diffusion-fundamentals.org

Matrix Factorisations for the Estimation of NMR Relaxation Distributions

Paul D. Teal

Victoria University of Wellington, Wellington, New Zealand
paul.teal@vuw.ac.nz

Abstract

The two most successful methods of estimating the distribution of NMR relaxation times from two dimensional data are firstly a data compression stage followed by application of the Butler-Reeds-Dawson (BRD) algorithm, and secondly a primal dual interior point method using a preconditioned conjugate gradient (PCG). Both of these methods have been presented in the literature as requiring a truncated singular value decomposition of matrices representing the exponential kernels. Other matrix factorisations are applicable to each of these algorithms, and which demonstrate the different fundamental principles behind the operation of the algorithms. In the case of the data compression approach the most appropriate matrix decomposition specifically designed for this task is the rank-revealing QR (RRQR) factorisation. In the case of the interior point method, the most appropriate method is the LDL factorisation with diagonal pivoting, also known as the Bunch-Kaufman-Parlett factorisation. The details of these differences are discussed, and the performances of the algorithms are compared numerically.

1. Introduction

An important approach to the interpretation of nuclear magnetic resonance (NMR) measurements involves the estimation of the distribution of relaxation times. This is typically performed using a numerical inversion of a Laplace transform, which is known to be an ill-conditioned procedure.

The observed data can be expressed in terms of the spectra of relaxation times using an integral. A common experiment uses two dimensional longitudinal (T_1) and transverse (T_2) experiment [1], in which case this integral takes the form:

$$Z(\tau_1, \tau_2) = \iint k_1(\tau_1, T_1) k_2(\tau_2, T_2) S(T_1, T_2) \, dT_1 \, dT_2 \qquad (1)$$

with kernels k_1 and k_2 given by

$$k_1(\tau_1, T_1) = 1 - 2e^{-\tau_1/T_1}$$
$$k_2(\tau_2, T_2) = e^{-\tau_2/T_2} \qquad (2)$$

The spectra of other time constants, such as diffusion, may also be of interest, and T_2–T_2 experiments are also common. The methods presented here are equally applicable to these other experiments. Experimental data is collected at discrete values of time, and so (1) is frequently expressed as

$$Z = K_1 S K_2^T + E \qquad (3)$$

where $K_1 \in \mathcal{R}^{N_1 \times G_1}$, $K_2 \in \mathcal{R}^{N_2 \times G_2}$, $S \in \mathcal{R}^{G_1 \times G_2}$, $Z \in \mathcal{R}^{N_1 \times N_2}$, and where $E \in \mathcal{R}^{N_1 \times N_2}$ represents white Gaussian noise having zero-mean. If the columns of each of Z, S and E are stacked to obtain $z = \mathrm{vec}(Z)$, $s = \mathrm{vec}(S)$ and $\eta = \mathrm{vec}(E)$, and if $K = K_2 \otimes K_1$, then (3) can be expressed as

$$z = Ks + \eta \qquad (4)$$

Since the noise η is white and Gaussian, the maximum likelihood estimate of s is given by the squared error minimising solution. The matrices K_1 and K_2 are very ill-conditioned, and hence solving this solution is known to be an ill-posed problem.

Historically there have been two main approaches to solve such problems; both involving some form of capacity control. The first approach is known as truncated singular value decomposition (TSVD) and avoids the amplification of noise associated with small singular values by replacing the inverse of K by a matrix of reduced rank. It was recognised by proponents of this approach that the singular value decomposition itself was not actually required, and that the same result could be achieved with the computationally less expensive QR factorisation [2–4]. We will return to this important point later.

The second approach uses Tikhonov regularisation, with the solution $\hat{s}$ to (4) being given by

$$\hat{s} = \arg\min_s \|z - Ks\|^2 + \alpha R(s) \qquad (5)$$

with $\alpha > 0$ controlling the extent of the imposed regularity, and $R(s)$ may equal $s^T s$ for simple energy control, although other possibilities include curvature control using $R(s) = s^T Q s$ where Q is a matrix implementing an approximate first or second derivative operation. Entropy control can be implemented using $R(s) = \sum s_i \log s_i$ [5].

