Computer Vision II - Homework Assignment 2

Stefan Roth, Uwe Schmidt, Tobias Plötz, GRIS, TU Darmstadt

May 14, 2014

This homework is due on May 26, 2014 at 20:00.
Please read the instructions carefully!

Problem 1 - Graphical Models - 30 Points

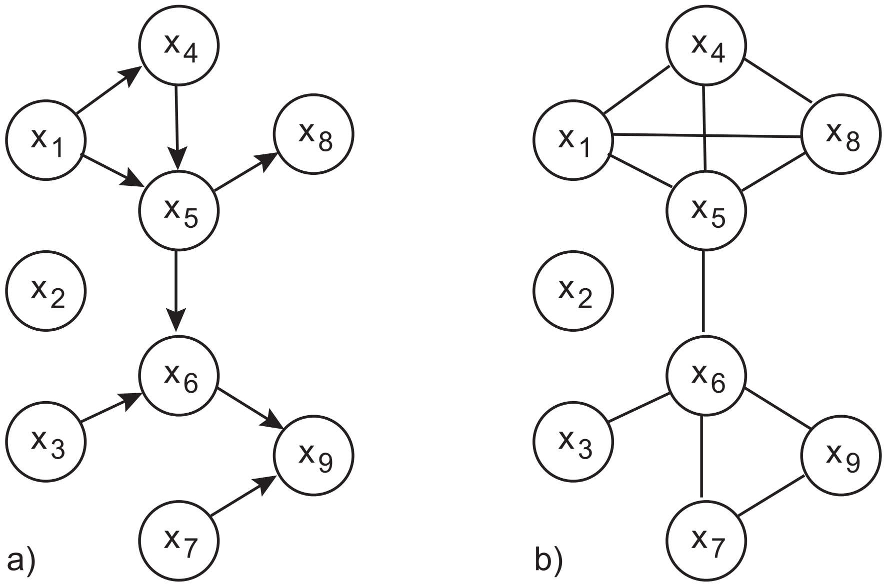

Figure 1: A directed (a) and an undirected (b) graphical model.

1. What is the main motivation of using graphical models?

   1 point

2. The joint probability model between variables $x_1, \ldots, x_7$ factorizes as

$$p(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = p(x_1)p(x_7)p(x_2|x_1)p(x_3|x_2)p(x_6|x_7)p(x_4|x_3, x_5)p(x_5|x_2, x_6)$$

   Draw a directed graphical model relating these variables. Which variables form the Markov blanket of variable x_2?

   2 points

Definition. In a directed graphical model, a *Markov blanket* of a random variable x is defined as a set of all parents, children, and all other parents of the children of x. For an undirected graphical model, the Markov blanket of a node is defined as the set of all direct neighbors of that node.

3. Write out the factorizations corresponding to the directed and the undirected graphical models in figure 1 a) and b).

   3 points

4. An undirected graphical model G has the form

$$p(x_1, \ldots, x_5) = \frac{1}{Z}\phi_1(x_1, x_2, x_3)\phi_2(x_3, x_4)\phi_2(x_3, x_5)\phi_3(x_1, x_5)\phi_4(x_4, x_5)$$

   Draw the undirected graphical model that corresponds to this factorization. A graphical model can have multiple factorizations. Give a more compact factorization of G.

   3 points

5. Consider the undirected graphical model of binary variables $\{x_i\}_{i=1}^3 \in \{0, 1\}$ defined by

$$p(x_1, x_2, x_3) = \frac{1}{Z}\phi(x_1, x_2)\phi(x_2, x_3)\phi(x_3, x_1)$$

   where the function ϕ is defined by

$$\phi(0, 0) = 1, \qquad \phi(0, 1) = 0.2, \qquad \phi(1, 1) = 3, \qquad \phi(1, 0) = 0.4$$

   Compute the probability of each of the 8 possible states of this system.

   3 points

6. What is the Markov blanket for each of the variables in figures 2 and 3?

   5 points

7. Show that the stated patterns of independence and conditional independence in figure 2 and figure 3 are true.

   5 points

8. A *factor graph* is a third type of graphical model that depicts the factorization of a joint probability. As usual it contains a single node per variable, but it also contains one node per factor (usually indicated by a solid square). Each factor variable is connected to all of the variables that are contained in the associated term in the factorization by undirected links. For example, the factor node corresponding to the term $p(x_1|x_2, x_3)$ in a directed model would connect to all three variables x_1, x_2 and x_3. Similarly, the factor node corresponding to the term $\phi_{12}(x_1, x_2)$ in an undirected model would connect variables x_1 and x_2. Figure 4 shows two examples of factor graphs. Draw the factor graphs corresponding to the graphical models in figures 2 and 3. You must first establish the factorized joint distribution associated with each graph.

