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 assigns each of N points to *one* of K different clusters. For cluster means $\mu = (\mu_1, \ldots, \mu_K)$, data $X = (x_1, \ldots, x_n)$ and cluster alloation matrix $\mathbf{Z}$, the objective function for k-means clustering is:

$$C(\mathbf{Z}, \mu) = \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik} ||x_i - \mu_k||^2.$$

$\mathbf{Z}$ is a $N \times K$ binary matrix, with

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

We use the following notation:

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

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

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

One can solve for C by iterating over:

$$z_{ik} = \mathbb{I}(k = \underset{\mathbf{k'}}{\operatorname{argmin}} ||x_i - \mu_{k'}||_2^2),$$

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N} z_{ik} x_i.$$

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:

$$C(\mathbf{Z}, \mu) = \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik} ||\phi(x_i) - \mu_k||^2.$$

Allocation and cluster means are as previously stated, except x_i is replaced with the features $\phi(x_i)$.

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

Define the $K \times K$ matrix $\mathbf{L}$ with $(k,k)th$ entries of $1/N_k$ and 0's elsewhere (i.e. the kth diagonal correspond the inverse of the number of points assigned to the kth cluster). Define also the gram matrix $\mathbf{\Delta}$ as the matrix with $(i,j)th$ entries $\delta(x_i, x_j)$ and the matrix Φ where $\Phi_{ij} = \phi_i(x_j)$. Define also the matrix $\mathbf{M}$:

$$\mathbf{M} = \Phi\mathbf{Z}L\mathbf{Z}^T.$$

$$\begin{aligned} L\mathbf{Z}^T &= \begin{bmatrix} \frac{1}{N_1} & & \\ & \ddots & \\ & & \frac{1}{N_K} \end{bmatrix} \begin{bmatrix} z_{11} & \cdots & z_{N1} \\ \vdots & \ddots & \vdots \\ z_{K1} & \cdots & z_{NK} \end{bmatrix} \\ &= \begin{bmatrix} \frac{z_{11}}{N_1} & \cdots & \frac{z_{N1}}{N_1} \\ \vdots & \ddots & \vdots \\ \frac{z_{1K}}{N_K} & \cdots & \frac{z_{NK}}{N_K} \end{bmatrix}, \\ \mathbf{Z}L\mathbf{Z}^T &= \begin{bmatrix} \sum_{k=1}^{K} \frac{z_{1k}z_{1k}}{N_k} & \cdots & \sum_{k=1}^{K} \frac{z_{1k}z_{Nk}}{N_k} \\ \vdots & \ddots & \vdots \\ \sum_{k=1}^{K} \frac{z_{Nk}z_{1k}}{N_k} & \cdots & \sum_{k=1}^{K} \frac{z_{Nk}z_{Nk}}{N_k} \end{bmatrix} \end{aligned}$$

As z_{ik} is a binary variable with the constraint that $\sum_{k=1}^{K} z_{ik} = 1$, this means that the only non-zero entries are the entries for which there is a common allocation. If one re-arranges the rows of $\mathbf{Z}L\mathbf{Z}^T$ such that points assigned to the same cluster are contiguous, i.e. into a block diagonal matrix, then the matrix has the form:

$$\mathbf{Z}L\mathbf{Z}^T = \begin{bmatrix} \frac{1}{N_1} & \cdots & \frac{1}{N_1} & & & & & & \\ \vdots & \ddots & \vdots & & & & & & \\ \frac{1}{N_1} & \cdots & \frac{1}{N_1} & & & & & & \\ & & & \frac{1}{N_2} & \cdots & \frac{1}{N_2} & & & \\ & & & \vdots & \ddots & \vdots & & & \\ & & & \frac{1}{N_2} & \cdots & \frac{1}{N_2} & & & \\ & & & & & & \ddots & & \\ & & & & & & & \frac{1}{N_K} & \cdots & \frac{1}{N_K} \\ & & & & & & & \vdots & \ddots & \vdots \\ & & & & & & & \frac{1}{N_K} & \cdots & \frac{1}{N_K} \end{bmatrix}$$

We consider the order of the points as stored in X to be independent of the analysis and consider the rows of each matrix rearranged to give the above block diagonal structure.

This means that multiplying by X^T or Φ gives a $p \times N$ matrix with the jth column being the mean of the cluster the jth point is assigned to. The jth column is of the form:

$$\text{col}_j(\Phi Z L Z^T) = \begin{bmatrix} \frac{1}{N_{c_j}} \sum_{i=1}^{N} z_{ic_j} \phi_j(x_i) \\ \vdots \\ \frac{1}{N_{c_j}} \sum_{i=1}^{N} z_{ic_j} \phi_j(x_i) \end{bmatrix} = \mu_{c_j}$$

This allows us to write the objective function in the form $C = \mathbf{tr}[(\Phi - M)(\Phi - M)^T]$(Gönen and Margolin 2014).

The jth diagonal entry of this matrix is given by:

$$C_{jj} = \sum_{j=1}^{p} (x_{ij} - \mu_{c_i j})^2.$$

Thus, one can see how the statement of the function as a trace minimisation problem is merely a restatement of our original objective function.

