7. Implementing EM for matrix completion

Project due Apr 26, 2023 08:59 -03 Completed

We need to update our EM algorithm a bit to deal with the fact that the observations ar
vectors. We use Bayes' rule to find an updated expression for the posterior probability
$p(j|u) = P(y = j|x_{C_u}^{(u)})$:

$$p(j \mid u) = \frac{p(u|j) \cdot p(j)}{p(u)} = \frac{p(u|j) \cdot p(j)}{\sum_{j=1}^{K} p(u|j) \cdot p(j)} = \frac{\pi_j N(x_{C_u}^{(u)}; \mu_{C_u}^{(j)}, \sigma_j^2 I_{C_u \times C_u})}{\sum_{j=1}^{K} \pi_j N(x_{C_u}^{(u)}; \mu_{C_u}^{(j)}, \sigma_j^2 I_{C_u \times C_u})}$$

This is the soft assignment of cluster j to data point u.

To minimize numerical instability, you will be re-implementing the E-step in the log-dom
calculate the values for the log of the posterior probability, $\ell(j, u) = \log(p(j|u))$ (th
of your E-step should include the non-log posterior).

Let $f(u, i) = \log(\pi_i) + \log\left(N(x_{C_u}^{(u)}; \mu_{C_u}^{(i)}, \sigma_i^2 I_{C_u \times C_u})\right)$. Then, in terms of f, the lo

$$\begin{aligned}
\ell(j|u) \quad &= \log(p(j \mid u)) = \log\left(\frac{\pi_j N(x_{C_u}^{(u)}; \mu_{C_u}^{(j)}, \sigma_j^2 I_{C_u \times C_u})}{\sum_{j=1}^{K} \pi_j N(x_{C_u}^{(u)}; \mu_{C_u}^{(j)}, \sigma_j^2 I_{C_u \times C_u})}\right) \\
&= \log\left(\pi_j N(x_{C_u}^{(u)}; \mu_{C_u}^{(j)}, \sigma_j^2 I_{C_u \times C_u})\right) - \log\left(\sum_{j=1}^{K} \pi_j N(x_{C_u}^{(u)}; \mu_{C_u}^{(j)}, \sigma_j^2 I_{C_u \times C_u}\right. \\
&= \log(\pi_j) + \log\left(N(x_{C_u}^{(u)}; \mu_{C_u}^{(j)}, \sigma_j^2 I_{C_u \times C_u})\right) - \log\left(\sum_{j=1}^{K} \exp(\log(\pi_j N(x\right. \\
&= f(u, j) - \log\left(\sum_{j=1}^{K} \exp(f(u, j))\right)
\end{aligned}$$

Once we have evaluated $p(j|u)$ in the E-step, we can proceed to the M-step. We wish
π, μ, and σ that maximize $\ell(X; \theta)$,

First, note that, by decomposing the multivariate spherical Gaussians into univariate sp
before, we can write, if $l \in C_u$:

$$\frac{\partial}{\ldots} N(x^{(u)} | \mu^{(k)}, \sigma^2 I_{|C_u| |C_u|}) \quad N(\ldots) \quad \frac{\partial}{\partial \mu_l^{(k)}} \left(\frac{1}{\sqrt{2\pi}\sigma_{l,(k)}} \exp\left(-\frac{1}{2\sigma_{l,(k)}^2}(x_l^{(u)} -\right.\right.$$

Following the EM algorithm's approach of maximizing a proxy likelihood function consider the following function:

where is the likelihood of generated by clust set is . The values are the ones as we computed in the E step and they are c

We now take the derivative of with respect to to find the optimal value o .

where if and if .

Setting the partial derivative equal to zero, we obtain that

We leave it as an exercise to the reader to obtain the estimates of and for

Implementation guidelines:

- To debug your EM implementation, you may use the data files `test_incomplete.tx` `test_complete.txt`. Compare your results to ours from `test_solutions.txt`.