Another approach to providing capacity control is to fit a finite number of exponentials to the data, rather than a distribution, and to limit the number of these exponentials [6].

For many measurements, S can be considered as a distribution, so that all of the elements of s must be non-negative, and imposing this constraint also improves the resolution of the estimated spectra. This constraint is used in all the methods presented here. However, it should be noted that under some conditions the T_1–T_2 spectrum cannot be considered to be strictly non-negative [7, 8], and so this condition cannot always be applied. An effective approach that penalises zero-crossings rather than demanding non-negativity is described in [9].

The amount of data collected for one measurement can be very large, with typical sizes N_1=30–50 and N_2=4000–10000. If a grid used for the estimated distribution has 100×100 elements, then the matrix K could have 5×10^9 elements, which at present is far too many for practical computation.

The two most successful methods of solving these problems of ill-conditioning and data size are the data compression approach of [10] and the interior point method of [11]. Both of these use singular value decompositions of the matrices K_1 and K_2, although the SVDs are used in very different ways. The fact that the SVD is such a general factorisation obscures the differences between the ways that the SVD is used in these methods.

In the remainder of this paper we demonstrate that less general matrix factorisations than the SVD can be used for each of these methods. In doing so we highlight the differences between the two approaches. These matrix factorisations have not been previously applied to the NMR inversion problem.

2. Data Compression

2.1 Outline of the method

The SVD of K_1 and K_2 are given by $K_1 = U_1 \Sigma_1 V_1^T$ and $K_2 = U_2 \Sigma_2 V_2^T$ respectively. Since the kernels k_1 and k_2 are smooth, the singular values decay very rapidly, and can be truncated at r_1 and r_2 non-zero values respectively, and hence obtain approximations $\tilde{U}_1 \in \mathbb{R}^{N_1 \times r_1}$, $\Sigma_1 \in \mathbb{R}^{r_1 \times r_1}$, $V_1 \in \mathbb{R}^{G_1 \times r_1}$.

The columns of U_1 are an orthonormal basis for the column space of K_1, and hence $P_1 = U_1U_1^T$ forms a projection onto this space [e.g., 12]. Analogous expressions hold for the decomposition of K_2. The projection of the data Z onto the column space of K_1 and K_2 is given by $U_1U_1^T Z U_2 U_2^T$, and the data can thus be compressed with little loss of information by considering the matrix $\tilde{Z} = U_1^T Z U_2 \in \mathbb{R}^{r_1 \times r_2}$.

K_1 is clearly unchanged by projection P_1, and so by considering the squared Frobenius norm of $P_1(Z - K_1S)$ as the sum of the squared vector norms of each of its columns, it follows that

$$\|P_1(Z - K_1S)\|_F^2 = \|Z - K_1S\|_F^2 - \|Z\|_F^2 + \|P_1Z\|_F^2 \tag{6}$$

A similar result holds for the projection $P_2 = U_2U_2^T$, and hence

$$\|Z - K_1SK_2^T\|_F^2 = \|P_1(Z - K_1SK_2^T)P_2\|_F^2 + \|Z\|_F^2 - \|P_1ZP_2\|_F^2$$

The above expression is inexact when a truncated SVD is used, but the rapid decay in singular values means the approximation is very accurate with only a few singular values. Since U_1 and U_2 have orthonormal columns, it follows that $U_1^T P_1(Z - K_1SK_2^T)P_2U_2 = U_1^T(Z - K_1SK_2^T)U_2$ has the same Frobenius norm as $P_1(Z - K_1SK_2^T)P_2$, and so

$$\arg\min_{S \geq 0} \|Z - K_1SK_2^T\|_F^2 = \arg\min_{S \geq 0} \|\tilde{Z} - \tilde{K}_1S\tilde{K}_2^T\|_F^2 \tag{7}$$