Note that:

$$\begin{aligned} Z^T Z &= \begin{bmatrix} \sum_{i=1}^{N} z_{i1} z_{i1} & \cdots & \sum_{i=1}^{N} z_{i1} z_{iK} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{N} z_{i1} z_{iK} & \cdots & \sum_{i=1}^{N} z_{iN} z_{iK} \end{bmatrix} \\ &= \begin{bmatrix} N_1 & & \\ & \ddots & \\ & & N_K \end{bmatrix} \\ &= L^{-1} \end{aligned}$$

This means that $(ZLZ^T)^2 = ZLZ^T$ which is the definition of a projection. Similarly $I - ZLZ^T$ is a projection on the complement. This combined with the fact that $\mathbf{tr}[AB] = \mathbf{tr}[BA]$ means that we can simplify the objective function:

$$\begin{aligned} C &= \mathbf{tr}[\Phi(I - ZLZ^T)^2 \Phi^T] \\ &= \mathbf{tr}[\Phi(I - ZLZ^T)\Phi^T] \\ &= \mathbf{tr}[\Phi\Phi^T] - \mathbf{tr}[\Phi Z L^{1/2} L^{1/2} Z^T \Phi^T] \\ &= \mathbf{tr}[\Phi^T\Phi] - \mathbf{tr}[L^{1/2} Z^T \Phi^T \Phi Z L^{1/2}] \\ &= \mathbf{tr}[\Delta] - \mathbf{tr}[L^{1/2} Z^T \Delta Z L^{1/2}] \end{aligned}$$

Here we have let $L^{1/2}$ be the matrix with entries of the square root of the diagonal entries of L.

As it is only the second part of the equation that depends upon the clustering matrix Z, one can formulate the equivalent maximisation problem:

$$\max \mathbf{tr}[L^{\frac{1}{2}} Z^T \Delta Z L^{\frac{1}{2}}],$$

subject to the contraints:

$$\begin{aligned} \mathbf{Z1}_K &= \mathbf{1}_N \\ z_{ik} &\in \{0, 1\} \end{aligned}$$

The binary variables, z_{ik}, make this problem very difficult to solve (Gönen and Margolin 2014). By setting $H = ZL^{1/2}$, we can restate the problem with matrix H which is no longer constrained to be binary, but as $Z^TZ = L^{-1}$ does have the constraint that $H^TH = I$. One hopes that a clustering solution may then be derived as an additional step after the optimisation problem is solved. Thus the relaxed formulation of the problem is:

$$\max_H \mathbf{tr}[H^T \Delta H]$$
$$\text{subject to } H^TH = I$$

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. This is done as so:

$$\hat{H} = \frac{H_{ik}}{\sqrt{\sum_{k=1}^K H_{ik}^2}}$$

One can then implements k-means clustering on this normalised matrix.

Multiple kernel k-means clustering

Gönen and Margolin (2014) extend this concept of kernel k-means clustering to multiple kernel k-means clustering. Considered in the context of *multiview learning*, this method assumes we have L different feature representations each with its own mapping function, i.e. $\{\Phi_m(\cdot)\}_{l=1}^L$. The aim is to combine these different views in a non-naive way; we wish to avoid simple concatenation. If instead we use a weighted sum such that the views that contribute the most signal are given the greatest weights with the restriction that weights are positive and sum to 1. This corresponds to replacing $\phi(x_i)$ with:

$$\phi_\theta(x_i) = \begin{bmatrix} \theta_1\phi_1(x_i)^T \\ \vdots \\ \theta_L\phi_L(x_i)^T \end{bmatrix}.$$

Here $\theta \in \mathbb{R}_+^L$ is the kernel weights that need to be optimised. The kernel function over the weighted mapping function becomes:

$$\begin{aligned}\delta_\theta(x_i, x_j) &= \langle \phi_\theta(x_i), \phi_\theta(x_j) \rangle \\ &= \sum_{l=1}^L \langle \theta_l\phi_l(x_i), \theta_l\phi_l(x_j) \rangle \\ &= \sum_{l=1}^L \theta_l^2 \delta_l(x_i, x_j)\end{aligned}$$

Where previously we could discard the $\mathbf{tr}[\Delta]$ component of the trace minimisation problem as being independant of the clustering matrix, it will now be dependent upon the kernel weights. Letting $\Delta_\theta = \sum_{l+1}^L \theta_l^2 \Delta_l$, our objective function becomes:

$$C = \mathbf{tr}[\Delta_\theta] - \mathbf{tr}[L^{1/2}Z^T\Delta_\theta Z L^{1/2}]$$
$$\text{subject to } H^TH = I, \sum_{l=1}^L \theta_l = 1$$

Coclustering matrices as kernels

Kernel learning integrative clustering

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 27*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, 1305–13. Curran Associates, Inc. http://papers.nips.cc/paper/5236-localized-data-fusion-for-kernel-k-means-clustering-with-application-to-cancer-biology.pdf.