OPEN ACCESS

International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-7, Issue-6, March 2019

An Efficient Approach for Iterative Learning Algorithms

G. Abinaya, Anirudh Sundararaman, R. Ashwin, Ravi Teja, Vinay Motghare

***Abstract:** In this paper, a framework which takes into account machine learning for the analysis of massive datasets is proposed. The framework maps the algorithms to the respective platform so as to extract maximum resource efficiency. In addition, the framework takes into account a data projection technique called as Elastic Dictionary to form sparse representation of the underlying data. By this way, the resource efficiency is optimized leading to reduction in the cost associated with the performance. The framework represents a model and shows the performance metrics in accordance with their respective runtime and storage. An additional application program interface takes into account the applicability of the framework to the underlying platform or datasets. The framework is based on the union of both the content and platform aware methodologies so as to make the machine learning algorithms to utilize the resources efficiently.*

***Index terms:** Cholesky Factorization, Elastic Dictionary, Orthogonal matching pursuit algorithm, Sparse Approximation*

I. INTRODUCTION

Learning algorithms tend to use the dependencies and patterns which are present across the datasets. In these cases the objective function is solved by iteratively updating the parameters of interests which may require matrix multiplication that involves gram matrix but in cases when data cannot fit on a single node and therefore can require a spread architecture, repetition based updation exhibit large computation and communication costs. A sustainable computing environment requires efficient utilization of resources. The most challenging environments are those which take into account iterative updates on non-sparse (dense) datasets. This conditions involve algorithms put into place for to solve regression, deep learning or SVM used in learning applications. Example of dense datasets may include video and image datasets which stands for the substantial amount of the generated data of the modern society.

In such challenging scenarios there constitutes two disjoint prior works which have taken into account efficient machine learning. (i) Methods which are content aware in the field of statistics and machine learning which factorize the corresponding computational raw data into a reduced dimensional representation of the respected datasets [9] and (ii) Techniques which are platform aware which assist to depict algorithms to a respective platform [1]. There exists machine learning methods that work on the sparsity which exists of the correlated matrix data representation but they tend to be inefficient for dense data which include Pregel [12] or Graphlab [11].

In this paper a framework is introduced which is a machine learning based framework for performance optimization which takes into account both platform and content aware techniques. The new viewing involving projection of data can play a vital role in the mapping onto modern multi-core and distributed architectures drives the efficiency. We observe that degree of freedom to factorize datasets to reduced dimensions involving the same error (approximated error) can be used to map computations with respect to the underlying platform. Elastic dictionary consists of parameters which can be tuned to project divergent subspaces also known as elastic dictionaries with some approximation error gives the dimensionality reduction methodology. The preprocessing phase involves the formation of the dictionary and the dictionary parameters which can be tuned with respect to the underlying platform by subsampling of the data. The information is then utilized for processing the entire large scale dataset.

The respective implementations of this paper can be given as:(i) Implementation of a framework which supports a range of applications in the machine learning domain are dependent upon the repetitive optimization over condensed datasets to achieve convergence. (ii) A performance evaluation metric for the framework which is used for the optimization in terms of runtime, energy and memory performance for the execution of a machine learning algorithms on a given system. (iii) Implementation of extensible dictionary which is a parametric data projection technique which can produce many possible lower dimensional subspaces for a given approximation error. To minimize the cost associated with the performance, the parameters in the extensible dictionary are tailored with respect to the corresponding platform. (iv) A mapping methodology to implement optimized execution of the machine learning techniques over the underlying architecture which takes into account load balancing and minimization of communication concurrently.

Revised Manuscript Received on 30 March 2019.
* Correspondence Author

G. Abinaya, Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India

Anirudh Sundararaman, B.Tech Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India

R. Ashwin, B.Tech Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India

Ravi Teja, B.Tech Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India

Vinay Motghare, B.Tech Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India

The performance hurdles corresponding to the communication and storage theoretical bounds is achieved by the mapping technique. (v) To involve quick customization of the framework to a given specific platform, is performed by automating the framework by open source application program interfaces built upon the MPI message passing interface. De-noising and super resolution exhibit a considerable improvement in costs corresponding to runtime, memory and energy constraints.

