Probabilistic Graphical Models in Computer Vision (IN2329)

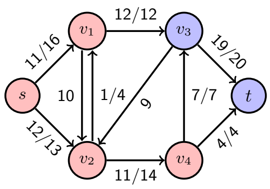
Csaba Domokos

Summer Semester 2017

5. Move making algorithms

Agenda for today's lecture *

In the **previous lecture** we learnt about the minimum $s-t$ cut problem

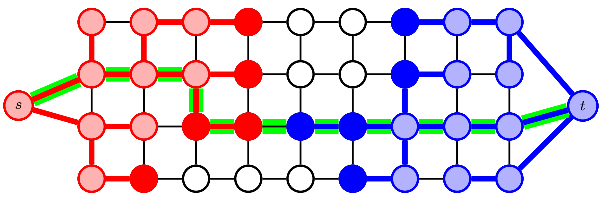


Today we are going to learn about

- The *Boykov–Kolmogorov algorithm*
- Exact solution for **binary image segmentation** via *graph cut*
- Approximate solution for the **multi-label problem** via $\alpha-\beta$ swap

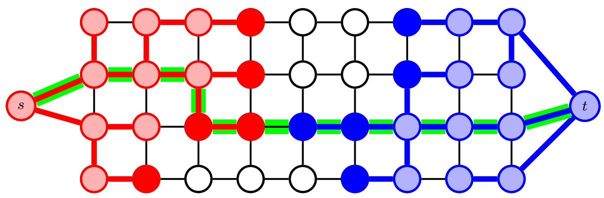
Boykov–Kolmogorov algorithm

Boykov–Kolmogorov algorithm



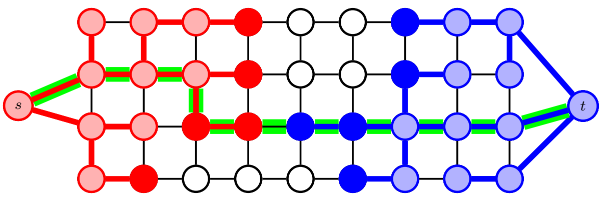
- Main idea: Never start building an *augmenting path* from scratch
- Two non-overlapping search trees S and T with roots at the terminals
- The edges of the trees are *non-saturated*, i.e. $f(i,j) < c(i,j)$
- Active nodes:
- Passive nodes:
- Free nodes:

Boykov–Kolmogorov algorithm



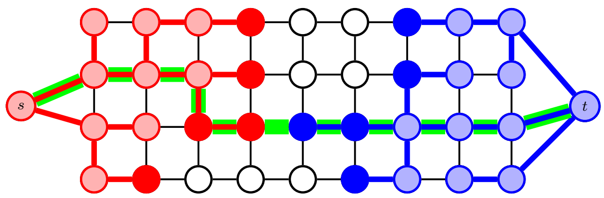

```
while true do
    grow S or T to find an augmenting path P from s to t
    if P = ∅ then
        terminate
    end if
    augment on P
    adopt orphans
end while
```


Growth stage



- The *active nodes* explore adjacent edges and acquire new children from a set of *free nodes*
- The newly acquired nodes become *active* members of the corresponding search trees
- The *active node* becomes *passive*, when all of its neighbors are explored
- If an *active node* encounters a neighboring node belonging to the opposite tree, the growth stage terminates

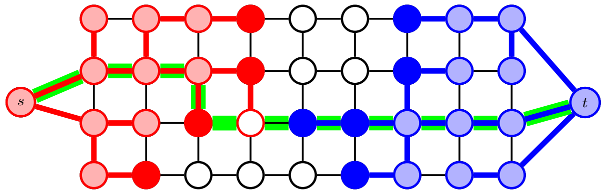
Augmentation stage



- Find the bottleneck capacity Δ on P
- Update the residual graph by pushing flow Δ through P

Adoption stage

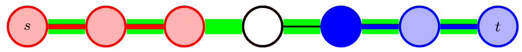
- *Orphan* ( ): the nodes such that the edges linking them to their parents are no longer valid (i.e. they are saturated)
- By removing them the search trees S and T may be split into *forests*



We are trying to find a *new valid parent* for p among its neighbors, such that a new parent should belong to the same set, S or T, as the *orphan*

