Graph Cuts

Srikumar Ramalingam
School of Computing
University of Utah
Outline

• Introduction
• Pseudo-Boolean Functions
• Submodularity
• Max-flow / Min-cut Algorithm
• Alpha-Expansion
Segmentation Problem

[Boykov and Jolly’2001, Rother et al. 2004]
Stereo Reconstruction

- Choose the disparities from the discrete set: \((1, 2, \ldots, L)\)
Image Denoising

Original

Denoised image
Semantic Labeling (Building, ground, sky)

[Hoiem, Efros, Hebert, IJCV, 2007]
Image Labeling Problems

Assign a label to each image pixel

Geometry Estimation  Image Denoising  Object Segmentation  Depth Estimation
Labeling is highly structured

Possible labeling

Impossible labeling

Image Courtesy: Lubor Ladicky
Labeling is highly structured

- Labelings highly structured
- Labels highly correlated with very complex dependencies

- Neighbouring pixels tend to take the same label
- Low number of connected components
- Classes present may be seen in one image
- Geometric / Location consistency
- Planarity in depth estimation
- … many others (task dependent)
Image Labeling Problems

- Labelings highly structured
- Labels highly correlated with very complex dependencies
- Independent label estimation too hard
- Whole labelling should be formulated as one optimisation problem
- Number of pixels up to millions
  - Hard to train complex dependencies
  - Optimisation problem is hard to infer
Outline

• Introduction
• Pseudo-Boolean Functions
  • Submodularity
  • Max-flow / Min-cut Algorithm
• Alpha-Expansion
Pseudo Boolean Functions (PBF)

- Variables: $x_1, x_2, \ldots, x_n \in \{0,1\}$

- Negations: $\overline{x_i} = 1 - x_i \in \{0,1\}$

- Pseudo-Boolean Functions (PBF): \( f : \{0,1\}^n \rightarrow \mathbb{R} \)
  - Maps a Boolean vector to a real number.

- Has unique multi-linear representation:
  - For example:
    \[
    f(x_1, x_2, x_3, x_4) = 2 - 3x_2x_4 + 5x_1x_2x_3
    \]

[Boros&Hammer’2002]
Posiforms for Pseudo-Boolean functions (PBF)

• Posiforms: Non-negative multi-linear polynomial except maybe the constant terms.

\[ f(x_1, x_2, x_3, x_4) = 2 - 3x_2x_4 + 5x_1x_2x_3 \]
\[ = 2 - 3(1 - x_2)x_4 + 5x_1x_2x_3 \]
\[ = 2 - 3x_4 + 3x_2x_4 + 5x_1x_2x_3 \]
\[ = 2 - 3(1 - x_4) + 3x_2x_4 + 5x_1x_2x_3 \]
\[ \phi = -1 + 3x_4 + 3x_2x_4 + 5x_1x_2x_3 \]

• Several posiforms exist for a given function.
• Provides bounds for minimization, e.g. \( \phi = -1 \)

[Boros&Hammer‘2002]
Set Functions are Pseudo Boolean Functions (PBF)

- Finite ground set $V = \{1, 2, \ldots, n\}$

- Set function (Input - subset of $V$, output - real number)
  $$f_s : 2^V \rightarrow R$$

- 1-1 correspondence exists between $x_1, x_2, \ldots, x_n \in \{0, 1\}$ and subset $S$ of $V$.
  
  \[
  \begin{align*}
  V &= \{1, 2, 3, 4\} \\
  \{x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1\} &\iff (1, 2, 4) \\
  x_i = 1 &\iff i \in S \\
  x_i = 0 &\iff i \notin S
  \end{align*}
  \]
Set Functions are Pseudo Boolean Functions (PBF)

- Consider a PBF \( f(x_1, x_2, x_3, x_4) = 2 - 3x_2x_4 + 5x_2x_3 \)

- Equivalent to a set function

\[
\begin{align*}
  f_s (\{1,2\}) &= 2 - 3(1)(0) + 5(1)(0) = 2 \\
  f_s (\{2,3\}) &= 2 - 3(1)(0) + 5(1)(1) = 7
\end{align*}
\]
Outline

• Introduction
• Pseudo-Boolean Functions
• Submodularity
• Max-flow / Min-cut Algorithm
• Alpha-Expansion
Submodular set functions (Union-Intersection)

A set function $f : 2^V \rightarrow \mathbb{R}$ is submodular if and only if:

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B), \forall A, B \subseteq V$$
Equivalent Definitions

- **Diminishing gains:** for all $A \subseteq B$

\[
F(A \cup s) - F(A) \geq F(B \cup s) - F(B)
\]

Slide Courtesy: Krause, Jegelka
Questions

How do I prove my problem is submodular?

