Graph matching algorithm

Chang Su

April 2021

1 Algorithm

Here is a brief description of the algorithm used in our graph matching analysis. This procedure is an application of the Hungarian algorithm. The Matlab implementation is available at https://github.com/changchangsu/pairwise_graph_matching.

1.1 Notation

- Let M_1, M_2 denote the two precision matrices to be matched. These are symmetric matrices.
- Let Π denote the permutation matrix used to match M_1 to M_2. By the definition of permutation matrix, there is only one '1' in any row or any column. In the formula below, $\Pi_{ij} = 1$ implies mapping column j to column i, or mapping row j to row i. Note that in general a permutation matrix does not have to be symmetric. For example, consider three columns: i, j, k. The following swapping is possible: column $j \to$ column i; column $i \to$ column k; column $k \to$ column j.
- Let D denote the penalty matrix where D_{ij} stands for the penalty for matching ROI j to ROI i. In particular, in our implementation each entry D_{ij} can be either 1 or 0, such that only ROI pairs with nonzero values will be penalized.
- λ is the general penalty weight.

1.2 Objective function

The objective function is defined as

$$f_1(\Pi) = \|\Pi^\top M_1 \Pi - M_2\|_F^2 + 2\lambda \text{trace}(\Pi^\top D), \tag{1}$$

which can be simplified into

$$f_1(\Pi) = \|M_1\|_F^2 + \|M_2\|_F^2 - 2\text{trace}(\Pi^\top (M_1^\top \Pi M_2 - \lambda D)). \tag{2}$$

Intuitively, $f_1(\Pi)$ evaluates the difference between permuted precision matrix $\Pi^\top M_1 \Pi$ and M_2, with penalty λ on matching column j to column i or row j to row i.

As the first two terms in 2 are constants, we can also define the objective function as

$$f_2(\Pi) = \mathrm{trace}(\Pi^\top (M_1^\top \Pi M_2 - \lambda D)). \tag{3}$$

The goal for optimization is then to maximize $f_2(\Pi)$. We use Hungarian algorithm to accomplish this, which is as simple as two steps:

1. Initialize Π as $\Pi_0 = \mathrm{diag}(1)$.
2. For $t = 1, \ldots, t_m$, solve $\Pi_t = \mathrm{argmax}_\Pi \left\{ \mathrm{trace}(\Pi^\top (M_1^\top \Pi_{t-1} M_2 - \lambda D)) \right\}$.

The second pair is solved by matchpair() in Matlab. As we observe almost no difference between doing 1 and more steps of updates, in our implementation we fix $t = 1$ to save computational time.