one got guesses from running 5 standard ALS iterations starting with 10 different sets of matrices of uniform distributed values and choosing the best fitting set.

3.2 Criteria of interest

4 different parameters considered, value of loss function, occurence of full recoveries and of 2FD's and mean squared errors of parameter estimates. For real set conc of different constitutes was available and Root Mean Squared Error in Calibration using linear regression.
One can consider a factor as completely recovered if there is one component in the estimated model having a congruence greater than a certain recovery threshold.
Occurence of a 2FD is established when the congruence between 2 components within the same solution is less than or equal to -0.8.
The second main aspec was computational efficiency, which uses as few floating point operations as possible. other aspec as memory traffic and data access

3.3 Results and discussion

3.3.1 simulated data

Overall iterative solution retrieve correct solution in more than 50% of the cases. Best agorithm in this respect is SWATLD, then damped Gauss-Newton w/compression. The direct method worked only 26% of the time. ALS folls in lower part of range with 57%
performance of ALS equivalent to those of derivative based algorithms and SWATLD when rank was correctly estimated. ALS was heavily influnced by incorrect estimation of the rank. Compression helped with that. SWATLD worked best wehn rank overestimated and equal or better when rank corretly set. ALS did worse in 21 setups in case of over-factoring. der-based methods seem more robust than ALS wrt over-factoring even when ALS has compression.

der have res to overfactoring because of regularization terms. Helps deal with 2FDs. SWATLD did not yield any 2FD in any of the models regardless of over-factoring.