UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Dominant Subspace Preconditioning for Kernel Ridge Regression Problems

Christopher R. Anderson

November 2016

CAM Report 16-77

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90095-1555

Abstract

In this paper we describe a subspace based preconditioner for the linear systems of equations arising during the application of the kernel ridge regression technique. Computational results are presented that demonstrate the efficacy of the preconditioner when used in conjunction with the method of preconditioned conjugate gradients.

1 Introduction

Kernel ridge regression is a technique that can be used to construct approximations of the form

$$\phi_{\vec{\alpha}}(\vec{x}) = \sum_{j=1}^{\mathrm{N}} \alpha_j \mathrm{K}(\vec{x}_j, \vec{x}) \tag{1}$$

to functions $\mathcal{F} : \mathrm{V} \to \mathrm{R}$ where V is a normed space. In the approximation K is the kernel operator and $\{\vec{x}_j\}_{j=1}^{\mathrm{N}}$ are samples associated with a training set. A common choice for the kernel operator are Gaussians, e.g. $\mathrm{K}(\vec{x}_j, \vec{x}) = \mathrm{e}^{-||\vec{x}_j - \vec{x}||^2/\sigma^2}$, or more generally functions of $||\vec{x}_j - \vec{x}||$ that have a shape similar to Gaussians. The vector of coefficients of this approximation, $\vec{\alpha}$, are determined as the minimizer of

$$||\sum_{i=1}^{\mathrm{N}} \phi_{\alpha}(\vec{x}_i) - y_i||_2 + \lambda ||\vec{\alpha}||_{\mathrm{M}}, \tag{2}$$

where $y_i = \mathrm{F}(\vec{x}_i)$, $\lambda > 0$ is a regularization parameter, M is the "kernel matrix" whose (i,j)th entry, $m_{i,j}$, is given by $m_{i,j} = \mathrm{K}(\vec{x}_j, \vec{x}_j)$, and $||\vec{\alpha}||_{\mathrm{M}} = < \vec{\alpha}, \mathrm{M}\,\vec{\alpha} >$. The solution to this minimization problem can be obtained as the solution to the linear system

$$[\mathrm{M} + \lambda \mathrm{I}\,]\ \vec{\alpha} = \mathrm{A}\vec{\alpha} = \vec{y} \tag{3}$$

The matrix M of (3) is symmetric, and for commonly used kernels, is positive-definite. However, M can be very close to being singular as training data that are not well separated lead to columns of M that are nearly identical. The inclusion of the regularization parameter λ insures that the matrix A will have a minimal eigenvalue larger than λ and thus the solution to the matrix problem (3) can be reliably computed using standard computational linear algebra methods. Since M is in general a dense matrix, popular choices are direct (non-iterative) methods such as compact Choleksy factorization or, in cases where extra stability of the solution process is required, Gaussian elimination with partial pivoting.

While direct methods are popular there may be applications where an iterative solution of (3) is desired. For example, when the size of the training data is large, the amount of memory required to store the dense matrix may necessitate using distributed clusters of machines to solve (3). With high performance libraries available [1] for the implementation of matrix vector products on distributed architecture machines, implementing an efficient solution technique using iterative methods is likely to be far easier than implementing a direct method solution technique. Another desirable aspect of iterative methods is that the matrix $[\mathrm{M} + \lambda\mathrm{I}\,]$ is not overwritten during the solution process and thus it's possible to utilize iterative methods for different values of λ without having to reconstruct the kernel matrix M.

With the inclusion of the regularization parameter the matrix A of (3) is symmetric positive-definite and the method of Conjugate-Gradients is a natural choice as an iterative method. Unfortunately, the convergence of the Conjugate-Gradient method can be very slow and, consequently one incurs a high computational cost to obtain a solution. In order to improve the convergence behavior one seeks a good preconditioner — a matrix $\tilde{\mathrm{A}}$ that approximates A and whose inverse can be applied to a vector efficiently. Good preconditioners are those that lead to rapidly converging iterative methods when Conjugate-Gradients is applied to the equivalent preconditioned system

$$\tilde{A}^{-1/2}\, A\, \tilde{A}^{-1/2} \alpha = \tilde{A}^{-1/2}\, \vec{y}, \tag{4}$$

The form of the preconditioned system (4) is chosen so that symmetry is preserved, and, although it appears that the computation of $\tilde{A}^{-1/2}$ is required, this is not the case. As discussed in [2, p. 651], when implementing Conjugate-Gradients for the preconditioned system (4) it is possible to rewrite the expressions of the method so that only the application of $\tilde{A}^{-1}$ to a vector is required.