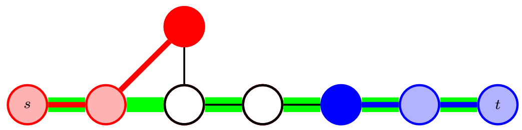
Adoption stage

If an orphan p does not find a valid parent then it becomes a *free node*



Scan all neighbors q of p such that q belong to the same tree as p:

- if tree $c(q,p) > 0$, add q to the *active set*
- if $\mathtt{parent}(q) = p$, add q to the set of *orphans* and set $\mathtt{parent}(q) = \varnothing$



Complexity

- The *Boykov-Kolmogorov algorithm* is also an *augmented path-based method* with worst case complexity $\mathcal{O}(|\mathcal{E}| \cdot |\mathcal{V}|^2 \cdot |C|)$, where $|C|$ is the capacity of the minimum cut.
- This complexity is worse than complexities of *Edmonds–Karp algorithm*, however, this algorithm *significantly* ($\sim$2-10$\times$) outperforms standard algorithms on typical problem instances in vision.

Binary image segmentation

Regular functions *

Let us consider a function f of two binary variables, then f is called **regular**, if it satisfies the following inequality

$$f(0,0) + f(1,1) \leqslant f(0,1) + f(1,0) .$$

Example: the **Potts-model** is *regular*, since

$$[\![0 \neq 0]\!] + [\![1 \neq 1]\!] = 0 \leqslant 2 = [\![0 \neq 1]\!] + [\![1 \neq 0]\!] .$$

Regular energy functions

Let us consider an *energy function* E of n binary variables which can be written as the sum of functions of up to two variables, that is $E : \mathbb{B}^n \to \mathbb{R}$

$$E(y_1, \dots, y_n) = \sum_i E_i(y_i) + \sum_{i<j} E_{ij}(y_i, y_j) .$$

E is *regular*, if each term $E_{ij} : \mathbb{B}^2 \to \mathbb{R}$ for all $i < j$ satisfies

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

If each term E_{ij} is *regular*, then it is possible to find the **global** minimum of E in *polynomial time* by solving a *minimum $s-t$ cut problem*.

Binary image segmentation

We have already seen that **binary image segmentation** can be reformulated as the minimization of an *energy function* $E : \mathbb{B}^{\mathcal{V}} \times \mathcal{X} \to \mathbb{R}$:

$$E(\mathbf{y}; \mathbf{x}) = \sum_{i \in \mathcal{V}} E_i(y_i; x_i) + \sum_{(i,j) \in \mathcal{E}} w \cdot [\![y_i \neq y_j]\!] .$$

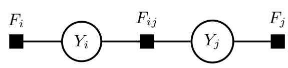
where $\mathcal{V}$ corresponds to the output variables, i.e. the pixels, and $\mathcal{E}$ includes the pairs of 4-neighboring pixels.

Assume probability densities f_{bg} and f_{fg} estimated for the background and the foreground, respectively. The **unary energies** E_i for all $i \in \mathcal{V}$ can be defined as

$$E_i(0, x_i) = 0 ,$$
$$E_i(1, x_i) = \log \frac{f_{\text{bg}}(x_i)}{f_{\text{fg}}(x_i)} .$$

Energy minimization via minimum $s-t$ cut

Let us consider the following example



Through this example we illustrate how to minimize *regular energy functions* consisting of up to pairwise relationships. In our example $\mathbf{y} \in \mathbb{B}^2$ and $E(\mathbf{y})$ is defined as

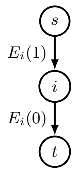
$$E(\mathbf{y}) = E_i(y_1) + E_j(y_2) + E_{ij}(y_i, y_j) .$$

We will create a *flow network* $(\mathcal{V} \cup \{s,t\}, \mathcal{E}', c, s, t)$ such that **the minimum** $s-t$ **cut will correspond to the minimization of our energy function** $E(\mathbf{y})$, where the *labeling* for each $i \in \mathcal{V}$ is defined as

$$y_i = \begin{cases} 0, & \text{if } i \in \mathcal{S} , \\ 1, & \text{if } i \in \mathcal{T} . \end{cases}$$

Graph construction: unary energies

Let us consider the unary energy function $E_i : \{0,1\} \to \mathbb{R}$.

