Submodular Maximization

advances in distributed/streaming computing

Jiecao Chen

Indiana University Bloomington

jiecchen@indiana

March 17, 2016

Overview

1. Introduction to Submodularity
   - Definitions
   - Properties
   - Constraints
   - Algorithms
2. Applications
3. Streaming Submodular Maximization
4. Distributed Submodular Maximization

Definitions of Submodularity

Definition (submodular concave)

A function $f : 2^V \to \mathbb{R}$ is submodular if for any $A, B \subseteq V$, we have that:

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B). \tag{1}$$

An alternate equivalent definition is more interpretable in many situations.

Definition (diminishing returns)

A function $f : 2^V \to \mathbb{R}$ is submodular if for any $A \subseteq B \subset V$, and $v \in V \backslash B$, we have that:

$$f(A + v) - f(A) \geq f(B + v) - f(B). \tag{2}$$

Modular Functions

Definition (Modularity)

A function $f : 2^V \to \mathbb{R}$ is modular if for any $A \subseteq B \subset V$, and $v \in V \backslash B$, we have that:

$$f(A + v) - f(A) = f(B + v) - f(B). \tag{3}$$

Notably, a modular function f can always be written as

$$f(S) = f(\emptyset) + \sum_{v \in S} (f(\{v\}) - f(\emptyset))$$

for any $S \subseteq V$. If we further assume $f(\emptyset) = 0$ (in this case, we call f normalized or proper), we have a simplified expression,

$$f(S) = \sum_{v \in S} f(\{v\}).$$

Monotonitcity

Definition (Monotonitcity)

A set function $f : 2^V \to \mathbb{R}$ is said to be non-decreasing if for any $A \subseteq B \subseteq V$, $f(A) \leq f(B)$. Non-increasing set functions are defined in the similar way.

When we say a submodular function is monotone, we mean it is non-decreasing.

Properties

Submodularity is closed under addition.

Property

Let $f_1, f_2 : 2^V \to \mathbb{R}$ *be two submodular functions. Then*

$$f : 2^V \to \mathbb{R} \quad \textit{with} \quad f(A) = \alpha f_1(A) + \beta f_2(A)$$

is submodular for any fixed $\alpha, \beta \in \mathbb{R}^+$.

Submodularity is preserved under restriction.

Property

Let $f : 2^V \to \mathbb{R}$ *be a submodular function. Let* $S \subseteq V$ *be a fixed set. Then*

$$f' : 2^V \to \mathbb{R} \quad \textit{with} \quad f'(A) = f(A \cap S)$$

is submodular.

Properties cont.

The following property can be useful when we show that the negative of the objective function of k-median problem is submodular.

Property

Consider V as a set of indices. Let $\mathbf{c} \in \mathbb{R}^V$ be a fixed vector, c_i its ith coordinate. Then

$$f : 2^V \to \mathbb{R} \quad \textit{with} \quad f(A) = \max_{j \in A} c_j$$

is submodular.

Constraints

Submodular Maximization Problem

A submodular maximization problem usually has the following form:

$$\arg\max_{I \in \mathcal{I}} f(I), \tag{4}$$

where f is a submodular function and $\mathcal{I} \subseteq 2^V$ is the collection of all feasible solutions. We call $\mathcal{I}$ the constraint of the optimization problem.

Constraints

$\mathcal{I}$ is important!

The structure of $\mathcal{I}$ plays a crucial role in submodular optimization:

- Different constraints have different hardness results.
- Normally the difficulty increases when the constraint becomes more general.

Constraints

$\mathcal{I}$ is important!

The structure of $\mathcal{I}$ plays a crucial role in submodular optimization:

- Different constraints have different hardness results.
- Normally the difficulty increases when the constraint becomes more general.

Popular constraints

Some popular constraints:

- Cardinality constraint
- Knapsack constraint
- Matroid constraint
- Matching
- p-System
- ...

Constraints cont.

First we define hereditary set systems.

