Quan Nguyen
quan@wustl.edu

Research report
April 10, 2020

Summary:

- Subsection 1.1 describes the process of applying a Gaussian kernel density estimator to fit the points in our data set while making sure the support is between 0 and 1.
- Subsection 1.2 includes the scatter plots for the pairwise distributions of data points in any two given interventions.
- Subsection 2.1 describes the unique characteristic of the leximin assignment when the rows in the cost matrix are sorted. This leads to a slighly new bound for PoF described in subsection 2.2.
- Subsection 2.3 includes results obtained from simulations comparing distributions of PoF when costs are drawn from various Beta distributions.

1 More on the real data set

1.1 KDE on bounded support

To make sure a kernel density estimator does not give positive mass to values outside of $[0, 1]$, we first map each point in the data set (which is between 0 and 1) to a real-valued number using the inverse of the Sigmoid function:

$$s^{-1}(x) = \ln(x) - \ln(1 - x)$$

An estimator can then be applied to fit the transformed data, which is now unbounded.

Our goal is to sample from the original data set. To do this, we sample from the fitted estimator in the new space and then transform the sampled data points back to be between 0 and 1 using the Sigmoid function.

The following Figure 1 compares the distribution of the original data (across all columns) and the distribution of 10,000 samples drawn from the KDE and transformed back:

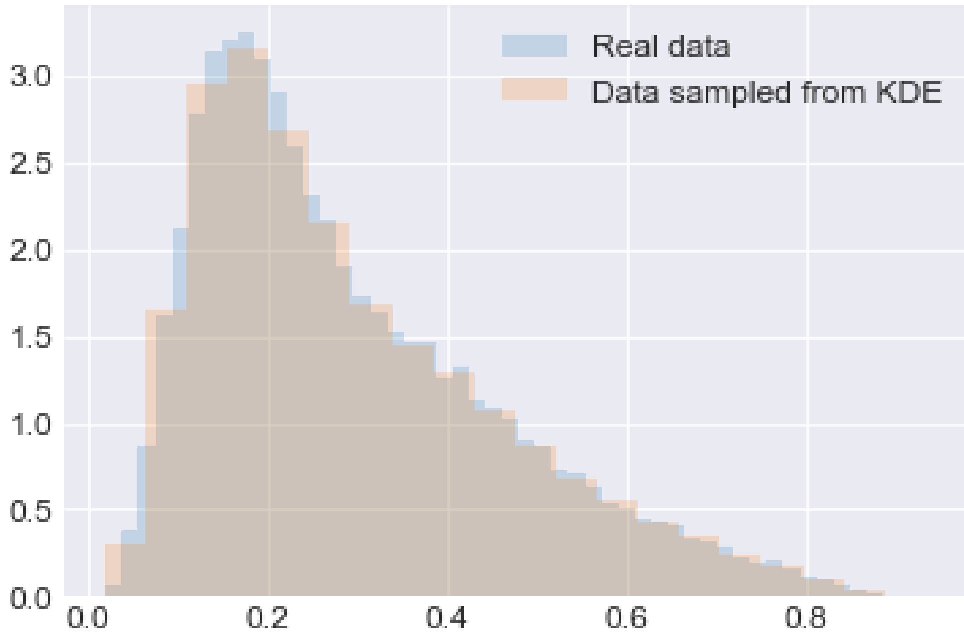

Figure 1: Real data vs. samples from KDE

1.2 Joint distributions

We are also interested in the joint distribution of an agent's probabilities for any two given interventions. This is illustrated by the following scatter plots denoting the pairwise distributions of probabilities:

