Submodular Minimisation using Graph Cuts

Pankaj Pansari

18 April, 2016
Overview

- Graph construction to minimise special class of submodular functions
- For this special class, submodular minimisation translates to constrained modular minimisation
- Given a submodular function via an oracle, NP-hard to determine if graph-representable
Outline

Introduction

Graph Construction and NP-hardness

Modular Minimisation and Minimum st-cuts
Part I - Introduction

- Submodular, modular and supermodular functions
- Cut functions, st-cut functions, minimum st-cuts
- Lattice
All discrete optimization problems are of the form:

\[ \max \{ f(X) : X \in \mathcal{F} \} \]

\[ \min \{ f(X) : X \in \mathcal{F} \} \]

where \( \mathcal{F} \) is a discrete set of feasible solutions and \( f \) is a set function, that is, \( f : 2^\mathcal{F} \rightarrow \mathbb{R} \).

We can try to

- deal with each problem individually, or
- capture some properties of \( f, \mathcal{F} \) that make it tractable
Submodular Functions

Equivalent definitions:

1. Define the *marginal value* of element \( j \):

\[
f_X(j) = f(X \cup \{j\}) - f(X)
\]

\( f \) is submodular if \( \forall X \subset Y, j \notin Y \):

\[
f_X(j) \geq f_Y(j)
\]

2. The set function \( f \) is submodular if for any \( X, Y \)

\[
f(X \cup Y) + f(X \cap Y) \leq f(X) + f(Y)
\]
Modular Functions

Equivalent definitions:

1. $f$ is modular if $\forall X \subset Y, j \notin Y$:

$$f_X(j) = f_Y(j)$$

2. $f$ is modular if for any $X, Y$

$$f(X \cup Y) + f(X \cap Y) = f(X) + f(Y)$$
Supermodular Functions

Equivalent definitions:

1. $f$ is supermodular if $\forall X \subset Y, j \notin Y$:

   $$f_X(j) \leq f_Y(j)$$

2. $f$ is supermodular if for any $X, Y$

   $$f(X \cup Y) + f(X \cap Y) \leq f(X) + f(Y)$$
An Observation

The whole is ______ the sum of its parts

Let $f(\emptyset) = 0$:

- Modular: “equal to”
  \[ f(A) = \sum_{i \in A} f(i) \]

- Submodular: “less than”
  \[ f(A) \leq \sum_{i \in A} f(i) \]

- Supermodular: “greater than”
A family $\mathcal{F}$ is a lattice if it is closed under union and intersection:

$$A, B \in \mathcal{F} \implies A \cup B \in \mathcal{F} \text{ and } A \cap B \in \mathcal{F}$$

**Example:** Let $S = \{a, b, c\}$

<table>
<thead>
<tr>
<th>$\mathcal{F}$</th>
<th>Lattice?</th>
</tr>
</thead>
<tbody>
<tr>
<td>${\phi, {a}, {b}, {a, b}}$</td>
<td>Yes</td>
</tr>
<tr>
<td>${\phi, {a, b}, {c}, {a, b, c}}$</td>
<td>Yes</td>
</tr>
<tr>
<td>${{a}, {c}, {a, c}}$</td>
<td>No</td>
</tr>
</tbody>
</table>

Derive lattices for $S = \{a, b, c, d\}$. 
Proper Lattice

\( \mathcal{F} \) is a proper lattice if
\( \cup \mathcal{F} = \emptyset \) and \( \cap \mathcal{F} = S \)
\( \implies \emptyset, S \in \mathcal{F} \)

<table>
<thead>
<tr>
<th>( \mathcal{F} )</th>
<th>Proper Lattice?</th>
</tr>
</thead>
<tbody>
<tr>
<td>( {\emptyset, {a}, {b}, {a, b}} )</td>
<td>Yes</td>
</tr>
<tr>
<td>( {{a}, {a, b}, {a, c}, {a, b, c}} )</td>
<td>No</td>
</tr>
</tbody>
</table>
Oracle

A black box which computes output $f(x)$ for any input $x$
Cut Functions

Given $G = (V, A)$, a cut is a partition of $V$ into $(X, \bar{X})$, $X \subset V$

$$f(X) = \sum_{i \in X, j \in \bar{X}} a_{ij}$$
Example

A directed graph with labeled edges:

