Exercise Sheet 04

Deadline: 16.05.2018, 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 **EX04** 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. 8 P.

Consider the graphical model in figure 1 with tree structure and variables $x_i \in \{0,1\}$ and $\Psi_1(x_1) = [11, 3]$, $\Psi_2(x_2) = [6, 3]$, $\Psi_3(x_3) = [1, 10]$, $\Psi_4(x_4) = [5, 7]$, $\Psi_{ij}(x_i, x_j) = \begin{bmatrix} 3 & -10 \\ -10 & 3 \end{bmatrix}$

a) Use belief propagation *by hand* and write down all steps and messages required to find the MAP solution. Although the final result will be the same no matter which node you select as the root node please pick x_1 for this exercise.

b) Write code that enumerates all configurations and calculates their energies. Confirm that the solution you found by hand corresponds to the configuration with minimal energy.

★ Bonus Exercise 2. 8 P.

Consider a graphical model with variables $x_i \in \{0, ..., D\}$ and chain structure with unary and pairwise terms.

Write a function which finds the optimal configuration for such graphical models by implementing Dijkstra's shortest path algorithm.

Test your algorithm for $D > 2$ and different unary and pairwise potentials.

Exercise 3. 8 P.

In this and the following exercise we will implement dual decomposition with subgradient descent[1] for the retina layer segmentation problem.

[1] see for example: MRF energy minimization and beyond via dual decomposition, Komodakis et al.

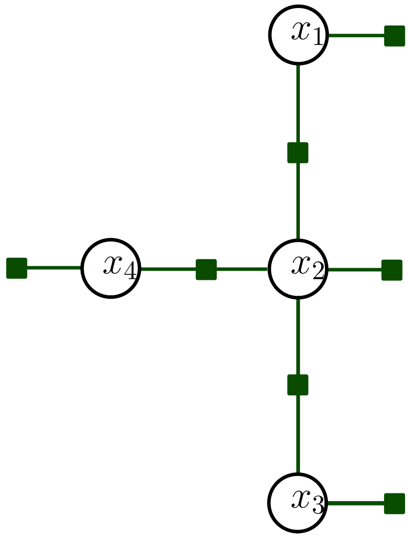

Figure 1: Graphical model with tree structure.

As in previous exercises, we want to classify each pixel in the images to either belong to background (class 0) or one of the 4 retina layers (classes 1 to 4). We formulate the problem as a graphical model where each pixel i corresponds to a node $x_i \in \{0, ..., 4\}$. We add smoothing pairwise potentials between neighbouring nodes along the rows of the image. Along the columns of the image we want to use pairwise potentials that favor the known order of the layers in the retina. We are hence aiming to solve the optimization problem

$$\begin{aligned} \min_x \; & \sum_i \psi_i(x_i) \\ & + \sum_i \sum_{j \in N^R(x_i)} \psi^R_{i,j}\left(x_i, x_j\right) \\ & + \sum_i \sum_{j \in N^C(x_i)} \psi^C_{i,j}\left(x_i, x_j\right) \end{aligned} \tag{1}$$

where $N^R(x_i)$ is the neighborhood of x_i along the row (i.e. the pixels to the left and right of x_i) and $N^C(x_i)$ the neighborhood along the column (i.e. the pixels above and below x_i).

This optimization problem does not have tree structure and is now decomposed into two subproblems: one problem consists only of the rows of the image and the pairwise interactions along the rows and the other problem consists of the columns of the image and the pairwise interactions along the columns.

a) Draw the graph corresponding to this model for an image with 4×4 pixels.

b) Plot the labels for the three training images and the test image (see exercise sheet 1) when only the unary terms are used. (just as a reminder what it looked like)

c) The pairwise smoothing along the rows is given by

$$\psi^R_{i,j}(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j \\ C_1 & \text{if } |x_i - x_j| = 1 \\ C_2 & \text{if } |x_i - x_j| > 1 \end{cases}$$

with C_1 and C_2 representing the costs for a class jump between neighboring pixels with $C_1 < C_2$.

