THU-70250043-0, Pattern Recognition (Spring 2021) **Homework: 3**

EM and GMM

Lecturer: Changshui Zhang zcs@mail.tsinghua.edu.cn

Hong Zhao vzhao@tsinghua.edu.cn

Student:

EM and GD

In this problem you will see connections between the EM algorithm and gradient descent. Consider a GMM with known mixture weight π_k and spherical covariances (but the radius of spheres might be different). It's log likelihood is given by

$$l\left(\{\mu_k, \sigma_k^2\}_{k=1}^K\right) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} \pi_k \ N(x_i|\mu_k, \sigma_k^2 I)\right).$$

A maximization algorithm based on gradient descent should be something like:

- Initialize μ_k and σ_k^2, $k \in \{1, \cdots, K\}$. Set the iteration counter $t \leftarrow 1$.
- Repeat the following until convergence:
  - For $k = 1, \cdots, K$,
  $$\mu_k^{(t+1)} \leftarrow \mu_k^{(t)} + \eta_k^{(t)} \nabla_{\mu_k} l\left(\{\mu_k^{(t)}, (\sigma_k^2)^{(t)}\}_{k=1}^K\right)$$
  - For $k = 1, \cdots, K$,
  $$(\sigma_k^2)^{(t+1)} \leftarrow (\sigma_k^2)^{(t)} + s_k^{(t)} \nabla_{\sigma_k^2} l\left(\{\mu_k^{(t+1)}, (\sigma_k^2)^{(t)}\}_{k=1}^K\right)$$
  - Increase the iteration counter $t \leftarrow t + 1$

Please **prove** that with properly chosen step size $\eta_k^{(t)}$ and $s_k^{(t)}$, the above gradient descent algorithm is essentially equivalent to the following *modified* EM algorithm:

- Initialize μ_k and σ_k^2, $k \in \{1, \cdots, K\}$. Set the iteration counter $t \leftarrow 1$.
- Repeat the following until convergence:
  - E-step:
  $$\tilde{z}_{ik}^{(t+0.5)} \leftarrow Prob\left(x_i \in cluster_k | \{(\mu_j^{(t)}, (\sigma_j^2)^{(t)})\}_{j=1}^K, x_i\right),$$
  - M-step:
  $$\{\mu_k^{(t+1)}\}_{k=1}^K \leftarrow arg \max_{\{\mu_k\}_{k=1}^K} \sum_{i=1}^{n} \sum_{k=1}^{K} \tilde{z}_{ik}^{(t+0.5)} \left(\log \ N(x_i|\mu_k, (\sigma_k^2)^{(t)} I) + \log \pi_k\right)$$
  - E-step:
  $$\tilde{z}_{ik}^{(t+1)} \leftarrow Prob\left(x_i \in cluster_k | \{(\mu_j^{(t+1)}, (\sigma_j^2)^{(t)})\}_{j=1}^K, x_i\right),$$

– M-step:

$$\{(\sigma_k^2)^{(t+1)}\}_{k=1}^K \leftarrow arg \max_{\{\sigma_k\}_{k=1}^K} \sum_{i=1}^n \sum_{k=1}^K \tilde{z}_{ik}^{(t+1)} \left(\log\ N(x_i|\mu_k^{(t+1)}, \sigma_k^2 I) + \log \pi_k\right)$$

– Increase the iteration counter $t \leftarrow t+1$

The main modification is inserting an extra E-step between the M-step for μ_k's and the M-step for σ_k^2's.

Hint: Find the exact algebraic form of step size $\eta_k^{(t)}$ and $s_k^{(t)}$ from M-step.

