Probabilistic Graphical Models in Computer Vision (IN2329)

Csaba Domokos

Summer Semester 2017

6. Primal-dual Schema

Agenda for today's lecture *

Let us consider an *undirected graphical model* given by $G = (\mathcal{V}, \mathcal{E})$, which takes values from an **arbitrary** (finite) label set $\mathcal{L}$. More specially, assume that the corresponding *energy function* $E : \mathcal{L}^{\mathcal{V}} \to \mathbb{R}$ is given by

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} E_i(\mathbf{x}_i) + \sum_{(i,j) \in \mathcal{E}} w_{ij} \cdot d(\mathbf{x}_i, \mathbf{x}_j) ,$$

where E_i stands for a *unary energy function*, $w_{ij} \in \mathbb{R}$ are *weighting factors*, and d is a *metric* or a *semi-metric* (i.e. the triangle inequality is not necessary satisfied).

In the **previous lecture** we learnt about $\alpha - \beta$ swap, which *approximately* solves this problem.

Today we are going to learn about

- α-expansion, which provides an approximate solution, and
- the *linear programming* formalization of the multi-labeling problem.

α-expansion

α-expansion

α-expansion allows each variable either to keep its current label or to change it to the label $\alpha \in \mathcal{L}$. We introduce the following notation

$$\mathcal{Z}_\alpha(\mathbf{y}, \alpha) = \{\mathbf{z} \in \mathcal{Y} : z_i \in \{y_i, \alpha\} \text{ for all } i \in \mathcal{V}\} .$$

The minimization of the *energy function* E can be reformulated as follows:

$$\hat{\mathbf{z}} \in \operatorname*{argmin}_{\mathbf{z} \in \mathcal{Z}_\alpha(\mathbf{y},\alpha)} E(\mathbf{z}) = \operatorname*{argmin}_{\mathbf{z} \in \mathcal{Z}_\alpha(\mathbf{y},\alpha)} \sum_{i \in \mathcal{V}} E_i(z_i) + \sum_{(i,j) \in \mathcal{E}} E_{ij}(z_i, z_j)$$

$$= \operatorname*{argmin}_{\mathbf{z} \in \mathcal{Z}_\alpha(\mathbf{y},\alpha)} \Big[\underbrace{\sum_{i \in \mathcal{V},\, y_i = \alpha} E_i(\alpha)}_{\text{constant}} + \underbrace{\sum_{i \in \mathcal{V},\, y_i \neq \alpha} E_i(z_i)}_{\text{unary}}$$

$$+ \underbrace{\sum_{\substack{(i,j) \in \mathcal{E} \\ y_i = \alpha,\, y_j = \alpha}} E_{ij}(\alpha, \alpha)}_{\text{constant}} + \underbrace{\sum_{\substack{(i,j) \in \mathcal{E} \\ y_i = \alpha,\, y_j \neq \alpha}} E_{ij}(\alpha, z_j)}_{\text{unary}} + \underbrace{\sum_{\substack{(i,j) \in \mathcal{E} \\ y_i \neq \alpha,\, y_j = \alpha}} E_{ij}(z_i, \alpha)}_{\text{unary}} + \underbrace{\sum_{\substack{(i,j) \in \mathcal{E} \\ y_i \neq \alpha,\, y_j \neq \alpha}} E_{ij}(z_i, z_j)}_{\text{pairwise}} \Big] .$$

Local optimization

Let us consider $E_{ij}(z_i, z_j)$ for a given $(i,j) \in \mathcal{E}$:

E_{ij}	α	y_j
α	$E_{ij}(\alpha, \alpha)$	$E_{ij}(\alpha, y_j)$
y_i	$E_{ij}(y_i, \alpha)$	$E_{ij}(y_i, y_j)$

If we assume that $E_{ij} : \mathcal{L} \times \mathcal{L} \to \mathbb{R}$ is a **metric** for each $(i,j) \in \mathcal{E}$, then

$$E_{ij}(\alpha, \alpha) + E_{ij}(y_i, y_j) = E_{ij}(y_i, y_j) \leqslant E_{ij}(y_i, \alpha) + E_{ij}(\alpha, y_j) ,$$

which means that E_{ij} is **regular** w.r.t. the labeling $\mathcal{Z}_\alpha(\mathbf{y}, \alpha)$.

