Advanced Topics in Data Sciences

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 2: Submodular Optimization

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)

EE-731 (Spring 2016)

lions@epfl

MARIE CURIE ACTIONS

FNSNF
FONDS NATIONAL SUISSE
SCHWEIZERISCHER NATIONALFONDS
FONDO NAZIONALE SVIZZERO
SWISS NATIONAL SCIENCE FOUNDATION

erc

EPFL

License Information for Mathematics of Data Slides

- This work is released under a Creative Commons License with the following terms:
- **Attribution**
  - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees must give the original authors credit.
- **Non-Commercial**
  - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees may not use the work for commercial purposes – unless they get the licensor's permission.
- **Share Alike**
  - The licensor permits others to distribute derivative works only under a license identical to the one that governs the licensor's work.
- Full Text of the License

Outline

- This lecture
  1. Submodular Function Maximization
  2. Performance of the Greedy Algorithm
  3. Submodular Function Minimization
  4. Lovász extension
  5. Interger programming

Recommended Reading

- *Submodular function maximization*, Krause and Golovin, 2012
- *An analysis of approximations for maximizing submodular set functions*, Nemhauser, 1978
- *Submodular functions and convexity*, Lovász, 1983
- *Learning with submodular functions: A convex optimization perspective*, Francis Bach, 2013 (Sections 3 & 10).

- *Lecture 3: Convex analysis and complexity*, Mathematics of Data: From Theory to Computation, 2015.

Submodular optimization

Recall the definition of submodular functions:

Definition (Submodularity)

A function $f : 2^V \to \mathbb{R}$ is said to be:

- *submodular* if, for all $S \subseteq T \subseteq V$ and $e \in V \backslash T$, it holds $\Delta(e|S) \geq \Delta(e|T)$;
- *modular* if it always holds that $\Delta(e|S) = \Delta(e|T)$;

where

$$\Delta(e|S) = f(S \cup \{e\}) - f(S)$$

Submodular optimization

Recall the definition of submodular functions:

Definition (Submodularity)

A function $f : 2^V \to \mathbb{R}$ is said to be:

- *submodular* if, for all $S \subseteq T \subseteq V$ and $e \in V \setminus T$, it holds $\Delta(e|S) \geq \Delta(e|T)$;
- *modular* if it always holds that $\Delta(e|S) = \Delta(e|T)$;

where

$$\Delta(e|S) = f(S \cup \{e\}) - f(S)$$

Several problems in theoretical computer science, game theory, machine learning and learning-based CS (c.f., lecture 1) can be cast as a submodular optimization problem.

Problem (Submodular Optimization)

Given a submodular function $f : 2^V \to \mathbb{R}$,

$$\min_{S \in I} f(S) \qquad \text{(SFMin)} \qquad\qquad \max_{S \in I} f(S) \qquad \text{(SFMax)}$$

Submodular optimization

Several problems in theoretical computer science, game theory, machine learning and learning-based CS (c.f., lecture 1) can be cast as a submodular optimization problem.

Problem (Submodular Optimization)

Given a submodular function $f : 2^V \to \mathbb{R}$,

$$\min_{S \in I} f(S) \quad \text{(SFMin)} \qquad\qquad \max_{S \in I} f(S) \quad \text{(SFMax)}$$

In this lecture:

- For $I = 2^V$ (unconstrained), SFMin can be solved in polynomial time, while SFMax is NP-Hard[1].
- For $I = \{S : |S| \leq k\}$, and f *monotone*, SFMax admits a $(1 - \frac{1}{e})$-approximation.
- For $I = 2^V$ or $I = \{S : |S| \leq k\}$, and f *modular*, SFMax/SFMin is easy to solve.
- For I decribed by *totally unimodular* linear constraints, and f *modular*, SFMax/SFMin, which in this case is an integer program (IP), can be solved by linear programming (LP).

[1] cannot be solved in polynomial time unless $P = NP$.

