

Python in the enterprise

Machine-learning-based silicon sensor quality evaluation

Dawid Gerstel, Marcin Lis, Michał Labuz, Konrad Zuchniak

July 16, 2016

1 Introduction

The goal was to implement software for classifying silicon sensors as either good or bad ones. The k-nearest neighbours algorithm was to be used to assess a sensor's response based upon a few attributes in the so-called feature space. These, in turn, were to be obtained in a preprocessing from sensor's response histograms in the ROOT framework.

2 Project organisation

At the beginning, the tasks were assigned to the authors of this report as follows:

task	person responsible
Analysis of the sensor's response histograms	all
Choosing specific ML algorithm	all
Histograming the measurements in the feature space	Dawid G.
Extracting feature space attributes from the histograms	Dawid G.
ML algorithm implementation	Michał L. & Konrad Z.
Testing accuracy of the algorithm	Marcin L.
Results analysis and deciding about further plans	all

Table 1: Summary of the tasks assigned to the team members at the project's start

3 Project evolution

The input data come from the real silicon sensors of the LHCb's Vertex Locator. Each sensor taken into account has 2048 channels at which noise has been observed producing 2-dimensional histograms as in the Fig. ???. Then, the former histogram was projected onto the y-axis and 1-dimensional histograms representing sensor's response (irrespective of its channels) were created (Fig. ???). Then, from these distributions four parameters have been extracted, namely: mean, rms, skewness and kurtosis. These comprised the attributes for a machine learning algorithm. For any supervised learning it is mandatory to provide an adequate class label. Initially, we decided to approach classification as binary (either good or bad) with a perspective to develop a continuous and probabilistic classification to either category later on (if at all).

After a brief analysis we agreed that all 42 (?) sensors are acceptable and we labelled them as '1' (good) class.

Next, we simulated response of 'bad' sensors by shifting attributes of the 'good' ones... (specifically how...)

A table summary of the results

k neighbours — % correct — train / test ...

To make the algorithm more useful a probabilistic approach needs to be taken in which a percentage that a given sensor is acceptable is inferred.¹ To satisfy this requirement we came up with an idea of deriving quality estimate (0-100 % range) based on a sensor's distance (in the feature space) from the most representative one – so-called the benchmark sensor – which could simply be represented by averaging all attributes of all good sensors dataset.

¹This objective is, unfortunately, beyond the scope of the knn classifier.