II. PRIOR WORK

Therefore, the framework is based on the union of both the content and platform aware methodologies so as to make the machine learning algorithms to utilize the resources efficiently. Prior works listed in this section are associated to the framework.

Techniques to Lessen Dimensions: Matrices involving a MxN dimensionality matrix computation methods using Singular value decomposition and principal component analysis becomes redundant on larger datasets as they involve $O(M^2N)$ complexities which also involves large constant parameters. Column Subset Selection (CSS) are randomized algorithms which have been developed to address these challenges [2]. In this methodology data is projected upon lower dimensional subspace. A subset of data columns is taken into account for learning the projection basis which is also known as dictionary. The randomized CSS techniques are applicable to large scale data as they scalable [10]. Methods such as Farahat [4] are adaptive and lower the dictionary size to achieve an approximation error. These methods are not applicable to larger datasets as they NxNmatrix which is correlational. There's one more method which is memory efficient and with linear runtime called Oasis [14] which is an adaptive methodology which takes into account the appropriate columns to include to the library. These techniques can be used and replace the extensible dictionary method being used here within the framework. In this paper performance is evaluated based on elastic dictionary.

Techniques that are Content Aware: For accelerating some machine learning algorithms previous work have exhibited the use of importance data factorization methods. But these techniques are however better at some targeted specific machine learning algorithms but are not generic in nature unlike the proposed framework. The proposed framework works upon generic learning algorithms. An earlier method called Rankmap highlights the effectiveness of the data transformation techniques and also works upon generic learning algorithms but the system in not platform aware unlike our framework. In this paper we use Rankmap [13] as a comparison basis. There exists one another method called Stochastic Gradient Descent [17] which does not take into account efficient memory usage and the results are not assured.

Techniques that are Platform Aware: There exists an extensive amount of previous research concerned with the depiction of the ML algorithmic techniques on to dynamic platforms which efficiently run the algorithms by mapping them on to distributed architectures. It does not involve changes to the underlying data configuration [1,15]. Data provided by the respective application present is worked upon instead which is not dependent upon the corresponding architecture.

III. FRAMEWORK

The framework provided here works upon large scale datasets or Gram matrices and facilitates iterative learning algorithms to run efficiently upon this massive and dense data. There exists many machine learning algorithms which operate upon the relations existing upon the present data samples such as Power method to evaluate PCA [6] or methods for solving SVM [5]. All these methods have an operational cost associated with them during an iterative update which arises majorly form the multiplication of the Gram matrix. A gram matrix is given as the multiplication of the transpose of a matrix to the matrix itself and can be given as $Z = B^T B$where B can be termed as the given data matrix. While operating on larger and denser datasets an amend tends have a lot of operational cost due to the huge number message passing mechanism amongst the participating nodes.An overall flow regarding the framework is given in Fig. 1. The framework relies on the fact that machine learning applications are open to dynamic solution outputs which offers a path to trade accurate results with optimized utilisation of the resources [2,10]. In this paper we also introduce a data transformation method called Elastic dictionary. Elastic dictionary is used to compute a reductional dimensional matrix A which is a dictionary matrix and a sparse coefficient matrix E so that

$$min||E||_0 \ \ s.t \ ||B - AE||_{F1} \leq \varepsilon_0 ||B||_{F1} \qquad \text{(i)}$$

such that A is MxI,E is $IxN$and $I \ll N.\varepsilon$ is a parameter which stands for approximated error $||E||_0$ depicts the non zeros in E and $||.||_{F1}$stands for Frobeniuus norms. Extensible dictionary operates to create a computation so that the continuous updation over the units being transformed which is, $(AE)^T AE$ becomes much more efficient compared to $B^T B$ which all happens during the preprocessing phase. By making dynamic changes in the I which is the size related to the dictionary we can achieve sparser E's. Therefore the technique is to control the redundant terms in the size I associated with the dictionary which can be taken into account.