In the following sections we present a preconditioner that is easy to construct and significantly improves the rate of convergence of the method of Conjugate-Gradients when applied to (4). The general idea behind the preconditioner is to use an approximation to A that consists of A projected onto it's "dominant subspace"; a subspace associated with the largest eigenvalues of A. The remarkable fact is that the construction of a good approximation to the dominant subspace that is satisfactory for use as a preconditioner requires just a one or two applications of the kernel matrix M to a collection of vectors with random entries. There is no need to find particularly accurate approximations to the subspace associated with the largest eigenvalues of A. In the following section we present the preconditioner and discuss the motivation for it's use. This section is then followed by a computational example where we demonstrate the utility of subspace preconditioning to improve the convergence rate of Preconditioned Conjugate-Gradients and also present results that demonstrate the effects that parameter choices associated with construction of the preconditioner have on the convergence behavior of the iterative method.

2 Subspace preconditioning

Our proposed preconditioner for (3) is a subspace preconditioner. The primary ingredient of a subspace preconditioner is an $\mathrm{N} \times p$ matrix $\tilde{\mathrm{U}}$ whose p columns are orthonormal, and hence span a p dimensional subspace. An approximation to a matrix A can be constructed that consists of the projection of the linear operator that A represents onto the subspace spanned by the columns of $\tilde{\mathrm{U}}$ and is the identity on the orthogonal complement of that subspace. The matrix representation of this approximation is given by

$$\tilde{\mathrm{A}} = \mathrm{I} - \tilde{\mathrm{U}}\tilde{\mathrm{U}}^T + \tilde{\mathrm{U}} \left[\tilde{\mathrm{U}}^T \mathrm{A} \tilde{\mathrm{U}} \right] \tilde{\mathrm{U}}^T \tag{5}$$

In order to allow for efficient application of the subspace preconditioner (5) and it's inverse to a vector, we transform $\tilde{\mathrm{U}}$ to another $\mathrm{N} \times p$ matrix U, so that the columns of U span the same subspace as the columns of $\tilde{\mathrm{U}}$ but application of the operator A is represented by a diagonal matrix. This transformed matrix is the matrix whose columns are the Ritz vectors associated with the diagonalization of the $p \times p$ matrix $\tilde{\mathrm{U}}^T \mathrm{A} \tilde{\mathrm{U}}$. Specifically, if C is the $p \times p$ matrix of orthonormal eigenvectors of $\tilde{\mathrm{U}}^T \mathrm{A} \tilde{\mathrm{U}}$ so that that $\mathrm{C}^T\, \tilde{\mathrm{U}}^T \mathrm{A} \tilde{\mathrm{U}}\, \mathrm{C} = \mathrm{D}$ where D is a diagonal matrix, then U by $\mathrm{U} = \tilde{\mathrm{U}}\mathrm{C}$. One can verify that $\mathrm{U}^T \mathrm{A} \mathrm{U} = \mathrm{D}$ so the subspace approximation to A based upon the columns of U has the form

$$\tilde{\mathrm{A}} = \mathrm{I} - \mathrm{U}\mathrm{U}^T + \mathrm{U}\left[\mathrm{U}^T\mathrm{A}\mathrm{U}\right]\mathrm{U}^T = \mathrm{I} - \mathrm{U}\mathrm{U}^T + \mathrm{U}\mathrm{D}\mathrm{U}^T \tag{6}$$

The inverse of this subspace approximation is given by

$$\tilde{\mathrm{A}}^{-1} = \mathrm{I} - \mathrm{U}\mathrm{U}^T + \mathrm{U}\mathrm{D}^{-1}\mathrm{U}^T \tag{7}$$

In iterative methods that utilize preconditioners it is only necessary to apply the inverse of the preconditioner to a vector. For subspace preconditioners, this requires the evaluation of

$$\left[\mathrm{I} - \mathrm{U}\mathrm{U}^T + \mathrm{U}\mathrm{D}^{-1}\mathrm{U}^T\right]\vec{w} \tag{8}$$

and is a task that can be efficiently accomplished by storing the matrix U and calling a sequence of BLAS routines for the component matrix vector multiplications.

