EQ2415 - Machine Learning and Data Science
Period 1, 2022/23

Homework Project – Graphical Models and VI R. Thobaben

Overview

The goal of this project is to explore and implement techniques from the Lectures RT-1, RT-2, and RT-3 dealing with *inference in graphical models and variational inference.* The project must contain three non-trivial elements that cover concepts from all the three lectures. Students are welcome to follow the following project assignment. However, students also have the freedom to choose another use case (e.g., tracking and prediction of time series data) and to explore other techniques than the ones suggested below.

The project will be carried out in groups of two students and is evaluated through a short oral presentation. Grades are given individually based on the performance during the presentation. The times for presentations are not pre-scheduled, and the groups are asked to contact the TAs for scheduling a presentation meeting.

1 Image De-noising Using Markov Random Fields

The goal of this first part of Project 1 is to generalize and implement the image de-noising algorithm presented in Section 8.3.3 in [1] to non-binary grayscale images. It is recommended that students use grayscale standard images (e.g., Lena) and adapt to the resolution to a reasonable size (somewhere between 3 and 8 bit depending on available processing power). Noisy versions of the images can then be obtained by adding artificial noise where the students can control the noise parameter (e.g., noise energy). For this setup, derive and implement an image de-noising algorithm that is based on Markov random fields. Evaluate the performance of your algorithm by measuring the mean-squared error (MSE) after de-noising, averaged over sufficiently many noise realizations, and plotting the MSE over the noise parameter. Note that in this part, it is sufficient to adjust / optimize the parameters of the potential functions offline (i.e., they do not need to be estimated as part of the image de-noising algorithm).

In this part of the project, you may find the following guiding questions and comments useful:

- The resolution of images can be adjusted by quantization. An easy way to implement a quantizer is by scaling and rounding.
- Different ways of modeling artificial noise are possible: additive white Gaussian noise with a given noise variance or a noise sequence defined over a finite field parameterized by its Hamming weight (e.g., if the image has a k-bit resolution, it makes sense to define the noise over the $\mathrm{GF}(2^k)$).
- You may want to implement the 1-bit case described in the book first and extend it then.
- Can you relate the parameters of the energy function to properties of the artificial noise and the image (e.g., noise variance, expected values, correlation coefficients)?
- You are not the first to look at this problem, and there are research articles out there that may be helpful.

2 Image De-noising Using the SP/MS Algorithm

The goal of the second part of Project 1 is to develop an image de-noising algorithm based on the sum-product or the max-sum algorithm for the framework introduced in Part 1. In order to derive an appropriate SP algorithm, start with a factorization of joint probability of the noisy image and the underlying image. For example, assuming an $N \times N$ image with pixels $x_{i,j}$ and noisy pixel observations $y_{i,j}$ you can write

$$\begin{aligned} p(y_{1,1}, \ldots, y_{N,N}, x_{1,1}, \ldots, x_{N,N}) &= \prod_i^N \prod_j^N p(y_{i,j}|x_{i,j}) p(x_{11}, \ldots, x_{N,N}) \\ &= \prod_i^N \prod_j^N p(y_{i,j}|x_{i,j}) p(x_{N,N}|x_{1,1}, \ldots, x_{N,N-1}) \cdot \\ &\qquad p(x_{N,N-1}|x_{1,1}, \ldots, x_{N,N-2}) \cdot \\ &\qquad\quad p(x_{N,N-2}|x_{1,1}, \ldots, x_{N,N-3}) \cdot \ldots, \end{aligned}$$

where $p(y_{i,j}|x_{i,j})$ is determined by the artificial noise and the terms $p(x_{i,j}|x_{1,1}, \ldots, x_{i,j-1})$ are determined by the correlation among pixels. Note that the factorization of $p(x_{11}, \ldots, x_{N,N})$ above can be interpreted as converting the 2D image into a chain of pixels. In order to obtain a tractable model, make simplifying assumptions on the correlation model (e.g., only horizontal correlation between neighboring pixel, only vertical correlation between neighboring pixel, horizontal and vertical correlation only between neighboring pixels). Start your implementation with the simplest correlation model (e.g., correlation only in one direction) and extend it gradually. Evaluate the performance and compare the results with the results from Part 1. As in Part 1, you can determine parameters offline (e.g., estimate probabilities $p(x_{i,j}|x_{i,j-1})$ using a histogram); they do not need to be estimated as part of the SP algorithm.

In this part of the project, you may find the following guiding questions and comments useful:

- In order to be able to implement and run the SP algorithm, factors like $p(x_{i,j})$, $p(x_{i,j}|x_{i,j-1})$, and $p(x_{i,j}|x_{i,j-1}, x_{i-1,j})$ (depending on the chosen factorization and correlation model) need to be determined. They can be either estimated from histograms or appropriately modeled with parametric models (e.g., jointly Gaussian, Gaussian mixtures).
- Which message schedule will work efficiently?
- How many different types of function nodes are there?

3 Variational Methods

The goal of the third part of Project 1 is to apply one variational method from Lecture 4 (Chapter 9.4 and 10 in [1]) to the problem studied in Part 1 and 2 of this project. As a suggestion, the distribution of the pixels $p(x_{i,j})$ can be modeled as a Gaussian mixture. Use a variational approach to estimate the parameters of the distribution. If successful, you can think about the following extensions:

- Can you extend your solution such that parameters of conditional probabilities $p(x_{i,j}|x_{i,j-1})$ or joint probabilities $p(x_{i,j}, x_{i,j-1})$ can be estimated as well?
- Can your solution be combined with your solution from Part 2 in order to jointly perform image de-noising and parameter estimation?

References

[1] C.M. Bishop, "Pattern Recognition and Machine Learning," Springer 2006, Chapter 8.4.