Machine Learning Engineer Nanodegree

CNC Mill Tool Wear Predictive Analysis and Building Closed-Loop Control System

Saravanan Balasubramanian
September 27^{th}, 2018

Proposal

Domain Background

Large industrial/manufacturing facilities operate multiple equipment with thousands of sensors/variables driving the manufacturing process. The equipment used in industrial/manufacturing process are subject to wear and tear. The tool wear directly affect the quality of the yield and introduces unscheduled production shut-down in shop floors leading to economical impacts to business.

Milling is a cutting process that uses cutter to remove material from the surface of a workpiece. Computer Numerical Control (CNC) milling machines are used widely in the milling process. The cutters in the CNC machines are subject to gradual failure due to regular operation. This project aims to reduce economical impacts to business caused by tool wear with CNC Milling machines. Manufacturing domain is fast adopting machine learning and artificial intelligence; the industry is transforming towards intelligence/smart manufacturing leveraging machine learning & data science capabilities. In their research paper "Tool Breakage Detection using Deep Learning" (reference 1), the authors propose applying Deep Learning for detecting CNC tool breakage on time.

I have reasonable experience working in a larger manufacturing environment implementing and supporting Manufacturing Execution Systems (MES). One of the key objectives of such MES systems are to facilitate the Operations & Quality personnel to improve yield quality and eliminate unexpected outages which cost millions of dollars. As I deal with multiple problems around forecasting anomalies detection, I am personally motivated for investigating the tool wear problem in this domain.

Problem Statement

Larger manufacturing facilities operate the facility through Distributed Control Systems (DCS). These DCS systems control all equipment by sending appropriate signals and drive automated manufacturing. Such facilities usually have a Historian which collects all data from DCS, i.e., data from all sensors would be collected in the Historian database.

For this project, I am trying to get data from CNC machines/Historian and predict tool wear so that Operations personnel could see the trend and take preventive actions. Predicting CNC machine tool wear is a binary classification problem. Multiple machine learning algorithms can be applied for this binary classification problem and they have quantitative metrics such as accuracy, precision, f1 score etc. to measure the results.

Datasets and Inputs

The CNC Milling Dataset is obtained from Kaggle donated by Sharon Sun. This data set was generated by running series of experiments on CNC milling machine in the System-level Manufacturing and Automation Research Testbed (SMART) at the University of Michigan.

Dataset link : https://www.kaggle.com/shasun/tool-wear-detection-in-cnc-mill/home

Details: General data from a total of 18 experiments are given in train.csv and includes:

Inputs (features)

- No : experiment number
- material : wax
- feed_rate : relative velocity of the cutting tool along the workpiece (mm/s)
- clamp_pressure : pressure used to hold the workpiece in the vise (bar)

Outputs (predictions)

- tool_condition : label for unworn and worn tools
- machining_completed : indicator for if machining was completed without the workpiece moving out of the pneumatic vise
- passed_visual_inspection: indicator for if the workpiece passed visual inspection, only available for experiments where machining was completed

Time series data was collected from 18 experiments with a sampling rate of 100 ms and are separately reported in files experiment_01.csv to experiment_18.csv. Each file has measurements from the 4 motors in the CNC (X, Y, Z axes and spindle).

The experiments gathered ~25300 data points; experiments conducted with worn tools are ~13,300 data points and the unworn tools have ~12,000 data points. So the data set is reasonably balanced for a binary classification problem. As these are time series data, the data set will be split in chronological order for the training/validation/test sets to avoid look ahead bias.

The features available in the machining datasets are:

- X1_ActualPosition: actual x position of part (mm)
- X1_ActualVelocity: actual x velocity of part (mm/s)
- X1_ActualAcceleration: actual x acceleration of part (mm/s/s)
- X1_CommandPosition: reference x position of part (mm)
- X1_CommandVelocity: reference x velocity of part (mm/s)
- X1_CommandAcceleration: reference x acceleration of part (mm/s/s)
- X1_CurrentFeedback: current (A)
- X1_DCBusVoltage: voltage (V)
- X1_OutputCurrent: current (A)
- X1_OutputVoltage: voltage (V)
- X1_OutputPower: power (kW)
- Y1_ActualPosition: actual y position of part (mm)
- Y1_ActualVelocity: actual y velocity of part (mm/s)
- Y1_ActualAcceleration: actual y acceleration of part (mm/s/s)
- Y1_CommandPosition: reference y position of part (mm)
- Y1_CommandVelocity: reference y velocity of part (mm/s)
- Y1_CommandAcceleration: reference y acceleration of part (mm/s/s)
- Y1_CurrentFeedback: current (A)
- Y1_DCBusVoltage: voltage (V)
- Y1_OutputCurrent: current (A)
- Y1_OutputVoltage: voltage (V)
- Y1_OutputPower: power (kW)
- Z1_ActualPosition: actual z position of part (mm)
- Z1_ActualVelocity: actual z velocity of part (mm/s)
- Z1_ActualAcceleration: actual z acceleration of part (mm/s/s)
- Z1_CommandPosition: reference z position of part (mm)
- Z1_CommandVelocity: reference z velocity of part (mm/s)
- Z1_CommandAcceleration: reference z acceleration of part (mm/s/s)
- Z1_CurrentFeedback: current (A)
- Z1_DCBusVoltage: voltage (V)
- Z1_OutputCurrent: current (A)
- Z1_OutputVoltage: voltage (V)
- S1_ActualPosition: actual position of spindle (mm)
- S1_ActualVelocity: actual velocity of spindle (mm/s)
- S1_ActualAcceleration: actual acceleration of spindle (mm/s/s)
- S1_CommandPosition: reference position of spindle (mm)
- S1_CommandVelocity: reference velocity of spindle (mm/s)
- S1_CommandAcceleration: reference acceleration of spindle (mm/s/s)

- S1_CurrentFeedback: current (A)
- S1_DCBusVoltage: voltage (V)
- S1_OutputCurrent: current (A)
- S1_OutputVoltage: voltage (V)
- S1_OutputPower: current (A)
- S1_SystemInertia: torque inertia (kg*m^2)
- M1_CURRENT_PROGRAM_NUMBER: number the program is listed under on the CNC
- M1_sequence_number: line of G-code being executed
- M1_CURRENT_FEEDRATE: instantaneous feed rate of spindle

Solution Statement

The solution to this problem would be to train multiple binary classifiers for this dataset and find the best one that generalize this dataset very well by measuring it. The dataset includes multiple files and includes some unwanted features. The dataset has to be prepared from the given files, relevant features selected, then training set and test set will be created.

As there are multiple classifiers that can be applied to this problem and since we don't know which one would be a good fit for this problem, I will evaluate the following algorithms.

- Logistic Regression (LR)
- Decision Trees
- Support Vector Machines (SVM)
- Gaussian Naïve Bayes (NB)
- Random Forests (RF)

The accuracy of the algorithm will be measured using the metrics outlined in Evaluation Metrics section. I will also make recommendations about deploying this model in real time production and building closed loop control systems where the model prediction is fed back to the MES systems to control/influence production.

Benchmark Model

The given dataset fits good with binary classifiers such as Logistic Regression. So I will be using Logistic Regression as the benchmark model and will try to produce better results than Logistic Regression with hyperparameter tuning.

Evaluation Metrics

I will evaluate the model accuracy by evaluating it with F1 score (derived by precision & recall using a confusion matrix). The model will also be evaluated against unseen data using a test set.

Confusion Matrix	Actual Positive	Actual Negative
Predicted Positive	TP	FP
Predicted Negative	FN	TN

Project Design

High level workflow for approaching a solution to this problem:

1. **Data Cleaning** : Consolidate data available on multiple files into one dataset, handle missing values.

2. **Visualize Data** : Visual representation of data to see ranges and visible patterns of predictors and variables

3. **Feature Selection** : Find relevant features, drop irrelevant features, enrich data with new features required.

4. **Training Set / Test Set Preparation** : Prepare training set and test set.

5. **Model Selection** : Train multiple binary classifiers against the training set to find out the best algorithm for this dataset.

6. **Model Tuning** : Fine tune the selected algorithm to increase performance without overfitting

7. **Testing** : Test the model against the testing data set

8. **Recommendations** : Make recommendations to business on deploying the model real-time and feeding back to production (closed loop control system)

References

1. Guang Li, Xin Yang, Duanbing Chen, Anxing Song, Yuke Fang, Junlin Zhou. "Tool Breakage Detection using Deep Learning", 2018.
2. Milling (machining) process Wikipedia https://en.wikipedia.org/wiki/Milling_(machining)
3. University of Michigan's SMART CNC Mill Tool Wear data set in Kaggle https://www.kaggle.com/shasun/tool-wear-detection-in-cnc-mill/home
4. Closed-loop (feedback) control systems – Wikipedia https://en.wikipedia.org/wiki/Control_theory#Open-loop_and_closed-loop_(feedback)_control