**PROCEDURE: NAÏVE-BAYES PARAMETER RANKING**

**NOTE TO SELF:** *Red texts indicate parts that I do not fully understand/can explain*

**[STEP 0]** *Select a set of key model parameters. Let = number of key model parameters which is indexed by . Thus the key parameters* *are indexed as follows:* *for*

* *If a model has many parameters the key parameter set can be selected on the basis of representativeness of R&D (parameters such as electricity price for example, while sensitive arent very representative of R&D and isnt something that is improved with research). However, technically any variable is a parameter as long as it is manipulable (is an independent variable) and represents a DOF of the CCU model.*
* *If information regarding R&D representativeness isn’t available for parameter selection, as many parameters should be selected (we propose that ALL DOF parameters be selected for initial set). Sobol Indices from* ***MC-Sobol*** *can be actually used to evaluate the parameter selection as it ranks parameters on the contribution to variance*
* *At TRL 2-3 the core phenomena is usually understood under First Order Principles, so it shouldn’t be difficult to distinguish the most important R&D parameters.*

**[STEP 1]** *Collect multiple sample points for each parameter in the process model*

* ***EX:*** *In the optimal operating conditions of a CCU process that has been designed via 1st principle model, what are the deterministic parameter values??*
* *The sample points need not necessarily be from optimality point and can be collected from pilot/lab-scale experiments or in the worst case, from literature study. The data collector should note discrepancies in the experimentation apparatus that might explain differences in parameter values.*
* *If repeated sampling of data is possible at the same conditions, then the engineer can opt to use parametric sample distributions (mean and variance) rather than kernel distributions.*

**[STEP 2]** *Establish probability distributions for each parameter using the Kernel Density Estimation (KDE) method. This is done by placing a Gaussian kernel at each sample point in* ***STEP 1*** *for each key parameter determined in* ***STEP 0****. The KDE generates a smooth continuous PDF as follows:*

*Where is the sample size (# of data samples), h is the bandwidth (curve spreadiness) and are the values of the data samples. With a Gaussian kernel, the equation becomes:*

* *One reason for using KDE for PDF generation is to avoid ambiguity issue in “assigning” a probability distribution for a parameter.*
* *Usually a PDF was assigned heuristically (assuming normal distribution, uniform etc). Generating a PDF that is defined by sample data points avoids the ambiguity issue as the PDF is tailored to available data. It can be proved that with infinite sample points, the KDE distribution converges to a Normal one*
* *Because the KDE distribution is sensitive to bandwidth h (which acts as “stdev” for parametric distributions), it can be argued that choosing a value for still retains some level of ambiguity. However, statisticians have addressed this issue by coming up with a way to calculate the* ***optimal bandwidth*** *for a given kernel sample set.*
* *The optimal bandwidth algorithm for the current code uses* ***Scott’s Rule*** *(Ref: Scott, D.W. (1979), “On optimal and data-based histograms”, Biometrika, 66, 605-610. The heuristic says that where is the number of sample points and is the number of spacial dimensions (for univariate data, ).*
* *The proportionality factor for Scott’s Rule is derived in a similar manner to Silverman’s rule and is calculated out to be 1.059*

**[STEP 3]** *Determine the total number of Monte Carlo simulations to be performed, , where the # of simulations equals the # of sampled sets of Key Parameters and the # of Process Model evaluations. Each Monte Carlo CCUS evaluation in* *is indexed with* *(simulation in )*

* *Usually, enough simulations are required to guarantee smoothness in the distribution of the model outputs (when plotted).*
* *Alternatively, the minimum number of Monte Carlo runs can be determined by the formula proposed by Gu et al., Proceedia Engineering 99, 2015:*

*Here, z is the confidence level as number of standard deviations (z=3) and is the evaluation error which can be something like*

* *In most cases, it is not necessary to calculate the minimum as sufficient MC simulations should be performed for sufficient accuracy. The limiting factor is time.*
* *Usually MC simulations in the order of 104 are generated but Wei et al., 2016 showed that 106 might be necessary*

**[STEP 4]** *Define and populate a Parameter Space for the Monte Carlo sampled parameter sets. The parameter space is a matrix where for each Monte Carlo simulation, each* *key parameters are random sampled from their respective PDFs defined by their respective distributions. For parameters with KDE PDFs, they must be sampled from the kernel distributions which are most likely discrete distributions.*

**[STEP 5]** *Using the sampled parameter values in the* ***“Parameter Space”*** *perform evaluations of the CCUS Process Model. In other words, parameters that are sampled from their respective PDFs in “Parameter Space” are used to solve the CCUS process model times, where each time we evaluate there are model outputs.*

