Estimating Feasibility Using Multiple Surrogates and ROC Curves

Anirban Chaudhuri*
University of Florida, Gainesville, Florida, 32601

Rodolphe Le Riche†
École Nationale Supérieure des Mines de Saint-Étienne, Saint-Étienne, France
and CNRS LIMOS UMR 6158

Mickael Meunier‡
Snecma SAFRAN, Villaroche Centre, France
CAE Methods and Tools Department

Constraint optimization aims at finding optimum points that satisfy equality or inequality constraints. An important part of constraint optimization is to estimate the feasibility of a point to be added in the next optimization cycle. This is especially evident in real-world problems which have multiple constraints with a very small, disconnected feasible space. The key issue, before seeking optimality, is to find a point in the feasible region. In this work we propose a family of methods for estimating feasibility at any new point in the design space using only the information from an initial design of experiment (DOE) when constraint calculations are computationally expensive, making the use of surrogates imperative. The method does not require additional resources and it is not limited to any particular choice of surrogate. Three different ways of predicting feasibility are described, where the choice of the DOE and surrogate uncertainties are taken into account through cross-validation and ROC curves. A way for combining feasibility predictions of multiple surrogates from their correlation and their confidence is also presented. These methods are compared using 2 analytic functions which have very small disconnected feasible regions.

Nomenclature

$g(x)$	=	True constraint value at point x
$PF(x)$	=	Probability of Feasibility at point x
$\hat{g}(x)$	=	Surrogate constraint prediction at point x
$s(x)$	=	Prediction error at point x
$\hat{s}(x)$	=	Surrogate prediction of error at point x
n	=	Number points in initial DOE
N_{CV}	=	Number of cases employed for each point selected in cross validation for error surrogate
N_S	=	Number of surrogates
n_g	=	Number of constraints
FPF	=	Feasibility Prediction Function
P	=	Positives or Feasible
N	=	Negatives or Infeasible
TP	=	True Positives
TN	=	True Negatives
FP	=	False Positives
FN	=	False Negatives
TPR	=	True Positive Rate
FPR	=	False Positive Rate
$T_{\hat{g}(x)}$	=	Threshold cut-off value on $\hat{g}(x)$
$T_{PF(x)}$	=	Threshold cut-off value on $PF(x)$

* Graduate Research Assistant, Mechanical & Aerospace Engineering, a.chaudhuri@ufl.edu, AIAA Student Member
† CNRS permanent research associate, Henri Fayol Institut, leriche@emse.fr
‡ Aerospace Engineer, CAE Methods and Tools Dept., Snecma SAFRAN, mickael.meunier@snecma.fr

w = Weight associated with each surrogate
V = Vote on feasibility of a point by a surrogate
V_M = Vote on feasibility of a point by a combination of multiple surrogates
x_{new} = Points where feasibility is predicted

I. Introduction

Constrained optimization aims at finding solutions in a design space that maximizes a performance function while satisfying equality or inequality constraint functions. Recent contributions have dealt with global constrained optimization[1-9]. But, in practical optimization problems, there are often multiple constraints and the feasible space, if any, is a small, disconnected fraction of the design space. Then the first issue to be addressed is that of finding a feasible design point.

This work concentrates on predicting the feasibility of any new point based on an initial design of experiments (DOE) when constraint calculations are computationally expensive. Such a feasibility function is the basic building block of constraint satisfaction procedures. We focus on the decision step about feasibility of a design point without using any additional resources (constraints evaluations) beyond the given DOE. Maximum amount of knowledge needs to be gained from available data. Since we are dealing with expensive function evaluations the use of surrogates (also known as, meta-models or response surfaces) becomes imperative. Our feasibility measures are not limited by any particular choice of surrogates.

Two techniques underlie our contributions in this article. First, cross-validation[10] is used to estimate the influence of the DOE choice on the predicted feasibility. Second, in order to account for the uncertainties associated with surrogate predictions, we use thresholds instead of directly trusting the surrogate predictions. Receiver Operating Characteristic (ROC)[11] curves based on cross-validation allow to calculate the thresholds. Based on these ingredients, three different ways of predicting feasibility are developed and compared. A method is then proposed that additionally combines the feasibility predictions of several surrogates based on their correlation and a confidence stemming from ROC curves.

The rest of the paper is arranged as follows. Section II describes the general principle for predicting feasibility used in this work. Section III focuses on how Receiver Operating Characteristic curves allow quantifying the quality of feasibility prediction and tuning thresholds. Section IV explains a way to combine feasibility predictions of multiple surrogates. Section V describes the analytical functions used to test the method. Section VI discusses the results. Concluding remarks for this research make Section VII.

II. Predicting Feasibility

We address the feasibility question where we need to decide whether a constraint function, $g(x)$, is being violated or not at a point x in the design space S (for all $x \in S$, decide if $g(x) \leq 0$ *(feasible)* or $g(x) > 0$ *(infeasible)*). The case of multiple constraints, $g_i(x)$ *for* $i = 1,\ldots,n_g$, can come down to single constraint formulation through $g(x) = \max_{i=1,..,n_g} g_i(x)$. In this work all constraints are set up as $g_i(x) \leq 0$.

As we are dealing with computationally expensive constraint functions, $g(x)$, we are led to the use of surrogates. This section presents our approach to predicting feasibility.

A. Surrogate for constraint

An initial set of data points or design of experiment (DOE) is generated and the true value of constraints at those points is evaluated. Then a surrogate is fit through these values. We can use any regression surrogate like CART (Classification and Regression Trees[12]), Kriging[13], Support Vector Regression (SVR)[14,15], Radial Basis Neural Networks (RBNN)[16], etc. This surrogate provides an estimate of the value of constraint, $\hat{g}(x)$ at any point in the design space. The easiest way to predict feasibility is to just use the surrogate prediction to check if the constraints are satisfied. But, as will be argued in this paper, these predictions can be improved, in particular for DOEs with few feasible points.

B. Surrogate for error in constraint prediction

Since we are using a surrogate prediction of the constraint at any point, there is some error associated with that prediction due to the finite set of points in the DOE and the choice of surrogate. The quantification of this error in surrogate prediction of constraint is done by building an error surrogate using cross-validation[10].

