



2019-04-18

Nankai-Baidu
Joint Laboratory



Parallel and Distributed
Software Technology Lab





$$Y = \{0, 1\} \quad \{-1, +1\}$$

- $Y = \{0, 1\}$ Andrew Ng
- $Y = \{-1, +1\}$ SNU C++ project:
 - ▣ 初始化 预处理数据: $y \cdot X^T \cdot w$ ($Y = \{-1, +1\}$)
- 增加一个算法: Nesterov with G



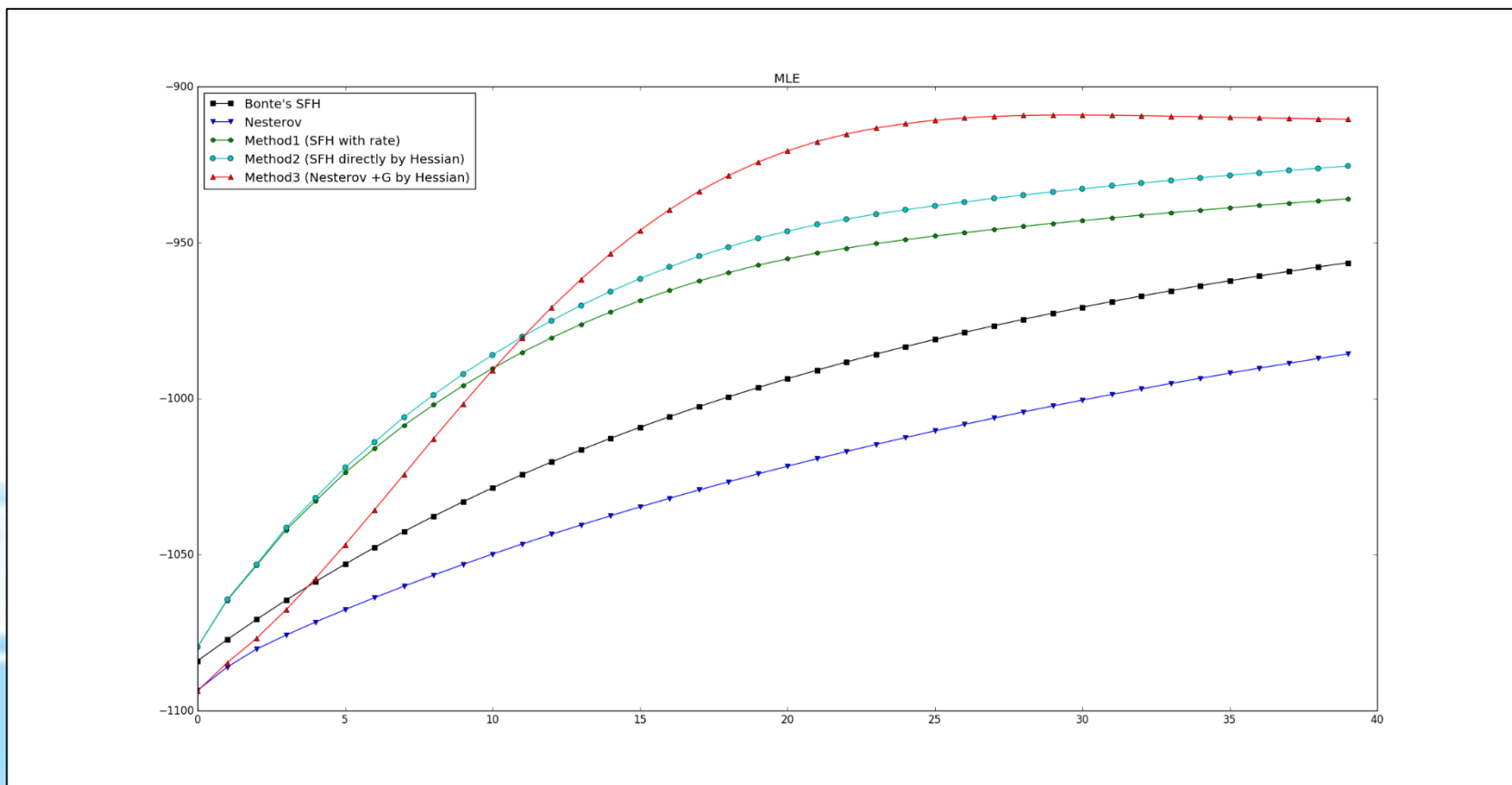
Bonte's Simplified Fixed Hessian

- $\text{weight} = 0$
- 使用的同态加密库: HEAAN
- 对(简化的)海森矩阵的替代矩阵求逆: HEAAN 自带函数
- 多项式拟合Sigmoid函数: 7次多项式
- (简化的)海森矩阵的替代矩阵(使对角矩阵的对角元非零)