Why is submodularity useful?

Slide Courtesy: Krause, Jegelka
Submodularity Example

Example: costs

breakfast??

Slide Courtesy: Krause, Jegelka
Submodularity Example

Example: costs

breakfast??

ground set $\mathcal{V}$

Market 1

Market 2

Market 3

Slide Courtesy: Krause, Jegelka
Submodularity Example

**Example: costs**

- Breakfast: time to shop + price of items

\[
F(\text{breakfast}) = \text{cost(orange juice, bread, coffee)}
\]

\[
= t_1 + 1 + t_2 + 2
\]

\[
= \text{#shops} + \text{#items}
\]

Is \(F\) submodular?
Submodularity Example

Shared fixed costs

\[ \Delta(b \mid A) = 1 + t_3 \]
\[ \Delta(b \mid B) = 1 \]

marginal cost:  #new shops + #new items decreasing  \(\Rightarrow\) cost is submodular!

- shops: shared fixed cost
- economies of scale

Slide Courtesy: Krause, Jegelka
Set cover is submodular

\[ A = \{s_1, s_2\} \]

\[ F(A \cup \{s'\}) - F(A) \geq \]

\[ F(B \cup \{s'\}) - F(B) \]

\[ B = \{s_1, s_2, s_3, s_4\} \]
Submodular set functions (Union-Intersection)

\[ f(A) + f(B) \geq f(A \cup B) + f(A \cap B), \forall A, B \subseteq V \]

Let us consider a very simple case with only two variables \( x_1 \) and \( x_2 \).

\[ V = \{1, 2\}, A = \{1\}, B = \{2\} \]

Using submodularity, we have:

\[ f(x_1 = 1, x_2 = 0) + f(x_1 = 0, x_2 = 1) \geq f(x_1 = 1, x_2 = 1) + f(x_1 = 0, x_2 = 0) \]

\[ f(0,0) + f(0,1) \geq f(1,1) + f(0,0) \]

Main diagonal elements are smaller than off-diagonal ones. Blue is larger than red.
Quadratic Pseudo Boolean Functions (QPBF)

• Example of quadratic pseudo Boolean functions

\[ f(x_1, x_2, x_3, x_4) = 1 + x_1 - 3x_2 + x_1x_2 + 5x_3x_4 \]

[Boros&Hammer’2002]
Submodular Quadratic Pseudo Boolean Functions

- A QPBF is submodular if and only if all quadratic coefficients are non-positive.

\[ f_3(x_1, x_2, x_3) = 15 + x_1 - 3x_2 - x_1x_2 - 5x_2x_3 \]
Example for submodular QPBF

\[ f_3(x_1, x_2, x_3) = 15 + x_1 - 3x_2 - 3x_1x_3 - 5x_2x_3 \]

\[ V = \{1, 2, 3\}, A = \{1, 2\}, B = \{2, 3\} \]

\[ A \cup B = \{1, 2, 3\}, A \cap B = \{2\} \]

\[ f(A) = 15 + 1 - 3(1) - 3(1)(0) - 5(1)(0) = 13 \]

\[ f(B) = 15 + 0 - 3(1) - 3(0)(1) - 5(1)(1) = 7 \]

\[ f(A \cup B) = 15 + 1 - 3(1) - 3(1)(1) - 5(1)(1) = 5 \]

\[ f(A \cap B) = 15 + 0 - 3(1) - 3(0)(0) - 5(1)(0) = 12 \]

\[ \Rightarrow f(A) + f(B) \geq f(A \cup B) + f(A \cap B), (13 + 7 > 5 + 12) \]
Outline

• Introduction
• Pseudo-Boolean Functions
• Submodularity
  • Max-flow / Min-cut Algorithm
• Alpha-Expansion
Max-flow/Min-cut

Image courtesy: Lubor Ladicky
Max-flow/Min-cut

cost = 18

\[
\min_{S,T} \sum_{i \in S, j \in T} c_{ij}
\]

s.t.  \( s \in S, \ t \in T \)
Max-flow/Min-cut

\[
\begin{align*}
\text{min}_{S,T} & \sum_{i \in S} \sum_{j \in T} c_{ij} \\
\text{s.t.} & \quad s \in S, \quad t \in T
\end{align*}
\]
Max-flow/Min-cut

\[
\begin{align*}
\text{cost} &= 23 \\
\min_{S,T} \sum_{i \in S, j \in T} c_{ij} \\
\text{s.t. } & s \in S, \ t \in T
\end{align*}
\]
Network model for submodular QPBF

