Midterm Report

LI, Ruizhe | 1155076990

Github: https://github.com/rzli6/ML-Storage.git (private)

1. Research Goal

The goal of the first part of summer research is to read papers, realize methods described in papers, try to rebuild programs, and finally compare my own result with the figures stated in the paper. Rebuilding codes of other scholars can help me start research and get an integrated picture about knowledge related to machine learning and disk storage.

Particularly, in the June, I managed to implement two papers assigned by professor: "Predicting Disk Replacement towards Reliable Data Centers" and "Improving Storage System Reliability with Proactive Error Prediction". These two papers gave me some insight to data storage, which was totally new to me, including how to use SMART attributes to monitor the health condition of a disk, how to explore data downloaded from Blackblaze Hard Drive website, and based on that, how to predict a whether a disk is going to fail. Moreover, at the same time, this experience sharpened my machine learning technique, which I just learned in last semester, including but not limited to using NumPy and Pandas to manipulate a csv file, implementing basic machine learning method on data set with the help of sklearn python kit, and evaluate a prediction using different professional scientific metrics such as f1 score, precision score and recall score.

In short, rebuilding others' method equipped me with basic knowledge of disk storage, developed my machine learning programming ability, and finally gave me not only a sketchy picture but also some insights of this summer research. It built me an essential foundation to construct my own experiment later.

2. Procedure and Achievements

2.1. Improving Storage System Reliability with Proactive Error Prediction

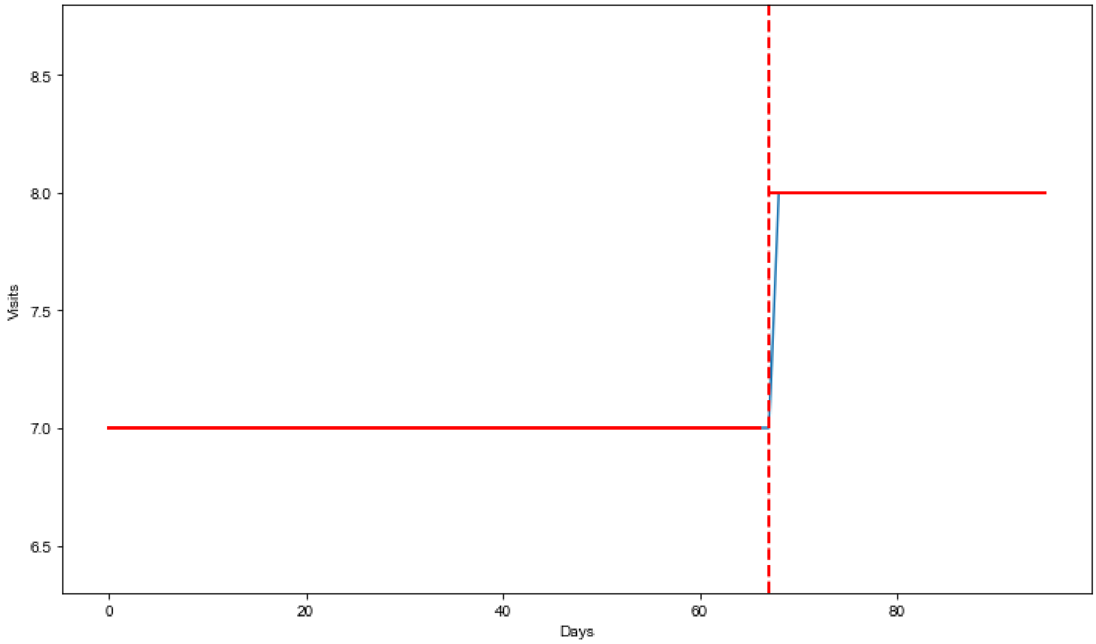
I started with "improving Storage System Reliability with Proactive Error Prediction". This paper aims at automatically predicting disk replacements based on historic disk replacement data. It totally contains five parts:

a) Selection of relevant SMART attributes

This part aims at finding the most relevant SMART attributes which later will be used as inputs of machine learning methods. The paper assumes that there should be a perceptible change on these relevant attributes, some time

Smart_5_raw: reallocated sectors count

prior to the failure of a disk. By counting how many fail disks have been detected a such change, we can determine attributes' relevance degree. Then, I selected the attributes whose relevance degree larger or equal to 1%, as paper did. There are 33 selected attributes, and 6 of them were not included in the paper, because either they are not available at the time paper was written, or they are too common to indicate anything about disks' health.



Smart_5_raw of JK1105B8GHWJZX has a change point on the 67th day.

b) Compact time series representation
Using the selected features, I grouped data by disk's serial number, and compacted the timeseries of each disk attributes. The timeseries were combined to a single number with the help of EWMA function in the Pandas package. After that, each disk had a health status code, either 1 for failure or 0 for health, together with the selected features represented by a single real number.

c) Class balancing via informative down sampling
The class balancing is essential because most disks are healthy. Even among the most prone-to-fail models, the failure rate can hardly be over 3%. This is where KMeans non-supervised was used. KMeans is able to separate the whole data set into a number of distinctive groups. And these groups are usually the most representative ones. We down sampled the healthy groups so that it nearly matched the size of the failed group.

d) Classification for disk replacement
In this section, I built several machine learning models from sklearn package.

