Gaussian Process Classification

Rohit Jain

Mentor - Dr. Swanand Ravindra Khare

Report for B.Tech Project

Department of Mechanical Engineering
Indian Institute of Technology
India
9th April, 2018

Department of Mechanical Engineering

Indian Institute of Technology Kharagpur

Certificate

This is to certify that this is a bonafide record of the project presented by Rohit Jain(Roll no. 14MF3IM13) during Spring 2018 in partial fulfillment of the requirements of the degree of Bachelor of Technology in Manufacturing Science And Engineering as a part of his Bachelor's Thesis.

Prof S.R.Khare
(Project Guide)

Prof. Anandroop Bhattacharya and Prof. Atul Jain
(Course Coordinator)

Date: 9th April 2018

1 Abstract

In our daily life we come across many situations where we need to predict something so as to prevent a cause or optimize the outcome. This is where predictive modelling and data analysis comes into picture. In the starting sections of the study, basics of Gaussian process classification and related rules have been discussed. Then a firm base is made on the modelling and procedural part of the algorithm. Also, an important application of this algorithm is shown in the form of novel gas identification. The modelling part is based on Gaussian process (GP) combined with principal components analysis which is also discussed briefly. The experimental setup for identification of different gases is an array of commercial Taguchi gas sensors (TGS) as well as microelectronic gas sensors. The proposed approach is shown to outperform both K nearest neighbour (KNN) and multilayer perceptron (MLP) classifiers.

2 Introduction

There are mainly 2 types of predictive models, regression and classification. Regression deals with prediction of continuous functions and classification deals with discrete function prediction. Gaussian process regression is often used to predict when non-linearity is present in the given data. But this could be further extended to a classification problem in a well defined manner. The main objective for classification type of analysis is to model posterior probability of the target variable for a new input. But using Gaussian process regression we predict values which are outside the range of 0 to 1. And that indeed creates main problem as probability value ranges from 0 to 1. The solution for this could be to use a non-linear activation function such as sigmoid to map the earlier predicted values to the range 0 to 1. We exploit this knowledge of activation functions in order to predict correct classes.

3 Literature Review

The books 'Introduction to mathematical statistics[HC95]' and 'An introduction to probability theory and its applications[Fel08]' provides lot of basic knowledge about the concepts of probability and statistics needed to understand each and every bit used in the related papers including the multivari-

able conditional probability[HC95] discussed here. The kernel based method and basics about GPR is also derived from 'Gaussian processes for machine learning'[RW06] and 'Advances in Gaussian processes' [Ras06].

4 Problem Description

The problem statement is the same as that of any topic in predictive modelling i.e. we have a dataset or set of information regarding some of the aspects of the outcome which we want to predict for future cases. The aspects or properties could be numerical or categorical(which has only discrete values). A mathematical model or function is to be approximated using these properties to give correct values of the future outcome and in this case correct classes. There are many algorithms for this function approximation and are used for different applications. As the degree and complexity of this approximation could be very high and thus very difficult to calculate we use these algorithms to make our task easier.

5 Mathematical Formulation

5.1 Classification Problem

The objective of pattern recognition is to set a decision rule that optimally partitions the data space into c regions, one for each class C_k. The boundaries between regions are the decision boundaries. A pattern classifier generates a class label for an unknown feature vector $x \epsilon R^d$ from a discrete set of previously learned classes. The most general classification approach is to use the posterior probability of class membership $p(C_k|x)$. To minimize the probability of misclassification, one should select the maximum a posterior rule and assign x to class C_k [DHS12], i.e.,

$$C_k = max_{1...c}[p(C_k|x)] = max_{1...c}[p(x|C_k)p(C_k)]$$

where, $p(x|C_k)$ is the class-conditional density and $p(C_k)$ is the prior probability. In the absence of prior knowledge, $p(C_k)$ can be approximated by the relative frequency of different class examples in the dataset. One way to

build a classifier is to develop a model that estimates the posterior probabilities directly, where the boundaries are learnt from data. Another alternative could be to estimate the class-conditional densities by using representation models for how each pattern class populates the feature space.

5.2 GP CLassifiers

A Gaussian process is a collection of random variables, any finite set of which have a joint Gaussian distribution. For a finite collection of inputs, $x = [x^{(1)}, ..., x^{(n)}]^T$, we consider a set of random variables $y = [y^{(1)}, ..., y^{(n)}]^T$ to represent the corresponding function values. A GP is used to define the joint distribution[Bis95]

$$p(y|x) \sim exp(\frac{-y^T \Sigma^{-1} y}{2}) \tag{1}$$

where the matrix Σ is given by the covariance function $\Sigma_{pq} = cov(y^{(p)}, y^{(q)}) = C(x^{(p)}, x^{(q)})$.

