Graphical Models and Learning

Lecture 3: Maximum Flow, Minimum Cut

Yuliya Tarabalka
yuliya.tarabalka@inria.fr

Slides courtesy of Pawan Kumar
Outline

• Preliminaries
  – Functions and Excess Functions
  – s-t Flow
  – s-t Cut
  – Flows vs. Cuts

• Maximum Flow
• Algorithms
• Energy minimization with max flow/min cut
Context

- Example: network optimization problems

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Arcs</th>
<th>Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intersections</td>
<td>Roads</td>
<td>Vehicles</td>
</tr>
<tr>
<td>Airports</td>
<td>Air lanes</td>
<td>Aircraft</td>
</tr>
<tr>
<td>Switching points</td>
<td>Wires, channels</td>
<td>Messages</td>
</tr>
<tr>
<td>Pumping stations</td>
<td>Pipes</td>
<td>Fluids</td>
</tr>
<tr>
<td>Work centers</td>
<td>Materials-handling routes</td>
<td>Jobs</td>
</tr>
</tbody>
</table>
Maximum flow problem

• Applications
  – Maximize the flow through a company’s distribution network from factories to customers
  – Maximize the flow of oil through a system of pipelines
  – Maximize the flow of vehicles through a transportation network
Functions on Arcs

\[ D = (V, A) \]

Arc capacities \( c(a) \)

Function \( f: A \rightarrow \text{Reals} \)

Excess function \( E_f(v) \)

- Incoming value
- Outgoing value
Functions on Arcs

$D = (V, A)$

Arc capacities $c(a)$

Function $f: A \rightarrow \text{Reals}$

Excess function $E_f(v)$

$$\sum_{a \in \text{in-arcs}(v)} f(a) - \sum_{a \in \text{out-arcs}(v)} f(a)$$
Functions on Arcs

D = (V, A)
Arc capacities c(a)
Function f: A $\rightarrow$ Reals
Excess function $E_f(v)$

$$f(\text{in-arcs}(v)) - f(\text{out-arcs}(v))$$

$E_f(v_1) = -6$
Functions on Arcs

\[ D = (V, A) \]

Arc capacities \( c(a) \)

Function \( f: A \rightarrow \text{Reals} \)

Excess function \( E_f(v) \)

\[ E_f(v) = f(\text{in-arc}(v)) - f(\text{out-arc}(v)) \]

\[ E_f(v_2) = 14 \]
Excess Functions of Vertex Subsets

Excess function $E_f(U)$

Incoming Value
- Outgoing Value
Excess Functions of Vertex Subsets

Excess function $E_f(U)$

$$\sum_{a \in \text{in-arcs}(U)} f(a) - \sum_{a \in \text{out-arcs}(U)} f(a)$$
Excess Functions of Vertex Subsets

Excess function $E_f(U)$

$$f(\text{in-arcs}(U)) - f(\text{out-arcs}(U))$$

$E_f(\{v_1, v_2\}) = 8$
Excess Functions of Vertex Subsets

Excess function \( E_f(U) \)

\[
f(\text{in-arcs}(U)) - f(\text{out-arcs}(U))
\]

\[
E_f(\{v_1,v_2\}) = -6 + 14
\]

\[
E_f(U) = \sum_{v \in U} E_f(v)
\]
Outline

• Preliminaries
  – Functions and Excess Functions
  – s-t Flow
  – s-t Cut
  – Flows vs. Cuts

