KLIC

Stephen Coleman

25/11/2019

Kernel learning integrative clustering

Elevator pitch

Takes independent clusterings of the individuals (i.e. local clusterings) and combines these for a global clustering. Allows different local clusterings to contribute with different strenghts to the final clustering. No specifications on the type of model used to create the original clusterings.

Intro

Kernel Learning Integrative Clustering (KLIC) is a **sequential analysis** method of integrative clustering. This in comparison to **post-processing** or **joint** methods. Are these labels a real thing?

KLIC is an extension of Cluster-of-Cluster Analysis (COCA). If one has L different datasets of measurements for the same N individuals to which one applies independent clustering methods, COCA then turns the similarity matrices that result from this into a global clustering by combining the matrices in a method similar to Consensus Clustering (the original paper). KLIC extends this by allowing different similarity matrices to have different weights in how they contribute to the global clustering. In short, KLIC applies multiple kernel k-means clustering to similarity matrices generated for individual datasets.

To understand KLIC one must understand the following:

- COCA;
- the kernel trick;
- k-means clustering; and
- multiple kernel k-means clustering.

The kernel trick

This is a computational trick to avoid operations. It aims to do an analysis in a high-dimensional space while only considering calculations in the original space.

Definition: Positive definite kernel

or simply a kernel, δ, is a symmetric map:

$$\delta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$

which for all $x_1, \ldots, x_N \in \mathcal{X}$, the matrix $\mathbf{\Delta}$ defined by entries $\mathbf{\Delta}_{ij} = \delta(x_i, x_j)$, is positive semi-definite.

Definition: Kernel matrix

or **Gram matrix**, $\mathbf{\Delta}$, is the positive semi-definite matrix defined by a kernel δ applied to data $\mathcal{X} = (x_1, \ldots, x_N)$ with entries $\mathbf{\Delta}_{ij} = \delta(x_i, x_j)$.

Definition: Feature map

For each kernel δ there exists a **feature map** $\phi(\cdot)$, which maps the original data $\mathcal{X} = (x_1, \ldots, x_N)$ to some new feature space taking values in some inner product of $\mathcal{X}$ defined by:

$$\delta(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle.$$

Thus if one is interested in working in some feature space that can be defined in terms of inner products, one may use kernels to avoid computations in analysing the data in said space.

k-means clustering

k-means clustering assignes N points to K different clusters. These clusters are defined by the points assigned to them - they are the centroid of the assigned points. The aim of this algorithm is to minimise the sum of a ll squared distance between the points and their assigned centroid.

Let m_k denote the kth centroid. Then if $\mathbf{Z}$ is the $N \times K$ classification matrix, with

$$z_{ik} = \begin{cases} 1 \text{ if point } x_i \text{ is assigned to cluster } k, \\ 0 \text{ else.} \end{cases}$$

In classic k-means clustering, each point can only be assigned to one cluster. We use the following notation:

$$\sum_{k=1}^{K} z_{ik} = 1 \text{ for all } i \in \{1, \ldots, N\}$$

$$N_k = \sum_{i=1}^{N} z_{ik}$$

$$m_k = \frac{1}{N_k} \sum_{i=1}^{N} z_{ik} x_i \text{ for all } k \in \{1, \ldots, K\}$$

We can consider this an optimisation problem.

$$\underset{\mathbf{Z}}{\operatorname{argmin}} \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik} ||x_i - m_k||_2^2$$

If we redefine this problem in some feature space defined by the feature map $\phi(\cdot)$, we can use the kernel trick in this context. First we write the problem in terms of the feature map:

$$\underset{\mathbf{Z}}{\operatorname{argmin}} \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik} ||\phi(x_i) - m_k^*||_2^2$$

$$m_k^* = \frac{1}{N_k} \sum_{i=1}^{N} z_{ik} \phi(x_i)$$

If we define the $K \times K$ matrix $\mathbf{L}$ with $(k,k)th$ entries of N_k and 0's elsewhere and the gram matrix $\mathbf{\Delta}$ as the matrix with $(i,j)th$ entries $\delta(x_i, x_j)$, then, according to Gönen and Margolin (2014), this can be rephrased as a trace maximisation problem:

$$\underset{\mathbf{Z}}{\text{argmax}}\ \text{tr}(\mathbf{L}^{\frac{1}{2}}\mathbf{Z}^{\mathbf{T}}\mathbf{\Delta}\mathbf{Z}\mathbf{L}^{\frac{1}{2}} - \mathbf{\Delta})$$

This is subject to the contraints:

$$\begin{aligned}\mathbf{Z}\mathbf{1}_K &= \mathbf{1}_N \\ z_{ik} &\in \{0, 1\}\end{aligned}$$

The binary variables, z_{ik}, make this problem very difficult to solve (Gönen and Margolin 2014). Relaxing this constraint and letting $\mathbf{Z}\mathbf{L}^{\frac{1}{2}} = \mathbf{H}$ and letting $\mathbf{H}$ take arbitrary real values constrained by $\mathbf{H}^{\mathbf{T}}\mathbf{H} = \mathbf{I_k}$ makes the optimisation problem more feasible:

$$\underset{\mathbf{H}}{\text{argmax}}\ \text{tr}(\mathbf{H}^{\mathbf{T}}\mathbf{\Delta}\mathbf{H} - \mathbf{\Delta}).$$

One can solve this by performing Kernel-PCA on the Gram matrix $\mathbf{\Delta}$ and setting $\mathbf{H}$ to the k largest eigevalues. To fianlly acquire a clustering solution, one can normalise all rows of $\mathbf{H}$ to be on the unit sphere and then implement k-means clustering on this normalised matrix.

References

Gönen, Mehmet, and Adam A Margolin. 2014. "Localized Data Fusion for Kernel K-Means Clustering with Application to Cancer Biology." In *Advances in Neural Information Processing Systems*, 1305–13.