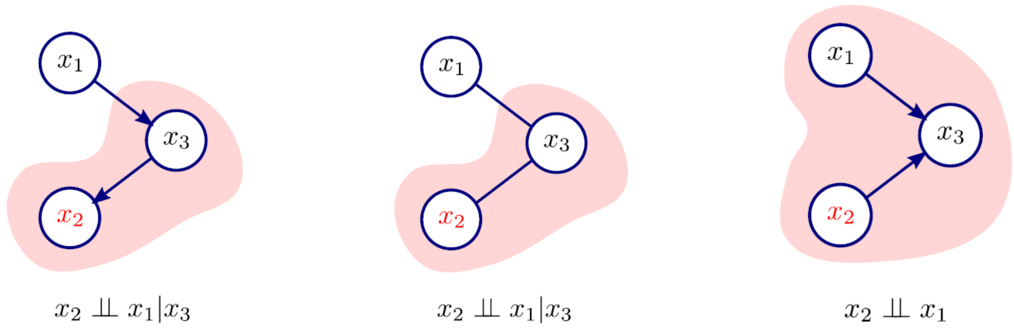

Figure 2: Directed vs. undirected graphical models. Left: Directed graphical model with three nodes. There is only one conditional independence relation implied by this model: the node x_3 is the Markov blanket of node x_2 (shaded area) and so $x_2 \perp\!\!\!\perp x_1|x_3$, where the notation $\perp\!\!\!\perp$ can be read as 'is independent of'. Middle: This undirected graphical model implies the same conditional independence relation. Right: Second directed graphical model. The relation $x_2 \perp\!\!\!\perp x_1|x_3$ is no longer true, but x_1 and x_2 are independent if we don't condition on x_3 so we can write $x_2 \perp\!\!\!\perp x_1$. There is no undirected graphical model with three variables that has this pattern of independence and conditional independence.

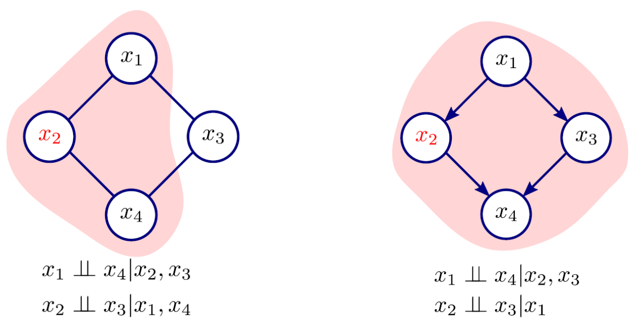

Figure 3: Directed vs undirected models. Left: This undirected graphical model induces two conditional independence relations. However, there is no equivalent directed graphical model that produces the same pattern. Right: This directed graphical model also induces two conditional independence relations but they are not the same. In both cases the shaded region represents the Markov blanket of variable x_2.

5 points

9. What is the Markov blanket of variable w_2 in figure 5 (left)?

1 points

10. What is the Markov blanket of variable w_8 in figure 5 (right)?

1 points

11. What can you say about a random variable x if you know its Markov blanket?

1 points

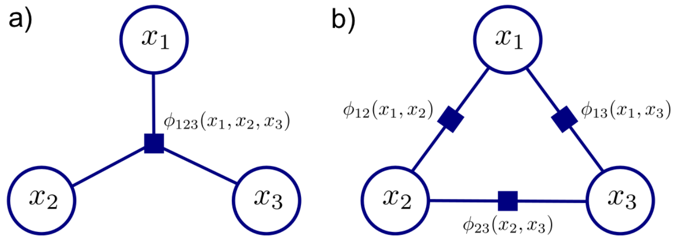

Figure 4: Factor graphs contain one node (square) per factor in the joint pdf as well as one node (circle) per variable. Each factor node is connected to all of the variables that belong to that factor. This type of graphical model can distinguish between the undirected graphical models a) $p(x_1, x_2, x_3) = \frac{1}{Z}\phi_{123}(x_1, x_2, x_3)$ and b) $p(x_1, x_2, x_3) = \frac{1}{Z}\phi_{12}(x_1, x_2)\phi_{23}(x_2, x_3)\phi_{13}(x_1, x_3)$.

