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

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

$X_v = 0 \text{ or } 1$ for each pixel $v$
Example: Segmentation

• Find the $X$ that minimizes the energy

$$X_v = 0 \text{ or } 1 \text{ for each pixel } v$$

**Data term**

Based on the data and the image formation model

$$\sum_{v \in V} g_v(X_v) \geq 0$$

All pixels

''Faithful to data''

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

**Data**

$$\theta(c, 0)$$

$$\theta(c, 1)$$

Faithful to data

smooth

Example: Segmentation

• Find the $X$ that minimizes the energy

$$X_v = 0 \text{ or } 1 \text{ for each pixel } v$$

0 if neighboring labels coincide

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

All pixels

Neighboring pairs of pixels

smooth

Energy Minimization

• Consider the energy of the form

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

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 $64 \times 64$ 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
- $\mathcal{C}$: set of cliques in $V$
- Probability distribution of MRF
  $$P(X) = \frac{1}{Z} \prod_{C \in \mathcal{C}} q_c(X_C) = \frac{p(X)}{Z}$$
  $$Z = \sum_{X \in \mathcal{C}} \prod_{C \in \mathcal{C}} q_c(X_C) = \sum_{X \in \mathcal{C}} p(X)$$
  $$X_C = (X_v)_{v \in C}$$

First order MRF

- The simplest (interesting) MRF
  Geman & Geman 1984; Besag 1986
- Model neighborhood relation
  $$\mathcal{C} \approx V \cup E$$
  $$p(X) = \prod_{v \in V} \psi_v(X_v) \prod_{(u,v) \in E} \phi_{uv}(X_u, X_v)$$
- Energy
  $$E(X) = \sum_{C \in \mathcal{C}} f_C(X_C) = \sum_{v \in V} g_v(X_v) + \sum_{(u,v) \in E} h_{uv}(X_u, X_v)$$

Markov Random Field

- Rewrite the MRF probability distribution
  $$P(X) = \frac{1}{Z} \prod_{C \in \mathcal{C}} q_c(X_C) = \frac{p(X)}{Z}$$
  as
  $$p(X) = e^{-E(X)}$$
  $E(X) = \sum_{C \in \mathcal{C}} f_C(X_C)$

Graphs and cuts

- Directed graph $G = (V, E)$
  - $V$: Finite set of vertices
  - $E \subset V \times V$: edges
- Edges are weighted
  - $c : E \mapsto \mathbb{R}$
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 $\geq 0$, the minimum cut can be found efficiently

Labeling problem

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

Simplest example

- Energy minimization by minimum cuts (binary case)
  - $E(X) = \sum_{v \in \mathcal{V}} g_v(X_v) + \sum_{(u,v) \in E} \kappa |X_u - X_v|$
  - $g_v(0) \rightarrow X_v = 0$
  - $g_v(1) \rightarrow X_v = 1$

Energy minimization by minimum cuts (binary case)

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

Original Noise-added Denoised

Original Noise-added Denoised

Original Noise-added Denoised

Original Noise-added Denoised
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 $\rightarrow$ Energy minimization
- The weight must be $\geq 0$
Energy minimization by minimum cuts (binary case)

- The edge weight must be $\geq 0$
  - 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

\[
g_v(0) = -2 \quad 3 \quad g_v(1) = -5 \quad 0
\]

Energy minimization by minimum cuts (binary case)

- The edge weight must be $\geq 0$
  - 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)

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

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

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

In general:

$$h_{uv}(l_i, l_j) = \tilde{h}_{uv}(i - j), \tilde{h}_{uv} : \text{convex}$$
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 \( L = \{l_0, l_1, \ldots, l_k\} \)
  • Globally minimizeable ⇔ \( h_{uv}(l_i, l_j) \) is a convex function of \( i - j \)

• Approximation algorithms
  • αβ swap, α-expansion
  • Minimizes within factor \( c = \max_{u,v \in \mathcal{V}} \left( \frac{\max_{X_u \neq X_v} h_{uv}(X_u, X_v)}{\min_{X_u \neq X_v} h_{uv}(X_u, X_v)} \right) \)
  • \( c = 1 \) with the Potts model

First-order energy

Choose the move that minimizes best: binary optimization

≥ 3 labels, approximation

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

Move
• αβ swap
  • Allows label changes \( \alpha \rightarrow \beta, \beta \rightarrow \alpha \) only
• α-expansion
  • Allows changing to \( \alpha \) only

Example: Good (Low Energy) vs Bad (High Energy)

In each iteration, the \( \alpha \) area expands

Choose the move that minimizes best: binary optimization

Courtesy Yuri Boykov
Higher-order energy

\[ E(X) = \sum_{C \in \mathcal{C}} f_C(X_C) \]
\[ = \sum_{v \in V} g_v(X_v) + \sum_{(u,v)} h_{uv}(X_u, X_v) \]
\[ + \sum_{(u,v,s,t)} k_{uvst}(X_u, X_v, X_s, X_t) \]

