Bayesian-Entropy Method for Probabilistic Diagnostics and Prognostics of Engineering Systems

by

Yuhao Wang

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy

Approved September 2020 by the
Graduate Supervisory Committee:

Yongming Liu, Chair
Aditi Chattopadhyay
Marc Mignolet
Yi Ren
Hao Yan

ARIZONA STATE UNIVERSITY

December 2020

ABSTRACT

Information exists in various forms and a better utilization of the available information can benefit the system awareness and response predictions. The focus of this dissertation is on the fusion of different types of information using Bayesian-Entropy method. The Maximum Entropy method in information theory introduces a unique way of handling information in the form of constraints. The Bayesian-Entropy (BE) principle is proposed to integrate the Bayes' theorem and Maximum Entropy method to encode extra information. The posterior distribution in Bayesian-Entropy method has a Bayesian part to handle point observation data, and an Entropy part that encodes constraints, such as statistical moment information, range information and general function between variables. The proposed method is then extended to its network format as Bayesian Entropy Network (BEN), which serves as a generalized information fusion tool for diagnostics, prognostics, and surrogate modeling.

The proposed BEN is demonstrated and validated with extensive engineering applications. The BEN method is first demonstrated for diagnostics of gas pipelines and metal/composite plates for damage diagnostics. Both empirical knowledge and physics model are integrated with direct observations to improve the accuracy for diagnostics and to reduce the training samples. Next, the BEN is demonstrated in prognostics and safety assessment in air traffic management system. Various information types, such as human concepts, variable correlation functions, physical constraints, and tendency data, are fused in BEN to enhance the safety assessment and risk prediction in the National Airspace System (NAS). Following this, the BE principle is applied in surrogate modeling. Multiple

algorithms are proposed based on different type of information encoding, such as Bayesian-Entropy Linear Regression (BELR), Bayesian-Entropy Semiparametric Gaussian Process (BESGP), and Bayesian-Entropy Gaussian Process (BEGP) are demonstrated with numerical toy problems and practical engineering analysis. The results show that the major benefits are the superior prediction/extrapolation performance and significant reduction of training samples by using additional physics/knowledge as constraints. The proposed BEN offers a systematic and rigorous way to incorporate various information sources. Several major conclusions are drawn based on the proposed study.

ACKNOWLEDGMENTS

Firstly, I would like to dedicate my sincere appreciation to Professor Yongming Liu, without whom the work would not have been possible. I would like to express my gratefulness to my committee members, Professor Aditi Chattopadhyay, Professor Marc Mignolet, Professor Yi Ren and Professor Hao Yan, for their continuous help and valuable discussions. I would also thank my dearest friends Dr. Chufeng Li, Dr. Minjing Yu, Dr. Eduardo Espiritu, Jingran Guo, etc. for their encouragement and advise, and the members in the Claremont Chinese Board for the support and entertainment they provided during my PhD career, as well as my colleagues in my research group for their discussions and feedbacks on my research study. Most importantly, I would like to raise my gratitude for my parents for the love, strength and understanding they have provided unconditionally.

TABLE OF CONTENTS

Page

CHAPTER Page

APPENDIX Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

1 Introduction

1.1 Backgrounds

Probabilistic events exist in almost every aspect in engineering systems. The intrinsic stochasticity often leads to potential extreme events, such as failures, surprises, and unpredictable behaviors. In many engineering systems, uncertainties could be introduced by model selection, model parameter, parameter dependencies, measurement, and computation. Significant amount of uncertainties makes it difficult for the evaluation and prediction for system health, which is important for the decision making and safety assurance of engineering systems. Therefore, systematic uncertainty quantification and probabilistic prediction are needed to enable accurate risk assessment and failure prevention.

Researchers have been working for decades on probabilistic methods trying to analyze the randomness in engineering problems. Bayesian statistics is one of the most important approaches in such circumstance. Bayesian statistics offers a different method to interpret probabilities. It is based on the Bayes' theorem [1]:

$$P(\theta \mid D) = \frac{P(D \mid \theta)P(\theta)}{P(D)} \tag{0.1}$$

Bayes' rule describes a conversion of a prior probability $P(\theta)$ to a posterior probability provided the evidence $P(\theta|D)$. The conditional probability $P(D|\theta)$ is referred as likelihood function, which describes the likelihood of the observed data D given a certain parameter θ. The denominator $P(D)$ can be regarded as a normalizing constant for the

posterior distribution. In another word, Bayes' theorem states that the posterior probability is proportional to the product of the prior and likelihood function.

Bayes' theorem has been extended to its network format for many engineering applications and is known as Bayesian Network (BN). The classical Bayesian Network (BN) is a probabilistic graphical model that describes the relations and dependencies between variables. The BN is formulated as a probabilistic direct acyclic graph (DAG). It represents a set of random variables by nodes and their conditional dependencies by edges. The DAG is a directed graph with vertices and directed edges but no directed cycles [2]. That is, there is no route to start at a vertex and follow a sequence of directed edges that eventually goes back to the starting vertex. BN is a robust tool in inference because of the ability to model causal relationships and update uncertainty based on observations. It has been applied in engineering problems to update and infer the model parameters through observations [3] and to predict system reliability [4]. The update is achieved by calculating the posterior probability based on experimental observations. The likelihood function is calculated through a known relation between the variable or a physical model. There has been a lot of tools for finding a BN given data [5][6]. Most of them construct the network by finding the correlations between variables. But correlation does not always mean causality of the variables.

BN also has its application in statistical machine learning [1], such as pattern recognition [7][8], image classification [9], automatic image segmentation [10] and Bayesian classifiers. There are different kinds of algorithm for Bayesian network classifier, such as Naïve Bayes, Tree Augmented Naïve Bayes and Selective Naïve Bayesian Network

[11], etc. Recently, the Bayesian method is combined with Neural Network (NN) to form a Bayesian deep learning. Instead of a scalar value, it handles the weights in a Neural Network as a distribution function. Such methods can handle data instances and utilize point observations. But in some scenarios, other types of information or data are available. These data may include historical data, physical constraints and human experience. The historical data may be abstracted statistic data of a population such as mean and variance. Physical constraints are certain for the range or specific value for a parameter. Human experience can be empirical information from an experienced engineer. Such information is not easy to be encoded into the traditional Bayesian method.

Some researchers tried to introduce additional information into the Bayesian network methods. One research focuses on updating with fuzzy range data [12]. It uses an integral over the range data as the likelihood function to represent the possibility of the range information. [13] used a Bayesian prior to control the posteriors by adding constraints. The method is to minimize the Kullback-Leibler divergence between the target distribution and the prior under constraint. The posterior is solved using expectation maximization (EM) algorithm. A posterior regularization framework was presented in [14] for structured, weakly supervised learning. The framework treats data-dependent constraint as information about model posteriors. The work was further developed as a regularized Bayesian inference in [15]. The work in [16] provided a method to update parameters using statistical moment information by using a network that directly inference for the parameter given statistical information. Although the method can successfully take advantage of the

moment information that can be interpreted from human information, the model requires a separate dataset to train the likelihood function.

The Maximum Entropy (ME) method [17][18] provides an alternative way of handling different types of information. The ME regards information in the form of constraints. The constraints can include point observations, moment information, and range data. It is also proved that the Bayesian theorem is just a special case of the ME method [19]. In the ME method, the target posterior distribution has an additional exponential term comparing to the classical Bayes' theorem. The equation can be expressed as:

$$P(\theta \mid D) \propto P(D \mid \theta) P(\theta) e^{\beta f(\theta)} \tag{0.2}$$

where D represents the observation data and θ is the parameter of concern. $P(\theta)$ is the prior probability of θ. $P(D|\theta)$ is the likelihood function and $P(\theta|D)$ is the posterior probability. Comparing to the Bayes' theorem, it has an additional exponential term. β is a constant and $f(\theta)$ is the constraint function. The exponential term can be analytically solved given specific constraint. The details will be discussed in the next chapter. The constraint can be used to encode additional information, such as the statistical data, physical constraints and human experience, into the Bayesian framework. But there has been many debates about the inconsistency of the ME method [16][20][21]. The debate is around the commute of constraints in ME method. Constraints are referred as commuting when the sequence of handling the constraints does not affect the result. As it is commonly known, the updating in Bayesian method gives the same result regardless of the updating sequence or updated simultaneously. Some argues that the result from the ME method would depend on the sequence of the applied constraints. [22] offered an explanation saying that multiple data

points are observations that cannot be undone, so the constraint corresponding to one observation should be considered in all following updates. While this is true, the accumulation of a large quantity of observations would yield the same number of constraints.

1.2 Research objectives

In this study, we propose a Bayesian Entropy method to introduce extra information into the Bayesian method in the form of constraints. The idea is to use a Bayesian part which deals with point observations as a classical Bayesian framework, and an entropy part that can encode extra information as a constraint. This way the commute issue of the ME method can be neglected since the point observations are handled using the Bayesian part. With the extra exponential term, any information that can be written in the form of a constraint can be encoded into the method. The detailed derivation for Bayesian Entropy method to handle various information, such as moment constraint and range constraint, is given. A new perspective of interpreting data and constraint will be given with the method. The key idea is that the entropy is only considered when the observation data is less, or when the belief of the extra information is high. The extra information may be beneficial when the number of observables is limited. Once the data is available in large quantities, we would tend to believe in the data. An adaptive algorithm is proposed to mitigate the strong effect of the constraint. The Bayesian Entropy method is applied in classification and inference for diagnostic and prognostic tasks in engineering problems. There exists a lot of information other than point data in these fields. The proposed method can take

advantage of them and improve the classification accuracy and enhance the prediction result.

The benefits and uniqueness of the Bayesian Entropy network are:

1. The method combines the classical probabilistic theory (Bayesian method) and information theory (ME method) to successfully encode extra information into the classical Bayesian approach.
2. In classification, the method can take advantage of the extra information to achieve a higher accuracy when the quantities of training data are limited. By the exponential term, a lot of other evidences can be utilized when inferring for the probability distribution of a parameter.
3. The extra exponential term does not add any computational complexity to the classical Bayesian framework. In general, the Bayesian Entropy approach can be applied in any method that is based on Bayesian theorem, such as Bayesian classifier and Bayesian network.

Figure 1-1. shows a flowchart for the proposed work. The Bayesian Entropy method is developed based on the hybrid of probabilistic theory (Bayesian method) and information theory (Maximum entropy method). With the compatibility with the classical Bayes' theorem and the ability to encode extra information, the proposed method will be used in diagnosis and prognosis in engineering tasks.

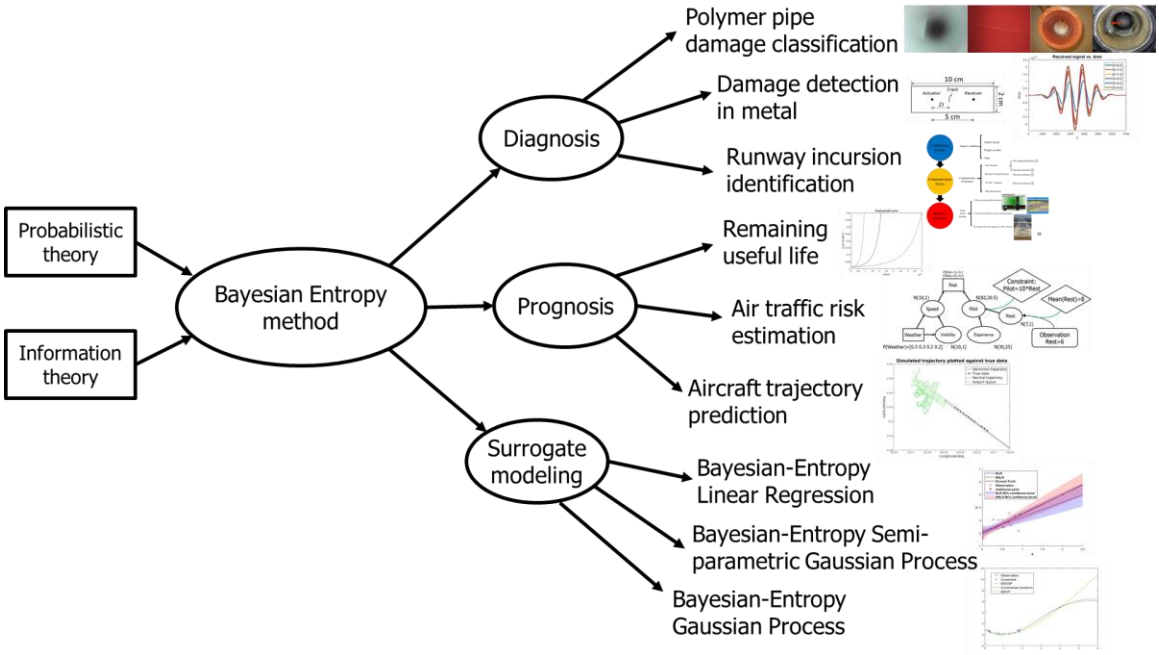


Figure 1-1. Flowchart for the proposed work.

The rest of the report will be divided into 5 chapters. In Chapter 2, the history of the information theory is briefly introduced, the definition of entropy will be stated. The information is regarded as constraint in this framework. The constraint is written in the form of an integral. The posterior is solved by maximizing the entropy under the available information constraints. The maximization problem is solved using Lagrange method. The resulting posterior has an additional exponential term compared to the Bayes' theorem. The derivation for moment information as constraint, range information as constraint and general function as constraint is given. The sequential effect of the ME method is due to the sequence of handling the constraints. The Bayesian Entropy method offers an alternative way of interpreting the equation. The point data is handled using the Bayesian part of the equation in the classical Bayesian manner, and the exponential term (the entropy part) encodes the extra constraint. The constraint information in the Bayesian Entropy

method is indeed a strong one, an adaptive method is introduced to mitigate the strong effect of the constraint. It is simply a weighing between the Bayesian posterior and Bayesian Entropy posterior. When the observation data becomes more, it will overwhelm the strong effect of the entropy constraint.

Chapter 3 introduces the application of the Bayesian Entropy method in diagnostic tasks. Firstly, the method is applied in classification for damage types for polymer pipes through 3D image reconstruction. The geometric features are selected as variables in the classifier. A Naïve network structure is used for the Bayesian classifier. Some common knowledge, such as a slit damage is long, is encoded into the Naïve Bayesian classifier as entropy constraint. A visible improvement can be observed by comparing the accuracy with regular Bayesian classifier. The second example involves the damage detection in metal using non-destructive method. The proposed method will be used in inference for the location of the damage. A crack damage is located in between two piezoelectric sensors. A back-propagate physical model can accurately calculate the damage location based on the signal generated and received by the sensors. The physical model would require a long computational time. A Bayesian network is used to estimate the damage location. The network includes some abstracted features from the signals. A set of simulated data set is used to train the probability distributions in the network. A regression function from the simulated data is assumed to be known as a physical relationship and encoded into the network as constraint. The network with entropy constraint is referred as Bayesian Entropy Network (BEN). The proposed method can achieve a relatively accurate estimate for the damage location.

Chapter 4 explores the application of the proposed method in prognostic tasks in aviation applications. The first example involves the identification for the cause of a common accident in air traffic system called runway incursion. From the Aviation Safety Reporting System (ASRS) database, some key features are found to be the influencing factor for such accident. A BEN is constructed with these variables and some empirical information encoded as constraints. Although the BEN can improve the classification result, but the accuracy is still low. This is due to the lack of useful data from the report. The second example is related to a causal Bayesian network for air traffic control. The network model predicts the occurrence probability of air traffic accident related to human performance. Additional information from various sources will be encoded into a Bayesian network model to demonstrate the influence of extra constraint. The result is compared with the classical Bayesian method. The third example is the prediction of landing location of an aircraft. The example involves a real accident where the aircraft mistakenly lined up with the taxiway and almost run into the four other aircrafts on the taxiway. Using Bayesian updating, the landing point can be predicted with the continuous observation of track points. The entropy is introduced to constraint the landing point, since the aircraft would only land on the runway or the taxiway. The posterior of the landing point would shift to random positions. And with entropy constraint, the posterior would look more reasonable. The posterior prediction would be more accurate. By using the Bayesian updating with entropy constraint, the accident can be predicted well before it would happen. This framework may be used as a computer aided air traffic control program for the NextGen.

Chapter 5 applies the Bayesian-Entropy principle into surrogate modeling to introduce extra constraints. Based on the classical Bayesian regression, a Bayesian-Entropy Linear Regression (BELR) is formed to consider extra information as constraints. The extra information could be value constraints, which may come from empirical knowledge, or derivative constraints, which may come from existing physical models, on the target regression function. The constraints can be written in an integral form with respect to the regression coefficient. The Bayesian-Entropy posterior accounting these constraints can be analytically solved. The regression result from BELR can strictly follow the constraints. The BELR methodology is then applied into the mean function of a Semi-parametric Gaussian Process in form a Bayesian-Entropy Semiparametric Gaussian Process (BESGP) to encode physical constraints into the surrogate model. The BESGP is applied as a trajectory surrogate to consider the waypoint constraints, flying direction as physical constraints. The BESGP can utilize the extra information and have a good extrapolation behavior compare to pure SGP. The Bayesian-Entropy method has also been found useful in directly impose constraints on the Gaussian Process. This method is proposed as Bayesian-Entropy Gaussian Process (BEGP) to enable a more general regression tool for adding constraints into the target regression function. The BEGP regards the hyperparameters of the GP as a random variable and finds its Bayesian-Entropy posterior given the constraints. However, a numerical solution is not ease. Hence, two different approaches are proposed for the BEGP solution. A sampling method is used to solve for the hyperparameter distributions, but numerical approximation can introduce errors for the prediction to follow the specified constraints. A double-loop optimization can efficiently solve the Lagrange multipliers and the hyperparameters, with an assumption that the

expected value of the constraint is equivalent to the solution from Maximum a posteriori (MAP). The constrained GP method can enable a good extrapolation taking advantage of the extra information. It can also enable the smooth connection of multiple local GPs.

Chapter 6 concludes the research findings. Future works and potential improvements are discussed. The presented research proposed multiple algorithms for the fusion of different types of information based on the Bayesian-Entropy method. The future research direction could be seeking the analytical solution or efficient sampling method for the Bayesian-Entropy posterior in the BEGP method. This can greatly benefit the efficiency and accuracy of the constrained surrogate modeling. The application of the Bayesian-Entropy method in deep learning such as Bayesian-Entropy Neural Network may also be a good research topic based on the Bayesian-Entropy principle.

2 Formulation of Bayesian-Entropy (BE) method

A hybrid method for information fusion combining the maximum relative entropy (ME) method with the classical Bayesian network is proposed. The key benefit of the proposed method is the capability to handle various types of information for classification and inference. Classical point observations, which is handled well using Bayesian method, and information in the form of moments, ranges, and general constraints can be easily fused using the proposed Bayesian Entropy method. The flexibility of the proposed method to handle different types of information is particularly useful for some engineering applications where only abstracted information on the mean and variance of raw data is available due to data reduction, and where experts' knowledge is important in determining the range of parameters. This type information/knowledges can be encoded into the Bayesian Entropy Network (BEN) in the form of constraint and is augmented with the classical likelihood function in the Bayesian method.

2.1 Introduction

Statistical machine learning has been extensively developed during the past decades and Bayesian Network (BN) was one of the most widely-used learning method [1]. BN is a robust tool in inference due to its ability to model causal relationship and update the posterior distribution with observations (most likely point observations). BN has been applied in engineering problems to update and infer the model parameters through observations [3] and to predict system reliability [4]. The application of Bayesian updating in engineering problems dates back to the 1970s. The potential of the method was discovered for updating the probability distribution for model parameters and the

prediction of system response [23][24][25]. The Bayesian method requires the evaluation of the posterior probability as a product of the prior and likelihood function. The prior is related to the previous belief in the parameter of concern and the likelihood related to the model and the observed data.

In practice, other types of information may be accessible especially for the complex engineering systems. For example, in the analysis of the failure in aircraft, the data required to train an accurate network is not available. While an expert engineer from industry may suggest that the aircraft would be in critical condition if no maintenance for 1000 hours of operation. Such empirical information from expert opinion or human concept is not easy to be encoded into the traditional Bayesian method. Other examples of available information include moment information (mean and variance etc.) and range information. There are existing researches about introducing additional information into the Bayesian network-based method. [12] presented a way of updating using fuzzy data by changing the likelihood function into an integral over the range data. [13] used a Bayesian prior to control the posteriors by adding constraints. The method is to minimize the Kullback-Leibler divergence between the target distribution and the prior under constraint. The posterior is solved using expectation maximization (EM) algorithm. A posterior regularization framework was presented in [14] for structured, weakly supervised learning. The framework treats data-dependent constraint as information about model posteriors. The work was further developed as a regularized Bayesian inference in [15]. [26] introduced a Bayesian maximum entropy (BME) that can integrate information from different sources. It was proven to have better accuracy in spatiotemporal problems [27][28]

and has successfully applied in various fields [29]. BME uses maximum entropy theory to construct the prior distribution based on general knowledge or physical law [30]. The method was found especially popular in geostatic problems to handle cite specific information [31][32][33]. In this paper, we propose using the maximum entropy (ME) method to encode extra information in the form of constraint in the Bayesian framework. The ME method is an alternative tool for updating for posterior distribution regarding given evidences as constraints. It was first introduced in Jaynes' information theory [17][18] and has been widely applied in science and engineering fields. It is shown that the Bayes' theorem is a special case of the ME algorithm [19] where only point data is available. Although there has been debates [20][21] about the inconsistency of the entropy method to the classical probability theory, i.e., the Bayesian method, it has been shown that the inconstancy depends on how the information as constraint is used [22]. By maximizing the entropy, the targeted posterior distribution has an additional exponential term comparing with the classical Bayes' theorem. The exponential term can be analytically solved given specific constraints. The constraint can be used to encode additional information into the Bayesian framework. The ME method incorporating moment constraints has been derived in [22] and successfully applied in a fatigue life prediction scheme via single parameter update [34]. The work in [16] provided a method to update parameters using statistical moment information. It builds a separate network to train the likelihood function between the moment data and the model parameters. Although the method can successfully take advantage of the moment information that can be interpreted from historical data, the model requires a separate step and an additional dataset for training.

In this paper, we propose a method of handling extra information via the ME method. The extra information will be introduced into the Bayesian method in the form of constraints. The idea is that the method has a Bayesian part which deals with point observations as a classical Bayesian framework, and an entropy part that can encode extra information as a constraint. The new developed method will be called Bayesian Entropy Network (BEN) from this point on. The rest of the paper will be organized as follows: in the next section, the definition of entropy will be reviewed and the derivation for the target posterior given moment and range constraint will be introduced. Following this, the discussion of the sequential effect of interpreting different types of information is given.

2.2 Brief review of the maximum entropy method

The information theory dates back to the 1960s when Jaynes discussed the maximum entropy estimate in [17]. As presented in [17], the ME method was originally defined for the purpose of assigning probability using information based on partial knowledge as constraints. It was later found that the Bayes' rule can be derived from the ME method using observations as a constraint [19]. The method was used in updating probability with moment constraint in [22]. The entropy between two distribution function $P(\theta)$ and $Q(\theta)$ is given as:

$$S[P,Q] = -\int_{\Theta} P(\theta) \log \frac{P(\theta)}{Q(\theta)} d\theta \tag{1.1}$$

where θ is the distribution parameter. The integral range is over the domain of the parameter θ. The entropy is used to describe the difference between the two distribution functions. The idea of the ME method is to find the posterior distribution that maximizes

the entropy under constraints. In this section, the derivation of the constraint term will be given under various types of information.

2.2.1 Maximizing entropy with point observations

When updating for a parameter θ with observable variable x, the traditional Bayesian method updates the belief on the parameter θ with some observed values of x as:

$$p(\theta \mid x = x') = \frac{p(x = x' \mid \theta)\, p(\theta)}{p(x = x')} \tag{1.2}$$

where x' represents the observation. $p(\theta|x=x')$ is the posterior distribution of θ given the observation x'. The $p(x=x')$ in the denominator is acting as a normalizing constant and the equation can be written in the proportional form as:

$$p(\theta \mid x = x') \propto p(x = x' \mid \theta)\, p(\theta) \tag{1.3}$$

In the ME method, the goal is to maximize the entropy under the observation constraint. In this case, the entropy involves the old and new joint distribution of parameter θ and the observable variable x. The entropy is measured between the posterior and prior of the joint distribution for θ and x as

$$S = -\iint_{X\times\Theta} p(x,\theta) \log \frac{p(x,\theta)}{q(x,\theta)}\, dx d\theta \tag{1.4}$$

where $p(x, \theta)$ and $q(x, \theta)$ is the posterior and prior joint distribution. The constraint given the observation data x' is described using a delta function:

$$p(x) = \int_{\Theta} p(x,\theta) d\theta = \delta(x - x') \tag{1.5}$$

Another constraint comes from the definition of a probability distribution function (PDF), i.e., the integral over the domain equaling to unity. We have the normalization constraint:

$$\iint_{X\times\Theta} p(x,\theta)dxd\theta = 1 \tag{1.6}$$

To maximize the entropy, we use the Lagrange method. The Lagrange function is formed as

$$\mathcal{L} = S + \alpha[\iint_{X\times\Theta} p(x,\theta)dxd\theta - 1] + \int_X \lambda(x)[\int_\Theta p(x,\theta)d\theta - \delta(x-x')]dx \tag{1.7}$$

where α and $\lambda(x)$ are Lagrangian multipliers. The Lagrange function is a function of the posterior $p(x,\theta)$. To find the target posterior $p(x,\theta)$, the variation of the Lagrange needs to be zero, i.e., $\delta\mathcal{L} = 0$. Thus, the derivative of the Lagrange function to $p(x,\theta)$ equals zero,

$$\frac{\partial\mathcal{L}}{\partial p} = \iint_{X\times\Theta} [-\log\frac{p(x,\theta)}{q(x,\theta)} - 1 + \alpha + \lambda(x)]dxd\theta = 0 \tag{1.8}$$

We can solve for the target posterior as:

$$p(x,\theta) = q(x,\theta)e^{-1+\alpha+\lambda(x)} \tag{1.9}$$

The multiplier α brings in a normalizing constant and can be dumped into a constant Z as

$$p(x,\theta) = \frac{1}{Z}q(x,\theta)e^{\lambda(X)} \tag{1.10}$$

where $Z = e^{1-\alpha} = \iint_{X\times\Theta} q(x,\theta)e^{\lambda(X)}dxd\theta$. By substituting this result to the observation constraint in Eq. (1.5), we can solve for the multipliers $\lambda(x)$ as

$$\frac{e^{\lambda(x)}}{Z}\int_{\Theta} q(x,\theta)d\theta = \frac{e^{\lambda(x)}}{Z}q_X(x) = \delta(x-x') \tag{1.11}$$

where $q_X(x)$ is the prior marginal distribution on x. Therefore, the posterior of the joint distribution can be expressed as:

$$p(x,\theta) = \frac{q(x,\theta)}{q_X(x)}\delta(x-x') \tag{1.12}$$

Integrating over x for the marginal distribution of θ:

$$p_\theta(\theta) = \int_X p(x,\theta)dx = \int_X \frac{q(x,\theta)}{q_X(x)}\delta(x-x')dx = q(\theta \mid x = x') \tag{1.13}$$

This is exactly the Bayes' theorem where the posterior distribution is the conditional probability based on the observed point data x'.

2.2.2 Maximizing entropy with moment information

The benefit of the ME method is that, theoretically, any information can be encoded if written in the form of a constraint. Statistical moment information can often be derived from historical database or due to the data reduction/abstraction. While the traditional Bayesian method may not be able to easily handle this type of information, it can be written

in a constraint form and used in the ME method. The moment information of the parameter θ can be expressed as:

$$\iint_{X\times\Theta} p(x,\theta)g(\theta)dxd\theta = G \tag{1.14}$$

Eq. (1.14) represents the expected value of a function $g(\theta)$. When $g(\theta)=\theta$, the equation represents the first order moment. When $g(\theta)=\theta^2$, the equation represents the second order moment and so on. With the normalization constraint and observation constraint as in Eq. (1.5) and Eq. (1.6), we have the Lagrange function as

$$\mathcal{L} = S + \alpha[\iint_{X\times\Theta} p(x,\theta)dxd\theta - 1] + \beta[\iint_{X\times\Theta} p(x,\theta)g(\theta)dxd\theta - G] + \int_X \lambda(x)[\int_\Theta p(x,\theta)d\theta - \delta(x-x')]dx \tag{1.15}$$

where α, β and $\lambda(x)$ are the Lagrangian multipliers. To find the maximum of the Lagrangian function, the derivative with respect to $p(x,\theta)$ should equal to zero, we have:

$$\frac{\partial\mathcal{L}}{\partial p} = \int[-\log\frac{p(f_j,C)}{\mu(f_j,C)} - 1 + \alpha + \beta g(\theta) + \lambda(x)]dxd\theta = 0 \tag{1.16}$$

We can solve for the target posterior as:

$$p(x,\theta) = \frac{1}{Z}q(x,\theta)e^{\lambda(x)+\beta g(\theta)} \tag{1.17}$$

where $Z = e^{-\alpha+1}$ is the normalizing constant. Substitute the result back to Eq. (1.14) to solve for β

$$\frac{1}{Z}\iint_{X\times\Theta} q(x,\theta)e^{\lambda(x)+\beta g(\theta)}g(\theta)dxd\theta = G \tag{1.18}$$

This leads to an implicit equation:

$$\frac{\partial \log Z}{\partial \beta} = G \tag{1.19}$$

Given the specific form of the prior joint distribution, the β term could be analytically solved. Similar to the process in section 2.1, the new marginal distribution for θ can solved as

$$p_\theta(\theta) = q(\theta \mid x = x')\frac{e^{\beta g(\theta)}}{Z} \propto q(\theta \mid x = x')e^{\beta g(\theta)} \tag{1.20}$$

As shown in Eq. (1.20) the solution from the ME method includes the Bayesian part (the first term on the right-hand side) and an additional part that includes the moment information (the exponential term). If there is no such information, i.e., β=0, the equation recovers the classical Bayesian updating rule. Detailed derivation can be found in Appendix A.

2.2.3 Maximizing entropy with range constraint

Another type of information commonly seen in engineering is the range information. For example, observation of some physical properties is within a certain range (i.e., crack length in a bridge is between 5mm and 6mm). Another example is that a parameter θ could have a physical constraint and its value should fall in a certain range. While it would lose some generality in assuming a specific bonded prior for the parameter, we introduce a range

constraint using the entropy method. Assume the parameter θ should be in the range from a to b, the constraint could be expressed as:

$$\int_{a}^{b}\int_{\mathrm{X}} p(x,\theta)dxd\theta = 1 \tag{1.21}$$

Along with the normalization constraint in Eq. (1.6). The idea of these two constraints setup is that the target posterior only takes value in the range from a to b for variable θ, and for $\theta \notin (a,b)$ the probability density is zero:

$$\int_{\theta\notin(a,b)}\int_{\mathrm{X}} p(x,\theta)dxd\theta = 0 \tag{1.22}$$

Forming the Lagrange function using these two constraints:

$$\mathcal{L} = S + \alpha[\iint_{\Theta\times\mathrm{X}} p(x,\theta)dxd\theta - 1] + \gamma[\int_{a}^{b}\int_{\mathrm{X}} p(x,\theta)dxd\theta - 1] \tag{1.23}$$

where α and γ are Lagrangian multipliers. The variation of the Lagrangian equals zero yields:

$$\frac{\delta\mathcal{L}}{\delta p} = -\iint_{\Theta\times\mathrm{X}} (\log(\frac{p(x,\theta)}{q(x,\theta)}) + 1)dxd\theta + \alpha\iint_{\Theta\times\mathrm{X}} dxd\theta + \gamma\int_{a}^{b}\int_{\mathrm{X}} dxd\theta = 0 \tag{1.24}$$

The target posterior needs to satisfy Eq. (1.24), which means that $p(x,\theta)$ should satisfy Eq. (1.24) when $\theta \in (a,b)$ and $\theta \notin (a,b)$. This provided us with two equations when $\theta \in (a,b)$ and $\theta \notin (a,b)$:

$$\begin{cases} -\int_a^b \int_X (\log(\frac{p(x,\theta)}{q(x,\theta)})+1)dxd\theta + \alpha \int_a^b \int_X dxd\theta + \gamma \int_a^b \int_X dxd\theta = 0 \\ \int_{\theta \notin (a,b)} \int_X (\log(\frac{p(x,\theta)}{q(x,\theta)})+1)dxd\theta + \alpha \int_{\theta \notin (a,b)} \int_X dxd\theta = 0 \end{cases} \tag{1.25}$$

This leads to a piecewise solution for the posterior as:

$$p(x,\theta) = \begin{cases} q(x,\theta)e^{\alpha-1+\gamma} & ,\theta \in (a,b) \\ q(x,\theta)e^{\alpha-1} & ,\theta \notin (a,b) \end{cases} \tag{1.26}$$

Substitute the result into Eq. (1.21), we have:

$$e^{\alpha-1+\gamma} = \frac{1}{Q_\theta(b) - Q_\theta(a)} \tag{1.27}$$

where $Q_\theta(\cdot)$ is the cumulative density function (CDF) of the prior distribution for θ. And substitute Eq. (1.26) back in Eq. (1.22) we have:

$$e^{\alpha-1}[1-(Q_\theta(b) - Q_\theta(a))] = 0 \tag{1.28}$$

Since we do not control or make any assumption about the prior distribution, the result in Eq. (1.28)indicates that the term $e^{\alpha-1}$ should be an infinitely small number. This is not hard to express in numerical calculations, it can simply be achieved by assigning a large negative number (e.g. -1000) to α. Hence, by integrating the joint posterior over x, the final solution for the posterior of θ given a range constraint from a to b is:

$$p_\theta(\theta) = \begin{cases} \frac{1}{Q_\theta(b) - Q_\theta(a)} q_\theta(\theta) & ,\theta \in (a,b) \\ 0 & ,\theta \notin (a,b) \end{cases} \tag{1.29}$$

It can be seen that the solution from the entropy method gives us a truncated distribution on the specified range.

