
Balaji Vasanth Srinivasan
Fast matrix-vector product based FGMRES for kernel machines

3368 AV Williams Bldg
University of Maryland
College Park
MD 20742
balajiv@umiacs.umd.edu
Ramani Duraiswami
Nail Gumerov

Algorithms based on kernel methods play a central role in statistical machine learning. At their core are a number of linear algebra operations on matrices of kernel functions which take as arguments the training and testing data. A kernel function $\Phi(x_i, x_j)$ generalizes the notion of the similarity between a test and training point. Given a set of data points, $\mathbf{X} = \{x_1, x_2, \dots, x_N\}, x_i \in R^d$, the kernel matrix is given by,

$$\mathbf{K} = \begin{pmatrix} k_{11} & \dots & k_{1N} \\ \vdots & \ddots & \vdots \\ k_{N1} & \dots & k_{NN} \end{pmatrix}, \quad (1)$$

where $k_{ij} = \Phi(x_i, x_j)$. Kernel matrices are symmetric and satisfy the Mercer conditions, $a^T \mathbf{K} a \geq 0$, for any a ; and hence \mathbf{K} is positive semi-definite. In many algorithms, it might be necessary to solve a linear system with the matrix \mathbf{K} ,

$$\mathbf{K}x = b. \quad (2)$$

Iterative Krylov approaches are often used with fast matrix vector products for efficient solution of such systems (de Freitas et al., 2005).

When the underlying system is ill-conditioned, there is a degradation of the performance of iterative approaches, necessitating the use of a preconditioner for the Krylov iterations. Popular preconditioners like Jacobi, SSOR, ILU can be used to improve the convergence, however, these preconditioners have an $O(N^2)$ space and computation requirement for dense matrices, which would ruin any advantage gained by the fast matrix-vector products. The preconditioner must have a representation that allows for a fast matrix-vector product just like the kernel matrix. We propose a *Tikhonov regularized kernel matrix* solved with a truncated conjugate gradient algorithm as a preconditioner for the kernel matrix, which can be accelerated using the aforementioned approaches. Because the symmetry of this preconditioner cannot be guaranteed, we use a flexible GMRES (Saad, 1993) algorithm.

A good preconditioner will improve the convergence of the iterative approach at the expense of an increased cost per iteration. The convergence with proposed preconditioner is shown to be an order of magnitude better than the unpreconditioned approach. But, for a preconditioner to be useful, the total time taken by the preconditioned approach should be less than the unpreconditioned approach. This is achieved by using the fast matrix-vector product algorithms, resulting in a computationally efficient solver with faster convergence. We use two classes of fast matrix vector products, approximation-based acceleration (eg. FIGTREE (Morariu et al., 2008) for Gaussian kernel) and GPU-based parallelization (Srinivasan et al, 2009). Each of these two approaches have their own advantages and disadvantages. We propose a strategy to choose the appropriate acceleration in the light of the desired accuracy.

The performance is further illustrated in popular learning approaches namely, radial basis function interpolation, Gaussian process regression and kernel classification. There is an improvement of upto $\sim 8X$ in the number of iterations to converge and $\sim 3.5X$ in the total time taken compared to a conjugate gradient based approach. The core preconditioning strategy proposed here will soon be released as an open source package.

Bibliography

- [1] N. DE FREITAS, Y. WANG, M. MAHDAVIANI, AND D. LANG, *Fast Krylov methods for n-body learning*, in Advances in Neural Information Processing Systems, 2005.
- [2] V. MORARIU, B. SRINIVASAN, V. RAYKAR, R. DURAIWAMI, AND L. DAVIS, *Automatic online tuning for fast Gaussian summation*, in Advances in Neural Information Processing Systems, 2008. Available at <http://sourceforge.net/projects/figtree/>.
- [3] Y. SAAD, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.
- [4] B. SRINIVASAN AND R. DURAIWAMI, *Scaling kernel machine learning algorithm via the use of GPUs*, in GPU Technology Conference, NVIDIA Research Summit. Available at <http://www.umiacs.umd.edu/balajiv/GPUML.htm>.