• Maximum Flow
• Algorithms
• Energy minimization with max flow/min cut
s-t Flow

Function flow: $A \rightarrow R$

Flow of arc $\leq$ arc capacity

Flow is non-negative

For all vertex except s,t

Incoming flow

= Outgoing flow
s-t Flow

Function flow: $A \rightarrow R$

$\text{flow}(a) \leq c(a)$

$\text{flow}(a) \geq 0$

For all $v \in V \setminus \{s,t\}$

$\sum_{(u,v) \in A} \text{flow}((u,v))$

$= \sum_{(v,u) \in A} \text{flow}((v,u))$
s-t Flow

Function flow: $A \rightarrow R$

$\text{flow}(a) \leq c(a)$

$\text{flow}(a) \geq 0$

For all $v \in V \setminus \{s,t\}$

$E_{\text{flow}}(v) = 0$
s-t Flow

Function flow: $A \rightarrow R$

$\text{flow}(a) \leq \text{c}(a)$

$\text{flow}(a) \geq 0$

For all $v \in V \setminus \{s,t\}$

$E_{\text{flow}}(v) = 0$
Function flow: $A \rightarrow R$

$\text{flow}(a) \leq c(a)$

$\text{flow}(a) \geq 0$

For all $v \in V \setminus \{s, t\}$

$E_{\text{flow}}(v) = 0$
s-t Flow

Function flow: \( A \rightarrow R \)

\[ \text{flow}(a) \leq c(a) \]

\[ \text{flow}(a) \geq 0 \]

For all \( v \in V \setminus \{s, t\} \)

\[ E_{\text{flow}}(v) = 0 \]

\( \checkmark \)
Value of s-t Flow

Outgoing flow of s
- Incoming flow of s
Value of s-t Flow

\[
\sum_{(s,v) \in A} \text{flow}((s,v)) - \sum_{(u,s) \in A} \text{flow}((u,s)) - E_{\text{flow}}(s) + E_{\text{flow}}(t)
\]
Value of s-t Flow

\[ \text{Value} = 1 \]

\[ \sum_{(s,v) \in A} \text{flow}((s,v)) - \sum_{(u,s) \in A} \text{flow}((u,s)) - E_{\text{flow}}(s) + E_{\text{flow}}(t) \]
Outline

• Preliminaries
  – Functions and Excess Functions
  – s-t Flow
  – s-t Cut
  – Flows vs. Cuts

• Maximum Flow
• Algorithms
• Energy minimization with max flow/min cut
Let $U$ be a subset of $V$

$C$ is a set of arcs such that

1. $(u,v) \in A$
2. $u \in U$
3. $v \in V \setminus U$

$C$ is a cut in the digraph $D = (V, A)$
What is $C$?

$D = (V, A)$

- $\{(v_1, v_2), (v_1, v_4)\}$
- $\{(v_1, v_4), (v_3, v_2)\}$
- $\{(v_1, v_4)\}$

$\checkmark$
What is $C$?

$D = (V, A)$

$\{(v_1,v_2),(v_1,v_4),(v_3,v_2)\}$ ?

$\{(v_4,v_3)\}$ ?

$\{(v_1,v_4),(v_3,v_2)\}$ ?
What is $C$?

$D = (V, A)$

$(v_1, v_2), (v_1, v_4), (v_3, v_2)$ ?

$(v_3, v_2)$ ?

$(v_1, v_4), (v_3, v_2)$ ?
Cut

\[ D = (V, A) \]

\[ C = \text{out-arcs}(U) \]
Capacity of Cut

Sum of capacity of all arcs in $C$
Capacity of Cut

\[ \sum_{a \in C} c(a) \]
Capacity of Cut

V_1 \rightarrow V_2: 10
V_1 \rightarrow V_3: 3
V_2 \rightarrow V_3: 2
V_3 \rightarrow V_4: 5
V_4 \rightarrow V_2: 3

U

V \setminus U
Capacity of Cut

U

V \ U

15

v_1 \rightarrow v_2 \quad 10

v_2 \rightarrow v_4 \quad 2

v_3 \rightarrow v_4 \quad 5

v_3 \rightarrow v_2 \quad 3

v_4 \rightarrow v_2 \quad 2
s-t Cut

\[ D = (V, A) \]

A source vertex “s”

A sink vertex “t”

C is a cut such that

- \( s \in U \)
- \( t \in V \setminus U \)