iDASH2017

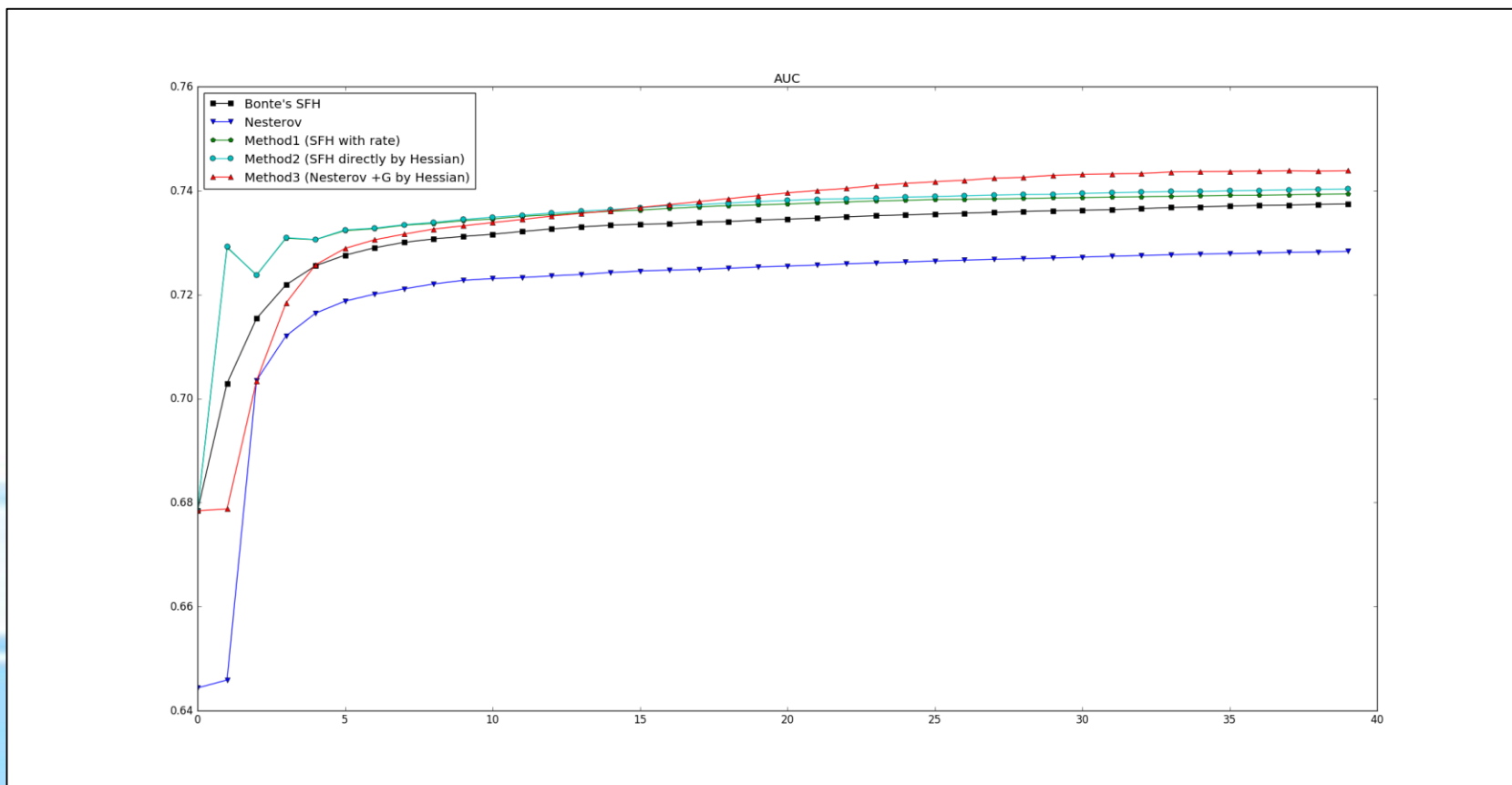
- 使用iDASH2017的数据，用多项式拟合Sigmoid函数：
$$0.5 + 0.214x - 0.00819x^3 + 0.000165861x^5 - 0.0000011958x^7$$





iDASH2017

- 使用iDASH2017的数据，用多项式拟合Sigmoid函数：
$$0.5 + 0.214x - 0.00819x^3 + 0.000165861x^5 - 0.0000011958x^7$$

