**Parameter Ranking by NBC-KDE Method (Gu et al., 2015)**

**Primary Reference:** *Jie Gu et al., Procedia Engineering 99, 2015, pp. 1082-1088*

Key Assumption: Input parameters are conditionally independent with each other

Function Spaces:

set of n-dimensional vectors corresponding to parameters

* is the random variable in the input space

be a set of class labels. In the framework by **Gu et al.,** we are labelling the input parameters as either a or a , so

* is the random variable in the output space

Methods: Training:

An early stage CCU technology is defined by its key R&D parameters (i.e. catalytic activity, Faradaic efficiency, etc) which are utilized to evaluate the performance of the technology via TEA and LCA.

Let denote a vector of R&D parameters where represents the index of a parameter. In the scope of classification, the parameters are equivalent to input features.

Each is an input to the classifier. The inputs of the Training Set are generated by performing Monte Carlo simulation runs where each simulation run is a stochastic evaluation of the CCU technology. It is stochastic because the values of the input parameters are sampled from their respective probability distribution.

Let denote the Monte Carlo simulation where . Thus, for a training set consisting of 1,000 Monte Carlo simulations:

where is a vector of features (parameters)

A target value for the performance metric of the CCU technology is set. The metric is most likely the Unit Production Cost (which can be set to the unit production cost of the non-CO2 based reference product) to ensure market competitiveness.

For each Monte Carlo run , the simulated performance metric value is observed and compared with the target value . The training instances are classified as follows

if (labelled as “success”)

if (labelled as “failure”)

Methods: PDFs of Parameters with Small Knowledgebase

A consequence of an emerging CCU technology is that there often isn’t enough information to justify a PDF underlying the uncertainty of a particular parameter. Instead of resorting to empirical distributions such as Triangular distribution, **kernel density estimation** allows a non-parametric way of generating smooth PDFs. This also removes any ambiguity in the assignment of parameter distributions for Monte Carlo analysis.

The Kernel Density function is as follows:

where represents the number of sample points where the kernel will be place at each sample point, and is the kernel function. There are a variety of kernel functions and many of them guarantees smoothness. The Gaussian kernel is the most basic and fundamental smooth kernel and is thus selected.

is a smoothing parameter which determines the “width” of each kernel where the wider the kernel the more overlap. There is no correct value for the smoothing parameter although Silverman’s rule is often adopted for deriving the optimal bandwidth for Gaussian kernels where:

is the standard deviation of the sample set for kernel density estimation. For highly multi-modal systems, the Improved Sheather & Jones method might be better for bandwidth selection. With the Gaussian kernel:

Thus for each R&D parameter whose uncertainty is to be characterized by Monte Carlo simulations, the PDFs are generated by the Gaussian KDE above given sample data point values.

Methods: Classifier:

The Maximum a posterioiri decision rule for minimizing Bayesian risk is as follows:

This function can optimally classify a given input assuming only two classs of . However this form of the expression does not utilize prior information, thus applying Baye’s Theorem:

The prior information can be estimated from the training sets. The Monte Carlo training instances are automatically labelled as explained in Methods: Training. After performing simulations, the prior probabilities are calculated by taking the probability of “success” and the probability of “failure” as a whole.

The **conditional density** term is normally calculated after Monte Carlo training by grouping all then generating a distribution for each (via Kernel Density, etc), then for each given input given the class label the probability is computed. The problem is that the number of distributions rises exponentially with the dimension of due to the need to consider conditionality between input features.

Consequently, the conditional independence assumption (Assumptions) is implemented. This assumes that all the R&D parameters of the target CCU technology to evaluate are conditionally independent. No systems truly have conditionally independent parameters, however we are still going to hold this assumption for the sake of learnability.

Applying the Naïve assumption:

Methods: Ranking

At the conclusion of the Monte Carlo simulations which act as the training data, for each outcome label “success” or “failure” is a list of sampled values for each characteristic parameter.

The posterior probabilities for EACH characteristic parameter are calculated with kernel density estimation:

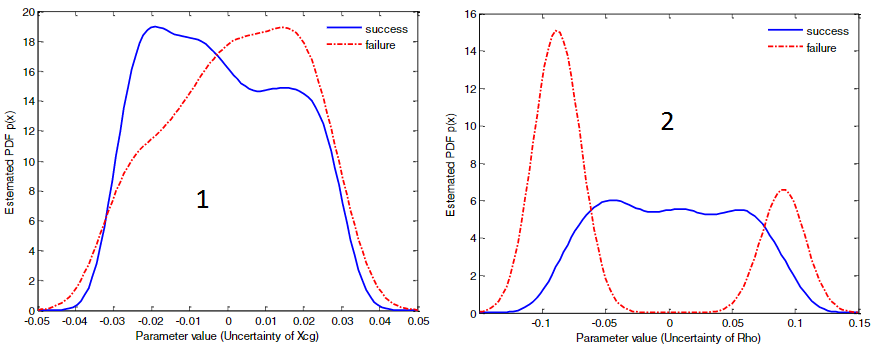
Each parameter is ranked according to their ranked score, which is defined by Gu et al. as the difference in the areas of the posterior probabilities. The greater the non-overlapping areas of the posterior probabilities the greater the rank score.

where depict the number of Monte Carlo sample points labelled “success” and the bandwidth (defined by Silverman’s Rule) for each parameter and depict the number of Monte Carlo sample points labelled “failure” and the bandwidth for parameter . is the number of discrete intervals which the PDFs are partitioned into and are summed up to approximate the area of the integrals.

Two additional parameters are defined here:

The graphs below are from Gu et al. which applied this parametric ranking framework to rank uncertain parameters for RLV flight in order of **contribution to failure.**

Note that the area of the “success” and “failure” posterior PDFs must equal to 1. Thus large area differences (high rank score) occurs when the behaviors such as that of the LEFT graph is observed and smaller area differences occur as the PDFs of “success” and “failure” become more similar.



Interpretation

Larger area differences in the posterior PDFs of success and failures of a parameter can be interpreted as that parameter being **significant** because:

* If the area differences are small, it means that the uncertainties of that parameter (by extension, variations and future improvements of that parameter) have little effect on improving the probability of achieving the target metric
* If large differences in the form of **1** are observed the parameter value should be reduced as much as possible by reducing the positive deviation uncertainty
* If large differences in the form of **2** are observed, uncertainty in the form of large tails should be mitigated as much as possible

Note that the PDF graphical representations of the ranking method provides benefits over classical correlation based methods and Sobol indice based methods because they provide insight in addition to ranking significant parameters as demonstrated above.

Summary

* Given characteristic R&D parameters of a CCU technology which we wish to evaluate where multiple data point values for each parameter is available:
* Given a target value for the performance metric which is not “too far off” from current performance of CCU technology:
* The PDF of each characteristic R&D parameter is generated non-parametrically via Kernel Density Estimation
* Monte Carlo simulations are performed where in each simulation, characteristic R&D parameters (vectorized as are sampled from their respective PDFs to compute the performance metric
* For each Monte Carlo simulation, is compared with and are labelled accordingly by comparing the two values.
* At the conclusion of Monte Carlo simulations, there is a input feature matrix of where there are columns of sampled characteristic parameter values. Additionally we have an output label matrix of which is a binary matrix of class labels
* The prior probabilities are computed by observing the entire training set. The posterior probabilities and for each characteristic parameter is computed by applying KDE to the Monte Carlo samples that are grouped to each class label.
* The difference in the posterior probabilities define the rank score and is computed numerically according to the formula proposed by Gu et al. In addition, graphically visualizing the difference in posterior PDFs allows one to gain additional insight as described in Interpretation