EM for MAP Estimation

The EM algorithm that we talked about in class was for solving a maximum likelihood estimation problem in which we wished to maximize

$$\prod_{i=1}^m p(x^{(i)};\theta) = \prod_{i=1}^m \sum_{z^{(i)}} p(x^{(i)}, z^{(i)};\theta) \tag{1}$$

where $x^{(i)}$ were visible variables, $z^{(i)}$ were hidden variables and m was the number of samples. Suppose we are working in a Bayesian framework, and wanted to find the MAP estimate of the parameters θ by maximizing

$$(\prod_{i=1}^m p(x^{(i)};\theta))p(\theta) = (\prod_{i=1}^m \sum_{z^{(i)}} p(x^{(i)}, z^{(i)}|\theta))p(\theta) \tag{2}$$

Here, $p(\theta)$ is our prior on the parameters. Please **generalize the EM algorithm** to work for MAP estimation. You may assume that $log\ p(x,z|\theta)$ and $log\ p(\theta)$ are both concave in θ, so that the M-step is tractable if it requires only maximizing a linear combination of these quantities. (This roughly corresponds to assuming that MAP estimation is tractable when x, z is fully observed, just like in the frequentist case where we considered examples in which maximum likelihood estimation was easy if x, z was fully observed.)

Make sure your M-step is tractable, and also **prove** that $(\prod_{i=1}^m p(x^{(i)};\theta))p(\theta)$ (viewed as a function of θ) monotonically increases with each iteration of your algorithm.

Programming 1 (EM and GMM)

Consider the case that the hidden variable $y \in \{1, ..., m\}$ is discrete while the visible variable $x \in R^d$ is continuous. In other words, we consider mixture models of the form

$$p(x) = \sum_{j=1}^m p(x|y=j)p(y=j) \tag{3}$$

We assume throughout that x is conditionally Gaussian in the sense that $x \sim \mathcal{N}(\mu_j, \Sigma_j)$ when $y = j$. We have provided you with an example EM code for mixture of Gaussians (with visualization) in *Matlab*. The command to run is:

[param, history, ll] = em_mix(data,m,eps);

where the input points are given as rows of *data*, m is the number of components in the estimated mixture, and *eps* determines the stopping criteria of EM: the algorithm stops when the relative change in log-likelihood falls below *eps*. In the output, *param* is a cell array with m elements. Each element is a structure with the following fields:

mean - the resulting mean of the Gaussian component,

cov - the resulting covariance matrix of the component,

p - the resulting estimate of the mixing parameter.

The value of *param* is updated after every iteration of EM; the output argument *history* contains copies of these subsequent values of *param* and allows to analyze our experiments. Finally, *ll* is the vector where the t-th element is the value of the log-likelihood of the *data* after t iterations (i.e. the last element is the final log-likelihood of the fitted mixture of Gaussians).

- Run the EM algorithm based on *data* provided by `emdata.mat` with m = 2, 3, 4, 5 components. Select the appropriate model (number of components) and give reasons for your choice. Note that you may have to rerun the algorithm a few times (and select the model with the highest log-likelihood) for each choice of m as EM can sometimes get stuck in a local minimum. Is the model selection result sensible based on what you would expect visually? Why or why not?
- Modify the M-step of the EM code so that the covariance matrices of the Gaussian components are constrained to be equal. Give detailed derivation. Rerun the code and then select a appropriate model. Would we select a different number of components in this case?

Hint: For the above two questions you are encouraged to google "BIC(Bayesian Information Criterion)" to help you with the model selection process. Of course other criteria are welcomed as long as you give convincing reasons.

Hint: For this assignment, you are allowed to implement EM algorithm manually in python, and you can use scipy.io.loadmat to load the data.

Programming 2 (Missing Data)

	ω_1		
point	x_1	x_2	x_3
1	0.42	-0.087	0.58
2	-0.2	-3.3	-3.4
3	1.3	-0.32	1.7
4	0.39	0.71	0.23
5	-1.6	-5.3	-0.15
6	-0.029	0.89	-4.7
7	-0.23	1.9	2.2
8	0.27	-0.3	-0.87
9	-1.9	0.76	-2.1
10	0.87	-1.0	-2.6

Suppose we know that the ten data points in category ω_1 in the table above come from a three-dimensional Gaussian. Suppose, however, that we do not have access to the x_3 components for the even-numbered data points.

1. Write an EM program to estimate the mean and covariance of the distribution. Start your estimate with $\boldsymbol{\mu}^0 = 0$ and $\boldsymbol{\Sigma}^0 = \mathbf{I}$, the three-dimensional identity matrix.

2. Compare your final estimation with the case when we remove all even-numbered data points (2, 4, 6, 8, 10).

3. Compare your final estimation with the case when there are no missing data, namely we have access to all x_3.