6. Primal-dual Schema ................................................................. 2
Agenda for today’s lecture * ....................................................... 3

α-expansion  ................................................................. 4
  α-expansion ............................................................... 5
Local optimization ............................................................. 6
Graph construction ............................................................ 7
Graph construction: t-links .................................................... 8
Graph construction: n-links .................................................... 9
α-expansion algorithm * ....................................................... 10
Optimality * ................................................................. 11

Stereo matching .............................................................. 12
Stereo matching * ............................................................ 13
Rectified images * ............................................................ 14
Stereo matching ............................................................... 15
Energy function ............................................................... 16
Energy function ............................................................... 17
Results * ................................................................. 18
Summary * ................................................................. 19
Equivalent integer linear program .................................................. 20
Interpretation of the constraints ..................................................... 21

Primal-dual LP 22
 LP relaxation * .............................................................................. 23
 LP relaxation: cost function * .......................................................... 24
 LP relaxation: uniqueness constraints * ............................................. 25
 LP relaxation: consistency constraints * ............................................ 26
 LP relaxation: constraints * ............................................................ 27
 Primal-dual LP ................................................................................. 28
 Dual LP .......................................................................................... 29
 Dual LP * ........................................................................................ 30
 An intuitive view of the dual variables ............................................. 31
 Balance variables and load ............................................................... 32
 Primal-dual LP for multi-label problem .......................................... 33

Primal-dual principle 34
 Primal-dual principle ....................................................................... 35
 The relaxed complementary slackness ............................................ 36

Primal-dual schema 37
 Primal-dual schema ................................................................. 38
 Summary * .................................................................................. 39
 Literature * .................................................................................. 40
Agenda for today’s lecture *

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

$$E(x) = \sum_{i \in V} E_i(x_i) + \sum_{(i,j) \in E} w_{ij} \cdot d(x_i, x_j),$$

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
- $\alpha$-expansion, which provides an approximate solution, and
- the linear programming formalization of the multi-labeling problem.
\(\alpha\)-expansion

\(\alpha\)-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

\[
Z_\alpha(y, \alpha) = \{z \in \mathcal{Y} : z_i \in \{y_i, \alpha\} \text{ for all } i \in \mathcal{V}\}.
\]

The minimization of the energy function \(E\) can be reformulated as follows:

\[
\hat{z} \in \arg\min_{z \in Z_\alpha(y, \alpha)} E(z) = \arg\min_{z \in Z_\alpha(y, \alpha)} \sum_{i \in \mathcal{V}} E_i(z_i) + \sum_{(i,j) \in \mathcal{E}} E_{ij}(z_i, z_j)
\]

\[
= \arg\min_{z \in Z_\alpha(y, \alpha)} \left[ \sum_{i \in \mathcal{V}, y_i = \alpha} E_i(\alpha) + \sum_{i \in \mathcal{V}, y_i \neq \alpha} E_i(z_i) \right.
\]

\[
\left. + \sum_{(i,j) \in \mathcal{E}, y_i = \alpha, y_j = \alpha} E_{ij}(\alpha, \alpha) + \sum_{(i,j) \in \mathcal{E}, y_i = \alpha, y_j \neq \alpha} E_{ij}(\alpha, \alpha) + \sum_{(i,j) \in \mathcal{E}, y_i \neq \alpha, y_j = \alpha} E_{ij}(\alpha, \alpha) + \sum_{(i,j) \in \mathcal{E}, y_i \neq \alpha, y_j \neq \alpha} E_{ij}(\alpha, z_j) \right].
\]
Local optimization

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

<table>
<thead>
<tr>
<th>$E_{ij}$</th>
<th>$\alpha$</th>
<th>$y_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>$E_{ij}(\alpha, \alpha)$</td>
<td>$E_{ij}(\alpha, y_j)$</td>
</tr>
<tr>
<td>$y_i$</td>
<td>$E_{ij}(y_i, \alpha)$</td>
<td>$E_{ij}(y_i, y_j)$</td>
</tr>
</tbody>
</table>

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

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

which means that $E_{ij}$ is regular w.r.t. the labeling $Z_\alpha(y, \alpha)$.

