Exercise Sheet 02

Deadline: 02.05.2018, 1pm
Regulations:

- You need $\geq$ **50% of all points** in the weekly exercises and successfully work on a $\geq$ **2 weeks fulltime project** and submit a **written report** in order to pass this class.
- Hand in your solutions in **groups of two or three people**.
- Make sure your code works with python 3.6 or later.
- For exercises that include coding, hand in a **Juypter Notebook** for each exercise containing all your answers, results, plots and code. Separate the different sub-exercises clearly by using headings and describing text. Label each file with the number of the exercise. Before submission, make sure your Jupyter Notebook compiles without any errors when you do Run All. Additionally hand in a PDF version of the Notebook.
- For exercises that can be solved without coding, hand in a **PDF file** for each exercise containing all your answers and results. Separate the different sub-exercises clearly by using headings and describing text. Label each file with the number of the exercise.
- Submit all your results in a **single** .zip archive. The title of the submitted ZIP-file as well as the subject line of your email **must** start with **EX02** followed by the full names of **all group members**. **People not mentioned in the subject will not get credits.**
- Submit all your results to mlcvss18@gmail.com.
- Submissions that do not adhere to the deadline and regulations, or that are chaotic, will not be graded.

Exercise 1. 5 P.

This exercise can be solved entirely without writing any code.

Let $E(x_0, x_1, x_2)$ be an energy-function of 3 binary variables $x_0 \in \{0, 1\}$, $x_1 \in \{0, 1\}$ and $x_2 \in \{0, 1\}$. $E(x_0, x_1, x_2)$ is a sum of potentials where each potential depends only on a subset of variables:

$$E(x_0, x_1, x_2) = \psi_0(x_0) + \psi_1(x_1) + \psi_2(x_1) + \psi_p(x_0, x_1) + \psi_p(x_0, x_2) + \psi_p(x_1, x_2)$$

The unary potentials are defined as:

$$\psi_0(x_0) = \begin{cases} 0.1 & \text{if } x_0 = 0 \\ 0.9 & \text{if } x_0 = 1 \end{cases} \qquad \psi_1(x_1) = \begin{cases} 0.8 & \text{if } x_1 = 0 \\ 0.1 & \text{if } x_1 = 1 \end{cases} \qquad \psi_2(x_2) = \begin{cases} 0.9 & \text{if } x_2 = 0 \\ 0.1 & \text{if } x_2 = 1 \end{cases}$$

And the pairwise potential is a Potts function:

$$\psi_p(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j \\ 1 & \text{if } x_i \neq x_j \end{cases}$$

a) Draw the corresponding factor graph for this model.

b) Evaluate $E(x_0, x_1, x_2)$ by hand for all possible configurations of x_0, x_1 and x_2.

c) Which configuration of x_0, x_1 and x_2 minimizes $E(x_0, x_1, x_2)$?

d) The probability for the state $x = (x_0, x_1, x_2)$ is given by

$$p(x) \quad = \tfrac{1}{\mathcal{P}} \exp(-E(x))$$

with the partition function

$$\mathcal{P} \quad = \sum_{\text{all possible } x} \exp(-E(x))$$

Use the results from b) to compute the likelihood $p(x_0, x_1, x_2)$ for all possible configurations of x_0, x_1 and x_2.

Exercise 2. 3 P.

This exercise can be solved entirely without writing any code.

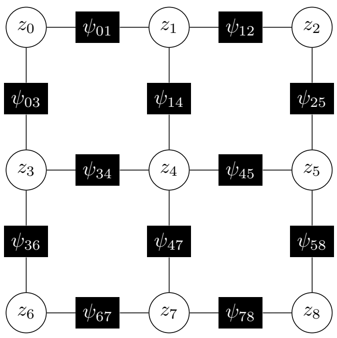

Figure 1: A graphical representation of a $N \times N$ graphical model with only binary terms as a factor graph. Each black box corresponds to a factor, each circle corresponds to a variable.