*For example, if Model Outputs = for 500 Monte Carlo simulations, then and we get a or 500x3 matrix, which we lavel as* ***M\_Evaluations.***

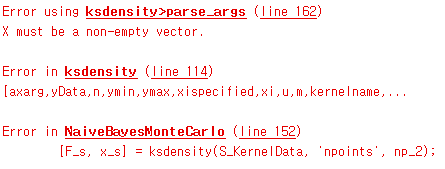
* *The outputs of the CCUS process model can be technoeconomic or LCA metrics (such as Unit Cost of Production or Specific GWI) or “process” metric such as Specific Yield. However TEA and LCA metrics such as Unit Prod. Cost and Specific GWI should be used as they represent the technology’s performance better than Specific Yield.*
* *For 2 or more Model Outputs, we have 2 or more separate ranked sets. A popular choice I plan to use is 1 TEA Metric and 1 LCA Metric, each with NBC ranked parameters.*

**[STEP 6]** *Identify target metric values for Bayesian Classification. If a CCU Process Model has 2 output metrics , where ) then a target metric value must be assigned for each output.*

* ***EX:*** *For the CCU Methanol process, we frequently evaluate the performance by comparing CCU methanol to Conventional Methanol process. This is because conventional methanol is still cheaper than CCU methanol while having higher unit emissions*
* *Thus, the target metric value for Unit Prod. Cost for CCU Methanol process can be set to that of Conventional Methanol unit production price, which is ~$320 USD/ton.*
* *The target metric value for Specific Global Warming Impact for CCU Methanol can be set to target CO2 reduction goals from CO2-based methanol process*

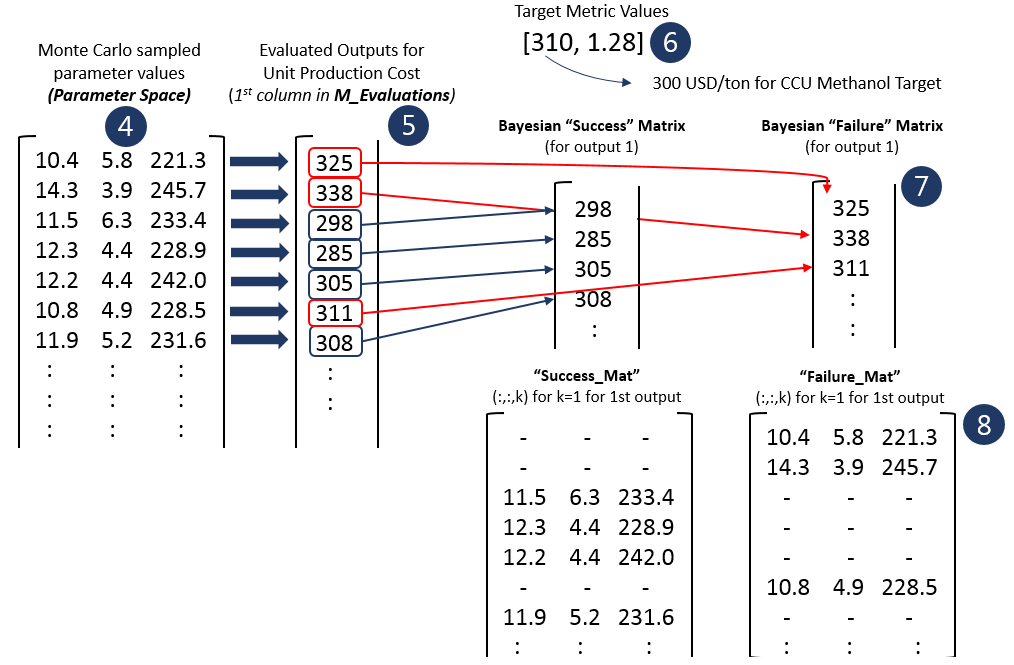
**[STEP 7]** *Generate a Bayesian classification matrix. Looping through the columns of* ***M\_Evaluations*** *for each output (each column in M\_Evaluations), for each Monte Carlo simulated output (each row), compare the MC output value with the target value. If the MC output value is* ***at or below*** *the target value, then label that MC output as “Success” = 1. If the MC output value is* ***above*** *the target metric value, label that output as a ”failure” = 0. Repeat for all MC simulations, and for all outputs in .*

