FEATURE SELECTION IN CLUSTERING PROBLEM

TECHNICAL UPDATES:

As a statistical tool, clustering analysis has been widely applied to a variety of scientific areas such as vector quantization,data mining, image processing,statistical data analysis, etc In this paper the clustering problem is formulated as a combination of clustering and feature selection based on Gaussian Mixture Model, which is further optimized by Expectation-maximization Algorithm.

Often, Expectation maximization Algorithm provides a general solution for the parameter estimate in density mixture models.Nevertheless, it needs to pre-assign an appropriate number of density components, that is, the number of clusters. Roughly, the mixture may overfit the data if too many components are utilized, whereas a mixture with too few components may not be flexible enough to approximate the true underlying model. Subsequently, the EM almost always leads to a poor estimate result if the number of components is misspecified. Unfortunately, from the practical viewpoint, it is hard or even impossible to know the exact cluster number in advance.

I here propose a method , which learns the model parameter via maximizing a weighted likelihood. Under a specific weight design ,we then introduce Rival Penalized EM (RPEM) algorithm for density mixture clustering.The RPEM algorithm learns the parameter by making mixture components compete each other at each time step.Comparing Rival Penalized EM algorithm to EM algorithm, the RPEM fades out the redundant densities from a density mixture during parameter learning process. Hence, RPEM automatically selects an appropriate number of densities.

The convergence speed of the RPEM relies on the value of the learning rate. Often, by a rule of thumb, we arbitrarily set the learning rate at a small positive constant. If the value of learning rate is assigned too small, the algorithm will converge at a very slow speed. On the contrary, if it is too large, the algorithm may even oscillate. In general, it is a nontrivial task to assign an appropriate value to the learning rate, although we can pay extra effort to make the learning rate dynamically change over time.

OVERVIEW OF MAXIMUM WEIGHTED LIKELIHOOD (MWL) LEARNING FRAMEWORK

Suppose that an input x ε R^d comes from the following density mixture model:

$$\begin{gathered} P(\mathbf{x} \mid \boldsymbol{\Theta}) = \sum_{j=1}^{k} \alpha_j p\left(\mathbf{x} \mid \theta_j\right), \quad \sum_{j=1}^{k} \alpha_j = 1, \\ \alpha_j > 0, \quad \forall 1 \leq j \leq k, \end{gathered} \tag{1}$$

Where Θ is the parameter set of $\{ \alpha_J , \theta_J \}^k_{J=1}$. Furthermore, Furthermore, is the number of components, is the mixture proportion of the J th component, and p(x | θ_J) is a multivariate probability density function of x parameterized by θ_J . As long as we know the value of Θ , an input x can be classified into a certain cluster via its posterior probability:

$$h(j \mid \mathbf{x}, \boldsymbol{\Theta}) = \frac{\alpha_j p\left(\mathbf{x} \mid \theta_j\right)}{P(\mathbf{x} \mid \boldsymbol{\Theta})} \tag{2}$$

Using the winner-take-all rule, that is, assigning an input x to cluster c if c =argmax$_J$h(J | x ,Θ) taking its soft version which assigns x to cluster J with the probability h(J | x, Θ) . Therefore, how to estimate the parameter set Θ , particularly without knowing the correct value of k in advance, is a key issue in density mixture clustering. In the MWL learning framework, the parameter set is learned via maximizing the following Weighted Likelihood (WL) cost function:

$$l(\boldsymbol{\Theta}) = \omega(\boldsymbol{\Theta}; \mathbf{x}) + \nu(\boldsymbol{\Theta}; \mathbf{x}) \tag{3}$$

$$\begin{aligned} \omega(\boldsymbol{\Theta}; \mathbf{x}) &= \int \sum_{j=1}^{k} g(j \mid \mathbf{x}, \boldsymbol{\Theta}) \ln \left[\alpha_j p\left(\mathbf{x} \mid \theta_j\right)\right] dF(\mathbf{x}), \\ \nu(\boldsymbol{\Theta}; \mathbf{x}) &= -\int \sum_{j=1}^{k} g(j \mid \mathbf{x}, \boldsymbol{\Theta}) \ln h\left(j \mid \mathbf{x}, \boldsymbol{\Theta}\right) dF(\mathbf{x}), \end{aligned} \tag{4}$$

