6.867 Machine learning and neural networks

Tommi Jaakkola
MIT AI Lab
tommi@ai.mit.edu

Lecture 22: markov random fields

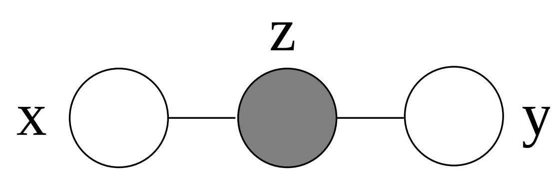
Topics

- Markov random fields
  - review: semantics, quantification
  - pattern completion example

Review: Markov random fields

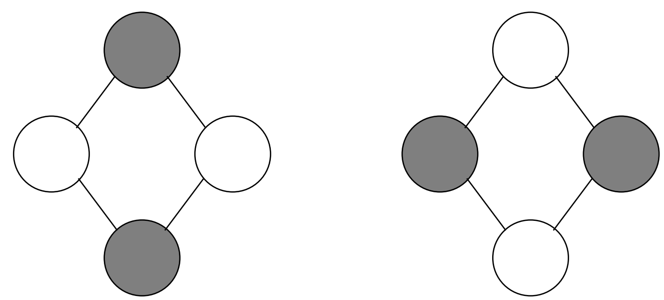
- The (conditional) independence properties can read from the graph via simple graph separation:

  x and y are conditionally independent given z if all paths between x and y go through z



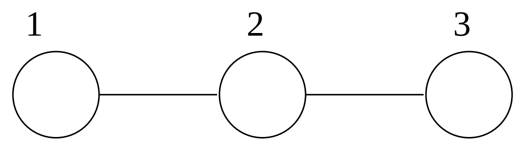
x and y are conditionally independent given z

- This graph semantics captures the simple example



Review: Markov random fields

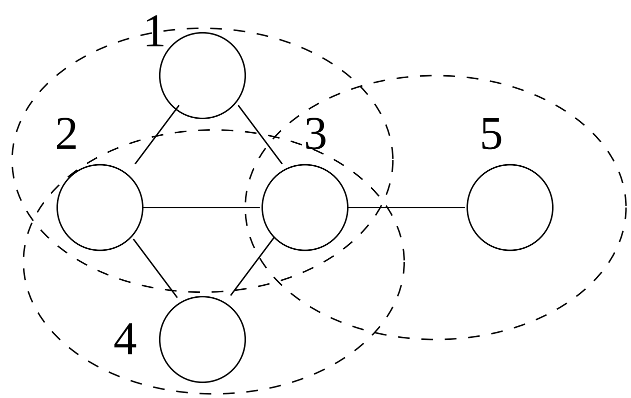
- A Markov chain:



$$P(x_1, x_2, x_3) = \underbrace{\frac{1}{Z}} \quad \underbrace{\psi_{12}(x_1, x_2)} \quad \underbrace{\psi_{23}(x_2, x_3)}$$

$$\begin{array}{ccc} 1 & P(x_1, x_2) & P(x_3 | x_2) \\ 1 & P(x_1 | x_2) & P(x_2, x_3) \end{array}$$

- More generally:

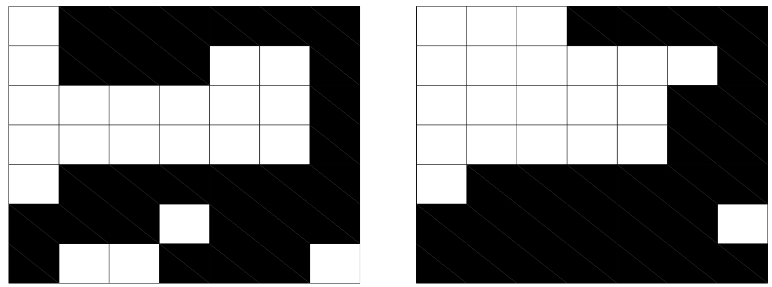


$$P(\mathbf{x}) = \frac{1}{Z} \psi_{123}(x_1, x_2, x_3) \, \psi_{234}(x_2, x_3, x_4) \, \psi_{35}(x_3, x_5)$$