- A submodular QPBF $f$ can be associated with a network $G_v$.
- There is 1-1 correspondence every edge in network and every term in $f$.
- Let us denote source by $s = 0$ and sink by $t = 1$.
- An edge that goes from $x_1$ to $x_2$ is denoted by $x_1 x_2$.
Network model for submodular QPBF

- Given a QPBF we rewrite it using a posiform representation using only three types of terms:

\[ f = 3x_1 + x_2 - 4x_1x_2 \]
\[ f = 3x_1 + x_2 + (-4x_1x_2 + 4x_2 - 4x_2) \]
\[ f = 3x_1 + x_2 + 4(1-x_1)x_2 - 4x_2 \]
\[ f = 3x_1 - 3x_2 + 4x_1x_2 \]
\[ f = 3x_1 + (-3x_2 + 3 -3) + 4x_1x_2 \]
\[ f = -3 + 3x_1 + 3(1-x_2) + 4x_1x_2 \]
\[ f = -3 + 3sx_1 + 3x_2t + 4x_1x_2 \]
Network model for submodular QPBF

- There is a one-one correspondence between values of $f$ and s-t cut values of $G_v$. [Hammer 1965]

$$f(x_1 = 0, x_2 = 1) = C(\{x_1, x_2\}) = 4$$

s-t mincut

[Ford&Fulkerson’62, Goldberg&Tarzan86]
There is a one-one correspondence between values of $f$ and s-t cut values of $G_v$. [Hammer 1965]

Thus we can compute the minimum of $f$ using maxflow/mincut algorithm on the associated $G_v$.

$$f(x_1 = 1, x_2 = 0) =$$

$$C(\{x_2, s\}, \{x_1, t\}) = 3 + 3 = 6$$

s-t mincut
[Ford&Fulkerson’62, Goldberg&Tarzan86]
Network model for non-submodular QPBF

- A non-submodular QBPF $f$ can be associated with a network $G_v$ as follows:
  
  $$f = 3x_1 + x_2 + 4x_1x_2$$
  
  $$f = 3x_1 + 5x_2 - 4(1 - x_1)x_2$$
  
  $$f = 3sx_1 + 5sx_2 - 4x_1x_2$$

- There is no polynomial-time algorithm for s-t mincut on a network with negative edge capacities.
- A submodular QBPF can always be associated with a network with non-negative edge capacities.
Minimizing Quadratic Pseudo Boolean Functions

• If QPBF is submodular, use maxflow algo..  
  [Ford&Fulkerson’62, Goldberg&Tarzan86]

• If QPBF is non-submodular, Belief propagation or other message passing algorithms.  
  [Boros&Hammer’2002]
Multi-label Problems

• Choose the disparities from the discrete set: \((1,2,...,L)\)
Multi-label Problems

Exact Methods:

Transform the given multi-label problems to Boolean problems and solve them using maxflow/mincut algorithms or QPBO techniques. [Not covered in this course!]

Approximate Methods:

Develop iterative move-making algorithms where each move corresponds to a Boolean problem.
Outline

• Introduction
• Pseudo-Boolean Functions
• Submodularity
• Max-flow / Min-cut Algorithm
• Alpha-Expansion
Boolean Energy Function

• Variables $x_1, x_2, ..., x_n \in \{0,1\}$.

$$\theta^j_{x_i} \quad - \text{cost of assigning } x_i = j \in \{0,1\}.$$  

$$\theta^{lm}_{x_i x_j} \quad - \text{cost of jointly assigning } x_i = l \text{ and } x_j = m.$$  

Energy function:

$$E(x_1, x_2) = \sum_{j=0}^{1} \theta^j_{x_1} \delta^j_{x_1} + \sum_{j=0}^{1} \theta^j_{x_2} \delta^j_{x_2} + \sum_{i=0}^{1} \sum_{j=0}^{1} \theta^{ij}_{x_1 x_2} \delta^i_{x_1} \delta^j_{x_2}$$
Multi-label Energy Function

- Variables $y_1, y_2, ..., y_m \in \{0, 1, ..., L\}$.

$$\delta^l_{y_i} = \begin{cases} 1 & y_i = l \\ 0 & \text{otherwise.} \end{cases}$$

$$\theta^l_{y_i} - \text{cost for assigning a single variable} \quad y_i = l.$$  

$$\theta^{lm}_{y_i, y_j} - \text{cost of jointly assigning} \quad y_i = l \quad \text{and} \quad y_j = m.$$  

