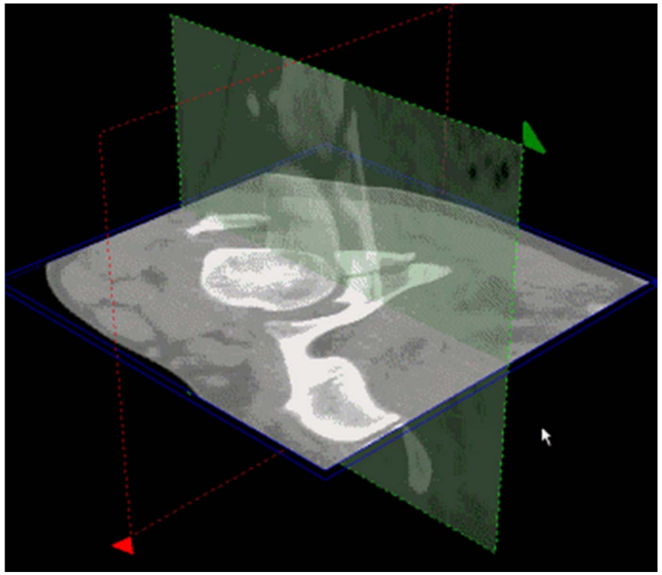
Higher-order Graph Cuts

Hiroshi Ishikawa　石川 博
Department of Computer Science & Engineering
Waseda University　早稲田大学

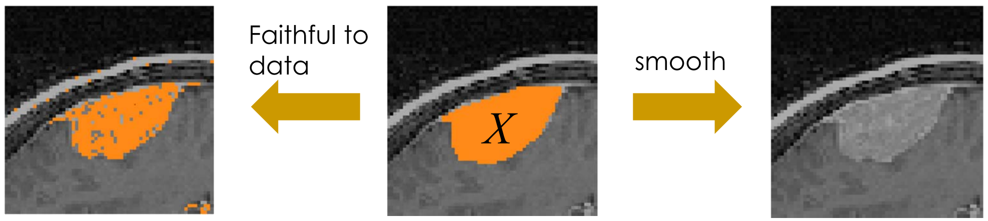
Example: Segmentation



Boykov&Jolly
ICCV2001

Example: Segmentation

- Local model
  - ex.: Models of pixel values for each kind of tissue
- Prior model / regularization
  - Assume smoothness
- Express the tradeoff by an energy $E(X)$
  - Faithful to the data and model and smooth



Example: Segmentation

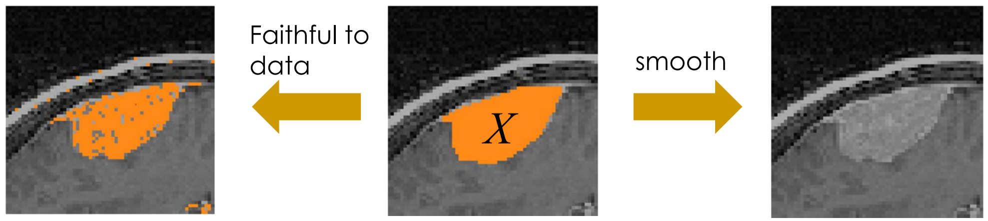
- Find the X that minimizes the energy

$X_v = 0$ or 1 for each pixel v

$$E(X) = \sum_{v \in V} g_v(X_v) + \sum_{(u,v) \in E} \kappa |X_u - X_v|$$

All pixels

Neighboring pairs of pixels



Example: Segmentation

- Find the X that minimizes the energy

$X_v = 0$ or 1 for each pixel v

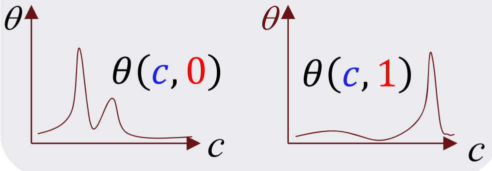
Data term
Based on the data and the image formation model