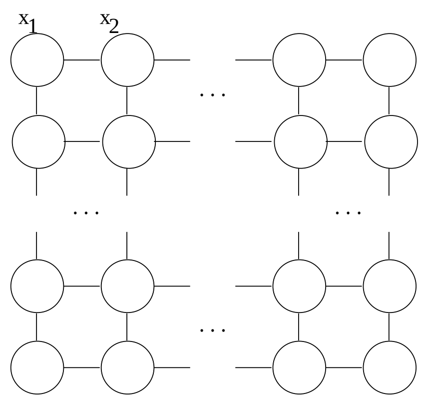
Image reconstruction example

- Modeling images with *Boltzmann machines*

- nearby pixels in images should be correlated

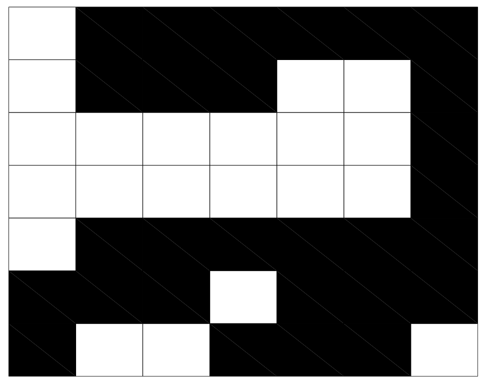


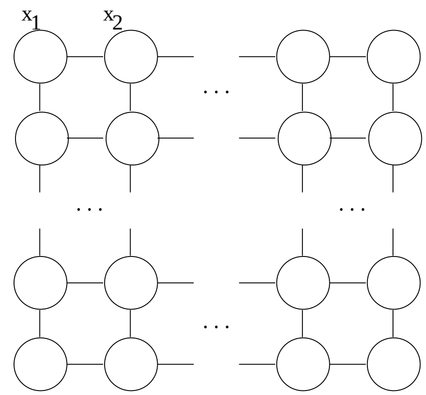
- we can capture such *nearest neighbor* dependences with the following lattice model



Example cont'd

- Each variable x_i in the model indicates whether a pixel is on ($x_i = 1$) or off ($x_i = 0$)





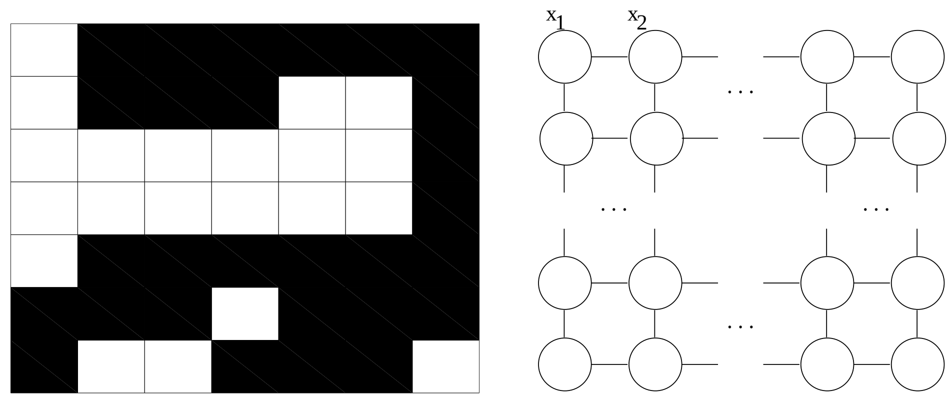
- For any pair of pixels linked with an edge in the graph we have the following potential function

$$\psi_{ij}(x_i, x_j) = \exp(\, J_{ij}\, x_i\, x_j + h_i\, x_i + h_j\, x_j\,)$$

where J_{ij} defines the *connection strength* between pixels i and j (large values of J_{ij} imply highly correlated pixels)

h_i permit us to bias individual pixel values towards on or off

Example cont'd



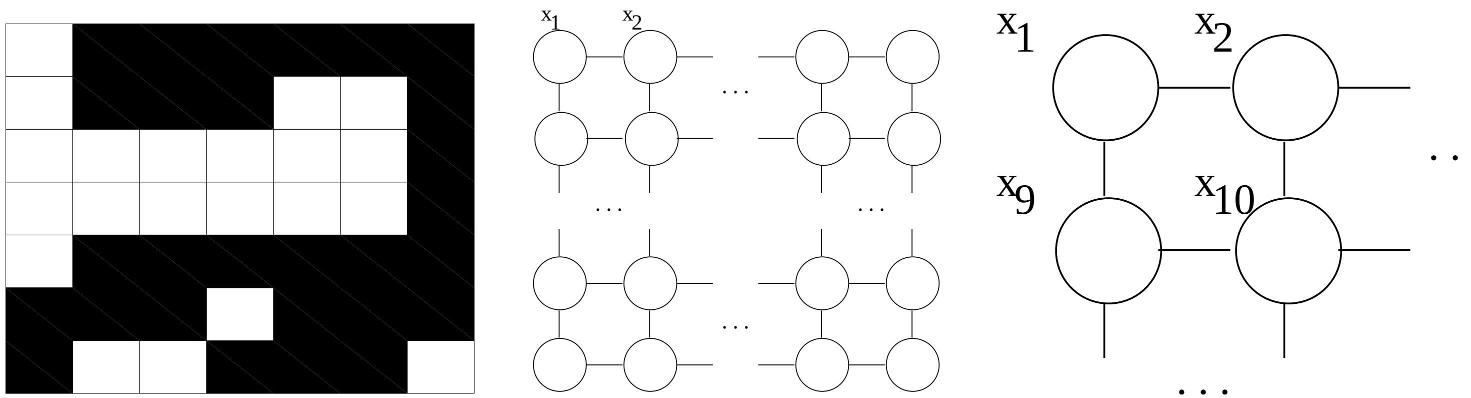
- The full joint distribution over the pixels in the image is given by