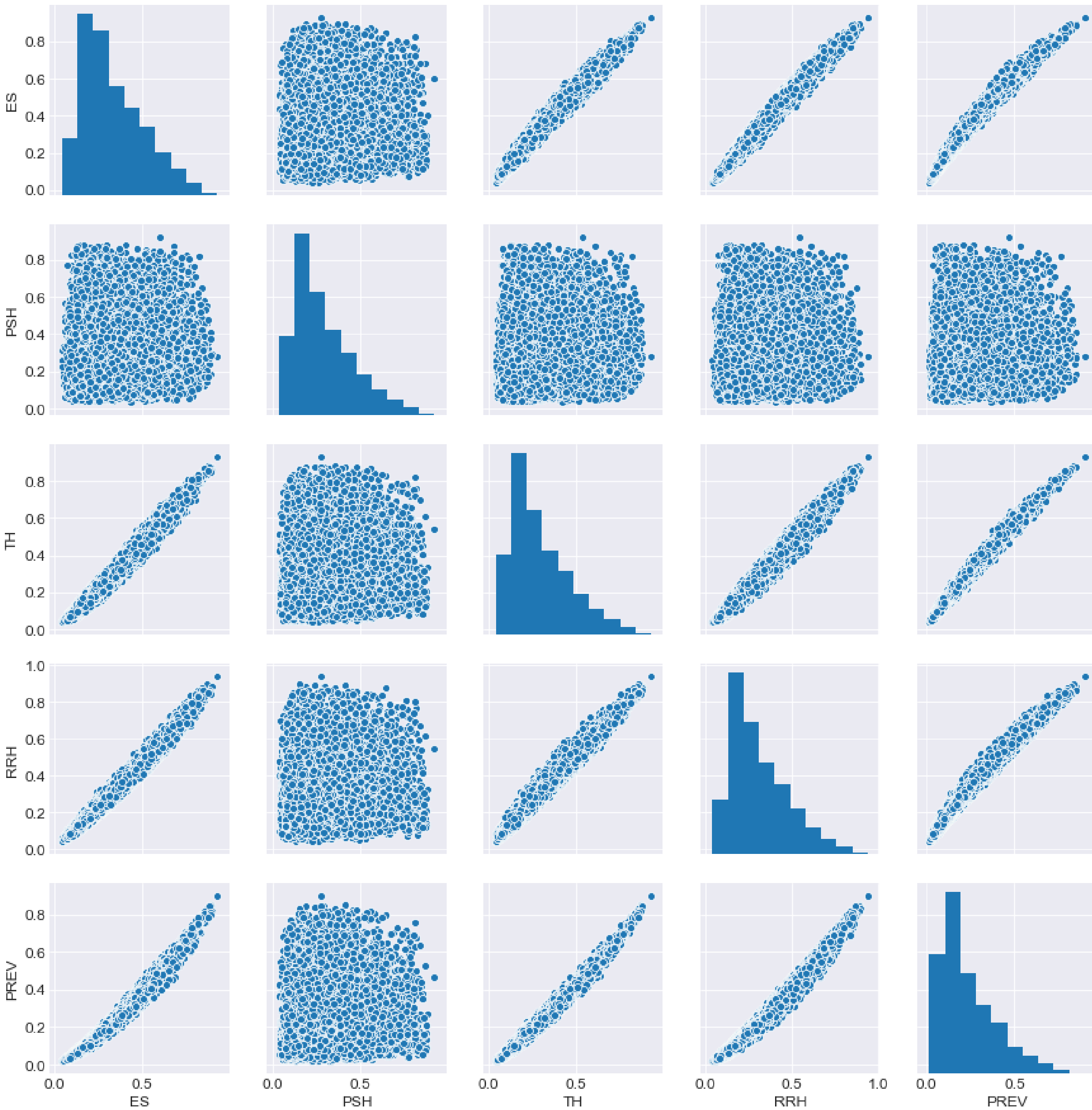

Figure 2: Pairwise bivariate distributions

2 PoF when the rows are sorted

We now inspect the behavior of the PoF quantity under the assumption that each row in the cost matrix defining an instance of the assignment problem is sorted in increasing order.

This is to say that for every agent, intervention 1 is more effective than intervention 2, which is more effective than intervention 3, and so on.