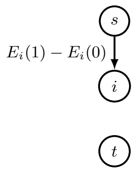


Obviously, the minimum $s-t$ cut of the flow network will correspond to

$$\operatorname*{argmin}_{y_i \in \{0,1\}} E_i(y_i) \,.$$

When $E_i(1) > E_i(0)$ holds, then we can write

$$\operatorname*{argmin}_{y_i \in \{0,1\}} E_i(y_i) = \operatorname*{argmin}_{y_i \in \{0,1\}} E_i(y_i) - E_i(0) \,.$$



Graph construction: pairwise energy

Let us consider the pairwise energy function $E_{ij}(y_i, y_j) : \mathbb{B}^2 \to \mathbb{R}$. The possible values of $E_{ij}(y_i, y_j)$ are shown in the table:

E_{ij}	$y_j = 0$	$y_j = 1$
$y_i = 0$	A	B
$y_i = 1$	C	D

We furthermore assume that $E_{ij}(y_i, y_j)$ is regular, that is

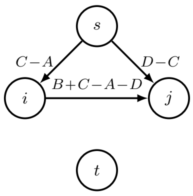
$$E_{ij}(0,0) + E_{ij}(1,1) \leqslant E_{ij}(0,1) + E_{ij}(1,0)$$
$$A + D \leqslant B + C \,.$$

Let us note that $E_{ij}(y_i, y_j)$ can be decomposed as:

$$\begin{array}{|c|c|}\hline A & B \\ \hline C & D \\ \hline\end{array} = A + \begin{array}{|c|c|}\hline 0 & B-A \\ \hline C-A & D-A \\ \hline\end{array} = A + \begin{array}{|c|c|}\hline 0 & 0 \\ \hline C-A & C-A \\ \hline\end{array} + \begin{array}{|c|c|}\hline 0 & B-A \\ \hline 0 & D-C \\ \hline\end{array}$$

$$= A + \underbrace{\begin{array}{|c|c|}\hline 0 & 0 \\ \hline C-A & C-A \\ \hline\end{array}}_{E_i(1)} + \underbrace{\begin{array}{|c|c|}\hline 0 & D-C \\ \hline 0 & D-C \\ \hline\end{array}}_{E_j(1)} + \underbrace{\begin{array}{|c|c|}\hline 0 & B+C-A-D \\ \hline 0 & 0 \\ \hline\end{array}}_{B+C-A-D \geqslant 0}$$

Graph construction: pairwise energy, $C-A \geqslant 0,\ D-C \geqslant 0$ *

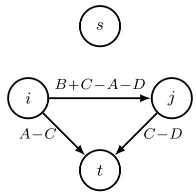


E_{ij}	$y_j = 0$	$y_j = 1$
$y_i = 0$	A	B
$y_i = 1$	C	D

Labeling: $y_i = y_j = 0$.

$$C - A \geqslant 0 \;\Rightarrow C \geqslant A \,.$$
$$D - C \geqslant 0 \;\Rightarrow D \geqslant C \;\Rightarrow\; D \geqslant A \,.$$
$$0 \leqslant B + C - A - D \leqslant B - A \;\Rightarrow B \geqslant A \,.$$

Graph construction: pairwise energy, $A-C \geqslant 0,\ C-D \geqslant 0$ *

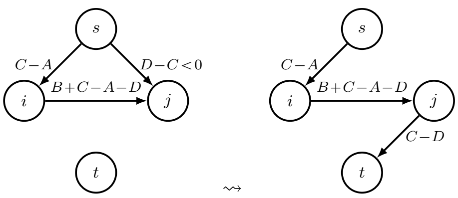


E_{ij}	$y_j = 0$	$y_j = 1$
$y_i = 0$	A	B
$y_i = 1$	C	D

Labeling: $y_i = y_j = 1$.

$$C - D \geqslant 0 \;\Rightarrow C \geqslant D \,.$$
$$A - C \geqslant 0 \;\Rightarrow A \geqslant C \;\Rightarrow\; A \geqslant D \,.$$
$$0 \leqslant B + C - A - D \leqslant B - D \;\Rightarrow B \geqslant D \,.$$

Graph construction: pairwise energy, $C-A \geqslant 0,\ C-D \geqslant 0$ *

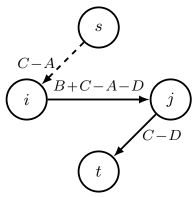


Note that the labeling $y_i = 1,\ y_j = 0$ is not possible in this case, since

$$C - A \geqslant 0 \;\Rightarrow\; C \geqslant A \,.$$

Graph construction: pairwise energy, $C-A \geqslant 0,\ C-D \geqslant 0$ *

Assume that $\min\{C-A, B+C-A-D, C-D\} = C-A$.