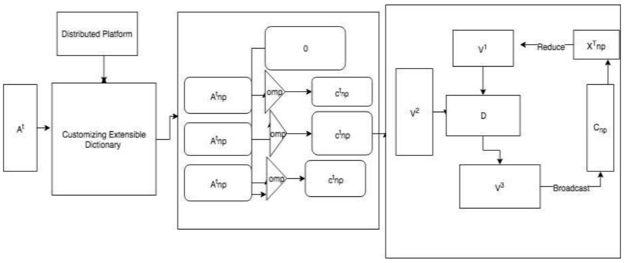

Fig. 1: Flow of the proposed framework

In this paper we show that the dictionary size affects the communication cost and there exists a tradeoff relationship between the communication overhead and the and the present number of non zeroes in the data to be projected. So we propose a distributed computing model which acts iteratively upon DC. In this paper we put forward metrics which determines the operational cost for our model in terms of runtime, energy and memory constraints. We also establish the technique to make dynamic changes in the extensible dictionary parameters to lower the operational costs and we provide evaluations for the same.

IV. FRAMEWORK TRANSFORMATION

The framework projection method is shown by Algorithm 1. The initial step involves randomly sampling the columns associated with B so as to create dictionary matrix A. When the matrix A is achieved equation (i) turns into a problem associated with sparse approximation. The individual columns of E denoted by $c_k$can be defined as a sparse approximation of the individual columns of the individual columns of the matrix B denoted by a_i constrained to the matrix D and the projection error given as ε. The next step involves solving the equation (i). To solve the equation a sparse routine called Orthogonal Matching Pursuit algorithm [16] so as to find the matrix E such that the criteria involving approximation error is fulfilled. This takes place when the number of spans in the matrix D is close to similar to that of the matrix A which takes place after enough data sampling. Noteworthy is that for platform aware methods sparse approximation is never used. The same projection error is achieved as least squares approaches when the approximation error is initialized with zero ($\varepsilon = 0$).

Algorithm 1: Elastic dictionary Transformation

Input: A data matrix $A \epsilon R^{MxN}$, number of columns to be selected L,number of processors N_P,error tolerance ε.

Output: A sparse matrix $C \epsilon R^{LxN}$,a dictionary $D \epsilon R^{MxL}$such that$||B - AE||_F \leq \varepsilon ||B||_{F1}$.

```
0. $id$= $0^1$forms a subselection of dices.
1. $id = k\ loads\ A = B(I_L)$
2. $id = k\ loads\ A_i = A(:, iN/N_P : (i+1)N/N_P)$
    3. $id = k\ applies\ OMP\ to\ solve\ a_i$
            $= AE_i\ for\ tolerance\ error\ \varepsilon$
        3.0. $Initialize\ p = ai, \beta = \emptyset$
        $while\ ||r2|| < \varepsilon ||a_i||_2\ do$
            3.1. $t = argmax_j |d_j . p|$
            3.2. $\beta = (\beta, t)$
            3.3. $y = D_\varphi^+ a_i$
            3.4. $p = a_i - D_\phi y$
        $end\ while$
```

To meet the criteria for the approximation error and sparsity selecting appropriate number of columns is important. Orthogonal Matching Pursuit (OMP) meets the error criteria when matrix A is fully ranked when $L > M$. In the subspace sampling corresponding to the matrix rank we need to find the bounds of L. It has been established that $L \leq \Omega(klogk/(1-\delta)^2)$ which is the number of columns by sub sampling them an error equal to the $1/\delta$ of the norm with an approximated data matrix involving k dimensions is gained. So to vary the degree of sparsity of matrix E we can obtain it by an increased L that is the size corresponding to the dictionary. This characteristic has been taken into account to tune the parameters of the extensible dictionary to lower the performance cost associated with the iterative updates so as it is platform aware.

Analysis of Complexity: Primary complexity arises from executing OMP with the help of Cholesky factorization. Nnz(c) is considered as no of 0's and the complexity of the upper bound is $O(LMN + L^2 nnz(C))$where C is computed separately. The complexity of OMP decreases to$O(\frac{N}{Np}\ (LMN + L^2\ \frac{nnz(C)}{Np}))$where Np is considered as number of cores which process parallel.

Algorithm 2: Gram Matrix Updation and Distribution.

Input: Matrices $y_{Nx1}, A_{MxI}, E_{IxN}$.

