Statistical Machine Learning (BE4M33SSU)
Lecture 11: Markov Random Fields

Czech Technical University in Prague

- Markov Random Fields & Gibbs Random Fields
- Approximated Inference for MRFs
- (Generative) Parameter learning for MRFs

1. Motivation: Two Examples from Computer Vision

Example 1 (Image segmentation)**.** Consider the following image segmentation model, where $x\colon V \to \mathbb{R}^3$ denotes an image and $s\colon V \to K$ denotes its segmentation (K is the set of segment labels)

$$p(s) = \prod_{i\in V} p(s_i) = \frac{1}{Z(u)} \exp\Big[\sum_{i\in V} u_i(s_i)\Big] \quad \text{and} \quad p(x \,|\, s) = \prod_{i\in V} p(x_i \,|\, s_i)$$

This model is pixel-wise independent and, consequently, so is the inference.

We want to take into account that:

- neighbouring pixels belong more often than not to the same segment,
- the segment boundaries are in most places smooth, . . .

Hence, we consider a more complex prior model for segmentations

$$p(s) = \frac{1}{Z(u)} \exp\Big[\sum_{i\in V} u_i(s_i) + \sum_{\{i,j\}\in E} u_{ij}(s_i, s_j)\Big],$$

where E are edges connecting neighbouring pixels in V.

1. Motivation: Two Examples from Computer Vision

Example 2 (Motion Flow)**.** Given two images $x, x' \colon V \to \mathbb{R}^3$ from a video, determine the motion flow, i.e. find a displacement vector $v_i \in \mathbb{Z}^2$ for each pixel $i \in V$.

- projections of the same 3D points look similar in x and x'.
- 3D points projected onto neighbouring image pixels move more often than not coherently.
- We consider a discriminative model $p(v \,|\, x, x')$ since we do not intend to model the image appearance.

$$p(v \,|\, x, x') = \frac{1}{Z(x, x')} \exp\Big[- \sum_{i \in V} \|x_i - x'_{i+v_i}\|^2 - \alpha \sum_{\{i,j\} \in E} \|v_i - v_j\|^2 \Big]$$

The first term in the model can be generalised, by using $f(x_{c_i})$ instead of x_i, where $f(x_{c_i}) \in \mathbb{R}^n$ denotes a feature vector computed by a CNN for the image patch c_i centered at pixel $i \in D$.

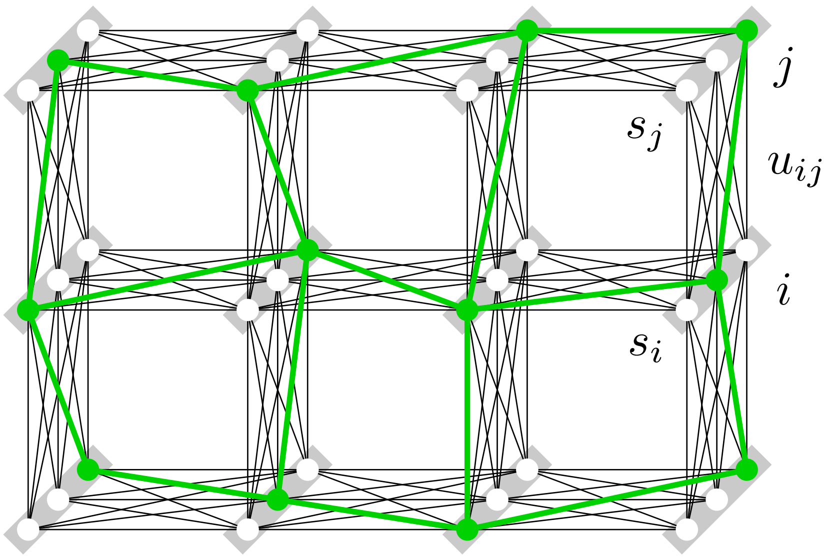
Such models can be generalised for stereo cameras and combined with segmentation approaches.

2. Markov Random Fields & Gibbs Random Fields

Let (V, E) denote an undirected graph and let $s = \{s_i \mid i \in V\}$ be a field of random variables indexed by the nodes of the graph and taking values from a finite set K. (Given a set of nodes $C \subset V$, we denote the field configuration on it by s_C)

Definition 1. A joint probability distribution $p(s)$ is a Gibbs Random Field on the graph (V, E) if it factorises over the the nodes and edges, i.e.