Solve the subproblem

$$\min_x \sum_i \psi_i(x_i) + \sum_i \sum_{j \in N^R(x_i)} \psi^R_{i,j}(x_i, x_j) \tag{2}$$

for your own choice of C_1 and C_2. Use Dijkstra's shortest path algorithm. Plot the resulting labeled images.

d) Write down the pairwise potential for the columns $\psi^C_{i,j}$ (as a matrix). Use the following constraints:

- neighboring pixels of the same class should not be penalized
- class transitions in the correct order should have a small cost
- transitions to classes with a class difference bigger than 1 should have a larger cost
- class transitions in the wrong order should be penalized hard
- the correct class order is $0 \to 1 \to 2 \to 3 \to 4 \to 0$. The background class 0 appears both, above and below the retina layers.

e) Solve the subproblem for the columns

$$\min_x \sum_i \psi_i(x_i) + \sum_i \sum_{j \in N^C(x_i)} \psi^C_{i,j}(x_i, x_j) \tag{3}$$

Use Dijkstra's shortest path algorithm. Plot the resulting labeled images.

Exercise 4. 4 P.

Consider again the problem introduced in Exercise 3.

Let ψ^R_i be the unaries for the row problem and ψ^C_i those for the column problem. Initialize $\psi^R_i = \psi^C_i = \frac{1}{2}\psi_i$, where ψ_i are the unary potentials given by the predictions of the random forest, saved in the file predictions.h5. Additionally let x^R_i denote the random variables in the first and x^C_i those in the second MRF and x^{R*}_i and x^{C*}_i their MAP solution found by Dijkstra's shortest path algorithm.

After solving both subproblems independently the subgradient update can be written as

$$\Delta\psi_i^R(n) = \begin{cases} 0 & \text{if } x_i^{R*} = x_i^{C*} \\ \epsilon & \text{if } x_i^{R*} \neq x_i^{C*} \text{ and } x_i^{R*} = n \\ -\epsilon & \text{if } x_i^{R*} \neq x_i^{C*} \text{ and } x_i^{C*} = n \end{cases}$$

$$\Delta\psi_i^C(n) = \begin{cases} 0 & \text{if } x_i^{R*} = x_i^{C*} \\ -\epsilon & \text{if } x_i^{R*} \neq x_i^{C*} \text{ and } x_i^{R*} = n \\ \epsilon & \text{if } x_i^{R*} \neq x_i^{C*} \text{ and } x_i^{C*} = n \end{cases}$$

$$\psi_i^R = \psi_i^R + \Delta\psi_i^R \qquad \psi_i^C = \psi_i^C + \Delta\psi_i^C$$

This update only affects those areas with disagreement: The current solutions are penalized while the solutions of the opposite problem are encouraged. This slowly moves the problems closer together and should increase the area in which they agree.

a) Implement the subgradient update step and plot for the test image

- The number of disagreements
- The areas in which both subproblems have the same solution
- The resulting labeled images

for different iterations.

b) Compute the F_1 score on the test image for all four cases: 1) only unaries, 2) only potentials along rows, 3) only potentials along columns, 4) with subgradient update steps.

Hints:

- You can either use your own implementation of Dijkstra's algorithm or use e.g. `networkx.dijkstra_path` (https://networkx.github.io/documentation/networkx-1.10/reference/generated/networkx.algorithms.shortest_paths.weighted.dijkstra_path.html) (keep in mind that you need a directed graph for the shortest path algorithm)
- Try $C_1 = 10$, $C_2 = 100$ and $\epsilon = 0.5$.
- When two solutions are equally likely the subgradient update might produce negative unaries. Just increase the energy of all values by one then to ensure that $\psi_i \geq 0$. This does not change the MAP solution since $\arg\min_x E(x) = \arg\min_x (E(x) + c)$.