2.1 Computing the leximin assignment

Under the assumption of sorted rows, the leximin assignment becomes very easy to compute. This is because of the following claim: under the leximin assignment, the c_m agents with the lowest costs in the last (m^{th}) column are assigned to intervention m (where c_m is the capacity of intervention m). Assume this is not the case, and under leximin, there exists an agent i among these c_m agents that is assigned to intervention $j \neq m$ and another agent i' who are not among these c_m agents that is assigned to intervention m.

Consider an alternative assignment where the assignments of these two agents i and i' are switched and the other assignments are kept the same. The cost of agent i will increase to $x_{i,m}$ while the cost of agent i' will decrease to $x_{i',j}$. Both of these two new costs are lower than the old cost of agent i', which was $x_{i',m}$. Therefore, this cost $x_{i',m}$ is not a valid leximin, hence contradicting our initial assumption.

As such, we can also prove that out of the remaining $n - c_m$ agents, the c_{m-1} agents with the lowest costs with respect to intervention $m-1$ are assigned to intervention $m-1$ under leximin, and so on. Thus the leximin assignment becomes the following:

Algorithm 1 Compute the leximin assignment of a row-sorted matrix

```
for j = m, ..., 1 do
    Assign the c_j agents with lowest costs in the j^th column out of the remaining agents to
    intervention j
end for
```

Quick note: This algorithm has a greedy flavor to it. Is it an approximation of the minimal assignment?

2.2 More on the PoF bound

Due to the characteristic of the leximin assignment described above, the total cost under the minimal assignment of the agents assigned to intervention m is at least that under the leximin assignment. Denoting $C_j(\Pi)$ as the total cost of the agents assigned to intervention j under a given assignment Π, we have:

$$C(E) \geq C_m(E) \geq C_m(L) = C(L) - \sum_{j=1}^{m-1} C_j(L)$$

Isolating the term $\mathrm{PoF}(L) = C(L)/C(E)$, we have:

$$\begin{aligned}
\mathrm{PoF}(L) &\leq 1 + \frac{\sum_{j=1}^{m-1} C_j(L)}{C(E)} \\
&\leq 1 + \frac{(n - c_m)\bar{b}_L}{C(E)} \text{ (where } \bar{b}_L \text{ is the largest cost under leximin)} \\
&\leq 1 + \frac{(n - c_m)\bar{b}_L}{C_m(E)} \\
&\leq 1 + \frac{(n - c_m)\bar{b}_L}{C_m(L)} \\
&\leq 1 + \frac{(n - c_m)\bar{b}_L}{c_m \ \underline{b}} \text{ (where } \underline{b} \text{ is the lowest cost in the last column)}
\end{aligned}$$

$\bar{b}_L$, the minimized bottleneck of leximin, is at most the maximum cost in the matrix, which is at most 1. We can then loosen the bound above to get a simpler form:

$$\mathrm{PoF}(L) \leq 1 + \frac{(n - c_m)}{c_m \ \underline{b}}$$

For a given instance of the assignment problem, the RHS is a decreasing function of c_m. A direct interpretation is that, for any two instances of an assignment problem with the same cost matrix, the instance where the least effective intervention has more capacities has a lower upper-bound for the PoF quantity.

This makes intuitive sense as when both the minimal and leximin assignments have to assign many agents to the least effective intervention, the difference in total cost among the two assignments should decrease, as the minimal assignment does not benefit as much from the assignments in the other interventions.

2.3 Experiments with different distributions

In the following experiments, we aim to inspect the distribution of PoF when the individual costs are drawn from specific distributions (and then sorted row-wise). In each experiment, a cost matrix denoting an instance of an assignment problem with 30 agents and 5 different interventions is used. Individual costs are drawn i.i.d. from the specific distributions that we consider to make up the cost matrix. Individual rows are subsequently sorted.

