Mini Project 2: More fire

Instructions

- The assignment is due by March 4, 2026 at midnight.
- There is a Google Colab with helper code located at `https://colab.research.google.com/drive/1ISpalCCDw-E8aF-NdA1bbWqGKeKD7cI-#scrollTo=WhfyTJ_r3UHz`. Open the notebook, go to "File" in the top left, and **save a copy**.
- Please submit to Gradescope (accessible via Canvas). There will be **two** assignments on Gradescope which you will need to submit to: one for your written answers, and one for code upload.
- Your written answers should be in a single PDF file, ideally produced with LaTeX. Your code should also be in a single PDF. To save the Colab to PDF, go to "File" in the top left, click "Print" and then save as PDF.
- The goal here is learning. You can work with other students, but be sure that, in the end, the solutions you submit were your work and that you understand them completely.
- You may use LLMs to help with this assignment; note, though, that we've tried these questions in several LLMs and they often produce incorrect answers, so the correctness is up to you!

1 Bigger fires

We are going to continue the theme from mini-project 1, in terms of a grid of cells with fire in them, but now model the relationship between states of neighboring cells probabilistically, instead of with hard constraints.

We can model the joint distribution of the state of all the locations in our (rectangular grid map) using a Markov random field (MRF). For simplicity, we'll ignore smoke and assume each location is either on fire or not.

We will assume strong locality in our MRF, using a pairwise potential between every neighboring pair of locations to model how fire in one location affects the probability of fire in neighboring locations.

Each pair of neighboring cells in this model shares the same binary potential $\phi(X_i, X_j)$. Each node also has a unitary potential $\psi_i(X_i)$ that may vary for each cell. Overall, the model expresses the joint distribution:

$$P(X) \propto \prod_{X_i} \psi_i(X_i) \prod_{X_j \in \texttt{neighbors\_rb}(X_i)} \phi(X_i, X_j),$$

where `neighbors_rb`(X_i) is the set of two neighbors to the right and below X_i. In essence, `neighbors_rb` ensures that we include a potential along each edge in the grid only once.

1.1 Exact Marginalization

1. (14 points) Write a python function that takes in a grid of observations, like

```
grid = [
    ['F', 'F', 'U', 'U'],
    ['U', 'F', 'U', 'U'],
    ['F', 'U', 'C', 'U'],
    ['U', 'U', 'C', 'U']
]
```

a pairwise potential, and a prior belief of fire at each cell and computes the marginal probabilities of fire at each of the unobserved locations. Use the factor-multiplication and marginalization methods you implemented in HW 2 Question 4.

Assume that your observations are perfect. If the pairwise potential is $[[0.7, 0.3], [0.3, 0.7]]$ and we use a unary potential of $[0.6, 0.4]$ to indicate a prior belief that locations are more likely to be clear than on fire, what values do you get on the problem above?

Note: Since the pairwise potential is symmetric, which way around we express the prior ($[0.6, 0.4]$ or $[0.4, 0.6]$) does not really matter. What matters is that we are internally consistent throughout the rest of the implementation. We will think of "Clear" as acting as a sort of "0" label and "Fire" as being a "1" label and thus choose to index the potential prior as we have in the problem.

2. (6 points) When you multiply all the potentials together, but before you normalize, what is the smallest non-zero potential in the table? Roughly, what will it be for a 10 x 10 grid? Explain what kind of trouble this might cause.

3. (10 points) Consider the marginal distributions on a 2×2 and 3×3 grid, when there are no observations. Explain, intuitively, why they have the values they do. How do they relate to the unary potential values? What, roughly, would you expect to happen on a 10 x 10 grid with these same potentials and no observations?

4. (8 points) How could you model imperfect observations? What might be a sensible potential to use if $P(\text{observe fire} \mid \text{fire}) = 0.8$ and $P(\text{observe fire} \mid \text{no fire}) = 0.05$?

1.2 Loopy belief propagation

Our exact marginalization is very slow for larger grids. So instead, we will use loopy belief propagation to compute the marginals.

We have provided you with a function `fire_mrf_lbp_marginals` that implements loopy belief propagation to compute the marginals approximately. This function is specialized to our 2D grid model, using NumPy operations to avoid Python loops. The function works in the most general form for our problem of one pairwise potential and potentially different unary potentials for each cell.

5. (12 points) Please complete the implementation of `fire_mrf_lbp_conditionals`. This function should take in a grid of observations, a unary potential that is related to prior belief of fire at each cell, as well as a length two vector `sensor_metrics` which gives $P(\text{observe fire} \mid \text{fire})$ and $P(\text{observe fire} \mid \text{no fire})$ to allow for potentially noisy observations. It should return the marginal probability of fire at each cell in the grid, conditioned on having the observations that we recieved, as well as knowing the prior and sensor metrics.

Using the same pairwise potential $[[0.7, 0.3], [0.3, 0.7]]$, the same prior unary potential at each cell of $[0.6, 0.4]$, and observation potentials from Question 4, run it on this example:

```
grid = [
    ['U', 'F', 'U', 'C', 'U', 'U', 'U'],
    ['U', 'F', 'F', 'U', 'C', 'U', 'C'],
    ['U', 'F', 'F', 'U', 'U', 'C', 'U'],
    ['U', 'F', 'U', 'F', 'U', 'U', 'U'],
    ['U', 'U', 'F', 'U', 'U', 'U', 'U'],
```

```
    ['F', 'F', 'U', 'U', 'F', 'U', 'U'],
    ['U', 'C', 'C', 'U', 'U', 'U', 'F']
]
```

Using `tol=1e-6`, does it converge within 500 iterations? If so, on what iteration?

6. (8 points) Compare the accuracy of this approximate method to the exact method on our original 4 x 4 example. The mean squared error of the marginal probability values is a reasonable error measure.

7. (6 points) Compare the computational speed of the exact method and loopy BP on 3x3, 4x4, and 5x5 grids. What trends do you see?

1.3 Gibbs sampling

An alternative approximate inference method is Gibbs sampling.

8. (2 points) What is the Markov blanket of cell i, j in the grid?

9. (14 points) Complete the python function `fire_mrf_gibbs_conditionals` that uses Gibbs sampling instead of LBP to estimate the marginals.

   How many samples are necessary (with a burn-in of 1000 samples) to achieve the same accuracy, on average, as 100 iterations of LBP on our original 4 x 4 problem?

1.4 Messages

10. (10 points) In this project we made some modeling approximations and some inference approximations. List two of each.

11. (10 points) Describe the probability distribution encoded by a single pairwise potential in this model, or argue that it has no such direct interpretation.