In most real design applications, we have limited resources and the idea here is to predict feasibility using only the initial set of points provided. We use cross-validation by leaving out 2 points at a time and fit a surrogate for

constraint violations through the rest of the points in order to get an estimate at the points left out. This leave-two-out strategy was chosen because leaving one point out would not be sufficient to estimate a local error (there would be only one measure of the error at each point of the DOE) and leaving three or more points out can change too much small DOEs. This was also substantiated by tests.

The total number of combinations of DOE for leaving out 2 points out of *n* total points is C_2^n. Each point in the DOE can be left out C_1^{n-1} = *(n-1)* times. So we can obtain *(n-1)* predictions at each point of the DOE. For large DOEs, this number can become large and surrogate learning is computationally too expensive. To handle these situations, we fix our computational budget at leaving out each point not more than N_{CV} times (this number can be user-defined based on the available resources). In order to select N_{CV} out of the possible *(n-1)* cases we use bootstrapping[17]. The predicted constraint values are assumed to be normally distributed and their true mean is assumed to be the true constraint value at that point (which is already known as it is an initial data point in the DOE). The error in the constraint prediction is the standard deviation of this distribution which is estimated by using Maximum Likelihood Estimation (MLE). More details about the implementation of MLE can be found in Appendix A. This gives the value of errors (*s(x)*) in surrogate prediction of constraint at the data points.

A surrogate is fitted through the *s(x)* values at the data points in order to estimate the error, $\hat{s}(x)$ at any point in the design space. In this work we use the same type of surrogate to fit the error and the constraints. Other surrogate choices for the error are also possible. Figure 1 summarizes the process used to find $\hat{s}(x)$.

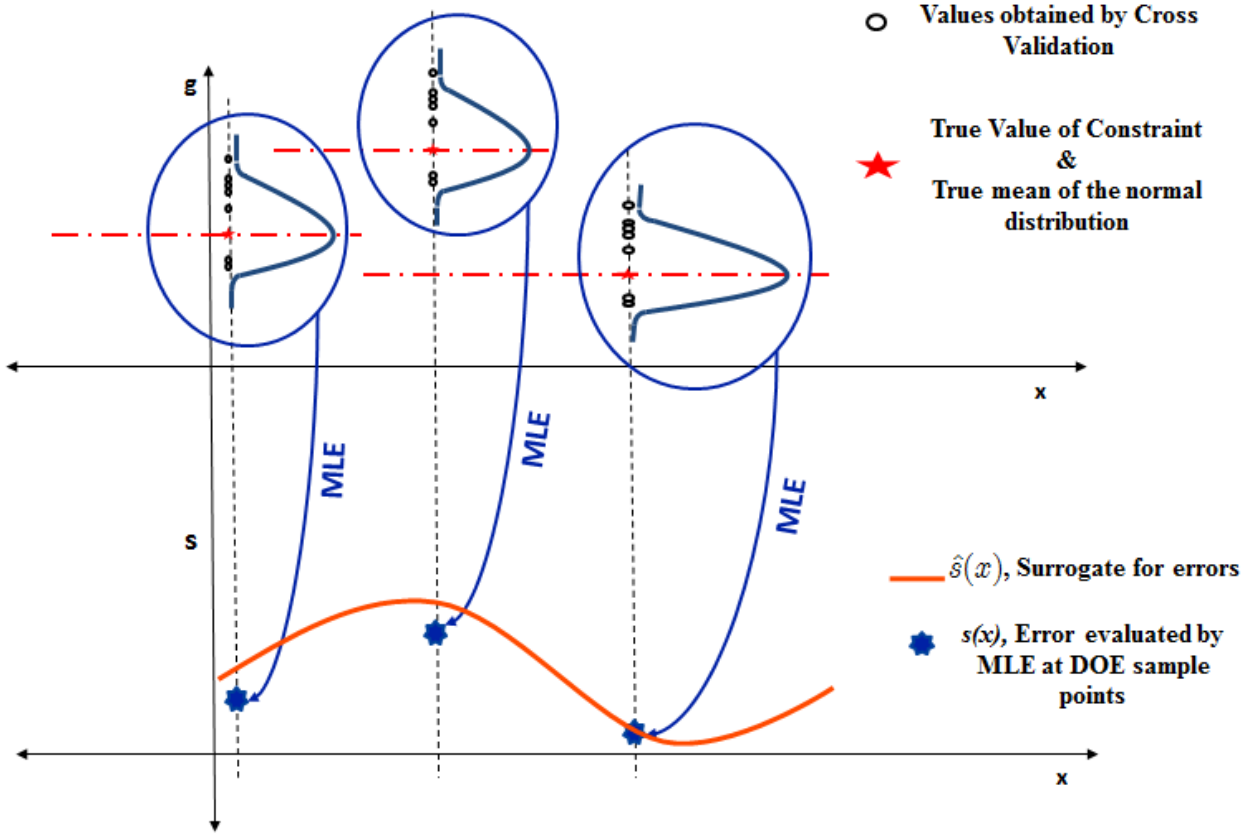

Figure 1. Method for building surrogates of the constraint prediction error, $\hat{s}(x)$, at any point *x* in the design space.

C. Probability of Feasibility (PF)

A measure of feasibility used in this work is the probability of being feasible or Probability of Feasibility (*PF*)[2]. For any new test point, *x*, the probability of feasibility can be computed by using Equation (1), where a point is feasible if the associated constraint is negative, $g(x) \leq 0$. $\hat{g}(x)$ is the surrogate constraint prediction explained in Section II.A and $\hat{s}(x)$ is the value of the error in the constraint prediction described in Section 0. Φ is the cumulative density function of a standard centered Gaussian law.

$$PF(x) = \Phi\left(\frac{0 - \hat{g}(x)}{\hat{s}(x)}\right) \tag{1}$$

D. Methods for predicting feasibility

The functions used to predict feasibility, dubbed *Feasibility Prediction Functions (FPF)*, are: (1) Constraint prediction, $\hat{g}(x)$, and (2) Probability of feasibility, $PF(x)$.

