ISE

Industrial and
Systems Engineering

Grow Your Samples and Optimize Better via Distributed Newton CG and Accumulating Strategy

Majid Jahani[1], Xi He[1], Chenxin Ma[1], Aryan Mokhtari[2], Dheevatsa Mudigere[3], Alejandro Ribeiro[4], and Martin Takáč[5]

[1]Lehigh University

[2]University of Texas at Austin

[3]Facebook

[4]University of Pennsylvania

[5]Department of Industrial and Systems Engineering, Lehigh University

ISE Technical Report 19T-029

LEHIGH
UNIVERSITY

Grow Your Samples and Optimize Better via Distributed Newton CG and Accumulating Strategy

Majid Jahani
Lehigh University
maj316@lehigh.edu

Xi He
Lehigh University
heeryerate@gmail.com

Chenxin Ma
Lehigh University
machx9@gmail.com

Aryan Mokhtari
University of Texas at Austin
mokhtari@austin.utexas.edu

Dheevatsa Mudigere
Facebook
dheevatsa@fb.com

Alejandro Ribeiro
University of Pennsylvania
aribeiro@seas.upenn.edu

Martin Takáč
Lehigh University
Takac.MT@gmail.com

Abstract

In this work[1], we propose a Distributed Accumulated Newton Conjugate gradiEnt (DANCE) method in which sample size is gradually increasing to quickly obtain a solution whose empirical loss is under satisfactory statistical accuracy. We give various iteration complexity results, communication efficiency and stopping criteria of the method, and perform extensive numerical experiments.

1 Introduction

In the field of machine learning, solving the expected risk minimization problem has received lots of attentions over the last decades, which is in the form of

$$\min_{w\in\mathbb{R}^d} L(w) = \min_{w\in\mathbb{R}^d} \mathbb{E}_z[f(w,z)], \tag{1.1}$$

where z is a $d+1$ dimensional random variable containing both feature variables and a response variable. $f(w,z)$ is a loss function with respect to w and any fixed value of z. One general idea is to estimate the expectation with a statistical average over a large number of independent and identically distributed data samples $\{z_1, z_2, \ldots, z_N\}$ where N is the total number of samples. The problem in (1.1) can be rewritten as the Empirical Risk Minimization (ERM) problem

$$\min_{w\in\mathbb{R}^d} L_N(w) = \min_{w\in\mathbb{R}^d} \frac{1}{N}\sum_{i=1}^{N} f_i(w), \tag{1.2}$$

where $f_i(w) = f(w, z_i)$. Many studies have been done on developing optimization algorithms to find an optimal solution of the above problem under different setting. For example, the studies by [2, 13, 19, 24] are some of the gradient-based methods which require at least one pass over all data samples to evaluate the gradient $\nabla L_N(w)$. As the sample size N becomes larger, these methods would be less efficient compared to stochastic gradient methods where the gradient is approximated based on a small number of samples [12, 17, 18, 25, 27, 29]. Second order methods are well known to share faster convergence rate by utilizing the Hessian information. Recently, several papers by [3–6, 10, 16, 22, 26, 28] have studied how to apply second orders methods to solve ERM problem.

[1] The full-length paper including additional results, proofs of statements and details of experiments, is available in [15].

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

However, evaluating the Hessian inverse or its approximation is always computationally costly, leading to a significant difficulty on applying these methods on large-scale problems.

The above difficulty can be addressed by applying the idea of adaptive sample size methods by recent works of [14, 21, 23], which is based on the following two facts. First, the empirical risk and the statistical loss have different minimizers, and it is not necessary to go further than the difference between the mentioned two objectives, which is called *statistical accuracy*. More importantly, if we increase the size of the samples in the ERM problem the solutions should not significantly change as samples are drawn from a fixed but unknown probability distribution.

