Combinatorial Optimization in Computer Vision (IN2245)

Frank R. Schmidt
Csaba Domokos

Winter Semester 2015/2016

15. Higher-order potentials

Compact parameterization

Transforming higher-order energies

Consider a **higher-order** energy function E_c for **a given clique** c, which assigns a cost θ_1 if the variables $Y_c = \{Y_v \mid v \in c\}$ take a particular labeling $\mathbf{u}_c \in \mathcal{Y}_c$ and $\theta_{\text{max}} \geqslant \theta_1$ otherwise, that is

$$E_c(\mathbf{y}_c) = \begin{cases} \theta_1, & \text{if } y_c = u_c \\ \theta_{\text{max}}, & \text{otherwise.} \end{cases}$$

Goal: is to transform $E_c(\mathbf{y}_c)$ to an equivalent quadratic one.
Observe that the minimization of $E_c(\mathbf{y}_c)$ can be transformed to the minimization of a quadratic function using *one additional* **switching variable** z as:

$$\min_{\mathbf{y}_c} E_c(\mathbf{y}_c) = \min_{\mathbf{y}_c \in \mathcal{Y}_c, z \in \mathbb{B}} f(z) + \sum_{i \in c} g_i(z, y_i) \ ,$$

where the **selection function** $f(0) = \theta_1$ and $f(1) = \theta_{\text{max}}$ and $g_i(z, y_i)$ is called the **consistency function**.

Transforming higher-order energies

$$\min_{\mathbf{y}_c} E_c(\mathbf{y}_c) = \min_{\mathbf{y}_c \in \mathcal{Y}_c, z \in \mathbb{B}} f(z) + \sum_{i \in c} g_i(z, y_i) \ ,$$

The *consistency function* $g_i(z, y_i)$ is defined as:

$$g_i(z, y_i) = \begin{cases} 0, & \text{if } z = 1 \\ 0, & \text{if } z = 0 \text{ and } y_i = u_i \\ \infty, & \text{otherwise.} \end{cases}$$

General higher-order energies

Observation: in many computer vision problems higher order energies assign a low cost to **only a few** label assignments and the rest of the labelings are given a high cost, i.e. the energy function is **sparse**.

Let $\mathcal{U} = \{\mathbf{u}_c^1, \mathbf{u}_c^2, \ldots, \mathbf{u}_c^t\}$ be a list of possible labelings of the clique c and their corresponding costs $\Theta = \{\theta_1, \ldots, \theta_t\}$. For all other labelings we assign a high constant cost θ_{max}.

The energy function can be defined as:

$$E_c(\mathbf{y}_c) = \begin{cases} \theta_q, & \text{if } \mathbf{y}_c = \mathbf{u}_c^q \in \mathcal{U} \\ \theta_{\text{max}}, & \text{otherwise,} \end{cases}$$

where $\theta_{\text{max}} \geqslant \theta_q$ for all $q = 1, \ldots, t$.

The previous idea can be generalized and used to transform any higher order function to a quadratic one.

General higher-order energies

The minimization of the above defined higher order function can be transformed to a quadratic function using a $(t+1)$-state *switching variable* as:

$$\min_{\mathbf{y_c}} E_c(\mathbf{y}_c) = \min_{\mathbf{y}_c \in \mathcal{Y}_c, z \in \{0,1,\ldots,t\}} f(z) + \sum_{i \in c} g_i(z, x_i) \ ,$$

where

$$f(z) = \begin{cases} \theta_q, & \text{if } z = q \in \{1, \ldots, t\} \\ \theta_{\text{max}}, & \text{if } z = 0, \end{cases}$$

and

$$g_i(z, y_i) = \begin{cases} 0, & \text{if } z = q \in \{1, \ldots, t\} \text{ and } y_i = u_i^q \\ 0, & \text{if } z = 0 \\ \infty, & \text{otherwise.} \end{cases}$$

Note that the first state of the *switching variable* z does not penalize any labeling of the clique c.

Compact parameterization

In general, the previous approach can be used to transform any general higher order potential.

It reduces the complexity of performing inference in higher order cliques. However, the addition of a switching variable with $|\mathcal{L}|^{|c|}$ states is required, which is generally **infeasible**.

Observation: many low cost label assignments tend to be close to each other in terms of the difference between labelings of pixels.

The cost of such groups of similar labelings can be encoded without increasing the number of states of the switching variable z in the transformation to quadratic functions.