C is an s-t cut
Capacity of s-t Cut

\[ \sum_{a \in C} c(a) \]
Capacity of s-t Cut

The diagram shows a network with vertices s, v_1, v_2, v_3, v_4, and t. The capacities of the edges are as follows:

- s to v_1: 1
- v_1 to v_2: 6
- v_2 to t: 8
- v_1 to v_3: 3
- v_2 to v_4: 2
- v_3 to v_4: 5
- v_3 to t: 7
- v_4 to t: 3

The cut set is highlighted in blue, and the capacity of the cut is 5.
Capacity of s-t Cut
Outline

• Preliminaries
  – Functions and Excess Functions
  – s-t Flow
  – s-t Cut
  – Flows vs. Cuts

• Maximum Flow

• Algorithms

• Energy minimization with max flow/min cut
Flows vs. Cuts

An s-t flow function : $A \rightarrow \text{Reals}$

An s-t cut C such that $s \in U, t \in V \setminus U$

Value of flow $\leq$ Capacity of C
Flows vs. Cuts

Value of flow

\[ \text{Value of flow} = -E_{\text{flow}}(s) \]

\[ = -E_{\text{flow}}(s) - \sum_{v \in U \setminus \{s\}} E_{\text{flow}}(v) \]

\[ = -E_{\text{flow}}(U) \]

\[ = \text{flow(out-arcs(U))} - \text{flow(in-arcs(U))} \]

\[ \leq \text{Capacity of C} - \text{flow(in-arcs(U))} \]
Flows vs. Cuts

Value of flow

\[ \text{Value of flow} = -E_{\text{flow}}(s) \]

\[ = -E_{\text{flow}}(s) - \sum_{v \in U \setminus \{s\}} E_{\text{flow}}(v) \]

\[ = -E_{\text{flow}}(U) \]

\[ = \text{flow(out-arcs}(U)) - \text{flow(in-arcs}(U)) \]

\[ \leq \text{Capacity of } C \]

When does equality hold?
Flows vs. Cuts

Value of flow

\[ \text{Value of flow} = -E_{\text{flow}}(s) \]

\[ = -E_{\text{flow}}(s) - \sum_{v \in U \setminus \{s\}} E_{\text{flow}}(v) \]

\[ = -E_{\text{flow}}(U) \]

\[ = \text{flow(out-arcs(U))} - \text{flow(in-arcs(U))} \leq \text{Capacity of C} \]

flow(a) = c(a), a ∈ out-arcs(U)  \hspace{1cm} \text{flow(a)} = 0, a ∈ in-arcs(U)
Flows vs. Cuts

Value of flow

\[ \text{Value of flow} = -E_{\text{flow}}(s) \]

\[ = -E_{\text{flow}}(s) - \sum_{v \in U \setminus \{s\}} E_{\text{flow}}(v) \]

\[ = -E_{\text{flow}}(U) \]

\[ = \text{flow(out-arcs(U))} - \text{flow(in-arcs(U))} \]

\[ = \text{Capacity of C} \]

\[ \text{flow}(a) = c(a), \ a \in \text{out-arcs(U)} \quad \text{flow}(a) = 0, \ a \in \text{in-arcs(U)} \]
Outline

- Preliminaries

- **Maximum Flow**
  - Residual Graph
  - Max-Flow Min-Cut Theorem

- Algorithms

- Energy minimization with max flow/min cut
Maximum Flow Problem

Find the flow with the maximum value !!

\[ \sum_{(s,v) \in A} \text{flow}((s,v)) - \sum_{(u,s) \in A} \text{flow}((u,s)) \]

First suggestion to solve this problem !!
Passing Flow through s-t Paths

Find an s-t path where $\text{flow}(a) < c(a)$ for all arcs
Passing Flow through s-t Paths

Find an s-t path where \( \text{flow}(a) < c(a) \) for all arcs

Pass maximum allowable flow through the arcs
Passing Flow through s-t Paths

Find an s-t path where flow(a) < c(a) for all arcs

Pass maximum allowable flow through the arcs
Passing Flow through $s$-$t$ Paths

Find an $s$-$t$ path where $\text{flow}(a) < c(a)$ for all arcs.