In this paper, we propose an increasing sample size second-order method which solves the Newton step in ERM problems more efficiently. Our proposed algorithm, called Distributed Accumulated Newton Conjugate gradiEnt (DANCE), starts with a small number of samples and minimizes their corresponding ERM problem. This subproblem is solved up to a specific accuracy, and the solution of this stage is used as a warm start for the next stage in which we solve the next empirical risk with a larger number of samples, which contains all the previous samples. Such procedure is run iteratively until either all the samples have been included, or we find that it is unnecessary to further increase the sample size. Our DANCE method combines the idea of increasing sample size and the inexact damped Newton method discussed in the works of [31] and [20]. Instead of solving the Newton system directly, we apply preconditioned conjugate gradient (PCG) method as the solver for each Newton step. The DANCE method is designed to be easily parallelized and shares the strong scaling property, i.e., linear speed-up property. We formally characterize the required number of communication rounds to reach the statistical accuracy of the full dataset. In particular, Table 1 highlights the advantage of DANCE with respect to other adaptive sample size methods.

Table 1: Comparison of computational complexity between different algorithms for convex functions

Method	**Complexity**
AdaNewton	$\mathcal{O}(2Nd^2 + d^3 \log_2(N))$
***k*-TAN**	$\mathcal{O}(2Nd^2 + d^2 \log_2(N) \log k)$
DANCE	$\tilde{\mathcal{O}}((\log_2(N))^3 N^{1/4} d^2)$

2 Problem Formulation

In this paper, we focus on finding the optimal solution w^* of the problem in (1.1). As described earlier, due to difficulties in the expected risk minimization, as an alternative, we aim to find a solution for the empirical loss function $L_N(w)$, which is the empirical mean over N samples. Now, consider the empirical loss $L_n(w)$ associated with $n \leq N$ samples. In [9] and [7], it has been shown that the difference between the expected loss and the empirical loss L_n with high probability (w.h.p.) is upper bounded by the statistical accuracy V_n, i.e., w.h.p. $\sup_{w \in \mathbb{R}^d} |L(w) - L_n(w)| \leq V_n$, where $V_n = \mathcal{O}(1/n^\gamma)$ where $\gamma \in [0.5, 1]$ [1, 8, 30].

For problem (1.2), if we find an approximate solution w_n which satisfies the inequality $L_n(w_n) - L_n(\hat{w}_n) \leq V_n$, where $\hat{w}_n$ is the true minimizer of L_n, it is not necessary to go further and find a better solution (a solution with less optimization error). The reason comes from the fact that for a more accurate solution the summation of estimation and optimization errors does not become smaller than V_n. Therefore, when we say that w_n is a V_n-suboptimal solution for the risk L_n, it means that $L_n(w_n) - L_n(\hat{w}_n) \leq V_n$. In other words, w_n solves problem (1.2) within its statistical accuracy.

It is crucial to note that if we add an additional term in the magnitude of V_n to the empirical loss L_n, the new solution is also in the similar magnitude as V_n to the expected loss L. Therefore, we can regularize the non-strongly convex loss function L_n by $cV_n\|w\|^2/2$ and consider it as follows:

$$\min_{w \in \mathbb{R}^d} R_n(w) := \frac{1}{n} \sum_{i=1}^{n} f_i(w) + \frac{cV_n}{2} \|w\|^2. \tag{2.1}$$

In order to prove our results the following conditions are considered in our analysis.

Assumption 1. *The loss functions $f(w, z)$ are convex w.r.t w for all values of z. In addition, their gradients $\nabla f(w, z)$ are $M-$Lipschitz continuous.*

Assumption 2. *The loss functions $f(w, z)$ are self-concordant w.r.t w for all values of z.*

The immediate conclusion of Assumption 1 is that both $L(w)$ and $L_n(w)$ are convex and M-smooth. Also, we can note that $R_n(w)$ is cV_n-strongly convex and $(cV_n + M)$-smooth. Moreover, by Assumption 2, $R_n(w)$ is also self-concordant.