In this work, three ways of predicting feasibility are compared. Any point in the design space is predicted as feasible if:

(a) *Method 1*: $\hat{g}(x) \leq 0$

(b) *Method 2*: $\hat{g}(x) \leq Threshold\ on\ \hat{g}(x), T_{\hat{g}(x)}$

(c) *Method 3*: $PF(x) \geq Threshold\ on\ PF(x),\ T_{PF(x)}$

We propose to use thresholds on feasibility prediction functions (FPF) in order to accommodate unbalanced DOEs, where typically feasible points are seldom, and deal with the uncertainties in the predictions. The thresholds T are meta-parameters added to the feasibility prediction functions to create feasibility classifiers. The process for deciding feasibility is explained in Figure 2 below. The calculation of the optimum threshold values is discussed in Section III.

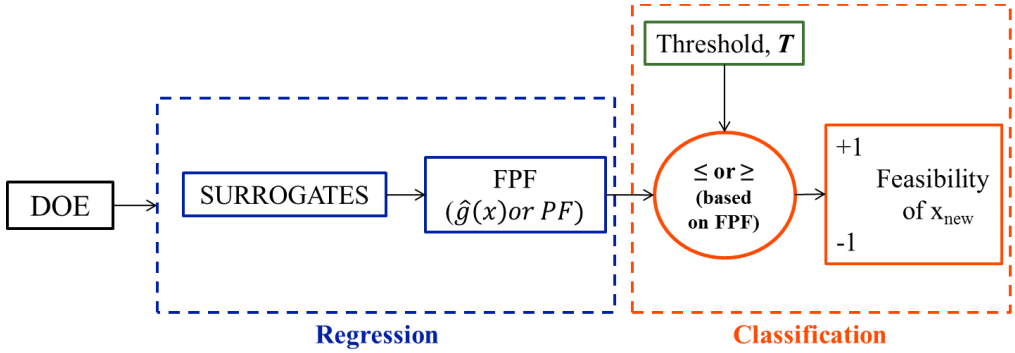

Figure 2. Method for deciding feasibility at a new point in the design space, x_{new}.

III. Receiver Operating Characteristic (ROC) curve and thresholds

The following section outlines how ROC curves are employed to quantify the feasibility thresholds. An additional measure of surrogate confidence is introduced which will be needed later as a weight when combining several surrogates.

A. ROC curves

Relative Operating Characteristic (ROC)[11,18] are two-dimensional graphs that analyze the performance of a binary classifier system with a parameter to be tuned. It is a curve describing the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) over a given set of labeled samples (into positives or negatives) as the classification parameter changes. In binary classification problems, such as our feasibility prediction problem, there could be two predicted outcomes which can be labeled as positive (*P*) or negative (*N*). Here, the positive event is associated to a feasible design. The four possible outcomes from a binary classifier are (also shown in Table 1):

(i) True Positive (TP): If the outcome from a prediction is *P* and the actual value is also *P*.

(ii) False Positive (FP): If the outcome from a prediction is *P* and the actual value is *N*.

(iii) True Negative (TN): If the outcome of prediction is *N* and the actual value is *N*.

(iv) False Negative (FN): If the outcome of prediction is *N* and the actual value is *P*.

Table 1. Outcomes of a binary classifier

		Actual Outcome	
		P	**N**
Predicted Outcome	**P**	**TP**	**FP**
	N	**FN**	**TN**

TPR is the fraction of True Positives (TP) out of all the positives and FPR is the fraction of False Positives (FP) out of all the negatives as shown in Equations (2) and (3), respectively.

$$True\ Positive\ Rate, TPR = \frac{TP}{TP + FN} \tag{2}$$

$$False\ Positive\ Rate, FPR = \frac{FP}{FP + TN} \tag{3}$$

In this work we use regression instead of classification. To convert the regression prediction at a certain point into a classification as feasible or infeasible solution point, a threshold is set. This threshold is varied to get a continuous ROC curve as can be seen in Figure 3 where *T* represents threshold values. On the no-discrepancy line the TPR and FPR values are equal suggesting that the classification is totally random. The further away from the no-discrepancy line the (TPR, FPR) point lies, the better is the classification. The working assumption here is that there is at least one feasible point in the initial DOE (so that the TPR can be defined). A method to handle DOEs without any feasible points to start with is provided in Appendix B.

The ROC curve compares the predicted feasibility decided by either *Method 2* or *Method 3* mentioned in Section II.D, with the true feasibility of the points. The threshold values are set at each value of the FPF used, varying from minimum to maximum value of the FPF, and a value of TPR and FPR is found for each such threshold. These constitute the points in the ROC curve.

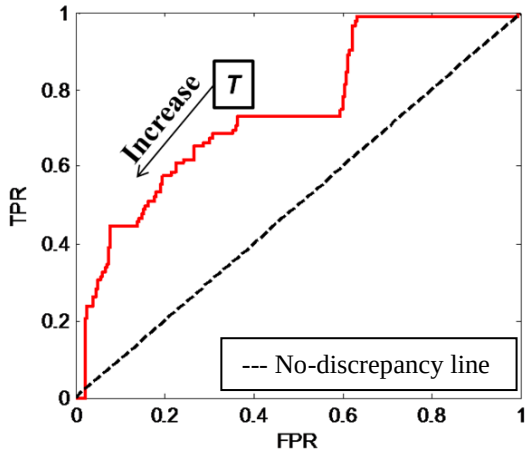

Figure 3. ROC curve built by continuously changing thresholds.

B. Building ROC curve using DOE points

ROC curves are usually built by using new test points. But with the current expensive simulation assumption, we don't have the liberty to test new points. So in order to make the ROC curve we use the already sampled DOE points with a one point cross-validation method.

We first fit a surrogate for error in constraint predictions as described in Section II.B. Then one point from the initial DOE, x_T, is left out to be the test point and a surrogate for constraint (Section II.A) is fit for the rest of the points. This provides $\hat{g}(x_T)$ and $\hat{s}(x_T)$ which allow the calculation of find $PF(x_T)$ at x_T. This is repeated for every point in the initial DOE to get $PF(x)$ and $\hat{g}(x)$ values for the n points in the initial DOE. These values of the FPF are used to build the ROC curve from which stems the optimum value of threshold and weights associated with a surrogate as

described in the next section. ROC curves are only needed for methods with a threshold, i.e. the 2nd and 3rd methods.

