Machine learning has become a primary mechanism for mining structured information and knowledge from data collections, turning them into automatic predictions and actionable hypotheses for diverse applications. With the increasing data from various sources, conventional machine learning research and development are now challenged by the growing prevalence of Big Data. The rise of Big Data is also being accompanied by an increasing appetite for more complex models with billions to trillions of parameters. Training Big models over Big Data is beyond the storage and computation capability of a single machine, and this gap has inspired a growing body of recent studies on distributed machine learning, where models are trained over commodity machines by partitioning both data and models into multiple parts. The complexity of statistical problems in the machine learning area and the system problems for distributed computations lead an emerging demand for distributed machine learning systems.

Given the importance and real demands of distributed machine learning, many of the existing platforms have provided their solutions. Data-flow systems, like Hadoop and Spark [1], simplify the programming of distributed algorithms and the integrated libraries, Mahout and Mllib, offer abundant ready-to-run machine learning algorithms. But they lack efficient mechanisms for parameter sharing in distributed machine learning. Petuum [2] integrates the parameter server architecture and the delayed synchronization protocol to tackle the problem of learning on Big Data and Big models. However, Petuum lacks the ability of fault tolerance to guarantee successful running in the production environment. ParameterServer [3] also builds a parameter server architecture and exploits live replication of parameters to support hot failover of servers. But its performance suffers dramatical degradation when dealing with dense data. TensorFlow [4]
employs data-flow programming, automatic derivation and GPU to simplify and accelerate the training of deep neural networks. But TensorFlow is designed to accelerate the computation-intensive tasks. It cannot handle sparse graphs with billions of nodes.

More recently, Cui’s group has developed a new distributed system, named Angel [5], to solve the problem faced by distributed machine learning. Angel employs hybrid parallelism to achieve both scalability and high performance. Parameter server architecture and efficient parameter pull-and-push operations have been established in Angel to improve the performance of model synchronization. Moreover, Angel can reinforce the performance of other machine learning systems by providing the service of model paralleling and asynchronous updates. To guarantee stable running in the production environment, Angel integrates mechanisms for fault tolerance and data management of training data. They also proposed a new distributed optimization algorithm, called DYNSGD, for Angel to accelerate the training speed of machine learning algorithms in the heterogeneous environment [6]. DYNSGD dynamically maintains a learning rate for each worker by incorporating the parameter staleness. By assigning smaller weight to a worker with large staleness, DYNSGD can help alleviate the impact of stragglers.

The Angel system has already been deployed in a world-leading internet company, Tencent, to support various business applications. A set of efficient machine learning algorithms have been designed and implemented in Angel, such as Gradient Boost Decision Tree (GBDT), Latent Dirichlet Allocation (LDA), Logistic Regression (LR) and so on. These algorithms are fully optimized to handle large data and high-dimensional models by exploiting either hybrid parallelism or the model synchronization mechanisms provided by Angel. Compared with the existing systems, such as Petuum [2], ParameterServer [3] and Tensorflow [4], Angel has some promising features such as the guarantees for running in the production environment and the ability to support a wide range of machine learning algorithms.

So far, there have been some products that utilize new hardware to accelerate the training of machine learning algorithms, especially deep learning. We hope that recent studies could stimulate more interest and effort in developing new techniques for distributed machine learning, thus providing better methods for large-scale data processing and mining.

Jianyong Wang
Department of Computer Science and Technology, Tsinghua University, China
E-mail: jianyong@tsinghua.edu.cn

REFERENCES
2. Xing EP, Ho Q and Dai W et al. SIGKDD 2015; 1335–44.