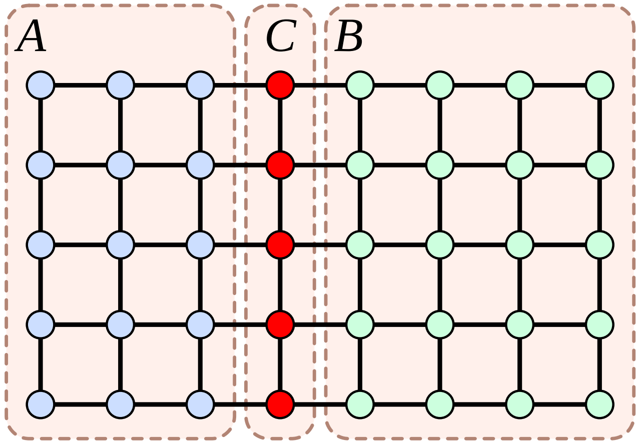
$$p(s) = \frac{1}{Z(u)} \exp\Big[\sum_{i \in V} u_i(s_i) + \sum_{\{i,j\} \in E} u_{ij}(s_i, s_j)\Big].$$



(V, E): 3x4 grid, grey bars: variables s_i, circles: values from K, green: a labelling s

2. Markov Random Fields & Gibbs Random Fields

A Gibbs Random Field w.r.t. the graph (V, E) is also a *Markov Random Field*, because it has the following Markov property:



$$p(s_A, s_B \,|\, s_C) = p(s_A \,|\, s_C)\, p(s_B \,|\, s_S)$$

holds for any subsets $A, B \subset V$ and a separating set C.

The following tasks for MRFs/GRFs are NP-complete

- Computing the most probable labelling $s^* \in \arg\max_{s \in K^V} p(s)$.
- Computing the normalisation constant

$$Z(u) = \sum_{s \in K^V} \exp\Big[\sum_{i \in V} u_i(s_i) + \sum_{\{i,j\} \in E} u_{ij}(s_i, s_j)\Big].$$

The same holds for computing marginal probabilities of $p(s)$.

3. Computing the most probable labelling of an MRF: Boolean case

Consider $\log p(s)$, replace $u \to -u$. The task reads then

$$\sum_{i\in V} u_i(s_i) + \sum_{\{i,j\}\in E} u_{ij}(s_i, s_j) \to \min_{s\in K^V}$$

The variables s_i, $i \in V$ are boolean: the pseudo-Boolean functions u_i, u_{ij} can be written as multi-linear polynomials. In particular, the functions $u_{ij}(s_i, s_j)$ can be written as

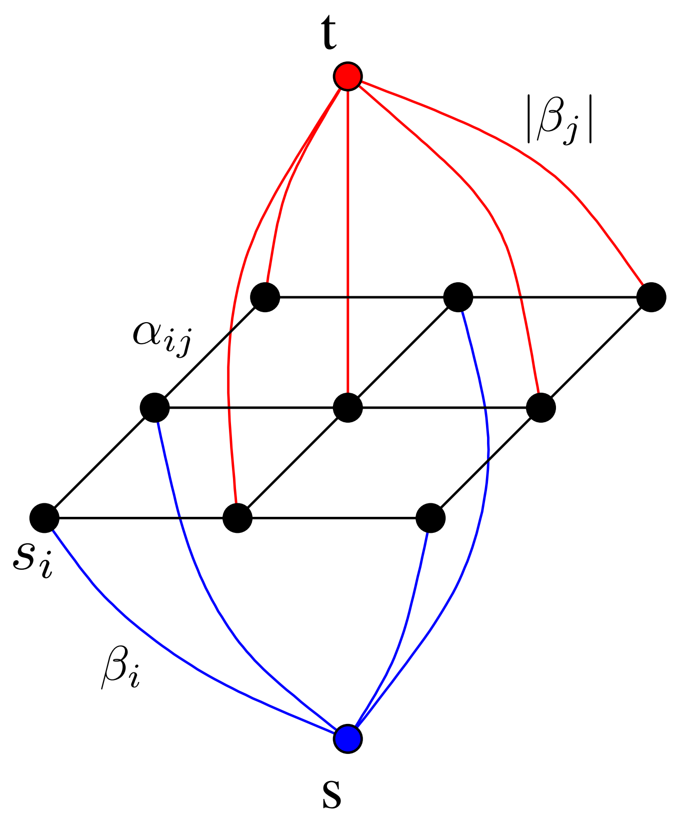
$$u_{ij}(s_i, s_j) = -2\alpha_{ij} s_i s_j + a_i s_i + b_i s_j = \alpha_{ij}|s_i - s_j| + a'_i s_i + b'_j s_j$$

up to additive constants. Thus, after re-defining the unary functions $u_i(s_i)$, the task reads as

$$\begin{aligned} s^* &= \arg\min_{s\in K^V} \sum_{\{i,j\}\in E} \alpha_{ij}|s_i - s_j| + \sum_{i\in V} \beta_i s_i \\ &= \arg\min_{s\in K^V} \sum_{\{i,j\}\in E} \alpha_{ij}|s_i - s_j| + \sum_{i\in V_+} \beta_i s_i + \sum_{i\in V_-} |\beta_i|(1 - s_i), \end{aligned}$$

where $V_+ = \{i \in V \mid \beta_i \geqslant 0\}$ and $V_- = V \setminus V_+$. This is a **s-t MinCut-problem!**

3. Computing the most probable labelling of an MRF: Boolean case



The binary labels $s_i = 0, 1$ encode the partition set to which $i \in V$ is assigned.

