Master in Computer Vision Barcelona

UAB UOC UPC upf.

Module: M2. Optimization and inference techniques for Computer Vision
Teachers: Juan F. Garamendi, Adrián Martín, Coloma Ballester, Karim Lekadir, Oriol Ramos
Final exam. Date: December 3rd, 2020 **Time: 2h30min**

- All sheets of paper should have your full name.
- The answer to all questions should be accurate and concise.
- You are allowed to use your notes, slides, papers, books, etc. The corresponding pdfs can be displayed on the PC screen but you are not allowed to use the internet nor any keyboard.
- You can answer the exercises one after another (in the order you choose) without leaving unnecessary blank spaces. Please use a horizontal continuous line to delimit each exercise and before the end of the exam cross out the extra text or calculations that do not make part of the final answer.
- If you finish before the 2 hours allocated for the exam, you will have to wait until the end when you will be informed that the task is opened for submissions.
- The exam delivery should be in .pdf format, uploaded to the task.
- Change the name of the file that Adobe Scan generates; the new name will be: Name_Surname.pdf

---

Problem 1 — *Juan F. Garamendi, 2 Points*

Imagine we are back to late 2017 and you work in the NVidia Labs. The Team is working in a new neural adversarial network architecture for generating synthetic faces (known as styleGAN). The team wants to create the training dataset using high resolution images downloaded from Flickr social network.

The faces will be automatically detected and cropped, but the problem of using directly these images is the miss-alignment. To overcome this issue, the team wants to register (align) all downloaded faces to a given template, such a way all faces are place in the same position. Imagine that you are in charge of performing such alignment, and for doing the task you decide to use the library DLib. For a given image with a face, DLib gives you the coordinates of the landmarks that you can see in figure 1. An example of drawn computed landmarks on Lenna image can be seen in figure 2.

Assume for the task that the function Move_Image(I, h, v) is already implemented. I is an image, h is a (positive or negative) integer number and v is also a (positive or negative) integer number. This function takes as input an image and returns the same image displaced in the vertical direction the number of pixels given by v and in the horizontal direction the number of pixels given by h, i.e., this function translates the image I in the direction of vector (h, v).

If you use only one landmark, for example landmark $lm^{[15]}$, with coordinates $(x_{lm}^{[15]}, y_{lm}^{[15]})$, computing the values (h, v) is easy, it is just

$$\begin{cases} h = x_I^{[15]} - x_{lm}^{[15]} \\ v = y_I^{[15]} - y_{lm}^{[15]} \end{cases}$$

where $(x_I^{[15]}, y_I^{[15]})$ are the coordinates of the computed landmark number 15 for the image I.

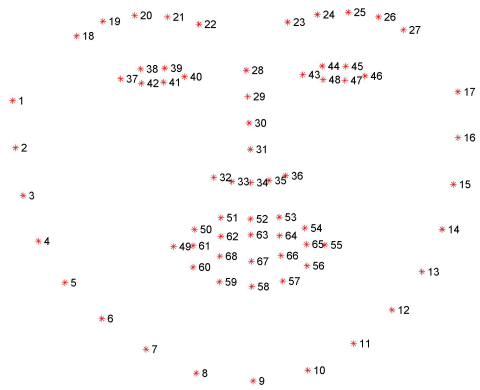


Figure 1: Landmarks computed by DLib

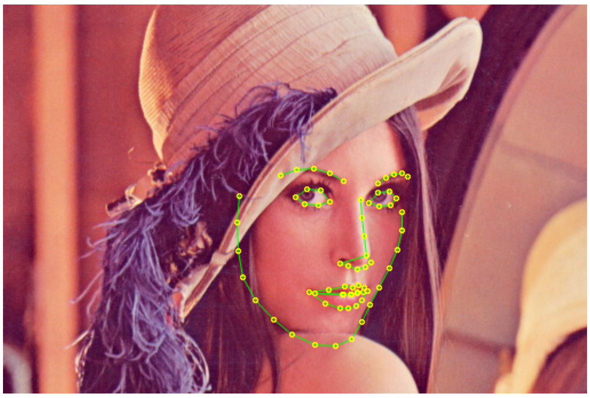


Figure 2: Examples of the landmarks over Lenna

But as you probably figured out, this will have miss aligned the rest of the landmarks, so your supervisor said that you should also use more landmarks. The problem of using more than one landmark is that you probably will not have a perfect match.