Let us introduce a list of **labeling deviation cost functions** $\mathcal{D} = \{d_1, d_2, \ldots, d_t\}$ encoding how the cost changes as the labeling moves away from some desired labeling.

Compact parameterization

The energy function $E_c(\mathbf{y}_c)$ can be defined as:

$$E_c(\mathbf{y}_c) = \min \left(\min_{q \in \{1,2,\ldots,t\}} \theta_q + d_q(\mathbf{y}_c), \theta_{\mathsf{max}} \right) ,$$

where *deviation functions* $d_q : \mathcal{L}^{|c|} \rightarrow \mathbb{R}$ are defined as:

$$d_q(\mathbf{y}_c) = \sum_{i \in c, l \in \mathcal{L}} w_{il}^q [\![y_i = l]\!] ,$$

where w_{il}^q is the **cost** added to the deviation function if label l is assigned to variable Y_i ($i \in c$) for a given clique c.
This function can be considered as the generalization of the *general higher-order energy functions*, when the weights w_{il}^q are set as:

$$w_{il}^q = \begin{cases} 0, & \text{if } u_i^q = l \\ \theta_{\mathsf{max}}, & \text{otherwise.} \end{cases}$$

Compact parameterization

$$E_c(\mathbf{y}_c) = \min \left(\min_{q \in \{1,2,\ldots,t\}} \theta_q + \sum_{i \in c, l \in \mathcal{L}} w_{il}^q [\![y_i = l]\!], \theta_{\text{max}} \right) .$$

One can apply the same transformation as before using a $(t+1)$-state *switching variable*. Therefore,

$$\min_{\mathbf{y}_c \in \mathcal{Y}_c} E_c(\mathbf{y}_c) = \min_{\mathbf{y}_c \in \mathcal{Y}_c, z \in \{0,1,\ldots,t+1\}} f(z) + \sum_{i \in c} g_i(z, y_i) ,$$

where

$$f(z) = \begin{cases} \theta_q, & \text{if } z = q \in \{1, \ldots, t\} \\ \theta_{\text{max}}, & \text{if } z = 0 , \end{cases}$$

and

$$g_i(z, y_i) = \begin{cases} w_{il}^q, & \text{if } z = q \in \{1, \ldots, t\} \text{ and } y_i = l \in \mathcal{L} \\ 0, & \text{if } z = t+1 . \end{cases}$$

P^n Potts model

The P^n **Potts model** is the higher-order generalization of the *Potts model*, which is used for modeling many computer vision problems.

The energy function of the P^n Potts model for cliques $c \subseteq \mathcal{V}$ of size n is defined as

$$E_c(\mathbf{y}_c) = \begin{cases} \theta_k, & \text{if } y_i = l_k, \quad \forall i \in c \\ \theta_{\text{max}}, & \text{otherwise,} \end{cases}$$

where $\theta_{\text{max}} > \theta_k$, for all $l_k \in \mathcal{L}$.

For a pairwise clique this reduces to the *Potts model*:

$$E_{ij}(y_i, y_j) = \begin{cases} \theta_k, & \text{if } y_i = y_j = l_k, \forall i \in c \\ \theta_{\text{max}}, & \text{otherwise.} \end{cases}$$

If $\theta_k = 0$, for all l_k, this P^n becomes a metric.

α-expansion for P^n Potts model

The P^n Potts model can be reformulated as

$$E_c(\mathbf{y}_c) = \max_{i,j \in c} E_{ij}(y_i, y_j) \ ,$$

where $E_{ij}(y_i, y_j)$ is a pairwise Potts model.

Theorem 1. *If $E_c(\mathbf{y}_c)$ is the P^n Potts potential, then the optimal α-expansion move for any $\alpha \in \mathcal{L}$ can be computed in polynomial time.*

Proof. Any configuration $\mathbf{y}_c$ can be decomposed as $\mathbf{y}_c = \{y_i, y_j\} \cup \mathbf{y}_{c \setminus \{i,j\}}$ for $i, j \in c$. Then

$$\begin{aligned} E_c(\mathbf{y}_c) =& E_c(\{y_i, y_j\} \cup \mathbf{y}_{c \setminus \{i,j\}}) \\ =& \max \Big(E_{ij}(y_i, y_j), \max_{k \in c \setminus \{i,j\}} E_{ij}(y_i, y_k), \max_{k \in c \setminus \{i,j\}} E_{ij}(y_j, y_k), \\ & \max_{k,l \in c \setminus \{i,j\}} E_{ij}(y_k, y_l) \Big) \ . \end{aligned}$$