3 Preconditioning for kernel ridge regression problems

In kernel ridge regression problems $\mathrm{A} = [\mathrm{M} + \lambda\mathrm{I}]$ and since the eigenvectors of $\tilde{\mathrm{U}}^T[\mathrm{M} + \lambda\mathrm{I}]\tilde{\mathrm{U}}$ are identical to the eigenvectors of $\tilde{\mathrm{U}}^T\,\mathrm{M}\,\tilde{\mathrm{U}}$, a subspace preconditioner based upon a subspace spanned by the columns of $\tilde{\mathrm{U}}$ can be expressed in the form

$$\tilde{\mathrm{A}} = \mathrm{I} - \mathrm{U}\mathrm{U}^T + \mathrm{U}\mathrm{D}\mathrm{U}^T \tag{9}$$

where U are the Ritz vectors associated with the diagonalization of the kernel matrix $\tilde{\mathrm{U}}^T\,\mathrm{M}\,\tilde{\mathrm{U}}$ and $\mathrm{D} = \mathrm{diag}(d_i + \lambda)$ with d_i the eigenvalues of $\tilde{\mathrm{U}}^T\,\mathrm{M}\,\tilde{\mathrm{U}}$. It is worth noting that for a given choice of subspace, $\tilde{\mathrm{U}}$, the diagonalization step required to create a useful preconditioner does not depend on the regularization parameter λ. Thus the preconditioner, and, perhaps most importantly, the inverse of the preconditioner, are easily constructed for any value of λ once the initial diagonalization step is done.

The choice of subspace to use as a preconditioner is dictated by the need to reduce the condition number of the preconditioned system (4). Specifically, as indicated by the error bounds for the Conjugate-Gradient method [3, p. 299], if the condition number, $\kappa_2 = \lambda_{max}/\lambda_{min}$, of the linear system is close to 1, then the method will converge rapidly. Therefore, we seek a subspace with the objective of reducing the condition number of the preconditioned system (4) to be close to 1.

The eigenvalues of $\mathrm{A} = [\mathrm{M} + \lambda\mathrm{I}]$ have the from $d_i + \lambda$ where d_i are the eigenvalues of the kernel matrix M. The typical distribution of eigenvalues for M consists of a large number of very small eigenvalues and a modest number of large (or dominant) eigenvalues. The condition number of A is approximately $(\lambda_{max} + \lambda)/\lambda$ where λ_{max} is the largest eigenvalue of M. If the subspace U is taken to be the subspace spanned by the orthonormal eigenvectors of M greater than $\bar{\lambda}$, then the inverse of the preconditioner will be exact on that subspace and the resulting condition number of the preconditioned system (4) will be reduced to $\max(1, \bar{\lambda})/\lambda$. If the dominant eigenvalues of M decay rapidly, then the size of the subspace needed to significantly reduce the condition number of the preconditioned system need not be especially large.

The task of finding the dominant eigenvalues and eigenvectors of a large dense matrix can be computationally expensive, and so instead of using a subspace explicitly formed from the eigenvectors associated with the dominant eigenvalues, we consider using a subspace comprised of vectors that have significant components of the eigenvectors associated with the dominant eigenvalues. In particular, one initially selects a set of p random vectors that have been orthonormalized, and then performs k_{max} steps of orthogonal iteration [2, p. 454] to obtain a subspace that is “rich” in components of the eigenvectors associated with the dominant eigenvalues. In the limit as the number of steps of orthogonal iteration $k_{max} \to \infty$, the orthogonal iteration will converge to the subspace containing the dominant eigenvectors, and the reduction in the condition number will be as described above. For smaller values of k_{max}, one may not see an identical reduction in condition number as when using a subspace of dominant eigenvectors, but as the computational experiments will demonstrate, it can still be a dramatic reduction, even with $k_{max} = 2$.