Graph construction

We need to minimize the following regular energy function:

$$\hat{z} \in \arg \min_{z \in \hat{Z}_\alpha(y, \alpha)} \sum_{i \in V} E_i(z_i) + \sum_{(i, j) \in \mathcal{E}} E_{ij}(\alpha, z_j) + \sum_{(i, j) \in \mathcal{E}} E_{ij}(z_i, \alpha) + \sum_{(i, j) \in \mathcal{E}} E_{ij}(z_i, z_j) .$$

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 $\mathcal{E}' = \{(\alpha, i), (i, \bar{\alpha}) : i \in \mathcal{V}' \setminus \{\alpha, \bar{\alpha}\}\} \cup \{(i, j) \in \mathcal{E} : i, j \in \mathcal{V}' \setminus \{\alpha, \bar{\alpha}\}\}$.
Graph construction: t-links

![Graph diagram]

t-links: for all $i \in \mathcal{V}\setminus \{\alpha, \bar{\alpha}\}$

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

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

Graph construction: n-links

![Graph diagram]

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

$$c(i, j) = E_{ij}(\alpha, y_j) + E_{ij}(y_i, \alpha) - E_{ij}(y_i, y_j).$$
\[ \alpha \text{-expansion algorithm } \]

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

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

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

\( \alpha \)-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 \( \alpha \)-expansion algorithm will find is at most twice the global minimum \( y^* \).

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

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

**Theorem 1.** Let \( \hat{y} \) be a local minimum when the expansion moves are allowed and \( y^* \) be the globally optimal solution. Then \( E(\hat{y}) \leq 2cE(y^*) \).
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 \( C_{\text{left}} \) 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 \( p_1p_2 \). Therefore \( p_1p_2 \) 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.

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

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

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

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

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

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

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

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

where

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

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 $\alpha$-expansion

It is worth noting that $\alpha$-expansion algorithm generally runs faster than $\alpha - \beta$ swap. There is optimality guarantee only for $\alpha$-expansion algorithm, however, the two algorithms perform almost the same in many practical applications.
A binary energy function $E$ consisting of up to pairwise functions is regular, if for each term $E_{ij}$ for all $i < j$ satisfies

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

The minimization of regular energy functions can be achieved via graph cut.

The multi-label problem for a finite label set $\mathcal{L}$

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

can be approximately solved by applying

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

Equivalent integer linear program

We are generally interested to find a MAP labelling $x^*$:

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

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

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

$x_{i;\alpha}$ indicates whether vertex $i$ is assigned label $\alpha$, while $x_{ij;\alpha\beta}$ indicates whether (neighboring) vertices $i, j$ are assigned labels $\alpha, \beta$, respectively.
Interpretation of the constraints

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

![Factor Graph Example](image)

**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:

$$
\min_{x_{i: \alpha}, x_{ij: \alpha \beta}} \langle c, x \rangle \\
\text{subject to } Ax = b, x \geq 0.
$$
LP relaxation: cost function *

\[
\min_{x_{i,a},x_{j,a}} \langle c, x \rangle \quad \text{subject to } Ax = b, x \geq 0. 
\]

We may write \( x = [x_1^T \ x_2^T]^T \), where

\[
x_1 = \begin{bmatrix}
  x_{1:1} & \cdots & x_{1:3} & x_{2:1} & \cdots & x_{2:3}
\end{bmatrix}^T \in \mathbb{R}^{mn},
\]

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

\[
x_2 = \begin{bmatrix}
  x_{12:11} & \cdots & x_{12:13} & x_{12:31} & \cdots & x_{12:33}
\end{bmatrix}^T \in \mathbb{R}^{|\mathcal{E}|m^2}.
\]

Similarly, we can write \( c = [c_1^T \ c_2^T]^T \), where

\[
c_1 = \begin{bmatrix}
  E_1(1) & \cdots & E_1(3) & E_2(1) & \cdots & E_2(3)
\end{bmatrix}^T \in \mathbb{R}^{mn}
\]

\[
c_2 = \begin{bmatrix}
  w_{12}d(1,1) & \cdots & w_{12}d(1,3) & \cdots & w_{12}d(3,1) & \cdots & w_{12}d(3,3)
\end{bmatrix}^T \in \mathbb{R}^{|\mathcal{E}|m^2}.
\]

Therefore, \( \langle c, x \rangle = \langle c_1, x_1 \rangle + \langle c_2, x_2 \rangle \).