Describe in a concise and clear way a method for getting the best possible values for h and v to align the image according to the left eye (landmarks 37 to 42). To solve the problem, follow the next steps

(a) (1 point) Mathematically write down an expression E that measures the L2 error, done when the image is moved using the values h and v. Write it in matricial compact form. Obviously, to compute the error only take into account the chosen landmarks to perform the alignment, in this case, the landmarks of the left eye 37 to 42.

(b) (1 point) Mathematically write the method to optimize the previous metric.

QUESTION 1

for landmarks from i to j, the ideal perfect case with error 0 is when

$$\begin{cases} h &= & x_I^{[i]} - x_{lm}^{[i]} \\ v &= & y_I^{[i]} - y_{lm}^{[i]} \\ h &= & x_I^{[i+1]} - x_{lm}^{[i+1]} \\ v &= & y_I^{[i+1]} - y_{lm}^{[i+1]} \\ \vdots &= & \vdots \\ h &= & x_I^{[j]} - x_{lm}^{[j]} \\ v &= & y_I^{[j]} - y_{lm}^{[j]} \end{cases}$$

so, if the matching is not perfect, there is an error on h as well as on v. Among different errors, we choose

$$E_h = \sum_{k=i}^{j} \left(h - \left(x_I^{[k]} - x_{lm}^{[k]} \right) \right)^2$$

$$E_v = \sum_{k=i}^{j} \left(v - \left(y_I^{[k]} - y_{lm}^{[k]} \right) \right)^2$$

$$E = E_h + E_v = \sum_{k=i}^{j} \left(h - \left(x_I^{[k]} - x_{lm}^{[k]} \right) \right)^2 + \left(v - \left(y_I^{[k]} - y_{lm}^{[k]} \right) \right)^2$$

In matricial form, $E = |A\bar{w} - \bar{z}|^2$ where $|\cdot|$ denotes usual eculidean norm and

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \end{pmatrix}_{(j-i+1)\times 2}, \quad \bar{w} = \begin{pmatrix} h \\ v \end{pmatrix}, \quad \bar{z} = \begin{pmatrix} x_I^{[i]} - x_{lm}^{[i]} \\ y_I^{[i]} - y_{lm}^{[i]} \\ x_I^{[i+1]} - x_{lm}^{[i+1]} \\ y_I^{[i+1]} - y_{lm}^{[i+1]} \\ \vdots \\ x_I^{[j]} - x_{lm}^{[j]} \\ y_I^{[j]} - y_{lm}^{[j]} \end{pmatrix}_{(j-i+1)\times 2}$$

QUESTION 2
This is a least squares problem so it can be solved with the normal equations

$$A^t A\bar{w} - A^t\bar{z} = 0$$

i.e,

$$\bar{w} = (A^t A)^{-1} A^t \bar{z}$$

Problem 2 *Juan F. Garamendi, 1 Point*

Say whether the next statements are true (**T**) or false (**F**) [Correct: +1/5, Incorrect: -1/5, unanswered: 0 points].

(a) Let $J(u)$ be an energy functional that we are minimizing. If we use a gradient descent scheme to minimize it, the found solution u satisfies that $J(u) = 0$

(b) Let $J(u)$ be an energy functional that we are minimizing. If we use a gradient descent scheme to minimize it, the found solution u satisfies that $\nabla u = 0$

(c) Let $J(u)$ be an energy functional that we are minimizing. If we use a gradient descent scheme to minimize it, the found solution u satisfies that $\Delta u = 0$

(d) Using Von Neumann boundary conditions is always better than using other kind of boundary conditions.

