Feature Transformation in Pattern Recognition

Sudhir K Singh

School of Computer Science and Electronic Engineering

Bangor University

Supervisor: Prof. Ludmila I Kuncheva

Aim:

Machine learning and pattern recognition algorithms depend heavily on the type of data fed into them. Data quality determines the performance and productivity of an algorithm.

The original features may be related to the class variable in a non-linear way. For example, "x^2" might be a better predictor than "x".

The purpose of this project is to explore, understand, and analyse the results generated from random feature transformations e.g. "x" to x^2, log(x), sin(x), 1/x etc, and monte carlo experiment.

Introduction:

Feature transformation is a well-known technique for creating a new set of features, using the existing features, to improve the classification accuracy. It can either be supervised or unsupervised. Supervised feature transformation involves discretising of continuous features into discrete features using the class label information. Unsupervised feature transformation consists of feature mapping, e.g. "x" to x^2, log(x), sin(x), 1/x.

Feature transformation can help with data variance stabilisation, creating better feature space separation between the classes and placing features on equal scales.

Software/Technology:

"Matlab" will be used for all the calculations in this project.

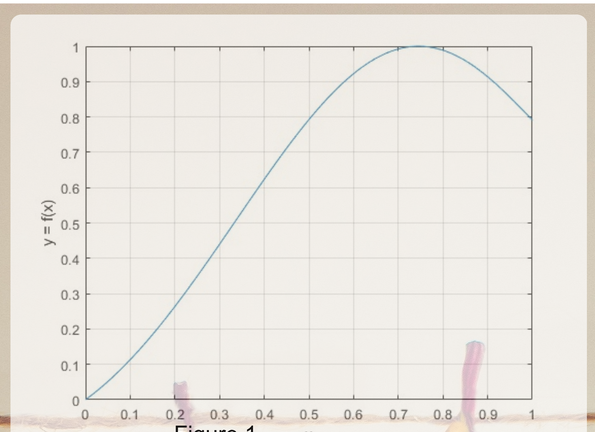

Figure 1

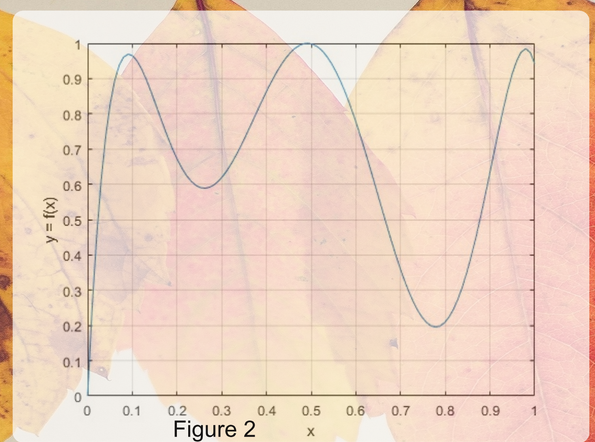

Figure 2

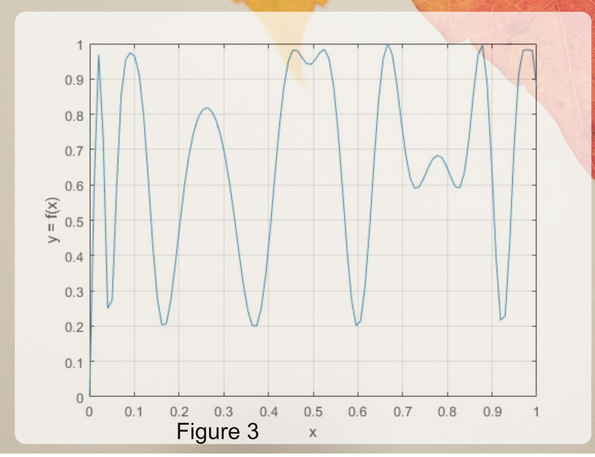

Figure 3

Method:

Uniformly distributed data of 100 values was created using "x = *linspace(0,1,100)*", and three random transformations were applied on it. Figure 1, 2, and 3 are the plotting of the data results after the transformations. Similar random transformations will be applied on each feature of the real datasets.

Initial transformation will be done on the data set "x" with the function "$y = a_6x^6 + a_5x^5 + ... + a_1x + a_0$", where the polynomial degree is going to be a random integer between 2 and 6. Data will be scaled between 0 and 1, by "x_s = (x - min(x)) / (max(x) - min(x))".

Similarly, further transformations will involve the functions like:
1) $s(x - r_1)(x - r_2)...(x - r_d)$
2) $s(x_s - 1 \times r_1)(x_s - 1 \times r_2)...(x_s - 1 \times r_d)$

Where, "s" is the randomly choosen sign, either -1 or +1, and r represents the random numbers between 0 and 1.

These transformations, along with scaling will be repeated as a part of the Monte Carlo Experiment. Subsequently, data will be classified using classifiers and classifiers ensembles e.g. binary decision tree, random forest bagging etc.

Consequently, error rates of the data with and without the tranformations will be compared.

Result and Conclusion:

Experiment is still in progress.

It is expected that some of the transformations are going to result in better accuracy and lower error rate.