- edge from e to a labeled 21
- edge from a to b labeled 5
- edge from a to c labeled 10
- edge from b to d labeled 10
- edge from d to f labeled 12
- edge from c to d labeled 6
- edge from c to f labeled 18
- edge from d to e labeled 30

Total paths and values:

- Path e to f via c and d: 21 + 10 + 12 = 43
- Path e to f via a and b: 21 + 5 + 12 = 38
- Path e to f via a and c: 21 + 10 = 31

Total values for e to f:

= 51

Total values for e to f:

= 28

- Path e to f via a and b: 21 + 5 + 12 = 38
- Path e to f via a and c: 21 + 10 = 31

Total values for e to f:

= 51

Total values for e to f:

= 28
• Cut functions are submodular
  (Proof on board)
Minimum Cut

- Trivial solution:
  \[ f(\phi) = 0 \]

- Need to enforce \( X, \bar{X} \) to be non-empty
  Source \( \{s\} \in X \), Sink \( \{t\} \in \bar{X} \)
\( f(X) = \sum_{i \in X, j \in \bar{X}} a_{ij} \)

\( \{s\} \in X, \{t\} \text{ in } \bar{X} \)
Minimum \textit{st}-Cut

\[ \min f(X) = \sum_{i \in X, j \in \bar{X}} a_{ij} \]

such that \( \{s\} \in X, \{t\} \text{ in } \bar{X} \)

Min cut value = 18
Given set $S$ and submodular function $f$, submodular minimisation is

$$\min_{A \subseteq S} f(A)$$

### Continuous
- Ellipsoid
  - $O(|S|^k \log(\max |f(A)|))$
- Min-norm point
  - $O(|S|^7)$

### Discrete
- Minimum Cut
  - $O(|S|^3)$
Introduction

Graph Construction and NP-hardness

Modular Minimisation and Minimum st-cuts
Part II - Graph Construction and NP-hardness

Is $f$ a cut function?

- **No**: NP-hard
- **Yes**: Polynomial time algorithm

Graph

Note: $f$ values available only via oracle
A Property of Cut Functions

For any three disjoint subsets $A$, $B$, $C$ of $S$

$$f(A \cup B \cup C) = f(A \cup B) + f(B \cup C) + f(C \cup A) - f(A) - f(B) - f(C) + f(\phi)$$

(Proof outline on board)

$\implies f$ is determined by its values on sets of cardinality at most 2
Recipe to construct directed graph $G = (V, A)$

- $V = S \cup \{s, t\}$
- To specify $A$
  1. Source to vertex arcs $a_{sv}$
  2. Vertex to vertex arcs $a_{uv}$
  3. Vertex to sink arcs $a_{vt}$

Graph construction and example on board
To certify if a submodular function is cut function

...is \textit{NP}-hard

\(\implies\) exponential oracle calls required to certify

Proof on board
Outline

Introduction

Graph Construction and NP-hardness

Modular Minimisation and Minimum \( st \)-cuts
Part III - Modular Minimisation and Minimum $st$-cuts

- Modular minimisation over lattice
- Minimum $st$-cut
Modular min over a lattice

Modular minimisation

- over power set $2^S$ is trivial
- over a lattice $\mathcal{F}$ is harder
Closure

Given \( G = (V, A), X \subseteq V \) is a closure if

\[ \delta(X) = \phi \]

\[ \implies \text{No outcoming arcs from } X \]
Minimum Weight Closure

Given $G = (V, A)$ with weights $w_v$ for $v \in V$

$$\min_{v \in X} \min w_v$$

s.t $u(\delta(X)) = \phi$
Modular min over a lattice as $st$-min cut

Strategy:
- Modular min over a lattice $\rightarrow$ Min weight closure
- Min weight closure $\rightarrow$ Min $st$-cut
Modular min over a lattice → Min weight closure

On board
Modular min over a lattice as $st$-min cut

Strategy:

- Modular min over a lattice $\rightarrow$ Min weight closure
- Min weight closure $\rightarrow$ Min $st$-cut
Min weight closure $\rightarrow$ Min \textit{st}-cut

On board
st-min cut as modular min over a lattice

On board
Reference