(e) Using homogeneous Von Neumann boundary conditions is always better than using other kind of boundary conditions.

All are false

(a) Using Von Neumann boundary conditions is always better than using other kind of boundary conditions. False because the boundary conditions depends on the problem. For example, In inpaintig you have homogeneous Neumann and Dirichlet.

(b) Using homogeneous Von Neumann boundary conditions is always better than using other kind of boundary conditions. False because the boundary conditions depends on the problem. For example, In inpaintig you have Neumann homogeneous and Dirichlet.

(c) Let $J(u)$ be an energy functional that we are minimizing. If we use a gradient descent scheme to minimize it, the found solution u satisfies that $\Delta u = 0$. False, $\Delta u = 0$ means that the gradient of a function is constant, something that has not to be necessarily true.

(d) Let $J(u)$ be an energy functional that we are minimizing. If we use a gradient descent scheme to minimize it, the found solution u satisfies that $\nabla u = 0$. False, $\nabla u = 0$ is a constant function, i.e. if u is an image, it will be a one color image. What is is satisfied is that $J(u)$ is minimum for all possible u.

(e) Let $J(u)$ be an energy functional that we are minimizing. If we use a gradient descent scheme to minimize it, the found solution u satisfies that $J(u) = 0$. False, $J(u)$ can have a minimum in u but its energy can be any value, although this value is the minimum.

Problem 3 Adrián Martín *1 Point*

Please briefly justify your answer for every question.

(a) In which situation would you choose an Stochastic Gradient Descent over a Mini-batch gradient descent? [0.2p]

(b) When would be a Mini-batch gradient descent equivalent to a Batch gradient descent? [0.2p]

(c) What is the main advantage of using a Batch gradient descent over the other two methods? [0.2p]

(d) Why do frequent/infrequent parameters need small/large updates in Adagrad? [0.2p]

(e) What is the main advantage of using RMSprop or Adam instead of Adagrad? [0.2p]

(a) *In which situation would you choose an Stochastic Gradient Descent over a Mini-batch gradient descent?*

Hardware (GPU memory) limitations are the main reason to choose SGD over mini-batch gradient descent. Avoiding fixing another hyper-parameter as the mini-batch size is also valid answer. Choosing SGD because is faster has been also considered partially correct, this is not completely true because in most implementations speed is equivalent when mini-batches fit on GPU memory.

(b) *When would be a Mini-batch gradient descent equivalent to a Batch gradient descent?*

Mini-batch gradient descent is a batch gradient descent if the size of the mini-batch is equal to the whole dataset.

(c) *What is the main advantage of using a Batch gradient descent over the other two methods?*

Batch gradient descent is guaranteed to converge to global minimum for convex error surfaces and to a local minimum for non-convex surfaces.

(d) *Why do frequent/infrequent parameters need small/large updates in Adagrad?*

When data is sparse or unbalanced it is typical to find parameters that are more frequently updated than others. If this is not corrected the optimization is mainly driven by large updates of these parameters resulting in negligible changes in parameters less frequently updated.

(e) *What is the main advantage of using RMSprop or Adam instead of Adagrad?*

Both algorithms, RMSprop and Adam, reduce the aggressive decreasing of gradients from Adagrad by substituting the accumulation of squared gradients by a moving average of squared gradients.

