DYNAMICALLY REGULARIZED MAXIMUM LIKELIHOOD LEARNING OF GAUSSIAN MIXTURES

Jinwen Ma, Hongyan Wang

Department of Information Science, School of Mathematical Sciences
Peking University, Beijing, 100871, China

ABSTRACT

The Gaussian mixture model is widely applied in the elds of data analysis and information processing. Recently, its parameter learning with adaptive model selection, i.e., the adaptive selection of number of Gaussian distributions in the mixture for a given sample dataset, has become an attracting and interesting topic. In this paper, we propose a dynamically regularized maximum likelihood learning (DRMLL) algorithm for Gaussian mixtures with adaptive model selection. The basic idea is that the Bayesian Ying-Yang (BYY) harmony learning is interpreted as the maximum likelihood learning regularized by the average Shannon entropy of the posterior probability per sample scaled by a positive parameter. As this scale parameter dy-

Based on the analysis of BYY harmony function on Gaussian mixtures under the BI-architecture [14]-[15], the BYY harmony learning can be regarded as a kind of regularization version of the maximum likelihood (ML) learning. The regularization term is just the average Shannon entropy of the posterior probabilities per sample. In fact, the entropy regularization methods could date back to the 1980s and have been widely used in the ill-posed problems involving in model selection. In the view of model selection and ML parameter estimation, the BYY harmony function can be decomposed into the likelihood function and the entropy regularized term. However, if the regularization scale keeps constant in the way of the existing regularized ML learning approaches [16]-[18], the regularized ML learning leads to a certain deviation between its estimation and the ML or true solution. In order to overcome this problem, we can adjust the regularization scale from 1 to 0, the learning process can transform from the BYY harmony learning into the ML learning. If we further maintain the increase of the regularization scale dynamically and properly, the regularized learning process can lead to the ML estimates of the parameters with adaptive model selection on Gaussian mixtures. Oppositely, the ML learning can be regarded as a kind of regularization version of the BYY harmony learning and the dynamically regularized BYY harmony learning algorithm can be established for Gaussian mixtures [20].

In the current paper, we propose a Dynamically Regularized Maximum Likelihood Learning (DRMLL) algorithm for Gaussian mixtures with adaptive model selection. By controlling the regularization scale to dynamically decrease from 1 to 0, the DRMLL algorithm transforms from the BYY harmony learning with adaptive model selection to the conventional maximum likelihood learning. It is demonstrated by the experiments that the DRMLL algorithm can not only select the correct number of actual Gaussian distributions in a given dataset, but also obtain ML estimates of the parameters in the original mixture.

2. DRMLL ALGORITHM

In this section, we first present the dynamic regularization mechanism to be used. Then, we introduce the fixed-point algorithm for the dynamic learning process. We further discuss the dynamic evolution of the regularization scale factor. Finally, we give the complete DRMLL algorithm.

2.1. Dynamic Regularization Mechanism

According to [15], for the Gaussian mixture model $P(x|\Theta_k) = \sum_{j=1}^{k} \pi_j q(x_t|m_j, \Sigma_j)$, the corresponding BYY harmony function $J(\Theta_k)$ can be divided into two parts,

$$J(\Theta_k) = L(\Theta_k) - O_N(p(y|x)), \tag{1}$$

where the first part is just the log-likelihood function, i.e.,

$$L(\Theta_k) = \frac{1}{N}\sum_{t=1}^{N} \ln(\sum_{j=1}^{k}(\pi_j q(x_t|m_j, \Sigma_j))), \tag{2}$$

while the second is the average Shannon entropy of the posterior probability $p(y|x)$ over the sample dataset $\mathcal{D} = \{x_t\}_{t=1}^{N}$,

$$O_N(p(y|x)) = -\frac{1}{N}\sum_{t=1}^{N}\sum_{j=1}^{k} p(j|x_t)\ln p(j|x_t). \tag{3}$$

According to Eq.(1), if $-O_N(p(y|x))$ is viewed as a regularization term, the BYY harmony learning, i.e., maximizing $J(\Theta_k)$, is a regularized ML learning which has already been investigated in [17, 18] by scaling the regularization term with a small positive number. However, since they keep the regularization scale constant just as in the case of the BYY harmony learning, these approaches must suffer from inconsistent parameter estimation.

