International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Machine Learning System Aware Optimization

Pratima Sharma[1], Priyanka Sharma[2]

[1]Assistant Professor, Aravali College of Engineering and Management (ACEM), Faridabad, Haryana, India
[2]Assistant Professor, Advanced Institute of Technology and Management (AITM), Palwal, Haryana, India

ABSTRACT
New computer systems have emerged in response to the increasing size and complexity of modern data sets. In order to ensure optimum performance, software approaches have to be closely matched with the basic features of systems. This research demonstrates the impact of system-sensitive machine learning using an optimizer lens, a crucial design and solution factor in the majority of machine learning problems. The exactness and convergence rates are traditionally measured for the optimization method. In contrast, a number of system-related variables are crucial to modern computing systems' overall efficiency. Specifications such as data or parameters for the device and higher-level meanings, such as communication and computer interconnections may be included. We propose CoCoA, an overall learning method that closely reviews and incorporates device parameters into the process and theory. We have shown the impact of CoCoA on two conventional distributed systems, that being the traditional cluster environment and the increasingly (founded) machine learning environment. Our results show that we get orders of magnitude quick, by combining system parameters and optimization techniques, to solve current machine learning difficulties. These empirical findings support the assumption that device parameters give more knowledge about the scientific performance.

Keywords: machine-learning, CoCoA, traditional cluster and optimization techniques

1. INTRODUCTION

Distributed computing architectures are the first to solve the challenges faced by a range of large apps in modern machine learning. Scalability is provided by distributed architectures due to increased equipment and storage space [1]. This commitment to scalability must be met by developing efficient information communication and synchronization methods between distributed machines which take the master study algorithms into account. Data sharing among devices is much more cost-effective than reading and measuring the local data from the main storage systems in most distributed networks. Furthermore, the optimal trade between interaction and measurement varies greatly depending on the data collection, the method used, and the optimized goal. As a result, when ensuring convergence, multiple coordination measurement profiles for distributed methods must be considered [1 & 2]. One of the most common paradigms for addressing this trade of contact measurement was optimization of small lot groups, which was one of the many methods of distributed optimization. The generalization of traditional stochastic methods also develops mini-batch methods for processing a large number of data points, allowing for more distributed computing through communication rounds and thus avoiding the communication bug. Although the requirement for reduced contact necessitates large miniature sizes,

theoretical convergence rates are reduced by increasing miniature sizes to conventional (batch) gradient rates in those methods. These theory rates are validated by empirical proof and only mini-batch methods in practice can be applied to maximize simultaneous measurement and interaction. Moreover, since mini-batch processes usually come with a system solver, they are often adapted to particular problems and disabled, theoretically and technically, when used in addition to the restricted setup of the problem [2]. In this analysis, we propose the CoCoA system to solve these two major constraints. First, we use arbitrary local solvers on each device at the same time. Individual solvers can be directly implemented into the distributed environment using our method. Second, we communicate with computers using a highly adaptable communication mechanism. This makes it simple to adjust the communication volume to the problem and system in use, particularly if communication within the distributed environment is significantly reduced. Necessary subproblems must be identified in order for each machine to be solved simultaneously and the updates from the subsets to be effectively combined in order to provide these functions. The focus of our approach and convergence findings is on the fact that in a distributed environment, some master learning goals are easier to break into sub-problems based on data distribution and whether the problem is solved primarily or dually. In certain cases, we identify certain master learning goals and use duality to help us achieve them. As we'll see, using primal-dual data not only ensures strong primal-dual convergence and practical advantages, but also ensures that the dual-distance usage and stop criterion are properly certified [1, 2 & 3].