Problem 4 Adrián Martín *1 Point*

Consider the following sets of constraints:

i) $x_1 \cdot x_2 \geq 0.$

ii) $\begin{cases} x_1^2 + x_2^2 - 4 \geq 0 \\ x_1 - x_2 + 4 \geq 0, \\ x_1 \cdot x_2 \leq 0. \end{cases}$

iii) $\begin{cases} x_1^2 + x_2^2 \leq 4 \\ x_1 \cdot x_2 \leq 0, \\ x_2 \geq 0. \end{cases}$

(a) Sketch the three sets of constraints. Are they convex? Justify your answer. [0.6p]

(b) Now consider the following minimization problem

$$\min_{x_1, x_2} \quad -x_1 + 2x_2$$

and the constrained optimization problem resulting from solving this minimization within **the set of constraints from the previous ones that is convex** and write its KKT optimality conditions. [0.4p]

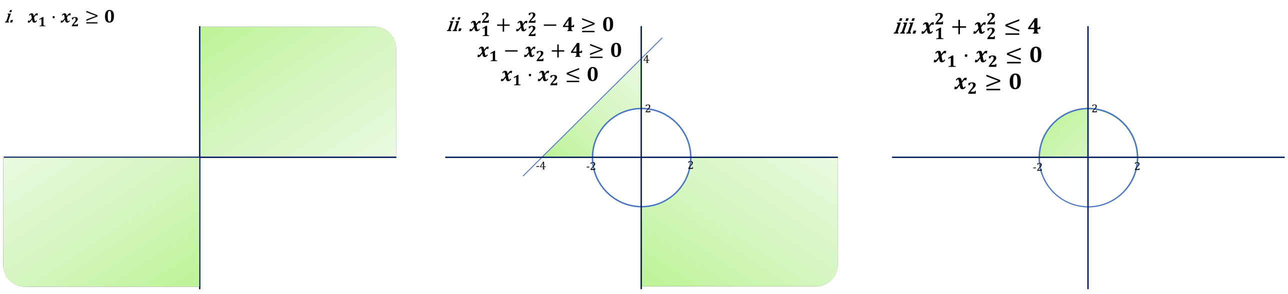


Figure 3: *i.* Non-convex. *ii.* Non-convex. *iii.* Convex

The resulting convex minimization problem is then

$$\begin{aligned} \min -x_1 + 2x_2 &\quad \text{subject to} \\ -x_1^2 - x_2^2 + 4 &\quad \geq 0 \\ -x_1 \cdot x_2 &\quad \geq 0 \\ x_2 &\quad \geq 0. \end{aligned}$$

The lagrange function is

$$\mathcal{L}(x, \lambda) = -x_1 + 2x_2 - \lambda_1(-x_1^2 - x_2^2 + 4) - \lambda_2(-x_1 \cdot x_2) - \lambda_3 x_2$$

and its KKT optimality conditions are:

$$\begin{cases} -1 + 2\lambda_1 x_1 + \lambda_2 x_2 = 0 \\ 2 + 2\lambda_1 x_2 + \lambda_2 x_1 - \lambda_3 = 0 \\ -x_1^2 - x_2^2 + 4 \geq 0 \\ -x_1 \cdot x_2 \geq 0, \\ x_2 \geq 0 \\ \lambda_1 \geq 0,\ \lambda_2 \geq 0,\ \lambda_3 \geq 0 \\ \lambda_1(-x_1^2 - x_2^2 + 4) = 0 \\ \lambda_2(-x_1 \cdot x_2) = 0 \\ \lambda_3 x_2 = 0 \end{cases}$$

Problem 5 Coloma Ballester *0.75 Points*

For the following statements indicate True or False. Each correct answer counts +0.25 points; incorrect answer counts -0.25 points; items with no answer will not add or subtract any point.

(a) The minimization problem

$$\min_x \|Dx\| + \frac{\eta}{2}\|x-a\|^2 + \langle b, x\rangle$$

(where $a, b \in \mathbb{R}^n$, $\eta > 0$, D real and symmetric matrix of size $n \times n$ are given and $x \in \mathbb{R}^n$) has a solution which can be obtained by a primal-dual strategy whose primal-dual algorithm to solve it is

$$\xi^{k+1} = P_C(\xi^k + \tau D x^k)$$
$$x^{k+1} = x^k - \theta\left(D\xi^{k+1} + \eta(x^k - a) + b\right)$$

where x^1, ξ^1 are given, $k \in \mathbb{N}, \tau, \theta > 0$, and $P_C(v) = \frac{v}{\max\{1, \|v\|\}}$.
True

