RnD Project Report

Aditya Golatkar - 14B030009
Rudrajit Das - 140020012
Supervisor - Prof. Suyash Awate

(I) Using the improved Nystrom Technique by Drineas and Mahoney

By utilizing some properties of the Gram matrix(G) like Symmetric Positive Definiteness and Low Rank structure, and using their algorithm, we can decompose G (of size N X N) into a low rank matrix of rank k in just O(N) time, which can be made arbitrarily close to the best possible k-rank approximation of G (using the k largest singular values) by a suitable choice of parameters. This algorithm decomposes G into a nice form, which we have utilized to obtain the eigen vectors and eigen values very efficiently.

Nystrom Method in Brief -

- This algorithm, when given as input a SPSD matrix $G \in \mathrm{R}^{n \times n}$, computes a low-rank approximation to G of the form

$$\widetilde{G_k} = CW_k^+C^T$$

- where $C \in \mathrm{R}^{n \times c}$ is a matrix formed by randomly choosing a small number c of columns (and thus rows) of G and $W_k \in \mathrm{R}^{c \times c}$ is the best rank-k approximation to W, the matrix formed by the intersection between those c columns of G and the corresponding c rows of G.
- The columns are chosen in c independent random trials (and thus with replacement) according to a judiciously-chosen and data-dependent non-uniform probability distribution.
- Let $\| \quad \|_2$ and $\| \quad \|_F$ denote the spectral norm and the Frobenius norm of a matrix respectively, and let G_k be the best rank-k approximation to G.
- It is then shown that under appropriate assumptions,

$$\left\|G - CW_k^+C^T\right\|_\xi \leq \|G - G_k\|_\xi + \epsilon \sum_{i=1}^{n} G_{ii}^2,$$

in both expectation and with high probability, for both ξ = 2, F, and for all k : 0 ≤ k ≤ rank(W). This approximation can be computed in O(N) space and time after two passes over the data from external storage.

The main theorem of Nystrom method in detail:

Suppose G is a nxn SPD matrix, let k ≤ c be a rank parameter, and let $\hat{G}_k = CW_k^+C^T$ *be constructed from the Nystrom Algorithm by sampling c columns of G with probabilities* $\{p_i\}_{i=1}^n$ *such that:*

$$p_i = G_{ii}^2 / \sum_{i=1}^{n} G_{ii}^2$$

Let $r = rank(W)$ *and let* G_k *be the best rank k approximation to G. In addition let* $\varepsilon \geq 0$ *and* $\eta = 1 + \sqrt{8 * log(1/\delta)}$.

If $c \geq 64k\eta^2/\varepsilon^4$ *the with probability at least* $1 - \delta$

$$||G - \hat{G}_k||_F \leq ||G - G_k||_F + \varepsilon \sum_{i=1}^{n} G_{ii}^2$$

For our application, epsilon in the range of $[1, 2]$ *is good enough in terms of accuracy.*

The $p_i's$ are obtained by minimizing an objective function (which turns out to be the variance of the estimator) under the constraint $\sum_i pi = 1$, using the method of Lagrange Multipliers.

A highly abridged (and hand-waving!) proof -

From the previous slide : $\widehat{G}_k = CW_k^+C^T$ & $G = X^TX$

After some painstaking algebra : $||G - \widehat{G_k}||_F \leq ||G - G_k||_F + ||AA - BB||_F$

where $A = XX^T$ & $B = C_X C_X{}^T$

To get C_X *choose (w.p* p_{i_c}*) only* c *scaled columns from* X

Let E *be the event that* $||AA - BB||_F \leq \frac{2\eta}{\sqrt{c}}||A||_F{}^2$

We need to show that $P[E] \geq 1 - \delta$, *where* δ *can be chosen to be arbitrarily small.*

After some more work : $E[\ ||AA - BB||_F\ ] \leq \frac{2\eta}{\sqrt{c}}||A||_F{}^2$