Energy function:

$$E(y_1, y_2) = \sum_{j=1}^{L} \theta^j_{y_1} \delta^j_{y_1} + \sum_{j=1}^{L} \theta^j_{y_2} \delta^j_{y_2} + \sum_{i=1}^{L} \sum_{j=1}^{L} \theta^{ij}_{y_1, y_2} \delta^i_{y_1} \delta^j_{y_2}.$$
Move Making Algorithms

[Image courtesy: Pushmeet Kohli, Phil Torr]
Move Making Algorithms

[Image courtesy: Pushmeet Kohli, Phil Torr]
$\alpha$ – Expansion

[Boykov et al. 2001]
\( \alpha \) – Expansion

- Let \( y_i \) and \( y_j \) be two adjacent variables whose labels are not \( \alpha \).

In the move space, we compute if the two variables should retain the same labels or move to label \( \alpha \).

[Boykov et al. 2001]
$\alpha - \text {Expansion}$

- In the move space, we use two Boolean variables $x_i$ and $x_j$ to denote $y_i$ and $y_j$ respectively. The encoding is shown below:

  \[
  y_i = l_i \iff x_i = 0 \quad y_j = l_j \iff x_j = 0 \\
  y_i = \alpha \iff x_i = 1 \quad y_j = \alpha \iff x_j = 1
  \]

- Submodularity condition states that the sum of main diagonal elements is less than or equal to the sum of elements in the off-diagonal:

\[
\begin{align*}
\theta^{00}_{x_i,x_j} & \quad \theta^{01}_{x_i,x_j} \\
\theta^{10}_{x_i,x_j} & \quad \theta^{11}_{x_i,x_j}
\end{align*}
\]

\[
\begin{align*}
\theta^{l_ia}_{y_i,y_j} & \quad \theta^{l_ia\alpha}_{y_i,y_j} \\
\theta^{la}_b & \quad \theta^{la\alpha}_b
\end{align*}
\]

$\Rightarrow$

\[
\begin{align*}
\theta^{00}_{x_i,x_j} & \quad \theta^{01}_{x_i,x_j} \\
\theta^{10}_{x_i,x_j} & \quad \theta^{11}_{x_i,x_j}
\end{align*}
\]

\[
\begin{align*}
\theta^{l_ia}_{y_i,y_j} & \quad \theta^{l_ia\alpha}_{y_i,y_j} \\
\theta^{la}_b & \quad \theta^{la\alpha}_b
\end{align*}
\]

[Boykov et al. 2001]
\( \alpha - \text{Expansion} \)

- Submodularity condition states that the sum of main diagonal elements is less than the sum of elements in the off-diagonal:

\[
\begin{array}{c|c|c}
\theta^{00}_{x_i x_j} & \theta^{01}_{x_i x_j} & \theta^{10}_{x_i x_j} \\
\hline
\theta^{01}_{x_i x_j} & \theta^{00}_{y_i y_j} & \theta^{11}_{x_i x_j} \\
\hline
\theta^{10}_{x_i x_j} & \theta^{11}_{x_i x_j} & \theta^{00}_{y_i y_j}
\end{array}
= 
\theta^{l_a l_b}_{y_i y_j} + \theta^{\alpha \alpha}_{y_i y_j} - \theta^{l_a \alpha}_{y_i y_j} - \theta^{\alpha l_b}_{y_i y_j} \leq 0
\]

If the multi-label potentials satisfy metric condition:

\[
\forall l_a, l_b \in L, \\
\theta^{l_a l_a}_{y_1 y_2} = 0, \\
\theta^{l_a l_b}_{y_1 y_2} = \theta^{l_a l_a}_{y_1 y_2} \geq 0, \\
\theta^{l_a l_b}_{y_1 y_2} + \theta^{l_b l_c}_{y_1 y_2} \geq \theta^{l_d l_d}_{y_1 y_2}
\]

[Boykov et al. 2001]
$\alpha -$ Expansion

Original Image  
Initial Solution  
After 1st expansion  
After 2nd expansion  
After 3rd expansion  
Final solution

[Image courtesy: Lubor Ladicky]  
[Boykov et al. 2001]
\( \alpha \beta \) Swap

- The variables having the labels \( \alpha \) and \( \beta \) can swap their labels or retain their previous states.

\[ y_i \xrightarrow{\text{retain}} \alpha \]
\[ y_j \xrightarrow{\text{retain}} \beta \]
\[ y_i \xrightarrow{\text{change}} \beta \]
\[ y_j \xrightarrow{\text{change}} \alpha \]