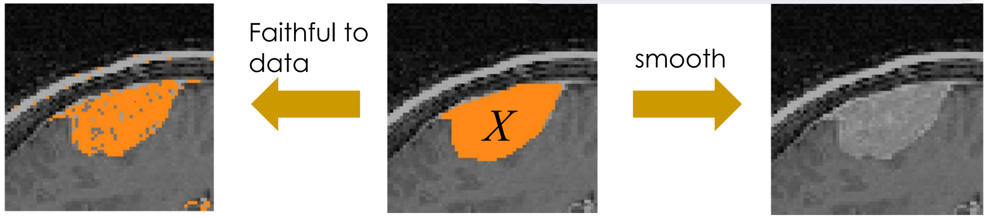
$$\sum_{v \in V} g_v(X_v)$$

All pixels

Data

$$g_v(l) = -\log \theta(I_v, l)\,,\, l = 0,1$$





Example: Segmentation

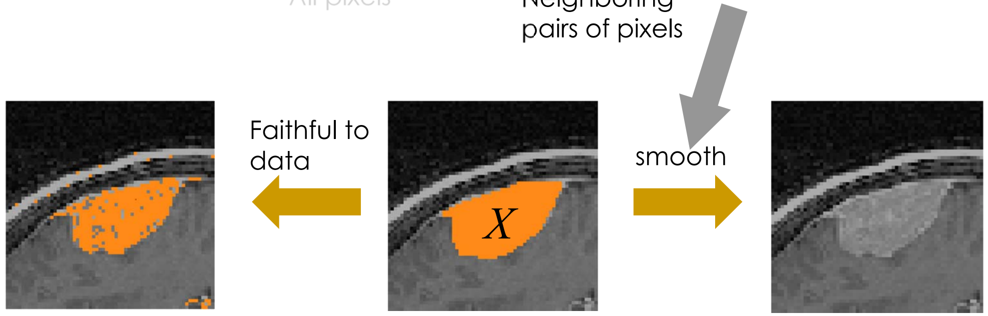
- Find the X that minimizes the energy

$X_v = 0$ or 1 for each pixel v

0 if neghiboring labels coincide
$\kappa > 0$ if they differ

$$\sum_{(u,v) \in E} \kappa |X_u - X_v|$$

Neighboring pairs of pixels



Energy Minimization

- Consider the energy of the form

$$E(X) = \sum_{v \in V} g_v(X_v) + \sum_{(u,v) \in E} h_{uv}(X_u, X_v)$$

Data term — Smoothing term

where V is the set of locations (sites)
E is the set of neighboring pairs of sites
X assigns a label to each site in V

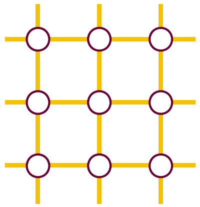
- 1st order Markov Random Field (MRF)
- Problem: Find the X that minimizes $E(X)$

Energy Minimization

- Problem: Find the X that minimizes $E(X)$
- Possible X
  - Combinations of labeling the sites
  - If V is 64×64 and labels $0,1$, $2^{4096} > 10^{1233}$
- NP-Hard in general
- Old method: Monte Carlo
- For some form of energy, Graph cut algorithms can globally minimize

Markov Random Field

- Graph $G = (V, E)$
  - V : each $v \in V$ has an L-valued random variable X_v
  - E : represents dependence
- $\mathscr{C}$: set of cliques in V
- Probability distribution of MRF



$$P(X) = \frac{1}{Z} \prod_{C \in \mathscr{C}} q_C(X_C) = \frac{p(X)}{Z}$$

$$Z = \sum_{X \in \mathscr{X}} \prod_{C \in \mathscr{C}} q_C(X_C) = \sum_{X \in \mathscr{X}} p(X)$$

$$X_C = (X_v)_{v \in C}$$

Markov Random Field

- Rewrite the MRF probability distribution

$$P(X) = \frac{1}{Z} \underbrace{\prod_{C \in \mathscr{C}} q_C(X_C)}_{p(X)} = \frac{p(X)}{Z}$$

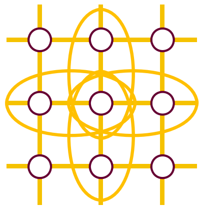
as

$$p(X) = e^{-E(X)}$$ ← Energy

$$E(X) = \sum_{C \in \mathscr{C}} f_C(X_C)$$ ← Potential

First order MRF

- The simplest (interesting) MRF
  Geman & Geman 1984; Besag 1986



- Model neighborhood relation

$$\mathscr{C} \approx V \cup E$$

$$p(X) = \prod_{v \in V} \psi_v(X_v) \prod_{(u,v) \in E} \phi_{uv}(X_u, X_v)$$

- Energy

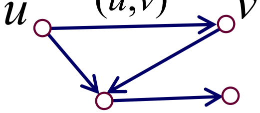
$$E(X) = \sum_{C \in \mathscr{C}} f_C(X_C) = \sum_{v \in V} g_v(X_v) + \sum_{(u,v) \in E} h_{uv}(X_u, X_v)$$