Output:$E^T A^T AE_x$

```
0.0 $id$ = k loads $E_i$= $E(:, iN/N_p : (i+1)N/N_P)$.
0.1 $id$ =$k\ loads\ y_i = y(iN/N_p : (i+1)N/N_p)$
1.$id$=$k\ computes\ v_i^1 = E_i y_i, v_i^1\ is\ a\ IxN\ vector$.
    Case 0: I > M
    2. $id = k.loads\ A$
    3.$id = k\ computes\ v_i^2 = Av_i^1, v_i^2\ is\ a\ M \times 1\ matrix$.
    4. Vectors $v_i^2 will\ be\ reduced\ in\ id = 0$.
    5. id = 0 computes $v^2 = S \sum_i^M v_i^2, v^2 = AE_x$
    6. id = 0 implements $v^2$to all other processors.
    7. id = i computes$E_i^T (A^T v^2)$.
    Case 1: L ≤ M
    2. id = 0 loads A.
    3. Vectors $v_i^1\ will\ be\ reduced\ in\ id = 0$
    4. id = 0 computes $v^2 = D(\sum_i^M v_i^1)$ ; $v^2 = AE_x$; is an M×1 vector.
    5. id = 0 computes $v^3 = A^T v^2$ ; $v^3 = A^T AE_x$ is an Lx1 matrix.
    6. id = 0 sends $v^3$to all other processors.
    7. id = i involves $E_i^T v^3$.
```

V. MODEL DESCRIPTION

We can depict from the 2nd Algorithm that $(AE)^T\ AEx = B^T\ Bx$ where x is N*1 vector. This frames the proposed data to perform update along with distributed computing model. Two different approaches can be followed depending on L>M or the opposite. In case of L>M it is considered as case 0 where matrix A is replicated in all process and redundant multiplication is supposed to be done to reduce communication $(AE)^T$.

The redundant multiplication $A^t v^2$is supposed to be done to reduce communication. In case of L<M computation computing is done by 0.

Arithmetic Bounds Cost is proportional to the quantity of operations $(Ac)^T\ AEx$.The quantity of operations in serial is$2(ML + \frac{nnz(c)}{p})$ where nnz(c) is considered as non zero with negligible cost of addition with Np<<L.

Messaging Bounds: The communication overhead of 2nd Algorithm is used to reduce and broadcast activities. All processors send messages corresponding to M words in case 0 and also step 4 and in case of step 0 and processor 6 the messages are then sent back with M words to corresponding processors. 2 × min(L, M) words are communicated simultaneously in total. The computational model achieves optical communication by exploiting numerical algebra. The communication- optimal parallel recursive rectangular matrix targets the problem where Z= XY and X, Y and belong to dimensional matrix(d1,d2 and d3)where (d1<=d2<=d3), if 2 d3 d2 > NP the communication lower bound is Ω(d1d2).On substituting d1 as 1 and d2 as min(M,L) we get words being distributed to min(M,L).

Performance: The overall $ML + \frac{nn(c)}{Np} + min(M,L)NP\ R_{b2f}^{time}$ is exactly proportional to runtime, where $R^{time}$is words per flops. The two terms at beginning shows computational operation and final terms shows adjusted communication overhead. The model gives a fair runtime approximation. Overall running time is affected by memory hierarchy and many other The ExD results in memory saving due to sparsifying effect. The memory footprint in both algorithm is bounded by $ML + \frac{N}{Np} + nnz(c)$.

ELASTIC DICTIONARY PLATFORM SPECIFIC CONFIGURATION

Algorithms which operate on the Gram matrices are optimized so as to limit the cost associated with their execution. To limit the cost the term (L, nnz(E)) needs to be reduced by choosing the appropriate L value with respect to the platform. The method involves computation only on a section of a matrix A by using nnz(E) function with respect to L. The degree of sparsity can be increased in E with a larger L value. The cost can be optimized if α(L, A, ε) undergoes a characterization.

VI. EVALUATION

The framework is implemented by using eigen libraries for carrying out computations involving linear algebra. In C++ a message passing system has been used. The inputs for the application program interface includes approximation error ε a data matrix A and the function which executes upon Gram matrix. In each continuous execution stage the procedure involves to select subsets, denoted by B_b,which is selected in a random manner from rows of B. A single execution step involves a subset $B_b$which is taken from rows of the matrix of B at random. It involves computation using $B_b^T B_x$ in place of $B^T Bx$. The model is operated by comparing the assumed running time trend with the corresponding precise measurement one. Fig 2 depicts the result for execution over a Gram matrix and for only one iteration, i.e., $(AE)^T AX$. The average is taken of the all iterations involved. The given tests are performed on distributed architectures.