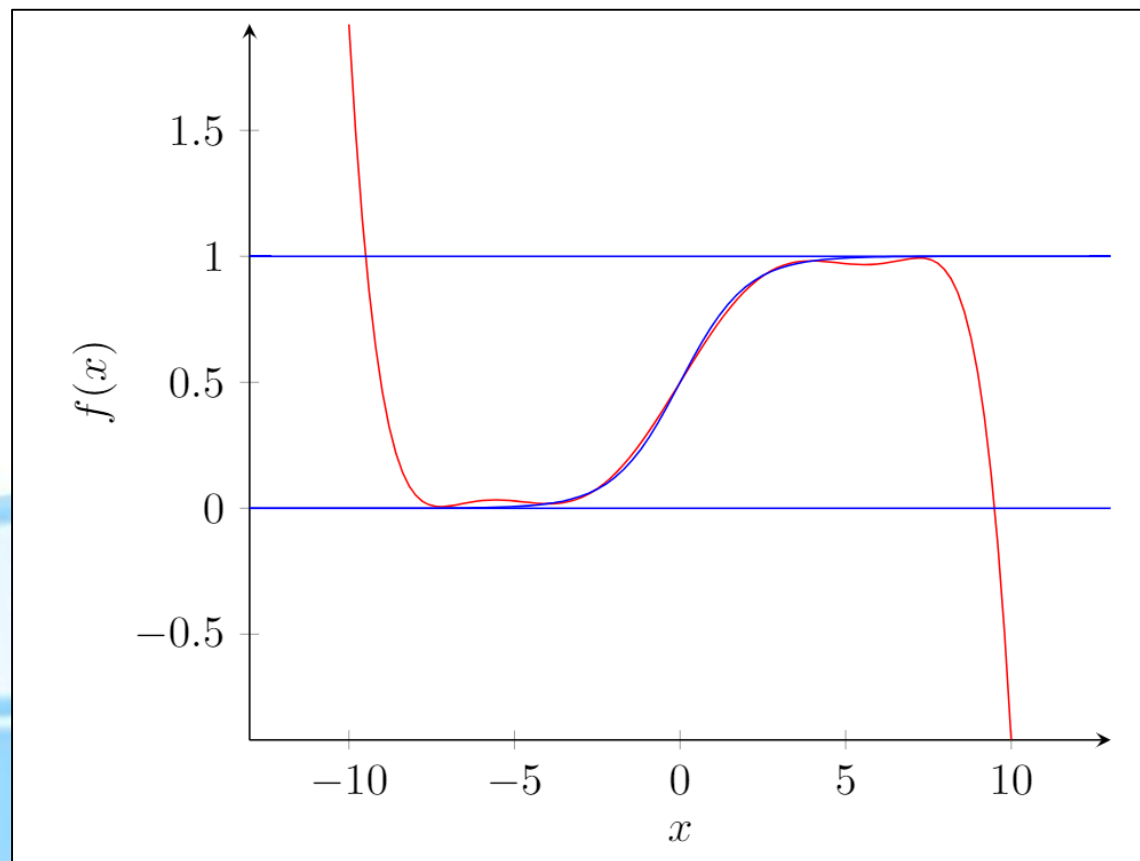




iDASH2017

- 使用iDASH2017的数据，用多项式拟合Sigmoid函数：

$$0.5 + 0.214x - 0.00819x^3 + 0.000165861x^5 - 0.0000011958x^7$$



Nankai-Baidu
Joint Laboratory



Parallel and Distributed
AI Technology Lab





MNIST

- 使用手写数字识别数据集MNIST的3(+1)8(-1)前500个样本数据:
Input range of Sigmoid : $[-13.3247343156, 2.00098210192]$
- python math overflow : 拟合Sigmoid的多项式 拟合的范围
- 数据集大小 内存不够 : SFH directly by Hessian 计算海森矩阵