No more paths. Stop.

Will this give us maximum flow? NO !!!
Passing Flow through s-t Paths

Find an s-t path where \( \text{flow}(a) < c(a) \) for all arcs

Pass maximum allowable flow through the arcs
Passing Flow through s-t Paths

Find an s-t path where $\text{flow}(a) < c(a)$ for all arcs

No more paths. Stop.

Another method?

Incorrect Answer !!
Outline

• Preliminaries

• Maximum Flow
  – Residual Graph
  – Max-Flow Min-Cut Theorem

• Algorithms

• Energy minimization with max flow/min cut
Residual Graph

Arcs where $\text{flow}(a) < c(a)$
Residual Graph

Including arcs to s and from t is not necessary
Inverse of arcs where flow(a) > 0
Maximum Flow using Residual Graphs

Start with zero flow.
Maximum Flow using Residual Graphs

Find an s-t path in the residual graph.
Maximum Flow using Residual Graphs

Find an s-t path in the residual graph.
Maximum Flow using Residual Graphs

For inverse arcs in path, subtract flow K.
Maximum Flow using Residual Graphs

Choose maximum allowable value of K.
For forward arcs in path, add flow K.
Maximum Flow using Residual Graphs

Choose maximum allowable value of K.
For forward arcs in path, add flow K.
Maximum Flow using Residual Graphs

Update the residual graph.
Maximum Flow using Residual Graphs

Find an s-t path in the residual graph.
Maximum Flow using Residual Graphs

Find an s-t path in the residual graph.
Choose maximum allowable value of K.
Add K to \((s,v_2)\) and \((v_1,t)\). Subtract K from \((v_1,v_2)\).
Maximum Flow using Residual Graphs

Choose maximum allowable value of $K$.
Add $K$ to $(s,v_2)$ and $(v_1,t)$. Subtract $K$ from $(v_1,v_2)$. 
Maximum Flow using Residual Graphs

Update the residual graph.
Maximum Flow using Residual Graphs

Find an s-t path in the residual graph.
Maximum Flow using Residual Graphs

No more s-t paths. Stop.
Maximum Flow using Residual Graphs

Correct Answer.
Maximum Flow using Residual Graphs

How can I be sure this will always work?
Outline

• Preliminaries

• Maximum Flow
  – Residual Graph
  – **Max-Flow Min-Cut Theorem**

• Algorithms

• Energy minimization with max flow/min cut
Let the subset of vertices $U$ be reachable from $s$.

$t$ is not in $U$.

Let the subset of vertices $U$ be reachable from $s$. 
Max-Flow Min-Cut

Or else a will be in the residual graph.

For all $a \in \text{out-arcs}(U)$, $\text{flow}(a) = c(a)$.
Max-Flow Min-Cut

For all $a \in \text{in-arc}(U)$, $\text{flow}(a) = 0$.

Or else inverse of $a$ will be in the residual graph.

For all $a \in \text{in-arc}(U)$, $\text{flow}(a) = 0$. 

Or else inverse of $a$ will be in the residual graph.
Max-Flow Min-Cut

For all $a \in \text{in-arcs}(U)$, $\text{flow}(a) = 0$.