Let's define $F(i_1 \ldots . i_C) = ||AA - BB||_F$, c *terms are chosen independently with* p_{i_c}

We can show that $|F(i_1 \ldots . i_k \ldots . i_c) - F(i_1 .. i_k^1 \ldots . i_c)| \leq \frac{1}{\beta C}||A||_F{}^2 = \Delta$. *This beckons for using BDC!*

Let $\gamma = \sqrt{2c\log(1/\delta)}\,\Delta$ *and consider the associated Doob's Martingale*

Invoking Azuma Hoeffding inequality, we get : $P[||AA - BB||_F \leq \frac{2\eta}{\sqrt{c}}||A||_F{}^2 + \gamma] \geq 1 - \delta$

$||A||_F{}^2 = (Trace(G^2)) = \sum_i G_{ii}{}^2$

After some more laborious work, the concentration result that we saw in the previous slide follows from the concentration result discussed in the previous point. We will skip those gory details and move on to the results!

Algorithm for obtaining $\widehat{G}_k$ is the following:

Data : $n \times n$ Gram matrix G, $\{p_i\}_{i=1}^n$ such that $\sum_{i=1}^n p_i = 1$, $c \leq n$, and $k \leq c$.

Result : $n \times n$ matrix $\tilde{G}$.

- Pick c columns of G in i.i.d. trials, with replacement and with respect to the probabilities $\{p_i\}_{i=1}^n$; let I be the set of indices of the sampled columns.
- Scale each sampled column (whose index is $i \in I$) by dividing its elements by $\sqrt{cp_i}$; let C be the $n \times c$ matrix containing the sampled columns rescaled in this manner.
- Let W be the $c \times c$ submatrix of G whose entries are $G_{ij}/(c\sqrt{p_i p_j}), i \in I, j \in I$.
- Compute W_k, the best rank-k approximation to W.
- Return $\tilde{G}_k = CW_k^+ C^T$.

This sampling algorithm has a time complexity of O(2N). Note that this time complexity is accounted for in the total complexity formula mentioned on the page below. We are skipping the proof of this randomised algorithm here because it based on 9 lemmas, the results of another paper which mainly depend on Concentration inequalities (not of interest to us in this course) and some highly non-trivial linear algebra.

Our Modification :

After we get $\widehat{G}_k$ using the improved Nystrom algorithm, we apply the following three steps to obtain the eigen vectors with the k largest eigen values efficiently.

EVD -> SVD -> EVD to get better complexity

$\widehat{G}_k = CW_k^+C^T$	W_k^+ *is c x c with rank k*
$\widehat{G}_k = CU\Lambda U^T C^T$	*Eigen Value Decomposition of Symmetric* W_k^+
$\widehat{G}_k = (CU)\Lambda(U^T C^T)$	*CU is n x k with rank k*
$\widehat{G}_k = U_1 S_1 V_1^T \Lambda V_1 S_1 U_1^T$	*Singular Value Decomposition of CU*
$\widehat{G}_k = U_1 M U_1^T$	*Let* $M = S_1 V_1^T \Lambda V_1 S_1$
$\widehat{G}_k = U_1 U_M \Delta U_M^T U_1^T$	*M is k x k with rank k*

$U_1 U_M$ *is the Eigen Vector matrix of* $\widehat{G}_k$
Δ *is the Eigen Value matrix of* $\widehat{G}_k$

The complexity of our algorithm $= O(kc^2 + nk^2 + k^3 + 2n) = O(nk^2)$
Complexity of obtaining top k Eigen Vectors of G $= O(n^2k)$

Computing the top k eigen vectors of G directly is computationally expensive requiring $O(n^2k)$ time. We can reduce this time complexity by exploiting the structure of $\widehat{G}_k$ as follows:

First compute the EVD of W_k , this computation has a time complexity of $O(kc^2)$ as the dimension of W_k^+ is c x c and the rank is k.