The algorithm for constructing a subspace preconditioner for the matrix problems of kernel ridge regression is given by

Algorithm 1: Kernel Matrix Subspace Preconditioner Construction

```
input : Kernel matrix M and regularization parameter $\lambda$. $\mathrm{W}_0$, an $\mathrm{N} \times p$ matrix
        whose columns are orthonormalized random vectors.
output: U and $\tilde{\mathrm{D}}^{-1}$, components of the subspace preconditioner inverse
        $\mathrm{I} - \mathrm{U}\mathrm{U}^T + \mathrm{U}\tilde{\mathrm{D}}^{-1}\mathrm{U}^T$

$\mathrm{Z}_0 = \mathrm{W}_0$ for $k \leftarrow 1$ to $k_{max}$ do
    Construct $p \times p$ matrix C and $p \times p$ diagonal matrix $\mathrm{D}_k$ such that
    $[\mathrm{Z}_{k-1}^T \mathrm{M} \mathrm{Z}_{k-1}]\, \mathrm{C} = \mathrm{D}_k\, \mathrm{C}$ ;
    $Z_k \leftarrow \mathrm{Z}_{k-1}\mathrm{C}$;
$\mathrm{U} \leftarrow \mathrm{Z}_{k_{max}}$;
$\{d_i\}_{j=1}^{p} \leftarrow$ diagonal elements of $\mathrm{D}_{k_{max}}$;
$\tilde{\mathrm{D}}^{-1} \leftarrow diag(1/(d_j + \lambda))$;
```

As mentioned previously, for the same kernel matrix M and differing values of λ the only component of this subspace preconditioner that requires modification are the diagonal entries of of $\tilde{\mathrm{D}}^{-1}$; the values $\{d_i\}_{j=1}^{p}$ and the subspace U do change and hence do not need to be recomputed.

4 Computational Results

The sample problem used to demonstrate the effectiveness of dominant subspace preconditioning concerns the construction of an approximation to the mapping associated with the minimum of a normalized quintic polynomial over the interval $[-1, 1]$ as a function of it's coefficients. Specifically the function $\mathcal{F} : \vec{x} \in \mathrm{R}^4 \to \mathrm{R}$ given by

$$\mathcal{F}(\vec{x}) = \min_{s \in [-1,1]} (x_0 \,+\, x_1 x \,+\, x_2 s^2 \,+\, x_3 s^3 \,+\, s^4) \tag{10}$$

where the coefficients are restricted so that $|x_i| \leq 1$.

The kernel used to construct an approximation of this function is the Gaussian kernel $\mathrm{K}(\vec{x}_j, \vec{x}) = \mathrm{e}^{-||\vec{x}_j - \vec{x}||^2/\sigma^2}$. For simplicity the standard Euclidean vector norm, $||\vec{x}||_2$, was used

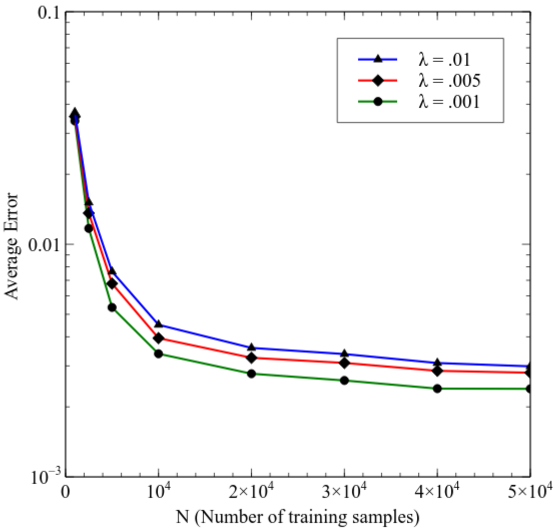

Figure 1: Average maximal error evaluated at 100 samples outside of training data.

as the distance metric. The training data consisted of N vectors of coefficients, $\vec{x} \in R^4$, with each component taken to be the value of a uniformly distributed random number in [-1,1]. The evaluation of the error in the approximation was obtained by evaluating the average maximal error at 100 randomly generated test points that were disjoint from the training set. In all computations reported the value of the hyper-parameter σ was set to $\sigma = 0.5$, and all matrix-vector and matrix-matrix products required by the iterative method were implemented using single-threaded level-1 and leval-2 BLAS routines.

That kernel ridge regression is capable of providing an approximation is revealed by the results in Figure 1, where we present the average maximal error as the number of training points increases. The individual curves correspond to different values of the regularization parameter λ. For smaller numbers of training points the error in the approximation exhibits 1/N convergence, while for larger values, the error saturates. This is not entirely unexpected since we are keeping the hyper-parameter σ fixed.