C. Optimum threshold and weights for each surrogate

An optimum threshold value, $T^{Optimum}$, can be decided from the ROC curves. The point which is farthest from the no-discrepancy line in the ROC curve makes the best classification for that surrogate. $T^{Optimum}$ lies between the thresholds classifying the point farthest from the no-discrepancy line, $T_{farthest}$ and the previous one, $T^{previous}_{farthest}$ as shown in Equation (4). More precisely, in order to account for the fact that most often feasible points will be under-represented in the DOE, the optimum threshold value will be chosen as in Equation (5) (ε is taken equal to 0.001 here).

$$T^{previous}_{farthest} \leq T^{Optimum} \leq T_{farthest} \tag{4}$$

$$T^{Optimum} = T^{previous}_{farthest} + \varepsilon \tag{5}$$

The weight, w_i, associated with each surrogate is the maximum distance from the no-discrepancy line. This distance can be interpreted as the amount of confidence we have in the surrogate. For example, an idealized classifier has FPR=0 and TPR=1 which is the farthest possible point from the no-discrepancy line. Figure 4 shows the method of deciding the optimum threshold and weight for a surrogate and for a particular FPF. In case of *Method 2*, the formulation actually used for generating ROC curves is $-\hat{g}(x) \geq -T_{\hat{g}(x)}$, which is the same as mentioned in Section II.D multiplied by (-)1 on both sides. This makes the threshold, T, which is being increased continuously to build the ROC curve (Figure 3), equal to $-T_{\hat{g}(x)}$, thus preserving the form in which ROC curves are solved to get optimum thresholds.

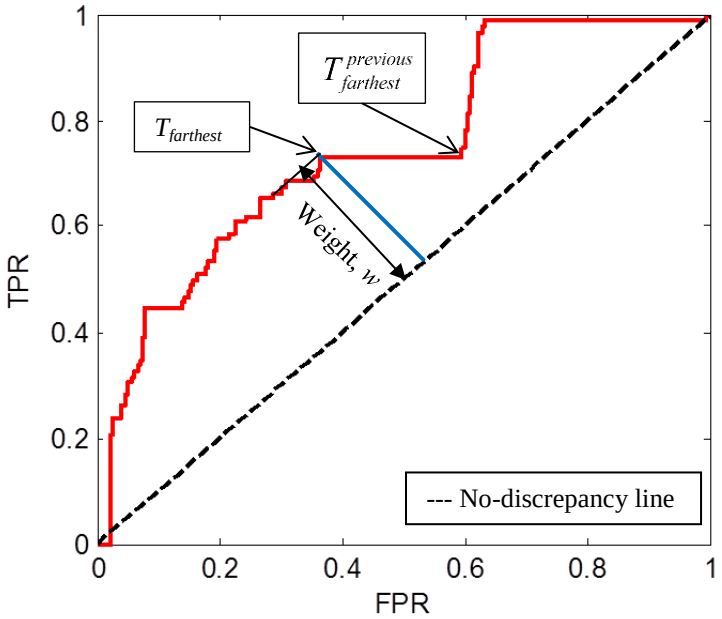

Figure 4. Illustration of the optimum threshold and weight for a surrogate and for a particular FPF.

IV. Combining feasibility predictions from multiple surrogates

When confronted to a new constraint optimization problem, it is not known which surrogate will perform the best. In order to increase the feasibility prediction accuracy and not being held back due to a bad surrogate choice, we propose a new approach for combining the feasibility predictions of multiple surrogates (number of surrogates, N_S).

This is done by first finding the optimum threshold value for each surrogate as described in Section III. This threshold value for the FPF is used to give a vote, V_i^k, as to whether the k^{th} DOE point is feasible or not according to surrogate or classifier *i*. V_i^k is given by Equation (6).

$$V = \begin{cases} -1 & \text{if not feasible} \\ 1 & \text{if feasible} \end{cases} \tag{6}$$

Second, the correlation between any two surrogates is taken into account when combining the classifiers. The correlation matrix, $[\rho_{ij}]$, is made of the correlation coefficients between the vector of votes on DOE points, V_i and V_j, of any two classifiers, *i* and *j*. Third, the weight, w_i, associated with each classifier *i* is calculated as explained in Section III and normalized to make, $\bar{w}_i$ by,

$$\bar{w}_i = \frac{w_i}{\sum_{i=1}^{N_S} w_i} \tag{7}$$

The combined value of the votes for multiple surrogates, V_M, is finally given by,

$$V_M = \{V_1, V_2, ..., V_{N_S}\}[\rho_{ij}]^{-1} \begin{Bmatrix} \bar{w}_1 \\ \bar{w}_2 \\ ... \\ \bar{w}_{N_S} \end{Bmatrix} \tag{8}$$

Some basic effects of this combination formula are described below for $N_S=2$ at some values of ρ:

- $\rho \to 1$: $V_M \to Average(V_1,V_2)$ as $\bar{w}_1 \to \bar{w}_2$
- $\rho \to 0$: $V_M \to \bar{w}_1 V_1 + \bar{w}_2 V_2$
- $\rho \to -1$: $\begin{cases} V_M \text{ strongly agrees if } V_1 = V_2 \text{ (agrees)} \\ V_M \to 0 \text{ if } V_1 = -V_2 \text{ (disagrees)} \end{cases}$

Then V_M is used to build a ROC curve and find the threshold for V_M in the same way as explained in Section III. Now the method of combination can be employed to predict the feasibility at any new point in the design space, x_{new}. First, votes of feasibility, V_i (Equation (6)), given by each of the N_S surrogates to x_{new} is obtained and substituted in Equation (8) with the same $[\rho_{ij}]$ and $\bar{w}_i$ as above to get V_M at x_{new}. After, threshold value for V_M is used to classify whether x_{new} is feasible or not. This process is illustrated in Figure 5. The combination of surrogates has been implemented for both *$\hat{g}$(x) (Method 2)* and *PF(x) (Method 3)*.