where $\tilde{K}_1 = \Sigma_1 V_1^T$ and $\tilde{K}_2 = \Sigma_2 V_2^T$. The truncation of singular values drastically reduces the number of rows in $\tilde{K}_1$ and $\tilde{K}_2$ compared to K_1 and K_2 (to r_1 and r_2 respectively), so the number of elements of the Kronecker product $\tilde{K} = \tilde{K}_2 \otimes \tilde{K}_1$ is few enough for computation to be feasible. If we form a vector of the compressed data $\tilde{z} = \text{vec}(\tilde{Z})$, then the problem can be expressed as

$$\min_{s \geq 0} \|\tilde{z} - \tilde{K}s\|^2 + \alpha R(s) \tag{8}$$

This problem can then solved using the BRD method of [13], which can be interpreted as solving the Legendre-Fenchel dual problem [5, App. A]. For the regularisation $R(s) = s^Ts$, this involves minimisation via Newton's method of the function

$$\chi(c) = \frac{1}{2}c^T [G(c) + \alpha I] c - c^T\tilde{z} \tag{9}$$

where

$$G(c) = \tilde{K}D\tilde{K}^T \tag{10}$$

where $D = \text{Diag}(u(\tilde{K}^Tc))$, $c = (\tilde{z} - \tilde{K}s)/\alpha$ and $u()$ is the Heaviside function. The Diag() operator constructs a diagonal matrix from the vector which is its argument. The application of Newton's method to this problem requires the solution of equations of the form

$$\Delta_c = c - (G(c) + \alpha I)^{-1}\tilde{z} \tag{11}$$

Further details of the method, and the selection of the regularisation parameter α are important but do not concern us here.

This method is in effect using both the truncated SVD and the Tikhonov regularisation approaches, although the SVD truncation is only applied to the point that it does not significantly change the results. Thus it is the Tikhonov regularisation that is primarily responsible for dealing with the ill-posed nature of the problem, while the SVD truncation is responsible for reducing the problem size.
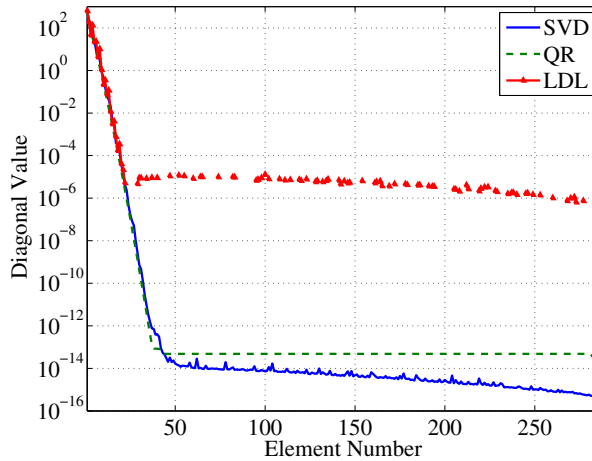
Figure 1: Typical decay of diagonal values of the various matrix decompositions.

2.2 Alternative Matrix Factorisation

A more simple matrix factorisation than the SVD is the QR decomposition, $K = QR$, where Q is a matrix of orthonormal columns and R is an upper triangular matrix. The diagonal elements of R are not guaranteed to be in decreasing order, although the rapid decrease in the values of the kernels k_1 and k_2 means that in practice the diagonal elements do decay rapidly. Rank-revealing forms of the QR factorisation (RRQR) include column pivoting or other means to obtain $K = QRE^T$ where E is a permutation matrix. This improves the numerical stability of the factorisation process and can be arranged to guarantee that the diagonal elements of R are in decreasing order.

We can use this rapid decay to truncate the number of columns of Q_1 and Q_2, and the number of rows of R_1, R_2 (where $K_1 = Q_1 R_1 E_1^T$ and $K_2 = Q_2 R_2 E_2^T$). Then if we set $\tilde{Z} = R_1^T Z R_2$, $\tilde{K}_1 = R_1 E_1^T$ and $\tilde{K}_2 = R_2 E_2^T$, we obtain data compression in an similar way to that obtained using SVD. Note that if the number of rows of K exceeds the number of columns, then a form of the factorisation should be used in which Q rather than R is of the same size as K.