(b) Given an optimization problem that can be solved by duality, and given the primal-dual algorithm to obtain a solution of it, the Duality Gap can be used as stopping criteria for the iterative process but only in the case that the objective function is quadratic with respect to the primal variable.
False

(c) Let A be an $m \times n$ matrix, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$. The minimization problem for $x \in \mathbb{R}^n$

$$\min -\|x-c\|^2$$
$$\text{subject to } Ax - b \geq 0.$$

can be solved by duality.
False

Problem 6 Coloma Ballester *0.75 Points*

Given $x_1, x_2, ..., x_n \in \mathbb{R}^p$, let's write them in a matrix X of size $n \times p$ where $x_1, x_2, ..., x_n$ are the rows of X. Let y be a vector of n entries equal to either $+1$ or -1, that is, $y \in \{-1, +1\}^n$. Then, the Support Machine Problem is

$$\begin{aligned} \min_{\beta,\beta_0,\xi} \quad & \frac{1}{2}\|\beta\|^2 + K\sum_{i=1}^n \xi_i \\ \text{subject to} \quad & \xi_i \geq 0,\ i = 1, ..., n \\ & y_i(<\beta, x_i> + \beta_0) \geq 1 - \xi_i,\ i = 1, ..., n \end{aligned}$$

where $\beta, \xi \in \mathbb{R}^n$, $\beta_0 \in \mathbb{R}$, and $K > 0$ a given constant. Let's write it in matrix form: by denoting $\mathbf{1}$ the vector of $\mathbb{R}^n$ with all entries equal to 1, and $\tilde{X} = \text{diag(y)}X$ where diag(y) is the diagonal matrix with the n entries of y in the diagonal, the previous optimization problem writes

$$\begin{aligned} \min_{\beta,\beta_0,\xi} \quad & \frac{1}{2}\|\beta\|^2 + K < \mathbf{1}, \xi > \\ \text{subject to} \quad & \xi \geq 0 \\ & \tilde{X}\beta + \beta_0 y \geq \mathbf{1} - \xi \end{aligned}$$

(a) Compute the dual function of this problem.

(b) Compute the dual problem.

The objective function of our constrained optimization problem is $E(\beta, \beta_0, \xi) = \frac{1}{2}\|\beta\|^2 + K < \mathbf{1}, \xi >$, with $\beta, \xi \in \mathbb{R}^n$, $\beta_0 \in \mathbb{R}$. E is: (1) quadratic with respect to $\beta \in \mathbb{R}^n$, with Hessian $H_\beta E = I =$ Identity matrix, that is, a positive definite matrix, thus convex with respect to β; (2) linear with respect to $\xi \in \mathbb{R}^n$, thus convex with respect to ξ; and (3) convex with respect to $\beta_0 \in \mathbb{R}$. Thus, E is convex with respect to its variables.

Introducing dual variables (associated to the inequality constraints), the Lagrange function associated to this constrained minimization problem writes

$$\mathcal{L}(\beta, \beta_0, \xi, \lambda, \mu) = \frac{1}{2}\|\beta\|^2 + K < \mathbf{1}, \xi > - < \lambda, \xi > - < \mu, \tilde{X}\beta + \beta_0 y - \mathbf{1} + \xi >$$

defined for $\beta, \xi, \lambda, \mu \in \mathbb{R}^n$, $\beta_0 \in \mathbb{R}$ with $\lambda_i, \mu_i \geq 0$ for all $i = 1, ..., n$ (written in the following as $\lambda, \mu \geq 0$). $\mathcal{L}$ is convex with respect to β, β_0, ξ (for λ, μ fixed), and linear, thus concave, with respect to λ, μ (for β, β_0, ξ fixed). Therefore, the duality gap is zero and there exists a saddle point. Therefore, the original Primal problem, the Primal-Dual problem, and the Dual problem are three equivalent problems. That is,