Graphs and cuts

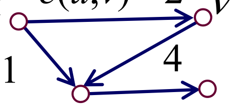
- Directed graph

$$\mathcal{G} = (\mathcal{V}, \mathcal{E})$$

$\mathcal{V}$: Finite set (vertices) $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ edges

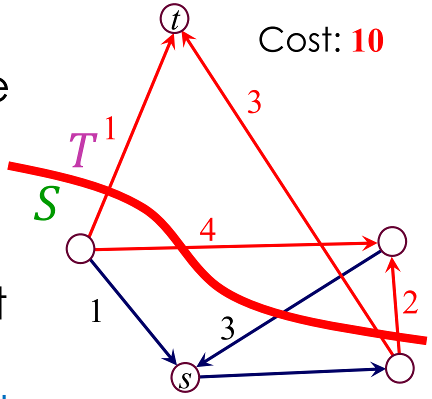


- Edges are weighted
  - $c : \mathcal{E} \to \mathbb{R}$



Graphs and cuts

- Choose $s, t \in \mathcal{V}$
- Cut: partition $\mathcal{V} = S \cup T$
  $S \cap T = \emptyset, s \in S, t \in T$
- Cost of cut: sum of the weights of the edges going from S to T
- Minimum cut: the cut with the minimum cost
- When all weights ≥ 0, the minimum cut can be found efficiently



Labeling problem

Original	Y Noise-added	X Denoised

- Assign a label to each pixel
  - Pixel $v \in V \rightarrow$ Label $X_v \in L$
- Set of possible labelings: $\mathscr{X} = L^V$

Simplest example

Original	Y Noise-added	X Denoised

$$E(X) = \sum_{v \in V} \lambda |Y_v - X_v| + \sum_{(u,v) \in E} |X_u - X_v|$$

- Globally optimizeable using graph cuts
  Greig, Porteous and Seheult '89

Energy minimization by minimum cuts (binary case)

