Mixture of GANs for Clustering

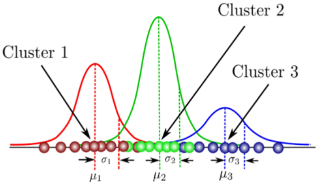
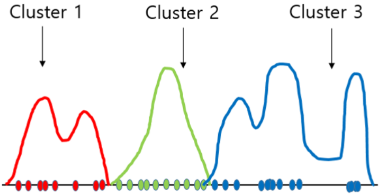
Y.Yu et al.(2018 IJCAI), cited by 8.

Presented by Insung Kong

Seoul National University

May 16, 2020

Summary

	Gaussian Mixture Model	GAN Mixture Model
Idea		
Optimize	EM Algorithm	ε −EM Algorithm

Introduction

- Among numerous clustering methods, **Gaussian Mixture Model** is a very classical and elegant one, which has still being consistently improved.
- But it has a fundamental **assumption** that the data is composed by **Gaussian distributions**. However, real data sets rarely satisfy this assumption, on which GMM can result in poor clusters.
- The ability of modeling **complex distributions** making **GAN** a potential replacement of the Gaussian distribution for learning mixture models.

Gaussian Mixture Model and EM Algorithm

Gaussian Mixture Model consists of N Gaussian components,

$$p(x) = \sum_{i=1}^{N} \alpha_i \mathcal{N}(x; \mu_i, \Sigma_i)$$

where $x \in \mathbb{R}^d$, $\boldsymbol{\mu}_i \in \mathbb{R}^d$, $\boldsymbol{\Sigma}_i \in \mathbb{R}^{d \times d}$, and $\sum_{i=1}^{N} \alpha_i = 1$

EM Algorithm

For $(X, Z) \sim p(X, Z|\theta)$, X : observable variable, Z : hidden variable

1. **E step** : Calculate the $l(\theta|X, \theta_{m-1}) = E_{Z \sim p_{\theta_{m-1}}}(l(\theta|Z)|X)$
2. **M step** : Maximize $l(\theta|X, \theta_{m-1})$

Gaussian Mixture Model and EM Algorithm

Let X : observable instances, D : Data $= \{x_1, x_2, ..., x_M\}$, $\theta = \{\alpha_i, \mu_i, \Sigma_i\}$, $z_{ij} = 1$ **iff** x_j is generated by i-th gaussian model.$(i = 1, \cdots N, j = 1, \cdots M)$

Then for j'th sample,

$$p(x_j, z_j|\theta) = \sum_{i=1}^{N} \alpha_i \mathcal{N}\left(x; \mu_i,\ \Sigma_i\right) I(z_{ij} = 1)$$

Apply to EM Algorithm,

1. **E step** : Findes a guess of q(Z) accroding to the posterior

$$p\left(z_{ij}|\boldsymbol{x}_j, \boldsymbol{\theta}^{(t)}\right) = \frac{p\left(\boldsymbol{x}_j, z_{ij}|\boldsymbol{\theta}^{(t)}\right)}{\sum_{i=1}^{N} p\left(\boldsymbol{x}_j, z_{ij}|\boldsymbol{\theta}^{(t)}\right)}$$

2. **M step** : $\boldsymbol{\theta}^{(t+1)} = \arg\max_{\theta} E_{\boldsymbol{Z}\sim p\left(Z|D,\boldsymbol{\theta}^{(t)}\right)} \ln(p(D, Z|\boldsymbol{\theta}))$

which can be calculated easily.

Drawback of Generalized mixture Model with EM

Assume we replace Gaussian model to other complex model.

- First, set initial model $\theta^{(0)}$ randomly. Then $p\left(z|D, \boldsymbol{\theta}^{(0)}\right)$ is calculated.
- Let $\boldsymbol{z}^{(0)} = \text{argmax}_{\boldsymbol{z}}\, p\left(\boldsymbol{z}|D, \boldsymbol{\theta}^{(0)}\right)$. If model can be **arbitarily capable**, there exists θ^* s.t. $p\left(z^{(0)}|D, \boldsymbol{\theta}^*\right) = 1$ and otherwise 0.
- $\theta^{(1)}$ become θ^*.
- Since $\underset{z}{\text{argmax}}\, p\left(z|D, \boldsymbol{\theta}^{(1)}\right) = \boldsymbol{z}^{(0)} = \underset{z}{\text{argmax}}\, p\left(\boldsymbol{z}|D, \boldsymbol{\theta}^{(0)}\right)$, we have $\theta^{(1)} = \theta^{(2)}$ and thus the algorithm has converged.
- That would be likely to happen for using GANs as the model.