2.2.4 Maximizing entropy with general function as constraint

Regarding two jointly distributed normal variables, the correlation between these two variables can only describe the linearity relations of the variables. But in often times, the true relation between two parameters is non-linear. When modeling such variables as a bivariate normal distribution, the deviation caused by linearity assumption could be large. While the explicit function could be known between the two physical quantities, the relations between variables could be encoded as a constraint using the entropy term. Assume that a known relation exists between the observable quantity x and the parameter θ and could be expressed as $\theta = f(x)$. The constraint could be written as:

$$\int_{\Theta} p(\theta \mid x)\theta d\theta = f(x) \tag{1.30}$$

The integral is over the domain of parameter θ. To interpret Eq. (1.30), it can be stated in human language as: The expected value for parameter θ given observational value x is equal to $f(x)$. Similar to the previous sections, to maximize the entropy with Eq. (1.30) and the normalization constraint as in Eq. (1.5), a Lagrange function can be formed:

$$\mathcal{L} = S + \alpha[\int_{\Theta} p(\theta \mid x)d\theta - 1] + \beta[\int_{\Theta} p(\theta \mid x)\theta d\theta - f(x)] \tag{1.31}$$

Note that the object in this case is the conditional probability of θ given x. To maximize the Lagrange function, the derivative with respect to the conditional probability of θ given x is equal to zero, which gives:

$$\frac{\delta \mathcal{L}}{\delta p} = -\int_{\Theta} (\log(\frac{p(\theta \mid x)}{q(\theta \mid x)}) + 1) d\theta + \alpha \int_{\Theta} d\theta + \beta \int_{\Theta} \theta d\theta = 0 \tag{1.32}$$

Solving Eq. (1.32) we can have the relation between the target likelihood function and the prior likelihood:

$$p(\theta \mid x) = q(\theta \mid x) \exp(-1 + \alpha + \beta\theta) \tag{1.33}$$

Assume normality for the distribution function and substituting Eq. (1.33) back into the constraint in Eq. (1.30) we can have the final form of the new distribution function given constraint:

$$p(\theta \mid x) = q(\theta \mid x) \exp(\frac{f(x) - \mu}{\sigma^2} \theta) \exp(\frac{f^2(x) - \mu^2}{2\sigma^2}) \tag{1.34}$$

Eq. (1.34) is the final result of posterior given the general function as a constraint. In which μ and σ are the mean and standard deviation for the conditional distribution of $q(\theta|x)$, which is the posterior from Bayesian method. Detailed derivation is very similar to that in section 2.2.2 and will not be given here.

2.3 The behavior of the BE method

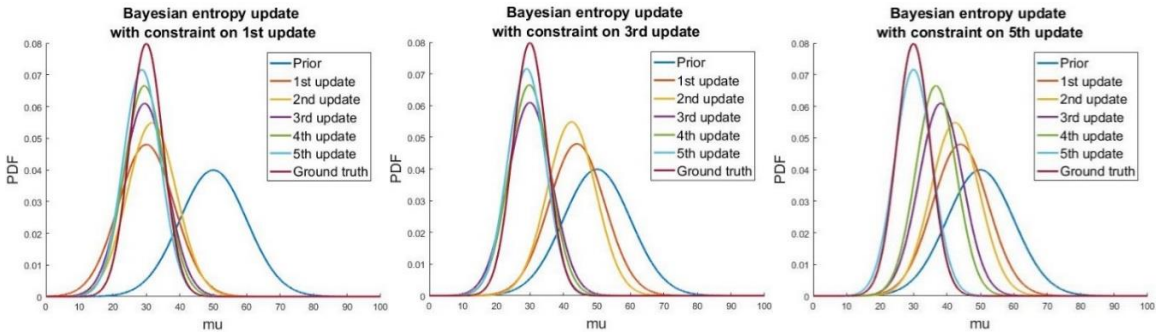
While it has been criticized in [16] about the inconsistency of the maximum entropy method with the Bayes' rule, this section will provide a new point of view to present the application of the entropy method in the Bayesian framework. The point observations will be handled in the classical Bayesian rule and the extra information will be encoded using the maximum entropy method. The demonstration will be given through a simple toy problem where the probability distribution of the mean for a Normal distribution with

known variance is updated. The variance of the parameter distribution is set to be 15^2. The prior distribution for the mean follows $N(50,10^2)$. Five samples are drawn from a Normal distribution $N(30,5^2)$ as observations from the population, namely (30.62 37.18 20.19 29.01 23.96).

2.3.1 Sequential or simultaneous updating with point data

It is known that the sequential updates for Bayesian method would always give the same result regardless of the order for the observation data. [20][21] has been pointed out that Jaynes' information theory (the entropy method) is inconsistent with the probability theory (Bayesian method).

Figure 2-1 has shown that putting the constraint information in different steps would yield different results for the posterior probability distribution. This kind of behavior is called the constraint is not commuting [22]. A constraint is regarded as commuting when the result does not depend on whether they are handled simultaneously or sequentially. It can be seen that the point data handled as constraint using delta function in Eq. (1.5) is commuting, which is the same as the Bayesian method.



a) b) c)

Figure 2-1. The sequential updates of data with constraint on a) the first update, b) the third update and c) the fifth update

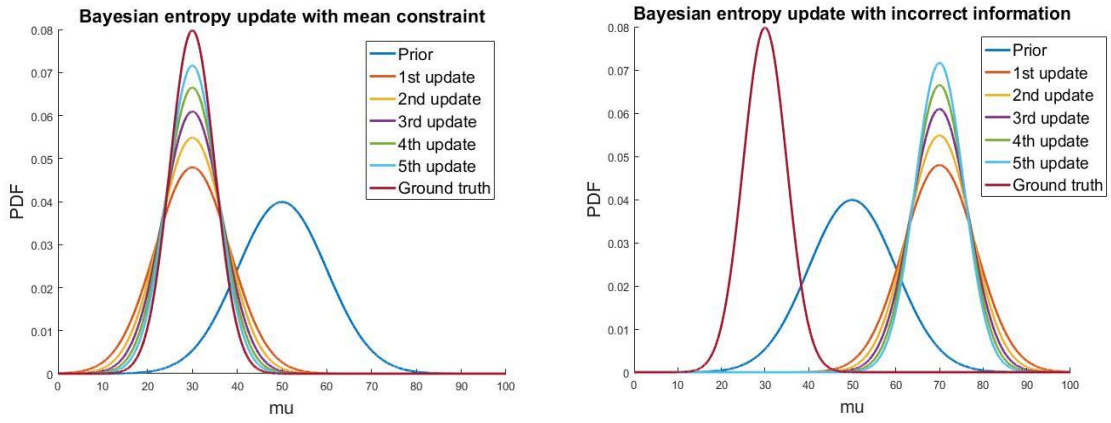
The idea of regarding information as constraint has been well explained in [22]. When sequentially update using data points, the entropy method would need to consider the points in previous updates to give the same result as the simultaneous updating. In our formulation of the BEN, the point observation is handled using the Bayesian part. The other types of information will be encoded using the entropy term. This way the BEN would keep the Bayesian behavior when dealing with point data and avoid the inconsistency debate, while still has the ability to incorporate additional constraint. How and when should constraint information be used will be explained in the following subsection.

2.3.2 Using data with constraint

In could be a great benefit to take advantage of the extra information apart from the observation. However, how a piece of incorrect information can affect the final updating result needs to be further studied. Below we give an illustration of the effect of a correct and incorrect constraint information on the posterior.

Following the updating case in the last section, since the samples are drawn from a distribution with mean equals 30, we assume this piece of information is known and encoded into each update as a mean constraint. In the first example, assume we were presented with correct information, i.e. the mean of the posterior is 30. The result can be seen in Figure 2-2, the posterior quickly converged to the true value that is specified by the constraint. But as in the second example, when a piece of false information was presented, in this case a mean constraint equals 70, the posterior would greatly deviate from the truth (Figure 2-2 b)). As can be concluded from the comparison, the constraint introduced by the exponential term is a hard constraint. It is so hard that the observational data has little effect

on the posterior. Incorporating constraint in this way would lead to unwanted result. In the next part, we explore a more reasonable way to handle such constraint along with observation data.



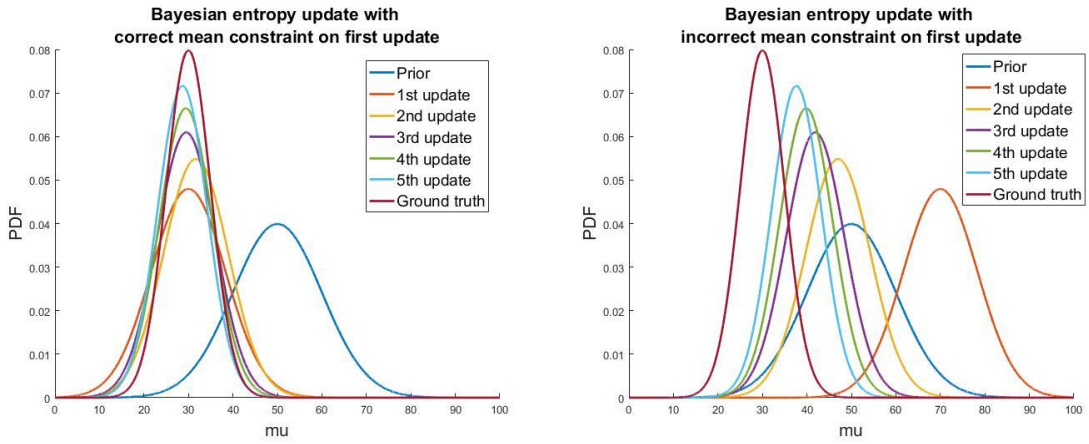
a) b)

Figure 2-2. Updating using data together with a) correct and b) incorrect information

The mean constraint, usually coming from historical data or expert opinion, could not always be trusted. These types of information could potentially help with the prediction when observation data is small. Usually when observation is becoming available in large quantities, we would tend to believe in the data. This can be done in the BEN network by only incorporate the constraint when not enough data is available, as shown in the following example.

In this case, we are considering the mean constraint only in the first step, which means the extra information is only considered in the first update. Similar to the example in the last part, two scenarios were considered, the first one with correct constraint information (mean equals 30) and the second one has an incorrect information (mean equals 70). The results are plotted in Figure 2-3. The results indicate that when a piece of correct

information is given, the constraint can help the posterior converge to the true solution much faster compared with only Bayesian update. While an incorrect constraint can lead to a deviated posterior in the beginning, the posterior can be somewhat corrected following a few more steps of update with observation data.



a) b)

Figure 2-3. Updating with constraint on the first update. a) with correct constraint, b) with incorrect constraint

Comparing with the results from the last example, where the constraint is imposed in every update, it can be seen that the constraints handled in this way is more flexible when dealing with both constraints and data. A more intuitive way to understand this is that in the example in 3.2 the constraint information has been used multiple times, while in the above example it is used only once (in the first update). Obviously, a same piece of information should not be re-used in multiple updates. It is not suggested to keep the constraint information in each update unless there are additional evidences.

2.4 An adaptive model for Bayesian and Bayesian-Entropy method

In the previous sections, it can be observed that the entropy constraint has a strong effect on the distribution function. The order of handling information is critical for the final posterior result. It might raise concerns that what if the given information is not correct. A wrong information could be misleading for human and could cause errors in the estimation for the posterior probability. In the Entropy method, the information comes from evidences. And the principle states that an outcome (experiment or observation) cannot be undone. A method of mitigating the strong effect of the constraint is proposed in this section.

The goal of this section is to mitigate the strong constraint imposed by the entropy term. Sometimes an expert's opinion may not be accurate, but a good estimate. Or the abstracted statistics may be outdated and does not reflect the new measurement data. An intuitive thinking is that when there is not much data, the distribution would lean towards the solution with the entropy constraint. And when observations are getting more, the distribution would tend to believe the data. A simple way of doing this is to add a weighing factor to the exponential term. The proposed equation is expressed as:

$$p(\theta \mid x) \propto q(\theta \mid x)\exp(k\beta g(\theta)) \tag{1.35}$$

where $q(\theta|x)$ is the Bayesian posterior, k is a weighing factor for the constraint. It is a factor to balance the entropy information and data. It is defined as:

$$k = \frac{N}{N+n} \tag{1.36}$$

N is called the confidence related to the constraint β and n is the number of observations available. For example, if a statistic data states that the mean value of a variable is 10 and this piece of information is abstracted from 50 historical data, then the confidence value could be set as $N=50$. According to Eq. (1.36) when no other data is available, k equals 1 and the resulting posterior in Eq. (1.35) is the entropy solution. When the data quantity is overwhelming, k goes to 0 and the resulting posterior is the Bayesian solution.

To demonstrate the mixture model, a numerical example is given. Assume a variable X follows a Normal distribution with known variance, $X \sim N(\mu, 15^2)$ and we are updating the mean μ. The prior for μ is a normal distribution $N(50,10^2)$. A piece of information in the form of moment constraint states the expected value for X is 90 and the information comes from the statistic of 30 historical data. So we set N=30. Random variables are drawn from a Normal distribution $N(80,10^2)$ as observations of X. The posterior distribution of the mean value is updated using the classical Bayesian method, the Entropy method and the mixture model of Bayesian Entropy. The result can be seen in Figure 2-4.

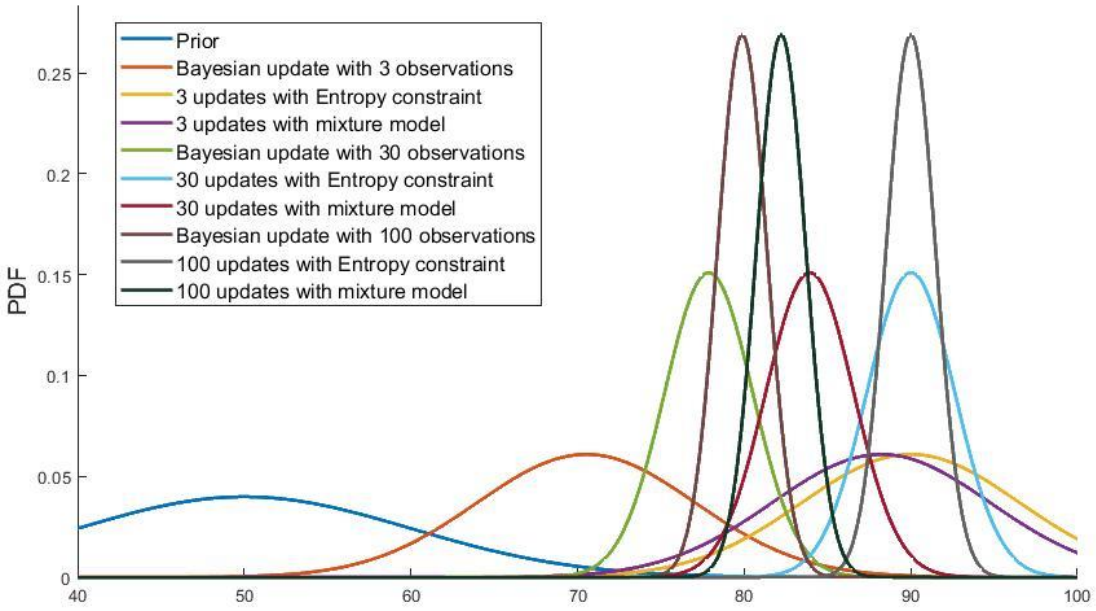


Figure 2-4. The posterior distribution updated with different amount of observations.

As we can see, the behavior for the mixture model is indeed as expected. When the data is less, it is closer to the Entropy result and when the number of observations increases, it is converging to the Bayesian result.

2.5 Conclusion

Based on the above derivation, the BEN method is formulated based on the posterior calculated from the maximum entropy method. The key idea is that in addition to the classical Bayesian method, the BEN method has an extra exponential term in the updating rule that can encode other types of information. Thus, the topology of a BEN is exactly the same with that of a Bayesian network. The BEN method can be regarded as a Bayesian network with encoded extra constraints. In next chapter, the method will be used in diagnostic tasks in engineering problems.

3 Bayesian-Entropy method for diagnostics

The benefit of the Bayesian Entropy method is its ability of adding constraint information into the parameter distributions. In this chapter, the BEN is demonstrated using a simple classification problem to illustrate the key ideas in the proposed method. Following this, the proposed BEN is applied to the damage detection in metal materials. The results are compared with the classical Bayesian method. The comparison shows that the proposed method is a generalized form of classical Bayesian method and can take advantage of the extra information especially when point observations are limited.

3.1 Damage type classification in polymer pipes

This section applies the BEN as classifier to achieve fast learning by the ability to handle extra types of knowledge. Such a classifier is based on the simple Bayes theorem and maximum entropy principle. The additional information such as mean, variance or range data about a certain feature can be coded together with the classical direct point observations. These knowledges were given in the form of constraints using the maximum entropy principle. The classifier is compared with a simple Naïve Bayes network classifier. The classification task is related to the damage type detection in polymer pipes using imaging method. Geometric features are extracted from the pipe imaging and used as variables in the classifier. By encoding extra information into the network, the classifier behaves better when the training data size is small.

3.1.1 Introduction

Damage diagnosis and remaining life prediction of pipeline infrastructure systems is still a challenging problem despite tremendous progress made during the past several

decades, such as the damage accumulation in plastic gas distribution pipes. In order to maintain the safety and integrity of the pipeline systems, accurate damage detection and classification is of great importance. Knowing the damage type and severity can help determine the right maintenance strategies and hence prolong the remaining useful life (RUL) of the pipes. In the pipeline system, the transmission pipelines are usually made of steels and the distribution pipelines are usually made of polymer materials. In this example, we are focusing on a novel damage detection and classification method based on imaging data. A device equipped with an endoscope camera and a laser pattern projector is built. The device can take photo frames as it moves along the pipe. The photos can be processed to reconstruct the pipe inner surface in 3D. Geometric features can be calculated based on the reconstruction. The features are used for classifications.

3.1.2 The Bayesian Entropy Network (BEN) as classifier

When applying the BEN method into classification, the constraint setup would be slightly different than the above derivation. The extra information we could derive from statistic data are often associated with a specific class. Recall a Naïve Bayes classifier, this type of classifier has the simplest network structure with one class node and several feature nodes. The features are assumed to be independent amongst each other and each feature are only and directly connected to the class nodes (Figure 3-1).

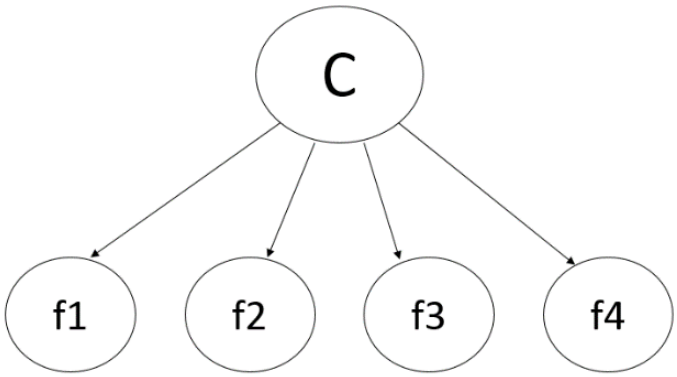


Figure 3-1. The network structure for a Naïve Bayes classifier.

The class node *C* is a discrete node representing the labels. Feature nodes *f1* to *f4* can be continuous or discrete. Each node contains the marginal distribution for the corresponding variable and each edge contains the likelihood function. The network needs a set of data to train each probability function. Giving a new data instance, classification is done by assigning the class label to the one that can achieve the highest posterior distribution.

$$c^* = \arg\max_{j=1\ldots m} p(c_i)\prod_{i=1}^{n} p(f_j \mid c_i) \tag{2.1}$$

Since the prior information about a certain feature would involve the class label, for example, the color of a lemon (class) is yellow (feature), the constraint should be imposed on the likelihood function. The integral form is expressed as:

$$\int_{F_j} p(f_j \mid C = c_i) g(f_j) df_j = G_i \tag{2.2}$$

where $p(f_j|C= c_i)$ is the likelihood function and f_j represents the jth feature and c_i is the ith class label. Consider the normalization constraint and a first order moment with $g(f_j)= f_j$, the posterior likelihood function in relation to the one trained by data is:

$$p(f_j \mid C) \propto \mu(f_j \mid C) e^{\beta f_j} \tag{2.3}$$

where β is the Lagrangian multiplier. Assume Gaussian distribution for the likelihood function, β can be analytically solved. Hence, the final solution for the posterior from BEN is:

$$p(f_j \mid C) \propto \mu(f_j \mid C) e^{\frac{G_i - \mu}{\sigma^2} f_j} \tag{2.4}$$

The solution for a range constraint in this case would be similar and not given in details.

3.1.3 Pipe imaging reconstruction and feature extraction

In the pipeline system, the transmission pipelines are usually made of steels and the distribution pipelines are usually made of polymer materials. The working condition of the pipes are under static pressure, damages in the pipe would accelerate the creep failure process in the pipeline system. Damages include dent, slit, rock impingement and squeeze-off are commonly seem in distribution pipelines (see Figure 3-2).

Figure 3-2. Four types of common damage in pipe: dent, slit, squeeze-off and impingement (from left to right)

A hardware device with an endoscope camera and a laser pattern projector was built for the imaging inspection for the pipe inner surface. Advanced image algorithm can reconstruct the surface in 3D, similar application exists in medical research [35]. The idea is to use triangulation to calculate the angel and distance of any point of the laser pattern relative to the camera. The laser pattern scans through the pipe as the camera is moving along. The 3D surface can be reconstructed based on the laser patterned image frames. In this task, we are trying to classify the damage types based on these reconstructed image data. Due to the laborious process in collecting real pipe imaging data, both simulated data and real data are used for the training and testing of the BEN classifier. The simulation is a Monte Carlo code to generate pipe sections with different damage types and random sizes and locations. White noises were added to the simulation to accommodate the potential noise in the camera sensor. A sample of the real and simulated damage reconstruction can be seen in Figure 3-3.

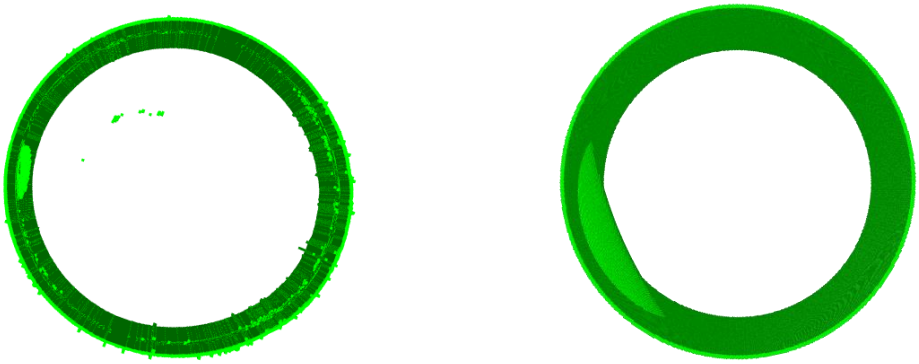


Figure 3-3. Real and simulated pipe 3D reconstruction data.

In order to calculate the geometric features of the damage, the damage needs to be isolated from the pipe. A frame averaging method [36] is used to process the image frames

to do back ground extraction and foreground isolation. It takes the average of a few numbers of the previous frames as the background of the current frame. The damage can then be isolated. Based on the damage, geometric features call be calculated from the 3D reconstruction. For demonstration purpose, four features were proposed, namely the surface area of the isolated damage, the maximum cross section (x-y plane) area, the length in z direction and the ratio of x-y plane projection to z direction length, respectively. The surface area is calculated using the pixel counts of the isolated damage. The maximum cross section is the area of projection for the isolated damage on the x-y plane. The length of the projection in z direction is calculated as the length of the damage. These four features will be used to construct a Naïve Bayesian network similar to Figure 3-1. A total of 110 simulated data instances and 90 lab generated data (a total of 200 data instances) are available for training and testing.

3.1.4 Constraints and classification results

For the four common damages, it is common sense that the slit damage is longer than other types. For this case, the constraint applied into the classifier is the length of a slit damage. It is found from the data that the length of a slit damage is around 100 unit. This constraint can be written as a mean constraint on the likelihood function for slit:

$$\int p(f_2 \mid C = slit)\, f_2 df_2 = 100 \tag{2.5}$$

This information can be regarded as a piece of empirical information from an experienced engineer. An experienced engineer could have some knowledge in certain characteristics of a damage. This piece of information is encoded into the BEN classifier

using the proposed method. The testing accuracy for both the BEN method and traditional Bayesian method is compared in Figure 3-4.

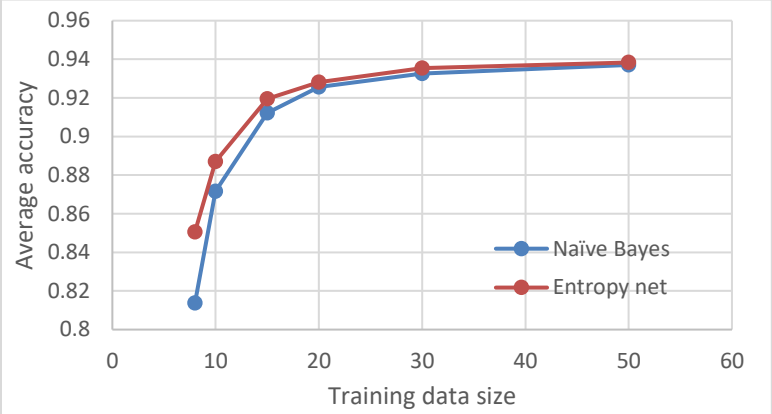


Figure 3-4. Comparison of average accuracy for BEN classifier and Naïve Bayes classifier

As we can see, due to the additional constraint information, the accuracy of the entropy network is significantly higher than that of a Naïve Bayes when there is less training data. When the training size increases, the accuracy from the two classifier tends to converge, for as the training data become available in large quantities, the learned distribution from NB will eventually converge to that of a BEN. And the effect of the additional constraint may become negligible. Figure 3-5 showed the plot for probability density functions (PDFs) of the updated likelihood functions in both cases against the sample's original population. And the mean of the updated likelihood function with constraint is much closer to the original, since the first order moment is the constraint enforced in this problem.

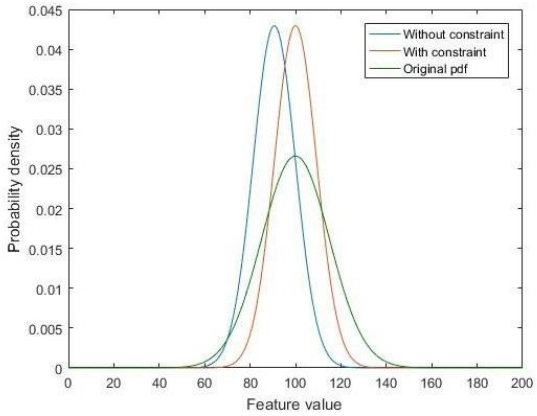


Figure 3-5. The comparison of the updated probability distribution with and without constraint with original fitted PDF from data

3.1.5 Conclusion for BEN classifier for damage classification

Based on the analysis of the result, conclusion can be drawn that the proposed BEN classifier behaves better when the training data size is small. The BEN provided a new way of handling information. Since the extra exponential term can modify and change the learned distribution to the desired position enforced by the constraint, BEN can encode empirical knowledge into the model and achieve fast learning.

3.2 Damage detection in metal plate

This section of the report gives an example in the damage detection in metal plate using the BEN method. The BEN is used as an inference tool for the damage location. The known relationship between the variables in the network can be encoded using the Entropy term. The constraint encoded this way could be understood as the underlying physics in the model. The results are compared with that from a regular Bayesian network (without encoded physics). The inference task involves a non-destructive damage detection framework. Acoustic waves are generated and passed through the metal plate. Based on

the received signal, the damage location can be calculated using an inverse physical model. Although accurate, such models are always time consuming in computation. By using statistical inference, the calculation time could be greatly reduced.

3.2.1 Introduction

Engineering structures have embedded defects such as cracks or voids. With the accumulation of damage through service time, the structures are subjected to fatigue failures. In time detection and maintenance is vital for the prognosis and health management of these engineering structures. Non-destructive testing (NDT) technologies are widely used in the health monitoring. The benefit of such method is that it can detect the damages without destroying the materials. These methods often use acoustic waves, ultrasound or thermography. The key idea is to use a stimulus and its responses to measure the physical properties in the material. Any discontinuity in the materials can be regarded as damage. Typically, the received signal can be used to back propagate via a physical model to calculate damage locations. In this example, piezoelectric sensors are used to generate and receive acoustic waves that passes through a metal plate. Features from the received signal can be extracted and used to statistically inference for the damage location.

3.2.2 Experimental setup

As illustrated in Figure 3-6, two piezoelectric sensors are installed on an aluminum plate. There is a crack damage on the plate in between the sensors. One of the sensors acts as a signal generator and actuates a signal. The signal then propagates through the plate to the other sensor that is acting as a receiver. The crack in the plate introduce a discontinuity in the material property and hence affect the received signal. Some existing study can use

this kind of non-destructive method to reconstruct the metal plate and calculate the position of the crack through a physical based model [37]. But usually the computational cost for a physics-based model is high, and, in most cases, it would take more than two sensors to be able to calculate the damage position. Our goal is to use a BEN model for this problem to predict the location of the crack.

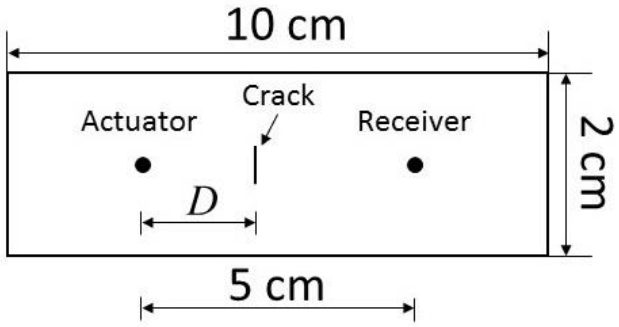


Figure 3-6. Simulation setup for the composite plate and sensors.

To get a training and test data set, numerical simulation was done for different damage location *D*. The received signal went through a few simple pre-processing such as de-noise and Fourier transform, some features can be calculated such as amplitude, frequency, phase angle and time of arrival. Amongst these features, three were chosen to fit a network model, namely the maximum amplitude (A_{max}), time of arrival (T) and frequency (f). The crack is simulated at 9 different location between the two sensors, namely at D=0.5cm, 1cm, 1.5cm to 4.5cm, each 0.5 cm apart. At each location, 30 simulations were done. A signal was generated at the actuator and a forward calculation is done using the K-space method for the signal at the receiver [37]. Each simulation has a varied crack length and noise were randomly added into the sensor. It can be observed from the data that there is a relation between the location of the damage and maximum amplitude. The maximum amplitude is an exponential function of the damage location d. The detail

of the physics for this relation is beyond the scope of this paper. This will be used as the prior knowledge for this case.

The received signal would differ for various damage location *D* as illustrated in Figure 3-7. Based on the characteristics of the received signal, three features were selected for building a network model, namely the maximum amplitude (A_{max}), time of arrival (*T*) and frequency (*f*). Numerical simulation was done at different damage location to generate data for training the network. The crack was set at different D values, from 0.5cm to 4.5cm, each 0.5cm apart. 30 simulations were done for each crack location. For each simulation, random noise has been added to the actuated signal to consider the environmental variability and the crack has a different length to consider the geometric uncertainty.

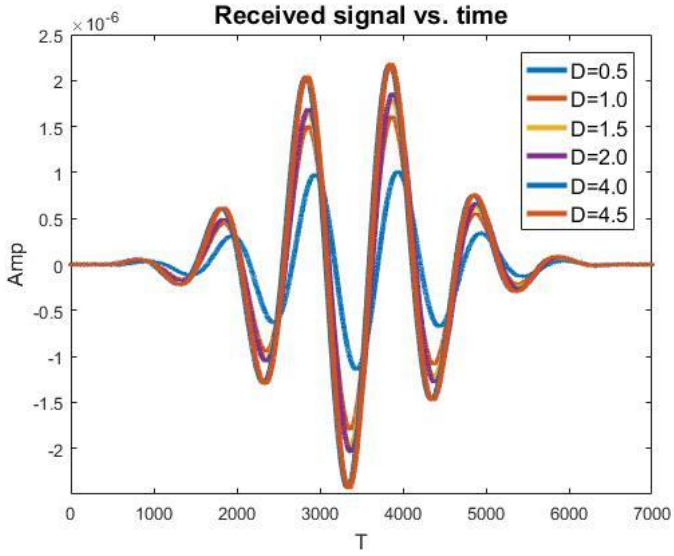


Figure 3-7. The received signal varies as the location of the damage changes.