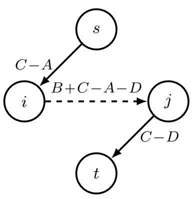
E_{ij}	$y_j = 0$	$y_j = 1$
$y_i = 0$	A	B
$y_i = 1$	C	D

Labeling: $y_i = y_j = 1$.

$$C - A \leqslant B + C - A - D \;\Rightarrow 0 \leqslant B - D \;\Rightarrow\; B \geqslant D \,.$$
$$C - A \leqslant C - D \;\Rightarrow A \geqslant D \,.$$
$$C - D \geqslant 0 \;\Rightarrow C \geqslant D \,.$$

Graph construction: pairwise energy, $C-A \geqslant 0,\ C-D \geqslant 0$ *

Assume that $\min\{C-A, B+C-A-D, C-D\} = B+C-A-D$.



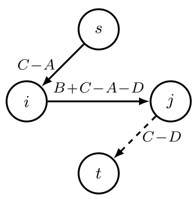
E_{ij}	$y_j = 0$	$y_j = 1$
$y_i = 0$	A	B
$y_i = 1$	C	D

Labeling: $y_i = 0,\ y_j = 1$.

$$B + C - A - D \leqslant C - A \;\Rightarrow B \leqslant D \,.$$
$$B + C - A - D \leqslant C - D \;\Rightarrow B \leqslant A \,.$$
$$C - A \geqslant 0 \;\Rightarrow A \leqslant C \;\Rightarrow\; B \leqslant C \,.$$

Graph construction: pairwise energy, $C-A \geqslant 0,\ C-D \geqslant 0$ *

Assume that $\min\{C-A, B+C-A-D, C-D\} = C-D$.

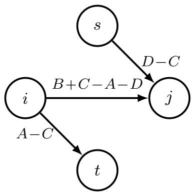


E_{ij}	$y_j = 0$	$y_j = 1$
$y_i = 0$	A	B
$y_i = 1$	C	D

Labeling: $y_i = y_j = 0$.

$$C - D \leqslant B + C - A - D \;\Rightarrow B \geqslant A \,.$$
$$C - D \leqslant C - A \;\Rightarrow D \geqslant A \,.$$
$$C - D \geqslant 0 \;\Rightarrow C \geqslant D \;\Rightarrow\; C \geqslant A \,.$$

Graph construction: pairwise energy, $A-C\geqslant 0$, $D-C\geqslant 0$ *



E_{ij}	$y_j=0$	$y_j=1$
$y_i=0$	A	B
$y_i=1$	C	D

Labeling: $y_i=1$, $y_j=0$.

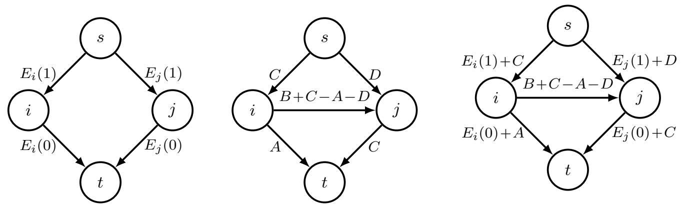
$$D-C\geqslant 0\;\Rightarrow D\geqslant C\,.$$
$$A-C\geqslant 0\;\Rightarrow A\geqslant C\,.$$
$$0\leqslant B+C-A-D\leqslant B-A\;\Rightarrow B\geqslant A\;\Rightarrow\; B\geqslant C\,.$$

All the other cases can be similarly derived.

Graph construction

Putting all together we get that

Unaries	Pairwise	Overall energy
$\operatorname{argmin}_{\mathbf{y}} E_i(y_i)+E_j(y_j)$	$\operatorname{argmin}_{\mathbf{y}} E_{ij}(y_i,y_j)$	$\operatorname{argmin}_{\mathbf{y}} E_i(y_i)+E_j(y_j)+E_{ij}(y_i,y_j)$



Remarks

Regularity is an *extremely important* property as is allows to minimize energy functions by making use of graph cut. Moreover, without the regularity constraint, the problem become intractable.

Let E_2 be a nonregular function of two binary variables. Then minimizing the energy function

$$E(y_1,\dots,y_n)=\sum_i E_i(y_i)+\sum_{i<j}E_2(y_i,y_j)\,,$$

where E_i are arbitrary functions of one binary variable, is NP-hard.