$$\begin{aligned} \min_{\beta,\beta_0,\xi} \quad & \frac{1}{2}\|\beta\|^2 + K < \mathbf{1}, \xi >= \min_{\beta,\beta_0,\xi} \max_{\lambda \geq 0, \mu \geq 0} \mathcal{L}(\beta, \beta_0, \xi, \lambda, \mu) = \max_{\lambda \geq 0, \mu \geq 0} \min_{\beta,\beta_0,\xi} \mathcal{L}(\beta, \beta_0, \xi, \lambda, \mu) = \max_{\lambda \geq 0, \mu \geq 0} g_D(\lambda, \mu) \\ \text{subject to} \quad & \xi \geq 0 \\ & \tilde{X}\beta + \beta_0 y \geq \mathbf{1} - \xi \end{aligned}$$

where $g_D(\lambda, \mu)$ denotes the dual function of the problem, which is defined as

$$g_D(\lambda, \mu) = \mathcal{L}(\beta^0(\lambda, \mu), \beta_0^0(\lambda, \mu), \xi^0(\lambda, \mu), \lambda, \mu),$$

where

$$(\beta^0(\lambda, \mu), \beta_0^0(\lambda, \mu), \xi^0(\lambda, \mu)) = \arg\min_{\beta,\beta_0,\xi} \mathcal{L}(\beta, \beta_0, \xi, \lambda, \mu)$$

$$= \arg\min_{\beta,\beta_0,\xi} \left(\frac{1}{2} < \beta, \beta > +K < \mathbf{1}, \xi > - < \lambda, \xi > - < \mu, \tilde{X}\beta > -\beta_0 < \mu, y > + < \mu, \mathbf{1} >- < \mu, \xi > \right).$$

The minimum $(\beta^0(\lambda, \mu), \beta_0^0(\lambda, \mu), \xi^0(\lambda, \mu))$ is the solution of the following zero gradient
$\nabla_\beta \mathcal{L} = \beta - \tilde{X}^t \mu = 0$,
$\nabla_{\beta_0} \mathcal{L} =< \mu, y >= 0$,
$\nabla_\xi \mathcal{L} = K\mathbf{1} - \lambda - \mu = 0$.
That is, $\beta^0(\lambda, \mu) = \tilde{X}^t \mu$, $< \mu, y >= 0$, and $K\mathbf{1} - \lambda - \mu = 0$. Thus, the dual function is

$$g_D(\lambda, \mu) = \begin{cases} -\frac{1}{2}\|\tilde{X}^t \mu\|^2 + < \mathbf{1}, \mu > & \text{if } < \mu, y >= 0, \text{ and } K\mathbf{1} - \lambda - \mu = 0 \\ -\infty & \text{otherwise.} \end{cases}$$

Finally, the dual problem is

$$\begin{aligned} \max_{\lambda,\mu} \; & -\frac{1}{2}\|\tilde{X}^t \mu\|^2 + < \mathbf{1}, \mu > \\ \text{subject to} \quad & < \mu, y >= 0 \\ & K\mathbf{1} - \lambda - \mu = 0 \\ & \lambda \geq 0 \\ & \mu \geq 0 \end{aligned}$$

(which has solution because it is a quadratic concave problem (with Hessian $H = -\tilde{X}\tilde{X}^t$ which is negative definite), with concave inequality constraints, where we have eliminated the primal variable, and therefore could be solved with a projected gradient ascent, for instance, or using the KKT optimality conditions).

Problem 7 Karim Lekadir *0.5 Points*

(a) What are the criteria that are encoded in the Chan-Vese segmentation method?
Solution: The segmentation result is an image. The segmentation image is similar to the original image. The segmented regions are homogeneous. The segmented regions have smooth boundaries. The segmentation image has two regions.

(b) Explain with your own words what are level set functions and their role in the Chan-Vese segmentation method. Be concise.
Solution: They are functions representing surfaces used to find contours by analyzing zero-crossings with the input images.