3.2.3 Feature calculation and statistical inference

The above features are extracted from the receiver signal. In feature-based damage detection method, the peak value of the waveform, A_{max}, and time of flight T are indicators of the defect existence [38], [39][40]. In this numerical model of a through-thickness crack, it's seen that the amplitude decreases due to the energy trapped in the crack and only part of the wave is transmitted to the receiver in Figure 3-7. The time of flight also increased as the mechanical property change around the crack area. The parametric study of the amplitude changes and time of flight T is observed by varying the crack location. The time of flight T are estimated by a Hilbert Transform on the received waveform [41]. The envelope of the first wave-packet and the source from the actuator is calculated. The time of arrival can be estimated by the time difference between the envelope of source signal and the one of the receiver signals. The peak amplitude is computed by finding the maximum in the time series of the receiver signal. The frequency is achieved by a Fourier Transform on the received signal. To simplify the Bayesian network model and the efficiency, the first arrived wave packet is used for feature extraction. It can be seen that the first-wave packet is strongly affected by the crack in Figure 3-7.

From the processed data, it can be observed that there exists a correlation between the location of the damage and the maximum amplitude ($A_{\max}$). By a simple regression analysis of the data, the correlation between D and $A_{\max}$ can be expressed by a logarithm function: $T = f(A_{\max}) = a\ln(A_{\max}) + b$. The detailed physics behind this relation is beyond the scope of this paper and this piece of information will be assumed as a known relation

and encoded into the BEN model. The constraint involving the relation between the two variables can be expressed as:

$$\int p(D \mid A_{\max}) D dD = f(A_{\max}) \tag{2.6}$$

If we assume normality for the prior distribution, the target posterior of the joint distribution for D and $A_{\max}$ can be solved:

$$p(D, A_{\max}) = q(D, A_{\max}) \exp\left(\frac{f(A_{\max}) - \mu}{\sigma^2} D\right) \exp\left(\frac{\mu^2 - f^2(A_{\max})}{2\sigma^2}\right) \tag{2.7}$$

where $q(D,A_{\max})$ is the joint distribution of D and $A_{\max}$ trained by data. μ and σ is the mean and standard deviation for the conditional distribution of D given $A_{\max}$, $q(D|A_{\max})$.

The network model is shown in Figure 3-8. We used a Naïve network where the three features, namely the maximum amplitude ($A_{\max}$) and the frequency (f) of the receiving signal, as well as the arrival time (T), are assumed to be independent and are solely related to the damage location D. There is a total of 270 instances from the simulation data. 170 of them are used to train the network for the likelihood function and rest 100 data are used to infer the damage location. In a Bayesian treatment, the feature values in each test data was used to update for a distribution for D. And the mode of the distribution is considered as the predicted damage location. In addition to the Bayesian method, the BEN model considers the constraint function and used Eq. (2.7) to calculate the posterior distribution for D.

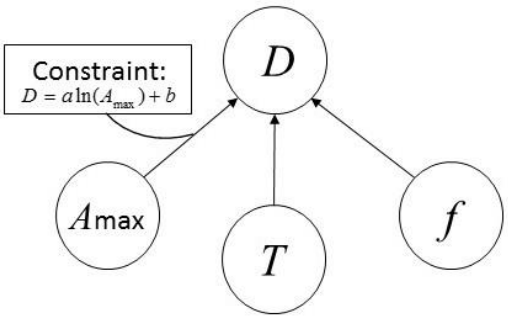
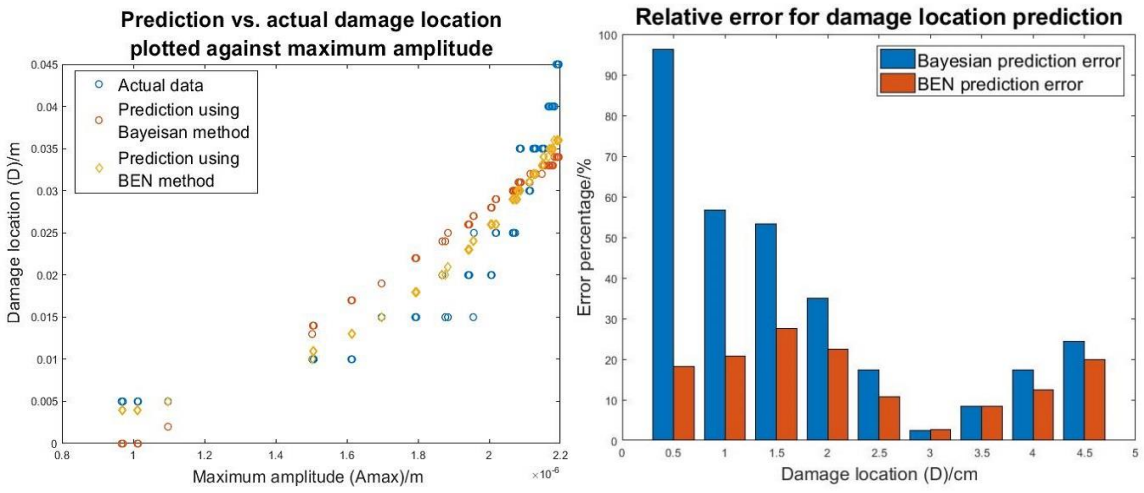


Figure 3-8. The network model for damage detection

The expected value for the updated distribution of *D* is regarded as our estimate for the damage location. In addition to the Bayesian updating, the BEN method considers the constraint given in Eq.(2.6) and updates the distribution with Eq.(2.7). As a result, with the given constraint, the prediction using BEN is closer to the actual value than that of the classical Bayesian method. As a matter of fact, the frequency *f* stays constant for all data instances, which is no surprising as the fundamental principle for wave propagation. So, this feature would have no effect on the prediction result. Figure 3-9 a) has shown the comparison of the prediction result from BEN and Bayesian with the true value. The prediction from the BEN method seems closer to the true value compared with the traditional Bayesian method due to the extra information. Figure 3-9 b) showed the average relative error for the prediction at each location. It can be seen that by using the extra information, the BEN can achieve a relatively accurate prediction than the traditional Bayesian.



a) b)

Figure 3-9. Prediction of the damage location for BEN and Bayesian method vs. actual position (left) and the average relative error at each location (right).

3.2.4 Conclusion for BEN for damage location inference

Although the improvement is only marginal, comparing with the high computational cost of a physics-based model, the proposed BEN can achieve a plausible accuracy with a much faster processing time. Also, only two out of the three features were useful in describing the joint relations. The author believes the estimation can be more accurate should more features were considered.

3.3 Bayesian text embedding for aviation accident classification

Text is another type of information that is not typically handled in the Bayesian framework. Existing method often treat text labels as discreet integers. Other popular text mining method creates dictionary for all available texts and represent each word or phrase via a vector. This section introduces a Bayesian text embedding method for analyzing and classifying the accident type in aviation.

3.3.1 Introduction

One of the easiest ways of processing text-oriented data is processing them as discrete categorical data by assigning random integers. However, the method can be questioned for the rationale behind the value for each integer. For example, as the damage classification problem presented in section 3.2, the class labels are randomly assumed as integers 1 to 4. Why would the value for slit damage in pipes smaller than an impingement? The random assignment for the class label may not cause a big issue. In [42], the relationship of the communication error to runway incursion accident at airport is studied through a Bayesian network. The study used random integer numbers to represent different types of communication error and assigned constraints on the expected value for the communication error. This, however, is questionable for why a certain type of communication error would has a larger value than the others. By randomly assign a different set of integer values for the communication error, the expected value constraint would change or sometimes, cannot be satisfied. This issue is the main motivation in this research for finding a rigorous way of handling text-related data.

Text is the most important way of documenting knowledge. Advancements in Machine Learning (ML) and Artificial Intelligence (AI) algorithms have enabled the development in text mining technologies. Text mining can increase the efficiency for discovering useful knowledge and extracting information and data mining from text-based documents such as webpage, text report etc. [43]. Comparing to manually reading and extracting information, text mining method can accelerate the process by magnitudes. In order for a ML algorithm to understand the underlying content within a text document,

Natural Language Processing is needed. Existing methods such as Bag-of-words [44][45], n-gram [46] and Word2Vec [47] has been developed for processing structured or unstructured text into arrays. This array can be used for the training of a ML model such as Neural Network and dimension reduction methods for the classification or semantic analysis for the text documents. Recently, Long Short-Term Memory (LSTM) method has found particularly useful for text mining [48][49].

The National Transportation Safety Board (NTSB) is a U.S. government agency dedicated to the accident investigations in civil transportations. The NTSB database contains rich text report for aviation accident in the US. The NTSB report includes narrative description as well as the event sequence. Aviation Safety Reporting System (ASRS) is also an information source containing text narratives that is voluntarily submitted by aviation personnel. The data archive is a mixture of numerical and text data. Existing research has been done in studying the accident/incident report aiming at analyzing the patterns in aviation accidents in the hope of the ability to predict and prevent unforeseeable situations. [50] used random forest method for analyzing the severity of accident related features based on NTSB report. The considered features are categorical variables such as flight phase, weather and aircraft type. The patterns in general aviation accidents are evaluated through text mining of NTSB reports in [51]. Two developed models, ASIAS Information Retrieval and Extraction System (AIRES) by MITRE, and a commercial software package STATISTICA, are evaluated for the performance in predicting the fatal and non-fatal accidents. [52] presented a combination of Support Vector Machine (SVM) and Deep Neural Network (DNN) for predicting abnormal event in

aviation. The study used text mining results from ASRS data to train an SVM and DNN for the prediction of event consequence (severity of accident). Statistical analysis has been done by data-mining the NTSB data in [53] focusing on helicopter accidents. An LSTM model has been applied in [54] for classifying accident type from NTSB report data. The classification is done on accident or incident, damage or non-damage and fatality or no fatality. The accuracy ranges from 70% to 90%. The NTSB database contains both narrative description and the event sequence as categorical features. The event sequence data is a combination of text and numerical features. In the above-mentioned researches, those related to text mining uses bag-of-words algorithm to translate text description into high dimensional array then use ML method or principle component analysis (PCA) to find patterns or correlations between the word vector. Those directly use labeled features treat them as categorical data or assign random values.

A Bayesian task embedding method was proposed in [55]. Originally, the method was developed as a tool for incorporating data from different models as text labels to enable fast design optimization along with available numerical data. The key idea of the approach is not only learning the inherent function relations of each model but also detecting the similarities between each model. In this research, the method is adopted as a tool for handling text labeled data from NTSB report to classify the severity of an aviation accidents/incidents along with the available numerical features.

3.3.2 Bayesian text embedding

The original Bayesian embedding for design optimization was proposed as a deep Gaussian Process (GP). The deep GP has two layers, the first layer is called the Bayesian

embedding GP that generate reasonable embeddings of the text features into scalar value or array. The second layer is another GP that takes the output scalar/array as well as other numerical futures as input and outputs the system response. The structure of the deep GP model is illustrated in Figure 3-10. Text features are referred as general input X_g and numerical features as X_r. The first GP map the text label into a latent variable Z. Z and X_r are concatenated into the full input for the second traditional GP model to predict system response.

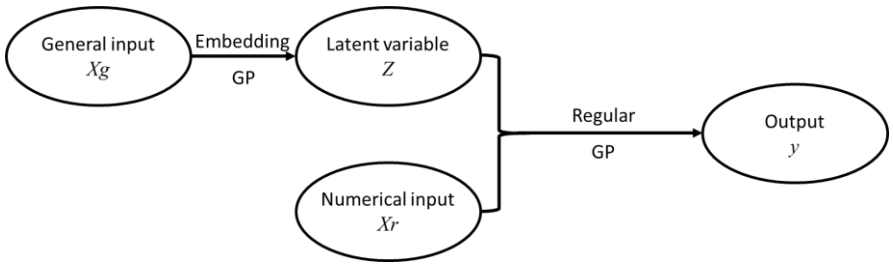


Figure 3-10. A schematic illustration for the structure of the deep GP.

Based on the previously developed research, this study builds a hybrid model by adopting the first GP layer for text embedding and replacing the second GP with a NN model for damage type classification. The output of the embedding GP is concatenated with the numerical features as the input to a feedforward NN. The output of the NN is the severity of the involved aircraft damage. An illustration of the model structure is shown in Figure 3-11.

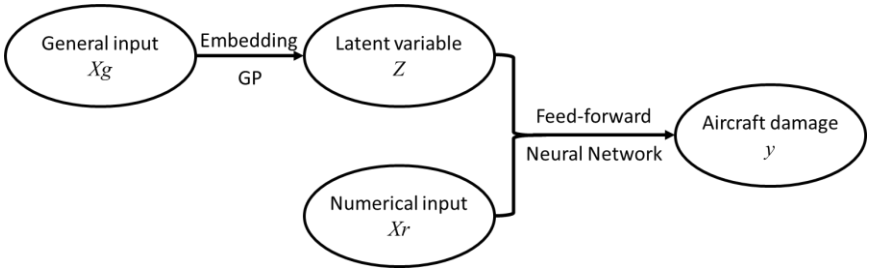


Figure 3-11. Schematic illustration for the proposed hybrid model structure for aviation accident classification.

GP is a statistical process that has been widely applied in function fitting and regression in engineering field. The GP, denoted as $GP(\mu(\cdot),k(\cdot,\cdot))$, is a distribution over functions with mean function μ and covariance function k that is defined through a set of hyperparameters. The function value at each input location is a Gaussian distribution. According to the definition in [56], the function output $\boldsymbol{f}$ from a GP regression model can be written as a Gaussian distribution:

$$\boldsymbol{f} \sim N(\boldsymbol{\mu}, \mathbf{K}) \tag{3.8}$$

where $\boldsymbol{\mu}$ is the mean vector and $\mathbf{K}$ is the covariance matrix. This is the GP prior used for inferring the function value at desired prediction point. The function prediction output $\boldsymbol{f}_*$ can be written as a joint distribution with the prior as:

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}_* \end{bmatrix} \sim N\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^{\,T} & \mathbf{K}_{**} \end{bmatrix} \right) \tag{3.9}$$

Hence, the conditional probability for the prediction $\boldsymbol{f}_*$ can be written as:

$$\boldsymbol{f}_* \mid \boldsymbol{f} \sim N(\boldsymbol{\mu}_* + \mathbf{K}_*^{\,T}\mathbf{K}^{-1}(\boldsymbol{f} - \boldsymbol{\mu}), \mathbf{K}_{**} - \mathbf{K}_*^{\,T}\mathbf{K}^{-1}\mathbf{K}_*) \tag{3.10}$$

The mean function and covariance function are defined through a set of hyperparameters θ, which are typically solved by Maximum Likelihood Estimator (MLE) by maximizing the likelihood function given the observed data:

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) \propto \exp\left[-\frac{1}{2}\log|\mathbf{K}| - \frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^T \mathbf{K}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}) - \frac{n}{2}\log(2\pi) \right] \tag{3.11}$$

The GP can be a flexible regression tool due to the variance types of covariance function that is defined through the hyperparameters. One commonly used covariance function is the squared exponential, which takes the form of a Gaussian kernel:

$$k(x, x'; \boldsymbol{\theta}) = \sigma^2 \exp\left[-\frac{1}{l}(x-x')^T (x-x') \right] \tag{3.12}$$

where l and σ are the hyperparameters. This covariance function reflects the correlation between inputs through the Euclidean distance. In the Embedding GP layer, the text input elements are embedded to a d_z-dimensional latent Z. To eliminate the effect of the Euclidean distance between the input, a white noise kernel is used as the covariance function defined as:

$$k_{white}(x, x'; \boldsymbol{\theta}) = \sigma^2 \delta_{x,x'} \tag{3.13}$$

where $\delta_{x,x'}$ is a delta function. $\delta_{x,x'}=1$ when $x=x'$, and equals zero otherwise. The embedding GP can be expressed as:

$$Z \sim GP(0, k_{white}(\cdot,\cdot)) \tag{3.14}$$

The output response from the zero mean white noise GP is then merged with the numerical input to form a Neural Network.

The NN model used in this study is a simple 2-layer feed forward NN. An illustration for the NN model is plotted in Figure 3-12. The NN has a hidden layer and the output layer. Both layers are linear layers, which means the value in each neuron y is calculated as a linear function of the neurons $\mathbf{x}$ in the previous layer as: $y=\mathbf{w}*\mathbf{x}$, where $\mathbf{w}$ is the weight that is to be trained. The last layer is a softmax layer that normalizes the output and selects the highest value as the class label.

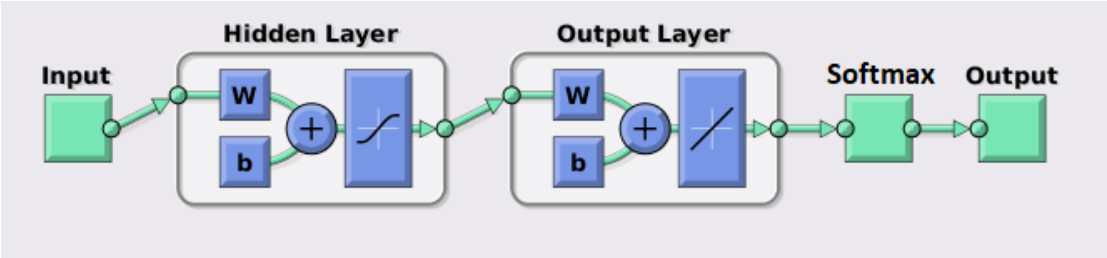


Figure 3-12. Schematic illustration for the feedforward NN structure.

The training of the hybrid model involves the optimization of the GP likelihood and the NN loss function with respect to the hyperparameters and the NN weight. For the NN loss function, since this is a classification type of problem, we use the Cross-Entropy loss function or log loss defined as:

$$H = -\sum_{i=1}^{C} y_i \log(p_i) \tag{3.15}$$

where p_i is a predicted probability from the softmax layer output and y is a logic indicator if the predicted class label is true (1) or false (0). C is the total number of class labels. The Cross-Entropy loss measures the divergence of the predicted probability to the actual label. A perfect model can have a zero Cross-Entropy loss.

The posterior for the latent variable **Z** may not be analytically tractable according to [57]. This has put a challenge on the inference for the hyperparameters. The posterior of the latent variable **Z** given the text input X_g can be expressed as:

$$p(\mathbf{Z} \mid \mathbf{X}_g) = \prod_{i=1}^{n} \prod_{j=1}^{d} p(z_{i,j} \mid 0, \sigma^2) \tag{3.16}$$

where n is the dimension of the text input and d is the dimension of the latent variable **Z**. The probability function $p(z_{i,j}|0,\sigma^2)$ represents the probability of $z_{i,j}$ given the zero mean and white noise variance Gaussian distribution. A variational posterior based on mean-field approximation is given to approximate the true posterior of the latent variable **Z** as:

$$q(\mathbf{Z}) = \prod_{i=1}^{n} \prod_{j=1}^{d} p(z_{i,j} \mid m_{ij}, s_{ij}) \tag{3.17}$$

where m_{ij} and s_{ij} are the variational parameters. The log likelihood of the embedding GP layer is then defined as the Kullback-Leibler (KL) divergence between the variational distribution and true posterior:

$$ll = \int_{Z} q(\mathbf{Z}) \log \frac{p(\mathbf{Z})}{q(\mathbf{Z})} d\mathbf{Z} \tag{3.18}$$

An evidence lower bound (ELBO) is then derived to replace the exact log likelihood for optimization:

$$ELBO = \int_{Z} q(\mathbf{Z})(\log p(\mathbf{Z}) - q(\mathbf{Z})) d\mathbf{Z} \tag{3.19}$$

The total loss of the hybrid model can then be defined as the sum of the Bayesian embedding GP loss in Eq. (3.19) and the NN loss in Eq. (3.15) as:

$$Loss = H + ELBO \tag{3.20}$$

By minimizing the total loss in Eq. (3.20) we can solve for the hyperparameters in the Bayesian embedding GP layer and the weights in the NN layer. $\in R^6$

3.3.3 NTSB data and data pre-processing

The NTSB provides an abundant data archive for aviation accidents from 1961 to present date and recorded all accident reports since 1982. A new NTSB accident database system (eADMS) was implemented after September 2008. The eADMS datasets are in the Microsoft Access format. The hierarchical architecture of the data base is shown in Figure 3-13. It includes a series of tables containing related information about the recorded event such as aircraft status, weather and event consequences etc. The tables are all linked by the keyword *ev_id*, which is a unique code representing each event. On the top level is the *event* table, which records the time and location of accidents or incidents. The *dt_event* on the second level is a "detailed table" that contains event overview such as the weather status, runway condition and event severity (damage to aircraft, injuries and fatalities). An aviation accident/incident can have one or more aircraft involved. The *aircraft* table links involved aircraft ID with the event ID. For each aircraft, the details such as the flight and cabin crew information, aircraft engine and maintenance information and detailed injury records are listed in the third level. The *Occurrences* table records the abnormal activities in each flight phases. Belonging to the *Occurrences* table is the *seq_of_event* table in the fourth level which records the most detailed of sequences that explained how each event eventually leads to the final accident/incident. In the *seq_of_event* table, each entry

includes a subject code and a modifier code or a personal code that specify an abnormal event.

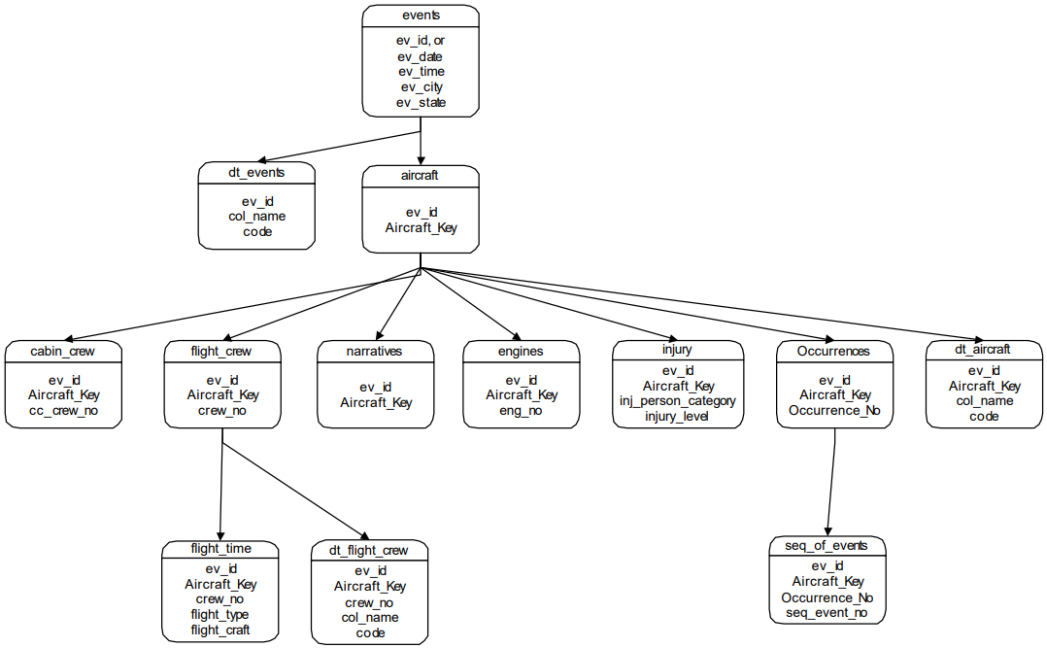


Figure 3-13. The data structure for the ADMS data from NTSB.

In this study, training data are filtered from the NTSB data. The subject codes in the *seq_of_event* table are regarded as one of the main text inputs. The length of the subject code in each event record may vary in length. We filtered out the event with less than 4 sequences of event. For event sequences that is longer than 4, we choose only the first 4 entries to store. Other selected text inputs are airspace type when the accident happen and load description which describes what the aircraft is used for. The numerical filtered includes engine power, visibility wind speed and wind direction, altimeter reading at the time of event, and temperature etc. The aircraft damage is selected as the class label for prediction. The aircraft damage has 4 type: Destroyed, substantial, minor and none. After

the filtering, a total number of 6736 data instances with 17 features (11 numerical input and 6 text input) are stored for the training and testing of the hybrid classification model.

3.3.4 Accident type classification for aviation accident

The filtered aviation accident/incident data has 17 features, 11 of which are numerical data related to weather information and aircraft parameters and the other 6 are event sequence, aircraft type and load description. Table 3-1 listed details of the filtered data columns and their meanings.

Table 3-1. Selected features and their meaning for the filtered data.

Feature type	Feature name	Description
Numerical input	Power_units	The power output of the aircraft engine
	Altimeter	Altitude from barometric pressure at the event
	Apt_dist	The distance of the involved airport to the event site in statute miles
	Apt_dir	The direction of the involved airport to the event site in degree
	Apt_elev	The elevation of the involved airport in feet
	Gust_kts	Wind gusts in knots
	Vis_sm	The visibility at the time of the event in statute miles
	Wind_dir_deg	The local indicated wind direction
	Wind_vel_kts	The local indicated wind speed
	Wx_dew_pt	Dew point temperature at the time of the event
	Wx_temp	Ambient air temperature at the time of the event
Text input	Airspc_type	The type of airspace the aircraft was operating in
	Load_desc	Description of the aircraft load
	Subj_code_1	Subject codes are used to identify the individuals, equipment, processes, or phenomena that contributed to the mishap event.
	Subj_code_2	
	Subj_code_3	
	Subj_code_4	

The 6 text inputs are embedded into a multi-dimensional latent vector. We arbitrarily chose the length of the latent vector to be 4. That is, each text input feature will be embedded into a 4-dimensional latent, so the total length of the latent variable ***Z*** is 24. Combining with the 11 numerical features, the input layer of the NN model has 35 neurons. We set the numbers of neurons in the hidden layer as 128. Figure 3-14 showed a detailed structure for the hybrid model.

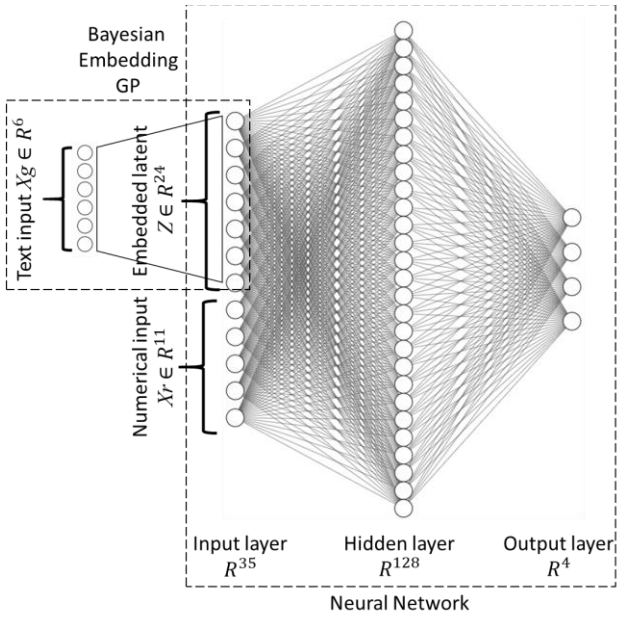


Figure 3-14. Detailed structure for the proposed hybrid model for aviation accident/incident classification.

From the total of 6736 available data instances, 60% are used for training the model, another 40% used for testing. Figure 3-15 showed the value of loss function as the number of epochs. It is observed that the loss function is converging at 300 epochs. For the current dataset, the classification accuracy can reach up to 72%.

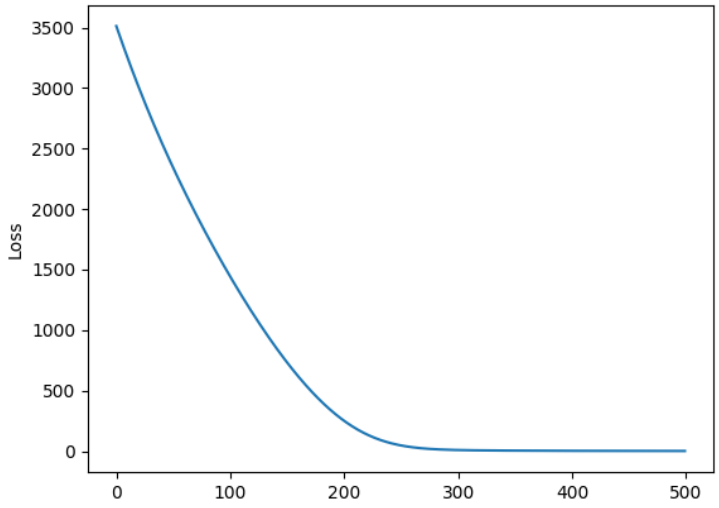


Figure 3-15. The value of loss function as a function of epochs for the hybrid model.

A comparison was done using a pure NN model. In the pure NN model, the text inputs are treated as categorical features and was processed into a multi-dimensional array. The length of the array equals to the unique elements in the text inputs. The text inputs for each data instances can all be assigned to such an array where the value equals one when the instance has the text content and zero otherwise. A separate NN model was built with 256 hidden layers. The accuracy is around 68%.

3.3.5 Conclusion for Bayesian text embedding

This section presented a novel hybrid modeling method for handling text and numerical information. The hybrid model consists of a Bayesian Embedding GP that maps the text input into latent vectors, and a Neural Network that takes the latent vectors together with the other available numerical features to predict the class label. Unlike assigning random integer values to text data or other bag-of-words-based embedding method, the

proposed method uses a white noise GP without considering the Euclidean distance to map the text input into an arbitrarily defined high dimensional latent vector. The method was trained and tested with filtered data from NTSB. The filtered data contains both numerical and text features related to aviation accidents/incidents. The hybrid model was used for the classification of aircraft damage severity in the accident/incident. According to the result, the test accuracy for the proposed model can reach 72%. Whereas under the similar training condition, an ordinary Neural Network model that takes text data as categorical input can achieve an accuracy around 68%. While the improvement is only marginal, this proves that the proposed method is a more rigorous way for text embedding. The test accuracy can be further improved by tuning the Neural Network parameters and changing the latent dimension for the Bayesian Embedding GP.

3.4 Conclusions

A novel BEN method as a general tool for classification and inference is presented in this chapter. The method combines a classical Bayesian part to handle point data and an exponential part to encode constraints. Several conclusions can be drawn based on the proposed study:

1) It is shown that the proposed Bayesian Entropy Network is a generalized Bayesian Network model and has the same modeling structure;
2) Different types of information can be encoded using the entropy principle with analytical expression of an exponential term and point observations are handled by the likelihood function. Various type of information constrains, including moment

constraints, range constraints, and general functional constraints have been developed;

3) It is observed that the encoded extra information can enhance the performance if the number of point observations is small. If point observations are huge, both BEN and BN converge to the same results;
4) When information constraints and point observations are both present, the final posterior distribution depends on the order of updating using different types of information;
5) In general, the proposed BEN shows its flexibility to handle multiple types of information commonly seen in engineering practice and can serve as a generalized information fusion tool for system reliability analysis.

A Bayesian Embedding Gaussian Process is used as a text embedding method for a hybrid classification model. The proposed method provides a rigorous way of handling text information. Unlike usual text embedding method, the logic behind the Bayesian Embedding GP does not require a Euclidean input. Comparing with the bag-of-words based method, the Bayesian Embedding has shown certain benefit for classifying aviation accidents based on text and numerical features. The proposed method can be used as an information fusion tool to handle text and numerical data.

4 Bayesian-Entropy method for prognostics

The prognosis and health management for a complex large-scale engineering system is a challenging problem. One example for a large engineering system is the national airspace system (NAS). The NAS is a complex system with coupled physical and human information. The research work reported in this section is a part of the NASA University Leadership Initiative (ULI) program. The ULI project aims to address the safety issues in the current NAS and develop new technologies for real-time or near real-time safety assessment and prediction for the Next Generation air transportation system (NextGen). The proposed work in this project includes the information fusion from various sources for the risk prediction as well as the uncertainty quantification and management in aircraft operations. In this chapter, the Bayesian Entropy method is used as a tool for information fusion in the NAS system for prognostic tasks such as risk assessment and flight trajectory prediction.

4.1 Runway incursion cause identification via BEN classifier

The air traffic control (ATC) system is critical in maintaining the safety and integrity of the National Airspace System (NAS). This requires the information fusion from various sources. This information can include human experience, historical data etc. These knowledges, once written in a mathematical format, can be incorporated into the classical Bayesian framework via the Bayesian Entropy network. This section introduces a hybrid network model called the Bayesian-Entropy Network (BEN) that can handle various types of information in the NAS system. The example is related to the prediction of the cause of runway incursion. A network model studying different sources of error is used to make

predictions for the cause of runway incursion. The training and validation data are extracted from existing accident report in the Aviation Safety Reporting System (ASRS) database. The results are compared with that of the traditional Bayesian method. It is found that the BEN can make use of the available information to modify the distribution function of the parameter of concern.