[Boykov et al. 2001]
\(\alpha \beta\) Swap

- In the move space, we use two Boolean variables \(x_i\) and \(x_j\) to denote \(y_i\) and \(y_j\) respectively. The encoding is shown below:

\[
\begin{align*}
y_i = \alpha & \iff x_i = 0 \\
y_j = \beta & \iff x_j = 0 \\
y_i = \beta & \iff x_i = 1 \\
y_j = \alpha & \iff x_j = 1
\end{align*}
\]

- Submodularity condition states that the sum of main diagonal elements is less than or equal to the sum of elements in the off-diagonal:

\[
\theta_{00}^{x_i x_j} \theta_{01}^{x_i x_j} = \theta_{\alpha \beta}^{y_i y_j} \theta_{\alpha \alpha}^{y_i y_j} = \theta_{\beta \beta}^{y_i y_j} \theta_{\beta \alpha}^{y_i y_j}
\]

[Boykov et al. 2001]
**αβ Swap**

- Submodularity condition states that the sum of main diagonal elements is less than or equal to the sum of elements in the off-diagonal:

\[
\begin{array}{cc}
\theta_{x_i x_j}^{00} & \theta_{x_i x_j}^{01} \\
\theta_{x_i x_j}^{10} & \theta_{x_i x_j}^{11}
\end{array}
\]

\[= \begin{array}{cc}
\theta_{y_i y_j}^{\alpha \beta} & \theta_{y_i y_j}^{\alpha \alpha} \\
\theta_{y_i y_j}^{\beta \beta} & \theta_{y_i y_j}^{\beta \alpha}
\end{array}\]

\[\theta_{y_i y_j}^{\alpha \beta} + \theta_{y_i y_j}^{\beta \alpha} - \theta_{y_i y_j}^{\alpha \alpha} - \theta_{y_i y_j}^{\beta \beta} \leq 0\]

- Semi-metric condition:

\[
\forall l_a, l_b \in L, \\
\theta_{y_i y_2}^{l_a} = 0, \\
\theta_{y_i y_2}^{l_a l_b} = \theta_{y_i y_2}^{l_b l_a} \geq 0
\]

[Boykov et al. 2001]
Foreground / Background Estimation using Graph Cuts

[Graph cuts slides adapted from Lubor Ladicky]
**Foreground / Background Estimation**

\[
E(x) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j)
\]

**Data term**

\[
\psi_i(x_i = 0) = -\log(p(x_i \notin FG))
\]

\[
\psi_i(x_i = 1) = -\log(p(x_i \in FG))
\]

**Smoothness term**

\[
\psi_{ij}(x_i, x_j) = K_{ij} \delta(x_i \neq x_j)
\]

where

\[
K_{ij} = \lambda_1 + \lambda_2 \exp(-\beta(I_i - I_j)^2)
\]

**Estimated using FG / BG colour models**

**Neighborhood terms**

Delta function is 1 when the condition is satisfied, and 0 otherwise

Parameters that are manually set or learned from data

**Intensity dependent smoothness**
Foreground / Background Estimation

\[ E(x) = \sum_{i \in V} \psi_i(x_i) + \sum_{i \in V, j \in N_i} \psi_{ij}(x_i, x_j) \]

**Data term**  **Smoothness term**

\[ x^* = \arg \min_{x \in L} E(x) \]
Foreground / Background Estimation

\[ E(x) = \sum_{i \in V} \psi_i(x_i) + \sum_{i \in V, j \in N_i} \psi_{ij}(x_i, x_j) \]

**Data term**  **Smoothness term**

\[ x^* = \arg \min_{x \in L} E(x) \]

How to solve this optimization problem?
Foreground / Background Estimation

\[ E(x) = \sum_{i \in V} \psi_i(x_i) + \sum_{i \in V, j \in N_i} \psi_{ij}(x_i, x_j) \]

Data term \hspace{1cm} Smoothness term

\[ x^* = \arg \min_{x \in L} E(x) \]

How to solve this optimization problem?

- Transform into min-cut / max-flow problem
- Solve it using min-cut / max-flow algorithm
What energy functions can be minimized using graph cuts?
http://www.cs.cornell.edu/~rdz/Papers/KZ-PAMI04.pdf

Boykov et al. Fast Approximate Energy Minimization via Graph Cuts, 1999

Tutorial on Energy functions (Longer version of the slides shown in the class with more details)

Ishikawa’s tutorial on graph cuts
Thank You