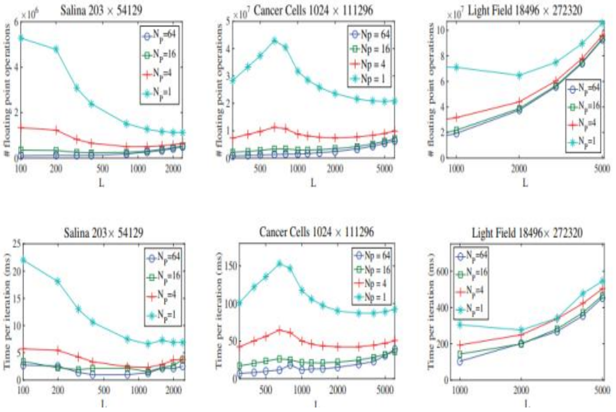

Fig. 2: Actual measured runtime vs runtime performance

VII. CONCLUSION

A framework which takes into account machine learning for the analysis of massive datasets has been proposed. The framework takes into account a data projection technique which is Elastic Dictionary which helps to form sparse representation of the underlying data so as to optimize the resource efficiency and so as to reduce the cost associated with the performance. An application programme interface has been provided which confirms the applicability of learning algorithms to respective platforms and datasets. We show that by tuning of parameters associated with Elastic Dictionary in terms of the underlying platform we can achieve ideal performance

REFERENCES

1 Demmel J, Eliahu D, Fox A, Lipshitz B, and Spillinger O,. “Communication optimal parallel recursive rectangular matrix multiplication”. IPDPS’13.
2 Drineas P ,and Mahoney M. “On the Nystrom method for approximating a gram matrix for improved kernel-based learning”. JMLR’05.
3 Dyer E, Goldstein T, Patel R, Kording K, and Baraniuk R. “Self-expressive decompositions for matrix approximation and clustering”. arXiv:1505.00824 (2015).
4 Farahat A, Elgohary A, Ghodsi A, and Kamel M, “Greedy column subset selection for large-scale data sets” Knowledge and Information Systems (2014).
5 Ferris M. C, and Munson, “T. S. Interior-point methods for massive support vector machines” SIAM J. on Optimization (2002).
6 Figueiredo M, Nowak R, and Wright S, “Gradient projections for sparse reconstruction: Application to compressed sensing and other inverse problems” IEEE J. Select. Top. Signal Processing 1, 4 (2007).
7 Fine S, and Scheinberg K, “Efficient SVM training using low-rank kernel representations” JMLR’02.
8 Fowlkes C, Belongie S, Chung F, and Malik J. “Spectral grouping using the Nystrom method.” TPAMI’04.
9 Gilbert A., Strauss M., Tropp J., and Vershynin R. “One sketch for all: Fast algorithms for compressed sensing”, STOC’07.
10 Gittens A., and Mahoney M. “Revisiting the Nystrom method for improved large-scale machine learning”, ICML’13.
11 Low Y, Gonzalez J, Kyrola A. Bickson, D. Guestrin, C. and Hellerstein, J. M. “GraphLab: A new parallel framework for machine learning”, UAI’10.
12 Malewicz G., Austern M., Bik, A., Dehnert J., Horn I., Leiser N., and Czajkowski G. “Pregel: a system for large-scale graph processing”, SIGMOD’10.

13 Mirhoseini A., Baraniuk R., and Koushanfar F., "Rankmap: A platform-aware framework for distributed learning from dense datasets". IEEE Transactions on Neural Networks and Learning Systems (2015).
14 Patel R., Goldstein T., and Baraniuk R. "Adaptive column sampling and nystrom approximation via oasis" SDM'16
15 Zaharia M., Chowdhury M., Franklin M., Shenker S., and Stoica I. "Spark: cluster computing with working sets", USENIX CHTCC'10.
16 Rubinstein R., Zibulevsky M., and Elad M. "Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit", CS Technion'08.
17 Zhang T. "Solving large scale linear prediction problems using stochastic gradient descent algorithms", ICML '04