4.1.1 Introduction

Runway incursion is defined as the incorrect of the presence of aircraft in landing and take-off area [58]. It can cause critical accidents and property damage. This example explores a Bayesian network model to classify the cause for runway incursion during take-off. The structure and the variables in the network are derived by the ASRS report database. The ASRS is a reporting system where pilots, controllers and operators voluntarily submit accidents and incidents during aviation operations. It can be acted as an educational source for the safety operations for pilots and controllers and for the overall NAS.

4.1.2 Data from ASRS report

The runway incursion can be regarded as an incident lead by a series of small errors. The network topology is built using features extracted from ASRS runway incursion accident report. The features are extracted by manually read the accident report. A total of 331 report involving runway incursion between 2014 and 2017 have been found. Due to the heavy workload of analyzing the report only a small number of reports were studied and 37 out of these reports were found to be useful with common descriptions of how the accident happened. It is found that, in the 37 reports, the runway incursion is caused by communication error between the pilot and ATC tower. The cause for runway incursion

can be abstracted using a procedure diagram shown in Figure 4-1. The runway incursion can be categorized into three types. The communication error are the leading factor of runway incursion. And a few basic features are chose as the influencing factor for communication error.

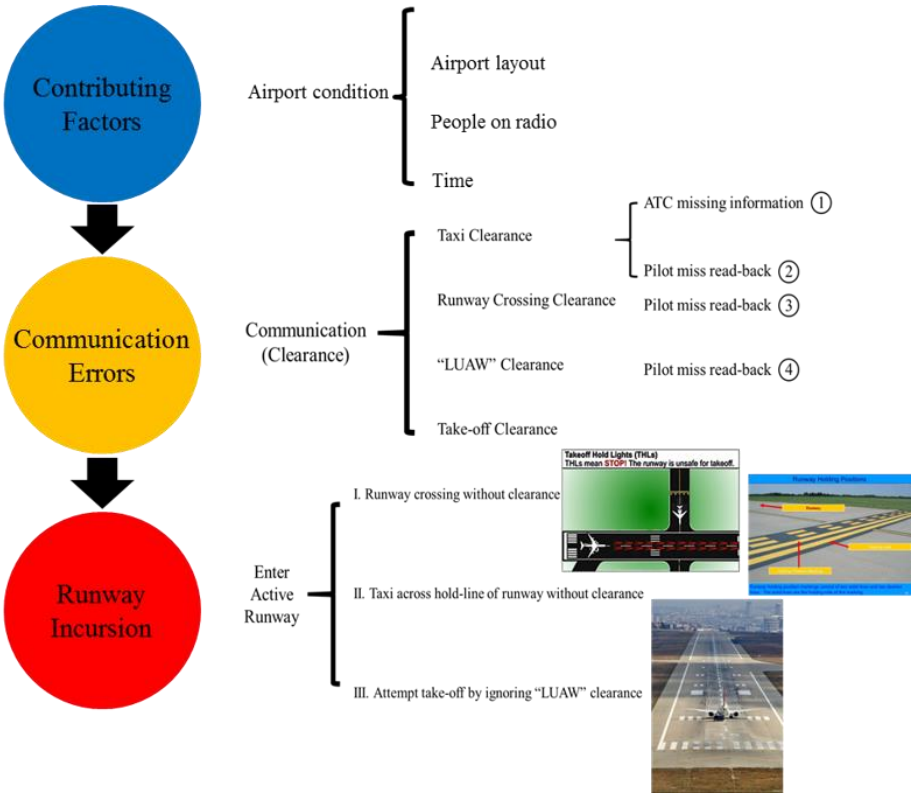


Figure 4-1. The procedure diagram for runway incursion accident

Three types of a runway incursion is identified:

1. Runway crossing without clearance,
2. Taxi across hold-line without clearance,
3. Attempt take-off by ignoring Line Up and Wait (LUAW).

Four types of communication error can be found in the 37 reported cases, which are:

1. ATC operator issues ambiguous taxi clearance (taxi clearance communication error on ATC side)
2. Pilot miss read-back on taxi clearance (taxi clearance communication error on the pilot side)
3. Pilot miss read-back on runway crossing clearance (runway crossing communication error)
4. Pilot miss read-back on LUAW clearance (LUAW communication error)

In addition to the communication error, some attributes in these 37 reports were extracted as basic features, they are: number of runways in the airport, the runway layout of the airport (whether there is intersection or not), number of people on the same radio frequency and the time of the day at the accident. Based on these available features, a network model for runway incursion classification is built in Figure 4-2. The network assumes that the four basic features are independent from the occurrence of runway incursions but can be a contributing factor of the communication error. The four basic features are all assumed to be independent with each other.

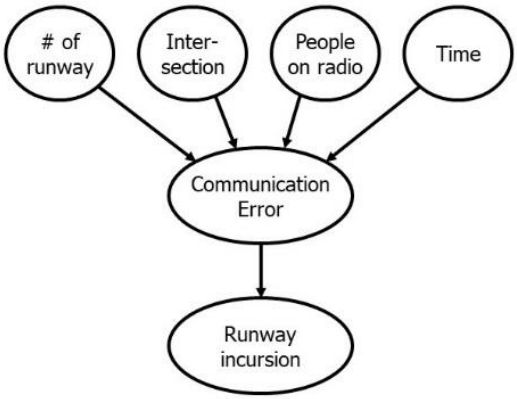


Figure 4-2. Bayesian network for runway incursion

A random train-test split is done to the 37 data instances. The training set is used to calculate the conditional probability table for the network and the test set is used to validate and test the classifier. The test was done using only the four basic features to infer for the communication error type and runway incursion. Due to the limited data, a Bayesian network cannot achieve plausible accuracy.

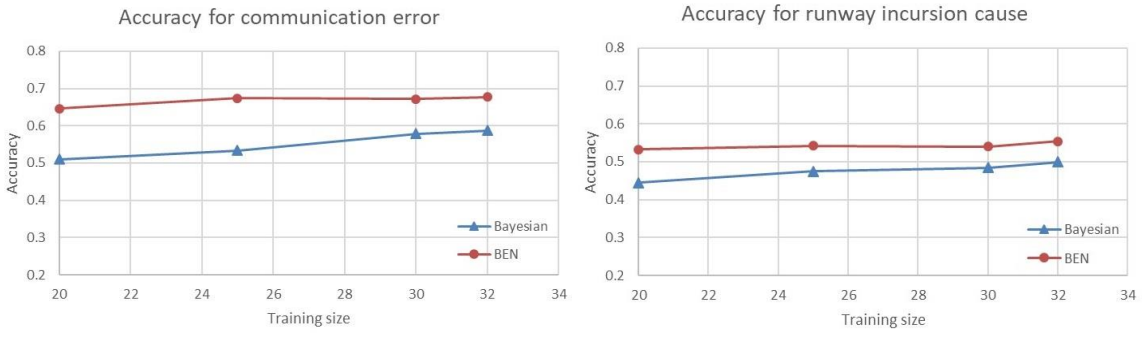
4.1.3 Constraint information from empirical information

When reading the accident report, it is found that there are certain patterns for the correlation between variables, for example, when the number of people on the radio frequency is less, taxi clearance communication error is more likely to happen on the pilot's side. Such information may come from an experienced operator, or a report reviewer who has read a lot of the accident report and was able to generate this type of empirical knowledge. These knowledges can be encoded into the network using BEN method as the entropy information.

The entropy information included in the BEN model are:

1. At night, a runway crossing communication error is more likely to happen.
2. When the number of people on the radio frequency is less, a taxi clearance communication error is more likely to happen on the pilot's side.
3. When the taxi clearance communication error is on ATC side, the cause for runway incursion is more likely to be cross runway without clearance.
4. When the taxi clearance communication error is on pilot side, the cause for runway incursion is more likely to be taxi across runway hold line.
5. LUAW communication error can only lead to and is the only reason for attempt take-off without clearance.

Since the communication error and runway incursion are all categorical nodes, integer values such as 1, 2, 3, 4 are assigned accordingly. The above constraints are all considered as mean constraints (1, 2, 3, 4) or range constraints (5). The training was done in a similar manner as the Bayesian approach. With the encoded constraints, the testing accuracy is plotted comparing with the accuracy from the Bayesian method in Figure 4-3.



a) b)

Figure 4-3. The average accuracy of classification for a) types of communication error and b) cause for runway incursion

Although the accuracy is still not satisfying, due to the encoded constraints, the BEN has around 10% improvement comparing with the classical Bayesian method. Since the lack of data, the result might not be representative. The author will keep working on analyzing reports to extract more data for sufficient training and testing set.

4.1.4 Conclusion for BEN in runway incursion identification

The state-of-the-art method for accident prediction in the NAS is mostly relying on human operator, such as Flight Risk Analysis Tool (FRAT) [59] and Safety Management System (SMS) [60]. Human are subject to fatigue and performance would vary for different operator. While the application of BEN can achieve an automated prediction scheme that can be robust and reliable. More data needs to be extracted from the report database for the validation of the proposed method and network structure.

4.2 BEN in ATM risk control

The risk in air transport may be coming from various factors such as aircraft conditions (maintenance and design of the aircraft), environmental conditions (weather and terrain), operation and management etc. [61]. Human error has always been considered as a critical influencing factor in air traffic management (ATM) [62]. Many research work [63][64] focuses on a causal network model to infer the air traffic risk probability. Bayesian network is a great tool for causal inference as its ability to model the conditional distributions between variables. In this example, we investigate in the application of the proposed method in air traffic risk assessment.

4.2.1 Introduction

The worldwide air traffic has seen a continuous increase in the past decades [65]. This puts a heavy burden on the air traffic management (ATM) for maintaining the safety of NAS. While a large portion of the air traffic accident is due to human error, human performance has been always considered as a critical influencing factor for ATM [62]. The Federal Aviation Administration (FAA) and other organizations have been heavily investigating in this research area. Since humans are irreplaceable in the air traffic control (ATC) system due to their ability of quick reactions to unusual scenarios [66]. The NextGen is looking for a computer assistant working along with ATC controllers to monitor and maintain the safety and predict accidents [67]. The information fusion is critical in achieving this goal. This section presents a BEN with encoded human information and studies the effect of such information in the prediction for air traffic risk.

4.2.2 Causal network for air traffic risk

Figure 4-4 showed a network model built to evaluate the risk of an aircraft. Note that this network is only for demonstration purposes and does not represent any real research work. The risk is related to two factors: the speed of the aircraft and the pilot performance. The speed of the aircraft can be affected by weather (e.g. wind speed, rain) and visibility. The weather and visibility are interconnected indicating the potential correlation between the two variables. The pilot performance is a measurement of the pilot status. The experience of the pilot and the rest (sleeping hours) of the pilot prior to the flight are two influencing factors that contribute to the pilot performance.

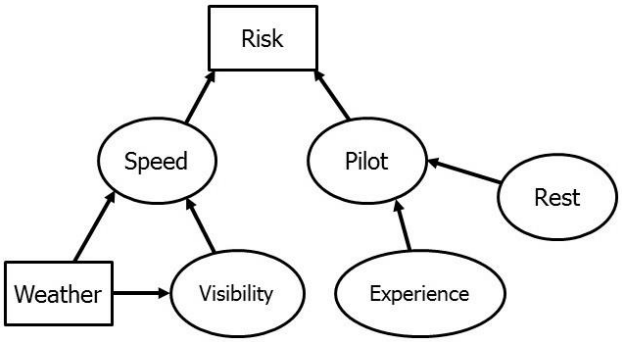


Figure 4-4. The topology for the ATC risk model.

The distribution for each variable in the network model is listed in Table 4-1. Risk and weather are considered as discrete node and the others are modeled as continuous. Risk is a binary node with 0 and 1 correspond to safe and accident, respectively. Weather can take four discrete values, each representing four possible weather conditions, such as sunny, cloudy, rain and snow. The continuous nodes are all modeled as Gaussian nodes. The Pilot can be a reference value for the evaluation of the pilot performance. Experience could be the years of experience of a pilot driving the aircraft. And rest is the sleeping hours of the pilot prior to the departure.

Table 4-1. Parameters and its distribution

Node name	Distribution type	Parameters	
		μ	σ
Risk	Discrete (2 values: 0 1)	[0.9, 0.1]	
Speed	Normal	51	20.5
Pilot	Normal	82	20.5
Weather	Discrete (4 values: 0 1 2 3)	[0.3, 0.3, 0.2, 0.2]	
Visibility	Normal	10	1
Experience	Normal	30	25
Rest	Normal	7	1

We assumed three scenarios to update for the risk probability:

1. An observation of Rest=6 is made about the pilot. This scenario uses only the Bayesian updating.

2. In addition to the observation of Rest=6, a first order moment (mean) of rest=8 is given. This scenario will use the BEN to incorporate this moment information.

3. This scenario includes the observation of Rest=6, and a new relation between the pilot performance and the rest hours expressed as a known function $\text{Pilot} = f(\text{Rest})$. The known function is specified as $\text{Pilot} = f(\text{Rest}) = 10 \cdot \text{Rest}$. BEN will use this information to change the likelihood between the two variables.

4.2.3 Bayesian Entropy network for the information fusion

The three pieces of information will be fused into the network model via the BEN method and update for the risk probability (Figure 4-5).

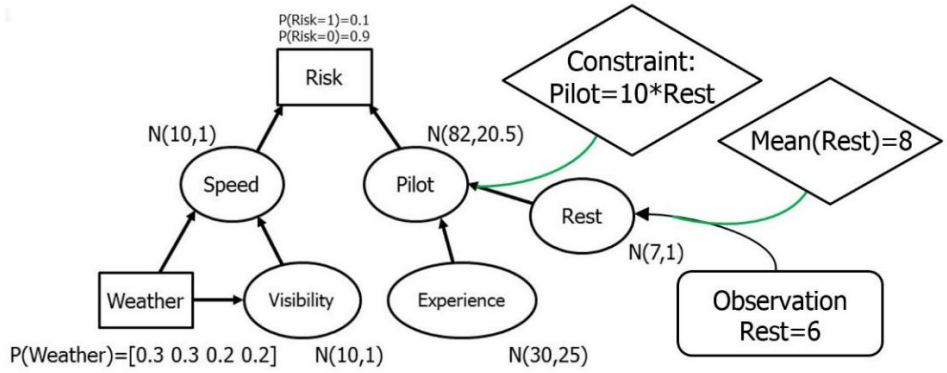


Figure 4-5. The topology of the ATC model with the information in three scenarios.

The updated result for the marginal distribution of Rest can be seen in Figure 4-6. It can be seen that when updating with only point observation (first scenario), the posterior

distribution is shifted towards the observed value and variance decreased. While the posterior from BEN has a similar shape but the mean value was shifted to the value specified by the mean constraint.

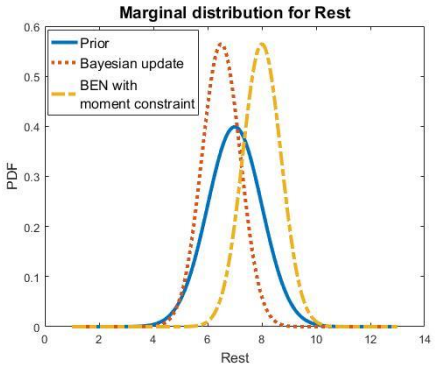


Figure 4-6. The posterior for Rest in the first two scenarios

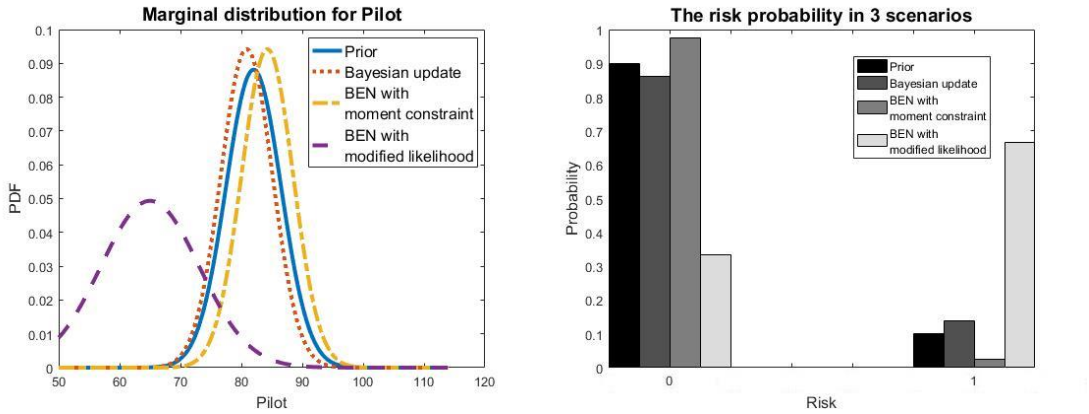
The update will propagate in the network along the edges. In the third scenario, a new correlation between Pilot node and Rest node is introduced as $\text{Pilot} = f(\text{Rest}) = 10 \cdot \text{Rest}$. Since it is a constraint imposed on the likelihood function, it is written as:

$$\int p(\text{Pilot} \mid \text{Rest}) \text{Rest} d\text{Pilot} = 10 \cdot \text{Rest} \tag{3.1}$$

The solution given this constraint is given as:

$$p(\text{Pilot}|\text{Rest}) \propto q(\text{Pilot}|\text{Rest}) \exp\left(\frac{10 \cdot \text{Rest} - \mu}{\sigma^2} \text{Pilot}\right) \tag{3.2}$$

where q is the old likelihood function and μ and σ are the distribution parameters (mean and variance) for the old likelihood function. Eq. (3.2) is used for updating in the third scenario. The results for the updated marginal distribution for Pilot and Risk can be seen in



a) b)

Figure 4-7. The marginal distribution for a) Pilot and b) Risk

To interpret the result, we can think of the observation information comes from a recording device that tracks the pilot's sleeping time. The moment constraint can be understood as the tracking device may be malfunctioned and we tend to believe that the pilot has followed his regular schedule for 8 hours of sleep. The information in the third scenario can be a new research finding of the correlation between pilot performance and pilot sleeping hours. From the result, we can see that: for the first scenario, the risk probability increased due to the observed low resting hours. For the second scenario the risk probability decreased since we tend to believe that the pilot had enough rest. The risk probability has a sudden increase in the third scenario because the constraint introduced a positive correlation. This acted as a penalty for the observed low sleeping time. Hence the risk probability increased.

4.2.4 Conclusion for BEN in ATM risk control

This example illustrated the ability of BEN to incorporate various sources of information into a network model to update for risk probability. According to the result, BEN can take advantage of the extra information to modify the probability distribution.

4.3 Aircraft Trajectory Prediction and Risk Assessment Using Bayesian Updating

Flight trajectory prediction is crucial in maintaining the safety and predicting accidents in the National Airspace System (NAS). The reported work used Bayesian updating to achieve flight trajectory prediction and real-time risk assessment in the NAS. The trajectory simulation is done using NATS, a novel flights simulation platform. The model can consider multiple sources of uncertainties such as weather, human performance etc. Through Bayesian updating, the uncertainty in the model can be reduced given observable quantities. In this article, the Bayesian framework in updating model parameter through observation is introduced. The NATS simulation for a real accident scenario at SFO airport will be presented. In the presented framework, the risk probability is updated continuously using the aircraft location tracking information. The accident can be predicted well before it happens. A criterion for assessing the risk probability is developed under the NATS platform. The risk probability is evaluated based on the separation between aircrafts. It can work as a computer-aided algorithm for Air Traffic Management (ATM) aiming to help the ATC operator in preventing potential accidents.

4.3.1 Introduction

Flight trajectory prediction has been recognized as an important factor in assessing the safety for each individual flight as well as the entire NAS. Researches has been done

in related fields including flight trajectory prediction, optimization and planning. [68] used a node based-pathfinding method to design flight plan to avoid possible turbulence. [69] presented a novel trajectory prediction algorithm based on the aircraft's intent. [70] used a neural network model to predict delay time due to ATC. There are other works, such as [71] and [72], that used probabilistic approach to consider the uncertainties in the prediction from various sources, such as weather [73] and aircraft operation parameters [74] etc. The statistical method is the best in modeling the unpredictable in complex systems such as NAS. And it is critical in understanding these uncertainties and how they can affect the NAS [75].

A physical based model is needed to simulate and predict the trajectory of the aircraft. The dynamics of an aircraft involves 6 degrees-of-freedom (DOF). Some commercial software such as X-plane used 6 DOF model to give detailed simulation for the aircraft dynamics. Since these models are complicated to be applied in trajectory analysis, they are often used for pilot trainings. Considering the aircraft as a point mass and with 3 DOF can significantly reduce the calculation complexity and require less aircraft data. This brings done the complex problem to 6 ordinary differential equations (ODEs). Some existing software such as Blue Sky uses the 3-DOF model for multiple aircraft simulation. A kinematic model replaced 3 of the ODEs related to the aircraft dynamics with table look-up from existing data base such as BADA [76]. This model can further increase the computational efficiency and is presented in a novel trajectory computational software called National Airspace Traffic-Prediction System (NATS) [77]. The NATS has detailed model of the airport infrastructures, flight plan and simple human performance models. It

has three subsystems: an equipment system for modeling aircrafts ground vehicles and communication systems, an entities system for human operation models such as pilots and ATC controllers, and an environment system to model the infrastructures at the airport, terrain and weather. NATS is a software designed for traffic prediction and prognosis.

In this chapter, we will use the NATS software for flight trajectory calculation and prediction. Uncertainties from various sources will be considered using probabilistic approach. As the modernization of the NAS, the tracking systems for aircrafts has shifted from radar based to satellite tracking systems such as ADS-B [78]. This enables the real-time location tracking with high precision for individual aircraft. Based on the broadcasted location, the uncertainties in the system will be updated using Bayesian-Entropy network (BEN) updating [79]. BEN is a tool that combines the classical Bayesian method and the Maximum Entropy (ME) method. It contains a Bayesian part that can update using point observations, and an Entropy part that can encode extra constraint information. Similar to Bayesian updating, it is a statistical tool to change the prior probability distribution based on observed evidence. The updated model will give a more accurate prediction of abnormality and can aid ATC controllers in preventing accident well before it happens. In this paper, a scenario from an accident at SFO will be simulated. In this accident, the pilot has mistaken the taxiway as the runway and intended to land while four other airplanes are on the taxiway. Both the ATC and pilot did not recognize the error. It was a plane on the taxiway reported that the landing aircraft is lining up with the taxiway that avoided the potentially fatal accident. The paper is organized as follow: the following section introduces the BEN method and shows how the real-time tracking can be used by Bayesian

updating for predicting the flight trajectory. The computational time is still high due to the numbers of model evaluation needed for updating. The fourth section discusses the real-time implementation of the proposed method by replacing the NATS software with a surrogate model. The fifth section further enhances the method by using Bayesian model selection and entropy constraint to predict the landing point. Conclusion and future research goal will be stated in the last section.

4.3.2 Simulation and Bayesian updating for the landing point

The scenario presented in the demonstration example comes from a real accident case happened on the evening of July 7th, 2017. Air Canada Flight 759 (AC759) was cleared to land on the runway 28R at SFO airport. The layout of the SFO airport can be seen in Figure 4-8. At that night, the runway 28L was closed down for maintenance and the lights on the runway were off. The AC759 pilot mistook the 28R as 28L and lined-up with the adjacent way which is taxiway C. Both the pilot and ATC are not aware of the wrong line-up. There were four aircraft on taxiway C waiting for take-off. Fortunately, the one of the four planes on the taxiway interrupted the radio and AC759 aborted the landing. The AC759 came dangerously close to the flights on the taxiway. The incident could have ended in the greatest aviation disaster in history [80].

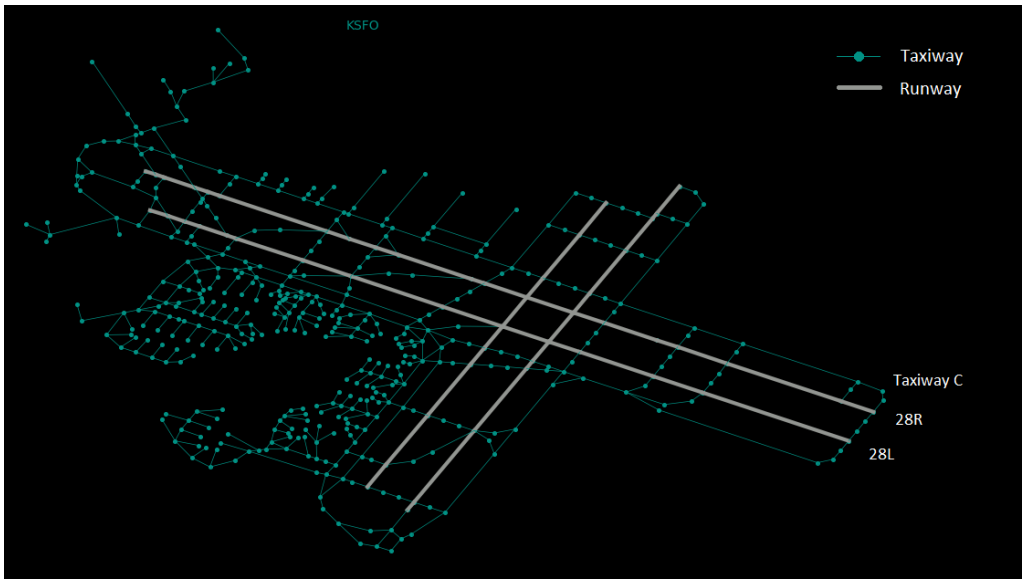


Figure 4-8. Runway and taxiway layout for SFO international airport at San Francisco, CA

The flight track data could be found in the Sherlock data warehouse. In Figure 4-9 is a zoomed in view near the airport in San Francisco. As it can be seen, the flight made a first attempt for landing but aborted and went around and did a second landing. The goal of this research is to be able to predict the landing point with the continuous observation of the aircraft location from ADS-B data.

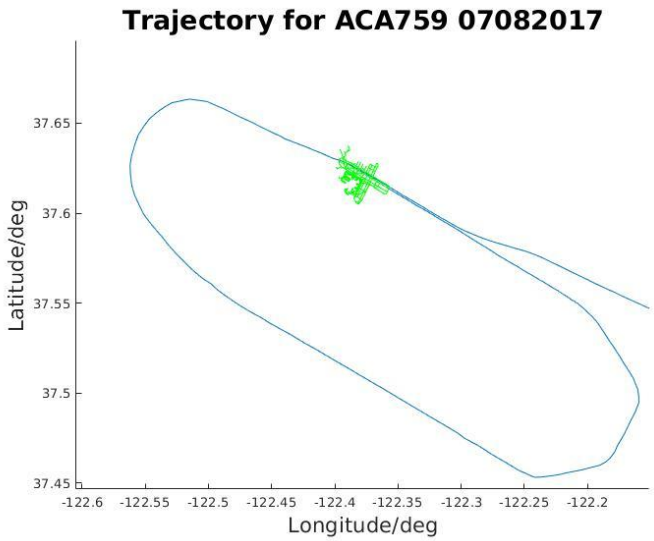


Figure 4-9. The trajectory data from Sherlock data warehouse for ACA759 plotted near the SFO airport.

To simulate this case, we used NATS to simulate the landing procedure at the SFO airport. The coordinates of the last waypoint is set up to the end at the taxiway C. Figure 4-10 showed the comparison of the simulated normal and faulty trajectory along with the real data. The selected 11 points, which is in a time duration of roughly one minute, are the track points prior to the time when the potential hazard was reported by the other pilot on taxiway. The example will show that these observations about the aircraft location can be used to update the uncertainty in the landing point and issue an early warning about the potential hazard.

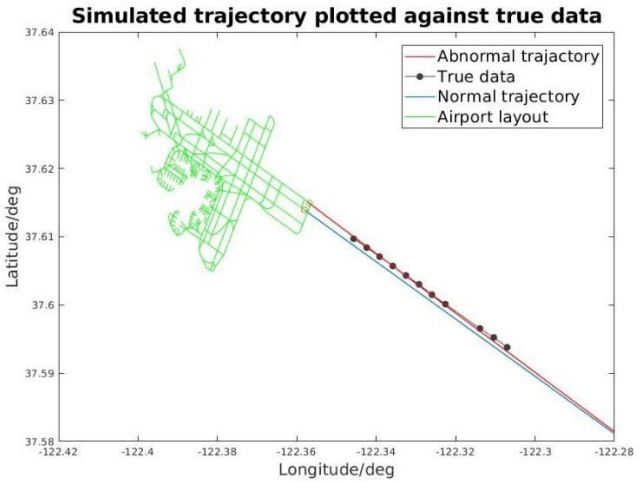


Figure 4-10. Trajectory simulation plot for normal and faulty conditions plotted against real data

In almost all cases, the aircraft would be landing on the runway and in rare cases on the taxiway. The landing point coordinate can be modeled as a mixture of two bivariate normal distribution:

$$\omega N(\mathbf{rwy}, \mathbf{\Sigma}) + (1-\omega) N(\mathbf{txy}, \mathbf{\Sigma}) \tag{6}$$

where N denotes a bivariate normal distribution. rwy and txy denotes the longitude and latitude coordinates for the end of runway 28R and taxiway. Σ is the covariance matrix. ω is a weighing parameter and is modeled as a Beta distribution. As for the prior, ω follows a Beta(9,1) distribution. Which indicates that most of the weight would be on the runway. The last waypoint of the flight plan will be modeled using the distribution in Eq. 6 and updated using the 11 observed track points in Figure 4-10.

The constraint in this case is the means of the two bivariate normal distributions, since the plan would either land on the runway or taxiway. This piece of information is enforced using the BEN method. Figure 4-11 showed the Monte-Carlo (MC) samples

drawn from the prior distribution. Our goal is to use the tracking position of the aircraft to update the uncertainty in the landing position.

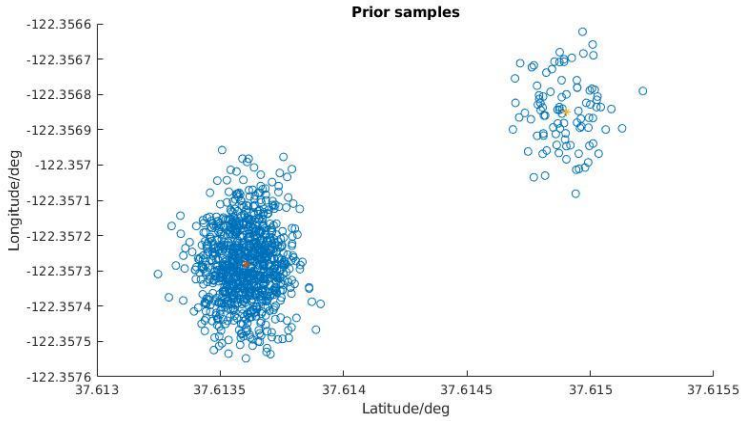


Figure 4-11. The bimodal distribution for the landing point.

4.3.3 Surrogate model for real-time model evaluation

In order to achieve the real-time or near real-time prediction, a surrogate model is proposed to replace the NATS software. The goal of the surrogate is to estimate as accurate as the NATS model with a minimum computational time. For simplification, the surrogate model was only trained for the landing procedure. Neural network fitting is used as the surrogate. The model has three input, the coordinates of the last waypoint (latitude and longitude) and the time t. The output of the surrogate is the location (latitude and longitude) of the aircraft at time t. Figure 4-12 showed the simple network structure.

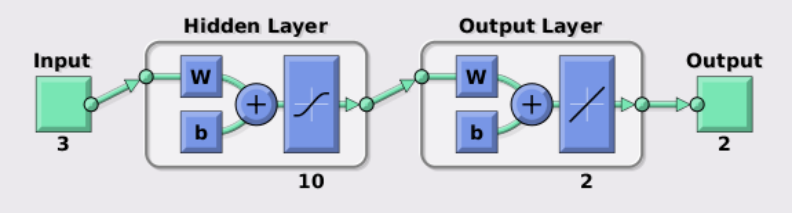


Figure 4-12. The Neural Network structure for the surrogate model

It is a two-layer feed-forward neural network with 10 hidden neurons. It used Levenberg-Marquardt (LM) algorithm as the optimizer. A dataset is needed to train the surrogate. The dataset is generated from several Monte Carlo (MC) simulation of NATS with the last waypoint as random variables. 10000 simulated trajectory point were selected as the training data, of which 70% were used to train, 15% as the testing data and 15% as the validation. The training is achieved using the machine learning toolbox in MATLAB built-in functions. The training loss and the error rate plot is shown in Figure 4-13.