Definition (Hereditary)

A constraint $\mathcal{I} \subseteq 2^V$ is said to be hereditary if

$$I \in \mathcal{I} \implies J \in \mathcal{I} \text{ for any } J \subseteq I.$$

A hereditary constraint is sometimes called an independent system and each $I \in \mathcal{I}$ is called an independent set.

All constraints we will discuss are hereditary.

Constraints cont.

Cardinality

Cardinality constraint: $\mathcal{I} = \{A \subseteq V \mid |A| \leq k\}$

Knapsack

Knapsack Constraint: each $i \in V$ is assigned a weight $w_i \geq 0$, $\mathcal{I} = \{S \subseteq V \mid \sum_{i \in S} w_i \leq W\}$.

Constraints cont.

Cardinality

Cardinality constraint: $\mathcal{I} = \{A \subseteq V \mid |A| \leq k\}$

Knapsack

Knapsack Constraint: each $i \in V$ is assigned a weight $w_i \geq 0$, $\mathcal{I} = \{S \subseteq V \mid \sum_{i \in S} w_i \leq W\}$.

Matching

Matching: given a graph $G = (V, E)$, a *Matching* is a set $S \subseteq E$ such that no edges in S share common vertex.

Matroid

Matroid is the generalization of the independence concept in linear algebra; omit details here ...

p-System

p-system is very general, it includes many other constraints as special cases.

Definition of p-System

Let $(V, \mathcal{I})$ be a set system and $\mathcal{I}$ hereditary. Let $\mathcal{B}(A)$ be the collection of all bases of A.

$$\mathcal{I} = \{A \subseteq V \mid \frac{\max_{S \in \mathcal{B}(A)} |S|}{\min_{S \in \mathcal{B}(A)} |S|} \leq p\}.$$

p-System

p-system is very general, it includes many other constraints as special cases.

Definition of p-System

Let $(V, \mathcal{I})$ be a set system and $\mathcal{I}$ hereditary. Let $\mathcal{B}(A)$ be the collection of all bases of A.

$$\mathcal{I} = \{A \subseteq V \mid \frac{\max_{S \in \mathcal{B}(A)} |S|}{\min_{S \in \mathcal{B}(A)} |S|} \leq p\}.$$

Note: a base of A is the maximal independent set included in A.

p-System

p-system is very general, it includes many other constraints as special cases.

Definition of p-System

Let $(V, \mathcal{I})$ be a set system and $\mathcal{I}$ hereditary. Let $\mathcal{B}(A)$ be the collection of all bases of A.

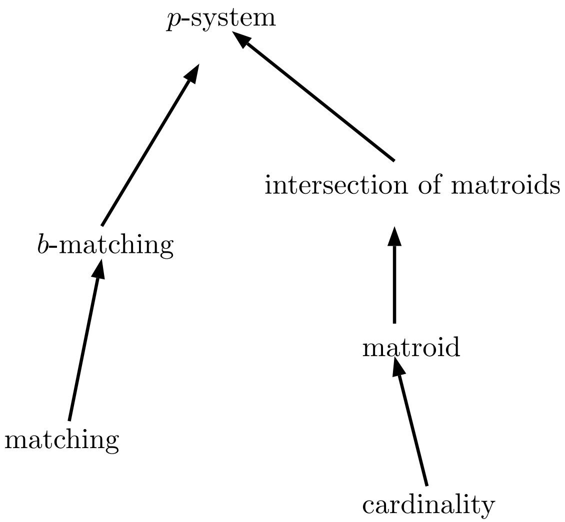
$$\mathcal{I} = \{A \subseteq V \mid \frac{\max_{S \in \mathcal{B}(A)} |S|}{\min_{S \in \mathcal{B}(A)} |S|} \leq p\}.$$

Note: a base of A is the maximal independent set included in A.