Consider a graphical model without unary terms (i.e. without a likelihood term) on a grid graph as in figure 1 with the size $N \times N$ where each variable z_i can take only two states: $z_i \in \{0, 1\}$. Assume

$$\psi_{ij}(z_i, z_j) = \begin{cases} \alpha & \text{if } z_i = z_j \\ \beta & \text{if } z_i \neq z_j \end{cases}$$

as the prior. The energy of this model is given as the following sum over all neighboring nodes:

$$E(Z) = \sum_{i \sim j} \psi_{ij}(z_i, z_j)$$

Show all global minima of E for

a) $\alpha = 0$ and $\beta = 1$

b) $\alpha = 1$ and $\beta = 0$

(Hints: You do not need to prove that you found all optima. Showing an example for a single $N \geq 3$ with instructions for how to construct the solutions for larger N is enough.)

★ **Bonus Exercise 3.** (+2 P.)

Consider a model with binary variables and without unary terms again. This time the structure of the graphical model is arbitrary and not necessarily a grid. The regularizer is again given by:

$$\psi_{ij}(z_i, z_j) = \begin{cases} \alpha & \text{if } z_i = z_j \\ \beta & \text{if } z_i \neq z_j \end{cases}$$

a) Is it possible to find Z such that $E(Z) = 0$ for $\alpha = 0$ and $\beta = 1$ for all such models with binary variables regardless of the structure / topology of the graphical model? Describe how.

b) Is it possible to find Z such that $E(Z) = 0$ for $\alpha = 1$ and $\beta = 0$ for all such models with binary variables regardless of the structure / topology of the graphical model? If not, which properties must be fulfilled to be able to find Z such that $E(Z) = 0$?

Exercise 4. 12 P.

The goal of this exercise is to use Gibbs sampling to sample from different priors and posteriors.

Assume a $N \times M$ grid where each variable can take S different states. Start with a random image and use the Gibbs sampling algorithm discussed in the lecture. You can choose suitable values for N, M, $\alpha \geq 1$ and $S > 3$ yourself. Use the following priors:

1. $\psi_{ij}(z_i, z_j) = \begin{cases} 0 & \text{if } z_i = z_j \\ \alpha & \text{if } z_i \neq z_j \end{cases}$
2. $\psi_{ij}(z_i, z_j) = \alpha |z_i - z_j|$

a) Run your sampler for the regularisation terms 1) and 2). Plot different samples from the prior distributions. Plot the mean taken over many samples. Comment on your results.

   Hints:

   - Use `numpy.random.choice` with the optional argument `p` to sample from a non-uniform distribution.
   - If your sampler does not converge you can increase α. I used $\alpha = 2$, $S = 4$, $N = 40$ and $M = 40$.

b) Now set the unary terms to the class probabilities from the first exercise sheet. Use the class probabilities you predicted on the first exercise sheet for the test image. If you do not have the probabilities you can load them from the file predictions.h5, dataset 'test', that you can find on the homepage. Use the priors from 1) and 2). Modify your Gibbs sampler to sample from the posterior distribution. Plot the mean taken over many samples. Comment on your results.

c) $\star$ Looping through all pixels in each sampling step makes the Gibbs sampling very slow. One possibility of speeding up the algorithm is to perform the updates on a checkerboard scheme (see e.g. "Outperforming the Gibbs sampler empirical estimator for nearest-neighbor random fields", section 5). In every iteration, first all pixels located on a black field are updated, conditioned on the white fields. Next, the white fields are updated conditioned on the black fields. As the updates for all black fields (all white fields, respectively) can be done in parallel, this is much faster than looping over all pixels. Modify the Gibbs sampler to use the checkerboard scheme. Use the priors 1) and 2) and sample from the posterior. Plot the mean taken over many samples. Comment on your results. (+4P.)

You can use the template template_exc_02_4.ipynb if you want to, which is also available on the website.