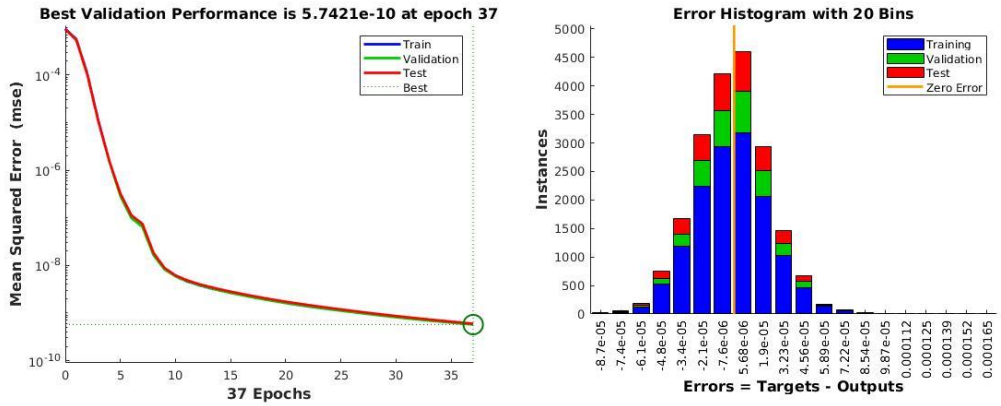


Figure 4-13. The training loss (left) and error histogram (right) for the NN training

It can be seen that the NATS model for the landing part is pretty linear and Neural Network fitting can capture the trend of the NATS function with minimum error. The output is used as a surrogate replacing the NATS calculation in the updating process. The update can be achieved in real-time. The next section will explore how to encode entropy constraint on the runway and taxiway coordinates.

4.3.4 Bayesian model selection and Entropy constraint

This section presents Bayesian model selection to handle the landing at runway and taxiway as different models. The prior distribution in section III is bimodal, it is hard to

put entropy constraint on such a probability function. Instead, we regard each point as a different model. This way the mixture distribution was de-coupled and entropy constraints can be added onto different models separately.

The Bayesian model selection is originally used to consider the uncertainty in model choice [81]. Different models may be available in describing the mechanism of an engineering problem. But each model may be applicable in different scenarios. The Bayesian model selection can update the model parameters and probability of choosing each model at the same time. Denote different models as Mk, the associated parameters in the model θk. Similar to the Bayesian updating and Bayesian Entropy method, the equation for updating the model parameters given observation data x' and constraint function C can be expressed as:

$$p(\theta_k \mid x', M_k) = p(x' \mid \theta_k, M_k)\, p(\theta_k \mid M_k) \exp(C(\theta_k, M_k)) \tag{7}$$

where p(θk|x',Mk) is the posterior of the parameters in model Mk. P(x'| θk,Mk) is the likelihood of the observation given the model Mk and its parameters θk. P(θk|Mk) C(θk,Mk) is the constraint function associated with θk in model Mk.

Our goal is to update the posterior probability for each model given available observations and constraint, P(Mk|x'). According to Bayesian theorem, we can have:

$$P(M_k \mid x') \propto p(x' \mid M_k) P(M_k) \tag{8}$$

The model posterior based on observation is the product of the model likelihood and the model prior. The evaluation of the model likelihood involves an integral:

$$p(x'|M_k) \propto \int_{\Theta_k} p(x', \theta_k | M_k) d\theta_k = \int_{\Theta_k} \frac{p(x'|\theta_k, M_k) p(\theta_k | M_k) P(M_k)}{P(M_k)} d\theta_k \tag{9}$$

The integral is over the domain of the parameter θk. Combining the above two equations, we can have the final form for the model posterior calculation:

$$P(M_k | x') \propto P(M_k) \int_{\Theta_k} p(\theta_k | M_k) p(x'|\theta_k, M_k) d\theta_k \tag{10}$$

A hierarchical structure is illustrated in Figure 4-14. The framework has a top level and a bottom level. The bottom level concerns the parameters in each model and the related likelihood function. The top level concerns the model probability. With the observation data and corresponding constraint for parameters in the models, the model probability and the parameter distributions can be updated according to Eq.7 and Eq. 10.

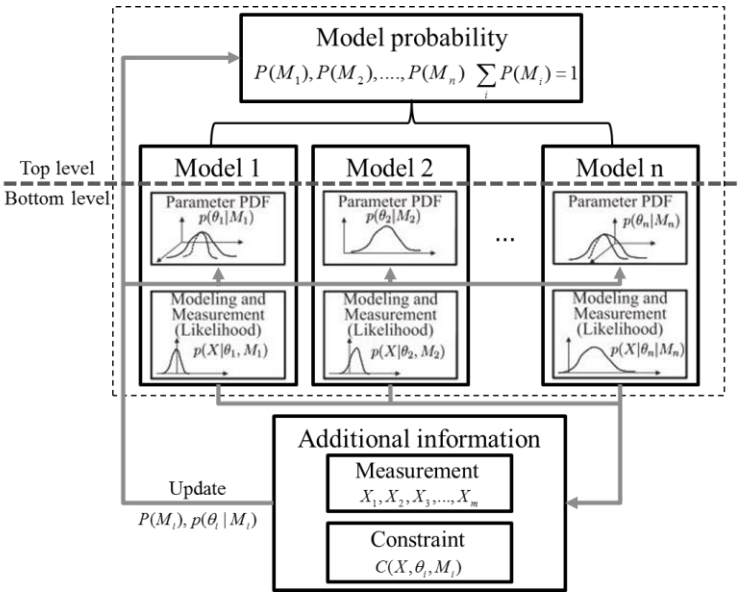


Figure 4-14. The hierarchical structure for Bayesian model selection

When applied in the prediction for the landing point, two models can be considered. The parameters for each model are the coordinates of the last waypoint. Prior distributions are set as bivariate normal distributions with means at the coordinates of the end of runway and taxiway. Now the constraint can be easily applied. The Entropy constraint enforces the mean value does not change in each update. The physical meaning of such constraints is that the coordinates of the runway and taxiway would not change since the pilot would only land either on the runway or on the taxiway. For most of the time, the pilot would have no problem in landing on the runway. Only in rare cases, the pilot would mistakenly land on the taxiway[82][83]. Conservatively, the model prior is set as P(runway)=0.9 and P(taxiway)=0.1.

Since the parameter distributions are enforced with a mean constraint, the model probability is the main concern in this problem. In Figure 4-15 showed the posterior for the model probability after each update with the sequential observation point. The horizontal axis indicates the model with 1 represents landing on runway and 2 represents landing on taxiway. Similar to the result using particle filter, the probability for landing on the runway decreased in only a few updates. This enables the controller to issue an early warning about the potentially hazardous accident.

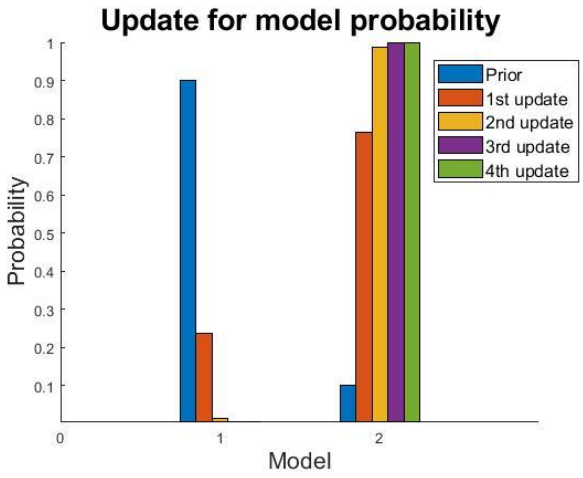


Figure 4-15. The update for model probability using faulty trajectory data

To further validate the model, a few observation points from the second landing attempt, which is a normal landing, are chosen as observations to update the model probability. The results are plotted in Figure 4-16. According to the result, the model probability for landing on the runway is increasing. This is because the aircraft is staying on the right track.

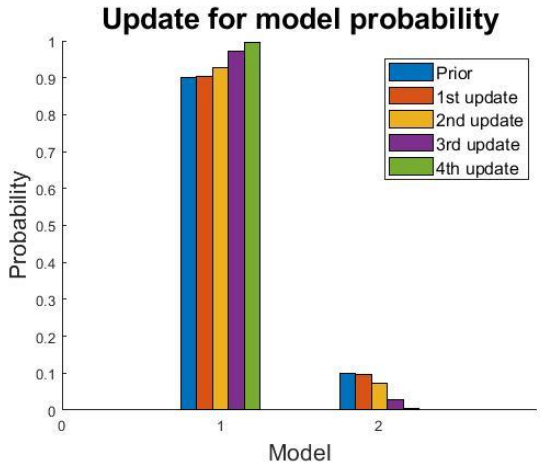


Figure 4-16. The update for model probability with normal trajectory data

4.3.5 Conclusion for BEN in trajectory prediction

The above research presented the simulation using NATS, a novel software platform for flight trajectory simulation. The software has a simplified kinematic model for the calculation of flight trajectory and included detailed model for the airport infrastructure, flight plan and simple human performance models. A real accident scenario is simulated in this case. The accident can be regarded as a fault caused by the uncertainties in the modeling parameter (the last waypoint). Based on the simulated real accident case, it can be found that through the observation of the aircraft location monitoring, the risk probability can be predicted using Bayesian updating. A surrogate model is used to replace NATS in the trajectory evaluation. The surrogate model is achieved via Neural Network fitting from MATLAB machine learning toolbox. After training with simulated trajectory data, it can capture the characteristics of the NATS software very well but requires much less computational time. This enable the real-time calculation for the posterior probability. The hierarchical Bayesian model selection framework is used to handle the bimodal behavior in the parameter distributions. The method treats different landing point as separate models, so that entropy constraint can be encoded on each parameter independently. With the distribution of parameters being constrained, the model probability is used as the criterion to predict the landing point. Based on this study, several conclusions can be drawn.

1. The Bayesian updating can be used to predict the aircraft landing location based on observed flight track points. The proposed framework can be used as a computer aid for ATC controllers to issue an early warning for unforeseeable accident.

2. A surrogate model can be used to replace NATS software in the landing process simulation. The surrogate model requires much less computational time than the NATS software evaluation. The fast model evaluation enables the real-time updating for the posterior distribution.
3. The Bayesian model selection framework decoupled the bimodal distribution in the model parameters and regarded each mode as an independent model. Hence, entropy constraint can be added into the model separately. This way the physical constraint of the runway and taxiway can be encoded into the model.

4.4 Conclusion

This chapter applied the BEN method in ATM to predict air traffic risk. The proposed method is able to encode various types of information into a classical Bayesian network. It provided an approach for the overall information fusion in NAS system. In the first example, the BEN utilized the tendency data from the ASRS report to help the identification for the cause of runway incursion. This demonstrated the ability of BEN in incorporate human empirical knowledge into classification tasks. The second example demonstrated the BEN in inferencing the risk in air traffic. Through three scenarios with different available information, the BEN achieved information fusion from: observation data, human information and new correlations. The third example presents the Bayesian model selection with entropy constraint controlling the parameters in the distribution. The method is applied in aircraft trajectory prediction. The aircraft operation has a lot of rules and physical constraints. The proposed method can restrict those parameters in the simulations. Overall, the BEN is a unique and important approach in achieving the

information fusion in NAS system and provide a computer aided framework for the safety management and risk prediction in the NextGen.

5 Bayesian-Entropy method for surrogate modeling

Surrogate modeling, also known as metamodels, is an important task in many engineering applications such as reliability analysis and design optimization. For such tasks, a good surrogate model can replace the computationally expensive physics model to accelerate necessary model evaluation. A surrogate model is often trained as a regression for the available data. But in addition to the point data, other types of information are available. Such as the underlying physics model, empirical knowledge and physical constraints etc. This chapter explore the Bayesian-Entropy method in incorporating various types of information in popular regression method for a more accurate surrogate modeling.

5.1 Bayesian-Entropy linear regression (BELR)

This section presents the application of BE method into the classical Bayesian linear regression. Bayesian linear regression is a statistical approach for fitting function with data. The method assumes a prior distribution for the model parameters and calculates the posterior based on the likelihood of the given data. Traditional Bayesian method only deals with point data, it is not easy to incorporate other types of information. This section reviews the classical Bayesian linear regression and then introduces the application of Bayesian-Entropy method to encode values and derivative information as constraints in the regression model.

5.1.1 Introduction

The classical Bayesian linear regression is a statistical data analysis tool for estimating the relationship between a set of input variable and output response. The method has found many applications in engineering fields. The main drawback, as has pointed out

in [84], is that the regression model is restricted by the selection of input variables and s selected form of basis. This has brought huge uncertainties related to model selection, as the fitted function will have poor performance when the selected model deviates from the actual physics. There have been many methods in the area of reducing model uncertainties, such as model averaging, feature selection etc. Despite, this section proposes the Bayesian-Entropy linear regression (BELR) where extra information, such as the value and derivative information of the target regression model, are introduced into classical Bayesian regression in the form of constraints.

The classical Bayesian regression considers a linear form of function given as:

$$y = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x} + \varepsilon \tag{5.1}$$

where $\mathbf{x}$ is the basis defined on the input parameters, $\boldsymbol{\beta}$ is the regression coefficient and ε is an error term that follows a specific distribution. With the assumption of prior distributions for the regression parameters, the posterior distribution can be calculated according to the Bayes' theorem:

$$p(\boldsymbol{\beta}, \sigma^2 \mid \mathbf{X}, \boldsymbol{y}) \propto p(\boldsymbol{y} \mid \mathbf{X}, \boldsymbol{\beta}, \sigma^2)\, p(\boldsymbol{\beta} \mid \sigma^2)\, p(\sigma^2) \tag{5.2}$$

If random distribution functions are assigned to the parameters, the solution may be hard for analytical derivation. To maintain conjugacy, that is the posterior is in the same type as the prior distribution, usually the error term ε is assumed to follow a Normal distribution with 0 mean and σ^2 variance, so that the likelihood function given data $\mathbf{X}$ and $\boldsymbol{y}$ can be written as:

$$p(\boldsymbol{y}\,|\,\mathbf{X},\boldsymbol{\beta},\sigma^2) \propto \sigma^{-n}\exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta})\right) \tag{5.3}$$

where n is the number of available data. The prior distribution for the conditional probability of the regression coefficient is Normal distribution as $N(\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0)$:

$$p(\boldsymbol{\beta}\,|\,\sigma^2) \propto (\sigma^2)^{-k/2}\exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)\right) \tag{5.4}$$

where k is the number of elements in $\boldsymbol{\beta}$. The prior for the noise variance σ^2 is given an Inverse-gamma distribution with Inv-Gamma(a_0,b_0):

$$p(\sigma^2) \propto (\sigma^2)^{-a_0-1}\exp\left(-\frac{b_0}{\sigma^2}\right) \tag{5.5}$$

Now, the posterior for the parameters of the linear model can be written as:

$$\begin{aligned} p(\boldsymbol{\beta},\sigma^2\,|\,\mathbf{X},\boldsymbol{y}) &\propto p(\boldsymbol{y}\,|\,\mathbf{X},\boldsymbol{\beta},\sigma^2)\,p(\boldsymbol{\beta}\,|\,\sigma^2)\,p(\sigma^2) \\ &\propto \sigma^{-n}\exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\mathbf{X}\boldsymbol{\beta})\right) \\ &\cdot(\sigma^2)^{-k/2}\exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)\right)\cdot(\sigma^2)^{-a_0-1}\exp\left(-\frac{b_0}{\sigma^2}\right) \end{aligned} \tag{5.6}$$

With some rearrangements, the equation could be rewritten as the product of a Normal and an Inverse-gamma distribution as:

$$\begin{aligned} p(\boldsymbol{\beta},\sigma^2\,|\,\mathbf{X},\boldsymbol{y}) &\propto p(\boldsymbol{\beta}\,|\,\mathbf{X},\boldsymbol{y},\sigma^2)\,p(\sigma^2\,|\,\mathbf{X},\boldsymbol{y}) \\ &\propto (\sigma^2)^{-k/2}\exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{\beta}-\boldsymbol{\mu}_n)^T\boldsymbol{\Sigma}_n^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_n)\right)\cdot(\sigma^2)^{-a_n-1}\exp\left(-\frac{b_n}{\sigma^2}\right) \end{aligned} \tag{5.7}$$

where $\boldsymbol{\mu}_n$ and $\boldsymbol{\Sigma}_n$, a_n and b_n are the posterior distribution parameters for $\boldsymbol{\beta} \sim N(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$ and $\sigma^2 \sim$ Inv-Gamma(a_n,b_n). The distribution parameters can be calculated according to:

$$\begin{aligned}
\boldsymbol{\mu}_n &= \boldsymbol{\Sigma}_n(\mathbf{X}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0) \\
\boldsymbol{\Sigma}_n &= (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \boldsymbol{\Sigma}_0^{-1})^{-1} \\
a_n &= a_0 + n/2 \\
b_n &= b_0 + (\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{\mu}_0^{\mathrm{T}}\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0 - \boldsymbol{\mu}_n^{\mathrm{T}}\boldsymbol{\Sigma}_n^{-1}\boldsymbol{\mu}_n)/2
\end{aligned} \tag{5.8}$$

The detailed derivations can be found in [85] and are not given in here. The posterior mean for $\boldsymbol{\beta}$ is often used as the predictor as the regression coefficient for the linear model.

5.1.2 Introducing value and derivative information as constraints

In the BELR framework, in addition to the point data as provided in $\mathbf{X}$ and $\boldsymbol{y}$, possible extra information such as value and/or derivative information are taken into account. An experienced engineer may have empirical knowledge about the expected value of the system response at a given point. This type of information is regarded as an unbiased estimate for the mean prediction of the given regression model in the BELR framework. That means, it is regarded as value constraint for the expected system response at the given input location $\mathbf{x}_0$ equals y_0:

$$E(\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_0 + \varepsilon) = E(\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_0) = y_0 \tag{5.9}$$

Since ε is a zero mean error term, its expected value yields zero. Since the coefficient $\boldsymbol{\beta}$ is assigned a Normal distribution, equation can be written as an integration form over the domain of $\boldsymbol{\beta}$ as:

$$\int_{\mathbf{B}} p(\boldsymbol{\beta})\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_0 d\boldsymbol{\beta} = y_0 \tag{5.10}$$

Sometimes, a known physics model may reveal the derivative information about the target regression model. This type of information can be regarded as the expected value of the derivative. Similarly, the derivative information in the *i*th dimension at location $\mathbf{x}_0$ can be expressed as:

$$\int_{\mathbf{B}} p(\boldsymbol{\beta})\boldsymbol{\beta}^{\mathrm{T}}\left(\frac{\partial \mathbf{x}}{\partial x_d}\bigg|_{\mathbf{x}=\mathbf{x}_0}\right) d\boldsymbol{\beta} = dy_{0,d} \tag{5.11}$$

where $\partial\mathbf{x}/\partial \mathrm{x}_d$ is the partial derivative of the basis function in the *d*th dimension and $dy_{0,d}$ is the corresponding derivative value. Following the framework of the BE method, we are finding the target distribution $p_{en}(\boldsymbol{\beta})$ for the regression coefficient $\boldsymbol{\beta}$ that maximizes the Entropy with respect to the posterior from Bayesian regression $p(\boldsymbol{\beta})$:

$$S[p_{en}, p] = -\int p_{en}(\boldsymbol{\beta}) \log \frac{p_{en}(\boldsymbol{\beta})}{p(\boldsymbol{\beta})} d\boldsymbol{\beta} \tag{5.12}$$

Under the constraints defined in Eqs. (5.10) and (5.11), as well as the normalization for $p_{en}(\boldsymbol{\beta})$. Following the same logic in Chapter 2, a Lagrange equation is formed:

$$\mathcal{L} = S + \alpha\left[\int_{\mathbf{B}} p_{en}(\boldsymbol{\beta}) d\boldsymbol{\beta} - 1\right] + \sum_{i=1}^{M} \eta_i \left[\int_{\mathbf{B}} p_{en}(\boldsymbol{\beta})\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i d\boldsymbol{\beta} - y_i\right] + \sum_{i=M+1}^{M+N} \eta_i \left[\int_{\mathbf{B}} p_{en}(\boldsymbol{\beta})\boldsymbol{\beta}^{\mathrm{T}}\left(\frac{\partial \mathbf{x}}{\partial x_{d_i}}\bigg|_{\mathbf{x}=\mathbf{x}_i}\right) d\boldsymbol{\beta} - dy_{i,d}\right] \tag{5.13}$$

In Eq. (5.13), α and η_i's are the Lagrangian multipliers associated with each constraint. *M* and *N* are the number of available value and derivative constraints, respectively. Similar to

the derivation in the Chapter 2, the deviation of the Lagrange function equals zero, $\delta \mathcal{L} = 0$, give us the solution to the target coefficient distribution function as:

$$p_{en}(\boldsymbol{\beta}) \propto p(\boldsymbol{\beta}) \exp[\sum_{i=1}^{M} \eta_i \boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_i] \exp[\sum_{i=M+1}^{M+N} \eta_i \boldsymbol{\beta}^{\mathrm{T}} (\frac{\partial \mathbf{x}}{\partial x_{d_i}}\Big|_{\mathbf{x}=\mathbf{x}_i})] \tag{5.14}$$

The next step is to solve the Lagrangian multipliers by back substituting Eq. (5.14) into each of the available constraints in Eqs (5.10) and (5.11). Eqs (5.10) and (5.11) can be regarded as a system of equations for η_i's regarding the expected value for $\boldsymbol{\beta}$. In the Bayesian regression framework, the posterior for $\boldsymbol{\beta}$ follows a Normal distribution $N(\boldsymbol{\mu}_{\mathrm{n}}, \boldsymbol{\Sigma}_{\mathrm{n}})$. By substitute the pdf of $p(\boldsymbol{\beta})$ into Eq. (5.14) we have:

$$p_{en}(\boldsymbol{\beta}) \propto \exp\left[-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_n)^{\mathrm{T}} \boldsymbol{\Sigma}_n^{-1} (\boldsymbol{\beta}-\boldsymbol{\mu}_n)\right] \exp\left[\sum_{i=1}^{M} \eta_i \boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_i\right] \exp\left[\sum_{i=M+1}^{M+N} \eta_i \boldsymbol{\beta}^{\mathrm{T}} (\frac{\partial \mathbf{x}}{\partial x_{d_i}}\Big|_{\mathbf{x}=\mathbf{x}_i})\right] \tag{5.15}$$

With some rearrangement, the kernel of the pdf can be written in the form of a new Normal distribution as $N(\boldsymbol{\mu}_{\mathrm{en}}, \boldsymbol{\Sigma}_{\mathrm{n}})$, where

$$\boldsymbol{\mu}_{en}{}^{\mathrm{T}} = \boldsymbol{\mu}_n{}^{\mathrm{T}} + \left(\sum_{i=1}^{M} \eta_i \mathbf{x}_i{}^{\mathrm{T}} + \sum_{i=M+1}^{M+N} \eta_i (\frac{\partial \mathbf{x}}{\partial x_{d_i}}\Big|_{\mathbf{x}=\mathbf{x}_i})^{\mathrm{T}}\right) \boldsymbol{\Sigma}_n \tag{5.16}$$

With the expression for $\boldsymbol{\mu}_{\mathrm{en}}$ in Eq. (5.16), the η_i's can be expressed as:

$$\boldsymbol{\eta} = \left(\mathbf{U}\boldsymbol{\Sigma}_n \mathbf{V}\right)^{-1} \left(\mathbf{B} - \mathbf{Y}\right)$$

with
$$\mathbf{U}=\left[\mathbf{x}_1{}^{\mathrm{T}},\ \cdots,\ \mathbf{x}_M{}^{\mathrm{T}},\ \left(\frac{\partial \mathbf{x}}{\partial x_{d_{M+1}}}\bigg|_{\mathbf{x}=\mathbf{x}_{M+1}}\right)^{\mathrm{T}},\ \cdots,\ \left(\frac{\partial \mathbf{x}}{\partial x_{d_{M+N}}}\bigg|_{\mathbf{x}=\mathbf{x}_{M+N}}\right)^{\mathrm{T}}\right]^{\mathrm{T}}$$

$$\mathbf{V}=\left[\mathbf{x}_1,\ \cdots,\ \mathbf{x}_M,\ \left(\frac{\partial \mathbf{x}}{\partial x_{d_{M+1}}}\bigg|_{\mathbf{x}=\mathbf{x}_{M+1}}\right),\ \cdots,\ \left(\frac{\partial \mathbf{x}}{\partial x_{d_{M+N}}}\bigg|_{\mathbf{x}=\mathbf{x}_{M+N}}\right)\right]$$

$$\boldsymbol{\eta}=\left[\eta_1,\ \cdots,\ \eta_{M+N}\right]^{\mathrm{T}}$$

$$\mathbf{B}=\left[\boldsymbol{\mu}_n{}^{\mathrm{T}}\mathbf{x}_1,\ \cdots,\ \boldsymbol{\mu}_n{}^{\mathrm{T}}\mathbf{x}_M,\ \boldsymbol{\mu}_n{}^{\mathrm{T}}\left(\frac{\partial \mathbf{x}}{\partial x_{d_i}}\bigg|_{\mathbf{x}=\mathbf{x}_{M+1}}\right),\ \cdots,\ \boldsymbol{\mu}_n{}^{\mathrm{T}}\left(\frac{\partial \mathbf{x}}{\partial x_{d_i}}\bigg|_{\mathbf{x}=\mathbf{x}_{M+N}}\right)\right]^{\mathrm{T}}$$

$$\mathbf{Y}=\left[y_1,\ \cdots,\ y_M,\ dy_{M+1,d_{M+1}},\ \cdots,\ dy_{M+N,d_{M+N}}\right]^{\mathrm{T}} \tag{5.17}$$

With specified constraints, the vector for Lagrangian multipliers $\boldsymbol{\eta}$ can be analytically solved. Back substituting η_i's into Eq. (5.14) we can have the BE distribution for the regression coefficient $\boldsymbol{\beta}$ with constraints. The mean of the distribution in Eq. (5.16) can act as the predictor for the regression model. Detailed derivation can be found in Appendix B.

5.1.3 Numerical demonstration

This section presents two numerical examples to illustrate the application of the proposed BELR method in incorporating value and derivative constraints.

5.1.3.1 BELR with a value constraint

In this example, a simple one-dimensional linear model is considered:

$$y=ax+b+\varepsilon$$

where x is the input and y is the response, a and b are regression coefficient, ε is a Normally distributed random noise with zero mean and σ=0.3. The true value for the regression coefficient is a=1.5 and b=2. A total of 15 observational data are collected in the range of

$x \in [0,1]$. A value constraint is given as the expected value at x_0=2 is y_0=5. The Bayesian linear regression (BLR) uses the 15 observations to calculate the coefficient according to Eq. (5.8). Based on the result from BLR, the BELR considers the extra constraint using Eq. (5.16) with analytically solved Lagrangian multiplier by Eq. (5.17). The comparison of the mean prediction and the associated confidence bond is plotted in Figure 5-1.

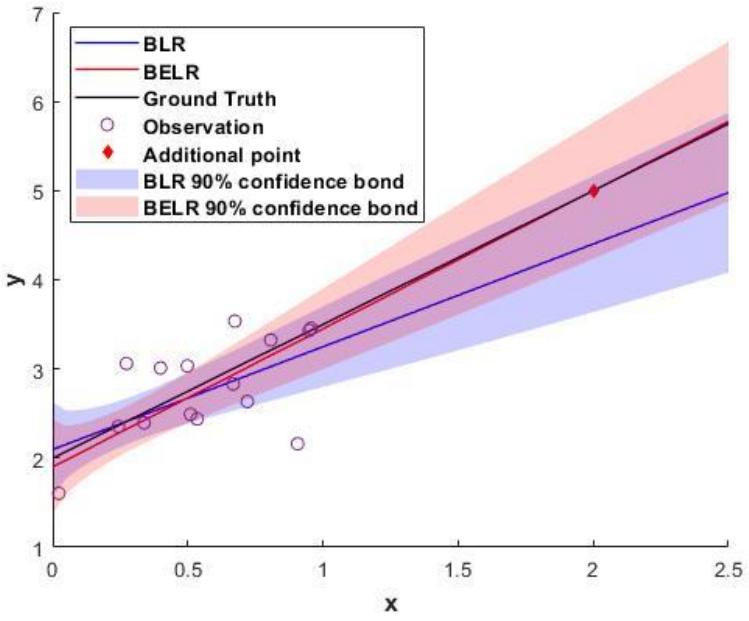


Figure 5-1. Comparison of regression result for BLR and BELR

For the results in Figure 5-1, it is observed that the mean prediction for BELR passes exactly through the additional point due to the constraint. Overall, the BELR has a better resemblance to the true solution thanks to the additional constraints.

5.1.3.2 BELR in smoothly connecting two regression models

Usually, the computational time for the BLR will increase as the order of the basis function, i.e. the number of regression coefficient. Also, it is not always easy to determine

the right order for an arbitrary model with only the observation data. In this example, we are using a lower order linear model to fit a high order function. By partition the data into groups, a lower order BLR model can have a good fit over the data. By adding value and derivative constraints, the BELR can ensure smooth connection between the two separate models.

The observation data are generated from a 6th order polynomial function:

$$y = -0.01(x-2)^5(x-7) + 2 + \varepsilon$$

where x is the input and y is the response, ε is a Normally distributed random noise with zero mean and σ=0.2. Two groups of observations are generated at $x \in [0,1]$ and $x \in [3,5]$, each group contains 20 data instances. The derivative and value at x=2 are assumed to be the extra information and is included as constraints in the BELR framework. The two groups of data are used to fit two separate regression model using a quadratic basis. That is, the basis for the linear model is $\mathbf{x}=[x^2\ x\ 1]$. With a second order function, it is obvious that a single regression model cannot match well with the data. So in this case, two separate models are used to match a local quadratic regression model for each cluster of data. The regression results and the comparison of BLR and BELR are plotted in Figure 5-2.

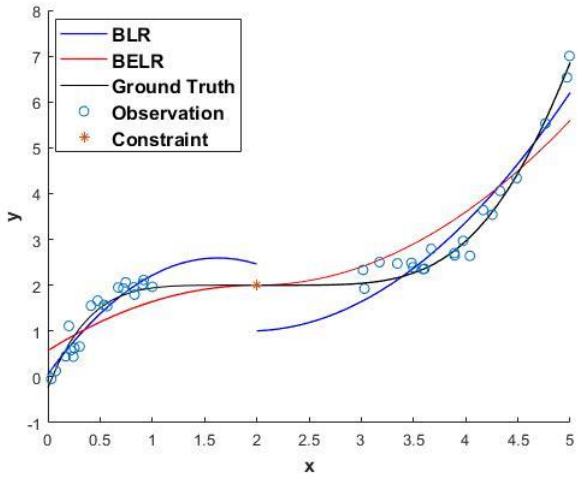


Figure 5-2. Comparison of the regression results for BLR and BELR using two separate models to fit two clusters of data.

The result shows that the BLR can have a good match for the data locally but discontinued globally. While the BELR made a smooth connection between the two local regression model at the specified constraint point. The BELR result has a better resemblance to the trend of the ground truth function. However, the fitness of the BELR to the data is less satisfying than the local models from the BLR method. This is because we are using a lower order basis to fit a higher order function. It is not likely for the regression model to have a perfect match.

5.1.4 Conclusion for BELR

This section presented the application of the Bayesian-Entropy method into Bayesian linear regression to form a Bayesian-Entropy linear regression method that can include value constraints and derivative constraints. The proposed method can utilize other types of information, such as physical or empirical information, other than point observations in building regression models. It has been demonstrated that the regression model from the

proposed method can strictly satisfy the given constraints. It is especially useful when such information is available outside the training data. One important drawback of the proposed method is that the number of constraints cannot exceed the number of regression coefficient. Otherwise it will form an over-determined system for the coefficient solution.

5.2 Bayesian-Entropy Semi-parametric Gaussian Process

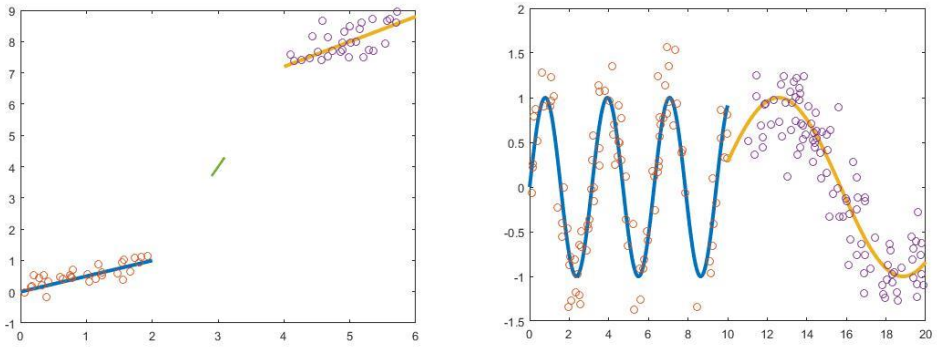
A Gaussian Process makes prediction based on the existing observed data. But in many cases, information is not limited to observations. Extra information, such as physical constraints and empirical knowledge, exists in many engineering problems. This section presents a Bayesian-Entropy method to encode constraints into a Semiparametric Gaussian Process. The Bayesian-Entropy method can encode various types of constraints by adding an additional term to the Bayesian equation. The Bayesian-Entropy regression method can incorporate values and derivative information into the classical Bayesian regression as constraint. By adjusting the mean function in Semiparametric Gaussian Process according to the Bayesian-Entropy regression principle, extra information, such as the expected value and/or the derivative at a specific point, can be encoded into the regression function. Comparing with the traditional method, the constrained Semiparametric Gaussian Process benefits from the available extra information and can make better prediction outside the range of training data.