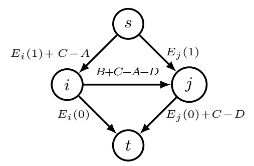
Graph construction

We need to minimize the following **regular** *energy function*:

$$\hat{\mathbf{z}} \in \operatorname*{argmin}_{\mathbf{z} \in \mathcal{Z}_\alpha(\mathbf{y},\alpha)} \underbrace{\sum_{\substack{i \in \mathcal{V} \\ y_i \neq \alpha}} E_i(z_i) + \sum_{\substack{(i,j) \in \mathcal{E} \\ y_i = \alpha,\, y_j \neq \alpha}} E_{ij}(\alpha, z_j) + \sum_{\substack{(i,j) \in \mathcal{E} \\ y_i \neq \alpha,\, y_j = \alpha}} E_{ij}(z_i, \alpha)}_{\text{unary}} + \underbrace{\sum_{\substack{(i,j) \in \mathcal{E} \\ y_i \neq \alpha,\, y_j \neq \alpha}} E_{ij}(z_i, z_j)}_{\text{pairwise}} .$$

Based on construction applied for *binary image segmentation*, we can also define a *flow network* $(\mathcal{V}', \mathcal{E}', c, \alpha, \bar{\alpha})$, where $\mathcal{V}' = \{\alpha, \bar{\alpha}\} \cup \{i \in \mathcal{V} : y_i \neq \alpha\}$ and $\mathcal{E}' = \underbrace{\{(\alpha, i), (i, \bar{\alpha}) : i \in \mathcal{V}' \backslash \{\alpha, \bar{\alpha}\}\}}_{\text{t-links}} \cup \underbrace{\{(i,j) \in \mathcal{E} : i, j \in \mathcal{V}' \backslash \{\alpha, \bar{\alpha}\}\}}_{\text{n-links}}$.

Graph construction: t-links

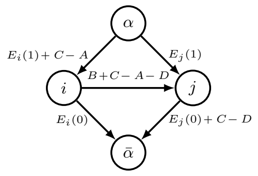


t-links: for all $i \in \mathcal{V}' \backslash \{\alpha, \bar{\alpha}\}$

$$c(\alpha, i) = E_i(y_i) + \sum_{(i,j) \in \mathcal{E},\, y_j = \alpha} E_{ij}(y_i, \alpha) + \sum_{(j,i) \in \mathcal{E},\, y_j = \alpha} E_{ji}(\alpha, y_i) + \underbrace{\sum_{(i,j) \in \mathcal{E},\, y_j \neq \alpha} E_{ij}(y_i, \alpha)}_{C} .$$

$$c(i, \bar{\alpha}) = E_i(\alpha) + \underbrace{\sum_{(j,i) \in \mathcal{E},\, y_j \neq \alpha} E_{ji}(y_j, \alpha)}_{C} - \underbrace{\sum_{(j,i) \in \mathcal{E},\, y_j \neq \alpha} E_{ji}(y_j, y_i)}_{D} .$$

Graph construction: n-links



n-links: for all $(i,j) \in \mathcal{E}$, where $i, j \in \mathcal{V}' \backslash \{\alpha, \bar{\alpha}\}$

$$c(i,j) = E_{ij}(\alpha, y_j) + E_{ij}(y_i, \alpha) - E_{ij}(y_i, y_j) \ .$$

α-expansion algorithm *

Input: An *energy function* $E(\mathbf{y}) = \sum_{i \in \mathcal{V}} E_i(y_i) + \sum_{(i,j)\in\mathcal{E}} E_{ij}(y_i, y_j)$ to be minimized, where E_{ij} is a **metric** for each $(i,j) \in \mathcal{E}$

Output: A local minimum $\mathbf{y} \in \mathcal{Y} = \mathcal{L}^{\mathcal{V}}$ of $E(\mathbf{y})$

1: Choose an arbitrary initial labeling $\mathbf{y} \in \mathcal{Y}$
2: $\hat{\mathbf{y}} \leftarrow \mathbf{y}$
3: **for all** $\alpha \in \mathcal{L}$ **do**
4: find $\hat{\mathbf{z}} \in \operatorname{argmin}_{\mathbf{z} \in \mathcal{Z}_\alpha(\hat{\mathbf{y}}, \alpha)} E(\mathbf{z})$
5: $\hat{\mathbf{y}} \leftarrow \hat{\mathbf{z}}$
6: **end for**
7: **if** $E(\hat{\mathbf{y}}) < E(\mathbf{y})$ **then**
8: $\mathbf{y} \leftarrow \hat{\mathbf{y}}$
9: Goto Step 2
10: **end if**

α-expansion is guaranteed to terminate in a finite number of cycles. This algorithm computes at least $|\mathcal{L}|$ graph cuts, which may take a lot of time, when the label space $\mathcal{L}$ is large.

Optimality *

The $\alpha - \beta$ swap does not guarantee any closeness to the global minimum. Nevertheless, the local minimum that the α-expansion algorithm will find is at most twice the global minimum $\mathbf{y}^*$.