The behavior of the Conjugate-Gradients and Preconditioned Conjugate Gradients methods when they are used to solve the equations (3) associated with our sample problem is revealed in Figure 2(a) and Figure 2(b). In Figure 2(a) we present the number of iterations required to obtain a relative residual size less than $\epsilon = 1 \times 10^{-4}$ for increasing numbers of training samples and varying values for the regularization parameter λ. Here the subspace size (dimension) was fixed at 400 and two steps of orthogonal iteration were used to create an approximation to the dominant subspace. The results clearly show that a substantial reduction in the number of iterations can be obtained when using a subspace preconditioner. This reduction is to be expected because of the reduction of the condition number associated with the preconditioned system. The number of iterations still increases with the number of training samples, but this is also expected, as the components of the solution that are added when the number of training samples increases are generally associated with the small eigenvalues; components that are not strongly effected when using dominant subspace preconditioning. Since the use of a dominant subspace preconditioner requires extra computational work per iteration, the number of iterations required to obtain convergence may be misleading. However, as revealed by the solution time comparison results given in Figure 2(b), the utility of using dominant subspace preconditioning is also reflected in substantially reduced computation times. In the reporting of the solution time, the solution time is taken to be the sum of the computational time spent carrying out the iteration and the initial time required to create the approximate dominant subspace U.

The results in Figure 2 where created using a dominant subspace dimension fixed at 400 and the number of orthogonal iterations fixed at two. The behavior of the preconditioned iterative method with respect to changing the subspace size is shown in Figure 3(a) and changes due to the number of orthogonal iterations used to construct the dominant subspace are shown in Figure 3(b). In these cases the value of the regularization parameter λ was fixed at .005. From the results in Figure 3(a), we observe that increasing the subspace size always reduces the time to compute the solution, but there is a notable trend of diminishing returns; the extra benefit of using a larger subspace diminishes as the size of the subspace increases. The results in Figure 3(b) demonstrate the somewhat remarkable property that very few orthogonal iterations are required to generate a dominant subspace approximation that is effective as a preconditioner. Using just two or three orthogonal iterations creates a dominant subspace that reduces the number of iterations to that which is obtained with a more accurate approximation to the dominant subspace. It is also worth noting that for this problem the reduction in iteration count does not decrease monotonically with an increasingly accurate dominant subspace approximation. We interpret this behavior as an indication that it can be useful to have remnants of the components of the solution associated with smaller eigenvalues in the dominant subspace approximation. This behavior is another example that reducing the condition number of the preconditioned system isn't the only factor that contributes to efficient iterative methods.

5 Conclusion

In this paper we've described an effective preconditioner for the iterative solution of the matrix problems that arise from applications of kernel ridge regression. The preconditioner is a subspace preconditioner where the subspace is that associated with the dominant eigenvalues. The success of this preconditioner depends on the observation that the number of algebraically large eigenvalues of the matrix problem of kernel ridge regression a typically a small fraction of the total number of eigenvalues, and so it's possible to remove the adverse effects of the dominant eigenvectors on the iterative process using a subspace that isn't of too high dimension. Moreover, approximations to the dominant subspace created with two or three steps of orthogonal iteration lead to a preconditioner that is as effective as that associated using an accurate approximation to the dominant subspace. An effective subspace preconditioner is therefore easy to construct and does not require an large amount of computational work.

For small to modest size problems, direct methods work quite satisfactorily and are likely to be the method of choice. However, for larger problems, a preconditioned iterative

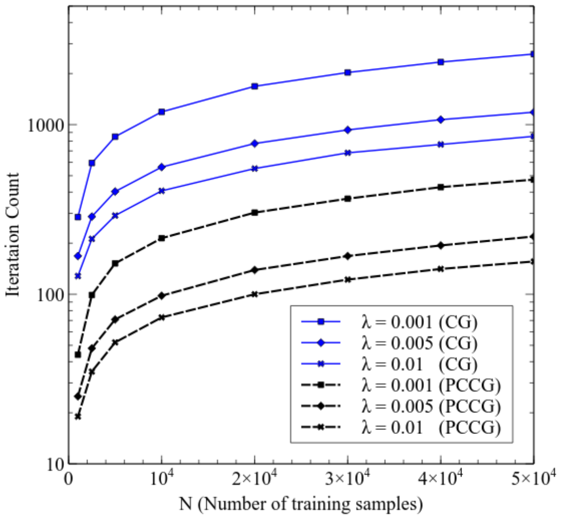

Figure 2(a)
Iteration count vs N for varying
regularization parameter.

