1 Exercises

Task 1.1

1. A clique in a graphical model is a subset obtained from the total set of nodes of the graph which represents the graphical model in which all nodes are connected between them. Therefore, a clique is a complete graph.

2. In the first graph $\mathcal{G}_1$, two cliques can be observed: the one conformed by nodes A and C, and the one conformed by B and **C**.

   In the second graph $\mathcal{G}_2$, a total of seven cliques are observed: two conformed by three nodes: the one built by the nodes A, B and D, and the one conformed by nodes B, C and D. Five conformed by two nodes: A and B, A and D, B and C, B and D, and C and D.

   The cliques can be observed in the following pictures:

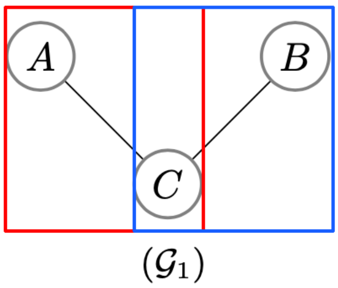

Figure 1: Cliques in $\mathcal{G}_1$.

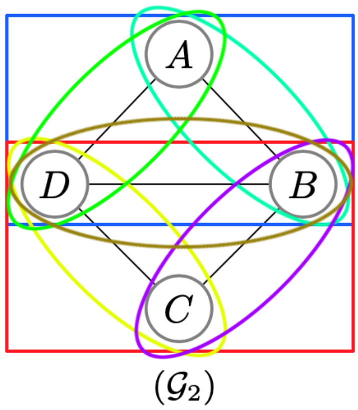

Figure 2: Cliques in $\mathcal{G}_2$.

Task 1.2

1. According to the structure of $\mathcal{G}_3$, the following nodes are independent, given x_5 and x_4:

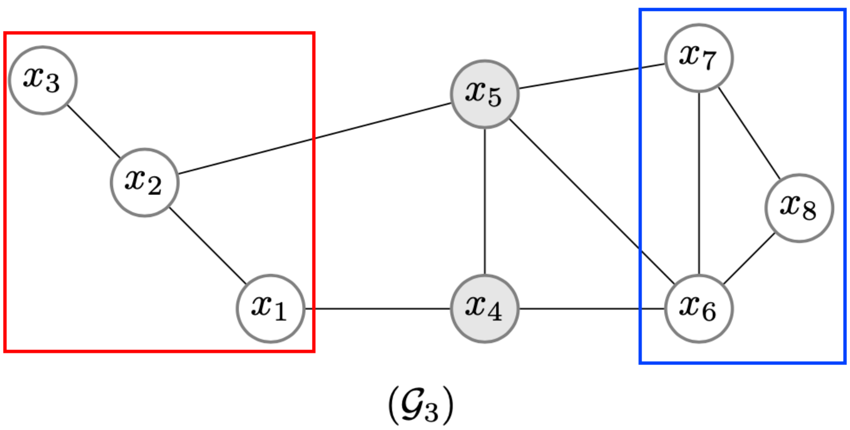

Figure 3: Independent nodes in $\mathcal{G}_3$.

The left nodes (surrounded in red) are mutually dependent and independent of the right nodes (surrounded in blue), which are also mutually dependent.

2. For x_2, it is necessary to know x_1, x_3 and x_5 to be totally independent of the rest of the graph. x_6 is totally independent of the rest of the graph given x_4, x_5, x_7 and x_8.

Task 1.3

1. Given nodes B and D, A and C are independent, because all paths between them are blocked. Nevertheless, given A and C, because of d-separation rules, B and D will be dependent given C.

2. This model, according to d-separation rules, makes A and B independent not given C, and dependent when given. Nevertheless, in undirected graphs, when a node is given, the path between two nodes is blocked, and if it is not, then they are dependent. This is why A and C cannot be independent in an undirected graph if C is not given.

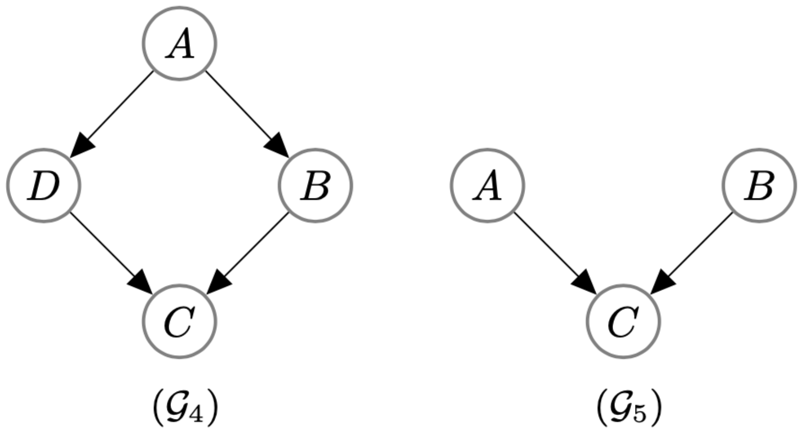

Figure 4: $\mathcal{G}_4$ and $\mathcal{G}_5$.

2 Exercises

Task 2.1

1. There are two types of cliques, those input-input, joining two x nodes, and those input-output, those joining x and y nodes. There is no output-output cliques in here.

2. x_5 will be independent of the rest of the graph given x_2, x_4, x_6, x_8 and y_6.