Numerical computation of the SVD is typically begun using the QR factorisation. In theory, omission of the later stages means that the result obtained using the truncated QR is not guaranteed to the closest approximation to K that can be obtained with a limited rank matrix. In practice however, the rate of decay of the diagonal elements of R is very similar to the rate of decay of the singular values of K, as shown in Fig. 1. Rank-revealing forms of the QR are designed to deal with reliably detecting when singular values reach the limit of machine precision, and perform this task very reliably [14]. The data compression and final results obtained using the QR factorisation are very similar to those obtained using the SVD. If the matrix K is square or nearly square, the large number of zeros it contains using the QR approach can speed up operation of the BRD algorithm, but usually this is not the case.

For typical sizes of K_1 and K_2, QR factorisation is faster than SVD decomposition by a factor of about 8 and 1.2 respectively, although the effect of this on the total processing time is small, as the factorisation time is only a small part of this total.

In summary: the important requirement of a matrix factorisation for the data compression approach is that it provide a low rank approximation to the matrices K_1 and K_2. The QR algorithm is an effective choice for this task, and is simpler and faster than the SVD.

3. Interior Point Methods

3.1 Outline of the method

In this section we follow standard convex optimisation theory (as presented in [15] for example) to the problem (5) with the constraint that $s \geq 0$. Both primal-dual and barrier methods are derived.

To solve the constrained problem

$$\min_{s \geq 0} \|z - Ks\|^2 + s^T Rs \tag{12}$$

we can write the Lagrangian as

$$L(s, \lambda) = \|z - Ks\|^2 + s^T Rs - \lambda^T s \tag{13}$$

where λ is a vector of non-negative Lagrange multipliers.

At optimality we must have

$$\frac{\partial L}{\partial s} = 2K^T Ks - 2K^T z + 2\alpha Rs - \lambda = 0 \tag{14}$$

and $\Lambda s = 0$ where $\Lambda = \mathrm{Diag}(\lambda)$. To obtain an iterative algorithm, the duality condition $\Lambda s = 0$ is relaxed to $\lambda_i s_i = \mu$, where μ a parameter which is reduced after each iteration to enforce duality gradually. For a given value of μ, Newton's method can be applied, each step of which requires solving

$$\begin{bmatrix} 2K^T K + 2\alpha R & -I \\ \Lambda & \mathrm{Diag}(s) \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} \lambda - 2K^T Ks + 2K^T z - 2\alpha Rs \\ \mu - \Lambda s \end{bmatrix} \tag{15}$$

for Δs and $\Delta \lambda$. We can eliminate $\Delta \lambda$ to obtain

$$(2K^T K + 2\alpha R + \mathrm{Diag}(s)^{-1}\Lambda)\Delta s = -2K^T Ks + 2K^T z - 2\alpha Rs + \mu(1/s) \tag{16}$$

Following a derivation which is only slightly different from that shown above for the primal-dual method, we can obtain an iteration in which non-negativity is instead enforced by adding a logarithmic barrier function scaled by $1/t$ [15]:

$$(2K^T K + 2\alpha R + \mathrm{Diag}(s)^{-2}/t)\Delta s = -2K^T Ks + 2K^T z - 2\alpha Rs + (1/ts) \tag{17}$$

The barrier method does not appear to have been previously applied to this problem.

For both the primal-dual method, and the barrier method, a simple backtracking algorithm, as described in [15] can be used to ensure that non-negativity is retained at each Newton iteration.

An important observation to make at this point is that the matrix K only occurs in (16) in the forms $K^T z$, $K^T Ks$ and $K^T K$. Both $K^T z = \mathrm{vec}(K_1^T Z K_2)$ and $K^T Ks = \mathrm{vec}(K_1^T K_1 S K_2^T K_2)$ are readily calculated. The matrix $H \triangleq K^T K = (K_2^T K_2) \otimes (K_1^T K_1)$ typically has of the order of 100 million elements, and so it is very much smaller than K, which typically has of the order of 1–10 billion elements, and so in this form the problem is feasible without truncation on ordinary desktop computers.