5.2.1 Introduction

Regression is a common task in high-computational engineering problems such as structural reliability analysis, computational fluid dynamics, and global optimization, where the numerical computational efficient surrogate model is built to fit the black box

function to describe the relationship between system response and input variables. Gaussian Process (GP) [86] or kriging model is a statistical method that has been proven to be versatile for regression tasks in many engineering applications. It is flexible and easy to implement. GP is a statistical process which can be regarded as a multivariate Normal distribution with infinite numbers of dimension, with the continuous function input $x_1 \ldots x_n$ each corresponds to one dimension. The function values $f(x_1) \ldots f(x_n)$ are measured as a Normal distribution. Unlike other data-driven method such as Neural Network (NN) or Support Vector Machine (SVM), the GP does not need a large number of training data. The GP prediction is an interpolation of the training data through the covariance matrix, which can take many forms to describe the spatial correlation between the training data and the prediction point. The GP surrogate modeling has benefited in numerical analysis and optimization. Usually, a GP surrogate is trained through a set of measurement of input and system response. However, sometimes information can be more than just point measurements. Presented in Figure 5-3.a) is a case where there may not be easy to get measurement data in a certain range, but a physical model may be available for providing the derivative information. There has been method that can utilize gradient information about the regression model, such as Gradient-Enhanced Kriging (GEK) [87], first-order kriging and cokriging method. But such models do not impose the gradient as a hard constraint. In another word, the gradient information is not necessarily satisfied with these methods. Figure 5-3.b) showed another scenario where there is a clear distinction for the frequency in the training data. Usually this type of problem is handled by building separate local GPs with different length scale parameter or by applying a GP with nonstationary covariance function [88]. However, the separate GP approach cannot ensure smooth

connection for the two local models. While the nonstationary GP method can adapt to variable smoothness, it requires more numbers of hyper parameters than traditional GP. The computational cost is much more expensive than two or more separate GPs due to the increased number of hyperparameters as well as the greater number of training data compared to training separate GPs.



a) b)

Figure 5-3. Two special cases for building a GP surrogates a) derivative information is known but data is limited, and b) different length scale for two clusters of data.

In this research, we are designated in adding introducing extra information in the form of constraints to solve the above-mentioned issues. By encoding the derivative constraints, the regression model for Figure 5-3.a) can satisfy the derivative information from the physical model where no training data is available. By properly set up a value and/or derivative constraints, the surrogate can have a smooth connection between the separate local GPs. In this section, we propose a Bayesian-Entropy (BE) method to encode constraint into a Semiparametric GP (SGP). The BE method has been used to incorporate different types of constraint into the classical Bayesian framework [89]. In the previous section, it is shown that by using the BE principle, it is possible to insert values and derivative information as constraints into the regression function. This method is applied

in the mean function of SGP. The method will be referred as Bayesian-Entropy Semiparametric Gaussian Process (BESGP). The potential benefit of the proposed method is that: 1) the regression function can utilize the extra information to enable a more accurate prediction based on the constraints. And 2) the proposed method can be used to smoothly connect multiple GPs with different hyperparameters. This way the computational complexity is reduced for each local GP models while maintaining the continuity of the overall regression.

5.2.2 Combining SGP with BELR

This section introduces the basics about SGP and the application of BELR into SGP to encode extra constraints. The SGP is a GP with a specific form of the mean function. The mean function is defined as the product a set of bases and the coefficient. A realization y of an SGP at $\boldsymbol{x}$ could be expressed as:

$$y(\boldsymbol{x}) = \hat{\beta}(\theta)^{\mathrm{T}} \bar{\mathbf{x}} + z(\theta, \boldsymbol{x}) \tag{5.18}$$

where $\hat{\beta}(\theta)$ is the regression parameters of the mean function. $\bar{\mathbf{x}}$ is the basis function of the input $\boldsymbol{x}$. $z(\theta,\boldsymbol{x})$ is a zero-mean GP. The regression parameters and the zero-mean GP are all functions of the hyperparameters θ for the SGP. The hyperparameters can be calculated from training data ($\mathbf{X}$,$\mathbf{y}$) by maximizing the likelihood function of the data. In often cases, a log-likelihood function is defined as:

$$ll(\theta) = -\frac{1}{2}\log|\mathbf{K}| - \frac{1}{2}(\mathbf{y} - \mathbf{X}\hat{\beta}(\theta))^{\mathrm{T}}\mathbf{K}^{-1}(\mathbf{y} - \mathbf{X}\hat{\beta}(\theta)) - \frac{1}{2}\log\left|\mathbf{X}^{\mathrm{T}}\mathbf{K}\mathbf{X}\right| \tag{5.19}$$

where K is the covariance matrix, and is also a function of the hyperparameters θ. The covariance function is used to describe the correlation between each data location according to their spatial distance. One of the common forms of the covariance function is a Squared exponential covariance function whereas a Gaussian kernel is used to describe the spatial correlation:

$$K(x, x'; \theta) = \sigma^2 \exp\left(-\frac{(x - x')^{\mathrm{T}} (x - x')}{2l^2} \right) + \sigma_n{}^2 \delta_{x,x'} \tag{5.20}$$

where θ=[l, σ, σ_n] are the hyperparameters and are often called length scale, variance and noise variance, respectively. The regression parameters as a function of the hyperparameters are expressed through:

$$\hat{\beta}(\theta) = (\mathbf{X}^{\mathrm{T}} \mathbf{K}^{-1} \mathbf{X})^{-1} \mathbf{X}^{\mathrm{T}} \mathbf{K}^{-1} \mathbf{y} \tag{5.21}$$

The regression coefficient is derived from the Mean Square Error (MSE) estimator. The log-likelihood function in Eq. 13 has three terms: The first term act as a complexity penalty term to prevent overfitting, the second term measures the fitness to the data and the third term is a log normalization term. The hyperparameters are solved by maximizing the log-likelihood function. This could be easily done with a gradient-based optimization method.

In this section, we take the solution in Eq. (5.21) as the mean predictor for the regression coefficient. This way the constraints can be added into the mean function as in Eq. (5.16):

$$\beta_{en} = \hat{\beta}(\theta) + \eta(\hat{\beta}(\theta), x_0, y_0, dy_0) \tag{5.22}$$

where $\eta(\cdot)$ is a function of the coefficient in Eq. (5.21) and the given value (y_0) and/or derivative constraints (dy_0) at x_0. By replacing the $\hat{\beta}(\theta)$ in Eq. (5.19) with β_{en} and maximizing the likelihood function, the hyperparameters incorporating the constraints can be solved. The realization of the constrained SGP at $x=x_t$ is then calculated according to:

$$\begin{aligned}\hat{y}(x_t) &= \bar{\mathbf{x}}_t^{\mathrm{T}}\beta_{en}(\theta) + \mathbf{K}(x_t, x;\theta)\mathbf{K}^{-1}(\theta)(\mathbf{y} - \mathbf{X}\beta_{en}(\theta)) \\ \varphi(z_t) &= \sigma^2(1 + (\mathbf{X}^{\mathrm{T}}\mathbf{K}^{-1}(\theta)\bar{\mathbf{x}}_t)^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{K}^{-1}(\theta)\mathbf{X})(\mathbf{X}^{\mathrm{T}}\mathbf{K}^{-1}(\theta)\bar{\mathbf{x}}_t) - \bar{\mathbf{x}}_t^{\mathrm{T}}\mathbf{K}^{-1}(\theta)\bar{\mathbf{x}}_t)\end{aligned} \tag{5.23}$$

where $\hat{y}(x_t)$ and $\varphi(x_t)$ are the mean and variance at $x=x_t$ respectively. $\mathbf{X}$ and $\mathbf{y}$ are training data, $\bar{x}_t$ is the basis function of x_t. $\mathbf{K}(\theta)$ is the covariance matrix calculated from training data and $\mathbf{K}(x_t,x;\theta)$ is the covariance vector between the test point and the training data.

5.2.3 BESGP as a trajectory surrogate

As the increase in the demand for air travel, the safety in the National Air Space (NAS) has been a hot topic over the decades. The NextGen air transportation system is aiming at a computer aided Air Traffic Management (ATM) tool for the safety assurance and risk mitigation in NAS. In this task, the trajectory prediction of the aircraft is critical in achieving this goal. A physic-based model for trajectory calculation tends to be time consuming and not ideal for uncertainty analysis. Current researches mainly focus on the trajectory prediction using data-driven method such as machine learning techniques. This type of method often requires an offline training process with a large amount of trajectory data. For a GP, the training needs an evaluation for the matrix inverse for the covariance matrix, the computational complexity increased in the order of $O(n^3)$ as the number of training data. Although the model evaluation for a trained machine learning model could be fast, there are certain constraints exists in the operation of aircraft. These constraints

include standard procedure routes, aircraft performance limitations and FAA regulations. This part introduces the application of the constrained SGP as a trajectory surrogate that can predict aircraft trajectory based on previous part of the flight with waypoint constraint.

As the development in the NAS, the surveillance in aviation is shifting from a radar-based tracking to an Automatic Dependent Surveillance-Broadcast (ADS-B) satellite-based tracking. This enables a more accurate and real-time tracking for the aircraft location. The broadcasted ADS-B data can be used to update and enhance the uncertainties in the trajectory prediction. In this example, an aircraft is descending to land in the San Francisco International Airport (KSFO). The aircraft's future trajectory is predicted using the proposed method based on previous observed ADS-B records, with the additional waypoint constraint at the destination airport. The observed trajectory points are selected from the Sherlock Data Warehouse (SDW), which is a data archive for all flight data in the NAS. The waypoint constraint is set as a value constraint at the coordinate of the landing runway. A quadratic function is chosen as the mean function, and the squared exponential as the covariance function. The regression coefficient for the SGP is revised based on the posterior calculated from Eq. (5.22) with value constraint. Figure 5-4 plotted the predicted trajectory from the proposed constrained SGP comparing with the classical SGP without constraint.

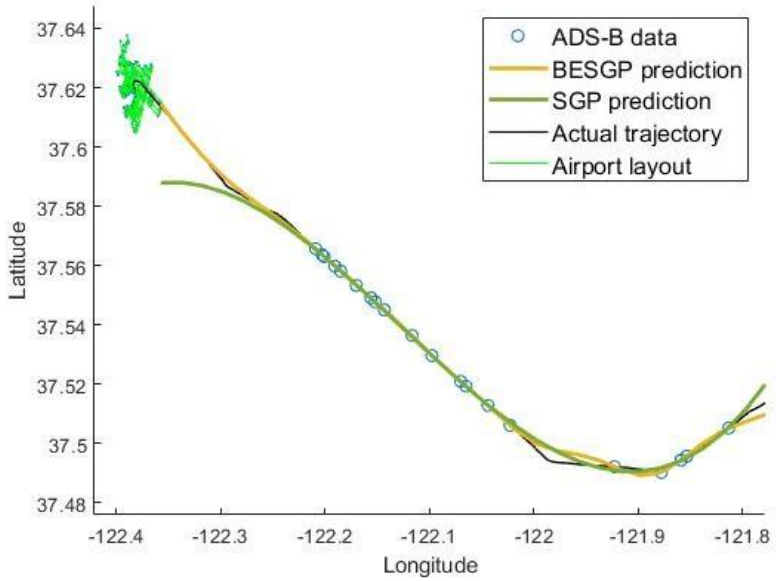


Figure 5-4. The trajectory prediction for future flight based on existing observed location.

The results in Figure 5-4 showed that the constrained SGP can have a more realistic trajectory prediction and better assembles the actual trajectory data. This is due to the waypoint constraint that set the exact location of the destination of the aircraft. The constrained SGP skipped the training with a large quantity of training data by only using the location records from previous part of flight. The proposed method enables a good trajectory prediction based on waypoint constraints. It could be used as a fast surrogate for the real-time or near real-time trajectory uncertainty estimation and reliability analysis in the NAS. Other potentially available constraints can also be considered using the proposed method, such as the direction of the runway.

5.2.4 Conclusion for BESGP

This paper presented a constrained Semiparametric Gaussian Process based on Bayesian-Entropy method. The proposed method can successfully incorporate values and

derivative information as constraints into the regression function. With the help of the encoded constraints outside the range of training data, the regression model can have a better prediction accuracy compared to the classical method. Through the demonstration in aircraft trajectory prediction, the proposed method can achieve a fast and accurate for an aircraft's future trajectory based on previous location records. The proposed method can have a better prediction without the need for training with a large dataset. The mean prediction of the proposed constrained Semiparametric Gaussian Process can satisfy the specified constraints.

5.3 Bayesian-Entropy Gaussian Process

Different from the SGP introduced in the previous section, an ordinary GP does not assume a specific form for the mean function. Instead, it uses a constant as the deterministic mean function. It is usually considered to be a more general case of the GP method. In this section, constraints are introduced into the ordinary GP with the Bayesian-Entropy method. The method will be referred as Bayesian-Entropy Gaussian Process (BEGP). In this framework, the hyperparameters of GP are treated as random variables. The BEGP will directly impose constraints on the GP prediction through the constraints on the hyperparameters' distribution. At last, several different methods for adding constraints into GP regression are compared.

5.3.1 Introduction

Gaussian Process (GP) has been used as a powerful surrogate modeling tool in many engineering applications. It is data efficient and can provide uncertainties for the prediction. The computational cost for a GP model increases in the order of $O(n^3)$ as the number of

training data. Many researches have been focusing on the acceleration and enhancement for training GP. With the mean and variance at a prediction as outputs, an active learning procedure that contains a double-loop optimization, i.e. minimizing the maximum or maximizing the minimum with different learning functions has been developed,

1) Build an initial GP model by the design of experiments (DoE) with a small number of training data.

2) Find a new best point with the acquisition function which trades off exploitation and exploration. Exploitation means sampling where the surrogate model predicts an optimal objective and exploration is to sample at locations where the prediction uncertainty is high. If the acquisition function satisfies some conditions, stop.

3) Evaluate the response of the chosen best sample and add that sample to the DoE. Update the Kriging model and go to Step (2).

The active learning concept is widely used in structural reliability and global optimization. In structural reliability analysis, the GP model is updated based on the learning function that describes the fitting accuracy between the GP model prediction and the real limit state function [90]. The typical method is the EGRA [91]. It selects the samples that can accurately fit the limit state function everywhere in the whole domain, which is a regression problem. An alternative method is to regrade the reliability analysis as a data-drive classification task, such as the AK-MCS [92]. The method proposes that only if the sign of the limit state function can be well predicted, it can meet the accuracy requirement of the failure probability. The acquisition function used in the global optimization field is based on the expected improvement function (EIF) [93]. EIF represents how much the true value

of the response at a sample could be expected to be better than the current best solution. Once the EIF is converged, the global optimal solution has then been obtained.

GP model is a non-parametric model. The computation complexity is $O(N^3+N^2d)$ for inverting the training data covariance matrix and $O(N^2+Nd)$ for the prediction, where N is the number of observations and d is the dimension of the problem. Similar to other surrogate models, it will surf the "curse of dimensionality" due to two reasons,

1) To well represent a high-dimensional response function, a large number of training samples are needed.

2) Building the GP model itself with plenty of observations and large dimensions can be also time-consuming.

Researchers have been pursuing strategies to reduce computational costs in the past few decades. The previous active learning concept is one of them, which is to build the as accurate as possible surrogate model with the minimum number of most relevant training data. Other approaches aim at including more information into GP model known as the cokriging model [94] or gradient-enhanced kriging model (GEK) [95], [96]. These methods are trying to improve the fitting accuracy with the existing data so that the overall needed training data can be reduced. Besides the response values, the GEK model can utilize the cheap gradient information estimated by automatic differentiation [97] to increase the fitting accuracy. Another way to reduce computing costs for large dataset is to build several local GP models [98] and the prediction at the unknown point is a weighted combination of local ones. This method firstly needs to determine the number of local GP models and

partition the domain or the dataset. One issue is that the unknown data at the intersection domain may have different predictions using different local models.

Sometimes there are other information in addition to the point data. For example, there may be existing physical models that can partially describe the regression problem or abstracted statistics from historical data such as mean and variance of the system response, an experienced engineer may have empirical knowledge about the expected value for a certain input. These types of information are essentially different from the direct observation and should be treated as constraints on the regression function rather than directly put into the training dataset. There has been existing research related to introducing constraints into GP models. Some studies add physical constraints by replacing the mean function with a PDE-based physical model and adds a zero-mean GP [99]. This type of method is only using GP to model the systematic error of the data from the PDE, which deviates the concept of the non-parametric nature of GP. [100] optimizes the likelihood function of GP during training subject to constraint on the hyperparameters. However, this method is likely to stuck at a local optimal solution. An interesting study multiplies the trained GP with another probability function to encode constraints [101]. The multiplied probability function is related to the constraint. It will give zero value when the constraint is not met. This method can make sure the realization satisfies the constraint but will likely change the statistical behavior (mean and variance etc.) from the trained GP. It is found that when all training data satisfies a linear constraint, the mean predictor of the trained GP will implicitly satisfy the linear relation. This is because the GP can inherently incorporate

explicit linear relations between input variables [56]. Some researches use this property to manually enforce constraints of the GP predictor [102].

In this section, a Bayesian-Entropy Gaussian Process (BEGP) is proposed to encode constraints into the ordinary GP model. The BE method has been used to incorporate different types of constraint into the classical Bayesian framework [89]. In the previous section, it is shown that by using the BE principle, it is possible to insert values and derivative information as constraints into the regression coefficient in the mean function of SGP. The BEGP, on the other hand, directly impose constraints on the mean prediction at a desired location. This is achieved by considering the hyperparameters for the GP as random variables, the mean prediction can be regarded as a function for the hyperparameters. The constraints on the predicted mean can be regarded as a statistic behavior for the distribution of the hyperparameters. Hence, can be translated into the constraints on the probability function for the hyperparameters. The potential benefit of the proposed method is that: 1) the BEGP can utilize the extra information to enable a more accurate prediction based on the constraints. And 2) the proposed method can be used to smoothly connect multiple GPs with different hyperparameters. This way the computational complexity is reduced for each local GP model while maintaining the continuity of the overall regression.

5.3.2 Bayesian-Entropy for constraining prediction

The previous section discussed using gradient-based optimization for solving the hyperparameters in SGP. This section considers the hyperparameters as random variables and use sampling approach to optimize the MLE solution. The ordinary GP model is a non-

parametric model that is determined through a set of hyper-parameters. The ordinary GP do not assume specific form for the mean function. Hence there is no basis function for the input x. The regression coefficient yields a constant and is calculated as:

$$\beta = (\mathbf{1}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{1})^{-1}\mathbf{1}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y} \tag{5.24}$$

where $\mathbf{K}$ is the covariance matrix of the input of training data, $\mathbf{1}$ is a column vector of 1's with the length equals the number of training data, $\boldsymbol{y}$ is the response of the training data. The likelihood function given data, without taking the logarithm, can be expressed as:

$$l(\theta) = \exp\left[-\frac{1}{2}\log|\mathbf{K}| - \frac{1}{2}(\mathbf{y}-\beta\mathbf{1})^{\mathrm{T}}\mathbf{K}^{-1}(\mathbf{y}-\beta\mathbf{1}) - \frac{1}{2}\log\left|\mathbf{X}^{\mathrm{T}}\mathbf{K}\mathbf{X}\right|\right] \tag{5.25}$$

The mean value for the realization of a GP at a location $x=x_0$ is determined through the training data ($\boldsymbol{x}$,$\boldsymbol{y}$) and hyperparameters (θ) as:

$$m(x_0; \boldsymbol{x}, \boldsymbol{y}, \theta) = \beta + \mathbf{K}(x_0, \boldsymbol{x}; \theta)\mathbf{K}^{-1}(\theta)(\boldsymbol{y} - \mathbf{1}\beta) \tag{5.26}$$

Given the training data and desired input, the mean prediction could be regarded as a function of the hyperparameters. Assume the hyperparameters follow a distribution with PDF $p(\theta)$, the value constraint at $x=x_0$ can be expressed as an integral:

$$\int_{\Theta} m(\theta)\, p(\theta) d\theta = y_0 \tag{5.27}$$

Here $p(\theta)$ presents the predicted mean at $x=x_0$ and y_0 is the corresponding value constraint. The integral is over the domain of the hyperparameters.

Assume the prior for the hyperparameters follows a distribution defined according to $p_0(\theta)$, the posterior after training from data can be calculated through Bayes theorem's:

$$q(\theta) \propto p_0(\theta) l(\theta) \tag{5.28}$$

Based on this posterior, we could find a target distribution function, or BE posterior $p(\theta)$ for the hyperparameters using the BE framework. That is maximize the entropy between the target $p(\theta)$ and the posterior from data $q(\theta)$ under the constraint in Eq. (5.27) as well as a normalization constraint for the PDF function:

$$S = -\int_{\Theta} p(\theta) \log \frac{p(\theta)}{q(\theta)} d\theta \tag{5.29}$$

Similar to the derivation given in Section 2.2, the solution for $p(\theta)$ can be expressed as:

$$p(\theta) \propto q(\theta) \exp(\eta m(\theta)) \tag{5.30}$$

The additional exponential term encodes the mean value constraint in Eq. (5.27). The Lagrange multiplier η is associated with the specific value of the constraint. Substituting the solution in Eq. (5.30) into the mean value constraint and normalization constraint, an equation with respect to η can be formed as:

$$\frac{\int_{\Theta} m(\theta) q(\theta) \exp(\eta m(\theta)) d\theta}{\int_{\Theta} q(\theta) \exp(\eta m(\theta)) d\theta} = y_0 \tag{5.31}$$

In the previous sections, the analytical solution to the BE posterior can be derived since the probability function in the previous application has a closed form for the distribution parameters. In this case, the hyperparameters are embedded in the constraint function $m(\theta)$ through the covariance function **K** and its inverse. It is not easy to seek an

analytical solution for the Lagrange multiplier. Hence, this section seeks numerical solution for Eq. (5.31).

One of the biggest challenges in solving η from Eq. (5.31) is the evaluation for the two integrals. In this study, we use a sampling-based method called importance sampling for calculating the integrals. Importance sampling is a widely used technique for sample from an arbitrary distribution. The importance sampling method draws samples from a known distribution function and calculates the integral as the sum of the value at each sample times the corresponding weight. For example, we want to evaluate an integral given as:

$$I = \int_X f(x)dx \tag{5.32}$$

N samples can be drawn from a proposal distribution according to $p(x)$, the integral is approximated as a summation according to:

$$I \approx Q = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)} \tag{5.33}$$

The benefit of applying the importance sampling is that the complex evaluation of the integrals can be turned into a summation. The accuracy of the method heavily depends on the selected proposal distribution. In this case, sampling method is used for evaluating the distribution for the hyperparameters θ. The posterior samples from $q(\theta)$ after training with data are directly used to measure the two integrals in Eq. (5.31). This way, the probability density in the denominator cancels out with the $q(\theta)$ term in the integrals. Hence, Eq. (5.31) can be evaluated according to:

$$\frac{\sum_i m(\theta_i)\exp(\eta\theta_i)}{\sum_i \exp(\eta\theta_i)} = y_0 \tag{5.34}$$

Eq. (5.33) is then solved with the MATLAB built-in function *fsolve* which utilizes the Trust-Region dogleg algorithm [103][104]. Plug in the solution for η into Eq. (5.30), we have the BE posterior for the hyperparameters. The predictions can be made by assessing the mean prediction over the distribution of the hyperparameters.

The BEGP encodes constraints via directly enforcing on the mean function. The above derivation discusses a constraint on the mean value at a specified input location. Similarly, other forms of constraints can also be added, such as the derivative:

$$\int_{\Theta} \left.\frac{\partial m(\theta)}{\partial x_i}\right|_{x=x_0} p(\theta)d\theta = y_{0,i} \tag{5.35}$$

where the mean prediction can take the derivative with respect to the *i*th dimension of the input *x*. Any arbitrary constraints that can be written in the integral form can be incorporated into proposed the BEGP method.

5.3.3 Constrained GP via adding constraints in likelihood

A straight-forward way for encoding constraints in GP is via a constrained optimization for the log-likelihood function over the hyperparameters. There has already been existing research utilizing a constrained optimization method [100]. But depending on the types of algorithm, the solution may be stuck at a local optimum and the constraints will not be satisfied. [101] ensures the satisfaction of the constraint by multiplying another probability density function based of the GP prediction. This method inevitably changed

the statistical behavior for the GP realization. But regarding the constraint as a probability function and adding it to the likelihood function for optimization is another way to solve the hyperparameters with constraints. This section introduces the method for solving a constrained likelihood function for encoding constraints into GP. The method will be referred as constrained posterior GP.

Following the concept presented in [101], the constraints can be modeled using a probability function where the probability density is high when the constraints are satisfied and yields a minimum value otherwise. Recall the likelihood function defined in Eq. (5.25), assume $p_0(\theta)$ is assigned to the hyperparameters as a prior, the posterior can be expressed as a Bayesian equation for the hyperparameters:

$$p(\theta \mid \mathbf{X}, \boldsymbol{y}) \propto p(\mathbf{X}, \boldsymbol{y} \mid \theta)\, p_0(\theta) \tag{5.36}$$

where $p(\mathbf{X},\boldsymbol{y}|\theta)$ is represents the likelihood of training data given the hyperparameters and is equivalent to the likelihood function $l(\theta)$ in Eq. (5.25).

In the MLE framework, the likelihood function $l(\theta)$ is maximized with respect to the hyperparameters. It is equivalent to the Maximum *a posteriori* (MAP) method for maximizing $p(\theta|\mathbf{X},\boldsymbol{y})$ when an uninformative prior is assigned to $p_0(\theta)$. For the proposed method, the constrained posterior for the hyperparameters $p(\theta|\mathbf{X},\boldsymbol{y},C)$ has an additional probability function associated with the constraints C that is added into Eq. (5.36) as:

$$p(\theta \mid \mathbf{X}, \boldsymbol{y}, C) \propto p(C \mid \mathbf{X}, \boldsymbol{y}, \theta)\, p(\mathbf{X}, \boldsymbol{y} \mid \theta)\, p_0(\theta) \tag{5.37}$$

The term $p(C|\mathbf{X},\boldsymbol{y},\theta)$ is called the constraint likelihood function. Since the training data is independent with the constraints, the posterior can be simplified as:

$$p(\theta \mid \mathbf{X}, \boldsymbol{y}, C) \propto p(C \mid \theta)\, p(\mathbf{X}, \boldsymbol{y} \mid \theta)\, p_0(\theta) \tag{5.38}$$

In this study, the constraint likelihood is assumed with a Normal distribution with zero mean and a small variance ε^2. For example, the value constraint for a mean prediction at $x=x_0$ can be expressed as:

$$p(C \mid \theta) = p(m(\theta; \mathbf{X}, \boldsymbol{y}, x_0) - y_0 \mid \theta) \sim N(0, \varepsilon^2) \tag{5.39}$$

The mean prediction $m(\theta|\mathbf{X},\boldsymbol{y},x_0)$ is a function of the hyperparameters as defined in Eq. (5.26). The strength of the constraint can be modified by changing the value for the variance ε^2. By plugging the constraint likelihood function into Eq. (5.38) and optimize the posterior for the hyperparameters via MAP, we can have the hyperparameters for the GP under the given constraint. Gradient-based method can also be applied for solving the MAP problem, but the optimum solution is not guaranteed depending on the constraint form and chosen variance ε^2.

5.3.4 Conclusion and comparison for different constrained GP method

This section gives a conclusion and comparison for the above-mentioned method for constrained GP modeling, namely the BESGP, BEGP and constrained posterior GP, through a simple one-dimensional numerical example. 20 observational data are randomly generated from a quadratic function $y=(x-1)^2+\varepsilon$ from the range of $x \in [0, 2]$, where ε is a random Gaussian noise with zero mean and $\sigma=0.1$. An arbitrary value constraint is given outside the training data at $x_0=4$ and $y_0=5$. Note this constraint is not a point on the function that generates the data. This example tests the constrained GP algorithm in handling an arbitrary value for the constraints. The SGP assumes a quadratic mean function, while the

BEGP and constrained posterior method uses an ordinary GP. The value constraint at (x_0, y_0) are incorporated into all three methods. For the constrained posterior method, the constraint variance is set to be ε=0.01.

The mean for each of the GP regression function is plotted in Figure 5-5. The BESGP mean prediction has a clear parabolic shape due to the form of the quadratic mean function. While the other two methods based on ordinary GP converge at a constant. It can be observed that all three models can satisfy the constraint but with a small error, i.e. the constraint is not strictly satisfied. For the BESGP method, the error is raised from the zero mean GP in the prediction. The BESGP mean prediction in Eq. (5.23) has two terms, the regression mean function and the zero mean GP. According to BELR, the constraints are strictly followed with the modified regression coefficient from BE method. The zero mean GP, however, will generate a small value at the prediction point, hence the error. The constrained posterior assigned a Normal distribution density to the constraint. Although the assumed variance for the PDF is a small number, there still exist probability when the constraint function is not exactly satisfied. Similar to a constrained optimization method where error may exist, and the specified constraints may not be exactly followed. Similar to the BELR method, the BEGP directly impose constraint of the prediction on the hyperparameters distribution. The BEGP method has the strongest theoretical background for strictly satisfying the given constraint. However, an analytical solution for the BE posterior distribution function for the hyperparameters cannot be achieved. The numerical approximation for integral using sampling-based method may cause error in the solution for the Lagrange multiplier, hence causing error in the prediction. This error could be

reduced by increase the number of samples when evaluating the integrals but will further increase the computational time.

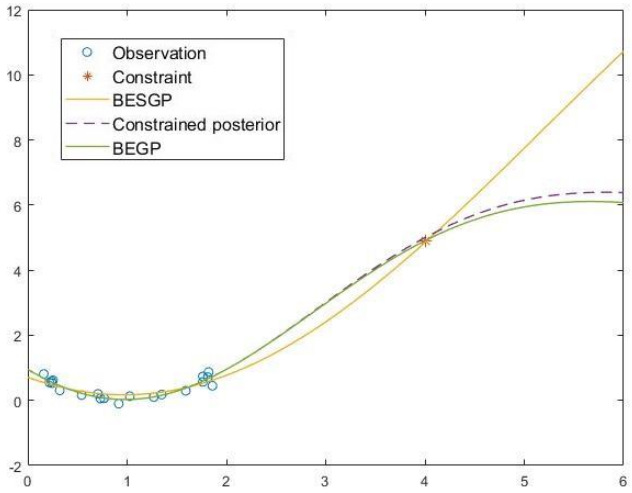


Figure 5-5. The comparison for the mean prediction of three constrained GP algorithm.

5.4 BEGP with multiple constraints

This section applies the BEGP theory into problems with multiple constraints. The computational effort for solving the Lagrange multipliers would increase as the number of constraints. This section proposes a double-loop optimization framework for solving the Lagrange multipliers and the hyperparameters for the BEGP to improve the efficiency. The method is demonstrated in a numerical example and an engineering problem.

5.4.1 Introduction

The previous section introduced the implementation of the Bayesian-Entropy framework into GP method as an alternative way to introduce constraints. Through the derivations, the BEGP has the strongest theoretical background for strictly follow the specified constraints. However, the BEGP considers hyperparameters as random variables

and the solution for the Lagrange multiplier involves numerical approximation for a complex integral over the hyperparameters. Hence, may cause error in the prediction. With multiple values and derivative constraints, solving the Lagrange multipliers involves a nonlinear system of n equations, with n denoting the number of constraints. And each equation includes the numerical approximation for the integral over the hyperparameters. The computational complexity increases as the number of constraints. This could be a potential issue for the practical application of the proposed method.

This section introduced a new algorithm of iteratively solving the Lagrange multipliers and the hyperparameters for the BEGP considering constraints. The algorithm finds the hyperparameters and the Lagrange multipliers solution by maximizing the Bayesian-Entropy posterior of the hyperparameter distribution while satisfying the constraints. This is done in a double-loop optimization scheme. The target is to optimize the error of prediction to the given constraints. When given a set of Lagrange multipliers, the solution for the hyperparameters can be achieved using gradient-based optimization method by maximizing the BE posterior. With the solved hyperparameters, the absolute difference of the prediction to the constraint can be evaluated. The solution to the Lagrange multipliers can be achieved by minimizing this difference. The following section will discuss the details of the algorithm. Two demonstration examples will be given in the third section to demonstrate the BEGP in handling multiple constraints.