$$E(X) = \sum_{v \in V} g_v(X_v) + \sum_{(u,v) \in E} \kappa |X_u - X_v|$$

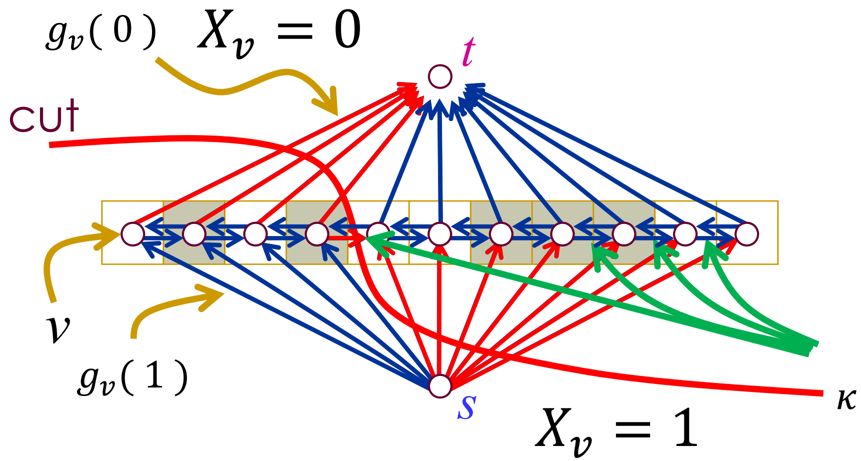


Image plane graph

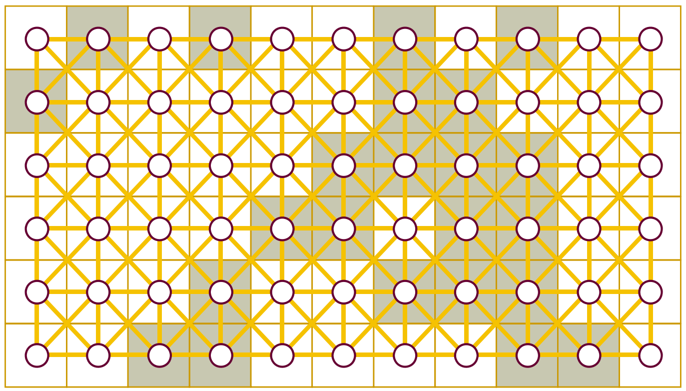
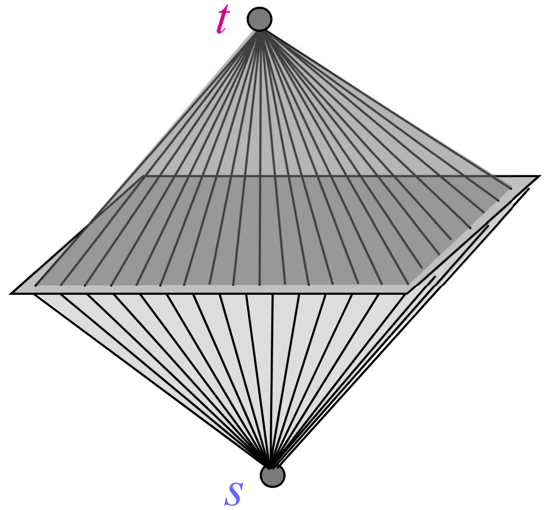
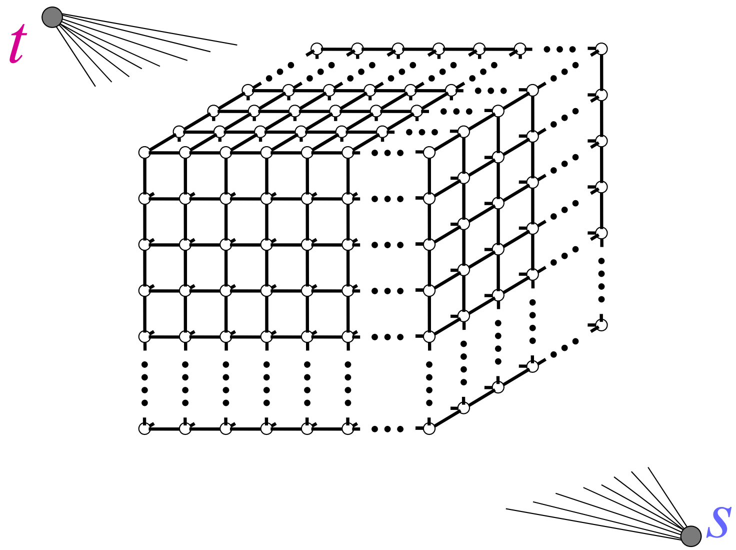


Image plane graph

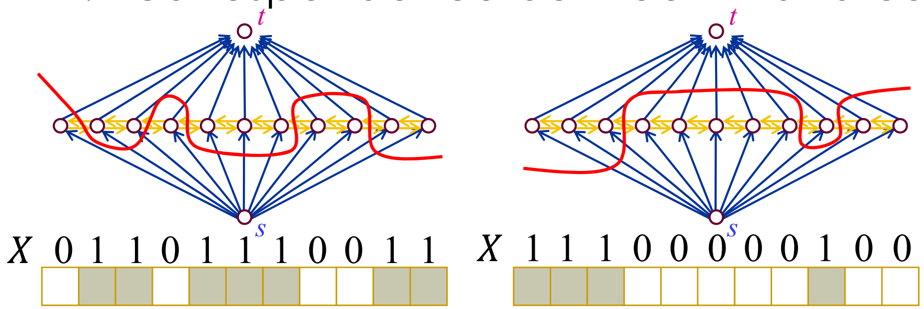


The 3D case



Energy minimization by minimum cuts (binary case)

- 1:1 correspondence between X and cuts



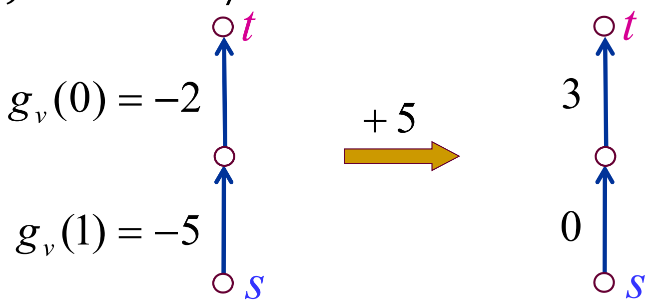
- Energy = Cut cost
- Minimum cut → Energy minimization
- The weight must be ≥ 0

Energy minimization by minimum cuts (binary case)

- The edge weight must be ≥ 0
  - What energy can be minimized?

$$E(X) = \sum_{v \in V} g_v(X_v) + \sum_{(u,v) \in E} h_{uv}(X_u, X_v)$$

- $g_v(x)$ is arbitrary



Energy minimization by minimum cuts (binary case)

- The edge weight must be ≥ 0
  - What energy can be minimized?

$$E(X) = \sum_{v \in V} g_v(X_v) + \sum_{(u,v) \in E} h_{uv}(X_u, X_v)$$

- $g_v(x)$ is arbitrary
- What about $h_{uv}(X_u, X_v)$?