1.1. Machine Learning: Modeling and Optimization

Mechanical learning breakthroughs and especially in-depth learning have recently become a major component of nearly every modern computing system (DL). Increase DL appliances have provoked a number of hardware based design problems on various platforms [3]. "What is the Deep Network (DNN) cost for latency or energy?" "Can latency or energy consumption be predicted before a model is even trained?" "If yes, how do the computer students create the optimal DNN for these models to be used?" The answers to these questions were very interesting, starting with the long life cycles of mobile devices and the reduction of the run-times of DL cloud models [3 & 4]. What is not properly modeled cannot be optimized. Therefore, the hardware effectiveness of DL models must be understood before the model is trained during the inference process. This main finding led to the use of predictive models for recording ML applications' hardware output or energy efficiency [4]. ML professionals often encounter problems in the design of a DNN, i.e. the DNN architecture Hyper Parameter Adjustment and the improvement of the DL model's precision and hardware performance. Hardware-conscious hyper-parameter optimization approaches are also proposed with recent methods. This paper provides an overview of machine learning system aware optimization. We also highlight open problems that will have significant effects on hardware systems and the related platforms in the coming years, as DL apps will continue to have new hardware solutions.

1.2. Modeling and optimization phase

1.2.1. The General framework

We construct a primary dual communication-efficient architecture for a wide range of convex optimization problems. Our simpler, united system, as opposed to earlier works [4]; and [5].

1. Specifically involves questioning L1 regulations and other convex regulations;
2. Provides the distribution by function or training points of the data flexibility; and

3. This can either be used with an elementary or dual formula that has major theoretical and practical implications.

1.2.2. Local solver and flexible connectivity

The stability of contact and the capacity to use individual system solvers internally are two of the major advantages of the proposed architecture. Communication costs differ greatly in real-world systems, and therefore a variable degree of communication can be permitted depending on the context [5]. That is exactly what our architecture has to offer. We also allow arbitrary solvers to be used on any computer, so that existing code can be reused and the benefits of multi-core or additional optimization.

1.2.3. The rates of primal-dual

When we research primal-dual rates for non-strict convex regulations, we derive convergence rates for our process. The proposed technology is significantly enhanced by adding a fast L2-term to the target, i.e. [6] and [7]. We demonstrate how previous work can be taken as a special case from our entire system and how the results are based on primary double rates and certificates for the general category of standard linear reductions.

1.2.4. An exploratory comparison

The proposed frame provides a variety of speeds in comparison with state-of-the-art machine learning methods (up to 50 unfaster). Via a systemic experimental comparison of distributed data sets in real time, we demonstrate these performance improvements. The characteristics of the frame itself are also discussed, including the impact of first or secondary frame structure operation. Apache Spark implements both algorithms and works with Amazon EC2 clusters [7].

1.2.5. The Learning Federated

In conclusion, the optimization of large networks of low power devices, including the new federal learning area, is studied in different distributed computing environments [6]. CoCoA Mocha is the perfect solution for dealing with federated world's special structures and statistical issues. The simulation of real-world data sets shows highest statistical efficiency and analytical accident of this approach and offers a sophisticated theoretical analysis to investigate the effects on our convergence guarantees of the system problems and failure tolerance [6 & 7].

2. Previous Related Research

2.1. Coordinates Solvers for a single computer

The RDCA (SDCA) with speedup variants is the latest state of the art to reduce empirical losses when regularization is strongly convex. The SDCA family is favored by the fact that it provides free parameters for learning rates and faster convergence in comparison with main stochastic grade descent (SGD) methods. The latest literature on lasso shows an interesting design of the coordinate solver with opposite original and double positions. Coordination descent methods have become state-of-the-art for problems such as Inglmnet and extensions [8]. However, only high convex regularization rates for unmodified coordination algorithms have been attained for both primary and dual convergence to date. The co-ordinated downward trends of the L1 system problem can be seen as an iterative smooth component minimization and a partial reduction. Their target is a smooth and straightforward component approach [9]. This is central to glmnet [28,106] which is commonly seen in a single coordinate update event in solvers based on main formulas for L1-regularised targets. This results in a two-language Inglmnet method of regression when several coordinates are simultaneously changed by the smooth,

quadratic top band. For the environment concerned, this concept is important. Where on the local machine, the set of co-ordinates is closely connected to the distributed system [8 & 9].

