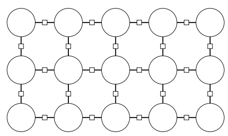
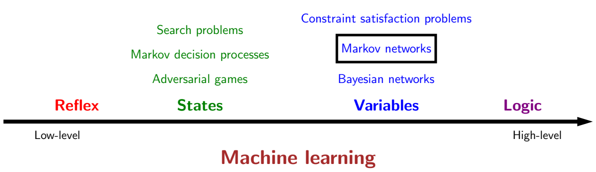
Markov networks: overview



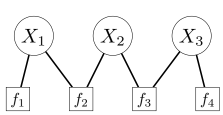
- In this module, I will introduce Markov networks.

Course plan



- So far, we have introduced CSPs, the first of our variable-based models.
- Markov networks are the second type of variable-based model, which will connect factor graphs with **probability** and serve as a stepping stone on the way to Bayesian networks.

Review: factor graphs



Definition: factor graph

Variables:

$X = (X_1, \dots, X_n)$, where $X_i \in \text{Domain}_i$

Factors:

$f_1, \dots, f_m$, with each $f_j(X) \geq 0$

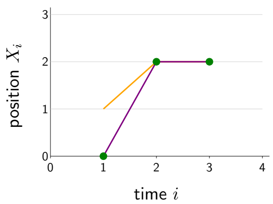
Definition: assignment weight

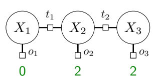
Each **assignment** $x = (x_1, \dots, x_n)$ has a **weight**:

$$\text{Weight}(x) = \prod_{j=1}^{m} f_j(x)$$

- Markov networks, like all variable-based models, are based on factor graphs.
- Recall that a factor graph contains a set of variables whose relationships are determined by a set of factors. For each assignment to all the variables, we have a non-negative weight, which captures how "good" a particular assignment is.
- Aside: Markov networks are also known as Markov random fields. They are typically defined as an undirected graph over variables, where we have a factor for each clique in the graph. But we use factor graphs to make the factors more explicit.

Example: object tracking





x_1	$o_1(x_1)$
0	2
1	1
2	0

x_2	$o_2(x_2)$
0	0
1	1
2	2

x_3	$o_3(x_3)$
0	0
1	1
2	2

$\|x_i - x_{i+1}\|$	$t_i(x_i, x_{i+1})$
0	2
1	1
2	0

[demo]

- Recall the object tracking example in which we observe noisy sensor readings 0, 2, 2.
- We have observation factors o_i that encourage the position X_i and the corresponding sensor reading to be nearby.
- We also have transition factors t_i that encourage the positions X_i and X_{i+1} to be nearby.

Maximum weight assignment

CSP objective: find the maximum weight assignment

$$\max_x \text{Weight}(x)$$

x_1	x_2	x_3	Weight(x)
0	1	1	4
0	1	2	4
1	1	1	4
1	1	2	4
1	2	1	2
1	2	2	8

Maximum weight assignment: $\{x_1 : 1, x_2 : 2, x_3 : 2\}$ (weight 8)

But this doesn't represent all the other possible assignments...

- In constraint satisfaction problems, we are interested in finding the maximum weight assignment.
- For the object tracking example, we show all the assignments with non-zero weight. The maximum weight assignment here is $\{x_1 : 1, x_2 : 2, x_3 : 2\}$ with weight 8.
- However, just returning this one assignment doesn't give us a sense of the alternatives, and how likely they are. In other words, we are not representing our **uncertainty**.

Definition

Definition: Markov network

A Markov network is a factor graph which defines a joint distribution over random variables $X = (X_1, \dots, X_n)$:

$$\mathbb{P}(X = x) = \frac{\text{Weight}(x)}{Z}$$

where $Z = \sum_{x'} \text{Weight}(x')$ is the normalization constant.

x_1	x_2	x_3	Weight(x)	$\mathbb{P}(X = x)$
0	1	1	4	0.15
0	1	2	4	0.15
1	1	1	4	0.15
1	1	2	4	0.15
1	2	1	2	0.08
1	2	2	8	0.31

$$Z = 4 + 4 + 4 + 4 + 2 + 8 = 26$$

Represents uncertainty!

- We now introduce **Markov networks** to capture the uncertainty over assignments.
- We've done most of the hard work by defining factor graphs, which endows each assignment $x = (x_1, \dots, x_n)$ with a weight Weight(x).
- We define the probability of an assignment x to be the fraction of weight relative to all assignments.
- Operationally, we first compute the **normalization constant** (also known as the partition function) Z, which is the sum of the weights over all assignments.
- Then we simply divide each weight by this normalization constant to get the probability.
- So the maximum weight assignment here only has 31% of the total probability.

Marginal probabilities

Example question: where was the object at time step 2 (X_2)?