To dynamically control the regularization, we use a dynamic regularization scale factor $\lambda(\geq 0)$ and have

$$J_\lambda(\Theta_k) = L(\Theta_k) - \lambda O_N(p(y|x)). \tag{4}$$

If $\lambda = 1$, $J_\lambda(\Theta_k) = J(\Theta_k)$ is just BYY harmony function on the Bi-architecture for Gaussian mixtures. If $\lambda = 0$, $L_\lambda(\Theta_k)$ is the log-likelihood function of the Gaussian mixture model. That is, with λ decreasing from 1 to 0, maximizing $J_\lambda(\Theta_k)$ changes from the BYY harmony learning to the ML learning. Here we try to control the decreasing of λ dynamically and appropriately to realize adaptive model selection at the previous learning stage and the ML estimation at the final learning stage.

2.2. Fixed-point Learning Algorithm

At each phase of the dynamically regularized maximum likelihood learning with a particular λ, we construct a fixed-point algorithm to maximize $J_\lambda(\Theta_k)$ as follows.

For convenience, we utilize the softmax representation for π_j, i.e., $\pi_j = e^{\beta_j}/\sum_{i=1}^{K} e^{\beta_i}$, $j = 1, \cdots, k$, where $\beta_j \in (-\infty, +\infty), j = 1, \cdots, k$. Letting the derivatives of $J_\lambda(\Theta_k)$ with respect to β_j, m_j and Σ_j, respectively, be zero, we get the following fixed-point (iterative) learning algorithm:

$$\hat{\pi}_j = \frac{\sum_{t=1}^{N} p(j|x_t)\gamma_j(t)}{N}; \tag{5}$$

$$\hat{m}_j = \frac{\sum_{t=1}^{N} p(j|x_t)\gamma_j(t)x_t}{\sum_{t=1}^{N} p(j|x_t)\gamma_j(t)}; \tag{6}$$

$$\hat{\Sigma}_j = \frac{\sum_{t=1}^{N} p(j|x_t)\gamma_j(t)(x_t - \hat{m}_j)(x_t - \hat{m}_j)^T}{\sum_{t=1}^{N} p(j|x_t)\gamma_j(t)}, \tag{7}$$

where

$$\gamma_j(t) = 1 + \lambda \ln p(j|x_t) - \lambda \sum_{i=1}^{k} p(i|x_t)\ln p(i|x_t). \tag{8}$$

In comparison with the conventional EM algorithm [1], this xed-point learning algorithm differs only at the augmenting term $\gamma_j(t)$. It can be easily veri ed that when $\lambda = 0$, $\gamma_j(t) = 1$, the xed-point learning algorithm is just the EM algorithm and when $\lambda = 1$, the xed-point learning algorithm returns to the original xed-point BYY learning algorithm [15] for maximizing the harmony function $J(\Theta_k)$.

Actually, $\gamma_j(t)$ implements a rival penalized competitive learning (RPCL) mechanism [19] so that model selection can be made adaptively during parameter learning. At the early learning stage, $\gamma_j(t) < 0$ may happen. According to Eq.(8), the mean vectors of the j-th Gaussian will move away from x_t. Otherwise, if $\gamma_j(t) > 0$, the mean vectors of the j-th Gaussian will be attracted to x_t. So, for x_t, Gaussians with $\gamma_j(t) > 0$ are winners while these Gausians with $\gamma_j(t) < 0$ are losers.

However, the xed-point learning algorithm cannot guarantee the positive de niteness of each covariance matrix during the iteration since $\gamma_j(t)$ may be negative. In order to overcome this problem, we use the EM update rule of the covariance matrixes, i.e., forcing all $\gamma_j(t) = 1$ in Eq.(7), in this degenerated case. In fact, this simpli cation is applicable and ef cient since the competition for adaptive model selection is mainly among mean vectors and controlled by the mixing proportions.