Problem 8

Oriol Ramos Terrades, 1.0 Points

We saw that color segmentation problem can be modeled as a problem of maximum a posteriori inference over the graphical model of figure 4. Given an image of size $N \times M$ pixels, the goal then was:

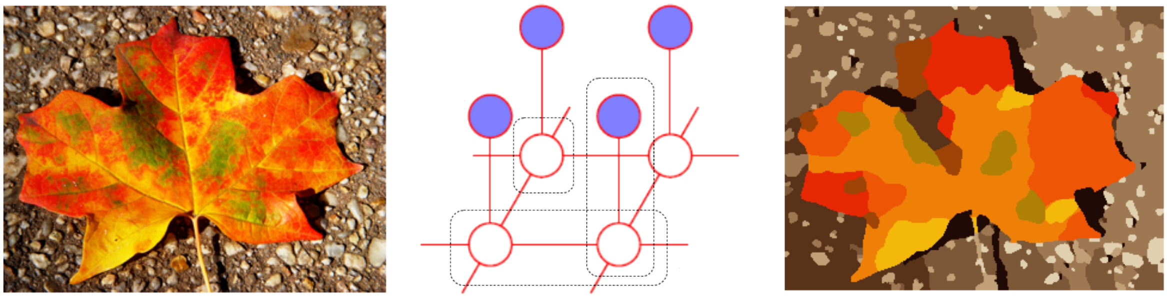


Figure 4: graphical model for color segmentation problem

$$\arg\max_y p(y|x) = \arg\max_y \prod_n p(y_n|x_n) \prod_{i,j} \phi_{i,j}(y_i, y_j)$$

(a) What are x and y ?

(b) What are the names for the $p(y_n|x_n)$ and $\phi_{i,j}(y_i, y_j)$ terms ?

(c) How many $p(y_n|x_n)$ and $\phi_{i,j}$ terms are?

(d) Propose a candidate $\phi_{i,j}$.

(e) Can $\phi_{i,j}$ have different values depending on the pixels position (i, j)?

(a) y is the sought segmented image and x is the observation or original input image.

(b) $p(y_n|x_n)$ are the unary terms, which are modeled by a conditional probability, and $\phi_{i,j}(y_i, y_j)$ are the pairwise potentials.

(c) There are $N \times N$ unary terms and $2N(N-1)$ pairwise potentials terms.

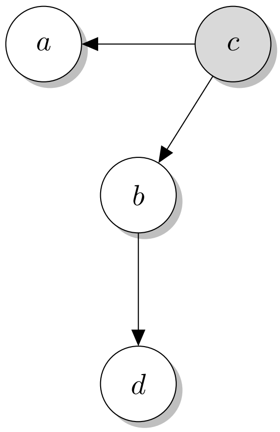
(d) If we want to segment to $k = 3$ colors, a possible $\phi_{i,j}(y_i, y_j)$ term is:

$$\begin{pmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 1.0 \end{pmatrix}$$

(e) Yes, although most of the cases is defined to not depend on pixels positions as in (d).

Problem 9 *Oriol Ramos Terrades, 1.5 Points*

Given the following Bayesian network:



$p(a\|c)$	a	c
0.3	0	0
0.7	1	0
0.2	0	1
0.8	1	1

$p(b\|c)$	b	c
0.1	0	0
0.9	1	0
0.4	0	1
0.6	1	1

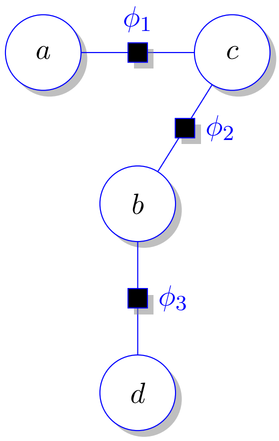
$p(d\|b)$	d	b
0.7	0	0
0.3	1	0
0.1	0	1
0.9	1	1

$p(c)$	c
0.8	0
0.2	1

with prior and conditional probabilities as defined above:

a) Convert it to a factor graph and define the joint probability [0.3 points].