For all $a \in \text{out-arcs}(U)$, $\text{flow}(a) = c(a)$.
Flows vs. Cuts

Value of flow

\[ \text{Value of flow} = -E_{\text{flow}}(s) \]

\[ = -E_{\text{flow}}(s) - \sum_{v \in U \setminus \{s\}} E_{\text{flow}}(v) \]

\[ = -E_{\text{flow}}(U) \]

\[ = \text{flow(out-arcs(U))} - \text{flow(in-arcs(U))} \]

\[ = \text{Capacity of C} \]

\[ \text{flow}(a) = c(a), \ a \in \text{out-arcs}(U) \]

\[ \text{flow}(a) = 0, \ a \in \text{in-arcs}(U) \]
Max-Flow Min-Cut

Minimum Cut

Capacity (C) = 

Maximum Flow

Value (flow)
Outline

• Preliminaries

• Maximum Flow

• **Algorithms**
  – *Ford-Fulkerson Algorithm*
  – *Dinitis Algorithm*

• Energy minimization with max flow/min cut
Ford-Fulkerson Algorithm

Start with flow = 0 for all arcs.

Find an s-t path in the residual graph.

Pass maximum allowable flow.

Subtract from inverse arcs.

Add to forward arcs.

REPEAT

Until s and t are disjoint in the residual graph.
Ford-Fulkerson Algorithm

Start with zero flow
Find an s-t path in the residual graph.
Ford-Fulkerson Algorithm

Find an s-t path in the residual graph.
Ford-Fulkerson Algorithm

Pass the maximum allowable flow.
Pass the maximum allowable flow.
Update the residual graph.
Ford-Fulkerson Algorithm

Find an s-t path in the residual graph.
Ford-Fulkerson Algorithm

Find an s-t path in the residual graph.
Ford-Fulkerson Algorithm

Complexity is exponential in $k$. 
Irrational arc lengths can lead to infinite iterations.

For examples, see Uri Zwick, 1993
Ford-Fulkerson Algorithm

Choose wisely.

There are good paths and bad paths.
Outline

- Preliminaries
- Maximum Flow

- **Algorithms**
  - Ford-Fulkerson Algorithm
  - Dinits Algorithm

- Energy minimization with max flow/min cut
Dinics Algorithm

Start with flow = 0 for all arcs.

Find the **minimum s-t path** in the residual graph.

Pass maximum allowable flow.

- Subtract from inverse arcs.
- Add to forward arcs.

Until s and t are disjoint in the residual graph.
Dinits Algorithm

Start with zero flow
Find the minimum s-t path in the residual graph.
Find the minimum s-t path in the residual graph.
Dinits Algorithm

Pass the maximum allowable flow.
Dinits Algorithm

Pass the maximum allowable flow.
Dinits Algorithm

Update the residual graph.
Find the minimum s-t path in the residual graph.
Find the minimum s-t path in the residual graph.
Pass the maximum allowable flow.
Dinits Algorithm