LP relaxation: uniqueness constraints *

\[
\min_{x_{i,a},x_{j,a}} \langle c, x \rangle \quad \text{subject to } Ax = b, x \geq 0. 
\]

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

\[
\begin{bmatrix}
  1 & 1 & 1 & 0 & 0 & 0 \\
  0 & 0 & 0 & 1 & 1 & 1 \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  A_{11}
\end{bmatrix}
\begin{bmatrix}
  x_{1:1} \\
  \vdots \\
  x_{2:3}
\end{bmatrix}
= A_{11} x_1 = 1_n = b_1,
\]

where \( 1_n \in \mathbb{R}^n \) is the vector of all-ones.
LP relaxation: consistency constraints *  

\[
\min_{x_{i\alpha}, x_{ij\alpha\beta}} \langle c, x \rangle \quad \text{subject to } A x = b, x \geq 0 .
\]

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

\[
\begin{bmatrix}
0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
x_{1:1} \\
x_{2:3} \\
x_{12:11} \\
\vdots \\
x_{12:33} \\
\end{bmatrix}
= 0,
\]

\[
\begin{bmatrix}
A_{21} & A_{22}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\end{bmatrix}
= 0_{2|\mathcal{E}|m} =: b_2 .
\]

LP relaxation: constraints *

\[
\min_{x_{i\alpha}, x_{ij\alpha\beta}} \langle c, x \rangle \quad \text{subject to } A x = b, x \geq 0 .
\]

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

\[
A x = \begin{bmatrix}
A_{11} & 0_{n \times |\mathcal{E}|m^2}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\end{bmatrix}
= \begin{bmatrix}
1_n \\
0_{|\mathcal{E}|m}
\end{bmatrix}
= \begin{bmatrix}
b_1 \\
b_2
\end{bmatrix}
= b .
\]

Hence, \( A \in \mathbb{R}^{n+2|\mathcal{E}|m \times mn + |\mathcal{E}|m^2} \) is a sparse matrix with elements -1, 0 and 1, furthermore \( b \in \mathbb{R}^{n+2|\mathcal{E}|m} \), where the first \( mn \) elements are equal to one and the others are equal to zero.
Consider a linear program (given in \textbf{standard form}):

\[
\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad \langle c, x \rangle \\
\text{subject to} & \quad Ax = b, x \geq 0 ,
\end{align*}
\]

for a \textit{constraint matrix} \( A \in \mathbb{R}^{m \times n} \), a \textit{constraint vector} \( b \in \mathbb{R}^m \) and a \textit{cost vector} \( c \in \mathbb{R}^n \).

The \textit{dual LP} is defined as

\[
\begin{align*}
\max_{y \in \mathbb{R}^m} & \quad \langle b, y \rangle \\
\text{subject to} & \quad A^T y \leq c .
\end{align*}
\]

For feasible solutions \( x \) and \( y \) \textbf{weak duality} holds:

\[
\langle b, y \rangle = b^T y = x^T (A^T y) = (y^T A)x \leq c^T x = \langle c, x \rangle .
\]
Dual LP

\[
\max_{y_i, y_{ij}, \alpha, \beta} \langle b, y \rangle \quad \text{subject to } A^T y \leq c.
\]

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

We can write \( y = [y_1^T \ y_2^T \ y_3^T]^T \), where \( y_1 = [y_1 \ldots y_n]^T \in \mathbb{R}^n \), and \( y_2 \in \mathbb{R}^{\mathcal{E}|m} \) and \( y_3 \in \mathbb{R}^{\mathcal{E}|m} \) 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

\[
\langle b, y \rangle = \langle b_1, y_1 \rangle + \langle b_2, [y_2^T \ y_3^T]^T \rangle = \langle 1_n, y_1 \rangle = \sum_{i=1}^{n} y_i.
\]