3 Distributed Accumulated Newton Conjugate Gradient Method

In order to utilize the important features of ERM, we combine the idea of increasing sample size and the inexact damped Newton method [31]. In our proposed method, we start with handling a small number of samples, assume m_0 samples. We then solve its corresponding ERM to its statistical accuracy, i.e. V_{m_0}, using the inexact damped Newton algorithm. In the next step, we increase the number of samples geometrically with rate of $\alpha > 1$, i.e., αm_0 samples. The approximated solution of the previous ERM can be used as a warm start point to find the solution of the new ERM. The sample size increases until it equals the number of full samples.

Consider the iterate w_m within the statistical accuracy of the set with m samples, i.e. $\mathcal{S}_m$ for the risk R_m. In DANCE, we increase the size of the training set to $n = \alpha m$ and use the inexact damped Newton to find the iterate w_n such that $R_n(w_n) - R_n(w_n^*) \leq V_n$ after K_n iterations. To do so, we initialize $\tilde{w}_0 = w_m$ and update the iterates according to the following

$$\tilde{w}_{k+1} = \tilde{w}_k - \tfrac{1}{1+\delta_n(\tilde{w}_k)} v_k, \tag{3.1}$$

where v_k is an ϵ_k-Newton direction. The outcome of applying (3.1) for $k = K_n$ iterations is the approximate solution w_n for the risk R_n, i.e., $w_n := \tilde{w}_{K_n}$. The favorable descent direction would be the Newton direction $-\nabla^2 R_n(\tilde{w}_k)^{-1}\nabla R_n(\tilde{w}_k)$; however, the cost of computing this direction is prohibitive. Therefore, we use v_k which is an ϵ_k-Newton direction satisfying the condition

$$\|\nabla^2 R_n(\tilde{w}_k) v_k - \nabla R_n(\tilde{w}_k)\| \leq \epsilon_k. \tag{3.2}$$

As we use the descent direction v_k which is an approximation for the Newton step, we also redefine the Newton decrement $\delta_n(\tilde{w}_k)$ based on this modification. To be more specific, we define $\delta_n(\tilde{w}_k) := (v_k^T \nabla^2 R_n(\tilde{w}_k) v_k)^{1/2}$ as the approximation of (exact) Newton decrement $(\nabla R_n(\tilde{w}_k)^T \nabla^2 R_n(\tilde{w}_k)^{-1} \nabla R_n(\tilde{w}_k))^{1/2}$, and use it in the update in (3.1). In order to find v_k which is an ϵ_k-Newton direction, we use Preconditioned CG (PCG).

Note that ϵ_k has a crucial effect on the speed of the algorithm. When $\epsilon_k = 0$, then v_k is the exact Newton direction, and the update in (3.1) is the exact damped Newton step (which recovers the update in Ada Newton algorithm in [21] when the step-length is 1). Furthermore, the number of total iterations to reach V_N-suboptimal solution for the risk R_N is $\mathbf{K}$, i.e. $\mathbf{K} = K_{m_0} + K_{\alpha m_0} + \cdots + K_N$. Hence, if we start with the iterate w_{m_0} with corresponding m_0 samples, after $\mathbf{K}$ iterations, we reach w_N with statistical accuracy of V_N for the whole dataset.

Algorithm 1 DANCE

```
 1: Initialization: Sample size increase constant $\alpha$, initial sample size $n = m_0$ and $w_n = w_{m_0}$ with $\|\nabla R_n(w_n)\| < (\sqrt{2c})V_n$
 2: while $n \leq N$ do
 3:     Update $w_m = w_n$ and $m = n$
 4:     Increase sample size: $n = \min\{\alpha m, N\}$
 5:     Set $\tilde{w}_0 = w_m$ and set $k = 0$
 6:     repeat
 7:         Calculate $v_k$ and $\delta_n(\tilde{w}_k)$ by Algorithm 2 PCG
 8:         Set $\tilde{w}_{k+1} = \tilde{w}_k - \frac{1}{1+\delta_n(\tilde{w}_k)} v_k$
 9:         $k = k + 1$
10:     until satisfy stop criteria leading to $R_n(\tilde{w}_k) - R_n(w_n^*) \leq V_n$
11:     Set $w_n = \tilde{w}_k$
12: end while
```