* ***NOTE1:*** *Because both “Unit Production Cost” and “Specific GWI” are cost metrics, i.e., the smaller the better, both outputs are classified as success if the MC output value is ≤ target value. If it was a “benefit metric” such as “Net Revenue”, the MC outputs should be classified as success if their values are ≥ the target metric value.*
* ***NOTE2:*** *The targets must be realistic enough so that there is a decent number of classifications for both labels. If the targets are too one-sided so that all the MC simulations for an output metric in are classified to one label, the parameter ranking fails (MATLAB will also return the below error:*)

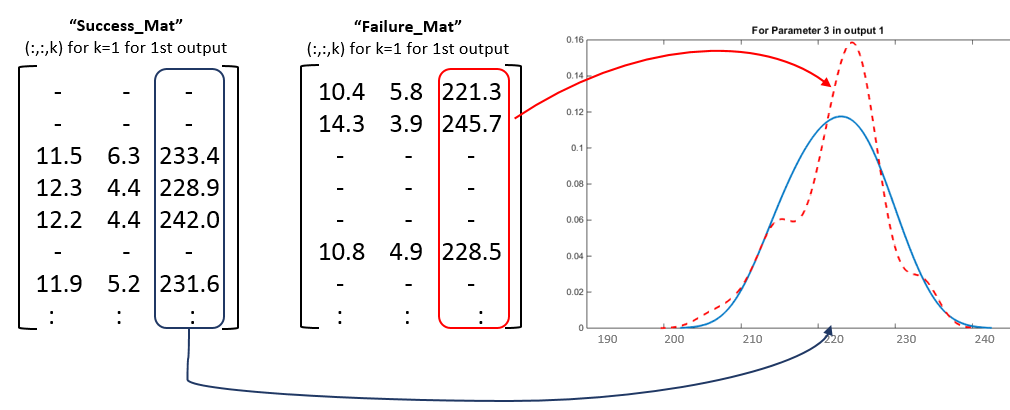
**

**[STEP 8]** *Create two matrices of size called* ***Success\_Mat*** *and* ***Failure\_Mat****.*

* *Populate the rows of* ***Success\_Mat*** *with the values from the rows of* ***“Parameter Space”*** *if for the index of the row, the corresponding Monte Carlo output value has been classified as “Success”. Else, populate the rows of* ***Success\_Mat*** *with zeros and repeat for all outputs in .*
* *Populate the rows of* ***Failure\_Mat*** *with the values from the rows of* ***“Parameter Space”*** *if for the index of the row, the corresponding Monte Carlo output value has been classified as “Failure”. Else, populate the rows of* ***Failure\_Mat*** *with zeros and repeat for all outputs in .*

**

**[STEP 9]** *For each key parameter in* *and for each CCU model output in , generate kernel probability density distributions using the corresponding vector of values in* ***Success\_Mat*** *and* ***Failure\_Mat****. There should be a total of kernel distributions.*

**

* The kernel density function used here is identical to that used to establish PDFs for model parameters:

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. The Gaussian kernel is the most basic kernel that guarantees smoothness and is thus selected.

is a smoothing parameter which determines the “width” of each kernel where the wider the kernel the more overlap, and is determined by Scott’s Rule.

* Intepretation: the Monte Carlo simulation acts as a training data, or a “sample population estimate” of the true outcome probability (probability of success and failure) for each key parameter. Kernel density method for posterior probabilities have the following form:

*where and*

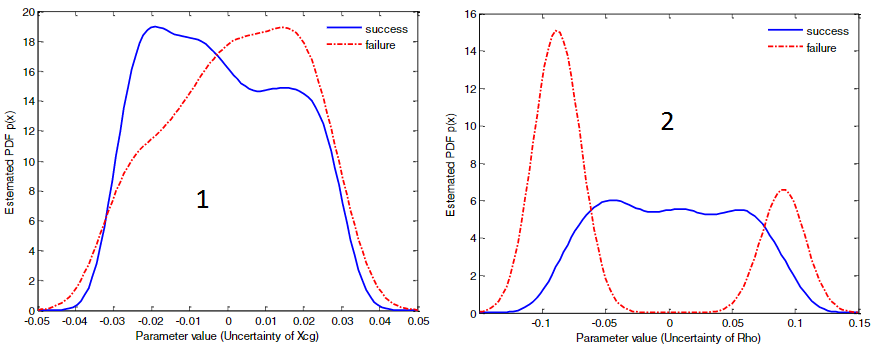
* *The naïve assumption for conditional independence is expressed in the classification procedure in* ***STEP 7*** *whereby:*

**[STEP 10]** *Compute the Naïve Bayes rank score for each parameter in and for each output in . The rank score is defined as the difference in the kernel probability distributions, i.e., the greater the non-overlapping areas of the posterior probabilities, the greater the rank score*