- the task can be solved in polynomial time via MinCut – MaxFlow duality if all edge weights are non-negative, i.e. $\alpha_{ij} \geqslant 0$, $\forall \{i, j\} \in E$,
- if some of the α-s are negative: apply approximation algorithms, e.g. relax the discrete variables to $s_i \in [0, 1]$, consider an LP-relaxation of the task and solve the LP task e.g. by Tree-Reweighted Message Passing (Kolmogorov, 2006)

4. Computing the most probable labelling: general case

Approximation algorithms for the general case, when $s_i \in K$

$$u(s) = \sum_{i \in V} u_i(s_i) + \sum_{\{i,j\} \in E} u_{ij}(s_i, s_j) \to \min_{s \in K^V}$$

Move making algorithms: Construct a sequence of labellings $s^{(t)}$ with decreasing values of the objective function by

- Defining neighbourhoods $\mathcal{N}(s) \subset K^V$ such that the restricted task

  $$\operatorname*{arg\,min}_{s \in \mathcal{N}(s')} \sum_{\{i,j\} \in E} u_{ij}(s_i, s_j) + \sum_{i \in V} u_i(s_i)$$

  is solvable in polynomial time for every s'.

- Iterating

  $$s^{(t+1)} \in \operatorname*{arg\,min}_{s \in \mathcal{N}(s^{(t)})} \sum_{\{i,j\} \in E} u_{ij}(s_i, s_j) + \sum_{i \in V} u_i(s_i)$$

  until no further improvement possible.

4. Computing the most probable labelling: general case

α-**Expansions** (Boykov et al., 2001)

- Define the neighbourhoods by choosing a label $\alpha \in K$ and setting

$$\mathcal{N}_\alpha(s') = \left\{ s \in K^V \;\middle|\; s_i = \alpha \text{ if } s_i \neq s'_i \right\}.$$

  Notice that $|\mathcal{N}_\alpha(s')| \sim 2^V$.

- The restricted task

$$\operatorname*{arg\,min}_{s \in \mathcal{N}_\alpha(s')} \sum_{\{i,j\} \in E} u_{ij}(s_i, s_j) + \sum_{i \in V} u_i(s_i)$$

  can be encoded as labelling problem with boolean variables $y_i = \begin{cases} 1 & \text{if } s_i = \alpha \\ 0 & \text{if } s_i = s'_i \end{cases}$

- It can be solved by MinCut-MaxFlow if

$$u_{ij}(k, k') + u_{ij}(\alpha, \alpha) \leqslant u_{ij}(\alpha, k') + u_{ij}(k, \alpha)$$

  holds for all pairwise functions u_{ij} and all $k, k' \in K$.

5. Learning parameters of MRFs

Learning task: Given i.i.d. training data $\mathcal{T}^m = \{s^\ell \in K^V \mid \ell = 1, \ldots, m\}$, estimate the parameters u_i, u_{ij} of the MRF.

The maximum likelihood estimator reads

$$\log p_u(\mathcal{T}^m) = \frac{1}{m}\sum_{\ell=1}^{m}\Big[\sum_{\{i,j\}\in E} u_{ij}(s_i^\ell, s_j^\ell) + \sum_{i\in V} u_i(s_i^\ell)\Big] - \log Z(u) \to \max_{u_i, u_{ij}} .$$

It is intractable: the objective function is concave in u, but we can compute neither $\log Z(u)$ nor its gradient (in polynomial time).

We can use the **pseudo-likelihood** estimator (Besag, 1975) instead. It is based on the following observation

- Let $\mathcal{N}_i$ denote the neighbouring nodes of $i \in V$.
- We can compute the conditional distributions

$$p(s_i \mid s_{V\setminus i}) \stackrel{!}{=} p(s_i \mid s_{\mathcal{N}_i}) \propto e^{u_i(s_i)} \prod_{j\in\mathcal{N}_i} e^{u_{ij}(s_i, s_j)}$$

5. Learning parameters of MRFs

The pseudo-likelihood of an single example $s \in \mathcal{T}^m$ is defined by

$$L_p(u) = \sum_{i \in V} \log p_u(s_i \mid s_{\mathcal{N}_i})$$

$$= 2 \sum_{\{i,j\} \in E} u_{ij}(s_i, s_j) + \sum_{i \in V} u_i(s_i) - \sum_{i \in V} \log \sum_{s_i \in K} \exp\Big[u_i(s_i) + \sum_{j \in \mathcal{N}_i} u_{ij}(s_i, s_j)\Big]$$

The pseudo-likelihood estimator is

- a concave function of the parameters u,
- tractable, i.e. both $L_p(u, \mathcal{T}^m)$ and its gradient are easy to compute,
- consistent.