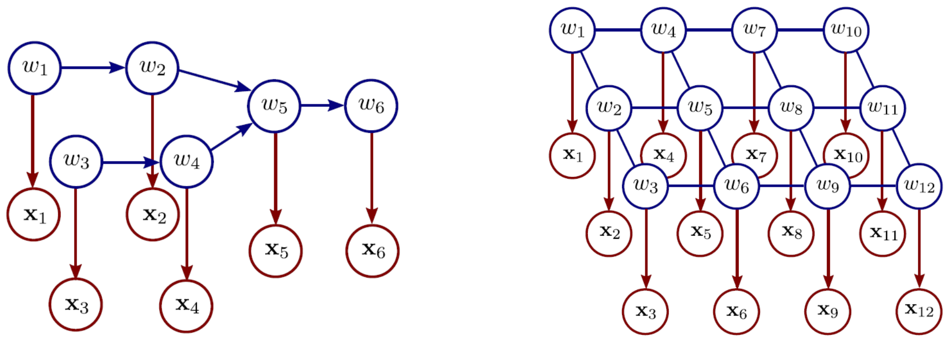

Figure 5: Commonly used graphical models in computer vision. Left: Markov tree. Right: Markov random field (MRF) prior with independent observations.

Problem 2 - Explaining away - 10 points

We want to do inference at lunch time! Assume that you just grabbed your meal in the cafeteria and are now looking for an empty seat for you and your mates. The probability of getting a seat should depend on two factors: First, if there has just been a lecture in the Audimax, the cafeteria tends to be full of people. Second, if it is rainy outside many students prefer to stay within the cafeteria to study instead of going to the Herrngarten, thus taking up additional spots. That said, our model contains three variables S, L and R that reflect the event of getting an empty seat, a lecture just being finished in the Audimax and rainy weather, respectively. All variables are binary and take the value 1 if the event occurred and 0 otherwise. We assume that L and R are a priori independent. This model can be cast as a directed graphical model as shown in figure 6.

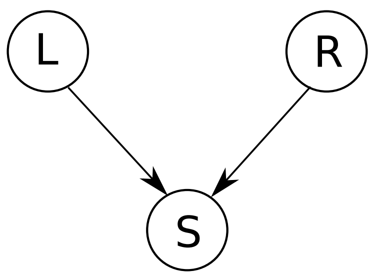

Figure 6: A graphical model capturing the independence relationships between the variables in the cafeteria model.

The probabilities for our model are defined as follows

$$p(R = 1) = 0.2$$
$$p(L = 1) = 0.1$$

$$p(S = 1|R = 0, L = 0) = 0.8$$
$$p(S = 1|R = 0, L = 1) = 0.3$$
$$p(S = 1|R = 1, L = 0) = 0.2$$
$$p(S = 1|R = 1, L = 1) = 0.1$$

Please answer the following questions (and show your calculations!)

- You observe that there is no empty seat in the cafeteria ($S = 0$). What is the probability that a lecture in the Audimax has just finished?

  4 points

- While still looking for a place to sit down (your food is cold by now) you walk by the window and observe that it is rainy outside ($R = 1$). Given this additional information, what is the probability that a lecture in the Audimax has just finished?

  4 points

- Give an intuition for your results.

  2 points

Problem 3 - Markov Random Field Priors - 15 points

In class we have seen how we can formulate a prior model of disparity maps using Markov random fields (MRFs). In particular we used pairwise MRFs, which are simple enough to realize, but still quite powerful. In this model we represent the pixels with nodes in a graphical model and connect every node with its 4 nearest neighbors (left, right, top, bottom). The edges between the horizontal and vertical edges induce a factor in the corresponding probability density, which describes the compatibility of the neighboring pixels. For a disparity map $\mathbf{x}$ of height N and width M, we write such a prior model of disparities as

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{i=1}^{N-1} \prod_{j=1}^{M} f_V(x_{i+1,j} - x_{i,j}) \cdot \prod_{i=1}^{N} \prod_{j=1}^{M-1} f_H(x_{i,j+1} - x_{i,j}). \tag{1}$$

To model the horizontal and vertical potentials f_H and f_V we use Student-t distributions:

$$f_{H/V}(d) = \left(1 + \frac{1}{2\sigma_{H/V}^2} d^2\right)^{-\alpha_{H/V}}. \tag{2}$$

For simplicity, we assume that the parameters of the horizontal and vertical potentials are the same, i.e., $\sigma_H = \sigma_V = \sigma$ and $\alpha_H = \alpha_V = \alpha$. Since we will not need the normalization term $1/Z$, we will ignore it in the following (i.e., you can assume that it is 1).

Tasks:

- Implement this MRF prior with Student-t potentials. In particular write a function

```
lp = mrf_log_prior(x, sigma, alpha)
```

  that computes the log of the unnormalized MRF prior density for a disparity map `x` with parameters `sigma` and `alpha`.