Better (Lower Energy) \[\xleftrightarrow{\text{}}\] Worse (Higher Energy)

\[ \text{A} \quad \text{B} \quad \text{C} \quad \text{D} \]

---

Higher-order energy

- Transform arbitrary higher-order binary energy
  \[ E(X) = E(X_1, \cdots, X_n) = \sum_{C \in \mathcal{C}} f_C(X_C) \]
  into an equivalent first-order energy
  \[ \tilde{E}(\tilde{X}) = \tilde{E}(X_1, \cdots, X_n) = \sum_{v} g_v(X_v) + \sum_{uv} h_{uv}(X_u, X_v) \]
- 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 \( x, y, z \):
  \[ E_{000} (1-x) (1-y) (1-z) + E_{001} (1-x) (1-y) z + \ldots + E_{111} x y z \]
- \( n \)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

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

\[
\begin{align*}
x & y & z & \\
0 & 0 & 0 & = \max_{w \in B} w \cdot (-2) = \max \{0, -2\} = 0 \\
0 & 0 & 1 & = \max_{w \in B} w \cdot (-1) = \max \{0, -1\} = 0 \\
0 & 1 & 1 & = \max_{w \in B} w \cdot 0 = 0 \\
1 & 1 & 1 & = \max_{w \in B} w \cdot 1 = \max \{0, 1\} = 1
\end{align*}
\]

Higher-Order Case

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

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

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

\[
x_1 \cdots x_d = \max_{w \in B} w(x_1 + \cdots + x_d - (d - 1))
\]

\[
\min ax_1 \cdots x_d = \min a w(x_1 + \cdots + x_d - (d - 1))
\]

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 $P(u,v)$:

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

ESP's

$$s_1 = x + y + z + t$$

$$s_2 = xy + yz + zx + xt + yt + zt$$

Quartic (degree 4) case

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

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

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

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

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

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

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

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

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

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

$$+ xy + yz + zx + xt + yt + zt$$
Similarly, and so on, until one can guess…

Quintic (degree 5) case

\[
xyztu = \min_{(v, w) \in \mathbb{B}} \left\{ v(-2r_1 + 3) + w(-r_1 + 3) \right\} + r_2
\]

\[
\begin{align*}
  r_1 &= x + y + z + t + u \\
  r_2 &= xy + yz + zx + xt + yt + zt + xu + yu + zu + tu
\end{align*}
\]

and so on, until one can guess...

General case

\[
x_1 \cdots x_d = \min_{w_1, \ldots, w_d} \left\{ \sum_{i=1}^{n} w_i \left( k_i^d (-S_1 + 2i) - 1 \right) \right\} + S_2
\]

where

\[
n_d = \left\lfloor \frac{d-1}{2} \right\rfloor \quad k_i^d = \begin{dcases} 
  1 & d \text{ is odd and } i = n_d \\
  2 & \text{otherwise}
\end{dcases}
\]

\[
S_1 = \sum_{i=1}^{d} x_i, \quad S_2 = \sum_{i=1}^{d-1} \sum_{j=i+1}^{d} x_i x_j
\]

Transformation

\[
x_1 \cdots x_d = \max \ w \left( S_1 - (d - 1) \right)
\]

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

\[
S_1 = \sum_{i=1}^{d} x_i, \quad S_2 = \sum_{i=1}^{d-1} \sum_{j=i+1}^{d} x_i x_j
\]

Depending on the coefficient, take the one that has \textbf{min} and makes it into a part of overall minimization

Example

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

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

\[
\min_{x, y, z, t, v \in \mathbb{B}} \left\{ xy + yz + 2v(- (x + y + z) + 1) + 2(xy + yz + zx) + 3w(- 2(x + y + z + t) + 3) + 3(xy + yz + zx + xt + yt + zt) \right\}
\]
Multiple labels: Fusion Move

Assume labels \( L = \{l_1, \ldots, l_N\} \)
Labeling \( Y \) assigns a label \( Y_v \) to each \( v \)

Fusion Move \( \text{Lempitsky et al. ICCV2007} \)

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”

Fusion Move with QPBO

QPBO (Roof duality)
\( \text{Hammer et al. 1984, Boros et al. 1991, 2006} \)
\( \text{Kolmogorov & Rother PAMI2007, Rother et al. CVPR2007} \)

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) \( \text{Roth & Black CVPR2005} \)
A higher-order prior for natural images

\[
E(Y) = \sum_{C \in \mathcal{C}} f_C(Y_C) \quad \mathcal{C} \text{: a set of cliques}
\]
\[
Y_C = (Y_v)_{v \in C}
\]
\[
f_C(Y_C) = \sum_{i=1}^{K} \alpha_i \log \left( 1 + \frac{1}{2} (J_i \cdot Y_C)^2 \right)
\]
\[
f_{i,v}(Y_{\{v\}}) = \frac{(N_v - Y_v)^2}{2\sigma^2}
\]
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
      - $\alpha$-$\beta$ swap • $\alpha$ 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