Vinay Gaba
UNI: vhg2105

COMS 4771: Machine Learning Assignment 5

1. We know that the Soft EM algorithm consists of the following two steps:

$$\textbf{E-step}: \textit{For each training example } x^{(i)} \textit{ and for all } j \in [k], \textit{compute}$$

$$q_j^{(i)} = Pr_{\theta}[Y = j, X = x^{(i)}]$$

$\boldsymbol{M - step}$: $Update$ θ as follows:

$$\theta \leftarrow \arg max_{\Theta}, \sum_{i=1}^{n} \sum_{j=1}^{k} q_j^{(i)} . \ln Pr_{\theta}[Y = j, X = x^{(i)}]$$

In hard EM, the E-step above is replaced by a hard assignment of the hidden variable to the most likely value for the given data point under the current distribution (i.e. with parameters θ). We will call this the E'-step. The M-step stays the same. The hard EM algorithm repeats the following two steps until convergence, starting from an arbitrary initialization of the parameters θ:

$\boldsymbol{E' - step}$: $For each training example \ x^{(i)}, compute$:

$$j^{(i)} = \arg max_{j \in [k]} Pr_{\theta}[Y = j \mid X = x^{(i)}],$$

$Additionally,$

$$q_j^{(i)} = \begin{cases} 1 \ if \ j = j(i) \\ 0 \ otherwise \end{cases}$$

$\boldsymbol{M - step}$: $The same as the M - step in the soft EM algorithm.$

We are told that we have a Gaussian mixture model where all the k Gaussians have the same covariance matrix, ***I***, the identity matrix. Thus,

$\theta = (\pi_1, \mu_1, \pi_2, \mu_2, \ldots, \pi_k, \mu_k)$

where,
π_j = mixing weights
μ_j = centers of the Gaussians.

<u>Part 1:</u> For the mixture of Gaussians setting described above, work out the precise implementation of the E'- and M-steps.

From our lecture slides, we have:

$$Pr_{\theta}[Y = j, X = x^{(i)}] = E[\emptyset_j | X = x^{(i)}] = w_j^{(i)}$$

It is given that the Gaussians have covariance matrix ***I,*** the identity matrix. Replacing the covariance matrix with ***I,*** from the slides, we have:

$$w_j^{(i)} = \frac{\pi_j \,.\exp(-\frac{1}{2}\,(x - \mu_j)^T(x - \mu_j))}{\sum_{j'=1}^{k} \pi_{j'} \exp(-\frac{1}{2}\,(x - \mu_{j'})^T(x - \mu_{j'}))}$$

Since we need to solve this for making hard assignments, we will solve by finding a value of j which would have the maximum probability for a point belonging to label j, we will compute

To compute $q_j^{(i)}$, we will compute the MLE of $w_j^{(i)}$ with respect to *j*.

$q_j^{(i)} = \arg max_{j \in [k]}\ w_j^{(i)}$

As we saw in the first half of the class, we can maximize the the log likelihood of $w_j^{(i)}$, which would evaluate to

$$\ln \pi_j \,.\exp(-\frac{1}{2}\,||(x - \mu_j)||^2)$$

$$= \ln \pi_j - \tfrac{1}{2}\,\left|\left|(x - \mu_j)\right|\right|^2$$

$$= -2.\ln \pi_j + \left|\left|(x - \mu_j)\right|\right|^2$$

M Step - $The\ same\ as\ the\ M -$
$step\ in\ the\ soft\ EM\ algorithm.\ However, we\ wont\ be\ required\ to\ compute\ the\ value\ of\ \Sigma.$

2. ***The hard EM algorithm you derived above should be almost (but not exactly) the same as another algorithm we have already studied in class. Which algorithm? What is the difference between the two algorithms***

   As we see from the equation we derived above, the equation for Hard EM Algorithm is very closely related to **K-Means Clustering Algorithm** that was discussed in class. In fact, it is almost identical to the K- Means Clustering Algorithm equation and just has an additional factor added to the equation.

   The difference between K-Means Clustering and Hard EM is that K-Means Clustering first assigns a class center μ_k for each class k. It then iteratively classifies each point x_t as belonging to a class k whose center is closest to x_t.

   Each iteration reduces the sum of squares of the difference between the point and its class center. The procedure terminates when the class labels stop changing their values (stabilization).

   Other observations about the difference is that K-Means clustering depends on the L2 norm when optimizing whereas Hard EM Algorithm is based on Expectation.

Collaborators

Dikha Vanvari dhv2108
Varun Shetty vs2567