

Advanced Machine Learning 2020 - Exercise 2

November 19, 2019

1. In this exercise you will implement the Loopy Belief Propagation (LBP), for a 2D image denoising task. In this task, we are given an image with a patch which contains noisy pixels, that need to be recovered. To do so, we define a Markov Network over a 2D grid graph (corresponding to the image pixels) as follows. The variables y_1, \dots, y_n correspond to the observed pixel values (i.e., the noisy image). The variables x_1, \dots, x_n correspond to the original (unobserved) pixel values (-1.0 to indicate black pixels and 1.0 to indicate white pixels).

We consider the following Markov network over the x, y variables. Define the following pairwise interaction between x_i and y_i , which reflects the fact that x_i is likely to be close to y_i (this is sometimes known as the data term):

$$\phi(x_i, y_i) = e^{\alpha x_i y_i}$$

Also, consider the pairwise interaction between x_i and x_j where i, j are neighboring pixels in the 2D grid:

$$\phi(x_i, x_j) = e^{\beta x_i x_j}$$

This reflects the fact that nearby pixels in the original image are likely to be similar, and is known as the smoothness term. The overall Markov network is then:

$$p(x_1, \dots, x_n, y_1, \dots, y_n) = \prod_{i=1 \dots n} \phi(x_i, y_i) \prod_{ij \in E} \phi(x_i, x_j) \quad (1)$$

Your goal is to use BP to approximate the MAP problem:

$$\max_{x_1, \dots, x_n} p(x_1, \dots, x_n | y_1, \dots, y_n) \quad (2)$$

This is the problem of interest since the y_i are observed and we would like to infer the x_i .

- (a) (no need to submit) Show $p(x_1, \dots, x_n | y_1, \dots, y_n)$ is a Markov network over variables x_1, \dots, x_n , with factors $\psi_{i,j}(x_i, x_j) = \phi_i(x_i, x_j)$ and $\psi_i(x_i) = \phi_i(x_i, y_i)$. The updates for LBP are (convince yourself

that these work for the tree case, since they contain the singleton term that we didn't use in class):

$$m_{ij}(x_j) \leftarrow \max_{x_i} \psi_i(x_i) \psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{ki}(x_i) \quad (3)$$

And then at convergence the approximate MAP is given by:

$$x_i^* = \arg \max_{x_i} \psi_i(x_i) \prod_{k \in N(i)} m_{ki}(x_i) \quad (4)$$

- (b) (no need to submit) Since you are updating message in a loopy fashion, they may grow to infinity or decay to zero. One way to avoid this is to normalize each message as follows:

$$m_{ij}(x_j) = \frac{m'_{ij}(x_j)}{\sum_{\bar{x}_j} m'_{ij}(\bar{x}_j)} \quad (5)$$

Convince yourself that this does not change the output of the algorithm (namely the approximated MAP assignment)

- (c) Implement the max-product algorithm on the above network. There are many options for choosing the sequence of messages. Here's a simple one you can try: choose a certain order on the variables (say row by row in the part of the image you want to recover), and then for each node update all messages from it to its neighbors). You can also choose a convergence criterion, such as message not changing by much.

Submission Guidelines:

- The image to be completed in this exercise is downloadable from the course Moodle under the name "digit.png"
- The output of your code should be a reasonable cleaned version of the noisy image.
- submit in the PDF submission file the path to your code's folder
- Make sure your code's folder has the proper read permissions.
- Your code should run by the command

```
>>> python mrf_denoising.py input_fname output_fname
```

where `input_fname` is the name of the input image and `output_fname` is the output image. For example, invoking

```
>>> python mrf_denoising.py digit.png clean_digit.png
```

will clean the image in the file `digit.png` and save the file `clean_digit.png`.

- For numerical stability, consider working in log-space.