In each experiment, the relative order among the costs is fixed among matrices whose costs are drawn from different distributions. For example, when we compare the uniform distribution and the KDE fitted on our data set (as described above), a matrix with uniformly drawn costs is first generated and has its rows sorted. A matrix with costs drawn from the KDE and transformed back is then generated, but now the costs are sorted in the same order as the costs in the first matrix. In other words, among the matrices used in the same experiment, their lowest costs are in the same location, so are their second lowest costs, and their third lowest costs, and so on.

Each graph is generated using results from 500 repeated experiments.

2.3.1 Uniform costs vs. KDE

Here we compare the distribution of PoF when costs are drawn from $U[0,1]$ against that when they are drawn from the KDE and transformed back. Here all five interventions have the same capacity: 6. The following figure denotes those distributions across 500 experiments:

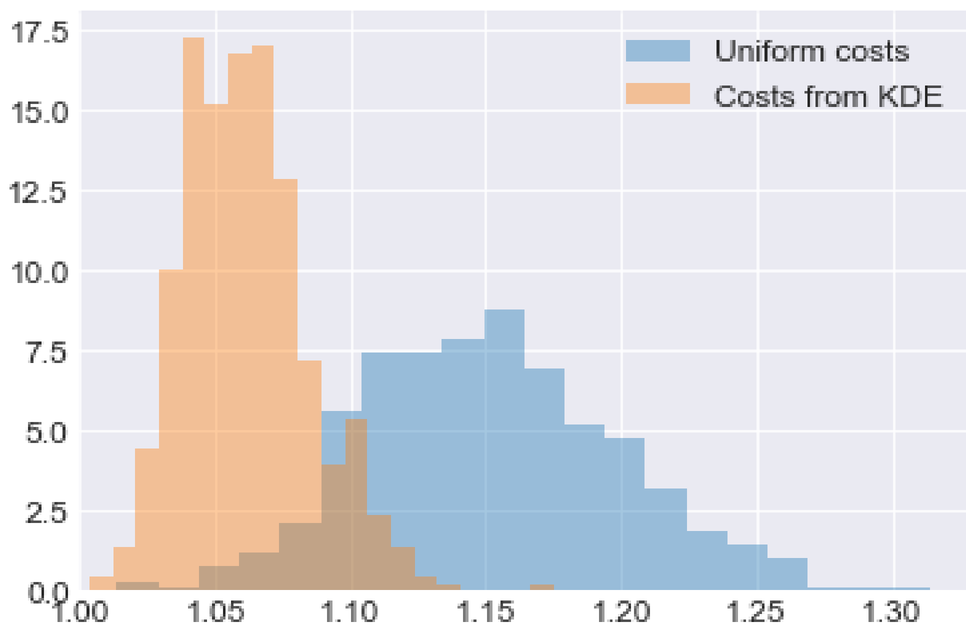

Figure 3: Distribution of PoF – $U[0,1]$ vs. KDE

It seems that the PoF quantity is more likely to have a lower value when the costs are drawn from the KDE than from $U[0,1]$. To see if this is the case in every experiment, the ratio between the PoF resulting from the matrix with uniformly drawn costs and the PoF resulting from the corresponding matrix with costs drawn from the KDE in each experiment is recorded and its distribution is visualized below:

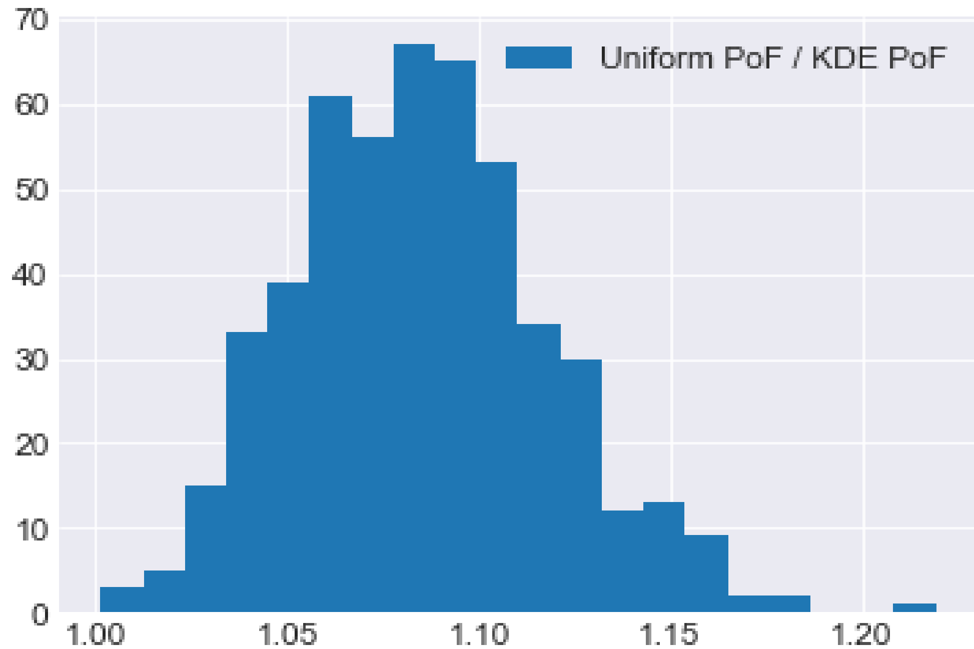

Figure 4: Distribution of ratio between two PoFs

It seems that most (if not all) ratios are at least 1, indicating that we always obtain a lower PoF when costs are drawn from the KDE as opposed to from $U[0, 1]$.

2.3.2 Different Beta distributions

Instead of comparing the PoF resulting from matrices from two different distributions, we can do the same for multiple distributions. Here we naturally limit ourselves to Beta distributions, whose support is $[0, 1]$.

The following boxplots visualize the distribution of PoF when the costs are drawn from a Beta distribution with corresponding parameters (while ensuring the matrix elements in each experiment have the same order).