α-expansion for P^n Potts model

Proof cont'd. Let us introduce the following notations

$$D_a = \max_{k \in c \setminus \{i,j\}} E_{ij}(a, y_k) \text{ , for } a \in \mathcal{L} \quad \text{and} \quad D = \max_{k,l \in c \setminus \{i,j\}} E_{ij}(y_k, y_l) \ .$$

Therefore, $E_c(\{y_i, y_j\} \cup \mathbf{y}_{c \setminus \{i,j\}}) = \max(E_{ij}(a, b), D_a, D_b, D)$.

The optimal expansion move can be computed in polynomial time if all projections of any α-expansion move energy on two variables of the clique are submodular (see Lecture 13). That is for $\alpha, a, b \in \mathcal{Y}_c$

$$E_c(\{\alpha, \alpha\} \cup \mathbf{y}_{c \setminus \{i,j\}}) + E_c(\{a, b\} \cup \mathbf{y}_{c \setminus \{i,j\}}) \leqslant$$
$$E_c(\{\alpha, b\} \cup \mathbf{y}_{c \setminus \{i,j\}}) + E_c(\{a, \alpha\} \cup \mathbf{y}_{c \setminus \{i,j\}})$$

should be satisfied. Thus we obtain

$$\max(E_{ij}(\alpha, \alpha), D_\alpha, D_\alpha, D) + \max(E_{ij}(a, b), D_a, D_b, D) \leqslant$$
$$\max(E_{ij}(\alpha, b), D_\alpha, D_b, D) + \max(E_{ij}(a, \alpha), D_a, D_\alpha, D) \ .$$

α-expansion for P^n Potts model

Proof cont'd.

$$\max(E_{ij}(\alpha,\alpha), D_\alpha, D_\alpha, D) + \max(E_{ij}(a,b), D_a, D_b, D) \leqslant$$
$$\max(E_{ij}(\alpha,b), D_\alpha, D_b, D) + \max(E_{ij}(a,\alpha), D_a, D_\alpha, D) \ .$$

- If $\alpha \in \{a, b\}$, then the above inequality is satisfied by a equality.
- If $\alpha \notin \{a, b\}$, then the RHS of the above inequality is equal to $2\theta_{\text{max}}$. The maximum value of the LHS is $2\theta_{\text{max}}$.

This implies that the above inequality is always true. □

Robust P^n Potts model

Robust P^n model

The P^n Potts model enforces label consistency rigidly. This can be resolved by **relaxing** the *step function* by a linear function.

We introduce notation $N_i(\mathbf{y}_c)$ for the number of variables in the clique c not taking the dominant label, that is

$$N_i(\mathbf{y}_c) = \min_k \left(|c| - \sum_{i \in c} [\![y_i = k]\!] \right) \ .$$

The **robust P^n potential** has the form

$$E_c(\mathbf{y}_c) = \begin{cases} \frac{\theta_{\text{max}}}{Q} N_i(\mathbf{y}_c), & \text{if } N_i(\mathbf{y}_c) \leqslant Q \\ \theta_{\text{max}}, & \text{otherwise,} \end{cases}$$

where Q is the truncation parameter which controls the rigidity of the clique potential.

Generalized robust P^n Potts model

Multiple robust P^n model can be combined to approximate any non-decreasing concave consistency potential up to an arbitrary accuracy.

This results in the (**generalized**) robust P^n Potts model, which can be formulated as

$$E_c(\mathbf{y}_c) = \min\Big(\min_{l\in\mathcal{L}}\big(\theta_l + \sum_{i\in c} w_i^l[\![y_i \neq l]\!]\big), \theta_{\mathsf{max}}\Big) \ ,$$

where $\theta_{\mathsf{max}} \geqslant \theta_l$ for all $l \in \mathcal{L}$.

The potential has a cost of θ_l if all pixels in the clique (segment) take the label l. Each pixel not taking the label l is penalized with an additional cost of w_i^l, and the maximum cost of the potential is truncated to θ_{max}.