- Change the values of α and β to tune the results.
 - You may use `mrf_denoising.py` available on the course Moodle.
 - Your code should be readable and well-documented.
 - add a README.txt file with the IDs, Names and Emails and if needed details about the code.
2. The question refers to sum-product LBP. Namely, the algorithm which on trees calculates exact marginals. Consider the pairwise MRF:

$$p(x_1, \dots, x_n) \propto \prod_{ij \in E} \phi_{ij}(x_i, x_j) \quad (6)$$

where E are edges of a given graph. Define the following “pseudo-marginals” $\mu_{ij}(x_i, x_j)$ and $\mu_i(x_i)$:

$$\mu_i(x_i) \propto \prod_{k \in N(i)} m_{ki}(x_i) \quad (7)$$

$$\mu_{ij}(x_i, x_j) \propto \phi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{ki}(x_i) \prod_{k \in N(j) \setminus i} m_{kj}(x_j) \quad (8)$$

- (a) Show that for tree graphs the above pairwise pseudo-marginals $\mu_{ij}(x_i, x_j)$ are indeed the correct marginals of the MRF (namely $p(x_i, x_j) = \mu_{ij}(x_i, x_j)$). You may use the fact that the messages are in that case the correct sum of assignments in the corresponding sub-tree. Note that it is also true that $p(x_i) = \mu_i(x_i)$ (we showed this when deriving BP for trees).
- (b) Note that generally the pairwise and singleton marginals might not be consistent. Namely it isn't true that $\sum_{x_j} \mu_{ij}(x_i, x_j) = \mu_i(x_i)$. Show that at a fixed point of LBP this property does hold.
- (c) Show that for any graph E (possibly non-tree) it holds that:

$$p(x_1, \dots, x_n) \propto \prod_i \mu_i(x_i) \prod_{ij \in E} \frac{\mu_{ij}(x_i, x_j)}{\mu_i(x_i) \mu_j(x_j)} \quad (9)$$

This is known as a reparameterization of the distribution p . Note that generally μ_{ij} and μ_i are not the marginals of p . But when p is a tree then they are (because of (a) above).

3. We say that a graph is “Chordal” if every cycle of length greater than 3 in the graph has a chord, namely an edge not in the cycle that connects two nodes in the cycle (this is also known as a shortcut in the cycle). For example, if a graph G is chordal, and it contains the edges $(1, 2), (2, 3), (3, 4), (4, 1)$ then it must also contain one of the edges $(2, 4)$ or $(1, 3)$.

It turns out that induced graphs and chordal graphs are closely related, as we show next.

- (a) Given a graph G and an elimination order π prove that the induced graph G_π is chordal.
- (b) Given a chordal graphs G , there exists an elimination order π such that $G_\pi = G$.

[11pt]book

Advanced Methods in Machine Learning**Homework Submission Guidelines:****General Instructions**

1. **Indicate at the top your names, ID numbers and e-mails.**
2. Assignments are to be submitted in a hard-copy fashion to the cell next to room number 210 in the wooden boxed on the 2nd floor in Schreiber building.
3. Assignments should be done in groups of up to 3 students.
4. Submit the theory questions and programming assignments (when relevant) clipped together (only one submission).
5. Print (using Word or LaTeX) all discussions and plots in the programming assignments. Discussions should be written right after their corresponding plots.

Theoretical Assignments

1. Follow the exercise specifications exactly. Seek clarifications if needed.
2. Your proofs should be both rigorous and clear.
3. You may use anything proved in class.
4. Your exercise should be submitted printed or in a clear and readable handwriting. Unreadable exercises will not be graded.

Practical Assignments

1. Zip your code, report and any output files and send to tau.aml.2020@gmail.com. The title of the email should be the exercise number and group IDs.
2. Include a README file with details on how to run your code (note that No additional parameter should be added aside for those who were requested).
3. Make sure to indicate in the email the names, IDs of all the group members in the email and in the README file.
4. Follow the exercise specifications exactly.
5. There is no need to print the code.
6. your code should be uploaded to NOVA. Make sure your code is running there and the folder has the proper read permissions. Add the path on the README file.
7. Some assignments will have skeleton code provided on Moodle.

8. Your code should be readable and well-documented.
9. For each practical assignment there might be specific instructions indicating how to execute the code. make sure to follow them exactly.