# Distributed Machine Learning: Foundations, Trends, and Practices

Tie-Yan Liu
Microsoft Research
No.5 Danling Street,
Haidian District, Beijing China
+86 1059175257
tyliu@microsoft.com

Wei Chen
Microsoft Research
No.5 Danling Street,
Haidian District, Beijing China
+86 1059174985
wche@microsoft.com

Taifeng Wang
Microsoft Research
No.5 Danling Street,
Haidian District, Beijing China
+86 1059173357
taifengw@microsoft.com

## ABSTRACT
In recent years, artificial intelligence has achieved great success in many important applications. Both novel machine learning algorithms (e.g., deep neural networks), and their distributed implementations play very critical roles in the success. In this tutorial, we will first review popular machine learning algorithms and the optimization techniques they use. Second, we will introduce widely used ways of parallelizing machine learning algorithms (including both data parallelism and model parallelism, both synchronous and asynchronous parallelization), and discuss their theoretical properties, strengths, and weakness. Third, we will present some recent works that try to improve standard parallelization mechanisms. Last, we will provide some practical examples of parallelizing given machine learning algorithms in online application (e.g. Recommendation and Ranking) by using popular distributed platforms, such as Spark MlLib, DMTK, and Tensorflow. By listening to this tutorial, the audience can form a clear knowledge framework about distributed machine learning, and gain some hands-on experiences on parallelizing a given machine learning algorithm using popular distributed systems.

## Keywords
Machine Learning, Distributed Machine Learning, Parallelization, Optimization Methods, Distributed System

## 1. INTRODUCTION
In recent years, artificial intelligence has demonstrated its power in many important applications. On one hand, these successes should be attributed to the adoption of novel machine learning algorithms (e.g., deep neural networks); on the other hand, the existence of big training data and the capability of training big models from the data also play a critical role. This corresponds to a new research discipline, called distributed machine learning. Around this topic, many papers have been published, many theories have been developed, and many systems/platforms have been built.


*WWW 2017*, April 3-7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
DOI: http://dx.doi.org/10.1145/3041021.3051099

The goal of this tutorial is to give a comprehensive review of this newly emerged research field, and to provide the audience with both theoretical guidelines and practical instructions on how to use distributed machine learning technologies to solve their own problems, which includes many big data problems in web scenario.

Specifically, in this tutorial, we will first review popular machine learning algorithms and the optimization technologies they use. For example, for algorithms, we will introduce deep neural networks, decision trees, and logistic regression; and for optimization techniques, we will introduce SGD [3][22], SCD [21][23], ADMM [4][27], ProxSGD [13], Frank-Wolf [30][31], Second-order optimizers [28][29], Nesterov's acceleration [20], momentum methods [32], variance reduction methods [11], and their combinations [14][15][18].

Second, we will introduce typical ways of parallelizing machine learning algorithms, including data parallelism and model parallelism. For data parallelism, we will introduce both synchronous mechanisms (such as BSP, model average, ADMM) and asynchronous mechanisms (such as downpour ASGD, asynchronous ADMM, etc.). For model parallelism, we will introduce model partitioning and model scheduling. During the introduction of these parallelization mechanisms, we will take optimization techniques like SGD [5][7], SVRG [35][34], and ADMM [4][33] as examples, so as to discuss how the parallelization mechanisms affect the convergence property of an optimization technique, and compare their pros and cons in different situations.

Third, we will discuss some recent research works that try to improve standard parallelization mechanisms, in the following three directions. (1) Parallelism of combined optimization techniques. It is analyzed in [14][15][18] how the parallelization mechanisms will affect the overall convergence rate when several different optimization techniques are simultaneously used in one machine learning algorithm. (2) Advanced synchronous and asynchronous algorithms. In [6], block momentum is used to speed up synchronous parallelization; in [5], backup servers are used to deal with stragglers; in [10], the staleness of local updates is controlled by a bounded asynchronous framework; and in [25], a delay-compensation method is proposed to recover the right information from the delayed gradients based on Taylor expansion. (3) New communication/aggregation methods. In [12], a more communication-efficient method via voting is proposed for distributed decision trees; in [24], ensemble is proposed to replace model average for the aggregation of locally trained neural

networks, which can better deal with parallelizing the non-convexity of the neural networks.