Generalized robust P^n Potts model

$$E_c(\mathbf{y}_c) = \min\Big(\min_{l\in\mathcal{L}}\big(\theta_l + \sum_{i\in c} w_i^l[\![y_i \neq l]\!]\big), \theta_{\mathsf{max}}\Big) \ ,$$

According to compact reparameterization,

$$E_c(\mathbf{y}_c) = \min_{\mathbf{y}_c\in\mathcal{Y}_c, l\in\mathcal{L}} f(l) + \sum_{i\in c} g_i(l, y_i) \ ,$$

where $f(l) = \theta_l$ and

$$g_i(l, y_i) = \begin{cases} 0, & \text{if } l = l_F \text{ or } y_i = l \in \mathcal{L} \\ w_i^l, & \text{otherwise.} \end{cases}$$

Here l_F stands for a "**free label**". This special label means that any possible label in $\mathcal{L}$ can be assigned to local nodes without any cost.

α-expansion for robust P^n model

Theorem 2. *The pairwise consistency function* $g_i(l, y_i)$

$$g_i(l, y_i) = \begin{cases} 0, & \text{if } y_i = l_F \text{ or } l = y_i \\ w_i^l, & \text{if } y_l \neq l_F \text{ and } l \neq y_i \end{cases}$$

can be written as:

$$g_i(l, y_l) = \frac{w_i^l}{2}[\![l \neq l_F]\!] - \frac{w_i^k}{2}[\![y_i = k \neq l_F]\!] + E(l, y_i) \ ,$$

where

$$E(y_i, l) = \begin{cases} 0, & \text{if } l = y_i \\ \frac{w_i^k}{2}, & \text{if } (y_i = l_F \text{ and } l = k \neq l_F) \text{ or } (y_i = k \neq l_F \text{ and } l = l_F) \\ \frac{w_i^k + w_i^l}{2}, & \text{if } y_i = k \neq l_F \text{ and } l \neq l_F \ . \end{cases}$$

α-expansion for robust P^n model

Proof.

$$g_i(l, y_l) = \frac{w_i^l}{2}[\![l \neq l_F]\!] - \frac{w_i^k}{2}[\![y_i = k \neq l_F]\!] + E(l, y_i) \ ,$$

where

$$E(y_i, l) = \begin{cases} 0, & \text{if } l = y_i \\ w_i^k, & \text{if } (y_i = l_F \text{ and } l = k \neq l_F) \text{ or } (y_i = k \neq l_F \text{ and } l = l_F) \\ \frac{w_i^k + w_i^l}{2}, & \text{if } y_i = k \neq l_F \text{ and } l \neq l_F \ . \end{cases}$$

- $l = l_F$ and $y_i = l_F$: $g(l, y_i) = 0$.
- $l = l_F$ and $y_i \neq l_F$: $g(l, y_i) = \frac{-w_i^k}{2} + \frac{w_i^k}{2} = 0$.
- $l \neq l_F$ and $y_i = l_F$: $g(l, y_i) = \frac{-w_i^l}{2} + \frac{-w_i^l}{2} = w_i^l$.
- $l = y_i \neq l_F$: $g(l, y_i) = \frac{w_i^k}{2} - \frac{w_i^k}{2} = 0$.
- $l \neq l_F$ and $y_i = k \neq l_F$: $g(l, y_i) = \frac{w_i^l}{2} - \frac{w_i^k}{2} + \frac{w_i^k + w_i^l}{2} = w_i^l$.

□

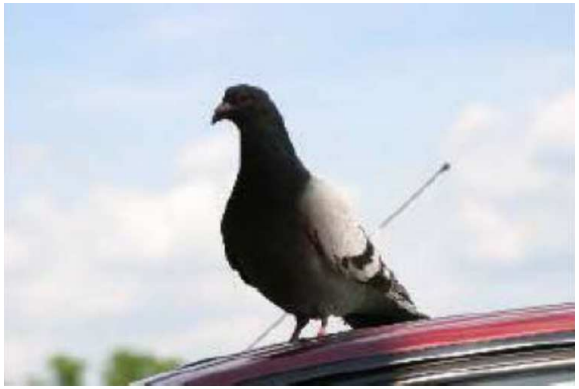
α-expansion for robust P^n model

It can be seen that $E(y_i, l)$ is a metric, as $\forall a, b, c \in \mathcal{L} \cup \{l_F\}$