Nankai-Baidu
Joint Laboratory

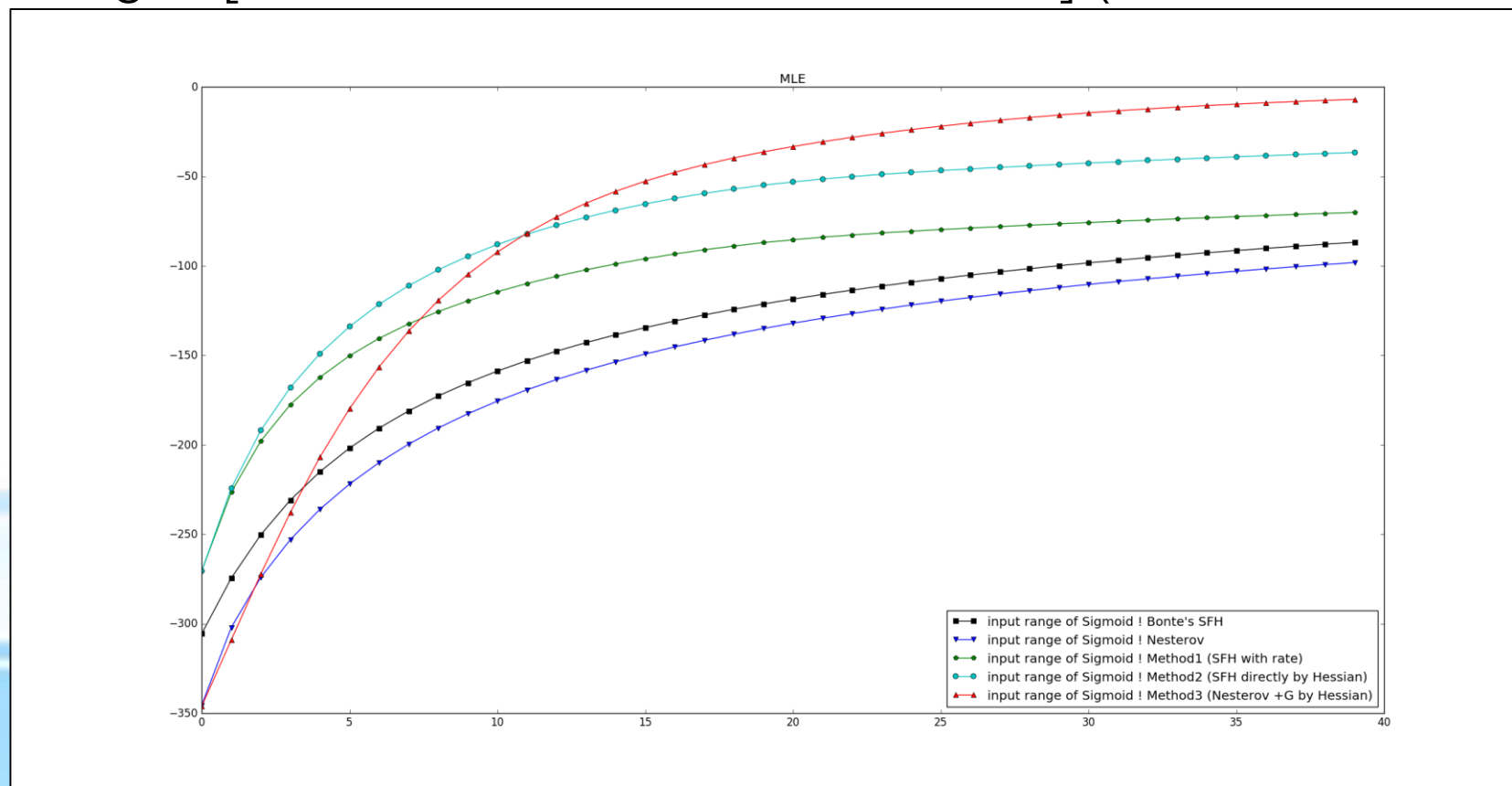
Parallel and Distributed
Software Technology Lab