However, the computation can be considerably further reduced by using the approach of [5]. Using the same TSVD as in Section 2.1, with $K^T K \approx \tilde{V}\Sigma^2 \tilde{V}^T$, and applying the matrix inversion lemma, the inverse of the matrix on the left of (16) is given by

$$P = A - A\tilde{V}((2\Sigma^2)^{-1} + M)^{-1}\tilde{V}^T A \tag{18}$$

where $A = (2\alpha R + \mathrm{Diag}(s)\Lambda)^{-1}$ and $M = \tilde{V}^T A \tilde{V}$. The truncation of the SVD means that P is not the exact inverse required, but is certainly accurate enough to provide an effective preconditioner for solving (16) or (17) using the preconditioned conjugate gradient (PCG) method [16]. With the exception of the preconditioner (18), all of the calculations of the PCG algorithm can be efficiently calculated using the exact but factored forms of K_1 and K_2, rather than $K = K_2 \otimes K_1$. M can be calculated using [5]

$$M_{ij} = \frac{1}{\alpha} \sum_{f=1}^{G_1} \sum_{g=1}^{G_2} S_{fg}(\tilde{V}_1)_{fa}(\tilde{V}_2)_{gb}(\tilde{V}_1)_{fb}(\tilde{V}_2)_{gd} \tag{19}$$

where (a, b) and (c, d) are row and column indices corresponding to the linear indices i and j respectively. This results in a very efficient algorithm.

Note that for large values of α, the interior point methods may not converge reliably. This difficulty can be overcome to some extent by commencing the algorithm with a small value of α, which is gradually increased towards the target value as the iterations progress. For the results in Section 4. this approach was used whenever the value of α was greater than 10^{-3}.

3.2 Alternative Matrix Factorisations

The interior-point method makes no use of the unitary matrices U_1 and U_2, and it is apparent that the required factorisations are the eigendecomposition of $K_1^T K_1$ and $K_2^T K_2$. The factors that can be obtained using the eigen-decomposition $K^T K = V D V^T$ are functionally identical to those obtained using the SVD of K, but are generally calculated much more rapidly (about 12 times faster for typical sizes of K_2, but 3 times slower for typical sizes of K_1).

As for the data compression method, the QR factorisation can be used instead of the SVD: if $K_i = QRE^T$, then $K_i^T K_i = ER^T Q^T QRE^T = ER^T RE^T$. If we define $D = \text{Diag}(\text{diag}(RR^T))$ and $V = ER^T D^{-1/2}$, we can obtain $K_i^T K_i = V D V^T$ which can be truncated and used in an identical way to the eigendecomposition. If the matrix K_1 or K_2 has more columns than rows this is a suitable method because $K_i^T K_i$ is actually rank deficient, rather than being only numerically ill conditioned, and this is accurately reflected in the sizes of D and V.

If K_1 or K_2 has more rows than columns, it is more computationally efficient to perform a factorisation of the square matrix $K_i^T K_i$, rather than of the matrix K_i itself. A suitable choice besides eigendecomposition is the LDL decomposition. The LDL can be performed using diagonal pivoting, resulting in $K_i^T K_i = ELDL^T E^T$ where D is a block diagonal matrix with blocks of size 1 or 2, L is lower triangular, and E is a permutation matrix. This is also known as the Bunch-Kaufman-Parlett factorisation [17] or Cholesky LDLt decomposition. The products $K_i^T K_i$ are positive semi-definite, and so any blocks in D of size 2 only occur as a result of numerical imprecision, and can be removed. As for the QR factorisation, the pivoting, leads to very rapid decay of the elements of D. Usually the diagonal elements of L are all equal to 1, but the rate of decay of D can be enhanced by instead scaling the columns of L to have unit norm. For typical sizes of K_1 and K_2, this factorisation is about 3 and 20 times faster respectively than using the SVD .