5.2.1 Covariance Function

There are many possible choices of prior covariance functions. From a modeling point of view, we wish to specify prior covariance that contains our prior beliefs about the structure of the function we are modeling. Formally, we are required to specify a function that will generate a non-negative definite covariance matrix for any set of input points. GP can handle interesting models by simply using a covariance function with an exponential term, i.e.,

$$C(x^{(p)}, x^{(q)}) = v_0 * exp[\frac{-1}{2} \Sigma \rho_l (x_l^p - x_l^q)^2] + v_1 \tag{2}$$

where, $\theta = (\rho_1, ...\rho_d, v_0, v_1)$ are the set of hyperparameters which would be optimized later. It expresses the idea that cases with nearby(similar) inputs will have highly correlated outputs.

5.2.2 Classifying With GP

For the two-class problem, we use the logistic function to produce an output that can be interpreted as $\pi(x) = \sigma(y(x))$, the probability of the input x belonging to class 1. The probability and the activation corresponding to

input x_i will be noted by π_i and y_i, respectively. We specify a GP prior over the activation function y that we have a probabilistic model as a discriminant classifier. Making classification for a test input x^* given a set of training data $D = \{x^{(i)}, t^{(i)} | i = 1, ..., n\}$, with a fixed set of hyperparameters θ, requires the calculation of the integral defined[PS02][Nas07] as

$$\pi^{**} = \int \pi^* p(\pi^* | t, \theta) d\pi^* \tag{3}$$

where $\pi^* = \pi(x^*)$. To find the posterior distribution of π^*, we use the distribution $p(y^*|t, \theta)$, which is defined by

$$\begin{aligned} p(y^*|t, \theta) &= \int p(y^*, y|t, \theta)\, dy \\ &= \frac{1}{p(t|\theta)} \int p(y^*, y|\theta) p(t|y)\, dy \end{aligned} \tag{4}$$

and apply the appropriate Jacobian to transform the distribution. Because we are using a logistic sigmoid output function, the appropriate noise model is no longer Gaussian but Bernoulli.

$$p(t|y) = \prod_{i=1}^{n} \pi_i^{t_i} (1 - \pi_i)^{1-t_i} \tag{5}$$

which means that (4) is no longer analytically tractable. Faced with this problem, there are two ways to evaluate the integral, namely 1) to use an analytic approximation or 2) to use Monte Carlo methods. We consider an analytic approximation based on Laplace's method. We approximate the function $p(y^*, y|t, \theta)$. Finding a maximum can be carried out using the Newton–Raphson algorithm, which then allows the approximation distribution of y to be calculated.

5.2.3 Training a GP

Let us assume that a form of covariance function has been chosen, but depends on undetermined hyperparameters θ. We would like to learn these hyperparameters from the training data. In a maximum likelihood framework,

we adjust the hyperparameters to maximize the log likelihood of the hyperparameters i.e. $logp(t|\theta)$. For the classification problem, we have to approximate this function by using Laplace's method. Let $\psi = logp(t|y) + logp(y)$. Using $p(t_i|y_i) = t_i y_i - log(1 + exp(y_i))$, we obtain

$$\psi = t^T y - \Sigma_{i=1}^{n} log(1 + exp(y_i)) - \frac{1}{2} y^T Q^{-1} y - \frac{1}{2} log(det(Q)) - \frac{n}{2} log(2\pi) \quad (6)$$

where $Q_{pq} = C(x^{(p)}, x^{(q)})$. By using Laplace's approximation about the maximum y, we find that

$$log(p(t|\theta)) = \psi(y) - \frac{1}{2} log(det(Q^{-1} + W)) - \frac{n}{2} log 2\pi \quad (7)$$

We initialize the hyperparameters to random values (in a reasonable range) and then use an iterative method, for example, conjugate gradient, to search for optimal values of the hyperparameters. We found that this approach is sometimes susceptible to local minima. To overcome this drawback, we randomly selected a number of starting positions within the hyperparameters space.

The extension of the method to multiple classes is essentially straightforward, although considerably more complex. A softmax activation function for each class $C_k(k = 1, ..., c)$ with a one-of-c coding scheme is also used. The probability of the input x belonging to class C_k is given by[Nas07]

$$\pi_k(x) = \frac{exp(y_k)}{\Sigma exp(y_k)} \quad (8)$$

which ensures that the GP outputs lay in the range [0, 1] and summed to 1. As for a two-class case, one can take the new log likelihood for similarly represented targets $t_k^i(k = 1, ..., c)$ and assume that GP prior operates in the activation space.