Submodularity condition

$$h_{uv}(0,0) + h_{uv}(1,1) \leq h_{uv}(0,1) + h_{uv}(1,0)$$

Energy minimization by minimum cuts ($\geq$ 3 labels)

Ishikawa IEEE TPAMI 2003

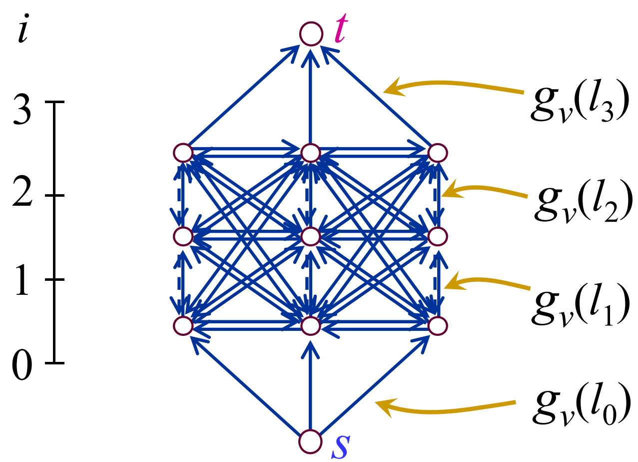
$$E(X) = \sum_{v \in V} g_v(X_v) + \sum_{(u,v) \in E} h_{uv}(X_u, X_v)$$

- If the L has linear order $L = \{l_0, l_1, \dots, l_k\}$
  - Globally minimizeable $\Leftrightarrow$ $h_{uv}(l_i, l_j)$ is a convex function of $i - j$

Energy minimization by minimum cuts ($\geq$ 3 labels)

Ishikawa IEEE TPAMI 2003

$$E(X) = \sum_{v \in V} g_v(X_v) + \sum_{(u,v) \in E} h_{uv}(X_u, X_v)$$



In general: $h_{uv}(l_i, l_j) = \tilde{h}_{uv}(i - j)$, $\tilde{h}_{uv}$:convex

Energy minimization by minimum cuts (≥ 3 labels)

$$E(X) = \sum_{v \in V} g_v(X_v) + \sum_{(u,v) \in E} h_{uv}(X_u, X_v)$$

- If the L has linear order $L = \{l_0, l_1, \dots, l_k\}$
  - Globally minimizeable ⇔ $h_{uv}(l_i, l_j)$ is a convex function of $i - j$
- Approximation algorithms

Boykov et al. IEEE TPAMI 2001

  - $\alpha\beta$ swap, α-expansion
  - Minimizes within factor $2c$ of the global minimum $c = \max_{u,v \in V} \left(\frac{\max_{X_u \neq X_v} h_{uv}(X_u, X_v)}{\min_{X_u \neq X_v} h_{uv}(X_u, X_v)} \right)$
  - $c = 1$ with the Potts model $h_{uv}(X_u, X_v) = \begin{cases} 0 & (X_u = X_v) \\ 1 & (X_u \neq X_v) \end{cases}$

≥ 3 labels, approximation

Move-making algorithms

- Iterative approximation algorithms
- In each iteration, finds the globally optimal move using binary graph cuts

Move

- $\alpha\beta$ swap
  - Allows label changes $\alpha \to \beta$, $\beta \to \alpha$ only
- α-expansion
  - Allows changing to α only

α-expansion

In each iteration, the α area expands

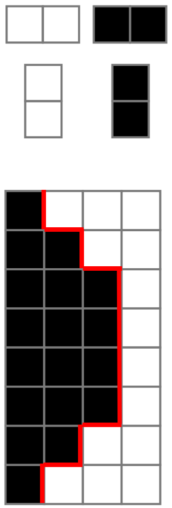
Initial
- -expansion
- -expansion
- -expansion
- -expansion
- -expansion
- -expansion
- -expansion

Choose the move that minimizes best : binary optimization

Courtesy Yuri Boykov

First-order energy

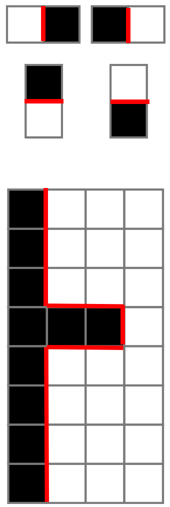
Good (Low Energy)



12 Bad

40 Good

Bad (High Energy)