2.2. Methods of parallel

Methods to minimize general regularized losses of concern based on stochastic subgraphical descent (SGD). For the parallel computing [10] a number of SGD versions were suggested, several focused on asynchronous communication. The disadvantage of this distributed environment approach is that communications involves a quantity of local information to access one data center per turn per device in spite of its simplicity and its competitive efficient common memory systems. These variants are virtually non-competitive for the most effective methods of communication, which allow more local updates through communications rounds. The proposal for parallel coordinate descent [11] (Shotgun) was drawn up in [12], with and without the use of mini-batches for special L1 controlled and widespread objectives; this descent is one of the best parallel solutions. If an internal solver updates a single sub problem (2.10) μ=1 and″, Shotgun may be a special case in our framework. Shootgun does not fall into our convergence theory; however, as β is used in its potentially uncertain top rather than μ′ which does not guarantee that the conditions of our convergence are met. This high-communication approach has an empirical influence in the distributed environment compared to the effects of Shotgun.

2.3. Communication of One-Shot Systems

On the other hand, only one round of touch is used [13, 14 & 15]. This calls for more hypothesis for data exchange, which is usually not done in practice when the data are transferred, i.e. when the data are not exchanged. Moreover, a convergence rate beyond the achievement of ignoring data from all but one machine cannot be guaranteed by some [14]. It's irrational. The minimum requirement of minimum contact rounds for the approx. standard listed is specified in [14 & 15] as other applicable lower limits.

2.4. Method of mini-batch

Miniatures are more versatile with alerts from a variety of training places and/or features located in parallel and one-shot communications. But the small slot and CD versions have a decline in the convergence rate as the size of the mini slot has been increasing. Instant local upgrades like CoCoA are not used for the previously outdated parameter of vector w [16]. The other problem with thumbnails is that the aggregation parameter is harder to define, because thumbnails can be found elsewhere. Actually, the best option remains uncertain or hard to calculate. The cacao framework does not have to modify its parameters, because it can specifically specify the aggregation and subproblem parameters using a safe bound.

2.5. The batch solvers

In district settings, ADMM [16] also mostly uses both downward gradients and newtons such as L-BFGS, owing to the relatively limited touch requests. Each round, there is a minimum requirement for calculation of one (distributed) batch gradient and the interdependency between coCoA communications and calculations is not increased. Research work presents the experimental contrasts between L1 sets, which have a minimum orthodox memory (3), ADMMs, decreasing gradients and L-BFGS. Finally the weaker theory of existing methods for load gradient is not necessarily able to accuracy local issues, while coCoA convergence rates represent a classical batch gradient in external round numbers. In addition, this approach is precisely incorporated in our convergence rate and local solutions are also much cheaper than batch approaches. This allows CoCoA to adapt in real systems to different communication costs. This enhances the distribution of the atmosphere. The versatility of competitive approaches results in significant changes in the efficiency [17].

2.6. The Distributed solvers

The CoCoA-v1 and CoCoA+ frameworks have been the first to allow local weavers to be used, using primarily two lines of work [17 & 18], in the distributed environment. DisDCA-p permits the introduction in the DisDCA [17] hand-on edition of additional changes like CoCoA, but its role is limited to proper local solvers coordinated (CD). All are DisDCA-p, CoCoA-v1, CoCoA+, convexly regularized, so CoCoA+ is not discusses the overall structure of this article. An approach to our method involves glmnet variants as distributed in a balanced L1 environment. The definition of the Bloc Diagonal Top Hessian method has been implemented in a diffused context, based on the glmnet and [18] works [19] and [20]. This approach to rare logistical regression was later focused on work. If hypothetically any G″ (all together, [k'] in our subproblems can be precisely reduced as described by the definition (2.10), the resulting steps can be viewed as Newton steps in any k block where the L1 regularize is amended to include Newton's subproblem. While provides fixed precision, not arbitrary approximations as we consider fit, [20] and [21] are supposed to have solved the quadratic substrates correctly. There is then no free exchange of communication and computing. Furthermore, arbitrary local solvers cannot be reused. Theoretically, the convergence rate results from [21] are unclear but asymptotic; security, as it is at our <our> quadratic boundaries, is therefore not specifically regulated.