6 Experimental Setup

The 5 datasets for the experiment is 'Gas Sensor Array Drift Dataset' from UCI Machine Learning repository. The experimental equipment for it consists of gas pumps, mass flow controllers, a sensor chamber, and a computer used for data acquisition and the experiment control. In a gas chamber, a

sensor array is placed based either on the commercial TGS or MHP microelectronic gas sensors. Vapors were injected into the gas chamber at a flow rate determined by the mass flow controllers. Measurement procedure consists of two steps. The first step consists of injecting the tested gas during the 10 min period, whereas 40 min is allocated to a cleaning stage with dry air. Data is collected from 5 different experiments with different gases.

- First experiment: A sensor array composed of five commercial tin oxide gas sensors, have been used to discriminate between ethanol and butanol. Different concentrations were used for both vapors.
- Second experiment: For the detection of CO in the presence of CH_4, we have used an array of eight microelectronic gas sensors. Different concentrations were used for each gas.
- Third experiment: It has the same experimental procedure as the second one but detects CO in the presence of hydrogen.
- Fourth experiment: The aim of fourth experimaent is to separate the signatures of carbon monoxide and the mixture of carbon monoxide and methane. The setup is similar to the one used in the previous two experiments.
- Fifth experiment: Methane, carbon monoxide, or their mixture vapors were injected into the gas chamber at a flow rate determined and accurately controlled by the mass flow controllers.

7 Solution Strategy

The most important multivariate tool for exploratory analysis is the PCA(principal component analysis). PCA was therefore used as a preprocessing stage for redundancy removing and feature reduction before applying a given classifier. By applying this technique, principal components and eigenvalues are calculated. The two first components allow us to take into account more than 95 percentage of the data variance, we limit our representation to these components. Figs. 1–5 present the first two PCA scores for each of the datasets so that we can judge the relative complexity of the different datasets. We note that the features can be separated into different distributions corresponding to each measurement gas type. However, the decision boundaries

are not well defined due to some overlapping, especially the third and the fifth datasets, having the most complex PCA scatter plots (Figs. 1–5). The inputs to each classifier are the projections of the data on the first two principal components. The hyperparameters of the GP classifier were adapted to the training data using conjugate gradient search algorithm. Table I reports the identification accuracy of the GP classifier in comparison to the one of KNN and the MLP classifiers. For the five datasets, the results using the GPs are better than both the KNN and MLP classifiers. These results can be explained by the fact that GP classifier succeeds in building a statistical model of the process that generates the data and, hence, exhibits better generalization performance than KNN and MLP. The GP classifier is also applied to the general case of more than two classes as shown in the fifth experiment.

8 Results and Discussion

In this report, a gas identification method was proposed based on GP models combined with PCA. We have shown that an excellent classification rate can be achieved on both TGS and microelectronic gas sensor data. Using GP classifiers, we obtained results that are superior to both KNN and MLP for gas identification applications. In addition, it was shown that the GP classifier can be applied to the general case of identifying more than two gases. However, a problem found in GP-based methods is the computational cost because they require computations (trace, determinants, and linear solutions) involving n n matrices, where n is the number of training examples. This represents a drawback when dealing with large datasets. It is of interest to investigate the possibility of speeding up Bayesian computations by using specific numerical methods.

Table 1: Gas Identification Results

Classifier	Datset1	Datset2	Datset3	Datset4	Datset5
GP	90.5	98	75	100	90
KNN	85	93.3	57.7	95.4	89.9
MLP	88.3	95.8	67.3	99.2	88.3