The constraints \( A^T y \leq c \) are given by

\[
A^T y = \begin{bmatrix} A_{11}^T & A_{21}^T \\ 0_{|\mathcal{E}|m \times n} & A_{22}^T \end{bmatrix} y \leq \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = c.
\]
Dual LP *

$$\max_{y_i, y_{ij}, y_{ji}, \alpha, \beta} \langle 1_n, y_1 \rangle$$

subject to

$$\begin{bmatrix}
A^T_{11} & A^T_{21} \\
0 & A^T_{22}
\end{bmatrix}
\begin{bmatrix} y_i \\
y_{ij} \\
y_{ji} \\
\alpha \\
\beta
\end{bmatrix} \leq \begin{bmatrix} c_1 \\
c_2
\end{bmatrix}.$$ 

Or equivalently, we can formulate the dual LP as

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

subject to

$$y_i - \sum_{j \in \mathcal{V}; (i,j) \in \mathcal{E}} y_{ij} \leq E_i(\alpha) \quad \forall i \in \mathcal{V}, \alpha \in \mathcal{L}$$

$$y_{ij} + y_{ji} \leq w_{ij}d(\alpha, \beta) \quad \forall (i,j) \in \mathcal{E}, \alpha, \beta \in \mathcal{L}$$

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

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

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

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

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

$$y_i = \min_{\alpha \in \mathcal{L}} h_i(\alpha) \quad \forall i \in \mathcal{V}.$$
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 \( h_i(\alpha) = E_i(\alpha) + \sum_{j \in \mathcal{N}_i} y_{ij}^\alpha \). Therefore, the role of balance variables is to raise or lower labels.

It is due to \( y_{ij}^\alpha + y_{ji}^\alpha \leq w_{ij} d(\alpha, \alpha) = 0 \implies y_{ij}^\alpha \leq -y_{ji}^\alpha \).

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:

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

subject to

$$\sum_{\alpha \in \mathcal{L}} x_{i\alpha} = 1 \quad \forall i \in \mathcal{V}$$

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

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

The dual LP:

$$\max_{y_i, y_{ij: \alpha \beta} \geq 0} \sum_{i \in \mathcal{V}} y_i$$

subject to

$$y_i - \sum_{j \in \mathcal{V} : (i,j) \in \mathcal{E}} y_{ij: \alpha} \leq E_i(\alpha) \quad \forall i \in \mathcal{V}, \alpha \in \mathcal{L}$$

$$y_{ij: \alpha} + y_{ji: \beta} \leq w_{ij} d(\alpha, \beta) \quad \forall (i,j) \in \mathcal{E}, \alpha, \beta \in \mathcal{L}$$
Theorem 2. If $x$ and $y$ are integral-primal and dual feasible solutions satisfying:

$$
\langle c, x \rangle \leq \epsilon \langle b, y \rangle
$$

for $\epsilon \geq 1$, then $x$ is an $\epsilon$-approximation to the optimal integral solution $x^*$, that is

$$
\langle c, x^* \rangle \leq \langle c, x \rangle \leq \epsilon \langle b, y \rangle \leq \epsilon \langle c, x^* \rangle .
$$
The relaxed complementary slackness

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

\[
\langle c, x \rangle \leq \epsilon \langle b, y \rangle
\]

relies on the **complementary slackness principle**.

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

\[
\forall j : (x_j > 0) \Rightarrow \sum_i a_{ij} y_i \geq \frac{c_j}{\epsilon_j},
\]

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

**Proof.** Exercise. \(\square\)

We aim to satisfy relaxed complementary slackness conditions in order to achieve an \(\epsilon\)-approximation solution.
Typically, primal-dual $\epsilon$-approximation algorithms construct a sequence $(x^k, y^k)_{k=1,...,t}$ of primal and dual solutions until the elements $x^t$, $y^t$ of the last pair are both feasible and satisfy the relaxed primal complementary slackness conditions, hence the condition $\langle c, x \rangle \leq \epsilon \langle b, y \rangle$ 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

Primal-dual schema