$$P(x_1, \ldots, x_{64} | J, h) = \frac{1}{Z(J,h)} \prod_{\text{edges}\, ij} \exp(\, J_{ij}\, x_i\, x_j + h_i\, x_i + h_j\, x_j\,)$$

$$= \frac{1}{Z(J,h)} \exp\left(\sum_{\text{edges}\, ij} J_{ij}\, x_i\, x_j + \sum_i \tilde{h}_i\, x_i \right)$$

(here, e.g., $\tilde{h}_i = 4h_i$ for pixels away from the borders)

Example cont'd



- The graph structure implies a number *Markov properties* such as

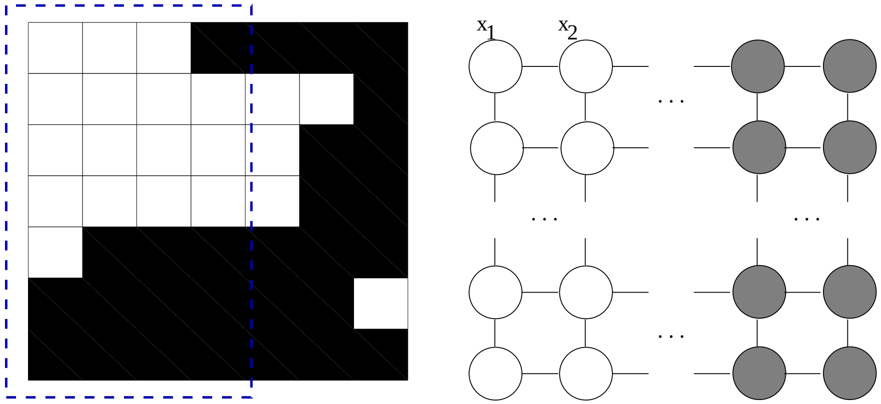
$$P(x_1|x_2, \ldots, x_{64}, J, h) = P(x_1|x_2, x_9, J, h)$$

- The specific potential functions we have used also turn these conditional probabilities into logistic regression models (verify)

$$P(x_1 = 1|x_2, x_9, J, h) = g(J_{12}x_2 + J_{19}x_9 + \tilde{h}_1)$$

where $g(z) = (1 + e^{-z})^{-1}$ is the logistic function.

The reconstruction problem

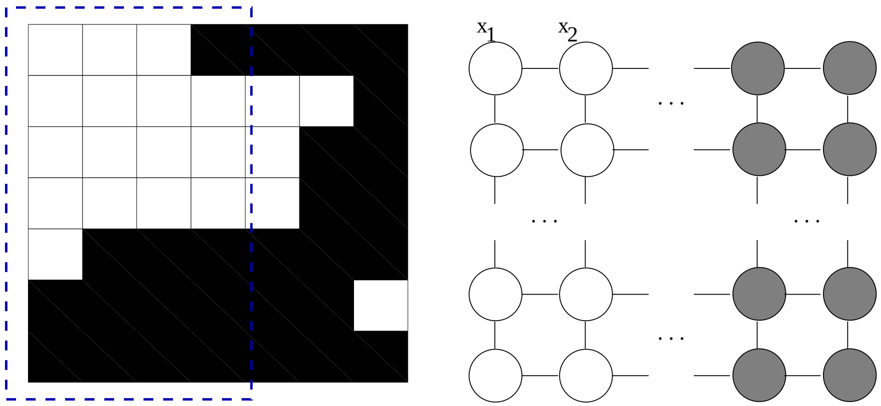


- Part of the image may be corrupted or missing (pixel values are unknown)
- We can use the probability model to infer what the missing pixels are (complete the image)