In summary: the important requirement of a matrix factorisation for interior point methods is that of providing a computationally efficient approximate inverse of $K_1^T K_1$ and $K_2^T K_2$ for the purpose of creating a preconditioner for the PCG algorithm. An effective choice for this task is the LDL decomposition.

4. Results

The primary purpose of this paper is to contrast the data compression and interior point approaches in terms of their use of matrix factorisations. However, this section will additionally demonstrate the effectiveness of the alternative matrix factorisations proposed, and compare the performance of the algorithms.

All three of the algorithms discussed in this paper produce visually identical recovered relaxation distributions (except as noted below). Hence we do not present figures of these distributions. However, we are interested in the relative speed of the various algorithms, and hence compare their speeds using plots of the objective function (5) versus computation time for small and large values of α.

The exact objective function can be efficiently calculated without truncation using the matrix trace operation as

$$\text{tr}\left(ZZ^T\right) - 2\,\text{tr}\left(S^T K_1^T Z K_2\right) + \text{tr}\left(S^T K_1^T K_1 S K_2^T K_2\right) + \alpha\|S\|_F^2 \tag{20}$$

some components of which can be precomputed. None of the algorithms explicitly require calculation of the objective function, and so the time required to do so is not included in the computation times shown.

The data used as input to the algorithms consisted of $N_1 = 50$ values of τ_1 logarithmically spaced on $[10^{-4}, 10]$ seconds by $N_2 = 10000$ values of τ_2 linearly spaced on $[3.5 \times 10^{-4}, 3.5]$ seconds. This was synthesised from a distribution consisting of various geometrical shapes on a logarithmically spaced grid of $G_1 = 100$ values on $[10^{-2}, 10]$ seconds by $G_2 = 101$ values on $[10^{-2}, 10]$ seconds.

In every case where matrix factorisation was required, the QR was used for the BRD algorithms, and LDL for the interior point methods. For the BRD method, the factorisation truncation threshold was set to 10^{-4}.
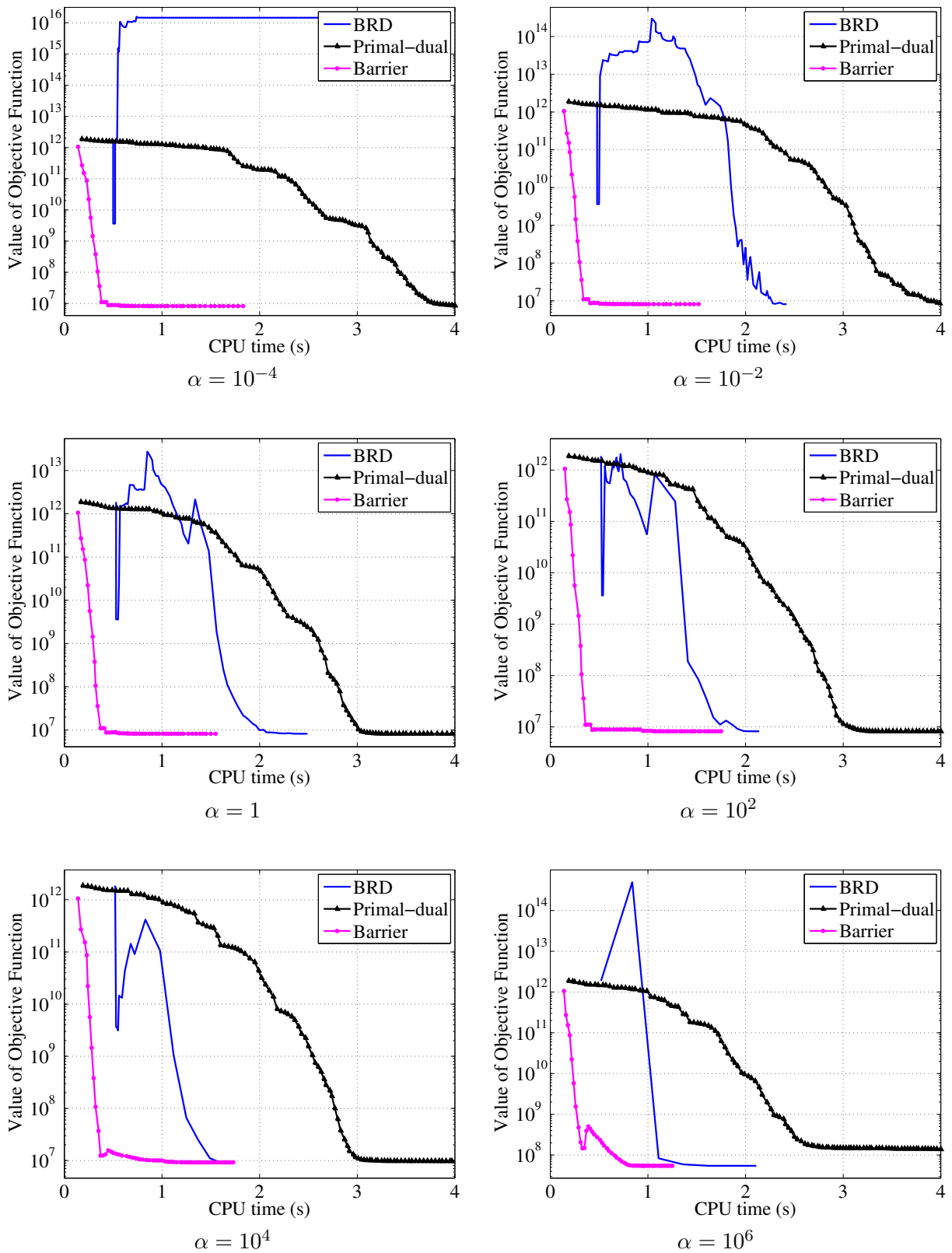
Figure 2: Comparison of the convergence of the algorithms for various values of the regularisation control parameter α.