*In discretized form, the above integral is as follows:*

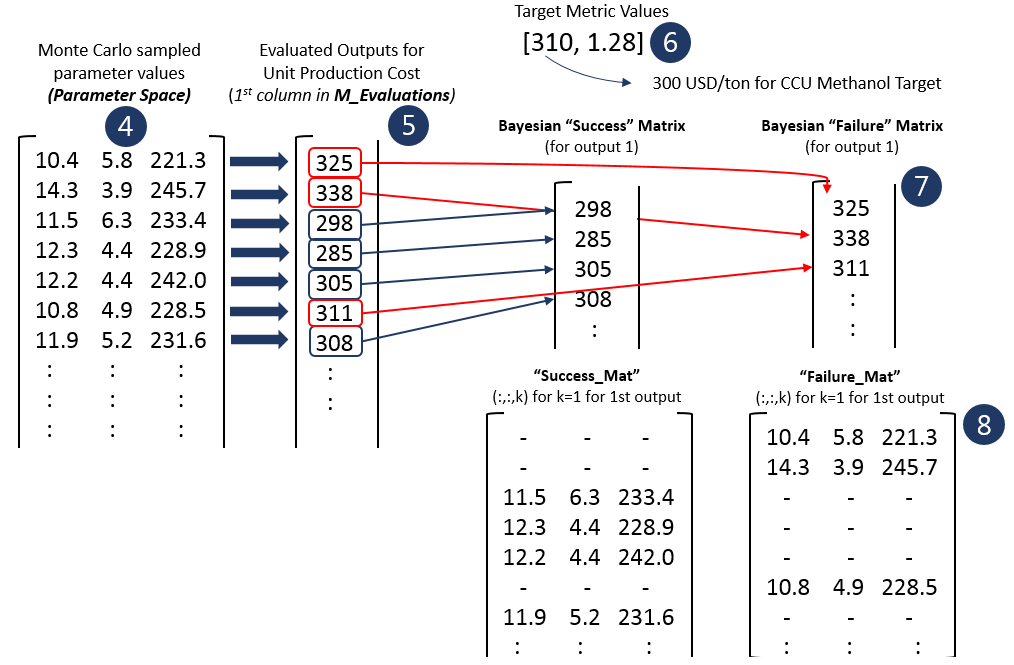
*where is the number of discrete intervals (~105 currently). and are the following:*

* *If the differences in kernel distributions for “success” and “failure” are small for a certain key parameter (i.e., lower rank score) it means that the value of said parameter has little impact or significance in determining the success rate or failure rate (failure rate >> RISK) of the output*
* *The Naïve Bayes rank order for each output can be different, and thus, compromising different rank orders will need to be addressed in this study.*
* **NOTE:** *The NBC ranking method has advantages in that graphical representation of the Bayesian kernal distributions can offer additional insights (Gu et al., 2016). Consider the following figures:*



* *If differences in the form of* ***“1”*** *are observed, the value of the parameter should be reduced as much as possible by reducing positive deviations in parameter uncertainty*
* *If differences in the form of* ***“2”*** *are observed, the uncertainty of large tails should be mitigated as much as possible*

**STUDY NOTES**

**

**Posterior Probability**  >> for a given evaluation outcome ( , ) what is the probability of parameter ?

* In other words, what is the probability of a class label ( or ) based on the value of a parameter ?? AKA, the distribution of success/failure rates depending on the value of parameter
* Unlike prior probability, this represents the uncertainty of parameter that has been revised with additional data. Here, the additional data is that depending on certain evaluation outcomes, the appearance of certain values of change, so we can revise the PDF of

**Prior Probability:** , what is the probability of the values parameter ?

* Basically take all sampled values of parameter and plot a histogram **(4 above),** AKA this is the probability distribution for each parameter which we define using kernel distributions using literature/bench-scale data
* The prior probability of parameter basically represents the uncertainty of parameter

**Rank Score:** The rank score is defined as the difference of the posterior PDFs (area difference) for “success” and “failure” classes. Parameters are ranked higher if the area difference in the posterior PDFs are larger

* AKA, for the exact same parameter ’s value, there is a noticeable effect on the evaluation outcome. Therefore it is more significant from the basis of \_\_\_\_. In addition, graphing the kernal distributions of the posterior PDFs can reveal more intricate details such as in Jie Gu et al., 2015
* Low rank score means small difference in the posterior kernel PDFs. This means regardless of parameter ’s value, there is little difference evaluation outcome classification.
* The smaller the rank score, the more idential the posterial kernel PDFs. AKA, regardless of ’s value, the probability of class outcome is virtually identical >> which means it is insignificant