2.3. Dynamic Evolution of λ

We further discuss the dynamic evolution of λ with time T during the learning process. According to our regularization mechanism, λ should start around 1 and decrease slowly at the early learning stage to realize adaptive model selection. Then, at the sequent stage, λ can attenuate to 0 at a higher speed so that the algorithm will nally converge to a ML solution. So, it is crucial to check whether the adaptive model selection has accomplished and when to change learning stage.

In order to detect the turning point, we introduce the Shannon entropy of mixing proportions in the Gaussian mixture model, $H_\pi = -\sum_{j=1}^{k} \pi_j \ln \pi_j$. It is obvious that H_π is sensitive to the structure of the Gaussian mixture model. If model selection is not completed, the difference of H_π between two iterations is considerable. Otherwise, the difference should be very small. This motivates us to adopt the absolute change rate of H_π between two iterations, de ned by

$$h_\pi(T) = |\frac{H_\pi(T) - H_\pi(T-1)}{H_\pi(T)}|, \tag{9}$$

as an indicator of model selection. Here, T is the time, i.e., the number of iterations. The whole learning process is divided into two learning stages according to a given threshold $\varepsilon_1(> 0)$ of this indicator. That is, if $h_\pi(T) > \varepsilon_1$, $\lambda(T)$ increases at a low speed; otherwise, it increases at a high speed. Since $\lambda(T)$ is assumed to increase exponentially, its dynamic evolution process can be given as follow:

$$\lambda(T) = \begin{cases} 1 - \lambda_0 * \eta_1^T, & \text{if } h_\pi(T) > \varepsilon_1; \\ 1 - \lambda_0 * (\frac{\eta_1}{\eta_2})^{T^*} \eta_2^T, & \text{if } h_\pi(T) \le \varepsilon_1, \end{cases} \tag{10}$$

where λ_0 is a very small positive constant, η_1, η_2 are two positive constants with the constraint that $1 < \eta_1 < \eta_2$, and T^* is the turning point such that $h_\pi(T^*) > h_0$ and $h_\pi(T^* + 1) \le h_0$. When λ becomes 0, we x it until the algorithm stops.

2.4. Complete DRMLL Algorithm

We nally summary our proposed DRMLL algorithm. Firstly, we should choose the parameters of the algorithm properly. As mentioned previously, λ_0, η_1, η_2 and ε_1 must be carefully selected to make the evolution of $\lambda(T)$ dynamic. θ_0 is a threshold value to lter out Gaussians with very small mixing proportions during the parameter learning process, while $\varepsilon_2(> 0)$ is a threshold value to terminate the iteration. If $\lambda = 0$ and the absolute increment of the log likelihood is smaller than ε_2, we af rm the convergence of the algorithm. In our learning paradigm, k is !exible. However, it should be larger than the number k^* of actual Gaussians or clusters in the dataset. As for the initial setting of the parameters Θ_k, i.e., $\Theta_k^{(0)} = \{\pi_i^0, m_i^0, \Sigma_i^0\}|_{i=1}^k$, some competitive learning mechanism may be helpful. For example, m_i^0 can be selected through a DSRPCL procedure [19] and then π_i^0 and Σ_i^0 can be estimated accordingly.

After initializing all the parameters, Θ_k will be updated in each phase of $\lambda(T)$ via the xed-point learning algorithm given by Eqs (12)-(14). At the end of each learning phase, the Gaussians with the mixing proportions less than θ_0 are annihilated immediately. After $\lambda(T)$ becomes 0, the algorithm goes on until the log likelihood function reaches its maximum value or its absolute increment is less than ε_2.