For the barrier method, the multiplier of the logarithmic barrier function was chosen to be initially 10^{-3}, and was increased by a factor of 4 whenever the proportional decrease in objective function plus

barrier function was less than 10^{-5}.

The results are presented in Fig. 2 for a computer having a CPU clock speed of 2.83 GHz. Although the QR decomposition is faster than the SVD, the LDL is faster still, and this explains the difference in the timing of the *first* iteration of each of the methods. Note that the convergence *rate* of the methods shown is practically indistinguishable from when the SVD is used. The only difference is that the traces shown start later (shift to the right) when the SVD is used. For this particular example on the computer used, using the SVD results in all three curves starting at about 0.6 second.

Most of the methods were found to fail (not converge on useful results) occasionally. This was more likely to occur with the BRD based methods for small values of α and more likely to occur for the primal-dual method for very large values of α. The results shown in Fig. 2 are typical in this respect. The possibility of failure highlights the importance of care in interpretation of the results of these experiments.

For small values of α, the barrier method is the fastest and most reliable. For large values of α, the compression-BRD method is the fastest, and the barrier method does not obtain as low a value of the objective function as the compression-BRD method. For values of α larger than about 10^2 this difference becomes visually significant in the resulting relaxation distribution.

5. Conclusions

In this paper we have demonstrated the use of different matrix factorisations for the estimation of NMR relaxation distributions.

The fact that both the data compression method has required a TSVD, and the interior-point method can be accelerated using a TSVD gives the superficial appearance of resemblance between the methods. The methods are in fact very different. One important contrast is the matrix which is inverted at each stage of these algorithms. In the case of the data compression method, using BRD, this matrix is $G + D_{dc}$ where D_{dc} is a diagonal matrix and G defined as in (10) by $K \operatorname{Diag}() K^T$. In the case of the interior point methods, the important matrix is $H + D_{ip}$ where D_{ip} is a diagonal matrix and $H = K^T K$. G is of size $r_1 r_2 \times r_1 r_2$ which decreases with stronger truncation of the SVD. By contrast, the size of H is fixed at $G_1 G_2 \times G_1 G_2$ (a closer analogy does exist between the matrices G and M, which does decrease in size with stronger truncation of the SVD, though G and M also have differences).