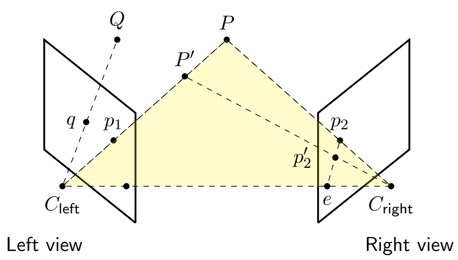
We have already assumed that E_{ij} is a metric for each $(i,j) \in \mathcal{E}$, hence $E_{ij}(\alpha, \beta) \neq 0$ for $\alpha \neq \beta \in \mathcal{L}$. Let us define

$$c = \max_{(i,j)\in\mathcal{E}} \left(\frac{\max_{\alpha \neq \beta \in \mathcal{L}} E_{ij}(\alpha, \beta)}{\min_{\alpha \neq \beta \in \mathcal{L}} E_{ij}(\alpha, \beta)} \right) \ .$$

Theorem 1. *Let $\hat{\mathbf{y}}$ be a local minimum when the expansion moves are allowed and $\mathbf{y}^*$ be the globally optimal solution. Then $E(\hat{\mathbf{y}}) \leqslant 2cE(\mathbf{y}^*)$.*

Stereo matching

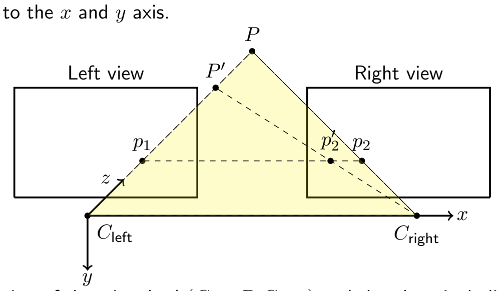
Stereo matching *



Given two images (i.e. left and right), an observed 2D point p_1 on the *left image* corresponds to a 3D point P that is situated on a line in $\mathbb{R}^3$. This line will be observed as a line on the *right image*.
P can be determined based on p_1 and p_2. We assume that the pixels p_1 and p_2, corresponding to P, have similar visual appearance.

Rectified images *

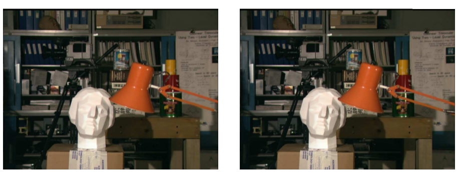
Suppose that we are given two cameras looking at *parallel direction*. Let C_{left} be the origin of the coordinate system and assume that the *image planes are co-planar* and parallel to the x and y axis.



The intersection of the triangle $\triangle(C_{\text{left}}, P, C_{\text{right}})$ and the plane including the images planes is the segment $\overline{p_1 p_2}$. Therefore $\overline{p_1 p_2}$ is parallel to the x-axis.
For more details you may refer to the course on **Computer Vision II: Multiple View Geometry**.

Stereo matching

The goal is to reconstruct 3D points according to corresponding pixels.
We assume **rectified images**, which means that the corresponding pixels are situated in **horizontal lines** according to some displacement.



Left view Right view

Therefore, we need to search for corresponding points in the same row of both views. We also assume that the pixels p_1 and p_2 corresponding to P have similar intensities.

Energy function

We define $\mathcal{L} = \{1, 2, \ldots, D\}$ as the **label set**, i.e. set of possible *horizontal displacement* of pixels on the *right view*), where D is a constant.
Therefore the output domain $\mathcal{Y} = \mathcal{L}^{\mathcal{V}}$ and the *energy function* has the following form

$$E(\mathbf{y}; \mathbf{x}) = \sum_{i \in \mathcal{V}} E_i(y_i; \mathbf{x}) + \sum_{(i,j)\in\mathcal{E}} E_{ij}(y_i, y_j; \mathbf{x}) \ ,$$

where $\mathbf{x}$ consists of the images (i.e. left and right view) denoted by $\mathbf{x}^{\text{left}}$ and $\mathbf{x}^{\text{right}}$, respectively.

Unary energies (a.k.a. **data terms**) E_i for all $i \in \mathcal{V}$ are defined as

$$E_i(y_i; \mathbf{x}) = \min(|x_i^{\text{left}} - x_{i+y_i}^{\text{right}}|^2, K)) \ ,$$

where K is a constant (e.g., $K = 20^2$).

Energy function

Pairwise energies (a.k.a. **smooth terms**) E_{ij} for all $(i,j) \in \mathcal{E}$ are defined as

$$E_{ij}(y_i, y_j; \mathbf{x}) = U(|x_i^{\text{left}} - x_j^{\text{left}}|) \cdot [\![y_i \neq y_j]\!] \, ,$$

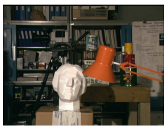
where

$$U(|x_i^{\text{left}} - x_j^{\text{left}}|) = \begin{cases} 2C, & \text{if } |x_i^{\text{left}} - x_j^{\text{left}}| \leqslant 5 \\ C, & \text{otherwise} \end{cases}$$

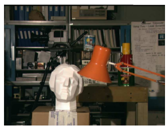
for some constant C.