3. Proposed methodology

Our method aims to globally minimize Objective A and to distribute computations based on the division into dataset a machines. In the first step, the modification to the gin objective function is straightforward (A). The term should be divided by dividing up our data. But it's not the same for the term f (Aα). To reduce this part of the objective, we propose minimizing a quadratic approach to the function that allows a division between the devices. This calculation is correct in the following paragraph [22].

Quadratic discussion of the data with the local. A data local optimization problem (A) subset is included in the overall CoCoA context for each computer. The problem on the computer is simpler to fix and only needs local data, i.e. columns A[k]. More formally, the local problem has only been assigned for each device according to the previous common vector v = Aα to Rd and local dates A[k]:

$$\Delta\alpha^{Min}_{[k]} \epsilon R^{n} = G_{k}\sigma_{(}\Delta\alpha_{[k]}, V, \alpha_{[k])} \quad \ldots\ldots (1)$$

W: = Absolute (v). In this case, [k] is moving αi to Pk by index I, and I=0 by local variables to all i/Pk (α[k]). α[k] must be noted for a change in local variables. The problem (2.10) is straightforward because it is always a quadratic objective [23]. It is worth noting: (apart from the gi term). It depends, not on the vector v which is fixed, on its linearization. The work of the local solver, in particular for complex functions, is also simplified f.

3.1. γ *and* σ Parameters of framework

Two parameters should be defined in our context: α, the parameter that regulates each device's combination of updates, and μ′, a terminology based on data that calculates {Pk}Kk=1 data partition issues. It is necessary to specify two parameters. In procedural convergence, these principles play an important role. These parameters are directly and robustly defined in practice: The ″ parameter is set to μ′= βK, although the details provided for in the β-aggregation parameter can be further enhanced (0.1). Sets μ: =1 and μ′=K ensures overall convergence at our fastest rate [24].

3.2. primal in the CoCoA

The frame is managed using the general CoCoA framework as described in the algorithm first edition by directly mapping the initial problem into the goal (Algorithm 2). This implies that we consider and solve

it directly as the main objective. Theory suggests that laws are not entirely convex from the initial viewpoint, as a less convex definition is possible. This context was not addressed in previous work [24 & 25] and we looked at this in our study, as more machines could also be used for the original dual relationship. The first version of the approach has a significant practical impact on the distributed system, as it typically involves distribution instead of training of data for each purpose. In this sense, the number of touch with each external iteration is O (number of training points). This will decrease interactions and improve overall efficiency if the number of features is significant (as usual when usages are usually causing regularity).

Table.3.1. Algorithm 1 CoCoA-Primal (mapping problem)

Algorithm 1 CoCoA-Primal (mapping problem)
1. Map: object input issue 2. Distributed: dataset A by columns (usually featured here) by {Pk}kk = 1 partition 3. Run: γ and sub problem parameter of σ algorithm aggregation

3.3. Dual in the CoCoA

We have frames for mapping the original problem to the goal of the two distributed systems versions (algorithm), then run algorithms in two formats to solve the problem. In other words, this is the key concern and the double issue that we are addressing. The new version of the Framework allows non-modern losses such as hinge loss or loss of total variance as the terms cannot be fluid. The data are usually distributed on a training site in this version of the system and a vector touch functional for every external iteration. This version can also be preferred if more than one feature is available on several training sites [24 & 25].

Table.3.2. Algorithm 1 CoCoA-Dual (mapping problem)

Algorithm 2 CoCoA-dual (mapping problem)
1. **Map:** object input issue 2. **Distributed:** dataset A by columns (usually featured here) by {Pk}kk = 1 partition 3. **Run:** γ and sub problem parameter of σ algorithm aggregation

3.4. Comparison between primal vs. dual

This section will discuss three examples showing how main and double versions of CoCoA can be used for various input problems based on functional features (u). You may decide to double or run the framework, particularly if you have a highly convex smooth andris (Algorithm). The loss of high algorithm 1 and algorithm 2 is intuitively desirable. However, device-related issues must also be considered. In algorithm 1 and algorithm 2 we generally exchange knowledge across training points. (It depends on how our mapping describes the terms and conditions). Algorithm 1 or algorithm 2 can be used to minimize contact costs [26], depending on the dominant word features or training points. These ideas are empirically tested in comparison to actual data groups for each release (primary vs. dual).

