Weekly Report
02/22/2015

Jingyu Deng

This Week

- Modified my previous random forest package
- Added a reader package to read data
- Added a high-level master program to implement Anil's algorithm
- Made some tests to see the performance

New Functions of my RF package

- predict(X)
    - X is the data set
    - return a vector containing result from each trees
- update_chunk(X, y)
    - inserting the latest data point, removing the oldest one.
- replace(flag, cnt)
    - flag is a bool vector with the length of n_trees
    - cnt is the number of trees which should be replaced
    - flag[i] == true means this tree need to replace
    - use last chunk to build a new random forest with cnt trees. Then use them to replace those trees whose flag is true.

DataReader Package

- __init__(num, data_path, target_path)
    - initialize the reader
- read_data_point()
    - read a data point from the file
    - return a dict containing data, target and index of this data point
- close()
    - close the reader

Master Program

- This master program is used to implement Anil's algorithm in high-level.
- Two versions are implemented:
    - A simple gradient descent algorithm
    - A complex correlation matrix based algorithm

Some Tests

- Because correlation matrix based algorithm may occur error due to no inverse of a singular matrix, I tested the gradient descent algorithm only.

- n_trees = 100

- chunk_size = 1000

- num of data points = 5000(no change point), 21000(1 change point), 61000(3 change points)

    - I didn’t test all data points since it cost a very long time if we need to train new trees

Some Tests

- I compared gradient descent + replace tree algorithm with naive no weight adjusting algorithm.
- I use mse to evaluate the performance

num of data points	gradient	naive
5000	0.02	0.02
20000	0.14	2.43
60000	0.08	0.08
110000	0.10	2.32

Question

- Why does naive algorithm get the same performance when n_data_points = 60000? I guess the trained RF is suitable for these later data by coincidence.

Next Step

- More experiments
- Improve our algorithm

Thanks