THU-70250043, Pattern Recognition (Spring 2016) **Homework: 3**

Gaussian Mixture Model and Expectation Maximization Algorithm

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

Student: XXX xxx@mails.tsinghua.edu.cn

EM and Gradient Descent

In this problem you will investigate connections between the EM algorithm and gradient descent. Consider a GMM where $\Sigma_k = \sigma_k^2 I$, i.e., the covariances are spherical but of different spread. Moreover, suppose the mixture weight π_k is known. The log likelihood then is

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

A maximization algorithm based on gradient descent is as follows:

- Initialize μ_k and σ_k^2, $k \in \{1, \cdots, K\}$. Set the iteration counter t=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 $t \leftarrow t + 1$

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

- Initialize μ_k and σ_k^2, $k \in \{1, \cdots, K\}$. Set the iteration counter t=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.

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 the $z^{(i)}$'s were latent random variables. 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. 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.

EM Application

Consider the following problem. There are P papers submitted to a machine learning conference. Each of R reviewers reads each paper, and gives it a score indicating how good he/she thought that paper was. We let $x^{(pr)}$ denote the score that reviewer r gave to paper p. A high score means the reviewer liked the paper, and represents a recommendation from that reviewer that it be accepted for the conference. A low score means the reviewer did not like the paper.

We imagine that each paper has some "intrinsic" true value that we denote by μ_p, where a large value means it's a good paper. Each reviewer is trying to estimate, based on reading the paper, what μ_p is; the score reported $x^{(pr)}$ is then reviewer r's guess of μ_p.

However, some reviewers are just generally inclined to think all papers are good and tend to give all papers high scores; other reviewers may be particularly nasty and tend to give low scores to everything. (Similarly, different reviewers may have different amounts of variance in the way they review papers, making some reviewers more consistent/reliable than others.) We let ν_r denote the "bias" of reviewer r. A reviewer with bias ν_r is one whose scores generally tend to be ν_r higher than they should be.

Points	ω_1			ω_2		
	x_1	x_2	x_3	x_1	x_2	x_3
1	0.42	-0.087	0.58	-0.4	0.58	0.089
2	-0.2	-3.3	-3.4	-0.31	0.27	-0.04
3	1.3	-0.32	1.7	0.38	0.055	-0.035
4	0.39	0.71	0.23	-0.15	0.53	0.011
5	-1.6	-5.3	-0.15	-0.35	0.47	0.034
6	-0.029	0.89	-4.7	0.17	0.69	0.1
7	-0.23	1.9	2.2	-0.011	0.55	-0.18
8	0.27	-0.3	-0.87	-0.27	0.61	0.12
9	-1.9	0.76	-2.1	-0.065	0.49	0.0012
10	0.87	-1.0	-2.6	-0.12	0.054	-0.063

Table 1: Data for Programming 1

All sorts of different random factors influence the reviewing process, and hence we will use a model that incorporates several sources of noise. Specifically, we assume that reviewers' scores are generated by a random process given as follows:

$$\begin{aligned} y^{(pr)} &\sim \mathcal{N}(\mu_p, \sigma_p^2) \\ z^{(pr)} &\sim \mathcal{N}(\nu_r, \tau_r^2) \\ x^{(pr)} | y^{(pr)}, z^{(pr)} &\sim \mathcal{N}(y^{(pr)} + z^{(pr)}, \sigma^2) \end{aligned} \tag{3}$$

The variables $y^{(pr)}$ and $z^{(pr)}$ are independent; the variables (x, y, z) for different paper- reviewer pairs are also jointly independent. Also, we only ever observe the $x^{(pr)}$'s; thus, the $y^{(pr)}$'s and $z^{(pr)}$'s are all latent random variables.

We would like to estimate the parameters $\mu_p, \sigma_p^2, \nu_r; \tau_r^2$. If we obtain good estimates of the papers' "intrinsic" values μ_p, these can then be used to make acceptance/rejection decisions for the conference.

We will estimate the parameters by maximizing the marginal likelihood of the data $\{x^{(pr)}; p = 1, \ldots, P, r = 1, \ldots, R\}$. This problem has latent variables $y^{(pr)}$ and $z^{(pr)}$, and the maximum likelihood problem cannot be solved in closed form. So, we will use EM. Your task is to derive the EM update equations. Your final E and M step updates should consist only of addition/subtraction/multiplication/division/log/exp/sqrt of scalars; and addition/subtraction/multiplication/inverse/determinant of matrices. For simplicity, you need to treat only $\{\mu_p, \sigma_p^2; p = 1, \ldots, P\}$ and $\{\nu_r, \tau_r^2; r = 1, \ldots, R\}$ as parameters. I.e. treat σ^2 (the conditional variance of $x^{(pr)}$ given $y^{(pr)}$ and $z^{(pr)}$) as a fixed, known constant.

- In this part, we will derive the E-step:
  - The joint distribution $p(y^{(pr)}, z^{(pr)}, x^{(pr)})$ has the form of a multivariate Gaussian density. Find its associated mean vector and covariance matrix in terms of the parameters $\mu_p, \sigma_p^2, \nu_r, \tau_r^2$ and σ^2.
  - Derive an expression for $Q_{pr}(y^{(pr)}, z^{(pr)}) = p(y^{(pr)}, z^{(pr)} | x^{(pr)})$ (E-step), using the rules for conditioning on subsets of jointly Gaussian random variables.
- Derive the M-step updates to the parameters $\{\mu_p, \sigma_p^2, \nu_r, \tau_r^2\}$

Programming 1

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.

- Write an EM program to estimate the mean and covariance of the distribution. Start your estimate with $\mu_0 = 0$ and $\Sigma_0 = I$, the three-dimensional identity matrix.
- Compare your final estimate with that for the case when there is no missing data

Suppose we know that the ten data points in category ω_2 in the table above come from a three-dimensional uniform distribution $p(x|\omega_2) \sim U(x_l, x_u)$. Suppose, however, that we do not have access to the x_3 components for the even-numbered data points.

- Write an EM program to estimate the six scalars comprising x_l and x_u of the distribution. Start your estimate with $x_l = (-2, -2, -2)^t$ and $x_u = (+2, +2, +2)^t$.
- Compare your final estimate with that for the case when there is no missing data.

Programming 2

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

$$p(x) = \sum_{j=1}^{m} p(x|y = j)p(y = j) \tag{4}$$

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 the EM code for mixture of Gaussians (with visualization). 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).

To overcome any numerical problems the code involves (slight) regularization. Specifically, the M-step of the EM algorithm solves a regularized weighted log-likelihood problem, where the prior distribution over the mixing proportions is a Dirichlet and the prior over each covariance matrix is a Wishart. The equivalent sample size is set to one. The returned log-likelihood values include the regularization penalties (log-priors). See the code for details if you wish to change any of these settings.

- Run the EM algorithm based on data2 provided by hw5em2.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?

  Notes: for the above two questions you are encouraged to google "BIC" to help you with the model selection process. Of course other criteria are welcomed as long as you give convincing reasons.