Image Denoising with BP

Alireza Mousavi

University of Toronto

Probabilistic Model

- An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
- The set of nodes $\mathcal{V}$
- The set of edges $\mathcal{E}$
- The set of neighbors of a node $i \in \mathcal{V}$, $N(i) = \{j : (i, j) \in \mathcal{E}\}$

$$p(x_1, ..., x_n) \propto \prod_{i \in \mathcal{V}} \psi_i(x_i) \prod_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j)$$

Message Passing

- When x_i is discrete with K possible values, $m_{j\to i}$ is a vector with K values
- If x_j is unobserved

$$m_{j\to i}(x_i) = \sum_{x_j} \psi_j(x_j)\psi_{ij}(x_i, x_j) \prod_{k\in N(j)\setminus\{i\}} m_{k\to j}(x_j)$$

- If x_j is observed with value $\bar{x}_j$

$$m_{j\to i}(x_i) = \psi_j(\bar{x}_j)\psi_{ij}(x_i, \bar{x}_j) \prod_{k\in N(j)\setminus\{i\}} m_{k\to j}(\bar{x}_j)$$

BP on Trees

1. Choose an arbitrary root
2. Pass messages from leaves to root
3. Pass messages from root to leaves
4. For every node x_i, we have

$$b(x_i) \propto \psi_i(x_i) \prod_{j \in N(i)} m_{j \to i}(x_i)$$

Loopy BP

1. Initialize all messages with $m_{j\to i}(x_i) = \frac{1}{k}$.
2. For some number of iterations, keep going through each edge and do BP updates

$$m_{j\to i}(x_i) = \sum_{x_j} \psi_j(x_j)\psi_{ij}(x_i, x_j) \prod_{j\in N(i)\setminus\{i\}} m_{k\to j}(x_j).$$

3. It will generally not converge, but that's ok.
4. Compute beliefs

$$b(x_i) \propto \psi_i(x_i) \prod_{j\in N(i)} m_{j\to i}(x_i).$$

Image Denoising

- A binary image is a $\sqrt{n} \times \sqrt{n}$ matrix where each entry is $+1$ or -1.
- We vectorize this matrix and denote the image as $x \in \mathbb{R}^n$.
- For example, the Mona Lisa below is a 128×128 image, vectorized to be $x \in \mathbb{R}^{16384}$.

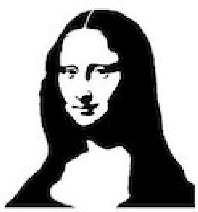

Image Denoising

- Assume that the image has been sent through a noisly channel, where each pixel is flipped with a small probability ϵ.
- The true pixels x are unobserved, and we observe the noisy pixels y
- We have the following Ising model (MRF)

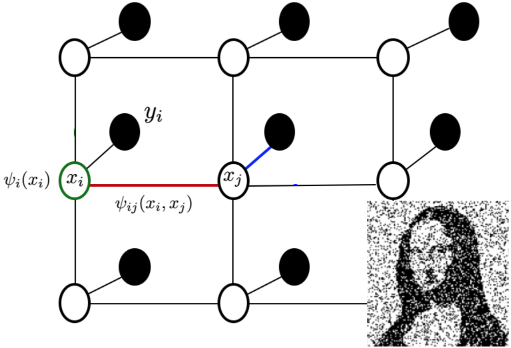

Image Denoising

$$
\begin{aligned}
\mathbb{P}(y_s|x_s) =& (1-\epsilon)^{\frac{1+y_s x_s}{2}} \epsilon^{\frac{1-y_s x_s}{2}} \qquad \text{for all } s. \\
=& \exp\{\frac{1+y_s x_s}{2}\log(1-\epsilon) + \frac{1-y_s x_s}{2}\log(\epsilon)\} \\
\propto& \exp\{y_s x_s \frac{1}{2}\log(\frac{1-\epsilon}{\epsilon})\} \\
=& \exp\{y_s x_s \lambda\} \quad \text{where} \quad \lambda = \frac{1}{2}\log(\frac{1-\epsilon}{\epsilon}).
\end{aligned}
$$

Ising Model

- The goal of the image denoising is to estimate x that maximizes $p(x|y) \propto p(x, y)$.
- We have the Ising model for the prior probability of x, i.e. $p(x) \propto \prod_{s \sim t} \psi_{s,t}(x_s, x_t)$, where $s \sim t$ means $(s, t) \in \mathcal{E}$, and

$$\psi_{s,t}(x_s, x_t) = \begin{pmatrix} e^J & e^{-J} \\ e^{-J} & e^J \end{pmatrix}.$$

Ising Model

- Therefore, we have

$$\begin{aligned} p(x|y) \propto & p(y, x) \\ = & p(x) \prod_s p(y_s|x_s) \\ \propto & \exp\{J \sum_{s \sim t} x_s x_t + \beta \sum_s y_s x_s\} \\ = & \prod_{s \sim t} \psi_{st}(x_s, x_t) \prod_s \psi_s(x_s) \end{aligned}$$

- Now, it is clear that node potentials are given by $\psi_s(x_s) = \exp(\beta y_s x_s)$

Loopy BP for Image Denoising

- Each message $m_{j \to i}(x_i)$ is stored as a 2-dimensional vector, where its first and second coordinates are $m_{j \to i}(+1)$ and $m_{j \to i}(-1)$ respectively.
- Similarly, beliefs $b(x_i)$ are two dimensional vectors, with $b(+1)$ and $b(-1)$ being the first and second coordinates.

Loopy BP for Image Denoising

- Initialize all messages uniformly $m_{j \to i}(x_i) = \frac{1}{2}$
- Keep doing BP updates until it (nearly) converges:

$$m_{j \to i}(x_i) = \sum_{x_j \in \{-1,+1\}} \psi_j(x_j) \psi_{ij}(x_i, x_j) \prod_{k \in N(j) \setminus \{i\}} m_{k \to j}(x_j)$$

  and normalize for stability $m_{j \to i}(x_i) = m_{j \to i}(x_i) / \sum_{x_i} m_{j \to i}(x_i)$.
- Compute beliefs after message passing is done:

$$b(x_i) \propto \psi_i(x_i) \prod_{j \in N(i)} m_{j \to i}(x_i).$$

Loopy BP for Image Denoising

- While loopy BP may not converge, 10-20 iterations suffice to perform approximate inference on the posterior.
- The computed beliefs correspond to $p(x_i|y)$.
- One decision rule to estimate x_i is

$$\hat{x}_i = \operatorname{argmax}_{x_i} b(x_i).$$

- The result is remarkable