  Inference problem: evaluate the marginal posterior probabilities for the missing pixels

  Learning problem: adjust the connection strengths for better reconstruction

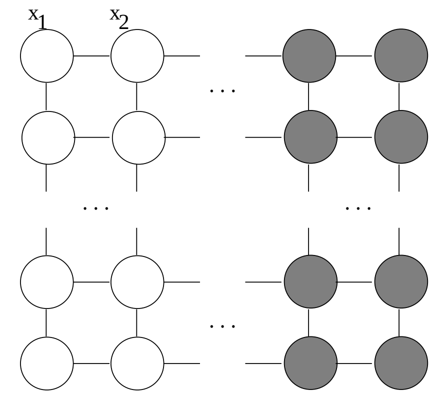
Second inference problem



- Assuming half the image is missing, we have 2^{32} possible configurations of the missing pixels
- We could apply the sampling method we already know about (importance sampling + likelihood weighting) to evaluate the posterior probabilies as in the context of medical diagnosis
- For Boltzmann machines there's a more natural sampling approach: Gibbs' sampling

Gibbs' sampling

- Given a set of known pixel values $x^* = \{x_i^*\}$, we wish to evaluate the posterior probabilities $P(x_i|\mathbf{x}^*, J, h)$ for all the missing pixels



- We can do this via Gibbs' sampling:
  1. Initialize all the missing pixels (e.g., set $x_i = 0$)
  2. Sequentially sample a new value for each missing pixel x_i based on the current setting of its *neighbors* (nb)

$$x_i \sim P(x_i|x_{nb_i}, J, h)$$

     In our case, these conditional probabilities are logistic regression models

$$P(x_i = 1|x_{nb_i}, J, h) = g\left(\sum_{j \in nb_i} J_{ij}x_j + \tilde{h}_i\right)$$

Gibbs' sampling cont'd

- Gibbs' sampling method generates a sequence of new pixel values

$$x_i^1, x_i^2, \ldots, x_i^t, x_i^{t+1}, \ldots$$

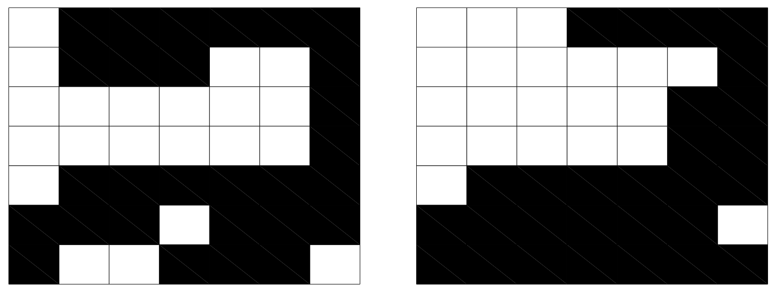
  where x_i^t is the sampled value for pixel i at the t^{th} iteration of the sampler (we assume that one iteration means generating a new value for each missing pixel)

- The posterior mean (expected value) of pixel i can be evaluated approximately from these samples

$$E\{\, x_i \,|\, x^*, J, h\} \approx \frac{1}{T} \sum_{t=1}^{T} x_i^t$$

Learning problem

- How do we set the connection strengths J_{ij} in the Boltzmann machine to improve reconstruction?
- Given a training set of images $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ we can set J_{ij} via maximum likelihood estimation



The connection strengths can be updated via stochastic gradient ascent

$$J_{ij} \leftarrow J_{ij} + \epsilon \frac{\partial}{\partial J_{ij}} \log P(\mathbf{x}_t | J, h)$$

so long as we can evaluate the gradients for any particular training image $\mathbf{x}_t$ (vector of known pixel values)

Learning problem cont'd

$$
\begin{aligned}
\frac{\partial}{\partial J_{ij}} \log P(\mathbf{x}_t | J, h) &= \frac{\partial}{\partial J_{ij}} \left[\sum_{\text{edges } ij} J_{ij} x_i^t x_j^t + \sum_i \tilde{h}_i \, x_i^t - \log Z(J, h) \right] \\
&= x_i^t x_j^t - \frac{\partial}{\partial J_{ij}} \log Z(J, h) \\
&\cdots \\
&= x_i^t x_j^t - E\{\, x_i x_j \,|\, J, h \,\}
\end{aligned}
$$

where $E\{\, x_i x_j \,|\, J, h \,\}$ is the expected co-occurence of the pixel values based on our current model (setting of J and h).

- We can evaluate $E\{\, x_i x_j \,|\, J, h \,\}$ via Gibbs' sampling