DATA.STAT.770

Exercise set 14

H2:

Kamada-Kawai cost function

$$Cost = \sum_{i<j} \frac{1}{2} k_{ij} (|y_i - y_j| - l_{ij})^2$$

where

$l_{ij} = L * d_{ij}$ (desired on-screen length of edge (ij))
$k_{ij} = \frac{K}{d_{ij}^2}$ (strength of the spring/edge)
$d_{ij} =$ length of the shortest parth from i to j
$y_i =$ on-screen pos. of node i

Sammon mapping cost function

$$\sigma_r = \sum_{i<j} \frac{1}{p_{ij}} (p_{ij} - d_{ij}(X))^2$$

where

$p_{ij} = d(y_i, y_j)$ (original distance between samples)
$d_{ij}(X) = d(x_i, x_j)$ (distances after projectoin on X)

Similarities between cost functions:

- Both methods compare projected distances to original ones simply by L2-norm
- Both methods prioritize the preservation of small distances – In both methods, the cost effect related to 'small distance' errors is relatively larger than in case of 'large distance' errors (k_{ij} vs $\frac{1}{p_{ij}}$).
- In general, both methods can be thought to be following a force-directed updating cycle in optimization stage.

Differences:

- The target distance p_{ij} in Sammon mapping is more 'well defined' since it has a solid ground truth value. However, in Kamada-Kawai, the target on-screen distance has to be estimated based on scaled length of the shortest path between nodes.
- In Sammon mapping we are always dealing with same kind of metric space ($\mathbb{R}^d$), but in Kamada-Kawai our original space and projection space are completely different (space defined for graphs vs $\mathbb{R}^d$, where $d \in [2,3]$)
- Kamada-Kawai introduces some hyperparameters that have large effect in end results.
- Kamada-Kawai uses separate factor k_{ij} to prioritize small distances, instead of l_{ij}. Sammon mapping uses simply the inverse of the original distances. Because k_{ij} increases more rapidly in small distances compared to l_{ij}, Kamada-Kawai empathises small distance preservation relatively more than in Sammon mapping.