Solution:



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

b) Propose the corresponding factor functions, given the factor graph proposed in a) [0.2 points].

Solution: a possible definition of factor functions are:

$$\phi_1(a, c) = p(a|c)p(c) = \begin{pmatrix} 0.3 & 0.7 \\ 0.2 & 0.8 \end{pmatrix} \begin{pmatrix} 0.8 & 0.8 \\ 0.2 & 0.2 \end{pmatrix} = \begin{pmatrix} 0.24 & 0.56 \\ 0.04 & 0.16 \end{pmatrix}$$

$$\phi_2(b, c) = p(b|c) = \begin{pmatrix} 0.1 & 0.9 \\ 0.4 & 0.6 \end{pmatrix}$$

$$\phi_3(b, d) = p(d|b) = \begin{pmatrix} 0.7 & 0.1 \\ 0.3 & 0.9 \end{pmatrix}$$

c) Compute the message received by a. Assume that c has been observed as $c = 1$ [0.2 points].

Solution:

$$m_{a\leftarrow 1} = \int \phi_1(a, c)^t m_{c\to 1}(c)dc = \begin{pmatrix} 0.24 & 0.04 \\ 0.56 & 0.16 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.04 \\ 0.16 \end{pmatrix}$$

d) Compute the belief of b. Remember to send messages from all variables that will have an impact in this belief [0.6 points].

Solution:
We can apply message passing since the PGM from c to d is a chain. we first send a message from c to b:

$$m_{b\leftarrow 2} = \int \phi_2(b,c)^t m_{c\rightarrow 2}(c)dc = \begin{pmatrix} 0.1 & 0.4 \\ 0.9 & 0.6 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.4 \\ 0.6 \end{pmatrix}$$

Now we sent a message from d to b:

$$m_{b\leftarrow 3} = \int \phi_3(b,d)^t m_{d\rightarrow 3}(d)dd = \begin{pmatrix} 0.7 & 0.3 \\ 0.1 & 0.9 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Finally, we combine both messages to get the belief.

$$b(b) = \frac{1}{Z_b} m_{b\leftarrow 2} \circ m_{b\leftarrow 3} = \frac{1}{Z_b} \begin{pmatrix} 0.4 \\ 0.6 \end{pmatrix} \circ \begin{pmatrix} 1.0 \\ 1.0 \end{pmatrix} = \frac{1}{1.0} \begin{pmatrix} 0.4 \\ 0.6 \end{pmatrix} = \begin{pmatrix} 0.4 \\ 0.6 \end{pmatrix}$$

where Z_b is the partition function associated to belief b and is computed as: $Z_b = 0.4 + 0.6 = 1.0$.

d) Is the belief computed in d) a true marginal probability? Argue your answers [0.2 points]

Solution:
Yes, since belief propagation is exact when applied to trees and chains as it is the case.

Problem 10 — *Oriol Ramos Terrades, 0.5 Points*

Say whether the next statements are true (**T**) or false (**F**) [Correct: +0.1, Incorrect: -0.1, unanswered: 0 points].

(a) Belief Propagation (BP) infers exact marginals in directed graphs.

(b) An initial distribution is said to be n*non stationary* if, in the Markov chain specified by this initial distribution, the conditional distribution of x_{n+1} given x_n depend on n.

(c) The Importance sampling method generates biased estimators.

(d) Be $u \sim U(0,1)$, a random value uniformly drawn from the interval $[0,1]$, in the Metropolis-Hasting algorithm, we accept a new sample y if $u \leq \max\left\{1, \frac{h(y)q(y,x)}{h(x)q(x,y)}\right\}$.

(e) The complexity of the message passing algorithm on a chain model is $O(N^2K)$, where N is the number of identically distributed random variables, X_i , and K is the number of states of X_i.

Solution:

(a) True.

(b) True.

(c) False.

(d) False.

(e) False.