Table.3.3. criteria for running algorithm 1 vs. algorithm 2

S.No.	description	Smooth l	Non-smooth and separable l
1.	Strongly convex r	Case i: algebraic – 1	Case i: algebraic – 1 & algebraic – 2
2.	No n-strongly convex and separable r	Case ii: algebraic – 2	-

3.5. Interpretation's

Different methods (A) and (B) have been developed to resolve both parallel and distributed contexts. We outline the work and illustrate the large algorithm between CocoA and other commonly employed paralleled methods [27]. In contrast to mini batch and batch processes, commonly used in the delivery of machines, compare CoCoA with mini stochastic down or coordinate downward descent, upwards gradients and near Newton. The CoCoA methods are alliterative, that is to say, by modifying the vector parameter to one functionality: On each iteration, Ron | ron > ron.

$$\alpha^{(t+1)} = h\,(\alpha^{(t)})\ t = 0,\ 1,\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (2)$$

Until convergence exists, from a coordinate point of view, the Jacobi System involves two methods in which α-coordinates shift does not include the most recent updates to the other co-ordinates and the latest details from Gauss-Seidel [28]. In this context, the parameter vector is updated iteratively by two approaches. The two paradigms make the following modifications to a t+1 coordinate:

$$\text{Jacobi: } \alpha_i^{(t+1)} = h_i(\alpha_1^{(t)},\ldots\ldots\ldots,\alpha_n^{(t)},\qquad I = 1,\ldots\ldots\ldots\ldots\ldots,\ n,$$

$$\text{Gauss-seidel: } \alpha_i^{(t+1)} = h_i(\alpha_1^{(t+1)},\ldots\ldots\ldots,\alpha_{i-1}^{(t+1)},\ \alpha_{I,}^{(t)},\ldots,\alpha_n^{(5)}\quad i= 1,\ldots,\ n,\ldots\ldots.(3)$$

The Jacobi method does not need data from the other coordinates to change the coordinates in order to adapt the style of parallelization [28 & 29]. However, iterations are usually more convenient for the Gauss Seidel, because data from other co-ordinates can be integrated more easily. This discrepancy, established and obvious in one single solver system, seems to exceed its batch equivalents using stochastic methods (benefit from new updated methods). Updates traditional mini load methods, such as the descent of mini loading coordinates, in the Jacobi-style to a sub-set of co-ordinates for each iteration. This enables these methods to be paralleled at high standards. With regard to the number of data points you have accessed, it cannot provide information as fast as its serial colleagues because it is necessary to expect to adjust synchronous co-ordinations. As the size of the mini packet increases, total time can be slowed down and function also realistically differs [28]. Instead, CoCoA tries with attractive features to combine both paradigms. Jacobi-style blocks are modified to α's cordoned in order that the Gauss-Seidel-style on each computer can be updated faster, but not necessarily needed. This parallel change is one of the main reasons for quality improvement by simple miniaturization or batch technology. CoCoA adds additional flexibility through arbitration of Gauss-Seidel iterations on each computer (or on any local solver in this connection), allowing the systems to operate from extremely low communications to higher media, with fewer internal items needed in advance of communication.

3.6. Comparison between CoCoA vs. ADMM

In this segment, you will also find an immediate comparison of CoCoA and ADMM [29]. A well known

distributed optimization mechanism is an alternative Multiplier Path Method (ADMM). Simile to CoCoA, in this ADMM, any problem must, rather than parallel a global batch or minigo batch update, be solved in parallel with the methods described in the previous section. It uses the duality structure close to the presented structure.