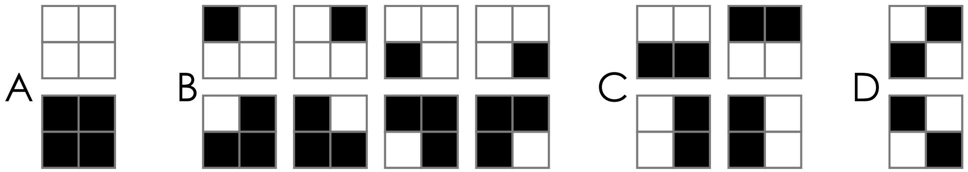
12 Bad

40 Good

Higher-order energy

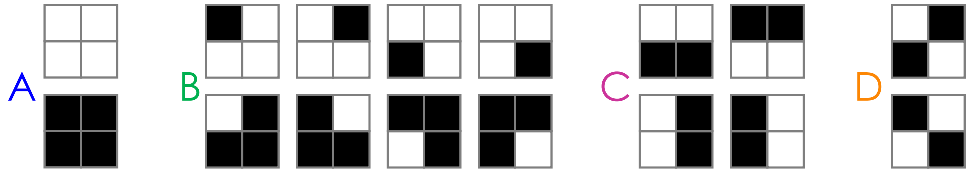
Good (Low Energy) Bad (High Energy)

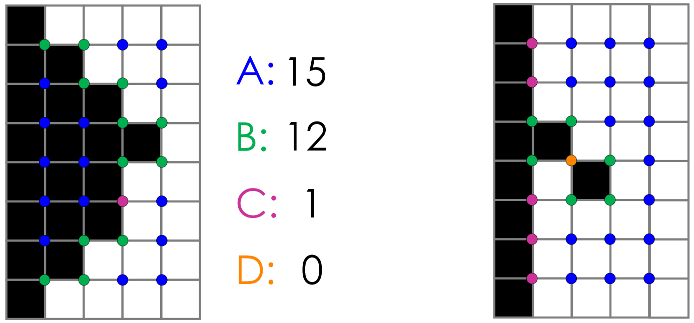
Better (Lower Energy) ⟷ Worse (Higher Energy)



Higher-order energy

Better (Lower Energy) ⟷ Worse (Higher Energy)





A: 15
B: 12
C: 1
D: 0

A: 16
B: 6
C: 5
D: 1

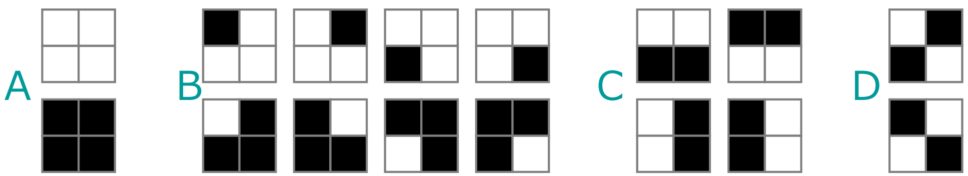
Higher-order energy

$$E(X) = \sum_{C\in\mathscr{C}} f_C(X_C)$$

$$= \sum_{v\in V} g_v(X_v) + \sum_{(u,v)} h_{uv}(X_u, X_v)$$

$$+ \sum_{(u,v,s,t)} k_{uvst}(X_u, X_v, X_s, X_t)$$

Better (Lower Energy) ⟷ Worse (Higher Energy)



Higher-order energy

- Transform arbitrary higher-order binary energy

Ishikawa CVPR 2009, PAMI 2011

$$E(X) = E(X_1, \cdots, X_n) = \sum_{C\in\mathscr{C}} f_C(X_C)$$

into an equivalent first-order energy

$$\tilde{E}(\tilde{X}) = \tilde{E}(X_1, \cdots, X_n, \cdots, X_m) = \sum g_v(X_v) + \sum h_{uv}(X_u, X_v)$$

- Adds variables
- More than 2 labels → Fusion moves

Functions of binary variables

- Pseudo-Boolean function (PBF)
  - Function of binary (**0** or **1**) variables
  - Can always write it uniquely as a polynomial
- One variable x : $E_0\,(1-x) + E_1\,x$
- Two variables x, y :

$$E_{00}\,(1-x)\,(1-y) + E_{01}\,(1-x)\,y + E_{10}\,x\,(1-y) + E_{11}\,x\,y$$

- Three variables x, y, z :

$$E_{000}\,(1-x)\,(1-y)\,(1-z) + E_{001}\,(1-x)\,(1-y)\,z + \dots + E_{111}\,x\,y\,z$$