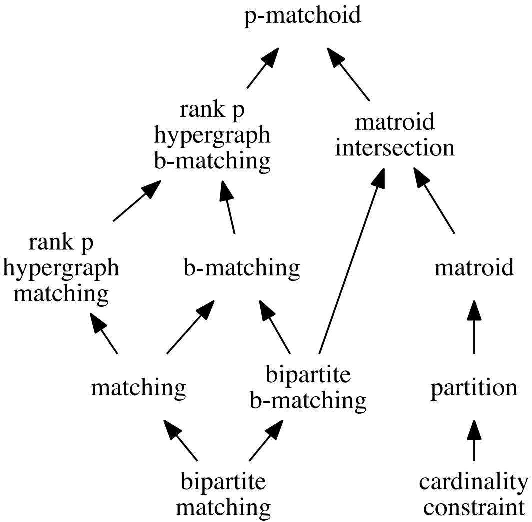
examples of p-system

- matroid is 1-system
- matching is 2-system
- intersection of p matroid is p-system
- ...

Hierarchy of constraints



Hierarchy of constraints (extended)



Notations

Some notations

- $\Delta_f(e|S) = f(S+e) - f(S)$ (or simply $\Delta(e|S)$ when f is clear from context)
- α-approximation: the returned solution S always satisfies $f(S) \geq \alpha \cdot \arg\max_{I \in \mathcal{I}} f(I)$
- When the algorithm is randomized, we normally say it guarantees α-approximation in expectation if

$$\mathbf{E}[f(S)] \geq \alpha \cdot \arg\max_{I \in \mathcal{I}} f(I).$$

The standard greedy algorithm

Algorithm 1: GREEDY algorithm for submodular maximization

Input: V the ground set, f the submodular function, $\mathcal{I}$ the constraint

Output: a set $S \subseteq V$

$S \leftarrow \emptyset$

while $\exists\ e \in V \setminus S$ *s.t.* $S \cup \{e\} \in \mathcal{I}$ **do**

  $e \leftarrow \arg\max_{e \in V \setminus S,\ S \cup \{e\} \in \mathcal{I}} \Delta_f(e|S)$

  $S \leftarrow S \cup \{e\}$

return S

Theorems of Algorithm 1

Theorem ([5], for cardinality constraint)

For a non-negative monotone submodular function $f : 2^V \to \mathbb{R}$*, let* $\mathcal{I}$ *be the cardinality constraint, Algorithm 1 returns a* $(1 - 1/e)$*-approximation to* $\arg\max_{I \in \mathcal{I}} f(S)$.

Theorems of Algorithm 1

Theorem ([5], for cardinality constraint)

For a non-negative monotone submodular function $f : 2^V \to \mathbb{R}$, let $\mathcal{I}$ be the cardinality constraint, Algorithm 1 returns a $(1 - 1/e)$-approximation to $\arg\max_{I \in \mathcal{I}} f(S)$.

Theorem ([5, 1], for p-system)

For a non-negative monotone submodular function f, given a p-system $(V, \mathcal{I})$, Algorithm 1 returns a $\frac{1}{p+1}$-approximation.

Theorems of Algorithm 1

Theorem ([5], for cardinality constraint)

For a non-negative monotone submodular function $f : 2^V \to \mathbb{R}$, let $\mathcal{I}$ be the cardinality constraint, Algorithm 1 returns a $(1 - 1/e)$-approximation to $\arg\max_{I \in \mathcal{I}} f(S)$.

Theorem ([5, 1], for p-system)

For a non-negative monotone submodular function f, given a p-system $(V, \mathcal{I})$, Algorithm 1 returns a $\frac{1}{p+1}$-approximation.

Theorem ([2], modular maximization s.t. p-system)

For a non-negative monotone modular function f, given a p-system $(V, \mathcal{I})$, Algorithm 1 returns a $\frac{1}{p}$-approximation.

Speedup - GreedyLazy

GreedyLazy

- Minoux [3] proposed Lazy-Greedy as a fast implementation for Algorithm 1.

Speedup - GreedyLazy

GreedyLazy

- Minoux [3] proposed Lazy-greedy as a fast implementation for Algorithm 1.
- GreedyLazy keeps an upper bound $\rho(e)$ on the marginal gain sorted in a heap.

