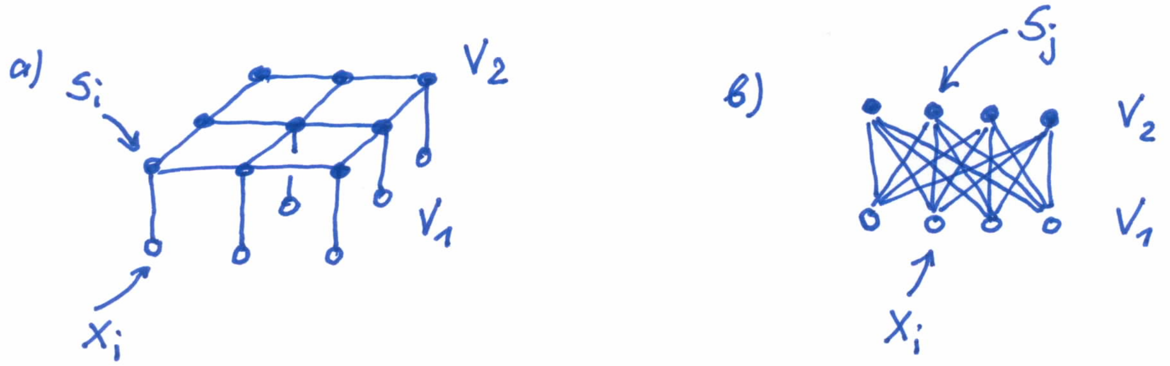
6. Unsupervised learning of GRFs

Consider a random field (X, S), where $X = \{X_i \mid i \in V_1\}$ is a subfield of F-valued random variables and $S = \{S_i \mid i \in V_2\}$ is a subfield of K-valued random variables. We assume that its joint p.d. $p(x,s)$ is a Gibbs random field w.r.t. the graph structure $(V = V_1 \cup V_2,\ E = E_1 \cup E_2 \cup E_{12})$

Example 1



The joint p.d. can be written as

$$p_u(x,s) = \frac{1}{Z(u)} \exp\Big[\sum_{ij \in E_1} u_{ij}(x_i, x_j) + \sum_{ij \in E_2} u_{ij}(s_i, s_j) + \sum_{ij \in E_{12}} u_{ij}(x_i, s_j)\Big]$$

or shorter

$$p_u(x,s) = \frac{1}{Z(u)} \exp\langle \varphi(x,s), u\rangle$$

We are given a sample of realisations of the subfield X, i.e.

$$\mathcal{T}_\ell = \{x^j \in F^{V_1} \mid j = 1, \dots, \ell\},$$

where the x^j are i.i.d. sampled from $p_u(x,s)$ (the corresponding realisations of the subfield S are hidden, i.e. not available).

The task is to estimate the model parameters u.

Applying a maximum likelihood estimator, we have

$$\frac{1}{\ell}\sum_{x\in T_\ell} \log p_u(x) = \frac{1}{\ell}\sum_{x\in T_\ell} \log \sum_{s\in K^{V_2}} p_u(x,s) \to \max_u$$

Substitution of p_u gives

$$\underbrace{\log Z(u)}_{g(u)} - \underbrace{\frac{1}{\ell}\sum_{x\in T_\ell} \log \sum_{s\in K^{V_2}} \exp\langle \varphi(x,s), u\rangle}_{h(u)} \to \min_u$$

We have to minimise a difference of convex functions

$$g(u) - h(u) \to \min_u$$

The DC-dual task is

$$h^*(\mu) - g^*(\mu) \to \min_\mu$$

See sec. 1E.

The DC-algorithm (see ibid.) constructs a pair of converging sequences $u^{(t)}, \mu^{(t)}$, $t = 1, 2, \ldots$

$$\begin{cases} \mu^{(t)} = \nabla h(u^{(t)}) \\ u^{(t+1)} \in \partial g^*(\mu^{(t)}) \end{cases}$$

Let us analyse the substeps:

a) $$\nabla h(u) = \frac{1}{\ell} \sum_{x \in \mathcal{T}_\ell} \frac{1}{\sum_{s \in K^{V_2}} \exp\langle \varphi(x,s), u \rangle} \cdot \sum_{s \in K^{V_2}} \exp\langle \varphi(x,s), u \rangle \, \varphi(x,s)$$

$$= \frac{1}{\ell} \sum_{x \in \mathcal{T}_\ell} \sum_{s \in K^{V_2}} p_u(s \mid x) \, \varphi(x,s)$$

$$= \frac{1}{\ell} \sum_{x \in \mathcal{T}_\ell} \mathbb{E}_u(\varphi \mid x) \quad \to \mu$$

b) $$g^*(\mu) = \inf_p \Big\{ \sum_{x,s} p(x,s) \log p(x,s) \;\Big|\; \mathbb{E}_p(\varphi) = \mu, \; p \in \mathcal{P} \Big\}$$

We know

$$u \in \partial g^*(\mu) \iff \mu = \nabla g(u) = \mathbb{E}_u(\varphi)$$

Hence, we obtain that the <u>DC-algorithm</u> applied to the considered learning task is nothing else than an <u>Expectation-Maximisation</u> algorithm.

Initialise with some $u^{(1)}$ and iterate

E-step: $u^{(t)} \rightarrow$ compute $\mu^{(t)} = \frac{1}{\ell} \sum_{x \in \mathcal{T}_\ell} \mathbb{E}_{u^{(t)}}(\varphi \mid x)$

M-step: $\mu^{(t)} \rightarrow$ compute $u^{(t+1)}$ s.t.

$$\mathbb{E}_{u^{(t+1)}}(\varphi) = \mu^{(t)}$$

The M-step solves a supervised learning task!

Algoritms & approximations

- Both steps are NP-hard for general GRFS, see sec. 4 and 5.
- If Gibbs sampling is used as approximation, the algorithm reads as follows

E-Step: for each $x \in \mathcal{T}_\ell$ run a Gibbs sampler to estimate the posterior statistics $\mathbb{E}_{u^{(t)}}(\varphi \mid x)$

$$\Rightarrow \mu^{(t)} = \frac{1}{\ell} \sum_{x \in \mathcal{T}_\ell} \mathbb{E}_{u^{(t)}}(\varphi \mid x)$$

M-step: Init $\tilde{u}^{(0)} = u^{(t)}$, iterate

- run a Gibbs sampler to estimate $\tilde{\mu}^{(m)} = \mathbb{E}_{\tilde{u}^{(m)}}(\varphi)$
- set $\tilde{u}^{(m+1)} = \tilde{u}^{(m)} + \alpha\left(\mu^{(t)} - \tilde{\mu}^{(m)}\right)$

until $\|\mu^{(t)} - \tilde{\mu}^{(m)}\| < \varepsilon$

set $u^{(t+1)} = \tilde{u}^{(m)}$

This results in a double loop algorithm.

Possible speed-ups & variants:

- replace the Gibbs sampler in the inner loop by some faster approximation, e.g. belief propagation
- try to estimate the gradient $\mu^{(t)} - \tilde{\mu}^{(m)}$ in the M-step directly e.g. by (persistent) contrastive divergence
- if the graph $(V_1 \cup V_2, E)$ is bipartite, i.e. $E = E_{12}$:
  - the E-step is tractable because
    $$p_u(s \mid x) = \prod_{i \in V_2} p_u(s_i \mid x)$$
  - We can easily sample from $p_u(s \mid x)$ for all $x \in \mathcal{T}_e \Rightarrow \ldots \Rightarrow$ replace the ML estimator in the M-step by the pseudo-likelihood estimator.