Implementing E-step (2)

1.0/1.0 point (graded)

In `em.py`, fill in the `estep` function so that it works with partially observed vectors w indicated with zeros, and perform the computations in the log domain to help with num

Available Functions: You have access to the NumPy python library as `np`, to the `Gau` and to typing annotation `typing.Tuple` as `Tuple`. You also have access to `scipy.s` `logsumexp`

Hint: For this function, you will want to use `log(mixture.p[j] + 1e-16)` instead of `l` avoid numerical underflow

```
def log_gaussian(x: np.ndarray, mean: np.ndarray, var: float) -> float:
    """Computes the log probablity of vector x under a normal distribution
    Args:
        x: (d, ) array holding the vector's coordinates
        mean: (d, ) mean of the gaussian
        var: variance of the gaussian
    Returns:
        float: the log probability
    """
    d = len(x)
    log_prob = -d / 2.0 * np.log(2 * np.pi * var)
    log_prob -= 0.5 * ((x - mean) ** 2).sum() / var
    return log_prob


```

Press ESC then TAB or click outside of the code editor to exit

Correct

Test results

CORRECT

Submit You have used 8 of 50 attempts

```

    Returns:
        GaussianMixture: the new gaussian mixture
    """
```

Press ESC then TAB or click outside of the code editor to exit

Correct

Test results

CORRECT

Submit You have used 6 of 50 attempts

Implementing run

1.0/1.0 point (graded)

In `em.py`, fill in the `run` function so that it runs the EM algorithm. As before, the conv you should use is that the improvement in the log-likelihood is less than or equal to absolute value of the new log-likelihood. Note: do not alter data 'X' in place. Deep copy values.

Available Functions: You have access to the NumPy python library as `np`, to the `Gau` and to typing annotation `typing.Tuple` as `Tuple`. You also have access to the `este` functions you have just implemented

```
def run(X: np.ndarray, mixture: GaussianMixture,
        post: np.ndarray) -> Tuple[GaussianMixture, np.ndarray, float]:
    """Runs the mixture model

    Args:
        X: (n, d) array holding the data
        post: (n, K) array holding the soft counts
            for all components for all examples

    Returns:
        GaussianMixture: the new gaussian mixture
        np.ndarray: (n, K) array holding the soft counts
            for all components for all examples
        float: log-likelihood of the current assignment
    """
```

Press ESC then TAB or click outside of the code editor to exit

Show all posts

Why is log(mixture.p[j] + 1e-16) useful to avoid numerical underflow ?
This is about the hint on "Implementing E-step (2)" > Hint: For this function, you will want to use log(mixture.

M-step - the condition for updating the mean (Implemetation guidelines #2)
If I understand the condition properly - it says: update the mean if n_hat > 1 (by n_hat I mean the denominato

Efficient code in e-step with different sizes of input vectors?
Hi, to describe the issue I will use the example from my previous post: We have an array of input vectors 5×1

Don't we lose too much after the observations are no longer complete vectors?
Hi, I have a problem with catching the idea of shrinking the input vectors to only non-zero values. It reduces

Derivation of d/du(N,x,sigma^2)

Partial grade on E-step?
Does anyone got a partial grade for E-step? I got all the answer correct except the first log likelihood. I have

How to write code for normal pdf with only observed values?

Grader Issue
Hi Staff, I'm wondering if the grader is just malfunctioning for my answer for Implementing M-step (2) questic

[STAFF]: Why doesn't the Grader's "See full output" for the M-step show the input mean for e
It would help if the Grader's "See full output" showed the input mean. I could then copy the input data into th

Staff: Grader error?
In m-step for the case of: X: [[0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0. 0. 0.] [0.13471579 0. 0.] [0.

M-step: Shallow or deepcopy the mean (mixture.mu)? What about the variance?
Shall we also deepcopy the mean in the M-step or just do shallow .copy() , i.e. mu = mixture.mu.copy() ? Do v

dimension and number of samples

edX

About
Affiliates
edX for Business
Open edX
Careers
News

Legal

Connect

Blog
Contact Us
Help Center
Security
Media Kit

GET IT ON Google play

Download on the App Store