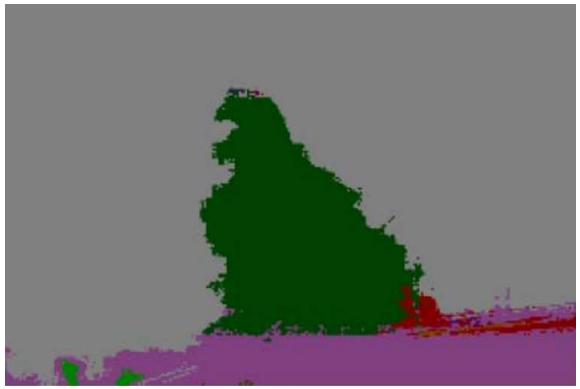
$$\begin{aligned} E(a,b) &\geqslant 0 \\ E(a,b) &= 0 \qquad \Leftrightarrow a = b \\ E(a,b) &= E(b,a) \\ E(a,b) + E(b,c) &\geqslant E(a,c)\ . \end{aligned}$$

Thus, every possible α-expansion move is submodular.

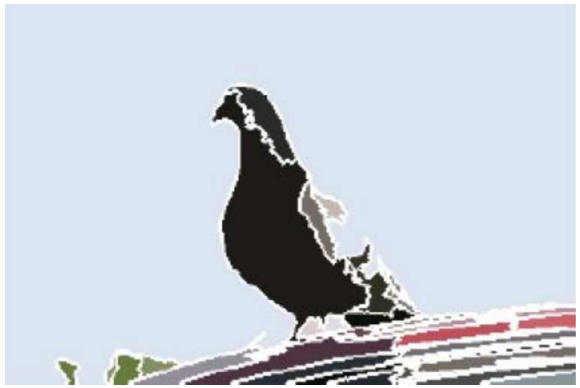
Example



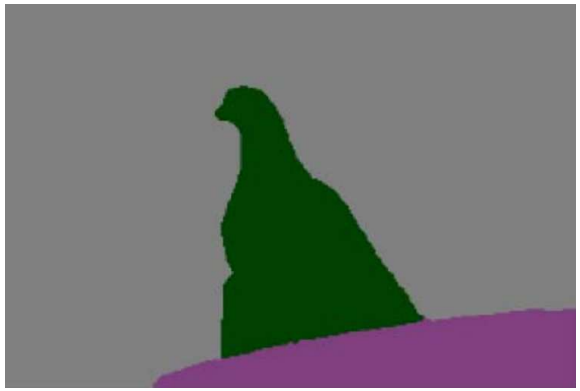
Input



Unaries

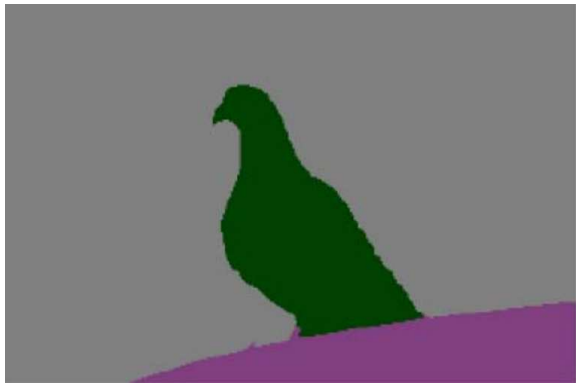


Super-pixels



Pairwise model

P^n Potts model



Robust P^n model

Region-based consistency potential

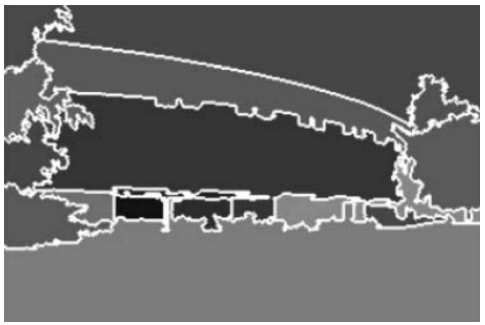
For a given clique c of variables we define the **label inconsistency cost** as $\lambda|c|^\gamma$ for some parameters $\lambda, \gamma \in \mathbb{R}$, i.e. the cost of a labeling in which different labels have been assigned to the pixels constituting the given clique.

This can be expressed by P^n Potts model, which favours all pixels belonging to a segment taking the same label.

One might also consider a **quality sensitive potential** which works by modulating the label inconsistency cost with a function of the quality of a given clique c.