Last, we will give a very practical demonstration on how to parallelize a given machine learning algorithm using popular distributed frameworks: *Iterative MapReduce* (such as spark Mllib [16], SystemML [1]), *Parameter Server* (such as DMTK [9], DistBelief [7], Petuum [26], DMLC [15]), and *Data Flow* (such as Tensorflow[2]). In this way, industrial practitioners will get the basic idea of different distributed machine learning systems and be able to choose the best one that fits their needs. Furthermore, we will take the most widely used machine learning model as example to show case the power of distributed machine learning and some practical tricks inside, which including logistic regression on billion scale features and deep neural network with millions of parameters.

## 2. Audience

We think both academic researchers and industrial practitioners in the artificial intelligence domain will be interested in this topic, especially for those who are working on/around deep learning, and large-scale machine learning.

## 3. Prerequisite Knowledge

A preliminary understanding of machine learning and optimization will be helpful and enough.

## 4. Relevance

Distributed machine learning targets at solving big data and big model challenges. WWW is a conference that may attract many machine learning and data mining researchers who are seeking distributed machine learning solution to solve their application at internet scale. This tutorial provides them with exactly such knowledge.

## 5. TUTORIAL HISTORY

Distributed machine learning has been developing very fast in recent years. Many related tutorials have been given in various venues, including the following ones:

- Distributed Machine Learning, by Wei Chen, Taifeng Wang, Tie-Yan Liu, AAAI 2017.
- Recent Advances in Distributed Machine Learning, by Taifeng Wang and Wei Chen, ACML 2016.
- Large Scale Distributed Systems for Training Neural Networks, by Jeff Dean and Oriol Vinyals, NIPS 2015
- Scaling Machine Learning, by Alex Smola and Amr Ahmed, AAAI 2014
- Emerging Systems for Large-Scale Machine Learning, by Joseph Gonzalez, ICML 2014
- Scaling Up Machine Learning, by Ron Bekkerman, Misha Bilenko and John Langford, KDD 2011

Although there have been a few tutorials as listed above, many new advances have not been covered yet. We believe our new tutorial will be a timely addition to the existing ones. In addition, the way that we organize the tutorial and our demonstration of open-source distributed machine learning systems are very unique as compared to previous tutorials.

## 6. Duration and Sessions

This tutorial will be a half-day tutorial, with the following outline:

Overview (15')

Machine learning algorithms and corresponding optimization techniques (40')

1. Algorithms: regularized regression, neural networks, decision trees (10')
2. Optimization techniques:
    a. (Prox-)SGD, SCD, second-order methods (20')
    b. Variance reduction and Nesterov acceleration (10')

Parallelization mechanisms (40')

1. Data parallelism and model parallelism (10')
2. Synchronous and asynchronous parallelization (20')
3. Convergence rate analysis and comparison (10')

Advanced research topics (45')

1. Parallelism of combined optimization techniques (10')
2. Towards better synchronous/ asynchronous parallelization (15')
3. New communication/aggregation methods (20')

Open-source platforms for distributed machine learning (40')

1. Iterative MapReduce (SparkML as example) (5')
2. Parameter Server (DMTK as example) (10')
3. Data Flow (Tensorflow as example) (10')
4. Deep dive on distributed logistic Regression and distributed deep learning (15')

The total presentation time will be 180 minutes, and there will be one or two breaks in the middle.

## 7. REFERENCES


[1] Ghoting Amol, et al. SystemML: Declarative machine learning on MapReduce. ICDE 2011.