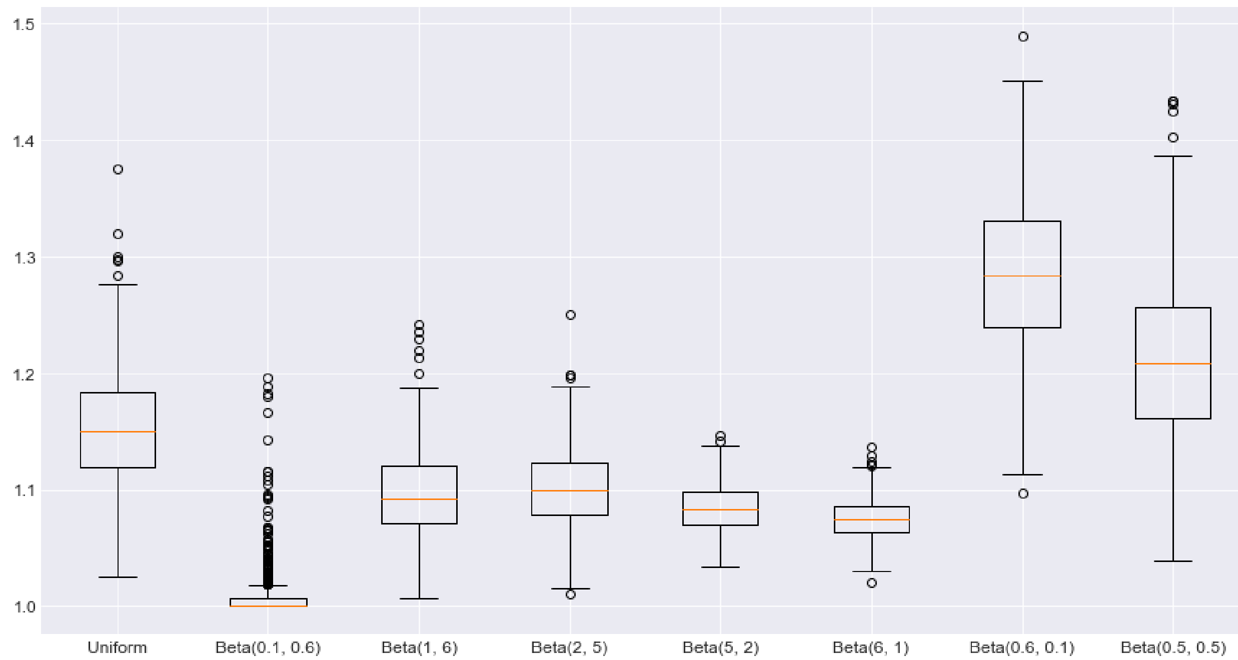

Figure 5: Distribution of PoF with different distributions

We observe that the distribution of PoF under $U[0, 1]$ has higher values than those under various Beta distributions with parameters $\alpha > 1$ and $\beta > 1$. Beta$(0.1, 0.6)$ and Beta$(0.6, 0.1)$ result in the most contrasting distributions: the PoF distribution of the former has significantly lower values than others, while that of the latter has significantly higher values.

2.3.3 Different capacities

So far, the capacities of the interventions in the experiments have been equal to each other. Here we want to see whether the behavior of PoF would change with increasing ($c_1 < c_2 < ... < c_m$) and decreasing capacities ($c_1 > c_2 > ... > c_m$).

The following boxplots were generated from the same experiments as above, but where the capacity vector $\mathbf{c} = [2, 4, 6, 8, 10]$.

