AAIA'16 Data Mining Challenge: Predicting Dangerous Seismic Events in Active Coal Mines

Genetic Algorithm Approach

Team: **dotjabber (Maciej Kowalski - mkowalski@opi.org.pl)**
Highest preliminary evaluation score: **0.9012**

Data normalization (variant 1)

Data were normalized using two different approaches. Fist one assumed that all values should be taken into account separately, so there were no dependencies between vector parts (for example 24h time series were treated as independent values). At the front of vector, an encoded set of metadata has been added. The numerical values of input vectors were rescaled to fit the range <0..1>. Other values were encoded as hot-one sub-vectors as following:

main_working_id	metadata	total_bumps_energy	total_tremors_energy	total_destressing_blasts_energy	total_seismic_energy	latest_progress_estimation_l	latest_progress_estimation_r	latest_seismic_assessment	latest_seismoacoustic_assessment	latest_comprehensive_assessment	latest_hazards_assessment	latest_maximum_yield	latest_maximum_meter	time series 1 (count_e2)	...	time series n (avg_difference_in_genergy)

Data normalization (variant 2)

Second approach to data normalization assumed, that there is a pattern in time series, so every part of data which contained time series were replaced by linear approximation (ax + b), so 24h values were at the end replaced by two numbers (a, b) of approximation. All the values and metadata were scaled and hot-one coded as it was done in variant 1.

main_working_id	metadata	total_bumps_energy	total_tremors_energy	total_destressing_blasts_energy	total_seismic_energy	latest_progress_estimation_l	latest_progress_estimation_r	latest_seismic_assessment	latest_seismoacoustic_assessment	latest_comprehensive_assessment	latest_hazards_assessment	latest_maximum_yield	latest_maximum_meter	a, b parameters of linear regression for time series 1(count_e2)	...	a, b parameters of linear regression for time series n (avg_difference_in_genergy)

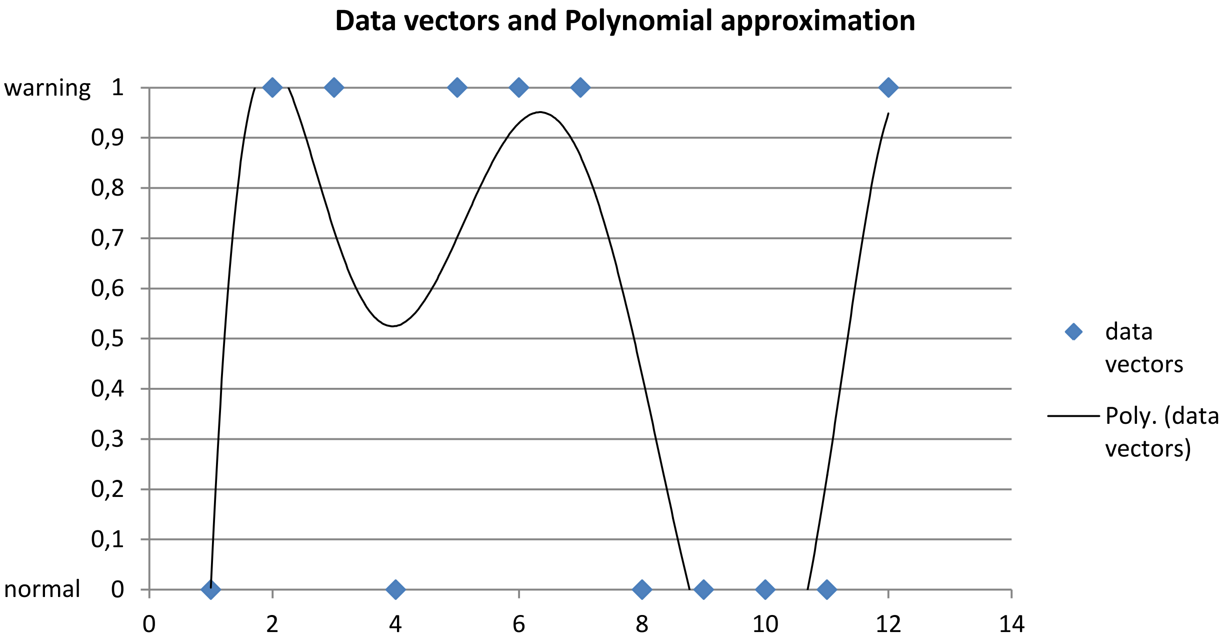
Polynomial approximation

This solution assumes that there exists a polynomial equation of given degree, which can approximate training data points:

$$P = \sum_{i=0}^{N} c(rx_i + a)^n + t$$

$$R = \begin{cases} 1 \; if \; P > 1 \\ P \; if \; 0 \leq P \leq 1 \\ 0 \; if \; P < 0 \end{cases}$$