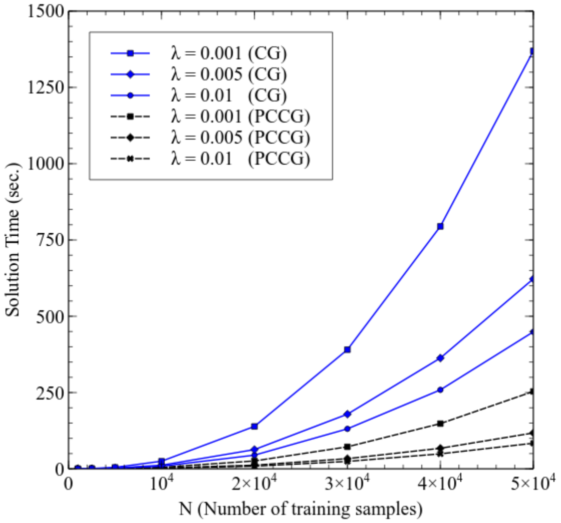

Figure 2(b)
Solution time vs. N for varying
regularization parameter.

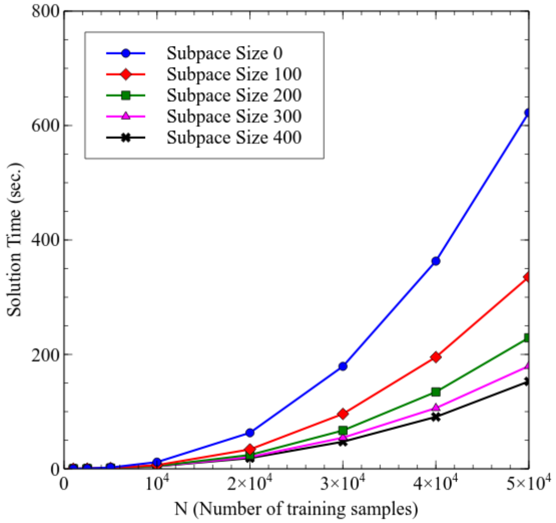

Figure 3(a)
Solution time vs N for varying
subspace sizes

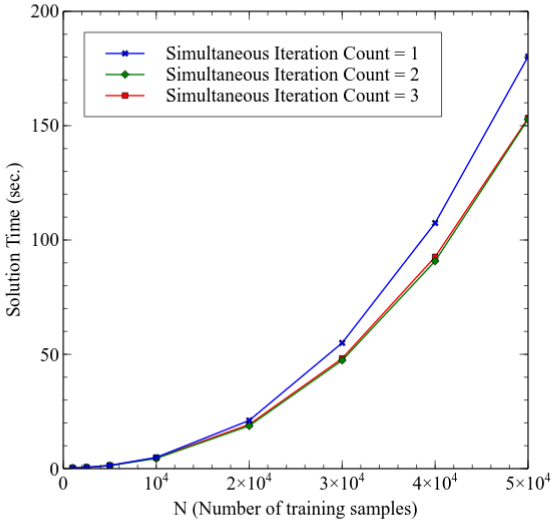

Figure 3(b)
Solution time vs. N for varying
orthogonal iteration count.

method of the type presented here may be an attractive option. Additionally, there are several parameters of the preconditioned method that can be varied and it's not too difficult to imagine that using problem specific parameter optimization and parallel matrix-vector operations, one could substantially reduce the size of the linear system where the resulting iterative method becomes competitive with direct methods of solution.

Bibliography

[1] S. Balay, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, and H. Zhang. PETSc users manual. *Argonne National Laboratory, Tech. Rep. ANL-95/11-Revision*, 2(5), 2004.

[2] G.H. Golub and C.F. Van Loan. *Matrix Computations 4th Edition.* Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 2013.

[3] Lloyd N. Lloyd Nicholas Trefethen and David Bau. *Numerical linear algebra.* Society for Industrial and Applied Mathematics, Philadelphia, 1997.