The (B) target is divided into a parameterization for consensus ADMM:

$$W_{1.wk.,w}^{max} \sum_{k=1}^{k} \sum_{i \in Pk}^{n} g * (-x_i^t wk) + f*(w)$$

$$S.t.\ W_{k=} w,\ k = 1,\ldots\ldots\ldots, (4)$$

This problem is solved through the construction of the Lagrangian increase which provides the following updates:

$$Wk\ (^{t}) = \arg \min_{wk} \sum_{i \in pk} g * (-x_i^T wk) + \frac{\rho}{2} ll\ w_k - (w^{(t-1)} - u_k^{(t-1)})\ ll^2,$$

$$Wk\ (^{t}) = \arg \min_{w} f* (w) + \rho \sum_{k=1}^{k} U_k^T (W_k - w) + \frac{\rho}{2} \sum_{k=1}^{k} IIW_k - W\ II^2,$$

$$U^{(t)}{}_k = u_k^{(t-1)} + w_k\ (t) - W^{(t),} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

4. RESULT AND DISCUSSION

The empirical development of CoCoA is apparent in the distributed data center sense. First, CoCoA has two popular machine learning applications, regression and SVM that are in opposition to competing methodology [30]. We research the output of CoCoA in the original versus double by explicitly solving an elastic net regression model with two different variants. Finally, we show that the general properties of other areas of the CoCoA system are applied empirically.

4.1. Information and configuration

CoCoA is compared with many advanced general-purpose optimization approaches, including:

1. Mb-SGD: stochastic gradient mini-batch. We equate Mb-SGD to L1-prox for our experiments with lasso.
2. DG: Complete downhill gradient. Prox-GD is Lasso's nearest variant.
3. L-BFGS: Near Newton limited memory. We use OWL-QN for lasso.
4. ADMM: Multiply alternative control method. For lasso experiments and SDCA experiments, we use a conjugate gradient internally.
5. Mb-CD: arrange simultaneous mini download. Mb-SDCA is used in SVM experiments (mini-batch stochastic dual coordinate ascent).

In Apache Spark's MLlib the three first processes are simplified and used [31] (v1.5.0). Each equipment is measured in large-scale experiments in accordance with Table 4.1 data sets. We separate from the optimum primary solution in contrast with other approaches. The optimal value of the results is determined using all the methods in a number of iterations (until progress is stopped). The entire code for Apache Spark is written and tests on Amazon EC2 m3.xlarge machines with one center per machine can be carried out in a public cloud format [32].

Table.4.1. datasets for empirical study

S.No.	datasets	Training size	Feature size	sparsity
1	Url	2 M	3 M	$3.5e^{-5}$
2	Epsilon	400 M	2 M	1.0
3	Kddb	19 M	29 M	$9.8e^{-5}$
4	Webspam	350 M	16 M	$2.0e^{-4}$

In order to enhance our test results, each competing strategy is carefully adapted. In order to choose a penalty parameter and to solve sub-specific problems, ADMM needs further tuning. For ADMM, it is prohibitively slow and thus we use a routine process internally to improve the performance at an early interval. The different activities mentioned in Boyd et al. [33] also apply penalty parameters [34 & 35]. We modify phase parameters for Mb-SGD. Updates to βb to [2, b] are calculated for Mb-CD and Mb-DCA for each batch size unit, and all β-band parameters are set. Details on the use of both methods are given. Strict CoCoA used the simple descent of the co-ordinate only as local solver in all of the following easy experiments. By connecting to state-of-the-art local solvers, we can see stronger empirical results for each application.