- n^{th} order binary MRF = $(n+1)^{th}$ degree PBF

2nd-Order (Cubic) Case

Kolmogorov & Zabih. PAMI2004
Freedman & Drineas. CVPR2005

Reduce cubic PBF into quadratic one using

$$xyz = \max_{w\in\mathbb{B}} w(x+y+z-2)$$

$x\ y\ z$	
0 0 0	$0 = \max_{w\in\mathbb{B}} w\cdot(-2) = \max\{0,-2\} = 0$
0 0 1	$0 = \max_{w\in\mathbb{B}} w\cdot(-1) = \max\{0,-1\} = 0$
0 1 1	$0 = \max_{w\in\mathbb{B}} w\cdot 0 = 0$
1 1 1	$1 = \max_{w\in\mathbb{B}} w\cdot 1 = \max\{0,1\} = 1$

2nd-Order (Cubic) Case

$$xyz = \max_{w\in\mathbb{B}} w(x+y+z-2)$$

If $a<0$ $\quad a\,xyz = \min_{w\in\mathbb{B}} a\,w(x+y+z-2)$

Thus $\min_{x,y,z\in\mathbb{B}} a\,xyz = \min_{x,y,z,w\in\mathbb{B}} a\,w(x+y+z-2)$

So, in a minimization problem, we can substitute

$a\,xyz$ by $a\,w(x+y+z-2)$

Higher-Order Case

$$xyz = \max_{w\in\mathbb{B}} w(x+y+z-2)$$

$$xyzt = \max_{w\in\mathbb{B}} w(x+y+z+t-3)$$

$$xyztu = \max_{w\in\mathbb{B}} w(x+y+z+t+u-4)$$

$$x_1\cdots x_d = \max_{w\in\mathbb{B}} w\big(x_1+\cdots+x_d-(d-1)\big)$$

$$\min\ a\,x_1\cdots x_d = \min\ a\,w\big(x_1+\cdots+x_d-(d-1)\big)$$

if $a<0$

Higher-Order Case

- How about the case $a > 0$ and $d > 3$?
- *Imagine* such a formula:

$$xyzt = \min_{w \in \mathbb{B}} w(1^{\text{st}} \text{ degree}) + (2^{\text{nd}} \text{ degree})$$

- Notice LHS is symmetric
  - i.e., if we swap the value of two variables,
  - LHS is unchanged
- So RHS might be symmetric, too.

Symmetric polynomials

Fact

Any symmetric polynomial can be written as a polynomial in terms of elementary symmetric polynomials.

If $f(x, y, z, t)$ is quadratic and symmetric, it can be written with a polynomial $P(u,v)$:

$$f(x, y, z, t) = P(s_1, s_2)$$

$$\text{ESPs} \begin{cases} s_1 = x + y + z + t \\ s_2 = xy + yz + zx + xt + yt + zt \end{cases}$$

Quartic (degree 4) case

$$xyzt = \min_{w \in \mathbb{B}} w(1^{\text{st}} \text{ degree}) + (2^{\text{nd}} \text{ degree})$$

$$xyzt = \min_{w \in \mathbb{B}} w P(s_1) + Q(s_1, s_2)$$

$$P(s_1) = as_1 + b$$

$$Q(s_1, s_2) = \alpha s_1{}^2 + \beta s_1 + \gamma s_2 + \delta$$

$$\begin{cases} s_1 = x+y+z+t \\ s_2 = xy+yz+zx+xt+yt+zt \end{cases}$$

$$s_1{}^2 = (x+y+z+t)^2 = x^2 + y^2 + z^2 + t^2 + 2s_2$$

$$= s_1 + 2s_2 \quad (\text{since } x^2 = x,\ y^2 = y, \text{etc.})$$

$$Q(s_1, s_2) = cs_1 + ds_2 + e$$

Quartic (degree 4) case

$$xyzt = \min_{w \in \mathbb{B}} w(1^{\text{st}} \text{ degree}) + (2^{\text{nd}} \text{ degree})$$

$$xyzt = \min_{w \in \mathbb{B}} w(as_1 + b) + cs_1 + ds_2 + e$$

An exhaustive search for a, b, c, d, e yields

$$xyzt = \min_{w \in \mathbb{B}} w(-2s_1 + 3) + s_2$$

$$= \min_{w \in \mathbb{B}} w\big(-2(x + y + z + t) + 3\big)$$

$$+ xy + yz + zx + xt + yt + zt$$

Quintic (degree 5) case

Similarly,

$$xyztu = \min_{(v,w)\in\mathbb{B}^2} \{v(-2r_1+3)+w(-r_1+3)\}+r_2$$

$$\begin{cases} r_1 = x+y+z+t+u \\ r_2 = xy+yz+zx+xt+yt+zt+xu+yu+zu+tu \end{cases}$$

and so on, until one can guess…

General case

$$x_1\cdots x_d = \min_{w_1,\cdots,w_{n_d}} \left\{ \sum_{i=1}^{n_d} w_i\left(k_i^d(-S_1+2i)-1\right) \right\} + S_2$$

where

$$n_d = \left\lfloor \frac{d-1}{2} \right\rfloor \qquad k_i^d = \begin{cases} 1 & d \text{ is odd and } i = n_d \\ 2 & \text{otherwise} \end{cases}$$

$$S_1 = \sum_{i=1}^{d} x_i, \qquad S_2 = \sum_{i=1}^{d-1}\sum_{j=i+1}^{d} x_i x_j$$