Speedup - GreedyLazy

GreedyLazy

- Minoux [3] proposed LAZY-GREEDY as a fast implementation for Algorithm 1.
- GREEDYLAZY keeps an upper bound $\rho(e)$ on the marginal gain sorted in a heap.
- In each step, only update the top element in the heap and push it back, if this element remains in the top, include it into solution.

Speedup - GreedyLazy

GreedyLazy

- Minoux [3] proposed Lazy-greedy as a fast implementation for Algorithm 1.
- GreedyLazy keeps an upper bound $\rho(e)$ on the marginal gain sorted in a heap.
- In each step, only update the top element in the heap and push it back, if this element remains in the top, include it into solution.
- Again gives $(1 - e^{-1})$-approximation.

Speedup - StocGreedy[4]

StocGreedy

- In each round, instead of considering all $V \backslash S$ to get

$$e \leftarrow \underset{e \in V \backslash S,\ S \cup \{e\} \in \mathcal{I}}{\arg\max} \Delta_f(e|S),$$

Speedup - StocGreedy[4]

StocGreedy

- In each round, instead of considering all $V \setminus S$ to get

$$e \leftarrow \underset{e \in V \setminus S,\ S \cup \{e\} \in \mathcal{I}}{\arg\max} \Delta_f(e|S),$$

- consider only $\frac{|V|}{k} \log \frac{1}{\epsilon}$ random samples from $V \setminus S$.

Speedup - StocGreedy[4]

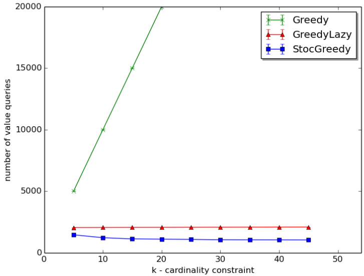
StocGreedy

- In each round, instead of considering all $V \setminus S$ to get

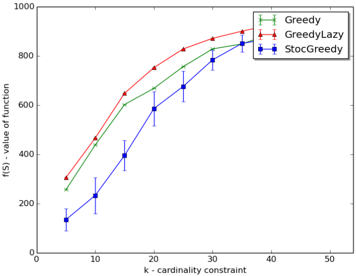
$$e \leftarrow \underset{e \in V \setminus S,\ S \cup \{e\} \in \mathcal{I}}{\arg\max} \Delta_f(e|S),$$

- consider only $\frac{|V|}{k} \log \frac{1}{\epsilon}$ random samples from $V \setminus S$.
- $(1 - e^{-1} - \epsilon)$-approximation in expectation.

Comparison



(a) Efficiency



(b) Quality

Figure : Experiment on SYNTHETIC dataset

Multiple Columns

Heading

1. Statement
2. Explanation
3. Example

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lectus nisl, ultricies in feugiat rutrum, porttitor sit amet augue. Aliquam ut tortor mauris. Sed volutpat ante purus, quis accumsan dolor.

Table

Treatments	Response 1	Response 2
Treatment 1	0.0003262	0.562
Treatment 2	0.0015681	0.910
Treatment 3	0.0009271	0.296

Table : Table caption

Theorem

Theorem (Mass–energy equivalence)

$E = mc^2$

Figure

Uncomment the code on this slide to include your own image from the same directory as the template .TeX file.

Citation

An example of the `\cite` command to cite within the presentation:

This statement requires citation [Smith, 2012].

References

John Smith (2012)
Title of the publication
Journal Name 12(3), 45 – 678.

The End

G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák.
Maximizing a monotone submodular function subject to a matroid constraint.
SIAM Journal on Computing, 40(6):1740–1766, 2011.

T. A. Jenkyns.
The efficacy of the greedy algorithm.
In *Proceedings of the 7th Southeastern Conference on Combinatorics, Graph Theory, and Computing, Utilitas Mathematica, Winnipeg*, pages 341–350, 1976.

M. Minoux.
Accelerated greedy algorithms for maximizing submodular set functions.
In *Optimization Techniques*, pages 234–243. Springer, 1978.

B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrak, and A. Krause.
Lazier than lazy greedy.

In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher.
An analysis of approximations for maximizing submodular set functionsi.
Mathematical Programming, 14(1):265–294, 1978.