Discussion

Hide Discussion

Topic: Unit 4 Unsupervised Learning (2 weeks) :Lecture 16. Mixture Models; EM algorithm / 5. Mixture Model - Unobserved Case: EM Algorithm

Add a Post

< All Posts

EM Algorithm - Gaussian Mixture Model (memo)

discussion posted 7 days ago by michael_x

Pinned

About indices

n : the number of data points ($i = 1, \ldots, n$)

k : the number of clusters ($j = 1, \ldots, k$)

d : dimension of each data point ; $dim\left(x^{(i)}\right)$

In this problem, $n = 5,\ k = 2,\ d = 1$

E-step

$$p\left(j \mid i\right) = \frac{p_j\,\mathcal{N}\left(x^{(i)}; \mu_j, \sigma_j^2\right)}{p\left(x^{(i)}|\theta\right)}$$

$$p\left(x^{(i)}|\theta\right) = \sum_{j=1}^{k} p_j\,\mathcal{N}\left(x^{(i)}; \mu_j, \sigma_j^2\right)$$

Note : $p\left(j \mid i\right)$ is the probability that the ith data point belongs to the jth cluster

just in case I won't be able to recognize the following notation in the future :

$$\mathcal{N}\left(x^{(i)}; \mu_j, \sigma_j^2\right) = p_{X^{(i)};\mu_j,\sigma_j^2}\left(x^{(i)}; \mu_j, \sigma_j^2\right)$$

where $X^{(i)} \sim \mathcal{N}\left(\mu_j, \sigma_j^2\right)$

Also, I needed to use a little bit sloppy notations, in order to see that $p\left(j \mid i\right)$ is actually given by Bayes' rule.

Here's Bayes' formula :

$$P\left(A \mid B\right) = \frac{P\left(B \mid A\right)\,P\left(A\right)}{P\left(B\right)}$$

Compare

$$p\left(j \mid i\right) = \frac{p_j\,\mathcal{N}\left(x^{(i)}; \mu_j, \sigma_j^2\right)}{p\left(x^{(i)}|\theta\right)}$$

$$p\left(x^{(i)}|\theta\right) = \sum_{j=1}^{k} p_j\,\mathcal{N}\left(x^{(i)}; \mu_j, \sigma_j^2\right)$$

with

$$p_{K|N}\left(j \mid i\right) = \frac{p_{N|K}\left(i \mid j\right) p_K\left(j\right)}{p_N\left(i\right)}$$

$$p_N\left(i\right) = \sum_{j=1}^{k} p_{N|K}\left(i \mid j\right) p_K\left(j\right)$$

where N and K are random variables, each of which takes i and j as realized values, respectively

M-step

$$\hat{n}_j = \sum_{i=1}^{n} p(j \mid i)$$

$$\hat{p}_j = \frac{\hat{n}_j}{n}$$

$$\hat{\mu}_j = \frac{1}{\hat{n}_j} \sum_{i=1}^{n} p(j \mid i)\, x^{(i)}$$

$$\hat{\sigma}_j^2 = \frac{1}{\hat{n}_j d} \sum_{i=1}^{n} p(j \mid i) \parallel x^{(i)} - \hat{\mu}_j \|^2$$

I wrote a R code according to the formulas above, and I got the right answers. The only function I defined in the code is bayes() for the 1st formula in the E-Step, namely :

$$p(j \mid i) = \frac{p_j \,\mathcal{N}(x^{(i)}; \mu_j, \sigma_j^2)}{p(x^{(i)}|\theta)}$$

```
# mixture components
mean <- c(-3, 2)
variance <- c(4, 4)

# mixture weights
p <- c(0.5, 0.5)

# observed data
x <- c(0.2, -0.9, -1, 1.2, 1.8)

# define p(j|i)
bayes <- function(p, x, mean, variance, i, j) {
...
}

# calculate p(j|i) for all n=5 data points
for (i in c(1:length(x))) {
 ... use bayes() here ...
}
```

and so on.

I didn't need to vectorize the code because the problem asks about the $j = 1$ case alone. It would be more time consuming to write a vectorized code that will calculate everything we need just by one click, even when we have multidimensional data points, in which case each $x^{(i)}$ is a vector and d is more than 1.

I added the following memo as requested

Likelihood and Log-Likelihood

Setup :

We're going to consider a string of words generation problem.

Suppose we have k clusters. Let us denote corresponding mixture weights by π_j (where $j = 1, \ldots, k$)

(In other words, we have k clusters, jth of which is chosen with probability π_j)

Suppose we have a string D that is comprised of n words, whose ith word is $x^{(i)}$.

Once one particular cluster j has been chosen, the probability that the word $x^{(i)}$ is generated is described as follows :

$$\mathcal{N}\left(x^{(i)}; \mu_j, \sigma_j^2\right) = p_{X^{(i)}; \mu_j, \sigma_j^2}\left(x^{(i)}; \mu_j, \sigma_j^2\right)$$

where $X^{(i)} \sim \mathcal{N}\left(\mu_j, \sigma_j^2\right)$

For example, if $x^{(i)}$ is the word "cat", the probability that the word "cat" is generated can be different for different clusters because

$\mathcal{N}($ "cat" $; \mu_j, \sigma_j^2)$ depends on mixture components μ_j and σ_j^2

Recall we can define θ such that it expresses all the mixture weights and mixture components :

$$\theta = \{\pi_1, \ldots, \pi_k, \mu_1, \ldots \mu_j, \sigma_1^2, \ldots, \sigma_j^2\}$$

Likelihood

The likelihood that we get the string D is :

$$L_n\left(D \mid \theta\right) = \prod_{i=1}^{n} \left(\sum_{j=1}^{k} \pi_j \, \mathcal{N}\left(x^{(i)}; \mu_j, \sigma_j^2\right) \right)$$

Log-Likelihood

The log-likelihood that we get the string D is :

$$\ln L_n\left(D \mid \theta\right) = \sum_{i=1}^{n} \left(\ln \left(\sum_{j=1}^{k} \pi_j \, \mathcal{N}\left(x^{(i)}; \mu_j, \sigma_j^2\right) \right) \right)$$

In some literature, the notations can be different :

Likelihood

$L_n\left(D \mid \theta\right) \;:=\; p\left(D \mid \theta\right)$ or $p\left(S_n \mid \theta\right)$ etc.

Log-Likelihood

$\ln L_n\left(D \mid \theta\right) \;:=\; \ell\left(D \mid \theta\right)$ etc.

For all intents and purposes, we didn't define our vocabulary size as N (for lack of a better alphabet).

It's confusing if we have both n and N at the same time. If the vocabulary size is 0, we can't produce any strings, but otherwise the vocabulary size N is NOT relevant to the string length n, while it could make us wonder that

a) "n is a realized value of random variable N ?" or

b) "the vocabulary size N is actually same as the cluster size k ?" etc.

(I believe neither of them is true)