Transformation

$$x_1\cdots x_d = \max_{w}\ w\left(S_1-(d-1)\right)$$

$$x_1\cdots x_d = \min_{w_1,\cdots,w_{n_d}} \left\{ \sum_{i=1}^{n_d} w_i\left(k_i^d(-S_1+2i)-1\right) \right\} + S_2$$

$$S_1 = \sum_{i=1}^{d} x_i, \qquad S_2 = \sum_{i=1}^{d-1}\sum_{j=i+1}^{d} x_i x_j$$

$$n_d = \left\lfloor \frac{d-1}{2} \right\rfloor \qquad k_i^d = \begin{cases} 1 & d \text{ is odd and } i = n_d \\ 2 & \text{otherwise} \end{cases}$$

Depending on the coefficient, take the one that has **min** and makes it into a part of overall minimization

Example

$$xyz = \min_{w\in\mathbb{B}} w\left(-(x+y+z)+1\right)+xy+yz+zx$$

$$xyzt = \min_{w\in\mathbb{B}} w\left(-2(x+y+z+t)+3\right) + xy+yz+zx+xt+yt+zt$$

$$\min_{x,y,z,t\in\mathbb{B}}\left(xy+yz+2xyz+3xyzt\right) =$$

$$\min_{x,y,z,t,v,w\in\mathbb{B}} \{xy+yz+2v\left(-(x+y+z)+1\right)+2(xy+yz+zx) +3w\left(-2(x+y+z+t)+3\right)+3\left(xy+yz+zx+xt+yt+zt\right)\}$$

Multiple labels: Fusion Move

Assume labels $L = \{l_1, \cdots, l_N\}$

Labeling Y assigns a label Y_v to each v

Fusion Move

Lempitsky et al. ICCV2007

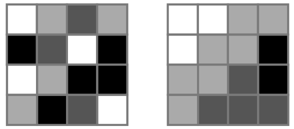
Iteratively update Y:

1. Generate a proposed labeling P
2. Merge Y and P

The merge defines a binary problem:

"For each v, change Y_v to P_v or not"

Multiple labels: Fusion Move



0	1	0	0
1	0	1	0
0	1	1	0
1	1	0	0

Fusion Move

Iteratively update Y:

1. Generate a proposed labeling P
2. Merge Y and P

The merge defines a binary problem:

"For each v, change Y_v to P_v or not"

Fusion Move with QPBO

QPBO (Roof duality)

Hammer et al. 1984, Boros et al. 1991, 2006

Kolmogorov & Rother PAMI2007, Rother et al. CVPR2007

Minimizes submodular E globally.

For non-submodular E, assigns each pixel

0, 1, or unlabeled

With fusion move, by not changing

unlabeled pixels to P, E doesn't increase

Example: Denoising by FoE

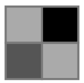
FoE (Fields of Experts) **Roth & Black CVPR2005**

A higher-order prior for natural images

$$E(Y) = \sum_{C \in \mathscr{C}} f_C(Y_C)$$

$\mathscr{C}$: a set of cliques

$Y_C = (Y_v)_{v \in C}$



$$f_C(Y_C) = \sum_{i=1}^{K} \alpha_i \log\left(1 + \frac{1}{2}(J_i \cdot Y_C)^2\right)$$

$$f_{\{v\}}(Y_{\{v\}}) = \frac{(N_v - Y_v)^2}{2\sigma^2}$$

Example: Denoising by FoE


Original


Noise-added


3rd order

Summary: Graph cuts

- **First order**: h_{uv} determines the applicability
  - Binary labels
    - Submodular: global minimization
    - Nonsubmodular: QPBO gives partial solution
  - More labels
    - Convex wrt linear order: global minimization
    - Approximation algorithms

      α-β swap ▪ α expansion ▪ fusion move
- **Higher order**
  - Binary labels: Transform to first order
  - More labels
    - Fusion move + transform binary energy to first order

Thank you very much

非常感謝您