5. CONCLUSION

A general communication-efficient primal dual optimization system has been developed, analyzed and evaluated in a distributed environment with the goal of facilitating large-scale machine learning. Our architecture, Cocoa, is based on duality for each machine to resolve parallel subproblems. These under-problems are closamente linked to the global interest problem that enables reusability in the distributed world of state-of-the-art single-machine solvers. Furthermore, in order to reach a highly scalable communication arrangement, our local solvers are allowed on each device to find solutions for arbitrary sub-problems. Because local solvers directly update their local parameters, it is necessary to communicate and update the system in the distributed environment to handle the communication bottle. We assessed the accuracy of our local solver approach and calculated worldwide primal dual convergence rates agnostic to local solver specifications. We took great care to expand our approach into non-strong convex regularization in order to ensure stable convergence by applying a minimal support adjustment technique. We have demonstrated the effectiveness of our scheme in a thorough experimental comparison with state-of-the-art distributed solvers.

In the data center setting for real world distributed data sets our architecture reaches a rate of up to 50 times over other commonly used approaches. Finally, the overall framework was expanded to include the increasing federal climate, which is a variety of new mechanisms and statistical challenges. The Mocha amendments allow the method of addressing practical issues such as non-IID data, sluggishness and failure tolerance. Our approach is supported by a system-sensitive study that discusses the impact of these issues on our guarantee of convergence. We have demonstrated the impact of the proposed amendments in Mocha using simulations on real-world federated data collection. All in all, the results of those research theses show that, with the development of methods and theories which reveal and consider system parameters and give important empirical speeds and creative theoretical guarantees, we are able to provide solutions for modern machine training that comply with systems.

REFERENCE

1. T. Parnell, C. D¨unner, H. Pozidis, D. Sarigiannis. (2018). *Machine Learning in Heterogeneous Processing Systems.* Patent Office: US, Patent Number: P201802616US01, Application Number: US 16/214918.
2. Sulin Liu, Sinno Jialin Pan, and Qirong Ho. (2017). *Distributed Multi-task Relationship Learning*. In: Conference on Knowledge Discovery and Data Mining.
3. Celestine Dünner et al. (2016). *Primal-dual rates and certificates*. In: International Conference on Machine Learning.

4. C. D¨unner, T. Parnell, H. Pozidis. (2018). *Machine Learning in Heterogeneous Processing Systems.* Patent Office: US, Patent Number: P201802665US01, Application Number: US 16/214639.
5. Christina Heinze, Brian McWilliams, and Nicolai Meinshausen. (2016). *DUAL-LOCO: Distributing Statistical Estimation Using Random Projections.* In: International Conference on Artificial Intelligence and Statistics.
6. Kirak Hong et al. (2013). *Mobile Fog: A Programming Model for Large-scale Applications on the Internet of Things.* In: SIGCOMM Workshop on Mobile Cloud Computing.
7. T. Parnell, C. D unner, D. Sarigiannis, H. Pozidis. (2018). *Machine Learning Implementation in Processing Systems.* Patent Office: US, Patent Number: P201802615US01, Application Number: US 16/144550.
8. Olivier Fercoq and Peter Richtarik. (2015). *Accelerated, Parallel, and Proximal Coordinate Descent.* In: SIAM Journal on Optimization 25.4, pp. 1997–2023.
9. C. D¨unner, T. Parnell, H. Pozidis. (2018). *A Memory-based Data-selection Scheme for Machine Learning Training on Limited Memory Resources.* Patent Office: US, Patent Number: P201704087US01, Application Number: US 15/953440.
10. ANDREW, Galen; GAO, (2007). *Jianfeng: Scalable training of L1-regularized log-linear models.* In International Conference on Machine Learning.
11. Joseph K Bradley et al. (2011). *Parallel coordinate descent for l1-regularized loss minimization.* In: International Conference on Machine Learning.
12. Martin Jaggi et al. (2014). *Communication-efficient distributed dual coordinate ascent*. In: Neural Information Processing Systems.
13. mr Ahmed, Abhimanyu Das, and Alexander J Smola. (2014). *Scalable hierarchical multi-task learning algorithms for conversion optimization in display advertising.* In: Conference on Web Search and Data Mining.
14. Jerome Friedman, Trevor Hastie, and Robert Tibshirani. (2010). *Regularization paths for generalized linear models via coordinate descent.* In: Journal of Statistical Software33.1, pp. 1–22.
15. BORWEIN, J M.; ZHU, (2005). *Techniques of Variation Analysis and Nonlinear Optimization.* Springer New York: Canadian Mathematical Society Books in Math.
16. Yatao Bian et al. (2013). *Parallel Coordinate Descent Newton Method for Efficient1-Regularized Minimization.* In: arXiv.org.
17. D UNNER, Celestine; PARNELL, Thomas P.; JAGGI, Martin (2017). *Efficient Use of Limited-Memory Accelerators for Linear Learning on Heterogeneous Systems.* In Advances in Neural Information Processing Systems (NIPS), p. 4261–4270.
18. Aaron Carroll and Gernot Heiser. (2010). *An Analysis of Power Consumption in a Smart-phone.* In: USENIX Annual Technical Conference.
19. FERCOQ, Olivier; RICHT´ARIK, Peter (2016). *Optimization in High Dimensions via Accelerated, Parallel, and Proximal Coordinate Descent.* In SIAM Review 58, no. 4, p. 739–771.
20. Galen Andrew and Jianfeng Gao. (2007). *Scalable training of L1-regularized log-linear models*. In: International Conference on Machine Learning.
21. Xin Jin et al. (2015). *Collaborating between local and global learning for distributed online multiple tasks*. In: Conference on Information and Knowledge Management.
22. Pedro Garcia Lopez et al. (2015). *Edge-centric computing: Vision and Challenges*. In: SIG-COMM Computer Communication Review 45.5, pp. 37–42.