Note the pairwise energies are defined by **weighted Potts-model**, which is a metric.

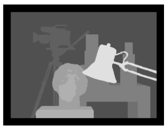
Results *



Left view



Right view



Ground truth

Result of $\alpha - \beta$ swap

Result of α-expansion

It is worth noting that α-expansion algorithm generally runs faster than $\alpha - \beta$ swap. There is optimality guarantee only for α-expansion algorithm, however, the two algorithms perform almost the same in many practical applications.

Summary *

- A binary energy function E consisting of up to pairwise functions is **regular**, if for each term E_{ij} for all $i < j$ satisfies

$$E_{ij}(0,0) + E_{ij}(1,1) \leqslant E_{ij}(0,1) + E_{ij}(1,0) \, .$$

- The *minimization of regular energy functions* can be achieved via *graph cut*.
- The *multi-label problem* for a finite label set $\mathcal{L}$

$$E(\mathbf{y};\mathbf{x}) = \sum_{i \in \mathcal{V}} E_i(y_i;\mathbf{x}) + \sum_{(i,j) \in \mathcal{E}} E_{ij}(y_i, y_j; \mathbf{x}) \, ,$$

can be approximately solved by applying

  - $\alpha - \beta$ `swap`, if E_{ij} is semi-metric;
  - α`-expansion`, if E_{ij} is metric.

Equivalent integer linear program

We are generally interested to find a *MAP labelling* $\mathbf{x}^*$:

$$\mathbf{x}^* \in \operatorname*{argmin}_{\mathbf{x} \in \mathcal{L}^{|\mathcal{V}|}} E(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{x} \in \mathcal{L}^{|\mathcal{V}|}} \left\{ \sum_{i \in \mathcal{V}} E_i(x_i) + \sum_{(i,j) \in \mathcal{E}} w_{ij} \cdot d(x_i, x_j) \right\} .$$

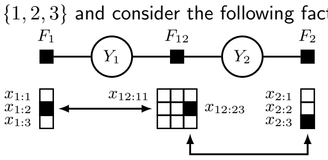
This can be equivalently written as an **integer linear program** (ILP):

$$\begin{aligned} \min_{x_{i:\alpha}, x_{ij:\alpha\beta}} \; & \sum_{i \in \mathcal{V}} \sum_{\alpha \in \mathcal{L}} E_i(\alpha) x_{i:\alpha} + \sum_{(i,j) \in \mathcal{E}} w_{ij} \sum_{\alpha,\beta \in \mathcal{L}} d(\alpha,\beta) x_{ij:\alpha\beta} \\ \text{subject to} \; & \textstyle\sum_{\alpha \in \mathcal{L}} x_{i:\alpha} = 1 && \forall i \in \mathcal{V} \\ & \textstyle\sum_{\alpha \in \mathcal{L}} x_{ij:\alpha\beta} = x_{j:\beta} && \forall \beta \in \mathcal{L}, (i,j) \in \mathcal{E} \\ & \textstyle\sum_{\beta \in \mathcal{L}} x_{ij:\alpha\beta} = x_{i:\alpha} && \forall \alpha \in \mathcal{L}, (i,j) \in \mathcal{E} \\ & x_{i:\alpha}, x_{ij:\alpha\beta} \in \mathbb{B} && \forall \alpha, \beta \in \mathcal{L}, (i,j) \in \mathcal{E} \end{aligned}$$

$x_{i:\alpha}$ indicates whether vertex i is assigned label α, while $x_{ij:\alpha\beta}$ indicates whether (neighboring) vertices i,j are assigned labels α, β, respectively.

Interpretation of the constraints

Let us assume that $\mathcal{L} = \{1,2,3\}$ and consider the following factor graph example:



Uniqueness: The constraints $\sum_{\alpha \in \mathcal{L}} x_{i:\alpha} = 1$ for all $i \in \mathcal{V}$ simply express the fact that each vertex must receive exactly one label.

Consistency: The constraints

$$\sum_{\alpha \in \mathcal{L}} x_{ij:\alpha\beta} = x_{j:\beta} \quad \text{and} \quad \sum_{\beta \in \mathcal{L}} x_{ij:\alpha\beta} = x_{i:\alpha} \quad \forall \alpha, \beta \in \mathcal{L}\, , (i,j) \in \mathcal{E}$$

maintain consistency between variables, i.e. if $x_{i:\alpha} = 1$ and $x_{j:\beta} = 1$ holds true, then these constraints force $x_{ij:\alpha\beta} = 1$ to hold true as well.