Our proposed method is summarized in Algorithm 1. We start with m_0 samples, and an initial point w_{m_0} which is an $V_{m_0}-$ suboptimal solution for the risk R_{m_0}. In every iteration of outer loop of Algorithm 1, we increase the sample size geometrically with rate of α in step 4. In the inner loop of Algorithm 1, i.e. steps 6-10, in order to calculate the approximate Newton direction and approximate Newton decrement, we use PCG algorithm which is shown in Algorithm 2, which the entire dataset is stored across $\mathcal{K}$ machines, i.e., each machine stores N_i data samples such that $\sum_{i=1}^{\mathcal{K}} N_i = N$. This process repeats till we get the point w_N with statistical accuracy of V_N. We now provide the theoretical results for DANCE.

Theorem 3.1. *By assuming $\alpha = 2$ and $\gamma \in [0.5, 1]$, the total number of communication rounds to reach a V_N-suboptimal solution of the full training set is w.h.p. $\tilde{\mathcal{T}} = \tilde{\mathcal{O}}(\gamma(\log_2 N)^2 \sqrt{N^\gamma} \log_2 N^\gamma)$.*

Algorithm 2 PCG

```
Master Node:                                                                  Worker Nodes ($i = 1, 2, \dots, \mathcal{K}$):
Input: $\tilde{w}_k \in \mathbb{R}^d$, $\epsilon_k$, and $\mathcal{A}_n$
Let $H = \nabla^2 R_n(\tilde{w}_k)$, $P = \frac{1}{|\mathcal{A}_n|} \sum_{i \in \mathcal{A}_n} \nabla^2 R_n^i(\tilde{w}_k) + \mu_n I$
Broadcast: $\tilde{w}_k$                      $\longrightarrow$                Compute $\nabla R_n^i(\tilde{w}_k)$
Reduce: $\nabla R_n^i(\tilde{w}_k)$ to $\nabla R_n(\tilde{w}_k)$       $\longleftarrow$
Set $r^{(0)} = \nabla R_n(\tilde{w}_k)$, $u^{(0)} = s^{(0)} = P^{-1} r^{(0)}$
Set $v^{(0)} = 0$, $t = 0$
repeat
    Broadcast: $u^{(t)}$ and $v^{(t)}$        $\longrightarrow$        Compute $\nabla^2 R_n^i(\tilde{w}_k) u^{(t)}$ and $\nabla^2 R_n^i(\tilde{w}_k) v^{(t)}$
    Reduce: $\nabla^2 R_n^i(\tilde{w}_k) u^{(t)}$ and $\nabla^2 R_n^i(\tilde{w}_k) v^{(t)}$ to $Hu^{(t)}$ and $Hv^{(t)}$        $\longleftarrow$
    Compute $\gamma_t = \frac{\langle r^{(t)}, s^{(t)} \rangle}{\langle u^{(t)}, Hu^{(t)} \rangle}$
    Set $v^{(t+1)} = v^{(t)} + \gamma_t u^{(t)}$, $r^{(t+1)} = r^{(t)} - \gamma_t H u^{(t)}$
    Compute $\zeta_t = \frac{\langle r^{(t+1)}, s^{(t+1)} \rangle}{\langle r^{(t)}, s^{(t)} \rangle}$
    Set $P s^{(t+1)} = r^{(t+1)}$, $u^{(t+1)} = s^{(t+1)} + \zeta_t u^{(t)}$
    Set $t = t + 1$
until $\|r^{(t+1)}\| \leq \epsilon_k$
Output: $v_k = v^{(t+1)}$, $\delta_n(\tilde{w}_k) = \sqrt{v_k^T H v^{(t)} + \gamma_t v_k^T H u^{(t)}}$
```