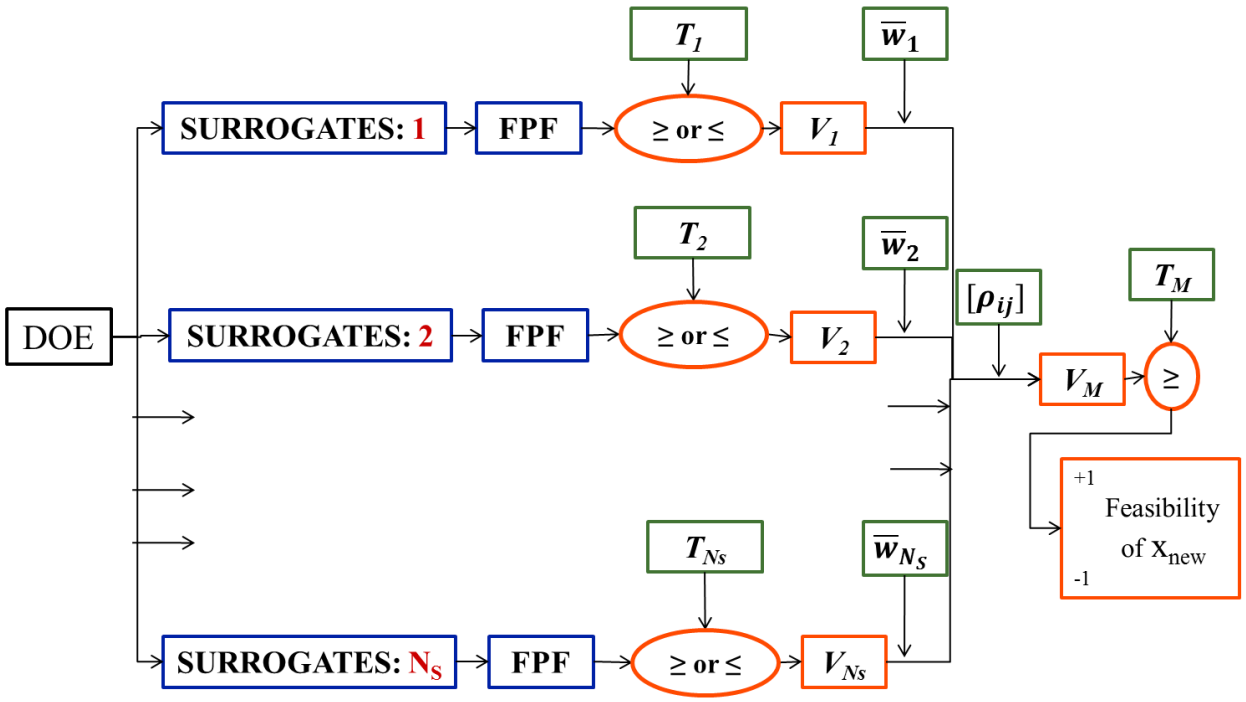

Figure 5. Flowchart for combining feasibility predictions of N_S surrogates.

V. Analytical examples

The test problems used in this work are the "New Branin" and "G9" problems. The New Branin problem is two dimensional so that the steps in the algorithm can be easily visualized. New Branin is not a very easy problem to solve as it has small disconnected regions of feasibility (~8% feasible space). Figure 6 illustrates this example. The feasible regions are the ones bounded by the black curves. We would like to effectively put points inside these black curves. Equation (9) shows the formulation of this problem.

$$\min_x f(x) = -(x_1 - 10)^2 - (x_2 - 15)^2$$

$$\text{such that : } g(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 5 \le 0 \tag{9}$$

$$-5 \le x_1 \le 10$$

$$0 \le x_2 \le 15$$

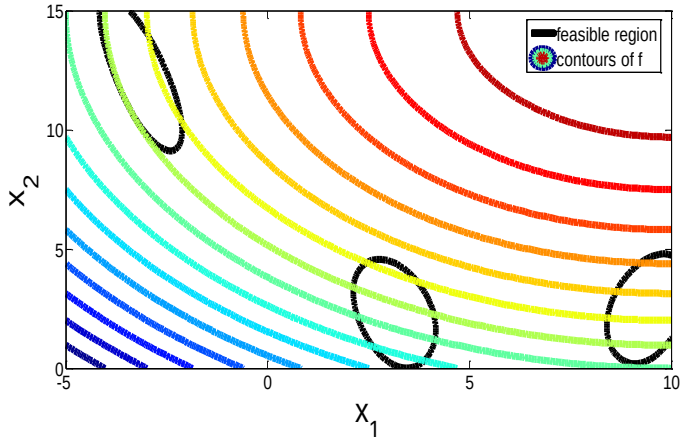

Figure 6. New Branin test problem.

The G9 problem has 7 design variables and 4 constraints (~7% feasible space). So it is a considerably more complex problem as compared to New Branin . The formulation of G9 is given below.

$$\begin{aligned}
&\min_x f(x) = (x_1-10)^2 + 5(x_2-12)^2 + x_3^4 + 3(x_4-11)^2 + \ldots \\
&\qquad 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \\
&such\,that, g_1(x) = 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 - 127 \le 0 \\
&\qquad g_2(x) = 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282 \le 0 \\
&\qquad g_3(x) = 23x_1 + x_2^2 + 6x_6^2 - 8x_7 - 196 \le 0 \\
&\qquad g_4(x) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0 \\
&-5 \le x_i \le 5 \; for\, i = 1, 2, \ldots, 7.
\end{aligned} \tag{10}$$

All the constraints are combined together to check for feasibility as,

$Maximum(g_i) \le 0$ where, $i = 1,..4$.

VI. Results and Discussion

The results for New Branin and G9 compare the 3 different methods of predicting feasibility (Section II.D). For the New Branin problem CART, SVR and Kriging (KRG) are used as the surrogates for all the methods while for G9, CART and KRG have been used. The number of points in the initial DOE, *n*, is 30 for New Branin and 70 for G9. In order to check the efficiency of the methods in predicting feasibility, we use 2500 new test points and 10000 new test points for New Branin and G9, respectively, and see how many are predicted correctly. The efficiency is judged to be good based on the point denoted by (TPR, FPR) being above the no-discrepancy line and farthest away from it. To average out the influence of design of experiments, 50 different DOEs are created and the results are presented as their mean.