3. EXPERIMENTAL RESULTS

In this section, various experiments are carried out to demonstrate the performance of the DRMLL algorithm for Gaussian mixtures. Moreover, it is compared with some typical existing learning algorithms. In these experiments, we always select $\varepsilon_1 = 1e-5$, $\varepsilon_2 = 1e-5$, $\eta_1 = 1.005$,$\lambda_0 = 1e-5$ $\eta_2 = 2$ and $\theta_0 = 0.05$. The other parameters will be speci ed in the particular experiments.

We begin to generate four typical synthetic datatsets from mixtures of four or three bivariate Gaussian distributions on the plane coordinate system (i.e., $d = 2$). Clearly, these

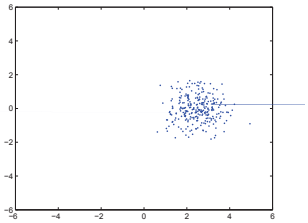
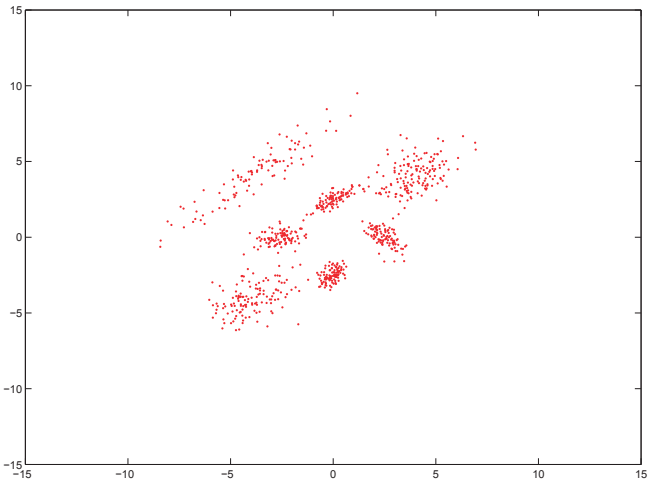

6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6

Table 1. The values of the parameters of the four synthetic datasets.

The dataset	Gaussian	m_i	σ_{11}^i	$\sigma_{12}^i(\sigma_{21}^i)$	σ_{22}^i	π_i	N_i
$\mathcal{S}_1$	G1	(2.50,0)	0.50	0.00	0.50	0.25	400
(N=1600)	G2	(0,2.50)	0.50	0.00	0.50	0.25	400
	G3	(-2.50,0)	0.50	0.00	0.50	0.25	400
	G4	(0,-2.50)	0.50	0.00	0.50	0.25	400
$\mathcal{S}_2$	G1	(2.50,0)	0.45	-0.25	0.55	0.34	544
(N=1600)	G2	(0,2.50)	0.65	0.20	0.25	0.28	448
	G3	(-2.50,0)	1.00	0.10	0.35	0.22	352
	G4	(0,-2.50)	0.30	0.15	0.80	0.16	265
$\mathcal{S}_3$	G1	(2.50,0)	0.10	-0.20	1.25	0.50	600
(N=1200)	G2	(0,2.50)	1.25	0.35	0.15	0.30	360
	G3	(-1,-1)	1.00	-0.80	0.75	0.20	240
$\mathcal{S}_4$	G1	(2.50,0)	0.28	-0.20	0.32	0.34	68
(N=200)	G2	(0,2.50)	0.34	0.20	0.22	0.28	56
	G3	(-2.50,0)	0.50	0.04	0.12	0.22	44
	G4	(0,-2.50)	0.10	0.05	0.50	0.16	32

Table 2. The comparison of the DRMLL and CEM^2 algorithms on model selection and runtime.

Datasets	DRMLL		CEM^2	
	CMS Frequency	runtime(s)	CMS Frequency	runtime(s)
$\mathcal{S}_1$	100%	707	84%	11290
$\mathcal{S}_2$	100%	764	56%	1825
$\mathcal{S}_3$	100%	405	72%	4317
$\mathcal{S}_4$	98%	250	56%	554



rithm is established for Gaussian mixtures. By controlling the scale factor of the regularization term to dynamically