ϵ-EM Algorithm

EM Algorithm

For $(X, Z) \sim p(X, Z|\theta)$, X : observable variable, Z : hidden variable

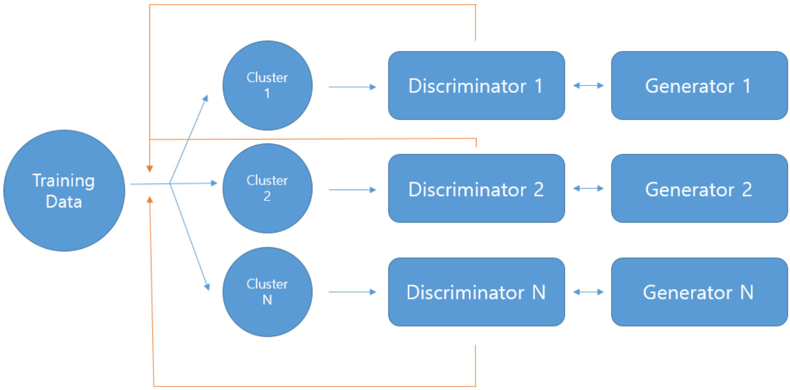
1. **E step** : Calculate the $l(\theta|X, \theta_{m-1}) = E_{Z \sim p_{\theta_{m-1}}}(l(\theta|Z)|X)$
2. **M step** : Maximize $l(\theta|X, \theta_{m-1})$

- In E step, we match the distribution $q(Z)$ to $p(Z|X, \theta^{(t)})$, so that $KL(q||p) = 0$
- However, we don't have to exactly match $q(Z)$ to $p(Z|X, \theta^{(t)})$.
- Specifically, for $q^{(t)}(Z)$ such that $KL(q^{(t)}||p^{(t)}) = \epsilon_t > 0$ and $\lim_{t \to +\infty} \sum_{i=0}^{t-1} \epsilon_i < \infty$, the algorithm converges.(Detail : appendix)

ϵ-EM Algorithm

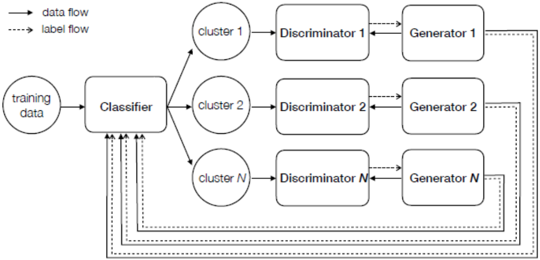
1. **ϵ-E step** : Calculate the $\tilde{l}(\theta|X, \theta_{m-1}) = E_{Z \sim \tilde{p}_{\theta_{m-1}}}(l(\theta|Z)|X)$
2. **M step** : Maximize $\tilde{l}(\theta|X, \theta_{m-1})$

GAN Mixture Model with EM Algorithm



1. **E step** : From current model $\theta^{(t)}$, assign training data to the cluster using discriminators.
2. **M step** : For the clusterd data $D = \{D_1, D_2, ..., D_N\}$, train i-th GAN model on D_i

GAN Mixture Model with ϵ-EM Algorithm



1. **ϵ-E step** :
   (1) From current model $\theta^{(t)}$, sample a data set $S^{(t)} = \{(\tilde{\boldsymbol{x}}_i, y_i)_{i=0}^n\}$ from the GAN generators, where $\tilde{\boldsymbol{x}}_i$ is generate by k-th generator and $y_i = k$.
   (2) Train a (not so perfect) classifier h_q from S.
   (3) Assign the cluster of each training instance x_i by $h_q(x_i)$.
2. **M step** : For the clusterd data $D = \{D_1, D_2, ..., D_N\}$, train i-th GAN model on D_i for several iterations.

GAN Mixture Model with ϵ-EM Algorithm

α: learning rate.
m: training set size.
N: cluster number.
n_{epoch}: number of epoch for GANs.
σ_t, number of augmented data points.

randomly divide D into $\{D_1^{(0)}, \ldots, D_N^{(0)}\}$
$\boldsymbol{\theta}_{D_i^{(0)}}^{(0)}, \boldsymbol{\theta}_{G_i}^{(0)} \leftarrow$ trainWGAN($D_i^{(0)}$, n_{epoch}) for each $i = 1, \ldots, N$
$t = 0$
while $\boldsymbol{\theta}_E$ not converged **do**
  $t = t + 1$
  $S = \{(\tilde{\boldsymbol{x}}_i, y_i)\}_{i=1}^m$ is sampled from $\{G_j\}_{j=1}^N$ each with probability $|D_i^{(t-1)}|/|D|$.
  $g_{\theta_h} \leftarrow \nabla_{\theta_h} [\frac{1}{m} \sum_{i=1}^m cross\_entropy(f_{\boldsymbol{\theta}_h}(\tilde{\boldsymbol{x}}_i), y_i)]$
  $\theta_h \leftarrow \theta_h + \alpha \cdot$ RMSProp(θ_h, g_{θ_h})
  assign D as $\{D_i\}_{i=1}^N$ by using $h_q(D; \theta_h)$
  for $i = 1$ to N **do**
    add σ_t instances from $D - D_i$ with highest posterior for cluster i by h_q to D_i
    $\boldsymbol{\theta}_{D_i}^{(t)}, \boldsymbol{\theta}_{G_i}^{(t)} \leftarrow$ trainWGAN($D_i^{(t)}$, n_{epoch}) for each $i = 1, \ldots, N$
  end for
end while