Where *P* is polynomial response value, and *R* is restricted polynomial response value, respecting range <0..1>. Parameters like *c*, *r*, *a*, *t*, *n* are established during process of selecting the best genome. The value x_i corresponds to the i^{th} value of data vector. At the end of selection process data points can be approximated with the least error possible as shown in the example below. For real data, there is 500+ dimensional space in which data points are approximated in the same way.



The parameters found can be used further to discover some information about seismic events – vector values with non-zero polynomial parameters assigned by genetic algorithm may be considered as essential in modeling of the seismic processes.

Genetic algorithm – Genome

The way to find polynomial parameters is to use generic algorithm. In the described solution, an individual (genome) which is processed by algorithm is constructed as following:

```
public class Individual implements Comparable<Individual> {
	private double[] coeficients; // c in the formula
	private double[] exponents; // n in the formula
	private double[] regulators; // r in the formula
	private double[] addictors; // t in the formula
	private double[] totalitarians; //t in the formula
	...
}
```

Genetic algorithm – Picking data

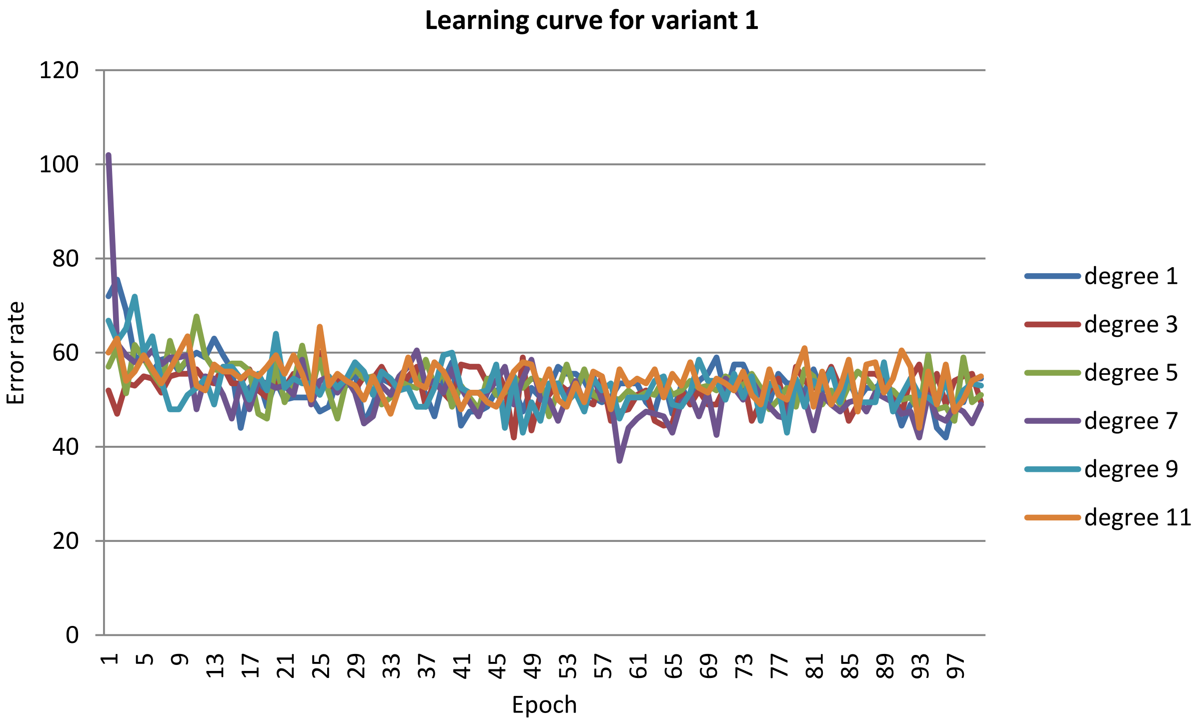
The data assigned to the competition is not equally distributed. For ~130 000 training vectors, only ~3 000 were labeled as "warning", the rest were labeled as "normal". To overcome this, a 25% of ~3 000 of "warning" data is picked randomly (around 750 vectors), and the same number of data is then picked from "normal" cases (also 750 vectors). Each genetic epoch is presented a new randomly picked pair of "normal" and "warning" sets.