Other differences are highlighted by the matrix factorisations that are applicable in each case: for the BRD method, a QR factorisation is applicable, whereas for interior point methods, LDL factorisation is the most useful.

The values of the regularisation parameter α for which the methods perform best is also different. The interior point barrier method is effectively solving the primal problem, whereas the BRD method is solving a form of dual. The interior point primal-dual method does not appear to excel for any values of the regularisation parameter.

References

[1] H. Peemoeller, K. Shenoy, M.M. Pintar, D.W. Kydon, and W.R. Inch. Improved characterization of healthy and malignant tissue by nmr line-shape relaxation correlations. *Biophysical J.*, 38:271–275, Jun 1982.

[2] L. Eldén. Algorithms for the regularization of ill-conditioned least squares problems. *BIT*, 17:134–145, 1977.

[3] P.C. Hansen. The truncated SVD as a method for regularization. *BIT*, 27:534–553, 1987.

[4] T.F. Chan and P.C. Hansen. Computing truncated singular value decomposition least squares solutions by rank revealing QR-factorizations. *SIAM J. Sci. and Stat. Comput.*, 11(3):503–518, 1990.

[5] É. Chouzenoux, S. Moussaoui, J. Idier, and F. Mariette. Efficient maximum entropy reconstruction of nuclear magnetic responance T1-T2 spectra. *IEEE Trans. Signal Processing*, 58(12):6040–6051, Dec 2010.

[6] Å. Ukkelberg, G.H. Sørland, E.D. Hansen, and H.C. Wideøe. ANAHESS, a new second order sum of exponentials fit algorithm, compared to the Tikhonov regularization approach, with NMR applications. *Int. J. Research and Reviews*, 2(3):195–210, Mar 2010.

[7] L. Monteilhet, J.-P. Korb, J. Mitchell, and P.J. McDonald. Observation of exchange of microp-ore water in cement pastes by two-dimensional T_2-T_2 nuclear magnetic resonance relaxometry. *Physical Review E*, 74:061404–1–9, 2006.

[8] D. Byrchenkoff and S. Rodts. Structure of the two-dimensional relaxation spectra seen within the eigenmode pertturbation theory and the two-site exchange model. *J. Magnetic Resonance*, 208(1):4–19, Jan 2011.

[9] J. Granwehr and P.J. Roberts. Inverse Laplace transform of multidimensional relaxation data without non-negativity constraint. *J. Chemcial Theory and Computation*, 8:3473–3482, 2012.

[10] L. Venkataramanan, Y.-Q. Song, and M.D. Hurlimann. Solving Fredholm integrals of the first kind with tensor product structure in 2 and 2.5 dimensions. *IEEE Trans. Signal Processing*, 50(5):1017–1026, May 2002.

[11] E. Chouzenoux, S. Moussaoui, J. Idier, and F. Mariette. Primal-dual interior point optimization for a regularized reconstruction of nmr relaxation time distributions. In *Proc. IEEE Conf. Acoust., Speech & Sig. Proc.*, Vancouver, May 2013.

[12] M. Vetterli, J. Kovačević, and V.K. Goyal. *Foundations of Signal Processing*. Cambridge University Press, 2014.

[13] J.P. Butler, J.A. Reeds, and S.V. Dawson. Estimating solutions of first kind integral equations with nonnegative constraints and optimal smoothing. *SIAM J. Numer. Anal.*, 18(3):381–397, Jun 1981.

[14] M. Gu and S.C. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM J. Sci. Comput.*, 17(4):848–869, 1996.

[15] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[16] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2nd edition, 2003.

[17] J.R. Bunch and L. Kaufman. Some stable methods for calculating inertia and solving symmetric linear systems. *Math. of Computation*, 31(137):163–179, Jan 1977.