Primal-dual LP

LP relaxation *

The ILP defined before is in general NP-hard. Therefore we deal with the **LP relaxation** of our ILP. The relaxed LP can be written in *standard form* as follows:

$$\min_{x_{i:\alpha}, x_{ij:\alpha\beta}} \langle \mathbf{c}, \mathbf{x} \rangle$$

$$\text{subject to } \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geqslant \mathbf{0} \, .$$

LP relaxation: cost function *

$$\min_{x_{i:\alpha}, x_{ij:\alpha\beta}} \langle \mathbf{c}, \mathbf{x} \rangle \quad \text{subject to } \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geqslant \mathbf{0} \, .$$

We may write $\mathbf{x} = \begin{bmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T \end{bmatrix}^T$, where

$$\mathbf{x}_1 = \begin{bmatrix} x_{1:1} & \cdots & x_{1:3} & x_{2:1} & \cdots & x_{2:3} \end{bmatrix}^T \in \mathbb{R}^{mn} \, ,$$

where $n = |\mathcal{V}|$ and $m = |\mathcal{L}|$, and

$$\mathbf{x}_2 = \begin{bmatrix} x_{12:11} & \cdots & x_{12:13} & \cdots & x_{12:31} & \cdots & x_{12:33} \end{bmatrix}^T \in \mathbb{R}^{|\mathcal{E}|m^2} \, .$$

Similarly, we can write $\mathbf{c} = \begin{bmatrix} \mathbf{c}_1^T & \mathbf{c}_2^T \end{bmatrix}^T$, where

$$\mathbf{c}_1 = \begin{bmatrix} E_1(1) & \cdots & E_1(3) & E_2(1) & \cdots & E_2(3) \end{bmatrix}^T \in \mathbb{R}^{mn}$$

$$\mathbf{c}_2 = \begin{bmatrix} w_{12}d(1,1) & \cdots & w_{12}d(1,3) & \cdots & w_{12}d(3,1) & \cdots & w_{12}d(3,3) \end{bmatrix}^T \in \mathbb{R}^{|\mathcal{E}|m^2} .$$

Therefore, $\langle \mathbf{c}, \mathbf{x} \rangle = \langle \mathbf{c}_1, \mathbf{x}_1 \rangle + \langle \mathbf{c}_2, \mathbf{x}_2 \rangle$.

LP relaxation: uniqueness constraints *

$$\min_{x_{i:\alpha}, x_{ij:\alpha\beta}} \langle \mathbf{c}, \mathbf{x} \rangle \quad \text{subject to } \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geqslant \mathbf{0} \,.$$

We can write the (uniqueness) constraints $\sum_{\alpha \in \mathcal{L}} x_{i:\alpha} = 1$ for all $i \in \mathcal{V}$ as

$$\underbrace{\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}}_{\mathbf{A}_{11}} \begin{bmatrix} x_{1:1} \\ \vdots \\ x_{2:3} \end{bmatrix} = \mathbf{A}_{11}\mathbf{x}_1 = \mathbf{1}_n =: \mathbf{b}_1 \,,$$

where $\mathbf{1}_n \in \mathbb{R}^n$ is the vector of all-ones.

LP relaxation: consistency constraints *

$$\min_{x_{i:\alpha}, x_{ij:\alpha\beta}} \langle \mathbf{c}, \mathbf{x} \rangle \quad \text{subject to } \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geqslant \mathbf{0} \,.$$

The (consistency) constraints $\sum_{\alpha \in \mathcal{L}} x_{ij:\alpha\beta} = x_{j:\beta} \Leftrightarrow -x_{j:\beta} + \sum_{\alpha \in \mathcal{L}} x_{ij:\alpha\beta} = 0$ and $\sum_{\beta \in \mathcal{L}} x_{ij:\alpha\beta} = x_{i:\alpha} \Leftrightarrow -x_{i:\alpha} + \sum_{\beta \in \mathcal{L}} x_{ij:\alpha\beta} = 0$ can be expressed as

$$\left[\begin{array}{cccccc|ccccccccc} 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ \hline -1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right] \begin{bmatrix} x_{1:1} \\ \vdots \\ x_{2:3} \\ x_{12:11} \\ \vdots \\ x_{12:33} \end{bmatrix} = \mathbf{0},$$

$$\left[\begin{array}{c|c} \mathbf{A}_{21} & \mathbf{A}_{22} \end{array}\right] \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \mathbf{0}_{2|\mathcal{E}|m} =: \mathbf{b}_2 \,.$$