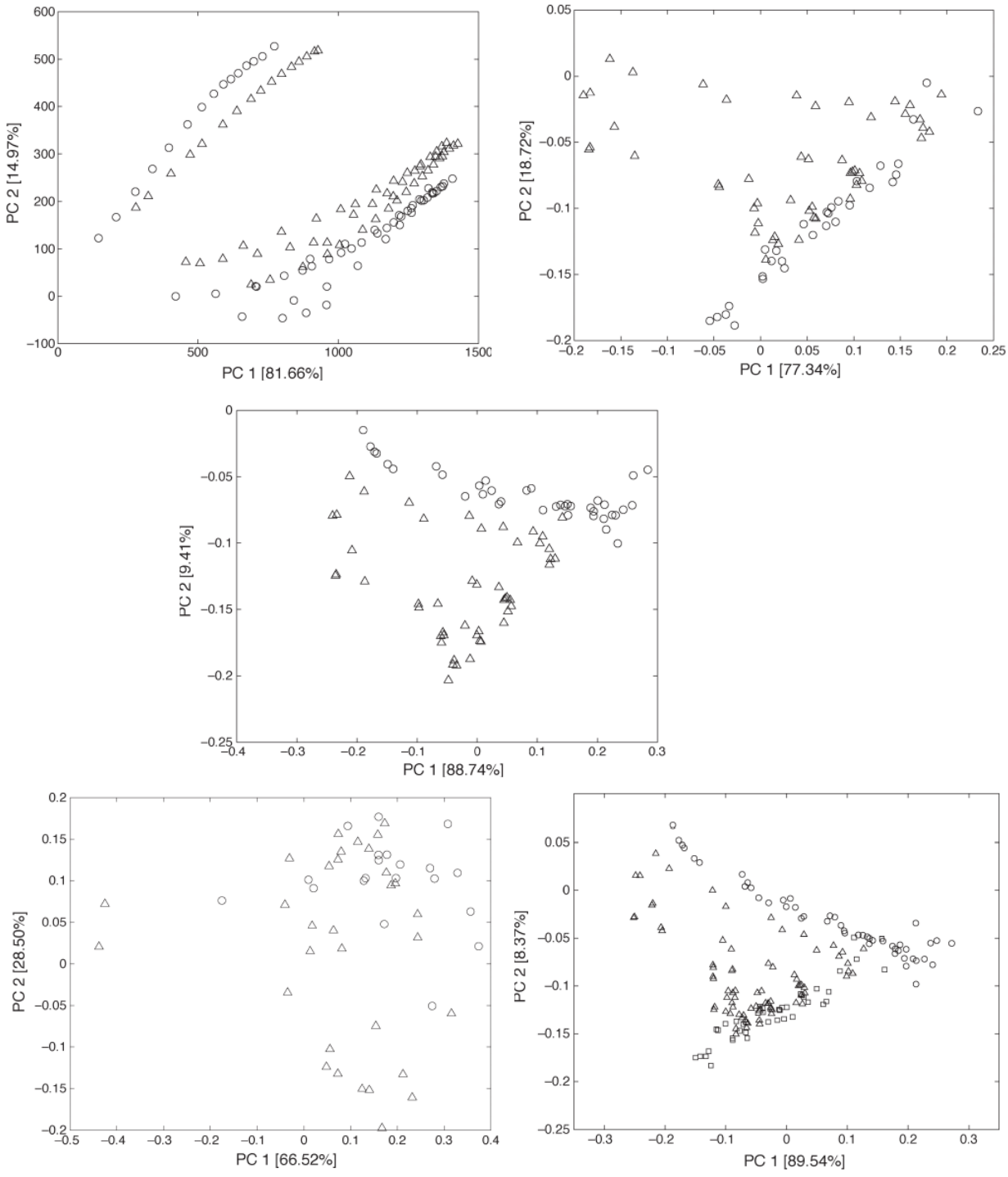

Figure 1: PCA Results

9 Conclusion

The algorithm discussed in this paper is universal in nature and thus could be used in any applications. The results also prove the effectiveness of this algorithm. But there could be more improvement in the results through more data pre-processing as seen in other related papers. Apart from that, the outcomes are more than satisfactory to be used in real life applications. However many new techniques could be investigated for the possibility of speeding up Bayesian computations by using specific numerical methods. There are also better algorithms for classification purposes which could be mixed with this approach to achieve better results.

References

[Bis95] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[HC95] Robert V Hogg and Allen T Craig. *Introduction to mathematical statistics.(5"" edition)*. Upper Saddle River, New Jersey: Prentice Hall, 1995.

[PS02] Matteo Pardo and Giorgio Sberveglieri. "Learning from data: A tutorial with emphasis on modern pattern recognition methods". In: *IEEE Sensors Journal* 2.3 (2002), pp. 203–217.

[Ras06] Carl Edward Rasmussen. "Advances in Gaussian processes". In: *Advances in Neural Information Processing Systems* 19 (2006).

[RW06] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*. Vol. 1. MIT press Cambridge, 2006.

[Nas07] Nasser M Nasrabadi. "Pattern recognition and machine learning". In: *Journal of electronic imaging* 16.4 (2007), p. 049901.

[Fel08] Willliam Feller. *An introduction to probability theory and its applications*. Vol. 2. John Wiley & Sons, 2008.

[DHS12] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.