[2] Martın Abadi, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv:1603.04467 (2016).

[3] Olivier Bousquet and Leon Bottou. The tradeoffs of large scale learning. NIPS 2008.

[4] Stephen Boyd, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning, 2011.

[5] Jianmin Chen, Rajat Monga, Samy Bengio, and Rafal Jozefowicz. Revisiting distributed synchronous SGD. arXiv:1604.00981 (2016).

[6] Kai Chen and Qiang Huo, Scalable training of deep learning machines by incremental block training with intra-block parallel optimization and blockwise model-update filtering, ICASSP 2016

[7] Jeffrey Dean, et al. Large scale distributed deep networks. NIPS 2012.

[8] Aaron Defazio, et al. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. NIPS 2014.
[9] Fei Gao, et al. http://www.dmtk.io.
[10] Qirong Ho, et al. More effective distributed ml via a stale synchronous parallel parameter server. NIPS 2013.
[11] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. NIPS 2013.
[12] Guolin Ke, et al. A Communication-Efficient Parallel Algorithm for Decision Tree, AAAI 2017.
[13] John Langford, et al. Sparse online learning via truncated gradient. NIPS 2009.
[14] Jason Lee, et a;. Distributed stochastic variance reduced gradient methods. arXiv:1507.07595 (2015)
[15] Mu Li, et al. Parameter server for distributed machine learning. Big Learning Workshop, 2013.
[16] Xiangrui Meng, et al. Mllib: Machine learning in apache spark. JMLR 2016.
[17] Qi Meng, et al. Asynchronous Accelerated Stochastic Gradient Descent, IJCAI 2016.
[18] Qi Meng, et al. Asynchronous Stochastic Proximal Optimization Algorithms with Variance Reduction, AAAI 2017.
[19] Arkadi Nemirovski, et al. Robust stochastic approximation approach to stochastic programming. In SIAM Journal on Optimization, 2009.
[20] Yurii Nesterov. Introductory lectures on convex optimization, Springer Science & Business Media, 2004.
[21] Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM Journal on Optimization, 2012.
[22] Alexander Rakhlin, et al. Making gradient descent optimal for strongly convex stochastic optimization. ICML 2012.
[23] Peter Richtarik and Martin Takac. Iteration complexity of randomized block coordinate descent methods for minimizing a composite function. Mathematical Programming, 2014.
[24] Shizhao Sun, et al. Ensemble-Compression: A New Method for Parallel Training of Deep Neural Networks, arXiv:1606.00575 (2016)
[25] Shuxin Zheng, et al. Asynchronous Stochastic Gradient Descent with Delay Compensation for Distributed Deep Learning, arXiv preprint (2016)
[26] Eric P. Xing, et al. Petuum: A new platform for distributed machine learning on big data. IEEE Transactions on Big Data, 2015.
[27] D. Gabay and B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximation, Computers & Mathematics with Applications, 1976.
[28] DC Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. Mathematical programming, 1989.
[29] R. H. Byrd, S.L. Hansen Jorge Nocedal, Y. Singer, A Stochastic Quasi-Newton Method for Large-Scale Optimization, SIAM Journal on Optimization.
[30] M. Frank, P. Wolfe, An algorithm for quadratic programming, Naval Research Logistics Quarterly, 1952.
[31] Jaggi, Martin, Revisiting Frank–Wolfe: Projection-Free Sparse Convex Optimization, Journal of Machine Learning Research, 2013.
[32] Sutskever, Ilya, et al. On the importance of initialization and momentum in deep learning. ICML 2013.
[33] Ruiliang Zhang, James T. Kwok, Asynchronous Distributed ADMM for Consensus Optimization, ICML 2014.
[34] Reddi, Sashank J., et al. On variance reduction in stochastic gradient descent and its asynchronous variants. NIPS 2015.
[35] Jason Lee, et al. Distributed stochastic variance reduced gradient methods. arXiv:1507.07595 (2015)