Lecture Pattern Analysis

Part 23: Markov Random Fields

Christian Riess

IT Security Infrastructures Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg

July 19, 2022

Introduction

- A Markov Random Field (MRF) is an undirected graphical model[1]
- Main applications are related to labeling tasks, e.g., stereo matching, multi-view stitching, segmentation, denoising, inpainting[2]
- The idea is to consider the unknown labels as a field of random variables with conditional independence

Fig. 2. Images used for our benchmarks. (a) Stereo matching: Tsukuba, Venus, and Teddy left images and true disparities. (b) Photomontage 1: Panorama. (c) Photomontage 2: Family group shot. (d) Binary image segmentation: Flower, Sponge, and Person. (e) Denoising and inpainting: Penguin and House.

[1] The literature source for this lecture is Bishop Sec. 8.3

[2] Figure from: Szeliski et al.: "A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 6, June 2008, pp. 1068–1080.

MRF Joint Distribution

- We again start with the joint distribution

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{z}_1, \ldots, \mathbf{z}_N) = p(\mathbf{x}_1, \ldots, \mathbf{x}_N | \mathbf{z}_1, \ldots, \mathbf{z}_N) \cdot p(\mathbf{z}_1, \ldots, \mathbf{z}_N) \quad (1)$$

  with observation sequence/field $\mathbf{x}_i$ and hidden variables $\mathbf{z}_i$
- Two assumptions help to further factorize this expression:
  1. Each observation only depends on a single hidden variable, i.e.,

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N | \mathbf{z}_1, \ldots, \mathbf{z}_N) = \prod_{i=1}^{N} p(\mathbf{x}_i | \mathbf{z}_i) \quad (2)$$

  2.a Each hidden variable only depends on its neighbors, i.e., $\mathbf{z}_i \perp\!\!\!\perp \mathbf{z}_j \mid \mathcal{N}(\mathbf{z}_i)$ for all $\mathbf{z}_j \notin \mathcal{N}(\mathbf{z}_i)$ where $\mathcal{N}(\mathbf{z}_i)$ is the set of neighbors of $\mathbf{z}_i$

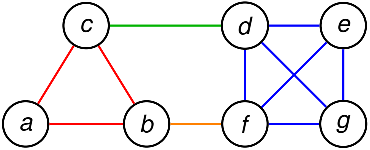
  2.b For tractability, we additionally require $p(\mathbf{z}_1, \ldots, \mathbf{z}_N)$ to split into pairs, i.e.,

$$p(\mathbf{z}_1, \ldots, \mathbf{z}_N) = \prod_{\mathbf{z}_l \in \mathcal{N}(\mathbf{z}_i)} p(\mathbf{z}_i, \mathbf{z}_l) \quad (3)$$

  Don't get confused: some sources use $p(\mathbf{z}_i | \mathbf{z}_l)$ instead of $p(\mathbf{z}_i, \mathbf{z}_l)$

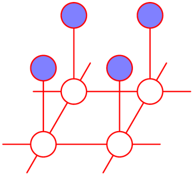
Factorization and Cliques

- Assumption 2.a highlights that the neighborhood determines the factorization
- Undirected graphs factorize into maximal cliques (fully connected subgraphs):



$$p(a, b, c, d, e, f, g) = p(a, b, c)p(c, d)p(b, f)p(d, e, f, g) \tag{4}$$

- Assumption 2.b implies that we model the dependencies with cliques ≤ 2:



Hammersley-Clifford: Potentials instead of Distributions

- The learning task for MRFs is to find distributions $p(\mathbf{x}_i|\mathbf{z}_i)$ and $p(\mathbf{z}_i, \mathbf{z}_j)$
- Unfortunately, **learning is intractable** for interesting problem sizes

- The inference task for MRFs is to find $\mathbf{z}^* = \underset{\mathbf{z}}{\operatorname{argmax}}\, p(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{z}_1, \ldots, \mathbf{z}_N)$
- Fortunately, **inference is tractable** for interesting problem sizes
- Unfortunately, the distributions $p(\mathbf{x}_i|\mathbf{z}_i)$ and $p(\mathbf{z}_i, \mathbf{z}_j)$ are difficult to hand-craft

- But there is a substantial simplification:
  - The Hammersley-Clifford Theorem states that a Markov Random Field and a Gibbs Random Field are equivalent under some mild constraints (Gibbs potentials need to be strictly positive)
  - Hence, instead of specifying $p(\mathbf{x}_i|\mathbf{z}_i)$ and $p(\mathbf{z}_i, \mathbf{z}_\mathbf{j})$, we can set potential functions $\psi(\mathcal{C})$, where $\mathcal{C} = \{\mathbf{z}_i\}$ are maximal cliques of the hidden variables
  - In many applications, such potential functions are much simpler to choose: just set a local maximum for desired behavior (more later)

MRFs as Gibbs Random Fields

- The potential functions $\psi(\mathcal{C})$ over a set of nodes $\mathcal{C} = \{\mathbf{z}_i\}$ are set as exponential functions

$$\psi(\mathcal{C}) = \exp(-E(\mathcal{C})) \ , \tag{5}$$

where $E(\mathcal{C})$ is an energy function (i.e., lower energy is closer to the goal)

