Optimization for Machine Learning

Exercise sheet 1

Dr. Bogdan Savchynskyy <`bogdan.savchynskyy@iwr.uni-heidelberg.de`>
Stefan Haller <`stefan.haller@iwr.uni-heidelberg.de`>

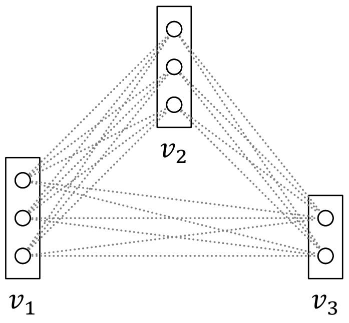
⋆

Exercise 1.1

Formulate the MAP-inference problem for graphical models. How is the optimization objective defined? Which role play the unary and pairwise costs?

Exercise 1.2

To get started, implement the computation of the energy for arbitrary graphical models. Evaluate the energy function for the given graphical model and parameter settings. What role does the parameter λ play for the "Potts" pairwise potential?



$$\theta_1 = [0.3, 0.9, 0.4],\quad \theta_2 = [0.8, 0.1, 0.3],\quad \theta_3 = [0.2, 0.5]$$

$$\forall \{u,v\} \in \mathcal{E} : \theta_{uv} = \begin{cases} 0 & \text{if } x_u = x_v, \\ \lambda & \text{otherwise.} \end{cases}$$

	$\lambda = 0$	$\lambda = 0.2$	$\lambda = 0.5$
$x = [0,0,0]$			
$x = [0,2,1]$			

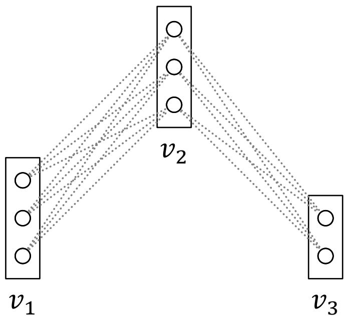
We provide example code that you can use on our website (`example_1_2.py`). Test your implementation with the provided bigger model (`test_1_2.py`).

Exercise 1.3

Implement a function `bruteforce_min(nodes, edges)` that computes $\min_{x \in \mathcal{X}_\mathcal{V}} E(\theta, x)$ programmatically using a brute-force technique. Test your implementation with the model from the previous exercise. Additionally, you should check that it works with more the complex model files that we provide on the website of the course. Derive the time complexity of the algorithm.

Exercise 1.4

For this exercise assume that your graph is chain-structured. As an example you can remove the edge $e = (v_1, v_3)$ from the graph provided in Exercise 1.2 and obtain the following chain-structured graph:



$$\theta_1 = [0.3, 0.9, 0.4],$$
$$\theta_2 = [0.8, 0.1, 0.3],$$
$$\theta_3 = [0.2, 0.5]$$

$$\forall \{u, v\} \in \mathcal{E} : \theta_{uv} = \begin{cases} 0 & \text{if } x_u = x_v, \\ \lambda & \text{otherwise.} \end{cases}$$

Implement the dynamic programming inference technique discussed in the lecture to compute all forward messages. Additionally, implement the backtracking pass to infer the optimal label assignment. What is the time complexity of this algorithm?

Exercise 1.5

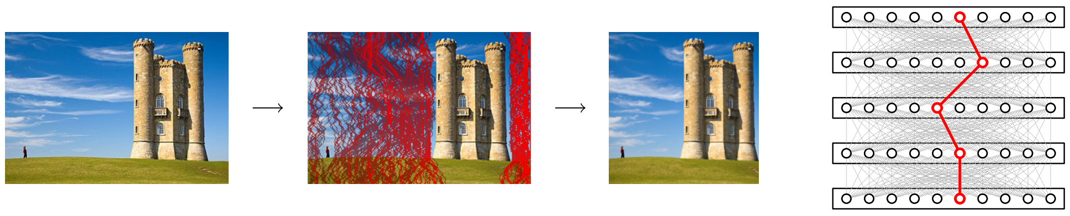
Extend your code of Exercise 1.4 to allow computation of min-marginals.

Exercise 1.6

Improve your code of Exercise 1.4 by extending it to handle tree-shaped graphs.

Exercise 1.7

Seam carving[1] is a technique for content-aware image rescaling. Implement this method by modeling the problem as a graphical model.



$$\theta_i(j) = |\Delta_{\text{hor}}(i,j)| + |\Delta_{\text{ver}}(i,j)| \qquad \theta_{i,i+1}(j,j') = \|j - j'\|^2$$
$$\Delta_{\text{hor}}(i,j) = p(i,j-1) - p(i,j+1) \qquad \Delta_{\text{ver}}(i,j) = p(i-1,j) - p(i+1,j)$$

You can solve the resulting chain-structured graphical model to remove a single pixel from each row efficiently by using your dynamic programming approach implemented in Exercise 1.4. Resize the provided image `tower.jpg` with this technique by iteratively removing pixels until the image width equals 100 pixels.

[1]Avidan, Shai, and Ariel Shamir. "Seam carving for content-aware image resizing." ACM Transactions on graphics (TOG). Vol. 26. No. 3. ACM, 2007. `https://dl.acm.org/citation.cfm?id=1276390`
Short overview: `https://en.wikipedia.org/wiki/Seam_carving`