Definition: Marginal probability

The marginal probability of $X_i = v$ is given by:

$$\mathbb{P}(X_i = v) = \sum_{x : x_i = v} \mathbb{P}(X = x)$$

Object tracking example:

x_1	x_2	x_3	Weight(x)	$\mathbb{P}(X = x)$
0	1	1	4	0.15
0	1	2	4	0.15
1	1	1	4	0.15
1	1	2	4	0.15
1	2	1	2	0.08
1	2	2	8	0.31

$\mathbb{P}(X_2 = 1) = 0.15 + 0.15 + 0.15 + 0.15 = 0.62$

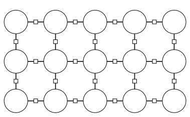
$\mathbb{P}(X_2 = 2) = 0.08 + 0.31 = 0.38$

Note: different than max weight assignment!

- The language of probability allows us to do more than just ask for the probability of complete assignments.
- It allows you to also ask for the **marginal probability** of partial assignments. In particular, we will focus on probability of single variables. This means asking for the probability of one variable X_i while marginalizing out others. Intuitively, while we don't ask for particular values on the marginalized variables, they still have a influence since factors still get multiplied into the weight.
- In the object tracking example, suppose we are interested in where the object was at time step 2 only, not caring about its position at other times.
- Then we would ask for the marginal probabilities $\mathbb{P}(X_2 = 1)$ and $\mathbb{P}(X_2 = 2)$. We compute these quantities by summing the probabilities of the complete assignment that match the condition on X_2.
- Interestingly, the result is that the object is 62% likely to be at position 1, even though the most likely complete assignment says the object is at position 2! Intuitively, this is because there are multiple assignments with $x_2 = 1$ with moderate weight (4), even though they don't have the maximum weight (4). There is kind of a "strength in numbers" phenomenon.
- The lesson is that you might get different answers depending on what you're asking.

Application: Ising model

Ising model: classic model from statistical physics to model ferromagnetism



$X_i \in \{-1, +1\}$: atomic spin of site i

$f_{ij}(x_i, x_j) = \exp(\beta x_i x_j)$ wants same spin

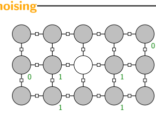
Samples as β increases:

Figure 2 from Perez (1998)

- A canonical example of a Markov network is the Ising model from statistical physics, which was developed by physicists in the 1920s to model ferromagnetism.
- The idea is that you have a large set of sites, each of which can either have an up or down spin.
- Assignments in which adjacent sites tend to have the same spin (resulting in a lower energy configuration) are favored, where the strength is given by β.
- Ising models are used to study phase transitions in physical systems. If $\beta = 0$, then the factors all evaluate to 1 independent of the assignment. Therefore, all assignments are equally likely, and there is simply no structure; every variable is completely random (probability $\frac{1}{2}$ up and probability $\frac{1}{2}$ down). As β increases, there starts to be more cohesion between sites, leading to larger blobs. As $\beta \to \infty$, equality becomes more like a hard constraint.
- Here we are showing samples from the Ising model (how we do this we will talk about in a future module).

Application: image denoising

Example: image denoising



- $X_i \in \{0, 1\}$ is pixel value in location i
- Subset of pixels are observed

$o_i(x_i) = [x_i = \text{observed value at } i]$

- Neighboring pixels more likely to be same than different

$t_{ij}(x_i, x_j) = [x_i = x_j] + 1$

- As another example, consider the problem of image denoising. This is one of the classic applications of Markov networks in computer vision before deep learning.
- In our stylized example, suppose we have a noisy image where only some of the pixels are observed and our goal is to recover our best guess of the clean image.
- We define a variable X_i for each pixel $i \in \{(1,1), (1,2), (1,3), \dots\}$.
- We then define an observation factor o_i on each pixel that is observed that constrains that pixel to be the observed value. For example, $o_{(1,1)}(x_i) = [x_i = 1]$.
- Then for every pair of neighboring pixels i and j (e.g., $i = (1,1)$ and $j = (2,1)$), we define a transition factor $t_{ij}(x_i, x_j)$ that encourages the pixel values to agree (both be 0 or both be 1). Weight 2 is given to those pairs which are the same and 1 if the pair is different.
- Note that the observation and transition factors should be reiminiscent of the object tracking example, just in two dimensions. In general, having factors that incorporate external evidence (observations) and factors that incorporate internal consistency (transitions) is a common template for building Markov networks, and variable-based models more generally.

Summary

Markov networks = factor graphs + probability

- Normalize weights to get probablity distribution
- Can compute marginal probabilities to focus on variables

CSPs	Markov networks
variables	random variables
weights	probabilities
maximum weight assignment	marginal probabilities

- In summary, we have introduced Markov networks, which connect factor graphs with probability.
- The connection is very natural: factor graphs already provide a way of specifying non-negative weights over assignments, which gets us most of the way there. We then normalize the weights to make them sum to 1 to get a probability distribution.
- Once we have a joint probability distribution, we can compute marginal probabilities of individual (or subsets of) variables.
- We can compare CSPs with Markov networks. Variables become random variables, which means that they have probabilities associated with them. Instead of weights, we have their normalized versions, a.k.a., probabilities. The big difference is that instead of focusing on just finding the maximum weight assignment, which might be not representative of the full set of possibilities, the goal is to look at marginal probabilities.