23. FRIEDMAN, Jerome; HASTIE, Trevor; TIBSHIRANI, (2010). *Robert: Regularization Paths for Generalized Linear Models via Coordinate Descent.* In Journal of Statistical Software 33, no. 1, p. 1–22.
24. M. Baytas et al. (2016). *Asynchronous Multi-Task Learning*. In: International Conference on Data Mining.
25. HUSH, Don; KELLY, Patrick; SCOVEL, Clint; STEINWART, (2006). *QP algorithms with guaranteed accuracy and run time for support vector machines.* In The Journal of Machine Learning Research7, p. 733–769.
26. Jianhui Chen, Jiayu Zhou, and Jieping Ye. (2011). *Integrating low-rank and group-sparse structures for robust multi-task learning*. In: Conference on Knowledge Discovery and Data Mining.
27. Chenxin Ma et al. (2015). *Adding vs. Averaging in Distributed Primal-Dual Optimization*. In: International Conference on Machine Learning.
28. Davide Anguita et al. (2013). *A Public Domain Dataset for Human Activity Recognition using Smartphone.* In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning.
29. Hamed Karimi, Julie Nutini, and Mark Schmidt. (2016). *Linear Convergence of Gradient and Proximal-Gradient Methods under the Polyak lojasiewicz Condition*. In: European Conference on Machine Learning.
30. John Duchi, Michael I Jordan, and Brendan McMahan. (2013). *Estimation, optimization, and parallelism when data is sparse.* In: Neural Information Processing Systems.
31. André R Gonçalves, Fernando J Von Zuben, and Arindam Banerjee. (2016). *Multi ask sparse structure learning with Gaussian copula models*. In: Journal of Machine Learning Research17.33, pp. 1–30.
32. Chenxin Ma, Rachael Tappenden, and Martin Takac. (2015). *Linear Convergence of the Randomized Feasible Descent Method under the Weak Strong Convexity Assumption*. In: arXiv.org.
33. ossi Arjevani and Ohad Shamir. (2015*). Communication complexity of distributed convex learning and optimization*. In: Neural Information Processing Systems.
34. NESTEROV (2012). *Efficiency of coordinate descent methods on huge-scale optimization problems.* In SIAM Journal on Optimization 22, no. 2, p. 341–362.
35. QU, Zheng; RICHT´ARIK, Peter; ZHANG, Tong (2015). *Quartz: Randomized Dual Coordinate Ascent with Arbitrary Sampling.* In Advances in Neural Information Processing Systems 28, (NIPS), p. 865–873.