Input

Over-segmentation



Segment qualities
(the darker the better)

Region-based consistency potential

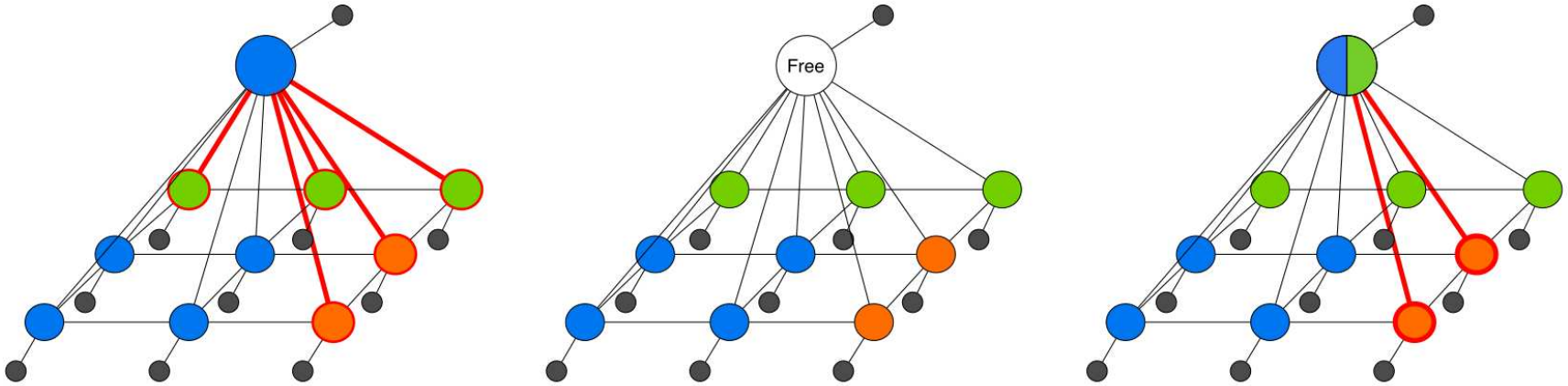
The (robust) quality sensitive potential has the form

$$E_c(\mathbf{y}_c) = \begin{cases} \frac{\theta_{\text{max}}}{Q} N_i(\mathbf{y}_c), & \text{if } N_i(\mathbf{y}_c) \leqslant Q \\ \theta_{\text{max}}, & \text{otherwise,} \end{cases}$$

where Q is the truncation parameter which controls the rigidity of the clique potential. $\theta_{\text{max}} = (\alpha g(c) + \lambda)|c|^\gamma$, where $\alpha \in \mathbb{R}$.

$g(c)$ is a function measuring the quality of a given clique c (e.g., the variance of feature responses evaluated on all constituent pixels of a segment).

Harmony potential



P^n Potts | Robust P^n Potts | Harmony potential

Idea: The global node takes a label from $\mathcal{P}(\mathcal{L})$, which is able to encode any combination of local node labels. The **Harmony potential** establishes a penalty for local node labels not encoded in the label of the global node. It is simply defined by

$$g_i(l, y_i) = \theta_i [\![y_i \notin l]\!] \, .$$

It penalizes when y_i is not encoded in $l \in \mathcal{P}(\mathcal{L})$, but not when a particular label in l does not appear in the y_i.

Harmony

As we have seen the generalized robust P^n Potts model is defined as

$$E_c(\mathbf{y}_c) = \min\Big(\min_{l\in\mathcal{L}}\big(\theta_l + \sum_{i\in c} w_i^l [\![y_i \neq l]\!]\big), \theta_{\mathsf{max}}\Big) \ ,$$