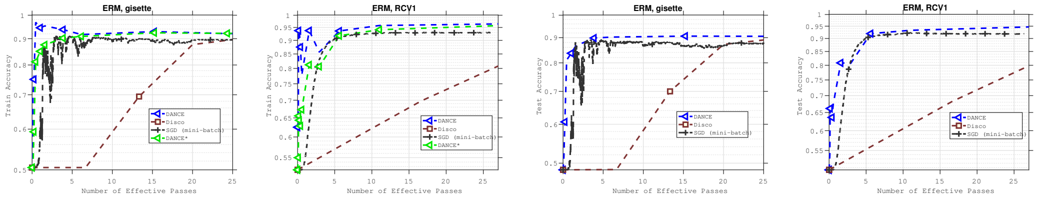

Figure 1: Performance of different algorithms on a logistic regression problem with rcv1 dataset. In the left two figures, the plot *DANCE** is the training accuracy based on the entire training set, while the plot *DANCE* represents the training accuracy based on the current sample size.

The rounds of communication for DiSCO in [31][2] is $\tilde{\mathcal{T}}_{DiSCO} = \tilde{\mathcal{O}}((R_N(w_0) - R_N(w_N^*) + \gamma(\log_2 N))\sqrt{N^\gamma} \log_2 N^\gamma)$ where $\gamma \in [0.5, 1]$. Comparing these bounds shows that the communication complexity of DANCE is independent of the choice of initial variable w_0 and the suboptimality $R_N(w_0) - R_N(w_N^*)$, while the overall communication complexity of DiSCO depends on the initial suboptimality. In addition, implementation of each iteration of DiSCO requires processing all the samples in the dataset, while DANCE only operates on an increasing subset of samples at each phase. Therefore, the computation complexity of DANCE is also lower than DiSCO for achieving the statistical accuracy of the training set.

Theorem 3.2. *The total complexity of DANCE Algorithm in order to reach a point with the statistical accuracy of V_N of the full training set is w.h.p. $\tilde{\mathcal{O}}((\log_2(N))^3 N^{1/4} d^2)$.*

All in all, the theoretical results highlight that DANCE is more efficient than DiSCO in terms of communication, and has lower complexity than previous adaptive sample size methods (see Table 1).

4 Numerical Experiments

In this section, we present numerical experiments on several large datasets to show that our DANCE algorithm can outperform other considered methods on solving both convex and non-convex problems. For the convex case, we use logistic regression model for binary classification tasks with *rcv1* and *gisette* [11] datasets.

In Figure 1, we observe consistently that DANCE has a better performance over the other two methods from the beginning stages. Both training and test accuracy for DANCE converges to optimality after processing a small number of samples. This observation suggests that DANCE finds a good initial solution and updates it over time. Compared with

[2] In order to have fair comparison, we put $f = R_N$, $\epsilon = V_N$, and $\lambda = cV_N$ in their analysis, and also the constants are ignored for the communication complexity.

DiSCO, our restarting approach helps to reduce computational cost for the first iterations. Moreover, we demonstrate that our DANCE method shares a strong scaling property. As shown in Figure 2, whenever we increase the number of nodes, we obtain acceleration towards optimality. We use the starting batchsize from 256 upto 4096, and the speed-up compared to the serial run (1 node) is reported. It indicates that as we increase the batchsize, the speed-up becomes closer to ideal linear speed-up. The advantage of the setting is to utilize the large batch over multiple nodes efficiently but not sacrifice the convergence performance. Regarding first order methods like SGD, it is hard to achieve nice linear scaling since the small batch is often required, which makes the computation time to be comparable with communication cost.

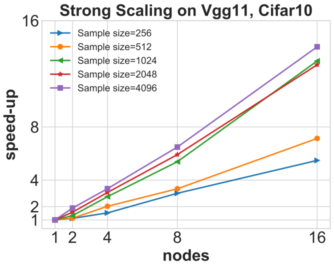

Figure 2: Performance of DANCE w.r.t. different number of nodes.

5 Acknowledgements

This work was partially supported by National Science Foundation grants CCF-1618717, CMMI-1663256 and CCF-1740796.

References

[1] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

[2] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[3] Albert S Berahas, Majid Jahani, and Martin Takáč. Quasi-newton methods for deep learning: Forget the past, just sample. *arXiv preprint arXiv:1901.09997*, 2019.

