12. (Min,+)-problems for graphical models

A. Map inference for GRFs

Let $\{x_j \mid j \in V'\}$, $x_j \in F$ be a random field of observables (features)

$\{s_i \mid i \in V\}$, $s_i \in K$ be a random field of hidden states

Assume their joint p.d. $p(x,s)$ is a GRF w.r.t. the system $\mathcal{C}$ of subsets of V

$$p(x,s) = \frac{1}{Z} \exp\Big[\sum_{c \in \mathcal{C}} u_c(x, s_c)\Big]$$

Inference: Given $x \in F^{V'}$ infer s w.r.t. 0/1 loss $\Rightarrow$ MAP

$$s^* = \operatorname*{argmax}_{s \in K^V} p(x,s) = \operatorname*{argmax}_{s \in K^V} \sum_{c \in \mathcal{C}} u_c(x, s_c)$$

- discrete optimisation problem for $|V|$ variables
- objective function $\triangleq$ sum of functions, each depending on a subset of variables.

Particular case: $\mathcal{C}$ is the structure of a graph (V,E), $s: V \to K$ are K-valued labellings, x is fixed $\Rightarrow$ Solve the task

$$s^* \in \operatorname*{argmin}_{s \in K^V} U(s) = \operatorname*{argmin}_{s \in K^V} \Big[\sum_{i \in V} u_i(s_i) + \sum_{ij \in E} u_{ij}(s_i, s_j)\Big],$$

where $u_i: K \to \mathbb{R}$, $u_{ij}: K^2 \to \mathbb{R}$.

- Easy to solve if (V,E) is acyclic (see Sec. 4 and 10)
- NP-complete in general (MaxClique)

Options:

- search for tractable subclasses
- search for approximation algorithms

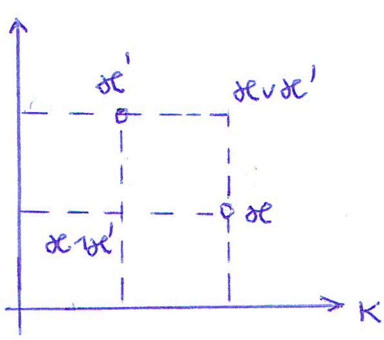
B. Submodular (Min,+)-problems

- Let K be completely ordered and denote min, max w.r.t. this order by $\wedge, \vee$
- K^n is a distributive lattice $\hat{=}$ poset with operations „infimum" and „supremum"

$x, x' \in K^n$

$x \wedge x' = (x_1 \wedge x'_1, \dots, x_n \wedge x'_n)$

$x \vee x' = (x_1 \vee x'_1, \dots, x_n \vee x'_n)$



Definition 1 Let K be completely ordered. A real valued function $u: K^n \to \mathbb{R}$ is submodular if

$$u(x \wedge x') + u(x \vee x') \leq u(x) + u(x')$$

holds $\forall x, x' \in K^n$.

Remarks

(1) if „$\leq$" is replaced by „$\geq$" $\to$ supermodular function

(2) any function $u: K \to \mathbb{R}$ is submodular and supermodular

(3) any function $u: K^2 \to \mathbb{R}$ can be decomposed into a sum of a super- and submodular part

(4) if $|K| = 2$, then K^V is a Boolean lattice and any $x \in K^V$ can be identified with a subset of V: $\{i \in V \mid x_i = 1\}$ (we assume $K = \{0,1\}$)

Examples

(1) Let $K = \{0,1\}$ be ordered. The function $u: K^2 \to \mathbb{R}$ defined by $u(k,k') = |k - k'|$ is submodular

(2) Let $K = \{0,1,2,\dots,m\}$ be ordered. Consider functions $u: K^2 \to \mathbb{R}$

$u(k,k') = |k - k'|$ is submodular

$u(k,k') = \mathbb{1}\{k \neq k'\}$ is not submodular

$u(k,k') = (k - k')^2$ is submodular

(3) Let $K = \{0,1\}$ be ordered. Consider the function $u: K^V \to \mathbb{R}$ defined by $u(x) = -\big|\,\|x\|_1 - m\big|$. It is submodular.

Theorem 1 (Iwata, Fleisher, Fujishige)

Any submodular function on $\{0,1\}^n$ can be minimised with complexity $\mathcal{O}((n^6 \mu + n^7)\log n)$, where μ denotes the time required for computing the function value

Theorem 2 (Schlesinger, Flach, 2006)

If all arity 2 functions $u_{ij}: K^2 \to \mathbb{R}$ of a (Min,+)-problem on a graph are submodular w.r.t. some ordering of K, then the (Min,+)-problem is equivalent to a MinCut-problem and solvable with complexity $\mathcal{O}(V^2 E K^4)$

Transforming a submodular (Min,+)-problem on a graph (V,E) into a MinCut problem: We assume $|K|=2$ for simplicity.

(1) Express the (Min,+)-problem in canonical form, using equivalent transformations,

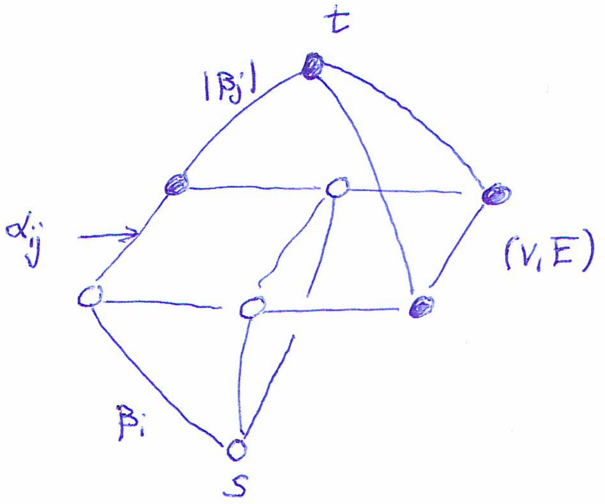
$$\sum_{ij \in E} \alpha_{ij} |s_i - s_j| + \sum_{i \in V} \beta_i s_i \to \min_{s \in K^V},$$

where $s_i = 0,1$. Submodularity ensures $\alpha_{ij} \geq 0 \;\forall \{i,j\} \in E$.

(2) Rewrite the linear terms. Let $V_+ = \{i \in V \mid \beta_i \geq 0\}$, $V_- = V \setminus V_+$

$$\sum_{i \in V} \beta_i s_i = \sum_{i \in V_+} \beta_i s_i + \sum_{i \in V_-} |\beta_i| (1 - s_i) + \text{const.}$$

(3) The task is now equivalent to an st-MinCut problem with positive edge weights



$\widetilde{V} = V \cup \{s,t\}$

$\widetilde{E} = E \cup E_+ \cup E_-$

$E_+ = \{\{s,i\} \mid i \in V_+\}$

$E_- = \{\{t,i\} \mid i \in V_-\}$

(4) Solve it by MinCut $\Leftrightarrow$ MaxFlow e.g.

- augmenting path algorithm
- pre-flow push algorithm
- V. Kolmogorov's algorithm