13. Approximation algorithms for \((\text{Min,}+)\)-problems

We shall consider the task

\[ S^{*} \in \arg\min_{S \in \mathcal{K}} U(s) = \arg\min_{S \in \mathcal{K}} \left[ \sum_{i \in V} U_{i}(s_{i}) + \sum_{j \in E} U_{j}(s_{i}, s_{j}) \right] \]

A. Iterated descent

- Define a family of neighbourhoods \( \mathcal{N}_{m}(s) \subseteq \mathcal{K} \), \( m = 1, \ldots, M \)
- Repeatedly solve the restricted problem

\[ S^{(t+1)} \in \arg\min_{S \in \mathcal{N}_{m}(s^{(t)})} U(s) \]

until no further improvement is possible, i.e.

\[ S^{(t)} \in \arg\min_{S \in \mathcal{N}_{m}(s^{(t)})} U(s) \quad \forall m = 1, \ldots, M \]

\( \alpha \)-expansions (Boykov et al. 2001)

For each label \( \alpha \in \mathcal{K} \) define the neighbourhood

\[ \mathcal{N}_{\alpha}(s) = \{ S' \in \mathcal{K} \mid S_1' = \alpha \text{ if } s_1' \neq s_1, \forall i \in V \} \]

Their size is exponential, i.e. \(|\mathcal{N}_{\alpha}(s)| \sim 2^{1V} \)

Is the task \( \arg\min_{S \in \mathcal{N}_{\alpha}(s)} U(s) \) solvable in polynomial time?

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

\[ V' = \{ i \in V \mid s'_{i} \neq \alpha \} \]
\[ E' = \{ i,j \in E \mid i,j \in V' \} \]

\( y_{i} = 0,1 \) encodes \( S \in \mathcal{N}_{\alpha}(s') \), i.e.

\[ S_{i} = S'_{i} \iff y_{i} = 0 \quad \text{and} \quad S_{i} = \alpha \iff y_{i} = 1 \]

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 $S$ be a fixpoint of $\alpha$-expansims $\forall \alpha \in K$. Then

$$U(S) \leq 2 \cdot \min_{\alpha \in K} U(S)$$

where $C$ is defined by

$$C = \max_{ij \in E} \frac{\max_{k \in K} U_{ij}(k, k')}{\min_{k \in K} U_{ij}(k, k')}$$

$d\beta$-Swaps (Boykov et al. 2001)

Let us define neighborhoods $N_{d\beta}$ for each pair of labels

$$N_{d\beta}(S) = \{ S' \subseteq K^V \mid S'_i = S_i \text{ if } S_i + d_{\beta} \text{ otherwise} \}$$

The reduced task $\arg \min_{S \subseteq K^V} U(S)$ is tractable if the restriction $S \subseteq N_{d\beta}(S')$

of each $U_{ij}: K^2 \rightarrow R$ to $\{d_{\beta}\} \subseteq K^2$ is submodular. $\forall d_{\beta} \in K$.

Example 2. Consider the truncated metric on $KC\overline{Z}$ given by

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

It is not submodular. It allows $d\beta$-Swaps, but does not allow $\alpha$-expansims.

Remark 1. Another class of approximation algorithms constructs

submodular upper bounds of the objective functions instead of considering restricted problems. I.e. given $S^{(n)} \in K^V$, construct an upper bound $\tilde{U}_e = U$ s.t. $\tilde{U}_e$ is submodular and $\tilde{U}_e(S) \geq U(S) \forall S \in K^V$ and $\tilde{U}_e(S^{(n)}) = U(S^{(n)})$.

Then solve

$$S^{(n+1)} = \arg \min_{S \subseteq K^V} \tilde{U}_e(S).$$
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

\[ U(s) = \sum_{i \in V} U_i(s_i) + \sum_{ij \in E} U_{ij}(s_i, s_j) \rightarrow \min \]

\[ \text{subject to} \quad s \in \mathcal{K} \]

A lower bound is given by

\[ \sum_{i \in V} \min_{k \in \mathcal{K}} U_i(k) + \sum_{ij \in E, k, k' \in \mathcal{K}} \min_{k' \in \mathcal{K}} U_{ij}(k, k') \leq \min_{s \in \mathcal{K}} U(s) \]

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

\[ B(\psi) = \sum_{i \in V} \min_{k \in \mathcal{K}} \left[ U_i(k) - \sum_{j \in V_i} \psi_{ij}(k) \right] + \]

\[ + \sum_{ij \in E, k, k' \in \mathcal{K}} \min_{k' \in \mathcal{K}} \left[ \psi_{ij}(k) + U_{ij}(k, k') + \psi_{ij}(k') \right] \rightarrow \max \psi \]

This can be expressed as a **linear optimisation task** by introducing additional variables

\[ \sum_{i \in V} c_i + \sum_{ij \in E} c_{ij} \rightarrow \max \psi, c \]

s.t.

\[ c_i + \sum_{j \in V_i} \psi_{ij}(k) \leq U_i(k) \quad \forall i \in V, \forall k \in \mathcal{K} \]

\[ c_{ij} - \psi_{ij}(k) - \psi_{ij}(k') \leq U_{ij}(k, k') \quad \forall i \in V, \forall j \in \mathcal{E}, \forall k, k' \in \mathcal{K} \]

Notice, that this LP-task is dual to the following direct relaxation of the discrete optimisation task. Encode the label \( s \in \mathcal{K} \) by \( 1 \)-out-of-\( K \) encoding with components denoted as \( \lambda_i(k) = 0,1 \) and, similarly, for edges by \( \lambda_{ij}(k, k') = 0,1 \)
\[ \sum_{i \in V} \sum_{k \in \mathcal{K}} \lambda_i(k) u_i(k) + \sum_{ij \in E} \sum_{k \in \mathcal{K}} \lambda_{ij}(k, k') u_{ij}(k, k') \rightarrow \min_{\lambda \geq 0} \]

s.t. \[ \lambda_i(k) = \sum_{k' \in \mathcal{K}} \lambda_{ij}(k, k') + \forall i \in V, k \in \mathcal{K} \]
\[ \sum_{k \in \mathcal{K}} \lambda_i(k) = 1 + \forall i \in V \]
\[ \sum_{k, k' \in \mathcal{K}} \lambda_{ij}(k, k') = 1 + \forall ij \in E \]

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

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