Smart_5_raw: reallocated sectors count

Use the compacted balanced data as input, and binary health degree as output to train our models. And compare the results with the ones in the paper. I also tried to apply the same method to different disk model. And basically, we can get similar results.

model	Metrics	GBDT	SVM	DT	LR	RF	RGF
ST4000DM000	P	0.99	0.98	0.92	0.60	0.99	0.99
	R	0.95	0.96	0.87	0.61	0.95	0.94
	F	0.97	0.97	0.89	0.60	0.97	0.97
	Sd	0.02	0.01	0.03	0.30	0.02	0.02
ST31500541AS	P	1.00	1.00	0.90	0.34	1.00	1.00
	R	0.92	0.92	0.76	0.97	0.92	0.92
	F	0.96	0.96	0.82	0.50	0.96	0.96
	Sd	0.01	0.01	0.07	0.02	0.01	0.01
Hitachi HDS722020ALA330	P	0.99	0.93	0.89	0.71	0.99	0.99
	R	0.86	0.87	0.86	0.85	0.85	0.85
	F	0.92	0.89	0.86	0.77	0.91	0.91
	Sd	0.04	0.07	0.09	0.08	0.04	0.04
Hitachi HDS5C3030ALA630	P	1.00	0.97	0.63	0.46	1.00	1.00
	R	0.71	0.71	0.74	0.56	0.71	0.69
	F	0.83	0.82	0.67	0.50	0.83	0.81
	Sd	0.06	0.07	0.05	0.05	0.06	0.04

Comparing the result from different disk model, we can see their f1 score is fluctuating in a range, despite the oscillation the figures stay at a relatively high level, which means this method is also suitable for other disk models. And the vacillation may be caused by the disparate sample size.

e) Transfer learning
In the paper, the machine learning model trained on one disk model can be used to predict another similar disk model. The transfer learning method is said to be able to increase the prediction accuracy. However, this high-tech is still abstruse to me despite one week's effort.

2.2. Improving Storage System Reliability with Proactive Error Prediction
As soon as I finished most of the first paper, except for transfer learning part, I started the second paper. Compared to the first paper, this one is much easier to implement. The features have been selected by authors of this paper intuitively, and the preprocessing method have been elaborated clearly. The Min Max Scaler

Smart_5_raw: reallocated sectors count

is used to preprocess data. And it uses FPR and FNR metrics to measure the prediction accuracy, different from f1, precision and recall scores used in the first paper.

	smart_5_raw	smart_187_raw	smart_196_raw	smart_197_raw	Capacity (TB)	# Drives
ST4000DM000	0.00738119	0.0154702	0	0.0150657	4	29670
ST3000DM001	0.120719	0.177226	0	0.0702055	3	1168
Hitachi HDS5C3030ALA630	0.0323491	0	0.032132	0.0123752	3	4606
Hitachi HDS722020ALA330	0.133675	0	0.133675	0.0365151	2	4683
Hitachi HDS5C4040ALE630	0.0150376	0	0.0154135	0.0075188	4	2660
HGST HMS5C4040ALE640	0.0060317	0	0.0060317	0.00266517	4	7129
HGST HMS5C4040BLE640	0.000966806	0	0.000966806	0.00257815	4	3103

The overview of each model to be explored.

a) Preprocessing

I used the data from the Backblaze Hard Drive website. And just as the paper described, the raw data was grouped into weeks. Then, I picked out the disk data on each Sunday as the first part of input. The second part of input is the increasement of each counting value. The output in our model is 1 if the Reallocated Sectors Count has increased in the next 7 days, or 0 otherwise. The Min Max Scaler from sklearn package is used to preprocess the raw value of each attributes, so that none of the selected attributes will dominate the whole prediction.

b) Down Sampling

As what is described in the paper, the training set is a random subset of the whole data set. This method mostly requires no technique and suffers a large inconstancy.

c) Classification

With the help of sklearn python package, we trained our data on logistic regression, SVM, decision tree, neuron network, and random forest. And use FPR and FNR to evaluate each model.

2.3. Differences between two papers

a) Down sampling

Down sampling in the first paper used KMeans method. However, the second paper just randomly selected some disks from the whole data set. KMeans methods can improve the stability of our training set, but it will cost a long time.

b) Preprocessing

The first paper does not do normalization on the data, instead it uses both

Smart_5_raw: reallocated sectors count

raw data and normalized data provided by disk manufacturer. On the contrary, the second paper uses the raw data and implements a Min Max Scaler to do normalization.

c) Metrics

F1 score, precision and recall score are used in the first paper, while the second paper implements its evaluation using FPR and FNR. Both of them are popular metrics for unbalanced data.

3. Future

In the following several weeks, I will continue to rebuild some methods in the given papers and try to do some improvements based on the paper we have verified. Hopefully, at the end of the summer research, we can have our own paper.

Smart_5_raw: reallocated sectors count