Next compute the SVD of CU , this computation has a time complexity of $O(nk^2)$ as the dimension of CU is n x k and its rank is k

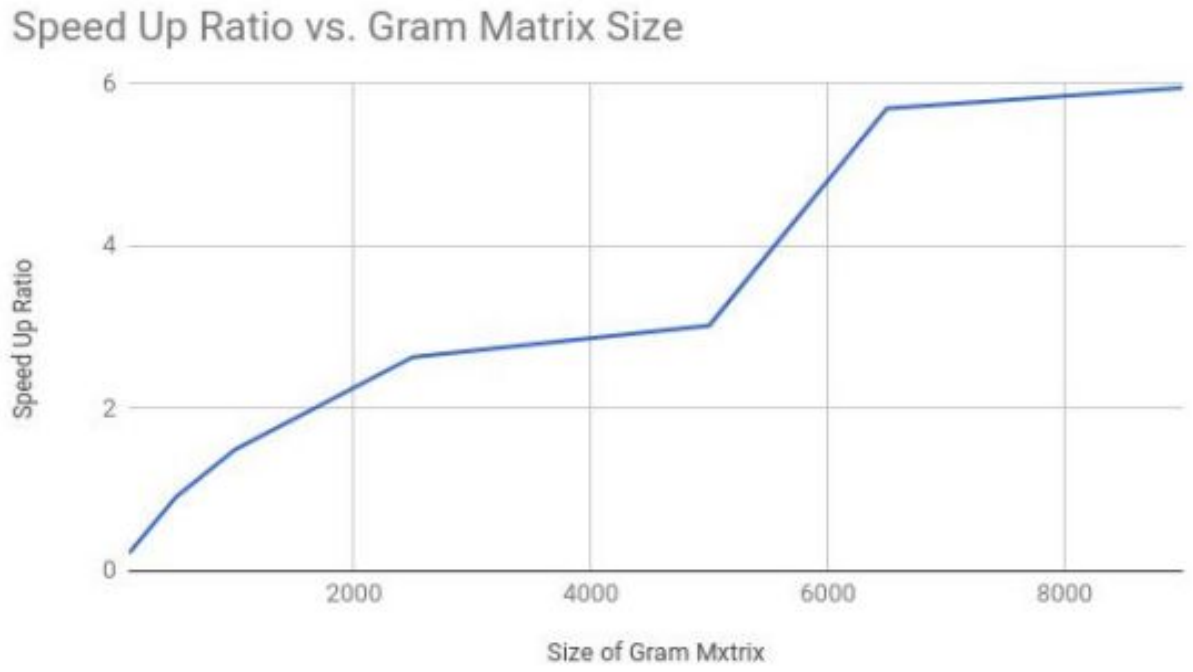
Finally, the last EVD of M has a time complexity of $O(k^3)$ as the dimension of M is k x k and its rank is also K.

Adding the three terms we can see that the second term is the largest

and hence the complexity of our algorithm is $O(nk^2)$ which is lesser than the time complexity of EVD of the actual Gram Matrix which is $O(kn^2)$. Thus, our algorithm is a sub-linear improvement over the naive method.

Simulations + Comparison with MATLAB's performance

Gram Matrix Size	Rank of Data	Time(Our Algo)	Time(MATLAB's eigs())	Speed of our Algo wrt Matlab	EigVec diff norm	EigVal diff(all EV)
100X100	20	0.009	0.002	0.2222222222	order e-11	0
500X500	30	0.012	0.011	0.9166666667	order e-11	0
1000X1000	40	0.04	0.06	1.5	order e-11	0
2500X2500	50	0.11	0.29	2.636363636	order e-11	0
5000X5000	65	0.42	1.27	3.023809524	order e-11	0
6500x6500	80	0.43	2.45	5.697674419	order e-11	0
9000x9000	80	0.76	4.52	5.947368421	order e-11	0



The above graph of Speed Up Ratio vs. Gram Matrix Size is sub-linear and hence consistent with the claim made in the previous page.