Some time ago I stumbled about a problem connected with the labels of a clustering. The partition an instance belongs to is labeled mostly through an integer ranging from 1 to K, where k is the number of clusters. The task at the time was to plot a map of the results from the clustering of spatial polygons where every cluster is represented by some color. Like in most projects the analysis was performed multiple time and we used plotting to monitor the changes resulting from the iterations. But after rerunning the clustering algorithm (k-means in this case) the assignment between the clusters and the labeling changed completely, even when use the same parameters. This is because there is no unique connection between a partition (a group of elements) and a specific label (eg. “1”). So even when two solutions match perfectly the assigned labels changed completely. So the graphical representations of two clustering’s (which only have some slight differences) look like they are completely different. This is because the coloring relates to the labels. The following R code depicts a simple example for this matter:

After searching the internet for a possible approach my first results point in the direction of methods for cluster validation (subsequently I found out that this problem is also evident when it comes to consensus clustering). In a research paper from Lange et. al. “[Stability-based validation of clustering solutions](http://www.ml.inf.ethz.ch/publications/lange.neco_stab.03.pdf)” the authors describe a sampling based approach for evaluating the stability of clustering solutions. They therefore have to compare partitions from different runs over the data which matches exactly the same question I described above. Here a method from [Kuhn](http://mike.mccreavy.com/hungarian-assignment-problem.pdf) called the ‘Hungarian method’ for minimum weighted bipartite matching is mentioned which should solve the assignment of two different clustering solutions onto each other. As a result we could rearrange the labels from one clustering.

But what is the idea of formulating this correspondence problem as an optimization exercise? You can relate this type of question to weighted bipartite graphs and subsets of them. In a bipartite graph a matching is a subset of the edges so that no two edges in the subset share a common vertex. It is called a minimum weighted bipartite matching when the graph is a weighted bipartite graph and the sum of all edges in the subset is minimal. This could be represented as a distance matrix having the dimension of the number of clusters where the value between two instances depicts the agreement between these two partitions (one constraint for this approach is that there is the same number of partitions in both clustering’s). So, one clustering is represented by columns and the other one by row or vice versa. The agreement can be calculated as follows: Calculate the number of elements in the intersection of the two partitions and subtract it twice from the sum of the number of elements in both clusters. The notion behind this computation is that if all elements are in the intersection, the value is zero and hence it is very likely that these two partitions are mapped on each other. The higher the value the more different are the two partitions. One approach for calculating these distance matrix in R looks like the following (herby we us the method solve\_LSAP from the package clue, where some additional explanations could also be found inside the associated paper “A CLUE for CLUster Ensembles”):

A simple example will illustrate the matching:

The mapping resulting for the example should be done in the following way: rename cluster 1 from the first partition as 2, cluster 2 as 3, cluster 3 as 1 and cluster 4 keeps its name.   
The last steps is to write some bit of code to carry out the mapping automatically. Looking at the final plot reveals, that besides the inherent instability from the k-mean method the clustering looks approximately identical and thus the mapping was done successfully.   
Hopping that this code will help also others I’m looking forward for any helpful comment.