The results for all the methods of predicting feasibility for a particular DOE with 5 feasible points to start with, for the New Branin test problem is presented to give an idea about how the method progresses. The ROC curves using only the DOE points for the different surrogates and their combination for *Method* 2 and for *Method 3* is shown in Figure 7. It can be seen from this that the Kriging individually and the combination for *Methods 2 & 3* show equally good performance (high weights). The optimum threshold values and the weights or the maximum distance from the no-discrepancy line are given in Table 2. The non-zero values of threshold for *Method* 2 indicates that there is compensation for error in surrogate prediction which also includes the fact that we do not have too many feasible points to start with. A similar computation is done for all DOEs.

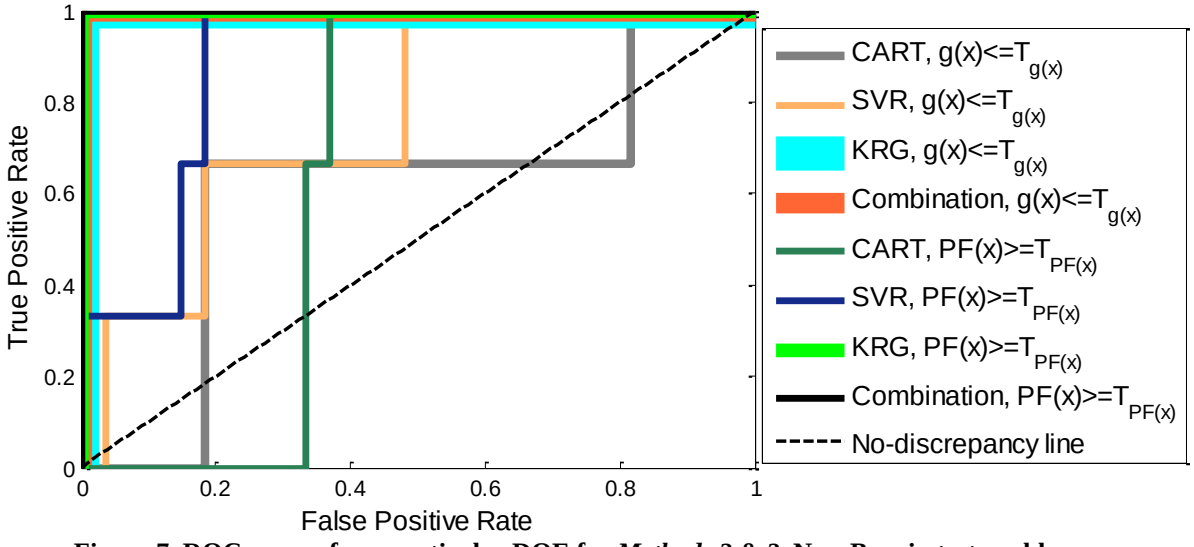

Figure 7. ROC curves for a particular DOE for *Methods 2 & 3*, New Branin test problem.

Table 2. Optimum threshold values and weights associated with each case for Methods 2 & 3 for a particular DOE for New Branin problem.

Surrogate used	Optimum threshold value	Weight (w)
Method 2: $\hat{g}(x) \leq T_{\hat{g}(x)}$		
CART	23.14	0.34
SVR	36.45	0.37
KRG	4.17	0.707
Combination	-0.32 (for V_M)	0.707
Method 3: $PF(x) \geq T_{PF(x)}$		
CART	0.10	0.45
SVR	0.08	0.58
KRG	0.001	0.707
Combination	-0.08 (for V_M)	0.707

The results showing the efficiency of the different methods of predicting feasibility for New Branin problem are given in Table 3 using 2500 test points with 207 feasible points (P) and 2293 infeasible points (N). The distance from no-discrepancy line for the predictions is highest for *Methods 1 & 2* using Kriging as the surrogate. But this is possible because Kriging fits this function very well. If some other surrogate was used, then results are not good as can be seen from results for CART and SVR using *Methods 1 & 2*. In the case of *Method 3*, Kriging individually performs the best. The combination also performs almost equally well using *Methods 2 & 3* both. The good thing about the combination is that it is not affected much by a bad surrogate like CART or SVR in this case and the idea is that the feasibility prediction is robust even if you don't know which one is the best surrogate to start with. This will become clearer from the results for G9 problem presented next.

Table 3. Efficiency in predicting feasibility of 2500 test points (P=207, N=2293) for New Branin problem, results from 50 DOEs.

Surrogate used	TPR		FPR		Distance from no-discrepancy line (maximum = $\sqrt{2}/2$)	
	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation
Method 1: $\hat{g}(x) \leq 0$						
CART	0.03	0.07	0.01	0.01	0.01	0.04
SVR	0.44	0.14	0.05	0.03	0.28	0.09
KRG	0.96	0.05	0.008	0.006	**0.67**	0.03
Method 2: $\hat{g}(x) \leq T_{\hat{g}(x)}$						
CART	0.49	0.28	0.35	0.20	0.10	0.20
SVR	0.83	0.25	0.25	0.16	0.41	0.15
KRG	0.98	0.15	0.06	0.05	**0.65**	0.04
Combination	0.94	0.12	0.10	0.12	**0.60**	0.10
Method 3: $PF(x) \geq T_{PF(x)}$						
CART	0.66	0.27	0.47	0.18	0.13	0.18
SVR	0.84	0.24	0.24	0.12	0.42	0.13
KRG	0.95	0.11	0.08	0.05	**0.62**	0.08
Combination	0.94	0.09	0.11	0.07	**0.59**	0.07

The results comparing the efficiency of the different feasibility prediction methods for G9 problem are given in Table 4 using 10,000 test points with 723 feasible points (P) and 9277 infeasible points (N). In this case *Method* 1 performs very badly using both Kriging and CART as it is very close to the no-discrepancy line showing that the surrogate prediction itself is not trustworthy. *Method 2* with CART surrogate individually performs the best. *Method 3* with CART also performs quite well. This shows that the threshold mechanism proposed in *Methods 2 & 3* is better than just using the surrogate prediction (*Method 1*). The combination of surrogates for both, *Methods 2 & 3*, performs quite well which establishes the fact that combining classifiers increases the classification robustness when no prior knowledge about the best surrogate is available.