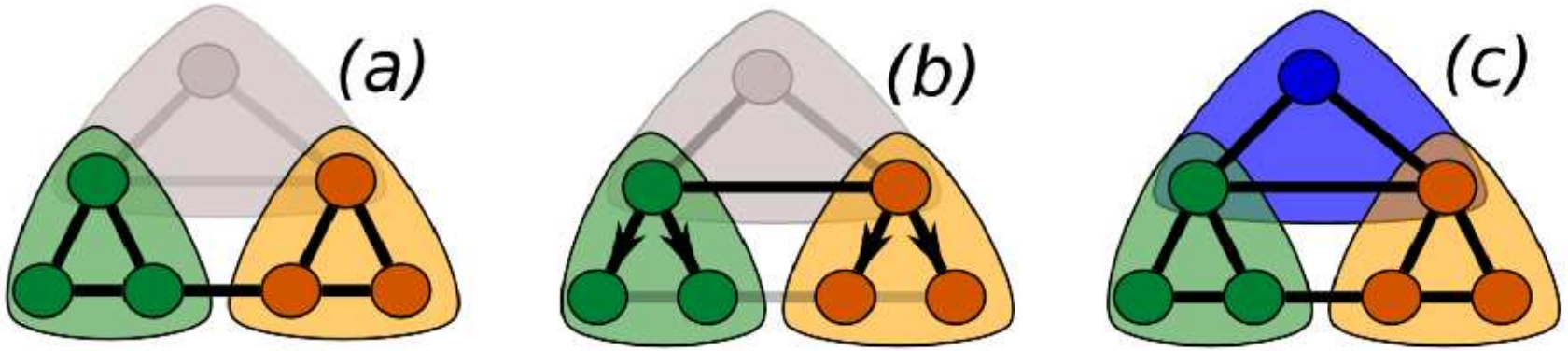
Analyzing the definition of the robust P^n-based potential, we see that l_F is essentially a "wildcard" label that represents any possible label from $\mathcal{L}$. It can be seen that the harmony potential can be written as

$$E_c(\mathbf{y}_c) = \min\Big(\min_{l\in\mathcal{L}}\big(\theta_l + \sum_{i\in c} w_i^l [\![y_i \notin l]\!]\big), \theta_{\mathsf{max}}\Big) \ ,$$

The harmony potential generalizes the robust P^n-based potential by admitting wildcard labels at the global node, while also allowing concrete and heterogeneous label combinations to be enforced by the global node.

Hierarchical model

By making use of higher order energies functions one may define a hierarchical model.



Hierarchical model

A **hierarchical model** is of the form:

$$E(\mathbf{y}) = \sum_{i \in \mathcal{V}} E_i(y_i) + \sum_{(i,j) \in \mathcal{E}} (y_i, y_j) + \min_{\mathbf{y}^{(1)}} E^{(1)}(\mathbf{y}, \mathbf{y}^{(1)}) \ ,$$

where $E^{(1)}(\mathbf{y}, \mathbf{y}^{(1)})$ is recursively defined as:

$$E^{(n)}(\mathbf{y}^{(n-1)}, \mathbf{y}^{(n)}) = \sum_{c \in \mathcal{S}^{(n)}} (\mathbf{y}^{(n-1)}, y_c^{(n)}) + \sum_{c,d \in \mathcal{S}^{(n)}} E_{cd}(y_c^{(n)}, y_d^{(n)})$$

$$+ \min_{\mathbf{y}^{(1)}} E^{(1)}(\mathbf{y}, \mathbf{y}^{(1)}) \ .$$

$\mathbf{y} = \mathbf{y}^{(0)}$ refers to the state of the base level, and $\mathbf{y}^{(n)}$, for $n \geqslant 1$ the state of **auxiliary variables**. The structure of the graph is chosen beforehand and for all layers n beyond the maximal layer in the hierarchy m (i.e. $n \geqslant m$)

$$E^{(n)}(\mathbf{y}^{(n-1)}, \mathbf{y}^{(n)}) = 0 \ .$$

Literature

- Carsten Rother, Pushmeet Kohli, Wei Feng, and Jiaya Jia. **Minimizing Sparse Higher Order Energy Functions of Discrete Variables**. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1382–1389, Miami, FL, USA, June, 2009.
- Pushmeet Kohli, M. Pawan Kumar, and Philip H. S. Torr. $\mathcal{P}^3$ **& Beyond: Solving Energies with Higher Order Cliques**. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Minneapolis, MN, USA, June, 2007.
- Pushmeet Kohli, Lubor Ladický, and Philip H.S. Torr. **Robust Higher Order Potentials for Enforcing Label Consistency**. In *International Journal of Computer Vision*, vol. 82, pp. 302–324, January, 2009.
- Xavier Boix, Josep M. Gonfaus, Joost van de Weijer, Andrew D. Bagdanov, Joan Serrat, and Jordi González. **Harmony Potentials Fusing Global and Local Scale for Semantic Image Segmentation**. In *International Journal of Computer Vision*, vol. 96(1), pp. 83–102, January, 2012.
- L'ubor Ladický, Chris Russell, Pushmeet Kohli, and Philip H. S. Torr. **Associative Hierarchical Random Fields**. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36(6), pp. 1056–1077, June, 2014.