5.4.2 Double-loop optimization for Lagrange multipliers and the hyperparameters

Following the formulation in Section 5.3, the derivative constraint can be expressed as the expected value of the derivative of the mean function with respect to the specified dimension:

$$\int_{\Theta}\left[\left.\frac{dm(\theta,x_t)}{dx_t}\right|_{x_t=x_0} p_{BE}(\theta)\right]d\theta = dy_0 \tag{5.40}$$

where $m(\theta)$ is the mean predictor for the GP at x_t. $p_{BE}(\theta)$ is the Bayesian-Entropy posterior for hyperparameters. dy_0 is the derivative at $x_t=x_0$. With multiple mean and derivative constraints, the constraint part for the Bayesian-Entropy posterior consists of the summation over multiple Lagrange multipliers. Each Lagrange multipliers corresponds to one specified constraint. The BE posterior is written as:

$$p_{BE}(\theta) \propto q(\theta)\exp(\sum_i \eta_i m(\theta;x_i) + \sum_j \eta_j \frac{d}{dx} m(\theta;x_j)) \tag{5.41}$$

where $q(\theta)$ is the posterior from data. Back substitute the result in Eq. (5.41) into each constraint, we can have the system of equations with respect to the Lagrange multipliers:

$$\begin{cases} \int_{\Theta} m(\theta;x_i)\, p_{BE}(\theta)d\theta \,/ \int_{\Theta} p_{BE}(\theta)d\theta = y_i \\ \int_{\Theta} \frac{d}{dx} m(\theta;x_j)\, p_{BE}(\theta)d\theta \,/ \int_{\Theta} p_{BE}(\theta)d\theta = dy_j \end{cases} \tag{5.42}$$

An analytical solution would be near impossible. Numerical approximations will add tremendous computational costs as the number of constraints increases.

For a faster calculation for the Lagrange multipliers and hyperparameters, we propose a double-loop optimization process to achieve the solutions. Recall in classical GP, the hyper-parameters are solved by maximizing the likelihood function defined as in Eq. (5.19). With a given set of Lagrange multipliers, the hyperparameters can be solved similarly by maximizing the BE posterior in Eq. (5.41). Given this pair of Lagrange multipliers and the hyperparameters, the error in the constraints can be calculated as:

$$e = \sum_i \left| m(x_i;\theta) - y_i \right| + \sum_j \left| \frac{d}{dx} m(x_j;\theta) - dy_j \right| \tag{5.43}$$

Hence, a double loop optimization for the Lagrange multipliers and hyperparameters can be formed. Assume uninformative prior for the hyperparameters, $q(\theta)$ can be replaced by the likelihood in Eq. (5.19). Hence, the BE posterior can be rewritten as:

$$\begin{aligned} p_{BE}(\theta,\eta_i,\eta_j) \propto \exp(-\frac{1}{2}\log|\mathbf{K}| - \frac{1}{2}(y-\mathbf{X}\hat{\beta}(\theta))^{\mathrm{T}}\mathbf{K}^{-1}(y-\mathbf{X}\hat{\beta}(\theta))) \\ \cdot \exp(\sum_i \eta_i m(\theta;x_i) + \sum_j \eta_j \frac{d}{dx} m(\theta;x_j)) \end{aligned} \tag{5.44}$$

The inner loop finds the hyperparameters by maximizing the BE posterior with a given set of η_i and η_j's:

$$\theta = \arg\max_{\theta}(p_{BE}(\theta;\eta)) \tag{5.45}$$

The outer loop minimizes the error in Eq. (5.43) with respect to η_i and η_j's using the result of the hyperparameters in Eq. (5.45):

$$\boldsymbol{\eta} = \arg\min_{\boldsymbol{\eta}} \left[\sum_i \left| m(x_i;\theta) - y_i \right| + \sum_j \left| \frac{d}{dx} m(x_j;\theta) - dy_j \right| \right] \tag{5.46}$$

The double-loop optimization framework can handle high numbers of constraints within a reasonable timeframe. The solution is based on an assumption that the difference calculated in Eq. (5.46) is equivalent to the constraints in Eq. (5.42) when equals to zero.

5.4.3 Application of BEGP

This section presents two applications of the proposed BEGP with the double-loop optimization algorithm. The first example explores the smooth connection between multiple local GPs. The second example applies BEGP into a structural problem in engineering.

5.4.3.1 Smooth connection of two local GPs

In many situations, there may exist clear distinction for the frequency of the model. Such as the one illustrated in Figure 5-3 b). In these cases, we may need separate local GPs for fitting the different physics within the range of subgroup data. But separate local GPs are not necessarily continuous. The proposed BEGP can ensure smooth connection of the multiple GP models by introducing value and derivative information as constraints into the adjacent GP models. Breaking down a large set of data may also benefit the computational cost compared to fitting a single GP with the whole dataset. In this numerical example, we demonstrate the ability of BEGP for smoothly connecting multiple GPs with value and derivative information.

The training data are generated from a piece-wise function as:

$$y = \begin{cases} 0.5*\sin(4x)+\varepsilon & x \in [0,10) \\ -\cos(x/2)+\cos(5)+\sin(40)/2+\varepsilon & x \in [10,20] \end{cases} \tag{5.47}$$

where x is the input dimension, y is the function response and ε is a random noise term that follows a Normal distribution $N\sim(0,0.1^2)$. The frequency of the first piece of the function is much higher than the other part. The constraints are chosen at x_0=10. The value constraint is specified as y_0=sin(40)/2, and the derivative constraint is specified as dy_0=sin(5)/2. Two groups of training data set are generated with the piece-wise function. Each group has 30 data instances in the range of $x \in [0,10]$ and $x \in [10,20]$, respectively. Two BEGPs with the same value and derivative constraints but with different data group are trained. Three ordinary GPs are fitted, each with group 1 data, group 2 data and the combined dataset, respectively. The comparison results of the BEGP against ordinary GPs are plotted in Figure 5-6.

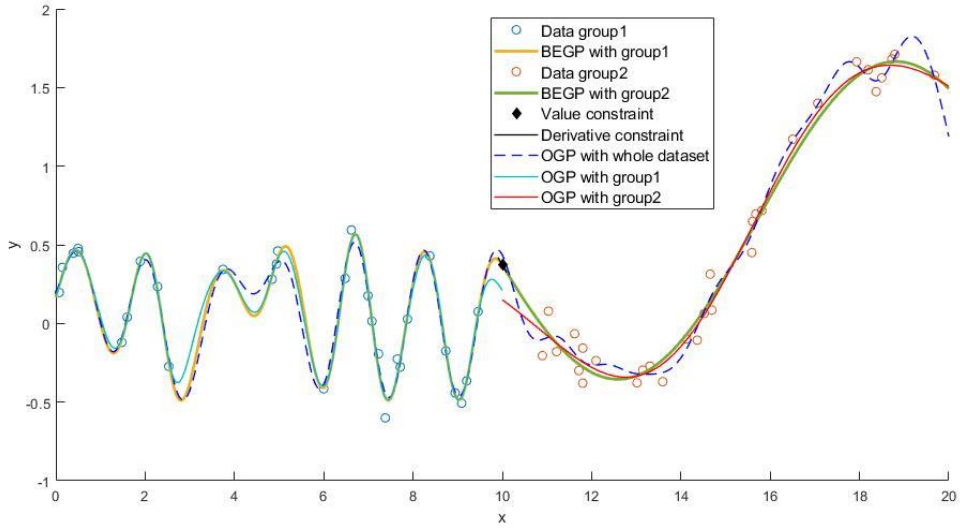


Figure 5-6. BEGP can smoothly connect two local GPs with the specified constraints.

The two BEGPs are smoothly connected at the constraint point. The oscillation in the OGP with the combined dataset has clear overfitting for the group 2 data. The two separate local GPs with group 1 and group 2 data can well captures the underlying physics behind the data in both regions but discontinuous at the connection point. The BEGP has shown its flexibility to incorporate value and derivative constraints for ensure smooth connection between multiple local GPs that has good representation of the function trend for each subgroup data.

5.4.3.2 Application in beam deflection function

This example explores the application of the proposed BEGP in engineering structural problem. The displacement of an engineering structure is a critical measurement related to the system safety. But in some circumstances, the displacement may not be measurable in certain regions of the structure. At the same time, clearly defined boundary conditions may be available due to physics constraints. In some studies, these extra constraints are treated as unbiased observation and put into the training data for the regression model. But these constraints are never truly "observed" so it is not reasonable to be used as training data. The proposed BEGP method can take such extra information as constraints on the GP predictor without use it as training data. This example demonstrates the BEGP in incorporating the boundary conditions as constraints in a structural beam under static loading.

Shown in Figure 5-7 is a structural beam under a static loading on the right end. The boundary condition of the beam is fixed at the left end and simply supported at x=2L. Assume the observable region for this structure is only limited in the dashed box ($x \in$

[0.5L,1.3L]). The displacement measurement or strain gauge can only made in this region. The goal is to fit a GP for the displacement function of the beam. From beam theory, we could get the analytical solution for the deflection function as:

$$w(x)=\begin{cases} -\frac{P}{8EI}x^3+\frac{PL}{4EI}x^2, & x\in[0,2L] \\ \frac{P}{6EI}x^3-\frac{3PL}{2EI}x^2+\frac{7PL^2}{2EI}x-\frac{7PL^3}{3EI}, & x\in(2L,3L] \end{cases} \tag{5.47}$$

where $w(x)$ represents the vertical displacement at x, P is applied force, E is the Young's modulus of the beam and I is the moment of inertia with respect to the bending axis. 10 data instances are randomly generated as training data using Eq. (5.47) from $x\in[0.5L,1.3L]$ with a small noises representing the measurement error and perturbation.

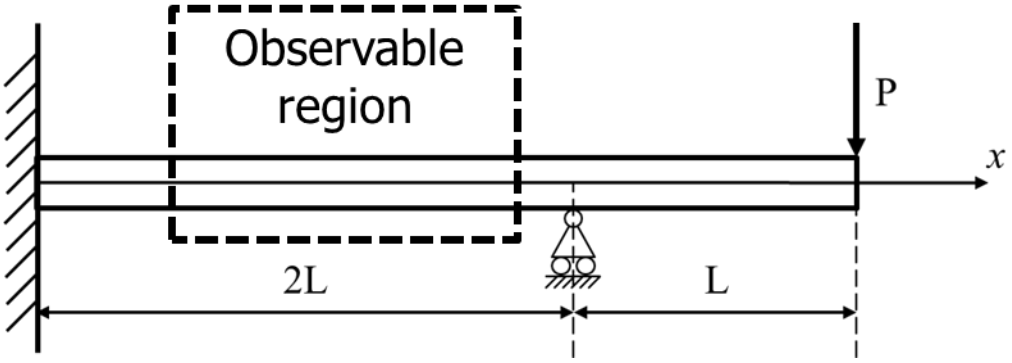


Figure 5-7. A beam under static loading.

The boundary conditions can be translated into a set of value and derivative constraints. That is, the deflection and slope at x=0 is $w(0)=0$, $w'(0)=0$, and the deflection at x=2L is $w(2L)=0$. These three boundary conditions are added into BEGP as constraints on the mean predictor. Plotted in Figure 5-8 are the comparison of the regression result from BEGP encoding the constraints and the ordinary GP with only the training data against the analytical solution from beam theory.

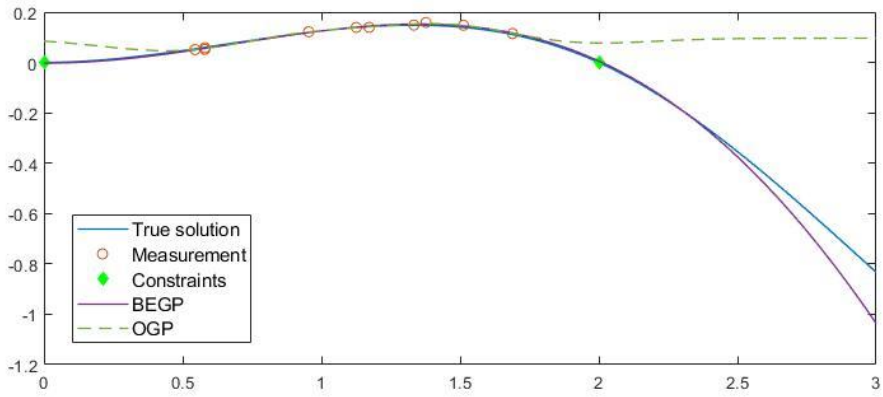


Figure 5-8. BEGP with mean and value constraint for the deflection in beam.

The result shows that the BEGP can strictly follow the specified constraints and can give a more accurate predictions from outside the training data due to the advantage of the encoded boundary conditions.

5.4.3.3 Effect of the noise on constraints

In previous examples, the constraint location is chosen to be exact. But in some cases, there may be errors in the constraint values. This section studies the effect of the randomness in the constraints on the regression result. The numerical example in Section 5.4.3.1 is used as demonstration. Random noise is added into the constraints to perturb the constraint location and constraint value and derivative. Table 5-1 listed a few cases with randomly generated noisy value to add as constraints.

Table 5-1. Different cases with noisy constraints

	Reference	Case 1	Case 2	Case 3	Case 4
x_0	10	9.9804	10.0013	10.0935	9.8919
y_0	0.4868	0.6306	0.5200	0.5037	0.3785
$dy/dx\|x= x_0$	-0.4561	-0.5042	-0.4860	-0.4875	-0.4364

The regression result for each case listed in Table 5-1 is plotted in [fig]. The general trend of the regression function does not change, since we are only introducing a small amount of noise in the constraints. But the BEGP will always follow the specified constraints at the given location. Hence, users need to be careful when choosing the constraint values as the BEGP will always satisfy the constraints. An erroneous constraint may lead to unwanted deviation from the true prediction.

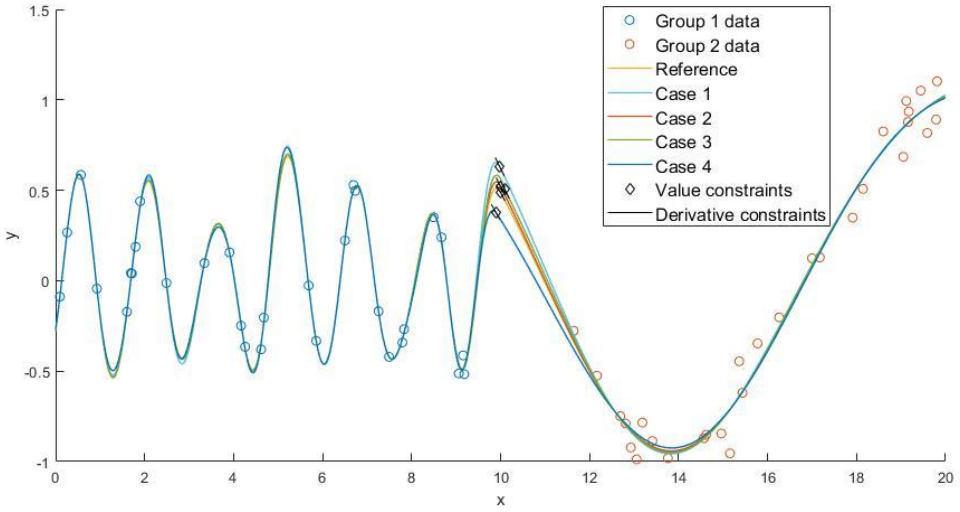


Figure 5-9. The regression results for BEGP with noisy constraint data

Another drawback for BEGP is found when regressing with a noisy constraint is that, when the constraint is too close to a data point, or too far away from the overall function value, the BEGP will suffer from long computational time with a non-optimal result (i.e. cannot satisfy constraints). This is true because intuitively, an arbitrary constraint that deviates too much from the function value will likely be not meaningful in real applications.

5.4.4 Comparison of BEGP with existing method for encoding constraints

As mentioned in the Introduction, the state-of-the-art method for encoding constraints into GP regression can be categorized into three types: 1) Adding constraints via fictitious data as noiseless observations; 2) Encoding PDE constraints by changing the covariance structure; and 3) By constraining on the GP realizations. The first method is not ideal since the constraint data may never be observed, hence not rational to be included in the training data. The second type of method achieves a different goal than the current BEGP framework. The third method sets the constraints on the upper and lower bonds of the GP realization. The realization of a GP is a response surface or function curve that is generated using the trained GP. This type of method can set bounds for both value and the derivative of the GP function realization. The method turns the constrained GP problem into a sampling task from a truncated normal distribution defined by the bounds.

In this section, we compare the BEGP method with the first and the third type of methods of constrained GP modeling. The numerical example for connecting two GP models in Section 5.4.3.1 is used as the demonstration for comparison. A Gradient-Enhanced Kriging (GEK) is used to add the constraint value and derivative at connecting interface of the two local GPs as noiseless observation data. A linear operator inequality constrained GP [105] is used to set bounds for GP realizations at the constrained location. The Upper Bound (UB) and Lower Bound (LB) are defined through an equation with respect to the input dimension x as:

$$\begin{aligned} \text{UB:} y &= 10 - (10 - y_0) \cdot 100^{-(x-x_0)^2} \\ \text{LB:} y &= -10 + (10 + y_0) \cdot 100^{-(x-x_0)^2} \end{aligned} \tag{5.48}$$

where x_0 and y_0 are the constrained value at $x = x_0$. The bounds defined in Eq. (5.48) will be large ($y \in [-10,10]$) when x is far from the constrained location and will quickly converge to the constrained value y_0 once close to x_0. Similarly, the bound function for derivative is defined as:

$$\begin{aligned} \text{UB:} \frac{dy}{dx} &= 10 - (10 - dy_0) \cdot 100^{-(x-x_0)^2} \\ \text{LB:} \frac{dy}{dx} &= -10 + (10 + dy_0) \cdot 100^{-(x-x_0)^2} \end{aligned} \tag{5.48}$$

The comparison result is plotted in Figure 5-10. The constrained GP realization used a Matern52 covariance function whereas the GEK and BEGP used squared exponential.

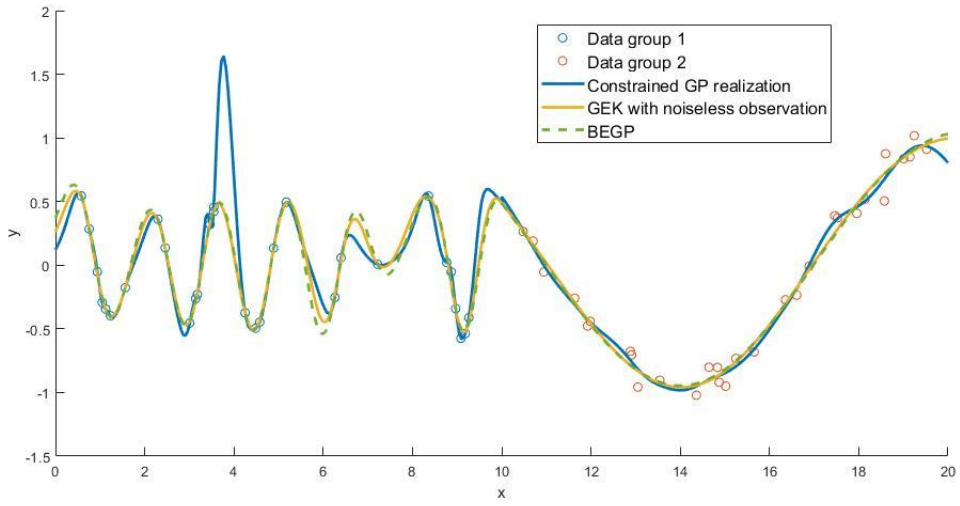


Figure 5-10. The comparison of BEGP with existing methods.

From Figure 5-10 we can see that the constrained GP realization is discontinuous at the intersection, which means that the constraints are not strictly satisfied. This is due to the method uses samples on the truncated normal defined by the bounds. It is likely that the sample mean does not agree with the constraints. On the other hand, this method is an

extra step after the training of GP, hence is more expensive than ordinary GP methods. The GEK with noiseless observation yields a very similar (almost identical) regression result compared to the BEGP regression. But, as aforementioned, the constraints are not reasonable to be treated as observation data. And adding constraints as noiseless data requires manual tuning with the GEK code. Whereas the BEGP offers a more general method for handling constraints.

The comparison proves that the BEGP can be used as an alternative method for encoding constraints into traditional GP method. It can successfully encode value and derivative information as constraint and is theoretically able to incorporate various types of constraint such as moment, range and empirical knowledge. Although the current BEGP framework still takes a long time to calculate comparing to normal GP method, the issue could be solved by seeking analytical derivation or more efficient approximation or optimization methods.

5.4.5 Conclusion for BEGP with multiple constraints

The BEGP with the double-loop optimization algorithm can efficiently encode multiple constraints into the classical GP method. The solution of the BEGP can successfully incorporate value and derivative constraints into the mean predictor for the GP. Through the two demonstration examples, the BEGP can be used for smoothly connecting local GPs with different function characteristics by adding value and derivative constraints at the connection point. The BEGP can also be used for adding physics constraints such as boundary conditions in mechanics problem to enhance the prediction ability.

5.5 Conclusion

This chapter proposed three different surrogate modeling algorithms based on the Bayesian-Entropy principle. The proposed algorithms can all successfully achieve information fusion by incorporating information other than point observations. Such extra information is treated as value or derivative constraints on the surrogate models. All three methods have pros and cons. Table 5-2 listed the comparison between the three constrained regression method.

Table 5-2. Comparison of three different constrained regression

	BELR	BESGP	BEGP
Pro	Strictly follows specified constraints.	Flexible regression. Enhanced extrapolation with constraints	Flexible regression. No need for basis function. Strictly follows constraints.
Con	Need to assume basis function. Not flexible as other regression methods	Need basis function. Constraints not strictly followed.	Higher computational cost. Assumes mean equals mode.

The first method applies the BE principle in the classical Bayesian linear regression to form the Bayesian-Entropy Linear Regression (BELR). The analytical solution for the regression coefficient incorporating value and derivative constraints is given. In the BELR framework, the constraints can be strictly satisfied. But the regression function is strictly limited by the functional form for the selected basis. The BELR framework is then applied in a more flexible regression method, Semiparametric Gaussian Process (SGP) to form a Bayesian-Entropy SGP (BESGP) that can successfully consider value and derivative constraints outside the range of training data. The constraints are added into the mean function of the SPG. The zero mean GP part of the SGP could produce a small error so that

the constraints are not strictly followed. The Bayesian-Entropy Gaussian Process (BEGP), which is a more general constrained GP method that is based on an ordinary GP, is proposed as another alternative for constrained surrogate modeling. The BEGP is a more generalized method since it does not assume any form for the mean function. The constraint is imposed directly on the prediction of the GP via the Bayesian-Entropy principle. Theoretically, the BEGP method should be the most rigorous method for constrained GP modeling. But due to the complex nature of the formulation, there exist some challenge in getting an analytical solution for the hyperparameters. Numerical method introduced error for the hyperparameters solution. Although a double-loop optimization setup could greatly enhance the computational time for the calculation, it is based on an assumption that the expected value of the constraints is equivalent to the result from Maximum a posteriori (MAP). Potential future research may be needed for finding possible analytical solution for the BEGP. A more efficient sampling method can also increase the accuracy for the methodology. The assumptions for the double-loop optimization needs to be verified.

6 Conclusion and Future work

The focus of the present research is on the information fusion into Bayesian framework. This research aims at utilizing various types of information that is not typically handled in the classical Bayesian method through the implementation of Bayesian/Bayesian-Entropy methodology. It is demonstrated that the Bayesian-Entropy framework can consider extra information such as empirical knowledge or known physics in the form of constraint based on the classical Bayesian method. The posterior probability function from the Bayesian-Entropy theory consists of two parts: A Bayesian part that handles point data as the classical Bayesian method, and an extra exponential Entropy part that encodes constraints. The Bayesian-Entropy posterior is analytically solvable given the specific form of constraints ranging from statistical moment constraints, range constraints and general function as constrains. The extra Entropy term will be dropped when no constraint is presented, and the Bayesian-Entropy posterior will yield the classical Bayes' theorem. Hence, the Bayesian-Entropy is compatible and can be easily extended into any Bayesian methods such as Bayesian Network, Bayesian classifier and Bayesian regression etc. In this research, the Bayesian-Entropy method is proposed as a generalized Bayesian method to achieve information fusion in various applications in diagnostics and prognostics in engineering fields.

In diagnostics, the Bayesian Network is often used as a classifier and as a statistical modeling tool for damage identification and classification. This research extended the Bayesian-Entropy method into a network format as a Bayesian-Entropy Network (BEN) to account for multiple types of information commonly seen in engineering practices. When the BEN is used as a classifier, it can accommodate empirical knowledge into the classifier

and can benefit the prediction accuracy when the training data size is small. The behavior will eventually converge to the classical Bayesian Network Classifier as the available training data increases. When used as a statistical model for damage location identification, it has been demonstrated that the BEN can compare with the computationally expensive physical model by encoding the known physics as constraints into the BEN model. The BEN has the same topology as the classical BN, but with the additional constraint term in the inferencing equation, the BEN can utilize the available extra information to enhance its performance. In general, the BEN is flexible in handling various types of information that is commonly seen in many engineering applications.

In prognostics, through the application in air traffic safety assessment, the BEN is able to encode conceptual information, human specified data as well as revising the correlation between variables through the added constraints in the inferencing equation. A Bayesian-Entropy Model Selection (BEMS) framework is developed based on Bayesian model selection to consider the effect of model uncertainty while incorporating the known constraints in model parameters. With the help of the Bayesian-Entropy method, the vastly available information of different types in National Airspace System (NAS) can be fused to better assess the safety and predict unwanted events. Through the demonstrated examples in different scenarios, the BEN is a unique and important approach in achieving the information fusion in NAS system and provide a computer aided framework for the safety management and risk prediction in the NextGen.

The Bayesian-Entropy principle has also been found beneficial in surrogate modeling. Information such as value and derivative can be easily encoded into the regression function

coefficient as constraints. The value constraints information may be available through human concept or empirical knowledge. The derivative information can be coming from available or creditable physics models. The value and derivative information can be regarded as the expected value of the target function, and hence can be written as an integral over the regression coefficient. This aligns with the Bayesian-Entropy principle in incorporating integral constraints. The result for the Bayesian-Entropy posterior of the regression coefficient based on Bayesian linear regression is analytically derived to form a constrained regression model called Bayesian-Entropy Linear Regression (BELR). The BELR can successfully incorporate value and derivative constraints. Later, the BELR is applied into the mean function in a Semi-parametric Gaussian Process (SGP). The SGP is a more flexible method for surrogate modeling under uncertainty. The proposed Bayesian-Entropy SGP (BESGP) can take in available constraints to help the extrapolation of the regression model.

At last, the Bayesian-Entropy principle finds its application into the ordinary GP, which is a more generalized regression model without the assumption for a mean function. This method is proposed as Bayesian-Entropy GP (BEGP). The hyperparameters in the GP model is regarded as a random variable. The BE principle is used to directly impose constraints at the predicted system response. This way the constraints are used to control the distribution of the hyperparameters. It is a more rigorous way of handling constraints and, theoretically, the constraint should be strictly satisfied by the regression model. Unfortunately, in this case, an analytical solution for the hyperparameter distribution function is not achieved due to the highly complex formulation of the likelihood function with respect to the hyperparameters. Despite, a numerical approximation can be done using

sampling method. According to the result of a toy problem, although the BEGP may suffer from the error raise by numerical approximation, it provides an alternative way of introducing extra information as constraints into surrogate modeling. In the later studies, it is found that a double-loop optimization algorithm can be used for efficiently solving the Lagrange multipliers and the hyperparameters. The benefits of the proposed models are: 1) Enhancing the surrogate model behavior by utilizing the extra information. This can enable a more accurate prediction for outside the training data range based on the constraints, and 2) the proposed method can be used to smoothly connect multiple GPs with different hyperparameters. This way the computational complexity is reduced for each local GP model while maintaining the continuity of the overall regression.

To conclude, a Bayesian-Entropy framework is proposed as a general tool for the fusion of different types of information. The proposed method can benefit in the diagnostic and prognostic in engineering problem as well as the surrogate modeling under constraints. The potential future direction of the research can be:

1) Exploration of efficient sampling algorithm or analytical approximation methods, such as variational Expectation Maximization, for the solution of hyperparameters for BEGP can reduce the computational time for finding the Bayesian-Entropy solution.
2) Analytical derivation for the solution of hyperparameters for the BEGP under constraints can significantly increase the efficiency and accuracy of the model training and prediction. Hence, will benefit the smooth connection of multiple local

GP models for enhanced performance of the surrogate and reduce the computational complexity comparing to building one overall GP.

3) The Bayesian-Entropy principle may also be applied to other popular data-driven methods such as Bayesian Neural Network to build a Bayesian-Entropy Neural Network. This will enable a novel method for deep learning under constraints with uncertainties.

Reference

[1] M. Bayes and M. Price, “An Essay towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, F. R. S. Communicated by Mr. Price, in a Letter to John Canton, A. M. F. R. S.,” *Philos. Trans. R. Soc. London*, vol. 53, pp. 370–418, Jan. 1763.

[2] F. V. Jensen and T. D. Nielsen, *Bayesian Networks and Decision Graphs*. New York, NY: Springer New York, 2007.

[3] T. Peng, A. Saxena, K. Goebel, Y. Xiang, S. Sankararaman, and Y. Liu, “A novel Bayesian imaging method for probabilistic delamination detection of composite materials,” *Smart Mater. Struct.*, vol. 22, no. 12, 2013.

[4] T. Peng, J. He, Y. Liu, A. Saxena, J. Celaya, and K. Goebel, “Integrated fatigue damage diagnosis and prognosis under uncertainties,” *Phm*, no. February 2016, pp. 1–11, 2012.

[5] M. Scanagatta, G. Corani, C. P. de Campos, and M. Zaffalon, “Learning Bounded Treewidth Bayesian Networks with Thousands of Variables,” in *Advances in Neural Information Processing Systems*, 2016, pp. 1864–1872.

[6] D. Heckerman, D. Geiger, and D. M. Chickering, “Learning Bayesian networks: The combination of knowledge and statistical data,” *Mach. Learn.*, vol. 20, no. 3, pp. 197–243, Sep. 1995.

[7] K. Jayech, “Application of Bayesian Networks for Pattern Recognition : Character Recognition Case,” pp. 748–757, 2012.

[8] L. Likforman-Sulem and M. Sigelle, “Recognition of degraded characters using dynamic Bayesian networks,” *Pattern Recognit.*, vol. 41, no. 10, pp. 3092–3103, 2008.

[9] K. Jayech, “Clustering and Bayesian network for image of faces classification,” *IJACSA Int. J. Adv. Comput. Sci. Appl.*, no. Special Issue, 2011.

[10] L. Zhang and Q. Ji, “A Bayesian Network Model for Automatic and Interactive Image Segmentation,” vol. 20, no. 9, pp. 2582–2593, 2011.

[11] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian Network Classifiers,” *Mach. Learn.*, vol. 29, pp. 131–163, 1997.

[12] S. Sankararaman and S. Mahadevan, “Likelihood-based representation of epistemic uncertainty due to sparse point data and/or interval data,” *Reliab. Eng. Syst. Saf.*, 2011.

[13] J. Graca, K. Ganchev, and B. Taskar, “Expectation Maximization and Posterior Constraints,” *Adv. Neural Inf. Process. Syst. 20*, pp. 569–576, 2008.

[14] K. Ganchev and J. Gillenwater, “Posterior Regularization for Structured Latent Variable Models,” *J. Mach. Learn. Res.*, vol. 11, no. MS-CIS-09-16, pp. 2001–2049,

2010.

[15] J. Zhu, N. Chen, and E. P. Xing, “Bayesian Inference with Posterior Regularization and applications to Infinite Latent SVMs,” *J. Mach. Learn. Res.*, vol. 15, pp. 1799–1847, 2014.

[16] E. VanDerHorn and S. Mahadevan, “Bayesian model updating with summarized statistical and reliability data,” *Reliab. Eng. Syst. Saf.*, vol. 172, no. December 2017, pp. 12–24, 2018.

[17] E. T. Jaynes, “Information theory and statistical mechanics,” *Phys. Rev.*, vol. 106, no. 4, pp. 620–630, 1957.

[18] E. T. Jaynes, “Information Theory and Statistical Mechanics. II,” *The Physical Review*, vol. 108, no. 2. pp. 171–190, 1957.

[19] A. Caticha and A. Giffin, “Updating probabilities,” in *AIP Conference Proceedings*, 2006, vol. 872, pp. 31–42.

[20] K. Friedman and A. Shimony, “Jaynes’s maximum entropy prescription and probability theory,” *Journal of Statistical Physics*, vol. 3, no. 4. pp. 381–384, 1971.

[21] A. Shimony, “The status of the principle of maximum entropy,” *Synthese*, vol. 63, no. 1, pp. 35–53, 1985.

[22] A. Giffin and A. Caticha, “Updating probabilities with data and moments,” in *AIP Conference Proceedings*, 2007, vol. 954, pp. 74–84.

[23] J. N. YANG and W. J. TRAPP, “Reliability Analysis of Aircraft Structures under Random Loading and Periodic Inspection,” *AIAA J.*, vol. 12, no. 12, pp. 1623–1630, Dec. 1974.

[24] J. L. Beck and L. S. Katafygiotis, “Updating Models and Their Uncertainties. I: Bayesian Statistical Framework,” *J. Eng. Mech.*, vol. 124, no. 4, pp. 455–461, Apr. 1998.

