

Computer Vision 2

Assignment 2



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Summer Semester 2019
Group 8
Moritz Fuchs & Diedon Xhiha

Problem 1 Graphical models (20 Points)

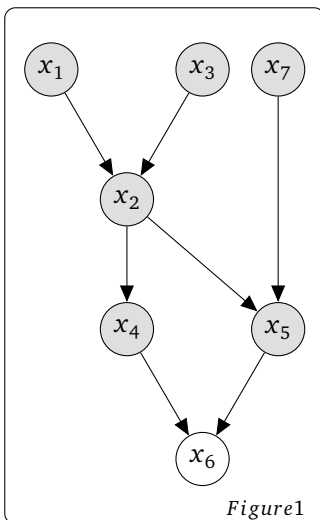
1. 1 Point

Two main ingredients the set of random variables describing the entities involved the problem, and the second ingredient is the set of conditional probabilities, that tell us the relation between a certain variable and another variable or variables from the set.

2. 1 Point

As we know that directed graphs are versatile, but not always appropriate. So they are not always be convenient to provide conditional distributions, and some of certain conditional independence structures that a directed graph can not represent. So for example Loopy graph, it is not possible to express the same conditional independence statements using the a directed graph model.

3. 2 Points



The Markov Blanket of variable x_2 is formed from the variables that are filled with gray color.

4. 2 Points

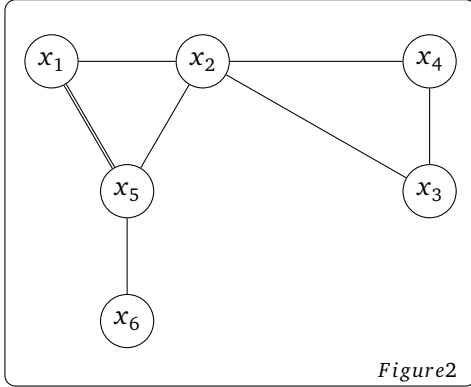
The factorization of the directed graphical model is as follows

$$p(x_1, x_2, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}) = p(x_1)p(x_2)p(x_3)p(x_6)p(x_7)p(x_4|x_1, x_2)p(x_5|x_2, x_3)p(x_{10}|x_7)p(x_{11}|x_{10})p(x_8|x_4, x_5)p(x_9|x_3, x_5, x_6)p(x_{14}|x_{11})p(x_{12}|x_8, x_9)p(x_{13}|x_9)p(x_{15}|x_{12})$$

The factorization of the undirected graphical model is as follows

$$p(x_1, x_2, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}) = \frac{1}{Z} \phi_0(x_3)\phi_1(x_1, x_4)\phi_2(x_2, x_4, x_5, x_8)\phi_3(x_7, x_{10})\phi_4(x_7, x_{11})\phi_5(x_{11}, x_{14})\phi_6(x_5, x_9)\phi_7(x_8, x_{12})\phi_8(x_9, x_{12}, x_{13}, x_{15})\phi_9(x_6, x_9)$$

5. 1 Point



6. 2 Points

x_1	x_2	x_3	x_4	$p(x_1, x_2, x_3, x_4)$	$\frac{1}{Z}p(x_1, x_2, x_3, x_4)$
0	0	0	0	$1*1*1*1=1$	0.1077
0	0	0	1	$1*1*0.1*0.1=0.01$	0.001077
0	0	1	0	$1*0.1*0.1*1=0.01$	0.001077
0	0	1	1	$1*0.1*2*0.1=0.02$	0.002155
0	1	0	0	$0.1*0.1*1*1=0.01$	0.001077
0	1	0	1	$0.1*0.1*0.1*0.1=0.0001$	0.00001077
0	1	1	0	$0.1*2*0.1*1=0.02$	0.002155
0	1	1	1	$0.1*2*2*0.1=0.04$	0.00431
1	0	0	0	$0.1*1*1*0.1=0.01$	0.001077
1	0	0	1	$0.1*1*0.1*2=0.02$	0.002155
1	0	1	0	$0.1*0.1*0.1*0.1=0.0001$	0.00001077
1	0	1	1	$0.1*0.1*2*2=0.04$	0.00431
1	1	0	0	$2*0.1*1+0.1=0.02$	0.002155
1	1	0	1	$2*0.1*0.1*2=0.04$	0.00431
1	1	1	0	$2*2*0.1*0.1=0.04$	0.00431
1	1	1	1	$2*2*2*2=8$	0.86206

We have computed Z as the sum of all the probabilities above, and after the calculation we found $Z=9.28$.

7. 2 Points

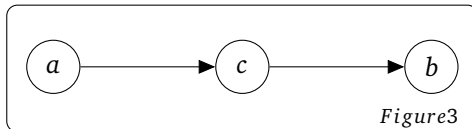
Based on those four records

a	b	c	$p(a,b,c)$
0	1	0	0.048
1	0	0	0.192
0	1	1	0.216
1	0	1	0.064

The first and second row c remain the same but a and b change and with them the distribution too, the same happens in the third and fourth row. So we can conclude that those variable a and b are marginally dependent, when $b=1$ and $a=0$ then the distribution is 0.048 but in the other hand, when $b=0$ and $a=1$ then the distribution is 0.192, so this means knowing event b does help in value of event a and vice versa.

