Probabilistically Forecasting Health of Hardware in a large-scale system

Table of Contents

[Abstract 2](#_Toc98118001)

[1. Proposed Solution 3](#_Toc98118002)

[1.2 Pseudocode 3](#_Toc98118003)

[2. Detailed Implementation 4](#_Toc98118004)

[2.1 Data Collection 4](#_Toc98118005)

[3. Novelty and Inventive Step 6](#_Toc98118006)

[3.1 Novelty 6](#_Toc98118007)

[3.2 Inventive Step 7](#_Toc98118008)

[3.3 Advantages over existing method 7](#_Toc98118009)

[4. Former approach and exiting method 7](#_Toc98118010)

[5. Utility and Use Case 7](#_Toc98118011)

[5.1 Efficient inventory management 7](#_Toc98118012)

[5.2 Modelling Component Failures 8](#_Toc98118013)

# Abstract

Hardware failure prediction problem has already been solved in the industry using statistical and machine leaning methods. Usually, classical statistical techniques catered for small scale, low-dimensional data and machine learning is primarily concerned with producing prediction rules. Many machine learning algorithms for classification are in fact scoring classifiers: they output a prediction score and the prediction is obtained by comparing the score to a threshold. Moreover, the problem with practitioners is that they often focus on the value of the prediction but overlook its uncertainty.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **“Ordinary” Classification** | **Conformal Classification** | **Venn-Aber Classification** |
| Method | minimize the loss function by predicting correct labels | Label prediction confidence | Probability distribution over the set of *k* classes along with confidence measure |
| Publication | ~ 5 Papers in USENIX  (2015 - present) | SNIA 2019 Santa Clara (First Author) | N/A |

Table 1: Study of techniques available to solve binary classification problems (HDD failure prediction)

We propose a method which offers a principled way of assigning a probability to predictions and the end goal is to predict the probability distribution of the label, given the training set and the test object. A more practical application will be for early detection of SSD failure and for other risk sensitive hardware, where an intuitive and explainable interpretation is required.

# Proposed Solution

We show a high-level design of our proposed method.

A screenshot of a cell phone

Description automatically generated

Figure 1: Flow diagram for multi-probability prediction for classification (analytics engine)

## 1.2 Pseudocode

**Algorithm**: Multi-Probabilistic Classification Prediction

**Data**: training set Z(n-1), testing example xn

**Result**: predicted label ŷn, probability interval [In, un]

K <- |Y|;

for y = 0 to K - I do

for i =1 to n do

Ti <- Taxonomy ((x1, y1), . . . (xi-1, yi-1), (xi+1, yi+1), (xn, y), (xi, yi));

end

C <- ϕ

for i=1 to n do

if Ti = Tn then

C <- AddToSet(C, yi);

end

end

Py <- CalcFrequency(C);

end

Jbest <- FindBestColumn(P);

[In, un] <- FindInterval(Jbest, P);

Ŷn <- Jbest;

return ŷn, [In, un]

# Detailed Implementation

## 2.1 Data Collection

We pulled data from open-source SMART dataset.

We implemented the method on the existing dataset and calculated **Brier Score**. Lower the brier score, better the accuracy of model. This works well for binary classification which is our required task.

jitu@ubuntu:~/venn$ ./va-cv -t 1 -v 10 -b transformed\_data\_hdd.txt

CV Accuracy: 94% (141/150)

Probabilities: [81.2966%, 98.3333%]

**Brier Score: 0.0706**

Logarithmic Loss: 0.1427

Time cost: 18 s



A screenshot of a cell phone

Description automatically generated

Table 1: Snapshot of disk failure forecast in next 5 days

We obtain the probability of the label of the test object being FAILED given the historical dataset (probability distribution on the set of the labels), where only one of these probabilities will be the valid one. The result can be represented with a matrix, in which each row contains the probability distribution over the labels associated with the hypothetical label assignment.

* Choose as best column the column with the highest quality
* Output the label of the column as label prediction

|  |  |  |
| --- | --- | --- |
|  | **FAILED** | **NORMAL** |
| **FAILED** | 0.890598 | 0.489321 |
| **NORMAL** | 0.701968 | 0.243526 |

Table 2: Predicting label for New data point

In this example, the prediction is FAILED and the interval for the probability is:

**[0.890598, 0.701968]**

Since all Venn predictors are valid in the long run, the size of the probability intervals is very important.

* Interval size – The tighter the interval is, the more informative
* Accuracy – The predictive performance of the model is of course vital in all predictive modeling

We also explored the method on synthetically created dataset and obtained satisfactory results as shown on the figure below.