Pass the maximum allowable flow.
Dinits Algorithm

Update the residual graph.
Dinits Algorithm

No more s-t paths. Stop.
## Solvers for the Minimum-Cut Problem

### Augmenting Path and Push-Relabel

<table>
<thead>
<tr>
<th>year</th>
<th>discoverer(s)</th>
<th>bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1951</td>
<td>Dantzig</td>
<td>$O(n^2 m U)$</td>
</tr>
<tr>
<td>1955</td>
<td>Ford &amp; Fulkerson</td>
<td>$O(m^2 U)$</td>
</tr>
<tr>
<td>1970</td>
<td>Dinitz</td>
<td>$O(n^2 m)$</td>
</tr>
<tr>
<td>1972</td>
<td>Edmonds &amp; Karp</td>
<td>$O(m^2 \log U)$</td>
</tr>
<tr>
<td>1973</td>
<td>Dinitz</td>
<td>$O(n m \log U)$</td>
</tr>
<tr>
<td>1974</td>
<td>Karzanov</td>
<td>$O(n^3)$</td>
</tr>
<tr>
<td>1977</td>
<td>Cherkassky</td>
<td>$O(n^2 m^{1/2})$</td>
</tr>
<tr>
<td>1980</td>
<td>Galil &amp; Naamad</td>
<td>$O(n m \log^2 n)$</td>
</tr>
<tr>
<td>1983</td>
<td>Sleator &amp; Tarjan</td>
<td>$O(n m \log n)$</td>
</tr>
<tr>
<td>1986</td>
<td>Goldberg &amp; Tarjan</td>
<td>$O(n m \log(n^2/m))$</td>
</tr>
<tr>
<td>1987</td>
<td>Ahuja &amp; Orlin</td>
<td>$O(n m + n^2 \log U)$</td>
</tr>
<tr>
<td>1987</td>
<td>Ahuja et al.</td>
<td>$O(n m \log(n \sqrt{\log U}/m))$</td>
</tr>
<tr>
<td>1989</td>
<td>Cheriyan &amp; Hagerup</td>
<td>$E(n m + n^2 \log^2 n)$</td>
</tr>
<tr>
<td>1990</td>
<td>Cheriyan et al.</td>
<td>$O(n^3/\log n)$</td>
</tr>
<tr>
<td>1990</td>
<td>Alon</td>
<td>$O(n m + n^{8/3} \log n)$</td>
</tr>
<tr>
<td>1992</td>
<td>King et al.</td>
<td>$O(n m + n^{2+\epsilon})$</td>
</tr>
<tr>
<td>1993</td>
<td>Phillips &amp; Westbrook</td>
<td>$O(n m(\log_{m/n} n + \log^{2+\epsilon} n))$</td>
</tr>
<tr>
<td>1994</td>
<td>King et al.</td>
<td>$O(n m \log_{m/(n \log n)} n)$</td>
</tr>
<tr>
<td>1997</td>
<td>Goldberg &amp; Rao</td>
<td>$O(m^{3/2} \log(n^2/m) \log U)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$O(n^{2/3} m \log(n^2/m) \log U)$</td>
</tr>
</tbody>
</table>

$n$: #nodes  
$m$: #arcs  
$U$: maximum arc length

[Slide credit: Andrew Goldberg]
Max-Flow in Computer Vision

- Specialized algorithms for vision problems
  - Grid graphs
  - Low connectivity ($m \sim O(n)$)

- Dual search tree augmenting path algorithm
  [Boykov and Kolmogorov PAMI 2004]
  - Finds approximate shortest augmenting paths efficiently
  - High worst-case time complexity
  - Empirically outperforms other algorithms on vision problems
  - Efficient code available on the web

http://www.adastral.ucl.ac.uk/~vladkolm/software.html
Outline

• Preliminaries

• Maximum Flow

• Algorithms

• Energy minimization with max flow/min cut
  – Two-Label Energy Functions
Interactive Binary Segmentation

Foreground histogram of RGB values FG

Background histogram of RGB values BG

‘1’ indicates foreground and ‘0’ indicates background
Interactive Binary Segmentation

More likely to be foreground than background
Interactive Binary Segmentation

\[ \theta_p(0) \text{ proportional to } -\log(BG(d_a)) \]
\[ \theta_p(1) \text{ proportional to } -\log(FG(d_a)) \]

More likely to be background than foreground
Interactive Binary Segmentation

More likely to belong to same label
Interactive Binary Segmentation

\[ \theta_{pq}(i,k) \text{ proportional to } \exp(- (d_a - d_b)^2) \text{ if } i \neq k \]

\[ \theta_{pq}(i,k) = 0 \text{ if } i = k \]

Less likely to belong to same label
Overview

Energy $E$

- One vertex per random variable
- + Additional vertices “s” and “t”