LP relaxation: constraints *

$$\min_{x_{i:\alpha}, x_{ij:\alpha\beta}} \langle \mathbf{c}, \mathbf{x} \rangle \quad \text{subject to } \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geqslant \mathbf{0} \,.$$

We can write all the constraints in a matrix-vector notation as follows.

$$\mathbf{A}\mathbf{x} = \left[\begin{array}{c|c} \mathbf{A}_{11} & \mathbf{0}_{n \times |\mathcal{E}|m^2} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{array}\right] \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{1}_n \\ \mathbf{0}_{2|\mathcal{E}|m} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} = \mathbf{b} \,.$$

Hence, $\mathbf{A} \in \mathbb{R}^{n+2|\mathcal{E}|m \times mn + |\mathcal{E}|m^2}$ is a **sparse matrix** with elements -1,0 and 1, furthermore $\mathbf{b} \in \mathbb{R}^{n+2|\mathcal{E}|m}$, where the first mn elements are equal to one and the others are equal to zero.

Primal-dual LP

Consider a linear program (given in **standard form**):

$$\min_{\mathbf{x} \in \mathbb{R}^n} \langle \mathbf{c}, \mathbf{x} \rangle$$
$$\text{subject to } \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geqslant \mathbf{0} \,,$$

for a *constraint matrix* $\mathbf{A} \in \mathbb{R}^{m \times n}$, a *constraint vector* $\mathbf{b} \in \mathbb{R}^m$ and a *cost vector* $\mathbf{c} \in \mathbb{R}^n$.

The *dual LP* is defined as

$$\max_{\mathbf{y} \in \mathbb{R}^m} \langle \mathbf{b}, \mathbf{y} \rangle$$
$$\text{subject to } \mathbf{A}^T\mathbf{y} \leqslant \mathbf{c} \,.$$

For feasible solutions $\mathbf{x}$ and $\mathbf{y}$ **weak duality** holds:

$$\langle \mathbf{b}, \mathbf{y} \rangle = \mathbf{b}^T\mathbf{y} = \mathbf{x}^T(\mathbf{A}^T\mathbf{y}) = (\mathbf{y}^T\mathbf{A})\mathbf{x} \leqslant \mathbf{c}^T\mathbf{x} = \langle \mathbf{c}, \mathbf{x} \rangle \,.$$

Dual LP

$$\max_{y_i, y_{ij:\alpha}, y_{ji:\beta}} \langle \mathbf{b}, \mathbf{y} \rangle \quad \text{subject to } \mathbf{A}^T\mathbf{y} \leqslant \mathbf{c} \,.$$

Note that the dual variables y_i for all $i \in \mathcal{V}$ and $y_{ij:\alpha}$, $y_{ji:\beta}$ for all $(i,j) \in \mathcal{E}$, $\alpha, \beta \in \mathcal{L}$ correspond to the constraints of the primal LP.

We can write $\mathbf{y} = \begin{bmatrix} \mathbf{y}_1^T & \mathbf{y}_2^T & \mathbf{y}_3^T \end{bmatrix}^T$, where $\mathbf{y}_1 = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}^T \in \mathbb{R}^n$, and $\mathbf{y}_2 \in \mathbb{R}^{|\mathcal{E}|m}$ and $\mathbf{y}_3 \in \mathbb{R}^{|\mathcal{E}|m}$ are the vectors consisting of the variables $y_{ji:\beta}$ and $y_{ij:\alpha}$ in the same order as it is defined in the case of the primal LP.

The cost function results in

$$\langle \mathbf{b}, \mathbf{y} \rangle = \langle \mathbf{b}_1, \mathbf{y}_1 \rangle + \langle \mathbf{b}_2, \begin{bmatrix} \mathbf{y}_2^T & \mathbf{y}_3^T \end{bmatrix}^T \rangle = \langle \mathbf{1}_n, \mathbf{y}_1 \rangle = \sum_{i=1}^n y_i \,.$$

The constraints $\mathbf{A}^T\mathbf{y} \leqslant \mathbf{c}$ are given by

$$\mathbf{A}^T\mathbf{y} = \left[\begin{array}{c|c} \mathbf{A}_{11}^T & \mathbf{A}_{21}^T \\ \hline \mathbf{0}_{|\mathcal{E}|m^2 \times n} & \mathbf{A}_{22}^T \end{array}\right] \mathbf{y} \leqslant \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} = \mathbf{c} \,.$$

Dual LP *

$$\max_{y_i, y_{ij:\alpha}, y_{ji:\beta}} \langle \mathbf{1}_n, \mathbf{y}_1 \rangle$$
$$\text{subject to } \left[\begin{array}{c|c} \mathbf{A}_{11}^T & \mathbf{A}_{21}^T \\ \hline \mathbf{0}_{|\mathcal{E}|m^2 \times n} & \mathbf{A}_{22}^T \end{array}\right] \mathbf{y} \leqslant \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} \,.$$