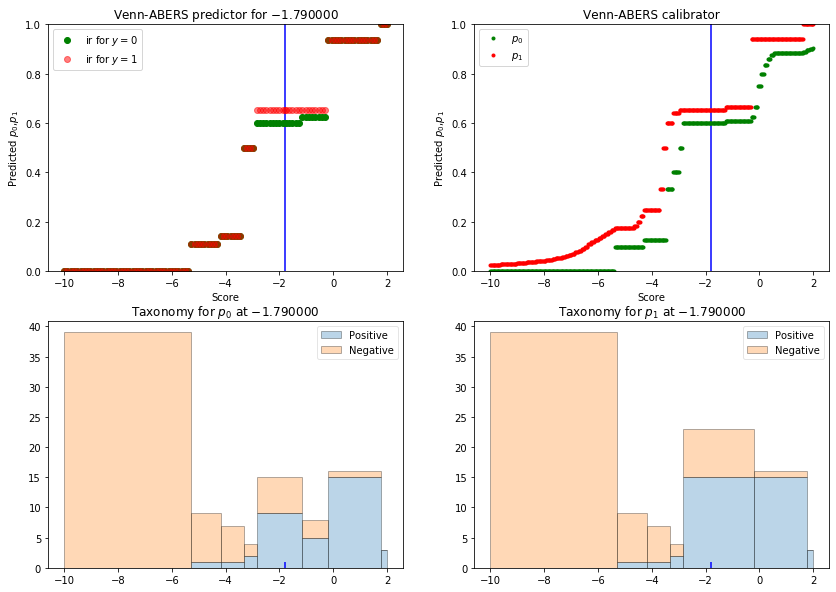


Figure 2: Multi-probability prediction for synthetic dataset with binary label

The plots show the taxonomies induced by the Venn-ABERS on the "calibration set plus (𝑠,0)(s,0)" on the left and "calibration set plus (𝑠,1)(s,1)" on the right.

Each bin corresponds to one category in the taxonomy. The bar heights show the counts of positive examples (in blue) and of negative examples (in beige).

In fact, we are only interested in the predicted probability in the bin in which the test example falls. This is the probability output by the Venn-ABERS predictor; 𝑝0p0 is taken from the taxonomy on the left and 𝑝1p1 from the taxonomy on the right.

# Novelty and Inventive Step

## 3.1 Novelty

* Probabilistic information on every prediction individually rather than estimate an error on all future examples as a whole
* Classification result is a prediction with its complementary probabilistic estimation which is more informative than a single prediction
* Method blends the learning and prediction together
* Much more confident and accurate prediction than the traditional methods (tighter prediction interval size)

## 3.2 Inventive Step

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Prior Art** | | |  | **Inventive Step** | |
| Classification | **Ordinary** | **Probabilistic** | **Conformal** | **Venn-Aber** | |
| Method | Single prediction | Single probability distribution | Set of predictions | Set of Probabilities | |
| Validity | Depends on data | Guaranteed under statistical assumption | Guaranteed | Guaranteed in online setting | |
| Efficiency | Algorithm dependent | Algorithm dependent | Dependent on Non- Conformity measure | Dependent on Taxonomy used | |

## 3.3 Advantages over existing method

* Complementing a prediction with its probability can enable better decision making
* Venn Predictors are (multi) probabilistic predictors with validity guarantee
* Venn-ABERS Predictors are Venn Predictors that can be applied on top of a Scoring Classifier

# 4. Former approach and exiting method

Many machine learning algorithms for classification are in fact scoring classifiers: they output a prediction score s(x) and the prediction is obtained by comparing the score to a threshold.

Methods are such as Naive Bayes classifier, **Logistic Regression** and Platt Scaling, which **can make probabilistic estimations for individual prediction**. However, these algorithms are only applicable under strong assumptions

Most of the publications in conferences like USENIX/ACM (Transactions on Storage) have used only variants of Ordinary binary classifiers (e.g. Random Forest, SGD, SVM) and comparative studies. This only gives point prediction.

There is a variety of machine learning algorithms that can make probabilistic predictions, which is to assign a probability distribution of the label for a new object. For e.g.

* **Probably approximately correct (PAC learning)**: the shortcoming is that its estimated bound of error is usually too loose to tell us any interesting information on the data set, especially when the dataset we deal with is not so large.
* “**hold-out estimate**”: drawback is that it rigidly separates the process of learning and prediction, and the estimates rely on the hold-out examples. On a similar note, both PAC learning and hold-out estimate, make probabilistic estimations for a whole group of predictions rather than provide information on each prediction

# Utility and Use Case

The proposed method can be used for any hardware failure diagnostics. A few of the use case is mentioned below:

## 5.1 Efficient inventory management

In a large-scale datacenter (on-premise or cloud), there are millions of disk drives (HDD/SSD), and there are various advanced techniques for replacing the failed disks or disks going to fail soon.

Although, the percentage of failure is approximately 4 – 8 % annually. Even then, it would be better if we could prioritize which all failed disks / soon to fail disks needs more attention and accordingly make the decision.

This is achieved from the suggested model where we get multi-probability distribution and a decision is made accordingly.

## 5.2 Modelling Component Failures

The proposed method is a use case of binary classification and can be extended to various components failures like CPU, Battery and Network. A dataset for each component should be created and analysed, and further fed to the model for forecasting.