[25] D. Kavetski, G. Kuczera, and S. W. Franks, “Bayesian analysis of input uncertainty in hydrological modeling: 2. Application,” *Water Resour. Res.*, vol. 42, no. 3, Mar. 2006.

[26] G. Christakos, “A Bayesian/maximum-entropy view to the spatial estimation problem,” *Math. Geol.*, vol. 22, no. 7, pp. 763–777, Oct. 1990.

[27] G. Christakos, *Integrative problem-solving in a time of decadence*. Springer Science & Business Media, 2010.

[28] A. Adam-Poupart, A. Brand, M. Fournier, M. Jerrett, and A. Smargiassi, “Spatiotemporal modeling of ozone levels in Quebec (Canada): a comparison of kriging, land-use regression (LUR), and combined Bayesian maximum entropy-LUR approaches.,” *Environ. Health Perspect.*, vol. 122, no. 9, pp. 970–6, Sep. 2014.

[29] S. Banerjee, B. Carlin, and A. Gelfand, *Hierarchical modeling and analysis for spatial data*. CRC press, 2014.

[30] A. Kolovos, J. Angulo, … K. M.-E., and undefined 2013, "Model-driven development of covariances for spatiotemporal environmental health assessment," *Springer*, vol. 185, no. 1, pp. 815–831, 2013.

[31] S.-J. Lee and E. A. Wentz, "Applying Bayesian Maximum Entropy to extrapolating local-scale water consumption in Maricopa County, Arizona," *Water Resour. Res.*, vol. 44, no. 1, 2008.

[32] K. P. Messier, T. Campbell, P. J. Bradley, and M. L. Serre, "Estimation of Groundwater Radon in North Carolina Using Land Use Regression and Bayesian Maximum Entropy," *Environ. Sci. Technol.*, vol. 49, no. 16, pp. 9817–9825, 2015.

[33] S. Tang, X. Yang, D. Dong, and Z. Li, "Merging daily sea surface temperature data from multiple satellites using a Bayesian maximum entropy method," *Front. Earth Sci.*, vol. 9, no. 4, pp. 722–731, 2015.

[34] X. Guan, R. Jha, and Y. Liu, "Probabilistic fatigue damage prognosis using maximum entropy approach," *J. Intell. Manuf.*, vol. 23, no. 2, pp. 163–171, 2012.

[35] C. Schmalz, F. Forster, A. Schick, and E. Angelopoulou, "An endoscopic 3D scanner based on structured light," *Med. Image Anal.*, vol. 16, no. 5, pp. 1063–1072, 2012.

[36] R. G. Gould, M. J. Lipton, P. Mengers, and R. Dahlberg, "Investigation of a video frame averaging digital subtraction fluoroscopic system," *Proc. SPIE*, vol. 0314, pp. 184–190, Nov. 1981.

[37] Q. Chang and Y. Liu, "A Novel Computational Method Modeling Wave propagation using K-space method and Damage Detection using Adjoint Method," in *58th AIAA/ASCE/AHS/ASC Structures, Structural*, 2017, pp. 1–6.

[38] J. He *et al.*, "A multi-feature integration method for fatigue crack detection and crack length estimation in riveted lap joints using Lamb waves," *Smart Mater. Struct.*, vol. 22, no. 10, p. 105007, Oct. 2013.

[39] Q. Chang, T. Peng, and Y. Liu, "Tomographic damage imaging based on inverse acoustic wave propagation using k-space method with adjoint method," *Mech. Syst. Signal Process.*, vol. 109, pp. 379–398, Sep. 2018.

[40] "Probabilistic Fatigue Damage Diagnostics and Prognostics for Metallic and Composite Materials by Tishun Peng A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy Approved May 2016 by the Graduate Supervis," no. June, 2016.

[41] M. R. Hoseini, X. Wang, and M. J. Zuo, "Estimating Ultrasonic Time of Flight Using Envelope and Quasi Maximum Likelihood Method for Damage Detection and Assessment," *Measurement*, vol. 45, pp. 2072–2080, 2012.

[42] Y. Wang, Y. Liu, Z. Sun, and P. Tang, "A Bayesian-Entropy Network for Information Fusion and Reliability Assessment of National Airspace Systems," in *PHM Society Conference*, 2018, vol. 10, no. 1.

[43] A. Hotho, A. Nürnberger, G. Paaß, and F. Ais, "A Brief survey of text mining," *Ldv Forum*, vol. 20, no. 1, pp. 19–62, 2005.

[44] Y. Zhang, R. Jin, and Z. H. Zhou, "Understanding bag-of-words model: A statistical framework," *Int. J. Mach. Learn. Cybern.*, vol. 1, no. 1–4, pp. 43–52, Dec. 2010.

[45] L. Wu, S. C. H. Hoi, and N. Yu, "Semantics-preserving bag-of-words models and applications," *IEEE Trans. Image Process.*, vol. 19, no. 7, pp. 1908–1920, Jul. 2010.

[46] W. B. Cavnar and J. M. Trenkle, "N-Gram-Based Text Categorization," *Proc. Third Annu. Symp. Doc. Anal. Inf. Retr.*, pp. 1–14, 2001.

[47] X. Rong, "word2vec Parameter Learning Explained," Nov. 2014.

[48] D. Wei *et al.*, "Research on Unstructured Text Data Mining and Fault Classification Based on RNN-LSTM with Malfunction Inspection Report," *Energies*, vol. 10, no. 3, p. 406, Mar. 2017.

[49] F. Ali, S. El-Sappagh, and D. Kwak, "Fuzzy Ontology and LSTM-based Text Mining: A Transportation Network Monitoring System for Assisting Travel," *Sensors*, vol. 19, no. 2, p. 234, Jan. 2019.

[50] N. Reddy and M. Padma, "NTSB Aviation Accidents Analysis Using R," *Int. J. Emerg. Trends Technol. Comput. Sci.*, vol. 6, no. 4, 2017.

[51] M. Bazargan, M. Johnson, and A. Vijayanarayanan, "An Evaluation of AIRES and STATISTICA Text Mining Tools as Applied to General Aviation Accidents," 2013.

[52] X. Zhang and S. Mahadevan, "Ensemble machine learning models for aviation incident risk prediction," *Decis. Support Syst.*, vol. 116, pp. 48–63, Jan. 2019.

[53] A. De Voogt and R. R. A. Van Doorn, "Helicopter accidents: Data-mining the NTSB database," in *33rd European Rotorcraft Forum 2007, ERF33*, 2007, vol. 4, pp. 2517–2523.

[54] P. Srinivasan, V. Nagarajan, and S. Mahadevan, "Mining and Classifying Aviation Accident Reports," in *AIAA Aviation 2019 Forum*, 2019.

[55] S. Atkinson, S. Ghosh, N. Chennimalai Kumar, G. Khan, and L. Wang, "Bayesian task embedding for few-shot Bayesian optimization," 2020.

[56] W. K. Christopher and C. E. Rasmussen, *Gaussian processes for machine learning*, vol. 2. Cambridge, MA: MIT press, 2006.

[57] H. Salimbeni and M. P. Deisenroth, "Doubly Stochastic Variational Inference for Deep Gaussian Processes," 2017.

[58] S. Wilke, A. Majumdar, and W. Y. Ochieng, "Modelling runway incursion severity,"

Accid. Anal. Prev., vol. 79, pp. 88–99, 2015.

[59] FAA, "Flight Risk Assessment Tools," 2007. [Online]. Available: http://www.faa.gov/documentLibrary/media/.

[60] FAA, "Safety Management System (SMS)," 2015. [Online]. Available: https://www.faa.gov/about/initiatives/sms/.

[61] J. J. H. Liou, G.-H. Tzeng, and H.-C. Chang, "Airline safety measurement using a hybrid model," *J. Air Transp. Manag.*, vol. 13, no. 4, pp. 243–249, Jul. 2007.

[62] M. Rodgers, *Human factors impacts in air traffic management*. Routledge, 2017.

[63] B. J. M. Ale *et al.*, "Further development of a Causal model for Air Transport Safety (CATS): Building the mathematical heart," *Reliab. Eng. Syst. Saf.*, vol. 94, no. 9, pp. 1433–1441, Sep. 2009.

[64] Y. Liu and K. Goebel, "Information Fusion for National Airspace System Prognostics," *PHM Soc. Conf.*, vol. 10, no. 1, pp. 1–13, Sep. 2018.

[65] M. Strohmeier, M. Schäfer, V. Lenders, and I. Martinovic, "Realities and challenges of nextgen air traffic management: The case of ADS-B," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 111–118, 2014.

[66] L. Rognin, I. Grimaud, E. Hoffman, and K. Zeghal, "Implementing changes in controller-pilot tasks distribution: the introduction of limited delegation of separation assurance," in *International Workshop on Human Error, Safety and Systems Development (HESSD)*, 2001.

[67] L. Martin, N. Bienert, L. Claudatos, V. Gujral, J. Kraut, and J. Mercer, "Effects of task allocation on air traffic management human-automation system performance," *2016 IEEE/AIAA 35th Digit. Avion. Syst. Conf.*, pp. 1–8, 2016.

[68] J. C. H. Cheung, "Flight planning: node-based trajectory prediction and turbulence avoidance," *Meteorol. Appl.*, vol. 25, no. 1, pp. 78–85, 2018.

[69] J. L. Yepes, I. Hwang, and M. Rotea, "New Algorithms for Aircraft Intent Inference and Trajectory Prediction," *J. Guid. Control. Dyn.*, vol. 30, no. 2, pp. 370–382, 2007.

[70] N. Takeichi, R. Kaida, A. Shimomura, and T. Yamauchi, "Prediction of Delay due to Air Traffic Control by Machine Learning," in *AIAA Modeling and Simulation Technologies Conference*, 2017.

[71] S. J. Landry, J. Archer, and N. Nguyen, "Enumeration of National Airspace System uncertainties within an agent-based, state-based model," in *2013 Aviation Technology, Integration, and Operations Conference*, 2013, p. 4224.

[72] B. Sridhar, S. Grabbe, and A. Mukherjee, "Modeling and optimization in traffic flow management," *Proc. IEEE*, vol. 96, no. 12, 2008.

[73] I. Lymperopoulos and J. Lygeros, "Sequential Monte Carlo methods for multi-

aircraft trajectory prediction in air traffic management," *Int. J. Adapt. Control Signal Process.*, vol. 24, no. 10, pp. 830–849, 2010.

[74] S. Sankararaman and M. Daigle, "Uncertainty Quantification in Trajectory Prediction for Aircraft Operations," *AIAA Guid. Navig. Control Conf.*, no. January, pp. 1–11, 2017.

[75] I. Roychoudhury *et al.*, "Predicting Real-Time Safety of the National Airspace System," in *AIAA Infotech@ Aerospace*, 2016, p. 2131.

[76] Eurocontrol, "Base of Aircraft Data (BADA)," 2018. [Online]. Available: http://www.eurocontrol.int/services/bada. [Accessed: 30-Apr-2018].

[77] P. K. Menon, B.-J. Yang, P. Dutta, S. G. Park, O. Chen, and V. H. L. Cheng, "A Computational Platform for Analyzing the Safety of the National Airspace System," in *PHM Society Conference*, 2018, vol. 10, no. 1.

[78] D. McCallie, J. Butts, and R. Mills, "Security analysis of the ADS-B implementation in the next generation air transportation system," *Int. J. Crit. Infrastruct. Prot.*, vol. 4, no. 2, pp. 78–87, Aug. 2011.

[79] Y. Wang and Y. Liu, "A Novel Bayesian Entropy Network for Probabilistic Damage Detection and Classification," in *AIAA Non-Deterministic Approaches Conference, 2018*, 2018, no. 209969.

[80] M. Gafni, "Exclusive: Air Canada near-miss at SFO sparks FAA probe," *San Jose Mercury News*, 2017. [Online]. Available: https://www.mercurynews.com/2017/07/10/exclusive-sfo-near-miss-might-have-triggered-greatest-aviation-disaster-in-history/.

[81] X. Guan, R. Jha, and Y. Liu, "Model selection, updating, and averaging for probabilistic fatigue damage prognosis," *Struct. Saf.*, vol. 33, no. 3, pp. 242–249, 2011.

[82] M. Gafni, "Plane lands on taxiway: Will FAA review cockpit recordings?," *San Jose Mercury News*, 2018. [Online]. Available: https://www.mercurynews.com/2018/01/26/the-mystery-at-pullman-airport-plane-lands-on-taxiway-but-was-key-evidence-inspected/.

[83] A. Moreno and R. Takeo, "Plane lands on taxiway instead of runway in Seattle," *USA Today*, 2015. [Online]. Available: https://www.usatoday.com/story/news/nation-now/2015/12/29/plane-lands-taxiway-instead-runway-seattle/78056520/.

[84] A. E. Raftery, D. Madigan, and J. A. Hoeting, "Bayesian Model Averaging for Linear Regression Models," *J. Am. Stat. Assoc.*, vol. 92, no. 437, p. 179, Mar. 1997.

[85] B. P. Carlin and T. A. Louis, *Bayesian methods for data analysis*. CRC Press, 2008.

[86] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures*

on machine learning, Springer, 2004, pp. 63–71.

[87] M. A. Bouhlel and J. R. R. A. Martins, "Gradient-enhanced kriging for high-dimensional problems," *Eng. with Comput.* , vol. 35, pp. 157–173, 2019.

[88] C. J. Paciorek and M. J. Schervish, "Nonstationary Covariance Functions for Gaussian Process Regression," in *Advances in neural information processing systems*, 2004.

[89] Y. Wang and Y. Liu, "Bayesian entropy network for fusion of different types of information," *Reliab. Eng. Syst. Saf.*, vol. 195, p. 106747, Mar. 2020.

[90] Z. Sun, J. Wang, R. Li, and C. Tong, "LIF: A new Kriging based learning function and its application to structural reliability analysis," *Reliab. Eng. Syst. Saf.*, vol. 157, pp. 152–165, 2017.

[91] B. J. Bichon, M. S. Eldred, L. P. Swiler, S. Mahadevan, and J. M. McFarland, "Efficient global reliability analysis for nonlinear implicit performance functions," *AIAA J.*, vol. 46, no. 10, pp. 2459–2468, Oct. 2008.

[92] B. Echard, N. Gayton, and M. Lemaire, "AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation," *Struct. Saf.*, vol. 33, no. 2, pp. 145–154, 2011.

[93] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *J. Glob. Optim.*, vol. 13, no. 4, pp. 455–492, 1998.

[94] J. Laurenceau and P. Sagaut, "Building efficient response surfaces of aerodynamic functions with kriging and cokriging," *AIAA J.*, vol. 46, no. 2, pp. 498–507, 2008.

[95] H. Yao, Y. Gao, and Y. Liu, "FEA-Net: A physics-guided data-driven model for efficient mechanical response prediction," *Comput. Methods Appl. Mech. Eng.*, vol. 363, p. 112892, 2020.

[96] Y. Gao and Y. Liu, "Adjoint Gradient-enhanced Kriging Model for Time-dependent Reliability Analysis," in *AIAA Scitech 2019 Forum*, 2019, p. 441.

[97] Y. L. Yi Gao, "'Adjoint-FORM for efficient reliability analysis of large-scale structural problems,'" *2018 AIAA Non-Deterministic Approaches Conf. AIAA SciTech Forum*, 2018.

[98] Z.-H. Han, Y. Zhang, C.-X. Song, and K.-S. Zhang, "Weighted Gradient-Enhanced Kriging for High-Dimensional Surrogate Modeling and Design Optimization," *AIAA J.*, pp. 1–17, 2017.

[99] H. Zhao, R. Jin, S. Wu, and J. Shi, "PDE-constrained Gaussian process model on material removal rate of wire saw slicing process," *J. Manuf. Sci. Eng. Trans. ASME*, vol. 133, no. 2, Apr. 2011.

[100] J. Matschek, A. Himmel, K. Sundmacher, and R. Findeisen, "Constrained Gaussian Process Learning for Model Predictive Control," Nov. 2019.

[101] J. Z. Liu, “Gaussian Process Regression and Classification under Mathematical Constraints with Learning Guarantees,” 2019.

[102] M. Salzmann and R. Urtasun, “Implicitly Constrained Gaussian Process Regression for Monocular Non-Rigid Pose Estimation,” in *Advances in Neural Information Processing Systems*, 2010, pp. 2065–2073.

[103] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*. Society for Industrial and Applied Mathematics, 2000.

[104] Z. Yingliang and X. Chengxian, “A NEW TRUST REGION DOGLEG METHOD FOR UNCONSTRAINED OPTIMIZATION,” 2000.

[105] C. Agrell, “Gaussian processes with linear operator inequality constraints,” *J. Mach. Learn. Res.*, vol. 20, pp. 1–36, Jan. 2019.

APPENDIX A

DERIVATION OF BAYESIAN-ENTROPY POSTERIOR GIVEN MOMENT CONSTRAINT

The definition of entropy is given as:

$$S = -\int dxd\theta\, p(x,\theta) \log \frac{p(x,\theta)}{\mu(x,\theta)}$$

where $p(\cdot)$ is the target distribution and $\mu(\cdot)$ is the prior distribution function. x represents the observable quantity and θ is the uncertainty parameter. The notation $\int dxd\theta$ means the definite integral of the variable's domain.

In our case the x represents the jth feature and theta represents class. Maximize the entropy gives us the new joint distribution $p(f_j, C)$ in relation to the old one $\mu(f_j, C)$.

Maximize the entropy $S = -\int df_j dC p(f_j, C) \log \frac{p(f_j, C)}{\mu(f_j, C)}$ under the following constraint:

1. The normalization constraint for joint distribution:

$$c_1 : \int df_j dC p(f_j, C) = 1$$

2. The moment constraint for likelihood function:

$$c_2 : \int df_j\, p(f_j \mid C = C_i) g(f_j) = G_i$$

Multiply p(C) on both side:

$$\int df_j\, p(f_j \mid C = C_i)\, p(C = C_i) g(f_j) = G_i\, p(C = C_i)$$

Note: for each class label C there could be one of this constraint.

3. The normalization constraint for the likelihood function

$$c_3 : \int df_j\, p(f_j \mid C = C_i) = 1$$

Multiply p(C) on both side:

$$\int df_j \, p(f_j \mid C = C_i)\, p(C = C_i) = p(C = C_i)$$

Note: for each class label C there could be one of this constraint.

Form the Lagrange function. $\alpha, \beta(C), \gamma(C)$ are Lagrange multipliers. $\beta(C), \gamma(C)$ are function of C.

$$\begin{aligned}\mathcal{L} = &-\int df_j dC p(f_j, C) \log \frac{p(f_j, C)}{\mu(f_j, C)} + \alpha[\int df_j dC p(f_j, C) - 1] \\ &+ \int dC \beta(C)[\int d f_j \, p(f_j, C) g(f_j) - G_i p(C)] \\ &+ \int dC \gamma(C)[\int d f_j \, p(f_j, C) - p(C)]\end{aligned}$$

The variation of the Lagrangian function is 0 $\delta \mathcal{L} = 0$ gives:

$$\frac{\partial \mathcal{L}}{\partial p} = \int df_j dC[-\log \frac{p(f_j, C)}{\mu(f_j, C)} - 1 + \alpha + \beta(C) g(f_j) + \gamma(C)] = 0$$

The above equation satisfies for any $p(f_j, C)$ which means:

$$-\log \frac{p(f_j, C)}{\mu(f_j, C)} - 1 + \alpha + \beta(C) g(f_j) + \gamma(C) = 0$$

This gives: $$p(f_j, C) = \mu(f_j, C) e^{-1+\alpha} e^{\beta(C) g(f_j)} e^{\gamma(C)} = \frac{\mu(f_j, C) e^{\beta(C) g(f_j)} e^{\gamma(C)}}{z}$$

where $$z = 1 \Big/ e^{-1+\alpha} = \int df_j dC \mu(f_j, C) e^{\beta(C) g(f_j) + \gamma(C)} = \sum_C \int df_j \mu(f_j, C) e^{\beta(C) g(f_j) + \gamma(C)}$$

We assume that the prior does not change: $p(C) = \mu(C)$

Hence the expression for likelihood function:

$$p(f_j \mid C) = \frac{\mu(f_j \mid C) e^{\beta(C) g(f_j) + \gamma(C)}}{z}$$

Back substitute into the two constraint c2 and c3:

$$\frac{\int df_j \mu(f_j \mid C) e^{\beta(C) g(f_j)} e^{\gamma(C)} g(f_j)}{z} = G_i \tag{A.1}$$

$$\frac{\int df_j \mu(f_j \mid C) e^{\beta(C) g(f_j)} e^{\gamma(C)}}{z} = 1 \tag{A.2}$$

(A.1)/(A.2):
$$\frac{e^{\gamma(C)} \int df_j \mu(f_j \mid C) e^{\beta(C) g(f_j)} g(f_j)}{e^{\gamma(C)} \int df_j \mu(f_j \mid C) e^{\beta(C) g(f_j)}} = G \tag{A.3}$$

Now let's focus on a specific case where the constraint on likelihood function is a first order moment, $g(f_j) = f_j$.

The old likelihood function is a normal distribution: $\mu(f_j \mid C) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(f_j - \mu)^2}{2\sigma^2}}$

Substitute into (A.3) we get:

$$\frac{\frac{e^{-\frac{\mu^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} e^{-\frac{1}{2\sigma^2} f^2 + (\frac{\mu}{\sigma^2} + \beta) f} f df}{\frac{e^{-\frac{\mu^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} e^{-\frac{1}{2\sigma^2} f^2 + (\frac{\mu}{\sigma^2} + \beta) f} df} = G_i \tag{A.4}$$

From basic calculus we have:

$$\int_{-\infty}^{\infty} e^{-ax^2-bx}dx = \sqrt{\frac{\pi}{a}}e^{b^2/4a}$$

$$\int_{-\infty}^{\infty} xe^{-ax^2-bx}dx = -\frac{\sqrt{\pi}}{2a^{3/2}}e^{b^2/4a}$$

Let: $$a = \frac{1}{2\sigma^2}, b = -(\frac{\mu}{\sigma^2} + \beta)$$

Substitute into (4): $$\frac{-\frac{\sqrt{\pi}b}{2a^{3/2}}e^{\frac{b^2}{4a^2}}}{\sqrt{\frac{\pi}{a}}e^{\frac{b^2}{4a^2}}} = -\frac{b}{2a} = G_i$$

Solve for β: $$\beta = \frac{G_i - \mu}{\sigma^2}$$

Substitute β into (2): $$\frac{e^{\gamma(C)}}{z} = \frac{1}{\int d f_j \mu(f_j \mid C)e^{\beta(C)g(f_j)}}$$

$$I = \int d f_j \mu(f_j \mid C)e^{\beta(C)g(f_j)} = \frac{e^{-\frac{\mu^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\int_{-\infty}^{\infty} e^{-\frac{1}{2\sigma^2}f^2 + (\frac{\mu}{\sigma^2}+\beta)f}df$$

Again, we let: $$a = \frac{1}{2\sigma^2}, b = -(\frac{\mu}{\sigma^2} + \beta) = -\frac{G_i}{\sigma^2}$$

So: $$I = \frac{e^{-\frac{\mu^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\sqrt{\frac{\pi}{a}}e^{\frac{b^2}{4a^2}} = e^{\frac{G_i^2 - \mu^2}{2\sigma^2}}$$

Hence: $$\frac{e^{\gamma(C)}}{z} = e^{\frac{\mu^2 - G_i^2}{2\sigma^2}}$$

The complete form of the new likelihood function:

$$p(f_j \mid C) = \mu(f_j \mid C) e^{\frac{G_i - \mu}{\sigma^2} f_j} e^{\frac{\mu^2 - G_i^2}{2\sigma^2}} \tag{A.5}$$

where μ and σ are the mean and standard deviation of $\mu(f_j \mid C)$.

APPENDIX B

DERIVATION FOR BAYESIAN-ENTROPY POSTERIOR FOR REGRESSION COEFFICIENT GIVEN VALUES AND DERIVATIVES CONSTRAINTS

The distribution function for the regression coefficient from the Bayesian-Entropy linear regression (BELR) can be written as:

$$p_{en}(\boldsymbol{\beta}) \propto \exp\left[-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_n)^{\mathrm{T}}\boldsymbol{\Sigma}_n^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_n)\right]\exp\left[\sum_{i=1}^{M}\eta_i\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i\right]\exp\left[\sum_{i=M+1}^{M+N}\eta_i\boldsymbol{\beta}^{\mathrm{T}}\left(\frac{\partial\mathbf{x}}{\partial x_{d_i}}\bigg|_{\mathbf{x}=\mathbf{x}_i}\right)\right]$$

$$= \exp\left[-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_n)^{\mathrm{T}}\boldsymbol{\Sigma}_n^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_n)+\sum_{i=1}^{M}\eta_i\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i+\sum_{i=M+1}^{M+N}\eta_i\boldsymbol{\beta}^{\mathrm{T}}\left(\frac{\partial\mathbf{x}}{\partial x_{d_i}}\bigg|_{\mathbf{x}=\mathbf{x}_i}\right)\right] \tag{B.1}$$

For the ease of illustrating the derivation, we now take out the term inside the exponential and omit the constant -1/2 as *H*. Next we expend *H* and combine like terms:

$$H = (\boldsymbol{\beta}-\boldsymbol{\mu}_n)^{\mathrm{T}}\boldsymbol{\Sigma}_n^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_n)-2\sum_{i=1}^{M}\eta_i\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i-2\sum_{i=M+1}^{M+N}\eta_i\boldsymbol{\beta}^{\mathrm{T}}\left(\frac{\partial\mathbf{x}}{\partial x_{d_i}}\bigg|_{\mathbf{x}=\mathbf{x}_i}\right)$$

$$= \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Sigma}_n^{-1}\boldsymbol{\beta}-2\left[\boldsymbol{\mu}_n^{\mathrm{T}}\boldsymbol{\Sigma}_n^{-1}\boldsymbol{\beta}+\sum_{i=1}^{M}\eta_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}+\sum_{i=M+1}^{M+N}\eta_i\left(\frac{\partial\mathbf{x}}{\partial x_{d_i}}\bigg|_{\mathbf{x}=\mathbf{x}_i}\right)^{\mathrm{T}}\boldsymbol{\beta}\right]+\boldsymbol{\mu}_n^{\mathrm{T}}\boldsymbol{\Sigma}_n^{-1}\boldsymbol{\mu}_n$$

$$= \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Sigma}_n^{-1}\boldsymbol{\beta}-2\left[\boldsymbol{\mu}_n^{\mathrm{T}}+\sum_{i=1}^{M}\eta_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Sigma}_n+\sum_{i=M+1}^{M+N}\eta_i\left(\frac{\partial\mathbf{x}}{\partial x_{d_i}}\bigg|_{\mathbf{x}=\mathbf{x}_i}\right)^{\mathrm{T}}\boldsymbol{\Sigma}_n\right]\boldsymbol{\Sigma}_n^{-1}\boldsymbol{\beta}+\boldsymbol{\mu}_n^{\mathrm{T}}\boldsymbol{\Sigma}_n^{-1}\boldsymbol{\mu}_n$$

The form for *H* resembles a quadratic form. By completing the square for *H*, we can have:

$$H = (\boldsymbol{\beta}-\boldsymbol{\mu}_{en})^{\mathrm{T}}\boldsymbol{\Sigma}_n^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_{en})-\boldsymbol{\mu}_{en}^{\mathrm{T}}\boldsymbol{\Sigma}_n^{-1}\boldsymbol{\mu}_{en}+\boldsymbol{\mu}_n^{\mathrm{T}}\boldsymbol{\Sigma}_n^{-1}\boldsymbol{\mu}_n \tag{B.2}$$

with

$$\boldsymbol{\mu}_{en} = \boldsymbol{\mu}_n^{\mathrm{T}}+\sum_{i=1}^{M}\eta_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Sigma}_n+\sum_{i=M+1}^{M+N}\eta_i\left(\frac{\partial\mathbf{x}}{\partial x_{d_i}}\bigg|_{\mathbf{x}=\mathbf{x}_i}\right)^{\mathrm{T}}\boldsymbol{\Sigma}_n \tag{B.3}$$

Substitute the rearranged expression for *H* into Eq. (B.1), the distribution function for **β** can be written as:

$$p_{en}(\boldsymbol{\beta}) \propto \exp\left[-\frac{1}{2}\left((\boldsymbol{\beta}-\boldsymbol{\mu}_{en})^{\mathrm{T}}\boldsymbol{\Sigma}_n^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_{en}) - \boldsymbol{\mu}_{en}{}^{\mathrm{T}}\boldsymbol{\Sigma}_n^{-1}\boldsymbol{\mu}_{en} + \boldsymbol{\mu}_n{}^{\mathrm{T}}\boldsymbol{\Sigma}_n^{-1}\boldsymbol{\mu}_n\right)\right]$$
$$\propto \exp\left[-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_{en})^{\mathrm{T}}\boldsymbol{\Sigma}_n^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_{en})\right]$$

The last two terms in Eq. (B.2) is a constant regarding $\boldsymbol{\beta}$ and is treated as part of the normalization constant for the distribution function. Hence, the Bayesian-Entropy posterior for $\boldsymbol{\beta}$ is another multivariate Normal distribution with $N(\boldsymbol{\mu}_{\mathrm{en}},\boldsymbol{\Sigma}_{\mathrm{n}})$.

The value and derivative constraints can be written as the expected value of the regression coefficient $\boldsymbol{\beta}$ as:

$$\begin{cases} E(\boldsymbol{\beta})^{\mathrm{T}}\mathbf{x}_i = y_i, & i = 1,\ldots,M \\ E(\boldsymbol{\beta})^{\mathrm{T}}\left(\dfrac{\partial \mathbf{x}}{\partial x_{d_j}}\bigg|_{\mathbf{x}=\mathbf{x}_j}\right) = dy_{j,d_j}, & j = M+1,\ldots M+N \end{cases}$$

The expected value for $\boldsymbol{\beta}$ can be replaced with the mean of the BE posterior $\boldsymbol{\mu}_{\mathrm{en}}$ given in Eq. (B.3). This will give a system of equations for the Lagrangian multipliers η_i's. Each parameter of the linear system is related to the given constraints:

$$\mathbf{B} - \mathbf{U}\boldsymbol{\Sigma}_n\mathbf{V}\boldsymbol{\eta} = \mathbf{Y} \tag{B.4}$$

where
$$\mathbf{B} = \left[\boldsymbol{\mu}_n{}^{\mathrm{T}}\mathbf{x}_1, \quad \cdots, \quad \boldsymbol{\mu}_n{}^{\mathrm{T}}\mathbf{x}_M, \quad \boldsymbol{\mu}_n{}^{\mathrm{T}}\left(\frac{\partial \mathbf{x}}{\partial x_{d_i}}\bigg|_{\mathbf{x}=\mathbf{x}_{M+1}}\right), \quad \cdots, \quad \boldsymbol{\mu}_n{}^{\mathrm{T}}\left(\frac{\partial \mathbf{x}}{\partial x_{d_i}}\bigg|_{\mathbf{x}=\mathbf{x}_{M+N}}\right)\right]^{\mathrm{T}}$$

$$\mathbf{U} = \left[\mathbf{x}_1{}^{\mathrm{T}}, \quad \cdots, \quad \mathbf{x}_M{}^{\mathrm{T}}, \quad \left(\frac{\partial \mathbf{x}}{\partial x_{d_{M+1}}}\bigg|_{\mathbf{x}=\mathbf{x}_{M+1}}\right)^{\mathrm{T}}, \quad \cdots, \quad \left(\frac{\partial \mathbf{x}}{\partial x_{d_{M+N}}}\bigg|_{\mathbf{x}=\mathbf{x}_{M+N}}\right)^{\mathrm{T}}\right]^{\mathrm{T}}$$

$$\mathbf{V} = \left[\mathbf{x}_1, \quad \cdots, \quad \mathbf{x}_M, \quad \left(\frac{\partial \mathbf{x}}{\partial x_{d_{M+1}}}\bigg|_{\mathbf{x}=\mathbf{x}_{M+1}}\right), \quad \cdots, \quad \left(\frac{\partial \mathbf{x}}{\partial x_{d_{M+N}}}\bigg|_{\mathbf{x}=\mathbf{x}_{M+N}}\right)\right]$$

$$\boldsymbol{\eta} = \left[\eta_1, \quad \cdots, \quad \eta_{M+N}\right]^{\mathrm{T}}$$

$$\mathbf{Y} = \left[y_1, \quad \cdots, \quad y_M, \quad dy_{M+1,d_{M+1}}, \quad \cdots, \quad dy_{M+N,d_{M+N}}\right]^{\mathrm{T}}$$

For interpreting Eq. (B.4), B can be regarded as the Bayesian posterior mean times the constraint terms, the second term is the constraint terms times the vector of Lagrangian coefficient. On the right-hand-side is the constraint values. So, the solution for η_i's is:

$$\boldsymbol{\eta} = \left(\mathbf{U}\boldsymbol{\Sigma}_n\mathbf{V}\right)^{-1}\left(\mathbf{B} - \mathbf{Y}\right) \tag{B.5}$$