Genetic algorithm – Score

Each breed of polynomial genome is tested on a set of 1 500 vector dataset and has a score assigned, using mean square error measure. To avoid fluctuations from the data picking scheme, both current-generation and previous-generation scores are take into account giving the final result as mean of these two.
For every epoch, 10 best genomes were taken from previous epoch, and 10 more were generated randomly.

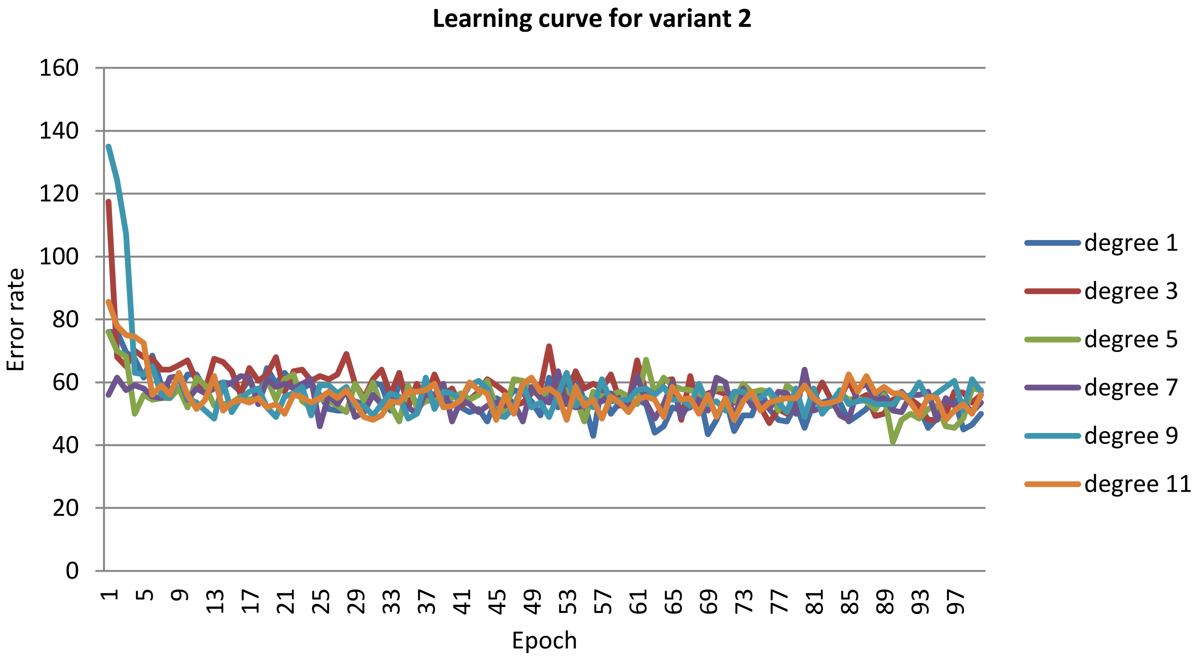
Generating genome (variant 1)

Tests were conducted using 6 different maximal polynomial degrees: 1, 3, 5, 7, 9, 11. Algorithm generated 300 epochs with 20 individuals per epoch. By using these parameters, generic algorithm produced a set of individuals, best of which drew following learning curve. The best genome has been generated for polynomial degree = 7.



Generating genome (variant 2)

Again tests were made using 6 different maximal polynomial degrees, as it was described in testing of variant 1. With Variant 2 case, it gave the same results as Variant 1 data.



Code

The genetic algorithm has been written using java language and is available on github: https://github.com/dotjabber/aaia16/

Results for both variants are published on project's "results" directory: https://github.com/dotjabber/aaia16/results/