Multi-label problem

Multi-label problem

We define a label set $\mathcal{L}=\{1,2,\dots,L\}$, where L is a (finite) constant. Therefore the output domain is defined as $\mathcal{Y}=\mathcal{L}^{\mathcal{V}}$. 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 an input image.

In order to ease to notation we will omit $\mathbf{x}$ and define the *energy function* simply as

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

Metric *

A function $d:\mathcal{L}\times\mathcal{L}\to\mathbb{R}^+$ is called a **metric** if the following properties are satisfied:

1. **Identity of indiscernibles**: $d(\ell_1,\ell_2)=0\quad\Leftrightarrow\quad\ell_1=\ell_2$ for all $\ell_1,\ell_2\in\mathcal{L}$.
2. **Symmetry**: $d(\ell_1,\ell_2)=d(\ell_2,\ell_1)$ for all $\ell_1,\ell_2\in\mathcal{L}$.
3. **Triangle inequality**: $d(\ell_1,\ell_3)\leqslant d(\ell_1,\ell_2)+d(\ell_2,\ell_3)$ for all $\ell_1,\ell_2,\ell_3\in\mathcal{L}$.

Example: the **truncated absolute distance** $d:\mathbb{R}\times\mathbb{R}\to\mathbb{R}$, $d(x,y)=\min(K,|x-y|)$ is a *metric*, where K is some constant. (See Exercise)

If a function $d:\mathcal{L}\times\mathcal{L}\to\mathbb{R}$ satisfies the first two properties (i.e. identity of indiscernibles and symmetric), then it is called **semi-metric**.

Example: the **truncated quadratic function** $d:\mathbb{R}\times\mathbb{R}\to\mathbb{R}$, $d(x,y)=\min(K,|x-y|^2)$ is a *semi-metric*, where K is some constant. (See Exercise)

$\alpha-\beta$ swap

$\alpha-\beta$ swap

$\alpha-\beta$ swap changes the variables that are labeled as $\ell\in\{\alpha,\beta\}$. Each of these variables can choose either α or β. We introduce the following notation

$$\mathcal{Z}_{\alpha\beta}(\mathbf{y},\alpha,\beta)=\{\mathbf{z}\in\mathcal{Y}:z_i=y_i,\ \text{if } y_i\notin\{\alpha,\beta\},\ \text{otherwise } z_i\in\{\alpha,\beta\}\}\,.$$

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

$$\begin{aligned}\mathbf{z}^*\in\ &\operatorname*{argmin}_{\mathbf{z}\in\mathcal{Z}_{\alpha\beta}(\mathbf{y},\alpha,\beta)}E(\mathbf{z})=\operatorname*{argmin}_{\mathbf{z}\in\mathcal{Z}_{\alpha\beta}(\mathbf{y},\alpha,\beta)}\sum_{i\in\mathcal{V}}E_i(z_i)+\sum_{(i,j)\in\mathcal{E}}E_{ij}(z_i,z_j)\\ &=\operatorname*{argmin}_{\mathbf{z}\in\mathcal{Z}_{\alpha\beta}(\mathbf{y},\alpha,\beta)}\Bigg[\underbrace{\sum_{i\in\mathcal{V},\,y_i\notin\{\alpha,\beta\}}E_i(y_i)}_{\text{constant}}+\underbrace{\sum_{i\in\mathcal{V},\,y_i\in\{\alpha,\beta\}}E_i(z_i)}_{\text{unary}}\\ &\quad+\underbrace{\sum_{\substack{(i,j)\in\mathcal{E}\\ y_i,y_j\notin\{\alpha,\beta\}}}E_{ij}(y_i,y_j)}_{\text{constant}}+\underbrace{\sum_{\substack{(i,j)\in\mathcal{E}\\ y_i\in\{\alpha,\beta\},\,y_j\notin\{\alpha,\beta\}}}E_{ij}(z_i,y_j)}_{\text{unary}}+\underbrace{\sum_{\substack{(i,j)\in\mathcal{E}\\ y_i\notin\{\alpha,\beta\},\,y_j\in\{\alpha,\beta\}}}E_{ij}(y_i,z_j)}_{\text{unary}}+\underbrace{\sum_{\substack{(i,j)\in\mathcal{E}\\ y_i,y_j\in\{\alpha,\beta\}}}E_{ij}(z_i,z_j)}_{\text{pairwise}}\Bigg].\end{aligned}$$