Or equivalently, we can formulate the dual LP as

$$\max_{y_i, y_{ij:\alpha}, y_{ji:\beta}} \sum_{i \in \mathcal{V}} y_i$$
$$\begin{aligned} \text{subject to} \quad & y_i - \sum_{j \in \mathcal{V},\, (i,j) \in \mathcal{E}} y_{ij:\alpha} \leqslant E_i(\alpha) && \forall i \in \mathcal{V}, \alpha \in \mathcal{L} \\ & y_{ij:\alpha} + y_{ji:\beta} \leqslant w_{ij} d(\alpha, \beta) && \forall (i,j) \in \mathcal{E}, \alpha, \beta \in \mathcal{L} \end{aligned}$$

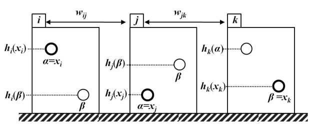
An intuitive view of the dual variables

We will refer to $x_i \in \mathcal{L}$ as the **active label** for a given the vertex $i \in \mathcal{V}$.

For each vertex we have a different copy of all labels in $\mathcal{L}$. It is assumed that all these labels represent **balls** floating at certain heights relative to a *reference plane*.

For this sake we introduce **height variables** defined as

$$h_i(\alpha) \triangleq E_i(\alpha) + \sum_{j \in \mathcal{V}, (i,j) \in \mathcal{E}} y_{ij:\alpha} \,.$$



The constraints $y_i - \sum_{j \in \mathcal{V}:(i,j) \in \mathcal{E}} y_{ij:\alpha} \leqslant E_i(\alpha)$ can be equivalently written as

$$y_i \leqslant E_i(\alpha) + \sum_{j \in \mathcal{V}:(i,j) \in \mathcal{E}} y_{ij:\alpha} = h_i(\alpha) \qquad \forall i \in \mathcal{V}, \alpha \in \mathcal{L} \,.$$

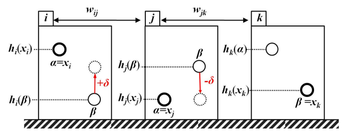
Since our objective is to maximize $\sum_{i \in \mathcal{V}} y_i$, the following relation holds

$$y_i = \min_{\alpha \in \mathcal{L}} h_i(\alpha) \qquad \forall i \in \mathcal{V} \,.$$

Balance variables and load

We will refer to the variables $y_{ij:\alpha}$, $y_{ji:\beta}$ as **balance variables**. Specially, the pair of $y_{ij:\alpha}$, $y_{ji:\alpha}$ is called **conjugate balance variables**.

The *balls* are not static, but may move in pairs through updating pairs of *conjugate balance variables* as $h_i(\alpha) = E_i(\alpha) + \sum_{j \in \mathcal{V}, (i,j) \in \mathcal{E}} y_{ij:\alpha}$. Therefore, the role of *balance variables* is to raise or lower labels.



It is due to $y_{ij:\alpha} + y_{ji:\alpha} \leqslant w_{ij} d(\alpha, \alpha) = 0 \quad \Rightarrow \quad y_{ij:\alpha} \leqslant -y_{ji:\alpha}$.

We will call the variables $y_{ij:x_i}$ as **active balance variable** and use the following notation for the **"load"** between neighbors i, j, defined as

$$\text{load}_{ij} = y_{ij:x_i} + y_{ji:x_j} \,.$$

Primal-dual LP for multi-label problem

The (relaxed) primal LP:

$$\min_{x_{i:\alpha}, x_{ij:\alpha\beta} \geqslant 0} \sum_{i \in \mathcal{V}} \sum_{\alpha \in \mathcal{L}} E_i(\alpha) x_{i:\alpha} + \sum_{(i,j) \in \mathcal{E}} w_{ij} \sum_{\alpha,\beta \in \mathcal{L}} d(\alpha, \beta) x_{ij:\alpha\beta}$$

subject to

$$\begin{aligned} \textstyle\sum_{\alpha \in \mathcal{L}} x_{i:\alpha} &= 1 && \forall i \in \mathcal{V} \\ \textstyle\sum_{\alpha \in \mathcal{L}} x_{ij:\alpha\beta} &= x_{j:\beta} && \forall \beta \in \mathcal{L}, (i,j) \in \mathcal{E} \\ \textstyle\sum_{\beta \in \mathcal{L}} x_{ij:\alpha\beta} &= x_{i:\alpha} && \forall \alpha \in \mathcal{L}, (i,j) \in \mathcal{E} \end{aligned}$$

The dual LP:

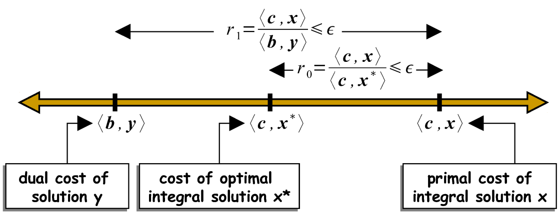
$$\max_{y_i, y_{ij:\alpha}, y_{ji:\beta}} \sum_{i \in \mathcal{V}} y_i$$

subject to

$$\begin{aligned} y_i - \sum_{j \in \mathcal{V}:(i,j) \in \mathcal{E}} y_{ij:\alpha} &\leqslant E_i(\alpha) && \forall i \in \mathcal{V}, \alpha \in \mathcal{L} \\ y_{ij:\alpha} + y_{ji:\beta} &\leqslant w_{ij} d(\alpha, \beta) && \forall (i,j) \in \mathcal{E}, \alpha, \beta \in \mathcal{L} \end{aligned}$$

Primal-dual principle

Primal-dual principle



Theorem 2. *If* $\mathbf{x}$ *and* $\mathbf{y}$ *are integral-primal and dual feasible solutions satisfying:*

$$\langle \mathbf{c}, \mathbf{x} \rangle \leqslant \epsilon \langle \mathbf{b}, \mathbf{y} \rangle$$

for $\epsilon \geqslant 1$*, then* $\mathbf{x}$ *is an* ϵ**-approximation** *to the optimal integral solution* $\mathbf{x}^*$*, that is*

$$\langle \mathbf{c}, \mathbf{x}^* \rangle \leqslant \langle \mathbf{c}, \mathbf{x} \rangle \leqslant \epsilon \langle \mathbf{b}, \mathbf{y} \rangle \leqslant \epsilon \langle \mathbf{c}, \mathbf{x}^* \rangle .$$

The relaxed complementary slackness

One way to estimate a pair $(\mathbf{x}, \mathbf{y})$ satisfying the fundamental inequality

$$\langle \mathbf{c}, \mathbf{x} \rangle \leqslant \epsilon \langle \mathbf{b}, \mathbf{y} \rangle$$

relies on the **complementary slackness principle**.

Theorem 3. *If the pair* $(\mathbf{x}, \mathbf{y})$ *of integral-primal and dual feasible solutions satisfies the so-called* **relaxed primal complementary slackness conditions***:*

$$\forall j : \ (x_j > 0) \quad \Rightarrow \quad \sum_i a_{ij} y_i \geqslant \frac{c_j}{\epsilon_j} ,$$

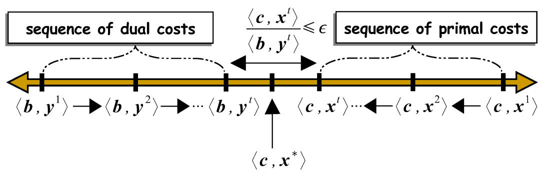
then $(\mathbf{x}, \mathbf{y})$ *also satisfies* $\langle \mathbf{c}, \mathbf{x} \rangle \leqslant \epsilon \langle \mathbf{b}, \mathbf{y} \rangle$ *with* $\epsilon = \max_j \epsilon_j$ *and therefore* $\mathbf{x}$ *is an* ϵ*-approximation to the optimal integral solution* $\mathbf{x}^*$*.*

Proof. Exercise. □

We aim to satisfy relaxed complementary slackness conditions in order to achieve an ϵ-approximation solution.

Primal-dual schema

Primal-dual schema



Typically, primal-dual ϵ-approximation algorithms construct a sequence $(\mathbf{x}^k, \mathbf{y}^k)_{k=1,\ldots,t}$ of primal and dual solutions until the elements $\mathbf{x}^t$, $\mathbf{y}^t$ of the last pair are both **feasible** and **satisfy the relaxed primal complementary slackness conditions**, hence the condition $\langle \mathbf{c}, \mathbf{x} \rangle \leqslant \epsilon \langle \mathbf{b}, \mathbf{y} \rangle$ will be also fulfilled.

Summary *

We have learned about primal-dual linear programming relaxation for the multi-labeling problem.

In the **next lecture** we will learn about the *Fast primal-dual algorithm* for the multi-labeling problem.

Literature *

Move-making algorithms

1. Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001
2. Sebastian Nowozin and Christoph H. Lampert. Structured prediction and learning in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3–4), 2010

Primal-dual schema

1. Nikos Komodakis and Georgios Tziritas. Approximate labeling via the primal-dual schema. Technical report, University of Crete, February 2005
2. Nikos Komodakis and Georgios Tziritas. Approximate labeling via graph-cuts based on linear programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1436–1453, August 2007