Table 4. Efficiency in predicting feasibility of 10000 test points (P=723, N=9277) for G9 problem, results from 50 DOEs.

Surrogate used	TPR		FPR		Distance from no-discrepancy line (maximum = $\sqrt{2}/2$)	
	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation
Method 1: $\hat{g}(x) \leq 0$						
CART	0.33	0.26	0.05	0.04	0.20	0.16
KRG	0.31	0.22	0.07	0.05	0.17	0.12
Method 2: $\hat{g}(x) \leq T_{\hat{g}(x)}$						
CART	0.83	0.15	0.24	0.13	**0.41**	0.08
KRG	0.70	0.30	0.32	0.20	0.27	0.15
Combination	0.77	0.21	0.30	0.27	**0.33**	0.15
Method 3: $PF(x) \geq T_{PF(x)}$						
CART	0.75	0.15	0.22	0.09	**0.37**	0.09
KRG	0.76	0.21	0.36	0.13	0.28	0.13
Combination	0.72	0.21	0.24	0.13	**0.34**	0.11

VII. Concluding Remarks

A practical method is devised to estimate the feasibility of any point in the design space using only the given DOE points and any surrogate. The method is based on ROC curves for tuning an additional threshold. It attempts to account for the error due to the limited size of the DOE and its potential unbalanced covering of the feasible and infeasible domains. Results have been shown for CART (Classification and Regression Tree), Support Vector Regression and Kriging surrogates for two analytical examples. In addition, a method for combining the predictions of multiple surrogates based on their confidence and correlation in order to get a better and more robust prediction is also presented.

The results for feasibility prediction of a set of new points in the design space for two benchmark constraint optimization problems in 2 and 7 dimensions show that best results are obtained when the Feasibility Prediction Function (FPF) of $\hat{g}(x)$ with threshold is used. The combination of surrogates using FPF of both $\hat{g}(x)$ with threshold and $PF(x)$ with threshold performed well in both test cases also and is especially useful when there is no prior idea as to which surrogate will perform the best.

One remaining aspect for future is to test the efficiency of feasibility prediction in case of multiple constraints when each constraint is handled separately rather than combining them together as has been done here.

Appendix A: Maximum Likelihood Estimator (MLE)

Maximum Likelihood Estimator[19] is a method of estimating the parameters of a statistical model. The objective function is to maximize the average log-likelihood of the observations by changing the parameters of the statistical model. For example, the unknown parameters can be the mean and standard deviation of a normal distribution.

We have a set of n sample points from the original distribution which is used to estimate the unknown parameters which are denoted by θ. All the sample points or observations are considered to be independent and identically distributed. So the joint density function of the observations is given by,

$$f(g_1, g_2, g_3, ..., g_n \mid \theta) = \prod_{i=1}^{n} f(g_i \mid \theta) = L(\theta \mid g_1, g_2, g_3, ..., g_n) \qquad (11)$$

In this work g_i's (actually $\hat{g}_I$ but for simplicity of formulas only g_i is being used in this section) are the surrogate constraint prediction values found by cross-validation for every point in the DOE.

The joint density function is also called the likelihood of getting those observations for a particular vector of θ. It is often easier to work with the logarithm of the likelihood or its scaled version which is called average log-likelihood given in Equation (12).

$$\hat{\ell} = \frac{1}{n} \sum_{i=1}^{n} \ln f(g_i \mid \theta) \qquad (12)$$

$$f(g_1, g_2, g_3, ..., g_n \mid \mu, \sigma) = \prod_{i=1}^{n} f(g_i \mid \mu, \sigma) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(g_i-\mu)^2}{(2\sigma^2)}}$$

$$= \frac{(2\pi)^{-\frac{n}{2}}}{\sigma^n} \exp\left[-\frac{\sum_{i=1}^{n}(g_i-\mu)^2}{2\sigma^2} \right]$$

$Log-likelihood$:

$$\ln f = -\frac{1}{2} n \ln(2\pi) - n \ln \sigma - \frac{\sum_{i=1}^{n}(g_i-\mu)^2}{2\sigma^2}$$

$So, Maximum \log-likelihood\ at$

$$\frac{\partial(\ln f)}{\partial \mu} = \frac{\sum_{i=1}^{n}(g_i-\mu_{MLE})}{\sigma^2} = 0 \quad or, \mu_{MLE} = \frac{\sum_{i=1}^{n} g_i}{n}$$

$and,$

$$\frac{\partial(\ln f)}{\partial \sigma} = -\frac{n}{\sigma_{MLE}} + \frac{\sum_{i=1}^{n}(g_i-\mu)^2}{\sigma_{MLE}^3} = 0 \quad or, \sigma_{MLE}^2 = \frac{\sum_{i=1}^{n}(g_i-\mu_{MLE})^2}{n} \tag{13}$$

In this paper, the distribution of the g_i*'s* is considered to be normal with the true mean, μ_{true}, which is the true constraint value at the particular point of the DOE. The standard deviation or *s(x)* at every point in the DOE is given by σ_{MLE} (Equation (13) & (14)).

$$\sigma_{MLE}^2 = \frac{\sum_{i=1}^{n}(g_i-\mu_{true})^2}{n} \tag{14}$$

Appendix B: Optimum thresholds for a DOE with no initial feasible points

When the general working assumption of at least one feasible point in the initial DOE is not satisfied then ROC curves cannot be built to get optimum threshold values. But there could be DOEs where no feasible points are present. In such cases a different method is employed to find the optimum threshold value for the FPF being used.

