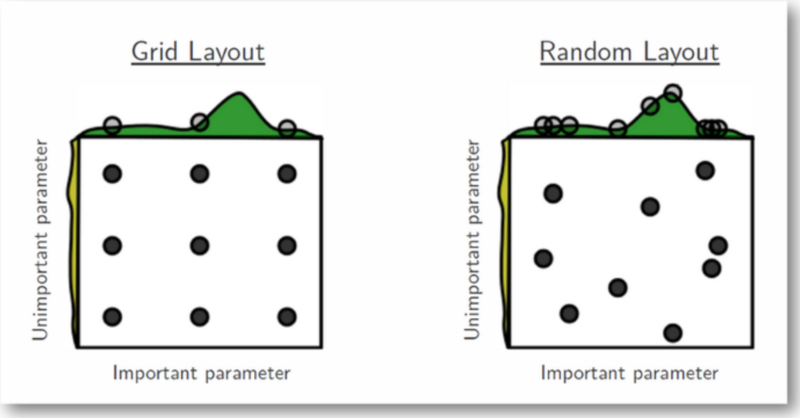
Calibration of Heston Model

After building Heston Model, it's time to do the calibration. We can see that Heston model requires several parameters to do the simulation and these parameters decide the accuracy of option pricing. The process of calibration can be treated as a non-linear optimization programming problem, so we applied least-square error as our objective. Inspired by the hyperparameter optimization of machine learning models, we used some Sequential Model-Based Optimization and genetical algorithm to calibrate the Heston Model.

Sequential Model-Based Optimization (SMBO)

There are many ways to do global hyperparameter optimization such as grid search and random search. Grid search will iterate each possibility and search for the best solution, which is super time-consuming. Random search will take shorter time but the result is relatively worse for big parameter spaces. Therefore, SMBO appears as one of the most efficient optimization method.



With one sentence to describe SMBO, it searches for parameters according to the minimum of the expectation of loss function and the maximum of variance within a limited iteration times. Considering a fitness function $f: X \rightarrow \mathbb{R}$ for evaluation where X is the parameter space, $D(x, y)$ refers to the dataset of each pair of parameter and its corresponding solution. S is called acquisition function which is used to choose parameters (X). M is the probabilistic regression model used to fit the dataset (D). T is iteration times. Below is the pseudo code of SMBO:

```
$Input: f, X, S, M$
$D \leftarrow INITSAMPLES(f, X)$
  $for\ i \leftarrow |D|\ FitModel(M, D)$
  $x_i \leftarrow \arg max_{x \in X}\ S(x, p(y|x, D))$
  $y_i \leftarrow f(x_i) \quad \rightarrow Expensive\ Step$
  $D \leftarrow D \cup (x_i, y_i)$
$end\ for$
```

In this project, we select Bayesian Optimization and Tree-Structured Parzen Estimator, which both use Gaussian Mixture model, to do the calibration. The acquisition function is expected improvement (EI) because its performance is the best in Gaussian distribution. EI's formula can be expressed with some threshold y*:

$$\mathrm{EI}(\mathrm{x}|\mathrm{D}) = \int_{\mathrm{y}^*}^{\infty} \max(\mathrm{y}^*\text{-}\mathrm{y}, 0)\mathrm{p}(\mathrm{y}|\mathrm{x})\mathrm{dy}$$

1. Bayesian Optimization(BO)

One probabilistic regression model we used is Gaussian Process (GP). In this method, f is assumed as a realization of a $GP(\mu, K)$ where μ is the mean function and K is the covariance kernel. Then it applies Gaussian process regression which is to interpolate on the surface with a GP with the covariance kernel function. An estimation can be given by the mean. Next point is chosen in terms of the maximum of EI then the global maximum can be found after iterations.

2. Tree-Structured Parzen Estimator (TPE)

TPE is sort of similar to Bayesian optimization but it uses $p(x|y)$, $p(y)$ to model while BO wields $p(y|x)$.
$p(x|y)$ is the conditional probability when the loss is y and the parameters is x. First, define a threshold of loss called y^*. For y greater or smaller than y^*, TPE defines two densities $l(x)$ and $g(x)$:

$$\mathrm{p}(\mathrm{x}|\mathrm{y}) = \begin{cases} \mathrm{l}(\mathrm{x}) & \text{if } \mathrm{y} < \mathrm{y}^* \\ \mathrm{g}(\mathrm{x}) & \text{if } \mathrm{y} \geq \mathrm{y}^* \end{cases}$$

Using $p(x|y)$ and $p(y)$, EI's formula becomes:

$$\mathrm{EI} = \int_{-\infty}^{\mathrm{y}^*} (\mathrm{y}^*\text{-}\mathrm{y})\mathrm{p}(\mathrm{y}|\mathrm{x})\mathrm{dy} = \int_{-\infty}^{\mathrm{y}^*} (\mathrm{y}^*\text{-}\mathrm{y}, 0)\frac{\mathrm{p}(\mathrm{x}|\mathrm{y})\mathrm{p}(\mathrm{y})}{\mathrm{p}(\mathrm{x})}\mathrm{dy}$$

Next, construct $\gamma = \mathrm{p}(\mathrm{y} < \mathrm{y}^*)$ and $\mathrm{p}(\mathrm{x}) = \int_{\mathbb{R}} p(x|y)p(y)dy = \gamma l(x) + (1-\gamma)g(x)$.

Then,

$$\int_{-\infty}^{\mathrm{y}^*} (\mathrm{y}^*\text{-}\mathrm{y})\mathrm{p}(x|y)p(y)\mathrm{dy} = l(x)\int_{-\infty}^{\mathrm{y}^*} (\mathrm{y}^*\text{-}\mathrm{y})\mathrm{p}(\mathrm{y})\mathrm{dy} = \gamma\, \mathrm{y}^*\, l(x) - l(x)\int_{-\infty}^{\mathrm{y}^*} \mathrm{p}(\mathrm{y})\mathrm{dy}$$

Finally we can get,

$$EI = \frac{\gamma\, \mathrm{y}^*\, l(x) - l(x)\int_{-\infty}^{\mathrm{y}^*} p(y)dy}{\gamma l(x) + (1-\gamma)g(x)} \propto (\gamma + \frac{g(x)}{l(x)}(1-\gamma))^{-1}$$

This formula indicates that we can find points x which has high probability under $l(x)$ and low probability in $g(x)$. The tree-structured l and g will return the candidate x* with greatest EI and keep minimizing $\frac{g(x)}{l(x)}$ until target iteration times.