MNIST

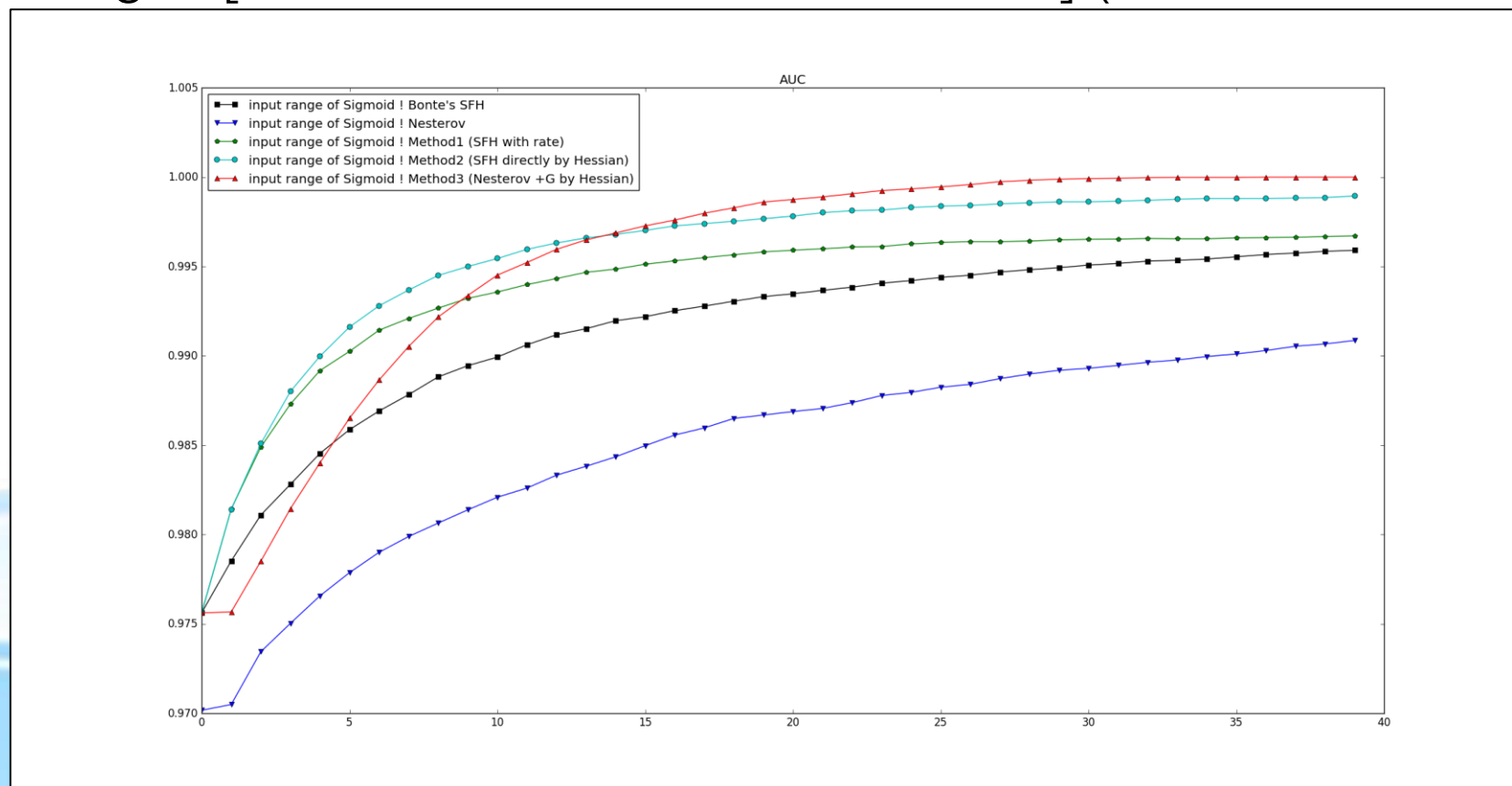
- 使用手写数字识别数据集MNIST的3(+1)8(-1)前500个样本数据:
Input range : $[-13.3247343156, 2.00098210192]$ (未使用拟合的多项式)





MNIST

- 使用手写数字识别数据集MNIST的3(+1)8(-1)前500个样本数据:
Input range : $[-13.3247343156, 2.00098210192]$ (未使用拟合的多项式)





Eclipse C++ project

- 数据 MNIST : 3(+1) 8(-1) >> .csv file (11982 * (14*14))
- HEAAN 整合到 C++ 项目中
- 进一步熟悉开源C++代码：尽量使用他们的函数，确保变量控制 (要使用他们的Nesterov算法，确保别的算法与之保持变量一致)
- 每一个算法的计算结果保存到一个.csv文件中，然后用python从5个.csv文件读取数据并画图

Nankai-Baidu
Joint Laboratory

Parallel and Distributed
Software Technology Lab