- The factorized MRF is hence

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{z}_1, \ldots, \mathbf{z}_N) = \frac{1}{Z} \prod_{\mathcal{C}} \psi(\mathcal{C}) \tag{6}$$

where $Z = \sum_{\mathbf{z}} \prod_{\mathcal{C}} \psi(\mathcal{C})$ is the **partition function** that normalizes the distribution by summing over all combinations of variable assignments

- The normalization Z enables us to use any positive function for the potentials
- Calculating Z is intractable (sum over all value combinations of $\mathbf{z}$), but Z is constant when maximizing $p(\mathbf{z})$, hence it does not prevent inference

MRF Inference and Gibbs Sampling

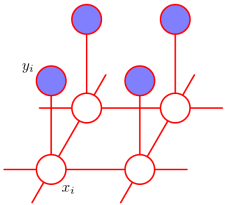
- MRFs are oftentimes used for labeling tasks
  - Each random variable is assigned one label from a **discrete** set of labels
  - We will look at Bishop's example of denoising a binary image:
    each pixel has a hidden variable, MRF inference decides for black or white
  - Another example is depth estimation from stereo vision:
    Here, the labels are depth values that are assigned to each pixel/variable
- The optimization task for inference is

$$\mathbf{z}^* = \operatorname*{argmax}_{\mathbf{z}} p(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{z}_1, \ldots, \mathbf{z}_N) = \operatorname*{argmax}_{\mathbf{z}} \frac{1}{Z} \prod_{\mathcal{C}} \psi(\mathcal{C}) \tag{7}$$

- Inference can be done with Gibbs Sampling:
  1. Select a node $\mathbf{z}_i$ (randomly, or following some pattern)
  2. Get the current labels from all neighbors (this forms a Markov blanket!)
  3. Assign a new label to $\mathbf{z}_i$ with minimal energy
  4. Goto 1 for a fixed number of iterations, or until convergence
- Gibbs sampling is only locally optimal
- Graph cuts find a globally optimal solution for binary labels ($\longrightarrow$ next lecture)

Example: Denoising a Binary Image — Graph Structure

Figure 8.31 An undirected graphical model representing a Markov random field for image de-noising, in which x_i is a binary variable denoting the state of pixel i in the unknown noise-free image, and y_i denotes the corresponding value of pixel i in the observed noisy image.



- A typical MRF graph treats the unknown solution as hidden variables
- For denoising, the unknowns are the pixel values of the denoised image
- The hidden variables are connected in a grid (like the pixel grid)
- Each hidden variable has an associated observation, i.e., a noisy input pixel
- Hence, the maximal clique size is 2, and there are 2 types of connections (observations-hidden and hidden-hidden)
- Bishop models the binary pixels as $-1, +1$ values

Example: Denoising a Binary Image — Energy Functions

- Energy functions for observations-hidden connections are chosen as

$$E(x_i, z_i) = -\eta x_i z_i \ , \tag{8}$$

  with weight constant η, i-th input pixel x_i, and i-th hidden variable z_i
- Energy functions for hidden-hidden connections are chosen as

$$E(z_i, z_j) = -\beta z_i z_j \ , \tag{9}$$

  with a weight constant β
- Both functions provide lower energy for identical pixel signs: we encourage
  1. solutions that are close to the input
  2. solutions that are smooth (neighboring pixels are identical)
- To prefer a label, add also a bias term $E(z_i) = hz_i$ with suitable h

Example: Denoising a Binary Image — Whole Energy Function and Result

- The whole energy function is then

$$E(\mathbf{z}, \mathbf{x}) = h \sum_{z_i} z_i - \beta \sum_{z_i, z_j \in \mathcal{N}(z_i)} z_i z_j - \eta \sum_{(x_i, z_i)} x_i z_i \tag{10}$$

- Top left: clean image
- Top right: noisy image
- Bottom left: Gibbs sampler output
- Bottom right: Graph cut output

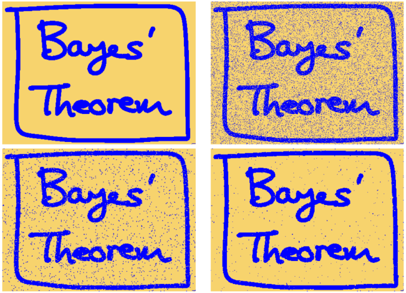


Figure 8.30 Illustration of image de-noising using a Markov random field. The top row shows the original binary image on the left and the corrupted image after randomly changing 10% of the pixels on the right. The bottom row shows the restored images obtained using iterated conditional models (ICM) on the left and using the graph-cut algorithm on the right. ICM produces an image where 96% of the pixels agree with the original image, whereas the corresponding number for graph-cut is 99%.

Remarks

- Potentials with energy terms $E(x_i, z_i)$ are also called **unary potentials**
- Potentials with energy terms $E(z_i, z_j)$ are also called **pairwise potentials**
- For anyone with an optimization background: one can identify unary potentials with the data term, and pairwise potentials with the regularizer
- The value range of the labels and the inputs constrains the potential functions:
  - Binary labels ± 1 (see example above) work well with product terms
  - Non-binary labels with meaningful linear distances work well with Minkowski norms
- Example non-binary labeling: Image denoising on intensities $[0; 255]$
  - Unary energy term can be $\eta \|x_i - y_i\|_2^2$: low energy for solutions $\mathbf{y}$ that are close to the input $\mathbf{x}$
  - Pairwise energy term can be $\beta \|y_i - y_j\|_2^2$: low energy for solutions $\mathbf{y}$ where neighboring pixels y_i, y_j are more similar