Local optimization

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

E_{ij}	α	β
α	$E_{ij}(\alpha,\alpha)$	$E_{ij}(\alpha,\beta)$
β	$E_{ij}(\beta,\alpha)$	$E_{ij}(\beta,\beta)$

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

$$E_{ij}(\alpha,\alpha) + E_{ij}(\beta,\beta) = 0 \leqslant E_{ij}(\alpha,\beta) + E_{ij}(\beta,\alpha) = 2E_{ij}(\alpha,\beta) ,$$

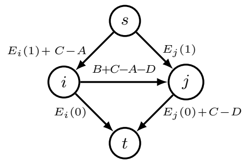
which means that E_{ij} is **regular** w.r.t. the labeling $\mathcal{Z}_{\alpha\beta}(\mathbf{y}, \alpha, \beta)$.

Graph construction for semi-metrics

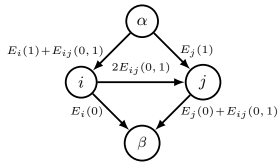
Let us consider the following *binary energy function*:

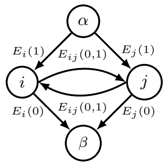
$$E(\mathbf{z}) = E_i(z_i) + E_j(z_j) + E_{ij}(z_i, z_j) ,$$

where E_{ij} is assumed to be a *semi-metric*.



Since E_{ij} is a *semi-metric*, we can construct a flow for $E(\mathbf{y})$ as follows:





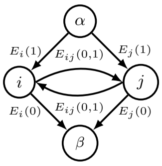
z_i	z_j	$E(\mathbf{z})$
0	0	$E_i(0)+E_j(0)$
0	1	$E_i(0)+E_j(1)+E_{ij}(1,0)$
1	0	$E_i(1)+E_j(0)+E_{ij}(1,0)$
1	1	$E_i(1)+E_j(1)$

Graph construction: t-links

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

$$\mathbf{z}^* \in \operatorname*{argmin}_{\mathbf{z} \in \mathcal{Z}_{\alpha\beta}(\mathbf{y},\alpha,\beta)} \sum_{\substack{i\in\mathcal{V}\\ y_i\in\{\alpha,\beta\}}} E_i(z_i) + \sum_{\substack{(i,j)\in\mathcal{E}\\ y_i\in\{\alpha,\beta\},\, y_j\notin\{\alpha,\beta\}}} E_{ij}(z_i, y_j) + \sum_{\substack{(i,j)\in\mathcal{E}\\ y_i\notin\{\alpha,\beta\},\, y_j\in\{\alpha,\beta\}}} E_{ij}(y_i, z_j) + \sum_{\substack{(i,j)\in\mathcal{E}\\ y_i, y_j\in\{\alpha,\beta\}}} E_{ij}(z_i, z_j).$$

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

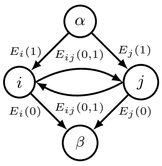


t-links: for all $i \in \mathcal{V}' \backslash \{\alpha,\beta\}$

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

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

Graph construction: n-links



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

$$c(i,j) = c(j,i) = E_{ij}(\alpha, \beta) .$$

$\alpha - \beta$ swap 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 **semi-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: $\mathbf{y}^* \leftarrow \mathbf{y}$
3: **for all** $(\alpha, \beta) \in \mathcal{L} \times \mathcal{L}$ **do**
4: find $\mathbf{z}^* \in \operatorname{argmin}_{\mathbf{z}\in\mathcal{Z}_{\alpha\beta}(\mathbf{y}^*,\alpha,\beta)} E(\mathbf{z})$
5: $\mathbf{y}^* \leftarrow \mathbf{z}^*$
6: **end for**
7: **if** $E(\mathbf{y}^*) < E(\mathbf{y})$ **then**
8: $\mathbf{y} \leftarrow \mathbf{y}^*$
9: Goto Step 2
10: **end if**

$\alpha - \beta$ swap algorithm is guaranteed to terminate in a finite number of cycles. This algorithm computes at least $|\mathcal{L}|^2$ graph cuts, which may take a lot of time, even for moderately large label spaces.

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 *minCut-maxFlow*.
- 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.

In the **next lecture** we will learn about

- α*-expansion*: approximate solution for the *multi-label problem*, if E_{ij} is metric
- *FastPD algorithm*: linear programming relaxation for *multi-label problem*

Literature *

1. Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of Min-cut/Max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, September 2004
2. Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, February 2004
3. 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
4. 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