[4] Albert S Berahas, Jorge Nocedal, and Martin Takác. A multi-batch l-bfgs method for machine learning. In *Advances in Neural Information Processing Systems*, pages 1055–1063, 2016.

[5] Albert S Berahas and Martin Takáč. A robust multi-batch l-bfgs method for machine learning. *arXiv preprint arXiv:1707.08552*, 2017.

[6] Raghu Bollapragada, Richard H Byrd, and Jorge Nocedal. Exact and inexact subsampled newton methods for optimization. *IMA Journal of Numerical Analysis*, 2016.

[7] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[8] olivier Bousquet. *Concentration Inequalities and Empirical Processes Theory Applied to the Analysis of Learning Algorithms*. PhD thesis, Biologische Kybernetik, 2002.

[9] Olivier Bousquet and Léon Bottou. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems 20*, pages 161–168, 2008.

[10] Richard H. Byrd, S. L. Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-Newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016.

[11] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[12] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems 27*, pages 1646–1654, 2014.

[13] Dmitriy Drusvyatskiy, Maryam Fazel, and Scott Roy. An optimal first order method based on optimal quadratic averaging. *SIAM Journal on Optimization*, 28(1):251–271, 2018.

[14] Mark Eisen, Aryan Mokhtari, and Alejandro Ribeiro. Large scale empirical risk minimization via truncated adaptive Newton method. In *International Conference on Artificial Intelligence and Statistics*, pages 1447–1455, 2018.

[15] Majid Jahani, Xi He, Chenxin Ma, Aryan Mokhtari, Dheevatsa Mudigere, Alejandro Ribeiro, and Martin Takáč. Efficient distributed hessian free algorithm for large-scale empirical risk minimization via accumulating sample strategy. *arXiv preprint arXiv:1810.11507*, 2018.

[16] Majid Jahani, Mohammadreza Nazari, Sergey Rusakov, Albert S Berahas, and Martin Takáč. Scaling up quasi-newton algorithms: Communication efficient distributed sr1. *arXiv preprint arXiv:1905.13096*, 2019.

[17] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, pages 315–323, 2013.

[18] Jakub Konečnỳ and Peter Richtárik. Semi-stochastic gradient descent methods. *Frontiers in Applied Mathematics and Statistics*, 3:9, 2017.

[19] Chenxin Ma, Naga Venkata C Gudapati, Majid Jahani, Rachael Tappenden, and Martin Takáč. Underestimate sequences via quadratic averaging. *arXiv preprint arXiv:1710.03695*, 2017.

[20] Chenxin Ma and Martin Takáč. Distributed inexact damped Newton method: Data partitioning and load-balancing. *arXiv preprint arXiv:1603.05191*, 2016.

[21] Aryan Mokhtari, Hadi Daneshmand, Aurelien Lucchi, Thomas Hofmann, and Alejandro Ribeiro. Adaptive Newton method for empirical risk minimization to statistical accuracy. In *Advances in Neural Information Processing Systems 29*, pages 4062–4070, 2016.

[22] Aryan Mokhtari and Alejandro Ribeiro. Global convergence of online limited memory bfgs. *Journal of Machine Learning Research*, 16(1):3151–3181, 2015.

[23] Aryan Mokhtari and Alejandro Ribeiro. First-order adaptive sample size methods to reduce complexity of empirical risk minimization. In *Advances in Neural Information Processing Systems 30*, pages 2057–2065, 2017.

[24] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[25] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2613–2621, 2017.

[26] Farbod Roosta-Khorasani and Michael W. Mahoney. Sub-sampled newton methods. *Mathematical Programming*, 2018.

[27] Nicolas L Roux, Mark Schmidt, and Francis R Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems 25*, pages 2663–2671, 2012.

[28] Nicol N Schraudolph, Jin Yu, and Simon Günter. A stochastic quasi-Newton method for online convex optimization. In *Artificial Intelligence and Statistics*, pages 436–443, 2007.

[29] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 2013.

[30] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[31] Yuchen Zhang and Xiao Lin. Disco: Distributed optimization for self-concordant empirical loss. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 362–370, 2015.