Digraph $D$

- Compute Minimum Cut

Labeling $\mathbf{x}$

- $v_p \in U$ implies $x_p = 0$
- $v_p \in V \setminus U$ implies $x_p = 1$

$U$ and $V \setminus U$
Digraph for Unary Potentials

\[ A \quad x_p = 0 \]

\[ B \quad x_p = 1 \]

\[ \theta_p(0) \]

\[ \theta_p(1) \]

\[ X_p \]
Digraph for Unary Potentials

\[ x_p = 0 \]

\[ x_p = 1 \]
Digraph for Unary Potentials

Let $A \geq B$

$x_p = 0$

$x_p = 1$

Constant

$A-B$

$0$

$B$

$A-B$

$0$

$v_p$

$s$

$t$

$0$

$B$

$B$
Let $A \geq B$

$x_p = 0$

$x_p = 1$

Let $A \geq B$

$x_p = 0$

$x_p = 0$

$A-B$

$A-B$
Let $A < B$

$x_p = 0$

$x_p = 1$

Let $A < B$

$x_p = 1$

$B - A$

Digraph for Unary Potentials
Digraph for Unary Potentials

Let $A < B$

$x_p = 0$

$0$

$x_p = 1$

Constant

$A$

$B - A$

$s$

$v_p$

$B - A$

$t$
Digraph for Pairwise Potentials

\[ x_p = 0 \quad x_p = 1 \]

\[ x_q = 0 \quad A \quad C \]

\[ x_q = 1 \quad B \quad D \]

\[ \theta_{pq}(1,1) \quad \theta_{pq}(0,1) \quad \theta_{pq}(1,0) \quad \theta_{pq}(0,0) \]

\[ X_p \quad X_q \]

\[ A + A + A + A \]

\[ 0 + 0 + 0 + 0 \]

\[ B-A + B-A + 0 + 0 \]

\[ 0 + D-B + D-B + 0 \]

\[ 0 + C+B-D-A + 0 + 0 \]
Digraph for Pairwise Potentials

\[
\begin{array}{c}
\text{x}_p = 0 & \text{x}_p = 1 \\
\text{x}_q = 0 & A & C \\
& B & D \\
\text{x}_q = 1 & & \\
\end{array}
\]

\[x_p = 0 \quad x_p = 1\]

\[
\begin{align*}
& + \quad 0 & 0 \\
& + \quad B-A & B-A \\
\end{align*}
\]

\[
\begin{align*}
& + \quad 0 & D-B \\
& + \quad 0 & D-B \\
\end{align*}
\]

\[
\begin{align*}
& + \quad 0 & C+B-D-A \\
& + \quad 0 & 0 \\
\end{align*}
\]

Constant

\[s\]

\[v_p\]

\[v_q\]

\[t\]
Digraph for Pairwise Potentials

Unary Potential $x_q = 1$

\[
\begin{array}{cc}
0 & 0 \\
B-A & B-A
\end{array}
\]

\[
\begin{array}{cc}
0 & D-B \\
0 & D-B
\end{array}
\]

\[
\begin{array}{cc}
0 & C+B-D-A \\
0 & 0
\end{array}
\]

$B-A$
Digraph for Pairwise Potentials

Unary Potential

\( x_p = 1 \)

\[
\begin{array}{c|c|c}
0 & D-B & + \\
0 & D-B & 0 \\
0 & C+B-D-A & 0 \\
\end{array}
\]
Digraph for Pairwise Potentials

Pairwise Potential

- \( x_p = 1, x_q = 0 \)
Digraph for Pairwise Potentials

\[ x_p = 0 \quad x_p = 1 \]
\[ x_q = 0 \quad A \quad C \]
\[ B \quad D \quad x_q = 1 \]

\[ C + B - D - A \geq 0 \]

Submodular Energy

General 2-label MAP estimation is NP-hard
Results – Image Segmentation

Boykov and Jolly, ICCV 2001
Results – Image Segmentation

Boykov and Jolly, ICCV 2001
Results – Image Segmentation

Boykov and Jolly, ICCV 2001
Results – Image Synthesis

Kwatra et al., SIGGRAPH 2003
Results – Image Synthesis

Kwatra et al., SIGGRAPH 2003
Outline

• Preliminaries

• Maximum Flow

• Algorithms

• Energy minimization with max flow/min cut
  – Multi-Label Energy Functions
St-mincut based Move algorithms

\[ E(x) = \sum_{i} \theta_i(x_i) + \sum_{i,j} \theta_{ij}(x_i, x_j) \]

\[ x \in \text{Labels } L = \{l_1, l_2, \ldots, l_k\} \]