Where g(J | x, Θ) ’s are the designable weights satisfying two conditions:

1. $\sum^{k}_{J=1}$ (J | x,Θ) = 1
2. For all, J , g(J | x, Θ) =0 , if h(J | x, Θ) =0

$$Q(\boldsymbol{\Theta};\mathbf{X}) = \omega(\boldsymbol{\Theta};\mathbf{X}) + \nu(\boldsymbol{\Theta};\mathbf{X}) \qquad (5)$$

$$\begin{aligned} \omega(\boldsymbol{\Theta};\mathbf{X}) &= \frac{1}{N}\sum_{t=1}^{N}\sum_{j=1}^{k} g\left(j \mid \mathbf{x}_t, \boldsymbol{\Theta}\right) \ln\left[\alpha_j p\left(\mathbf{x}_t \mid \theta_j\right)\right], \\ \nu(\boldsymbol{\Theta};\mathbf{X}) &= -\frac{1}{N}\sum_{t=1}^{N}\sum_{j=1}^{k} g(j \mid \mathbf{x}_t, \boldsymbol{\Theta}) \ln h(j \mid \mathbf{x}_t, \boldsymbol{\Theta}). \end{aligned} \qquad (6)$$

Moreover, the weights g(J | x ,Θ) ’s have been generally designed as:

$$g(j \mid x_t, \Theta) = (1+\varepsilon_t)\, I(j \mid x_t, \Theta) - \varepsilon_t h(j \mid x_t, \Theta) \qquad (7)$$

Subsequently, under a specific weight design, the papers [1,2] have presented the RPEM to learn Θ via maximizing the WL function of (5) using a stochastic gradient ascent method, which is able to fade out the redundant densities gradually from a density mixture. Consequently, it can automatically select an appropriate number of density components in density mixture clustering. We summarize the main steps of the RPEM in the following Algorithm. The experiments have shown the superior performance of the RPEM on automatic model selection; its convergence speed, however, relies on the value of learning rate.

ALGORITHM - RIVAL PENALIZED EXPECTATION MAXIMIZATION ALGORITHM

Initialization :
Given a specific k(k ≥ k* , k* is the true number of clusters), initialise the parameter Θ .

Step 1 :
Given the current input xt and the parameter estimate ,written as Θ^n , compute $h(j \mid x_t, \Theta^{(n)})$’s and $g(j \mid x_t, \Theta^{(n)})$ via (2) and (7), respectively.

Step 2:

Given , $h(j \mid x_t, \Theta^{(n)})$’s and $g(j \mid x_t, \Theta^{(n)})$, we Θ update by,

$\Theta^{(n+1)} = \Theta^{(n)} + \eta(\omega_t(\Theta; x_t)/\Theta)|_{\Theta^{(n)}}$, where η is a small positive learning rate.

Step 3:
Let n=n+1 and go to step 1 for the next iteration until Θ is converged.

Here I have proposed a MWL learning framework from the ML, through which a new RPEM algorithm has been proposed for density mixture clustering. The RPEM learns the density parameters by making mixture components compete with each other at each time step. Not only are the associated parameters of the winning density component updated to adapt to an input, but also all rivals' parameters are penalized with the strength proportional to the corresponding posterior density probabilities.

Compared to the EM algorithm, this intrinsic rival penalization mechanism enables the RPEM to automatically select an appropriate number of densities by fading out the redundant densities from a density mixture.Therefore using RPEM instead of EM may increase the accuracy and can be considered as a technical update in the model proposed in the paper.