  5 points

- Compute and report the log-prior density of the ground truth disparity of the Tsukuba dataset (from assignment 1) using $\sigma = 1$ and $\alpha = 1$.

  1 points

- Create a random "noise map" of the same dimensions as the Tsukuba disparity map by drawing uniform random numbers in $[0, 16]$ independently for each pixel. Compute and report the log-prior density for this "noise map".

  3 points

- Create a "constant map" ($\mathbf{x} \equiv 8$) again of the same dimensions and also compute and report the log-prior density.

  3 points

- Compare the values of the log-prior density for these three disparity maps and comment on what the relative values say about how well this MRF model captures the properties of the visual world.

  3 points

Note:

- Make sure that your implementation contains terms for *all* horizontally and vertically neighboring pixels.
- We assume here that the disparity map has disparities in the range $[0, 16]$. You should divide all values by 16 to obtain the actual disparity in pixels.
- Since your code needs to be reasonably fast, points will be subtracted for the usage of loops. Please use matrix/vector operations to implement everything[1].

[1] cf. http://www.mathworks.de/de/help/matlab/matlab_prog/vectorization.html

Bonus Problem 4 - Stereo with gradient-based optimization - 15 points

In this problem we will put things together and implement an algorithm for stereo estimation using gradient-based MAP estimation. We use the model of the previous problem consisting of:

- A pairwise MRF prior over disparity values with Student-t potentials
- Our stereo likelihood with Student-t potentials

In the current and previous exercise you have implemented code for evaluating the (unnormalized) log-probability of a disparity map given two input images. For the gradient-based optimization we now need – gradients! For the following implementation problems make sure that your code works with continuous disparities.

- Implement the function

  ```
  f = dx_studentt(x, sigma, alpha)
  ```

  that calculates the gradient of the log-density of the Student-t distribution. Write the function such that it can also operate on matrix-valued inputs for which it should calculate the gradient element-wise.

- Implement the gradient of the log-prior. Write a function

  ```
  g = mrf_grad_log_prior(d, sigma, alpha)
  ```

  that computes the gradient of the log-prior density w.r.t. every disparity value in **d** with the given parameters. Note that the output array `g` should have the same dimensions as the input image. You can (and should) make use of the previously implemented function `dx_studentt`.

- Implement the gradient of the log-likelihood. Write a function

  ```
  g = mrf_grad_log_likelihood(d, I0, I1, sigma, alpha)
  ```

  that computes the gradient w.r.t. every disparity value given the input images $\mathbf{I}^0$ and $\mathbf{I}^1$ and the parameters σ and α. Again, `dx_studentt` should help you with that.

- Now, put things together by implementing the gradient of the log-posterior distribution. Write a function

  ```
  g = mrf_grad_log_posterior(d, I0, I1, sigma, alpha)
  ```

  that computes the gradient w.r.t. every disparity value given the input images $\mathbf{I}^0$ and $\mathbf{I}^1$ and the parameters σ and α.

- With the log-posterior and its gradient at our fingertips, we can now estimate the disparity map. Implement a function

  ```
  d = stereo(I0, I1, sigma, alpha)
  ```

  that takes two input images $\mathbf{I}^0$ and $\mathbf{I}^1$ as well as parameters for the Student-t potentials. It should optimize the log-posterior to a local optimum by a gradient-based method. We recommend that you use `minFunc`[2] for this task.

[2] http://www.di.ens.fr/~mschmidt/Software/minFunc.html

Now, evaluate your algorithm with the supplied stereo image pair `i0.ppm` and `i1.ppm` with ground truth `gt.pgm`. Write a script `a2p4` that loads the images and invokes your stereo method. After your algorithm has finished, show the ground truth disparity, your estimated disparity map, and the differences between both.

- What are your findings when you initialize the disparity map with all zeros? What do you observe when you initialize with the ground truth disparity? Why does your algorithm (probably) perform so poorly?
- Now implement coarse-to-fine estimation. For this, use subsequently downsampled versions[3] of each input image and run the algorithm first on the coarsest scale. For the next iteration initialize with an upscaled version of the current estimate of the disparity map (do not forget to scale the disparity values after upsampling).
- Show the current estimate of the disparity map after each iteration.

Note: For this problem, you will get bonus points that you can use to make up for missing points from other problems and exercises. Code that you write for this problem will help you in further assignments, so we recommend that you give this problem a try, although it requires a certain effort to solve the tasks. Make sure all your external dependencies are bundled within your final submission and that your code runs out of the box, i. e. set up paths in `a2p4.m` if necessary.

[3] If available, consider reusing your CV1 code for computing the Gaussian pyramid.