13. Approximation algorithms for (Min,+)-problems

We still consider the task

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

A. Iterated descent

- Define a family of neighbourhoods $N_m(s) \subset K^V$, $m=1,..,M$
- Repeatedly solve the restricted problem

$$s^{(t+1)} \in \underset{s\in N_m(s^{(t)})}{\operatorname{argmin}}\, U(s)$$

until no further improvement is possible, i.e.

$$s^{(t)} \in \underset{s\in N_m(s^{(t)})}{\operatorname{argmin}}\, U(s) \quad \forall\, m=1,..,M$$

α-expansions (Boykov et al. 2001)

For each label $\alpha \in K$ define the neighbourhood

$$N_\alpha(s) = \{s' \in K^V \mid s'_i = \alpha \text{ if } s'_i \neq s_i \ \forall i \in V\}$$

Their size is exponential, i.e. $|N_\alpha(s)| \sim 2^{|V|}$.

Is the task $\underset{s\in N_\alpha(s')}{\operatorname{argmin}}\, U(s)$ solvable in polynomial time?

Yes, if $u_{ij}(k,k') + u_{ij}(\alpha,\alpha) \leq u_{ij}(k,\alpha) + u_{ij}(\alpha,k')$ holds $\forall \{ij\}\in E$ and $\forall\, k,k' \in K \setminus \alpha$. This can be seen by constructing a binary valued (Min,+)-problem that is equivalent to the restricted optimisation task

$$V' = \{i \in V \mid s'_i \neq \alpha\}, \quad E' = \{\{i,j\} \in E \mid i,j \in V'\}$$

$y_i = 0,1$ encodes $s \in N_\alpha(s')$, i.e.

$s_i = s'_i \Leftrightarrow y_i = 0$ and $s_i = \alpha \Leftrightarrow y_i = 1$

The pairwise functions of this equivalent problem are submodular if the condition given above holds.

Example 1 Consider the Potts model $u_{ij}(k,k') = a_{ij}(1-\delta_{kk'})$, $a_{ij} > 0$. It is not submodular if $|K| > 2$. However, it fulfills the above conditions. ■

Theorem 1 (w/o proof)
Let $\bar{s}$ be a fixpoint of α-expansions $\forall \alpha \in K$. Then

$$U(\bar{s}) \le 2C \min_{s \in K^V} U(s),$$

where C is defined by

$$C = \max_{ij \in E} \frac{\max_{k \neq k'} u_{ij}(k,k')}{\min_{k \neq k'} u_{ij}(k,k')}.$$ ■

$\alpha\beta$-swaps (Boykov et al. 2001)
Let us define neighbourhoods $N_{\alpha\beta}$ for each pair of labels

$$N_{\alpha\beta}(s) = \left\{ s' \in K^V \;\middle|\; s'_i = \begin{cases} s_i & \text{if } s_i \neq \alpha, \beta \\ \alpha, \beta & \text{otherwise} \end{cases} \right.$$

The reduced task $\underset{s \in N_{\alpha\beta}(s')}{\operatorname{argmin}} U(s)$ is tractable if the restriction of each $u_{ij} : K^2 \to \mathbb{R}$ to $\{\alpha,\beta\}^2 \subset K^2$ is submodular. $\forall \alpha, \beta \in K$.

Example 2 Consider the truncated metric on $K \subset \mathbb{Z}$ given by

$$u_{ij}(k,k') = a_{ij} \min(c, |k-k'|), \quad a_{ij} > 0.$$

It is not submodular. It allows $\alpha\beta$-swaps, but does not allow α-expansions. ■

Remark 1 Another class of approximation algorithms constructs submodular upper bounds of the objective functions instead of considering restricted problems. I.e. given $s^{(t)} \in K^V$, construct an upper bound $\tilde{U}_t \ge U$ s.t. $\tilde{U}_t$ is submodular and

$$\tilde{U}_t(s) \ge U(s) \;\; \forall s \in K^V \quad \text{and} \quad \tilde{U}_t(s^{(t)}) = U(s^{(t)}).$$

Then solve

$$s^{(t+1)} \in \underset{s \in K^V}{\operatorname{argmin}} \tilde{U}_t(s).$$ ■

B. Algorithms based on LP-relaxations

Loopy belief propagation (aka message passing): Apply equivalent transformations that resemble dynamic programming on trees until convergence. This is not well grounded (see next section).

More principled: Start from an LP-relaxation of the discrete optimisation problem

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

A lower bound is given by

$$\sum_{i\in V}' \min_{k\in K} u_i(k) + \sum_{ij\in E}' \min_{k,k'\in K} u_{ij}(k,k') \leq \min_{s\in K^V} U(s)$$

Let us combine it with equivalent transformations and then maximise the bound w.r.t. them

$$B(\varphi) = \sum_{i\in V}' \min_{k\in K}\Big[u_i(k) - \sum_{j\in N_i}' \varphi_{ij}(k)\Big] +$$

$$+ \sum_{ij\in E}' \min_{k,k'\in K} \Big[\varphi_{ij}(k) + u_{ij}(k,k') + \varphi_{ji}(k')\Big] \to \max_{\varphi}$$

This can be expressed as a <u>linear optimisation</u> task by introducing additional variables

$$\sum_{i\in V}' c_i + \sum_{ij\in E}' c_{ij} \to \max_{\varphi, c}$$

s.t.

$$c_i + \sum_{j\in N_i}' \varphi_{ij}(k) \leq u_i(k) \quad \forall i\in V,\ \forall k\in K$$

$$c_{ij} - \varphi_{ij}(k) - \varphi_{ji}(k') \leq u_{ij}(k,k') \quad \forall ij\in E,\ \forall k,k'\in K$$

Notice, that this LP-task is <u>dual</u> to the following direct relaxation of the discrete optimisation task. Encode the label $s_i \in K$ by 1-out-of-K encoding with components denoted as $\lambda_i(k) = 0,1$ and, similarly, for edges by $\lambda_{ij}(k,k') = 0,1$

$$\sum_{i\in V}\sum_{k\in K} \lambda_i(k)\, u_i(k) + \sum_{ij\in E}\sum_{k,k'\in K} \lambda_{ij}(k,k')\, u_{ij}(k,k') \to \min_{\lambda \geq 0}$$

s.t. $\quad \lambda_i(k) = \sum_{k'\in K} \lambda_{ij}(k,k') \quad \forall\, i,j\in E,\ \forall\, k\in K$

$$\sum_{k\in K} \lambda_i(k) = 1 \qquad \forall\, i\in V$$

$$\sum_{k,k'\in K} \lambda_{ij}(k,k') = 1 \qquad \forall\, ij\in E$$

Relaxing the integrality constraints $\lambda_i(k)=0,1$, $\lambda_{ij}(k,k')=0,1$ makes this an LP task.

Then, apply suitable algorithms for solving the primal task, or solving the dual task, or both simultaneously.