# Paper 9 Review: Map-Reduce for Machine Learning on Multicore

Tianyi Chen[1,*]

[1]Applied Mathematics and Statistic
306G, 100 Whitehead Hall MD 21210

**Abstract**

I reviewed this paper, and will summary it and provide some comments about it.

## 1 Summary of the paper

This paper mainly proposes a Map-Reduce framework, which can be suitable for a large class of machine learning algorithms for Multicore processors. The authors establish their framework based on one of their conclusions that any algorithm fitting Statistical Query Model may be written in a certain summation form. Even though such summation form doesn't depend on, but can be easily expressed in a map-reduce framework. Then a detailed architecture of their framework is offered, including engine, algorithm, master , etc. which is similar to original Map-Reduce framework. In the next sections, 10 algorithms which have been adopted to their framework with complexity analysis of algorithm time are run in the experiments. They realize the numerical experiments under both Multicore machines, and multiprocessor machines. The results show that Multicore framework can increase the computing speed dramatically, and is also faster than Multiprocessor machines, due to its lower cost communication among internal.

## 2 Numeric ratings

i. Significance and impact factor: 3.5 I can't find the journal or conference of it. But this paper has been cited about 750 times. It is a big number. So I think this paper is significant.

ii. Technical depth and quality of content: 3.5 It indeed provide a efficient framework, and several adopted algorithms on their framework.

[*]email: tchen59@jhu.edu

iii. Validation of claims:4 Some computational simulations are provided, but there is no experiment evidence provided to proof one of their conclusions that Multicore machine is faster than Multi-processor machines.

iv. Presentation and readability: 3 The architecture of this paper is clear. But the Figures are vague and not very readable.

## 3 Three strong points about the paper

i. The architecture of this paper is clear, which can let the reader understand easily.

ii.They provide many concrete adopted algorithms, which show how to realize their framework in realistic practice, and indeed prove the efficiency of the framework in the following numerical experiments.

iii. Their complexity analysis of algorithm time can offer some mathematical support to show multicore framework can reduce the algorithm time complexity, then improve the computation speed of algorithms.

## 4 Three weaknesses of the paper.

i. Lack of numerical experiment results to show the multicore machines are generally faster than multiprocessor machines, which cause this conclusion is not confident. Even though the cost of communication among multicore machines' internal is indeed less than multiprocessor machines, but other source of multicore machines, like internal memory, hard disk, are limited comparing with multiprocessor machines. These factors may affect their conclusion.

ii. The quality of the figures are so terrible that I even can't read the numerical values of the figures' ordinates. The authors don't provide clear annotation of their figures as well, which causes some difficulties to understand these pictures.

iii. Some sentences in this paper are vague. For example, the linear regression model, they say "We first compute sufficient statistic by summing over the data". But they don't give me what the sufficient statistic in this model is. It is not rigorous. Another example is they say one of advantages of the multicore framework comparing with multiprocessor machine is they can avoid the case of power-off. I don't agree with this argument, the multicore machine can't avoid emergence as well, and if emergence happened, we will suffer more from multicore machines, since all of the useful data are in a single machine. But for multiprocessor machine, useful data are stored in different machines.