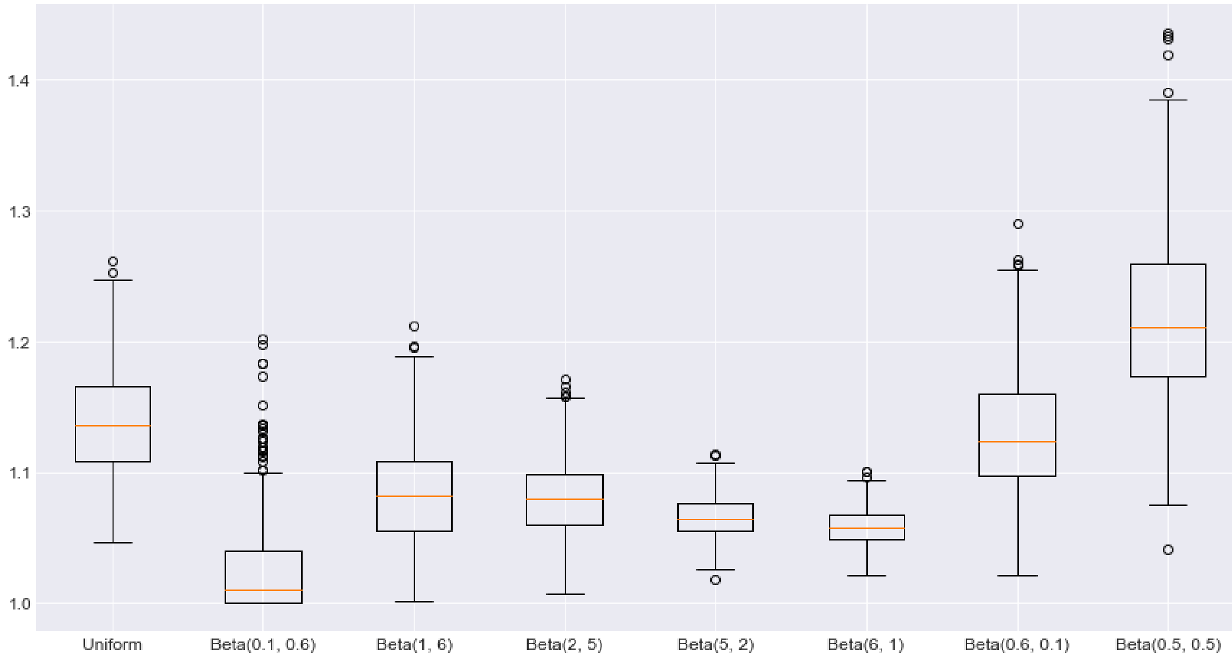

Figure 6: Distribution of PoF when $c_1 < c_2 < ... < c_m$

We see that compared to when $\mathbf{c} = [6, 6, 6, 6, 6]$, distributions of PoF in this case take on lower values (except for the case of $\text{Beta}(0.1, 0.6)$). This is consistent with the claim that with more capacity in the less effective interventions, the PoF might become lower.

The following boxplots were generated from the same experiments as above, but where the capacity vector $\mathbf{c} = [10, 8, 6, 4, 2]$.

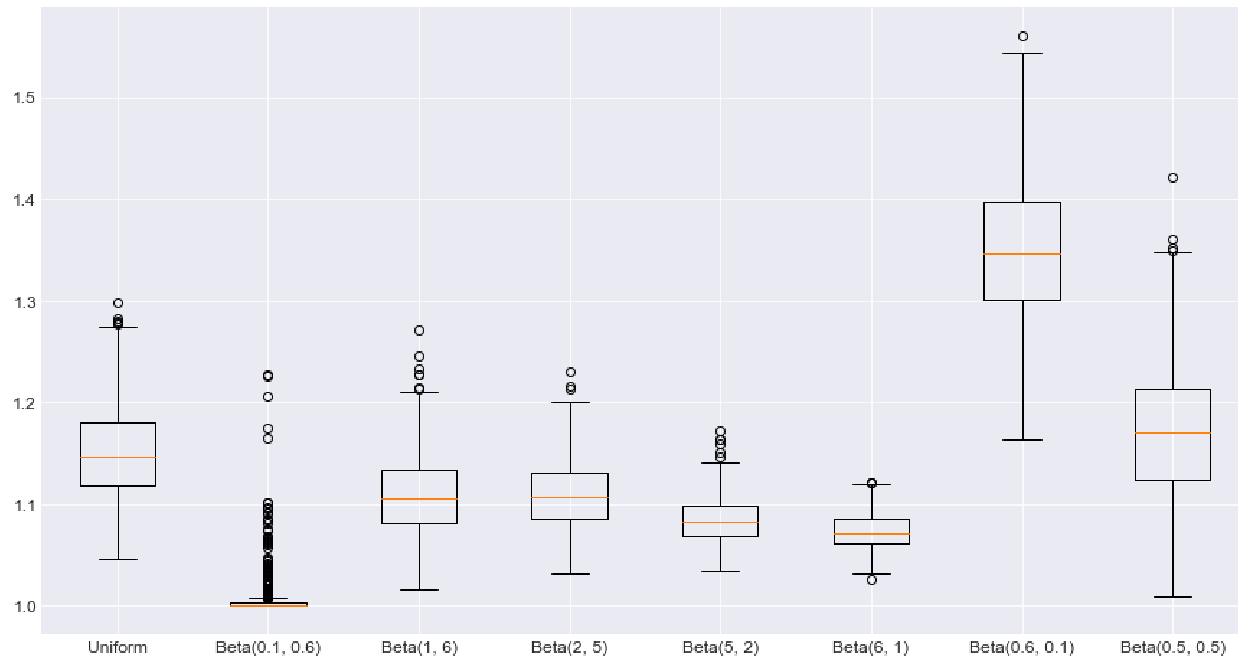

Figure 7: Distribution of PoF when $c_1 > c_2 > ... > c_m$

Here the contrast between Beta$(0.1, 0.6)$ and Beta$(0.6, 0.1)$ grows larger, while the other PoF distributions don't change much.