3. This expression is given by:

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \left[\prod_{i=i}^{4} \phi(y_i, x_i) \right] \phi(x_1, x_2)\, \phi(x_2, x_3)\, \phi(x_3, x_4)\, \phi(x_4, x_1)$$

$$\text{Where } Z = \sum_{x_1,\ldots,x_n} \prod_{c \in \mathcal{C}} \phi_c(x_c)$$

Task 2.2

1. MAP inference allows solving two major issues: non-parallelization and memory consumption. This allows to perform taking advantage of parallel architectures and outperform other methods in computational cost [1]. Moreover, the fact that only the maximum probability is stored allows to save memory since not all outcomes need to be stored.

2. The joint probability derived before can be expressed as:

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z}\, U(x_i, y_i)\, V(x_i, x_j)$$

This, according to what was explained before, means the following:

$$U(x_i, y_i) = \prod_{i=i}^{4} \phi(y_i, x_i)$$
$$V(x_i, x_j) = \phi(x_1, x_2)\, \phi(x_2, x_3)\, \phi(x_3, x_4)\, \phi(x_4, x_1)$$

Therefore, $U(x_i, y_i)$ represents the joint probability between the correspondent pairs of hidden states and pixels (x, y) and $V(x_i, x_j)$ between those hidden states that are related.

3 Exercises

Task 3.1

1. Implemented in matlab code.

2. The following results were obtained from the code:

Figure 5: Noisy figure to improve.

Figure 6: Improved figure.

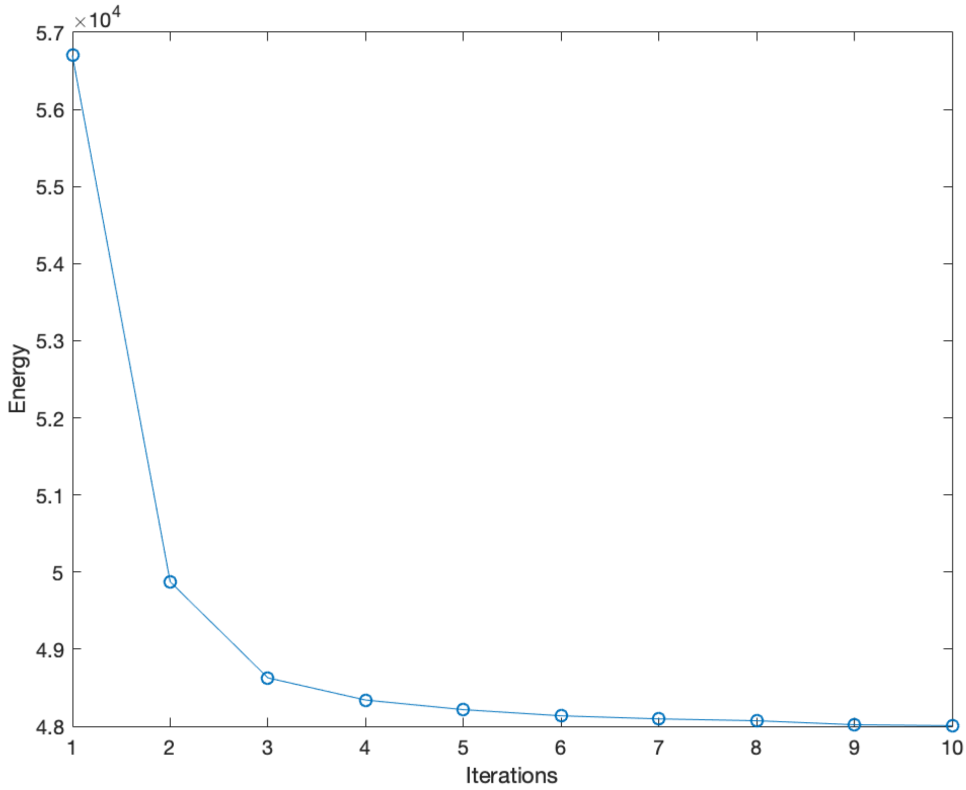

Figure 7: Energy vs iterations plot.

This results were obtained with the value of 1.0 for λ and τ. As it can be seen, the energy reduces per iteration, but from the fifth one, this reduction is much smoother, converging around 4.8x10^4. *tau* sets how closer the outcome will be to the original input, while λ determines how close it will be to the criterion selected (in this case, the maximum probability (Max-product algorithm)).

Task 3.2

After applying the algorithm, the following is obtained:

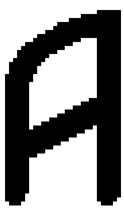

Figure 8: Original image.

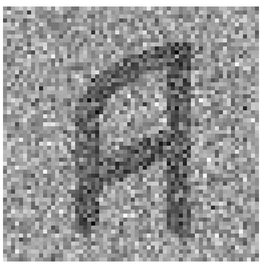

Figure 9: Original image after noise addition.

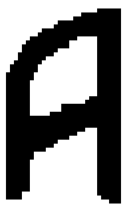

Figure 10: Obtained image after applying the algorithm.

λ sets the importance of the neighbor pixels. A big value of λ makes the image full white since most part of it is defined by white pixels. Lower values make this effect smoother. In this sense, λ acts a regularizer term.

References

[1] S. Alchatzidis, A. Sotiras, and N. Paragios. Efficient parallel message computation for map inference. In *2011 International Conference on Computer Vision*, pages 1379–1386. IEEE, 2011.

[2] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques.* MIT press, 2009.