8. 3 Points

Based on the given distributions $p(a)$ $p(b|c)$ $p(c|a)$ we can draw the directed graph as follows :



And from the figure above we can write:

$$p(a, b, c) = p(a)p(a|c)p(b|a, c) = p(a)p(a|c)p(b|c)$$

As we know from directed graph, because 'c' tell us about 'a', we can take away 'a' from $p(b|a, c)$, and therefore we get the desired probability.

9. 2 Points

Markov blanket of variables in undirected graph:

$$x_1 -> x_2, x_3$$

$$x_2 -> x_1, x_4$$

$$x_3 -> x_1, x_4$$

$$x_4 -> x_2, x_3$$

Markov blanket of variables in directed graph:

$$x_1 -> x_2, x_3$$

$$x_2 -> x_1, x_3, x_4$$

$$x_3 -> x_1, x_2, x_4$$

$$x_4 -> x_2, x_3$$

10. 4 Points

Based also in the previous question about for Markov blanket for each of the variables, we can conclude that in the figure 2, undirected graph, the first one that says x_1 is independent of x_4 and that x_2 and x_3 are the Markov blanket of x_1 , is completely true based also in the definition that, a node is conditionally independent of all other nodes given its Markov blanket. The second one that says x_2 is independent of x_3 , because the x_3 is not in the Markov blanket of x_2 , and from definition a node is conditionally independent of all other nodes given its Markov blanket, and that x_1 and x_4 are the Markov blanket of x_2 , because they are direct node of x_2 . And for the directed Graph, first one that says x_1 independent of x_4 , exactly as the definition from above, and that x_2 and x_3 are the Markov blanket of x_1 , because x_1 has 2 children and no other parents and also those children have just one parent and its x_1 . The second one

Problem 2 Markov random fields with Student-t potentials (10 Points)

Tasks 1-4 within code

2. 1 Point

As printed out: gt log prior: -9928.589720178397

3. 2 Points

As printed out: random disparity log prior: -478661.4937203116

4. 2 Points

As printed out: constant disparity log prior: 0.0

5. 3 Points

As we use MRF model, the potential function expresses the compatibility between neighbour pixels.

Therefore our random map has the lowest score as most neighbours don't fit each other.

The log probability of constant map is the lowest, because neighbouring pixels are as close together as they can get and we want them to be locally smooth.

The ground truth lies between those log-priors as it's not constant, therefore not maximally locally smooth and may have some small artifacts.

Problem 3 Stereo with gradient-based optimization (17 Points)

Everything else is in code.

1. 4 Points

When initialized with a constant disparity map it struggles to find another map which does not increase the function value as the prior increases in absolute value. It therefore only detects the edges which are farthest in the background and the nearest on the foreground. The rest stays constant.

The random disparity does not converge to something meaningful as it's too far away from such a solution, as neighbouring pixels are incompatible. It can be noticed that so many of the more compatible pixels merge together but to complete this would take very long.

The ground truth converges to something more blurry until it gets closer to the constant map and therefore loses its usefulness. This is probably because of the brightness constancy assumption we modeled in our prior which has the highest score when it's constant and therefore our gradient wants to converge to a smoother map, which is very smooth and does not account for object boundaries.

Problem 4 Stereo with a generalized robust function (10 Points)

Everything else is in code.

4. To obtain better results, also apply an image pyramid similar to Problem 3. (? Point/s)

Tuned the parameters to fit the normal algorithm and therefore doesn't show better results.

5. Based on your experiments, what recommendations can you give for choosing the parameters α and c ? How does each parameter influence the result? How did you go about finding good parameters (?? Point/s)

For prior: $\alpha = 0.5$ and $c = 10.0$ or ($\alpha = 3.0$ and $c = 3.0$)

For likelihood: $\alpha = 0.5$ and $c = 10.0$ or ($\alpha = 3.0$ and $c = 4.0$)

c does mainly influence how much influence an outlier has and therefore the result is smoothed. This limits the amount of detail we can e.g. see from the writing on the whiteboard. As it is small values (e.g. 1) smooth more and bigger values (e.g. 10) for sharper edges

α effects seem quite small. Results with smaller values seem to have less depth to it, but also less graininess.

We just started playing around with α and kept c fixed and then switched it. Repeated sometimes but didn't have anything bigger strategy than our intuition.

-
6. In general, using the generalized robust function and its gradients will not allow us to learn the shape parameters which is why we tune them manually. Why is it that gradient-based optimization using Eqs. (8) and (9) is not applicable here to obtain good shape parameters?(?? Point/s)
-

The proposed loss function approximates many different loss function and interpolates inbetween them. This means we would need to fix α to find the perfect c , and in next turn fix c and do the same for α . which means we would have the perfect c anymore. This result is because its a non convex function and we may only get stuck in a local optima. It is better to use the different properties of the incorpated loss functions by change the α between iteration and do as thex describe: we can initialize α such that our loss is convex and then gradually reduce α (and therefore reduce convexity and increase robustness) during optimization, thereby enabling robust estimation that (often) avoids local minima.