• Commonly used for solving non-submodular multi-label problems
• Extremely efficient and produce good solutions
• Not Exact: Produce local optima
Move-Making Algorithms

Space of All Labelings
Computing the Optimal Move

Key Property

Move Space

Energy

Solution Space

Current Solution

Search Neighbourhood

Optimal Move

Bigger move space

- Better solutions
- Finding the optimal move hard
Moves using Graph Cuts

Expansion and Swap move algorithms

[Boykov Veksler and Zabih, PAMI 2001]

- Makes a series of changes to the solution (moves)
- Each move results in a solution with smaller energy

Space of Solutions (x) : $L^N$

Move Space (t) : $2^N$

Current Solution
Search Neighbourhood

N  Number of Variables
L  Number of Labels
Moves using Graph Cuts

Expansion and Swap move algorithms
[Boykov Veksler and Zabih, PAMI 2001]

- Makes a series of changes to the solution (moves)
- Each move results in a solution with smaller energy

Current Solution

Move to new solution

Construct a move function

Minimize move function to get optimal move

How to minimize move functions?
Expansion Algorithm

Variables take label $l_\alpha$ or retain current label

Slide courtesy Pushmeet Kohli
Expansion Algorithm

Initialize labeling $x = x^0$ (say $x^0_p = 0$, for all $X_p$)

For $\alpha = 1, 2, \ldots, h-1$

$$x^\alpha = \operatorname{argmin}_{x'} E(x')$$

s.t. $x'_p \in \{x_p\} \cup \{l_\alpha\}$

Update $x = x^\alpha$

End

Boykov, Veksler and Zabih, 2001
Expansion Algorithm

Restriction on pairwise potentials?

- Move energy is submodular if:
  - Unary Potentials: Arbitrary
  - Pairwise potentials: Metric

\[ \theta_{ij}(l_a, l_b) + \theta_{ij}(l_b, l_c) \geq \theta_{ij}(l_a, l_c) \]

Example: Potts model

[Boykov, Veksler, Zabih]
Swap Move

- Variables labeled $\alpha, \beta$ can swap their labels

Swap Sky, House

[Boykov, Veksler, Zabih]
Swap Move

- Variables labeled $\alpha, \beta$ can swap their labels

- Move energy is submodular if:
  - Unary Potentials: Arbitrary
  - Pairwise potentials: Semimetric

\[
\theta_{ij}(l_a, l_b) \geq 0
\]
\[
\theta_{ij}(l_a, l_b) = 0 \quad \text{if} \quad a = b
\]

Example: Potts model

[Boykov, Veksler, Zabih]
General Binary Moves

\[ x = t \ x^1 + (1-t) \ x^2 \]

Minimize over move variables \( t \)

<table>
<thead>
<tr>
<th>Move Type</th>
<th>First Solution</th>
<th>Second Solution</th>
<th>Guarantee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expansion</td>
<td>Old solution</td>
<td>All alpha</td>
<td>Metric</td>
</tr>
<tr>
<td>Fusion</td>
<td>Any solution</td>
<td>Any solution</td>
<td>( \times )</td>
</tr>
</tbody>
</table>
Solving Continuous Problems using Fusion Move

\[ x = t x^1 + (1-t) x^2 \]

Optical Flow Example

Solution from Method 2

Solution from Method 1

Final Solution

(Lempitsky et al. CVPR08, Woodford et al. CVPR08)
Results – Denoising + Inpainting
Results – Denoising + Inpainting
Results – Denoising + Inpainting
Results – Denoising + Inpainting
Paper presentation