Experiments

Models

- GAN Mixture Model(GANMM)
- Gaussian Mixture Model(GMM)
- Deep Embedded Clustering(DEC)
- GAN Mixture Model with EM(GANMM-EM)

Evaluate Metrics : using the data with **oracle labels**.

- Purity $= \sum_{i=1}^{N} \frac{m_i}{m} \max_j \frac{m_{ij}}{m_i}$
  m_i : size of cluster i, m_{ij} : number of class j data in cluter i
- ARI $= \frac{m_{11}+m_{00}}{m}$
  m_{11} : the number of pairs in same cluster and same oracle label
  m_{00} : the number of pairs in diff cluster and diff oracle label
- NMI $= \frac{\sum d_{h,l} \log\left(\frac{|\Omega| \cdot d_{h,l}}{d_h c_l}\right)}{\sqrt{\left(\sum_h d_h \log\left(\frac{d_h}{d}\right)\right)\left(\sum_l c_l \log\left(\frac{c_l}{d}\right)\right)}}$

Experiments

	Purity	ARI	NMI
GANMM	**0.6430±0.0045**	**0.4924±0.0059**	**0.6159±0.0038**
GMM	0.3261±0.0006	0.0991±0.0003	0.1414±0.0004
DEC	0.3065±0.0003	0.1437±0.0005	0.1935±0.0003
GANMM(EM)	0.2786±0.0019	0.0665±0.0013	0.2414±0.0004

Table 1: Comparison of clustering performance in raw feature space on MNIST.

	Purity	ARI	NMI
GANMM	**0.8908±0.0015**	**0.8361±0.0025**	**0.8654±0.0008**
GMM	0.8617±0.0008	0.7933±0.0011	0.8451±0.0002
DEC	0.8673±0.0000	0.8091±0.0000	0.8457±0.0000
GANMM(EM)	0.5243±0.0024	0.3210±0.0034	0.4701±0.0019

Table 3: Comparison of clustering performance in embedded feature space on MNIST

Appendix : ϵ-EM Algorithm

We can observe r.v X which is parameterized by θ, and Z is hidden r.v.
Object : maximize $p(X|\theta) = \sum_Z P(X, Z|\theta)$. Let $\ln p(\mathbf{X}|\theta) = LL(\theta)$.
Let q(Z) is marginal distribution of Z. Then $LL(\theta) = \mathcal{L}(q, \theta) + \mathrm{KL}(q\|p)$,
where $\mathcal{L}(q, \theta) = \sum_{\mathbf{Z}} q(\mathbf{z}) \ln \frac{p(\mathbf{X},\mathbf{Z}|\theta)}{q(\mathbf{Z})}$ and $\mathrm{KL}(q\|p) = -\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{Z}|\mathbf{X},\theta)}{q(\mathbf{Z})}$.

In $\epsilon - E$ step, find $q^{(t)}$ s.t. $KL\left(q^{(t)}\|p^{(t)}\right) = \epsilon_t > 0$
Then, $\forall q$, $\mathcal{L}(q^{(t)}, \theta^{(t)}) + \epsilon_t = LL(\theta^{(t)}) \geq \mathcal{L}(q, \theta^{(t)})$

In M step, $\theta^{(t)} = \underset{\theta}{\operatorname{argmax}}\ \mathcal{L}(q^{(t-1)}, \theta)$, hence $\mathcal{L}(q^{(t-1)}, \theta^{(t)}) \geq \mathcal{L}(q^{(t-1)}, \theta^{(t-1)})$
Therefore, we have

$$\begin{aligned} LL(\theta^{(t)}) \geq\ & \mathcal{L}(q^{(t-1)}, \theta^{(t)}) \geq \mathcal{L}(q^{(t-1)}, \theta^{(t-1)}) \\ =\ & LL(\theta^{(t-1)}) - \epsilon_{t-1} \geq LL(\theta^{(0)}) - \sum_{i=0}^{t-1} \epsilon_i \end{aligned}$$

If we keep $\lim_{t\to+\infty} \sum_{i=0}^{t-1} \epsilon_i < \infty$, $\exists\ C > 0$ s.t. for all $t > C$,
$LL(\theta^{(t)}) \geq LL(\theta^{(t-1)})$ and thus the procedure converges.