An assumption of the proportion of feasible space in the entire design space, Pr_T, is made (in this paper, Pr_T =5%). Then a uniformly distributed sample of N_{Pr} points in the design space is used to calculate the FPF values. The value of the threshold is continuously changed in order to satisfy the proportion of design space that is assumed to be feasible by using Equation (15). An illustration showing this method can be seen in Figure 8. The testing of this method will be done in the near future.

$$\Pr_T = \frac{\text{Feasible Space, F}}{\text{Total design space, S}} = \frac{\sum_{i=1}^{N_{\Pr}} I(\hat{g}(x^i) \le T_{\hat{g}(x)})}{N_{\Pr}} or \frac{\sum_{i=1}^{N_{\Pr}} I(PF(x^i) \ge T_{PF(x)})}{N_{\Pr}} \tag{15}$$

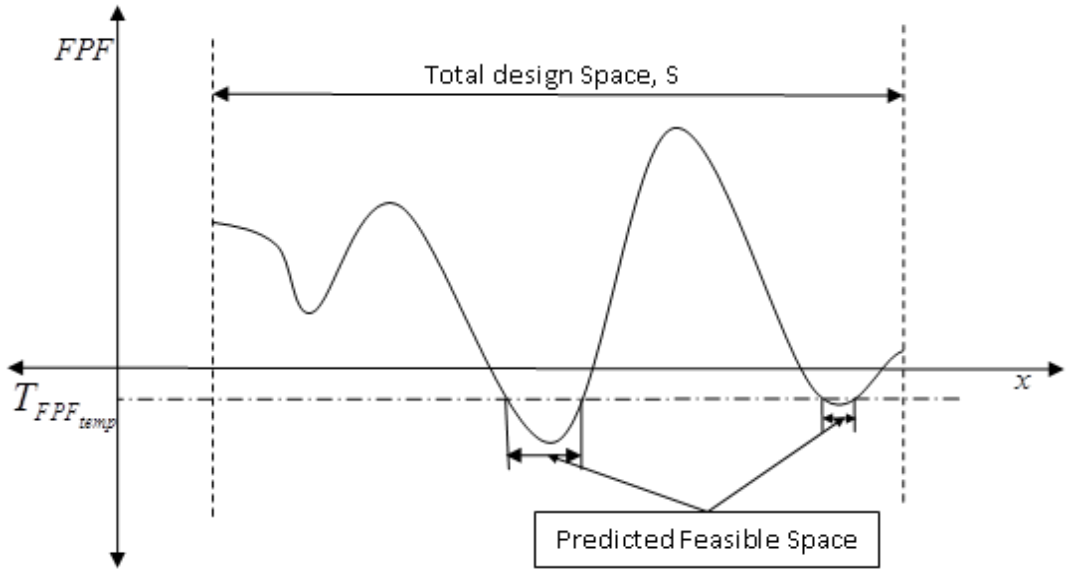

Figure 8. Illustration showing the method to tune threshold values when there are no feasible points in the initial DOE. $T_{FPF_{temp}}$ is changed until the estimated predicted feasible space volume equals a target fixed volume.

Acknowledgments

This work was supported in part by Snecma SAFRAN.

References

[1]Sasena MJ, Papalambros P, Goovaerts, "Global optimization of problems with disconnected feasible regions via surrogate modeling," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, Georgia, September 4-6,2002.

[2]Schonlau M, "Computer experiments and global optimization," *Ph.D. Thesis, University of Waterloo*, Waterloo, Ontario, Canada, pp. 100-108, 1997.

[3]Parr JM, Holden CME, Forrester AIJ, Keane AJ, "Review of efficient surrogate infill sampling criteria with constraint handling," *2nd International Conference on Engineering Optimization*, Lisbon, Portugal, September 6-9, 2010.

[4]Lee J, Jeong H, Choi DH, Volovoi V, Mavris D, "An enhancement of constraint feasibility in BPN based approximate optimization," *Computer methods in Applied Mechanics and Engineering*, Vol. 196, pp. 2147-2160, 2007.

[5]Runarsson TP, Yao X, "Search biases in constrained evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 35, Issue 2, pp. 233-243, 2005.

[6]Basudhar A, Dribusch C, Lacaze S, Missoum S, "Constrained efficient global optimization with support vector machines," *Structural and Multidisciplinary Optimization*, Vol. 46, pp. 201-221, 2012.

[7]Audet Jr C, Dennis JE, Moore DW, Booker A, Frank PD, "A surrogate based method for constrained optimization," *8th AIAA/NASA/USAF/ISSMO symposium on multidisciplinary analysis and optimization*, AIAA-2000-4891, 2000.

[8]Bichon BJ, Mahadevan S, Eldred MS, "Reliability-based design optimization using efficient global reliability assessment," *50th AIAA/ASME/ASCE/AHS/ASC on structures, dynamics and materials conference*, Palm Springs, California, 2009.

[9]Picheny V, Kim NH, Haftka RT, Queipo NV, "Conservative predictions using surrogate modeling," *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Material Conference*, Schaumburg, IL, April, 2008.

[10]Kohavi, R, " A study of cross-validation and bootstrap for accuracy estimation and model selection," *14th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, San Francisco, pp. 1137-1143, 1995.

[11]Egan JP, *Signal detection theory and ROC analysis*, Academic Press, New York, 1975.

[12]Breiman L, Friedman J, Olshen R, Stone C, *Classification and Regression Trees*, Chapman and Hall/CRC, 1984.

[13]Stein ML, *Interpolation of spatial data: some theory for Kriging*, Springer Verlag, 1999.

[14]Gunn SR, "Support vector machines for classification and regression," *ISIS technical report 14*, 1998.

[15]Smola AJ, Scholkopf B, "A tutorial on support vector regression," *Statistics and Computing*, Vol. 14, Issue 3, pp. 199-222, 2004.

[16]Fang H, Horstemeyer MF, "Global response approximation with radial basis functions," *Engineering Optimization*, Vol. 38, Issue 04, pp. 407-424, 2006.

[17]Mooney CZ, Duval RD, *Bootstrapping: A nonparametric approach to statistical inference*, Sage Publications, California, 1993.

[18]Flach P, Blockeel H, Ferri C, Hernandez-Orallo J, Struyf J, "Decision support for data mining: An introduction to ROC analysis and its applications," *Data Mining and Decision Support: Integration and Collaboration*, Kluwer publishers, Vol 745 (Part II), pp. 81-90, 2003.
[19]Aldrich J, "R. A. Fisher and the making of Maximum Likelihood 1912-1922," *Statistical Science*, Vol. 12, No. 3, pp. 162-176, 1997.