Unconstrained submodular maximization is NP hard

Given a graph $G(V, E)$, for any $S \subseteq V$, the *graph cut* function $f(S) = |\delta(S)|$ denotes the number of edges "cut" in the graph.

$$|\delta(S)| = |\{(u, v) \in E : u \in S, v \in V \setminus S\}|$$

Problem (Max cut)

Find $S \subseteq V$ such that the number of edges between S and the complementary subset is as large as possible.

$$\max_{S \subseteq V} |\delta(S)|$$

- Max Cut problem is *NP-Hard*.
- Graph cut function is *submodular*.
- Hence, SFMax is *NP-Hard*.

Unconstrained SFMax admits a $1/2$-*approximation* algorithm [2] which is *tight*; a $(1/2 + \epsilon)$-approximation requires exponentially many oracle calls [5].

Modular function maximization

While *submodular* maximization is hard, *modular* maximization is extremely easy.

Unconstrained modular maximization

Given constants $c_1, \ldots, c_n \in \mathbb{R}$,

$$\max_{S \subseteq V} f(S) := \sum_{i \in S} c_i$$

Optimal set $S^\star$ contains the indices i for which $c_i > 0$. This has complexity $O(n)$.

Cardinality Constrained modular maximization

$$\max_{|S| \leq k} \sum_{i \in S} c_i.$$

Optimal set $S^\star$ contains the indices i, among the k largest values of c_i, for which $c_i > 0$. This can be done by sorting, in $O(n \log n)$, or by Quickselect randomized algorithm in $\Theta(n)$ expected time [4].

Submodular maximization in Learning-based CS

LB-CS: Problem statement

Given a set of m *training signals* $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{C}^p$, find an index set Ω of a given cardinality n such that a related *test signal* $\mathbf{x}$ can reliably be recovered given the subsampled measurement vector $\mathbf{b} = \mathbf{P}_\Omega \boldsymbol{\Psi} \mathbf{x}$.

Average energy criterion

$$\hat{\Omega} = \arg\max_{\Omega : |\Omega| = n} \frac{1}{m} \sum_{j=1}^{m} \sum_{i \in \Omega} |\langle \boldsymbol{\psi}_i, \mathbf{x}_j \rangle|^2$$

This is a cardinality constrained modular maximization problem.

Submodular maximization in Learning-based CS

LB-CS: Problem statement

Given a set of m *training signals* $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{C}^p$, find an index set Ω of a given cardinality n such that a related *test signal* $\mathbf{x}$ can reliably be recovered given the subsampled measurement vector $\mathbf{b} = \mathbf{P}_\Omega \mathbf{\Psi} \mathbf{x}$.

Average energy criterion

$$\hat{\Omega} = \arg\max_{\Omega : |\Omega| = n} \frac{1}{m} \sum_{j=1}^{m} \sum_{i \in \Omega} |\langle \boldsymbol{\psi}_i, \mathbf{x}_j \rangle|^2$$

This is a cardinality constrained modular maximization problem.

What about generalized case?

Generalized average energy criterion

$$\hat{\Omega} = \arg\max_{\Omega : |\Omega| = n} \frac{1}{m} \sum_{j=1}^{m} g\Big(\sum_{i \in \Omega} |\langle \boldsymbol{\psi}_i, \mathbf{x}_j \rangle|^2 \Big).$$

where $g : \mathbb{R} \to \mathbb{R}$ be an increasing concave function with $g(0) = 0$. This is a cardinality constrained monotone submodular maximization problem.

Cardinality constrained submodular maximization

Cardinality constrained monotone submodular function maximization is also NP-hard.

$$S^\star \in \arg\max_{|S| \leq k} f(S)$$

Greedy algorithm [9]

1. Initialize $S_0 = \emptyset$
2. For $i = 1, \ldots, k$
   - Find $e_i = \arg\max_{e \in V \setminus S_{i-1}} \Delta(e|S_{i-1})$
   - Set $S_i = S_{i-1} \cup \{e_i\}$
3. Return S_k

Theorem (Approximation achieved by greedy algorithm [9])

For any monotone submodular function with $f(\emptyset) = 0$, after ℓ iterations of the greedy algorithm, it holds that

$$f(S_\ell) \geq (1 - e^{-\ell/k}) f(S^\star)$$

After k iterations, the greedy algorithm achieves a $(1 - 1/e)$*-approximation*, which is *tight*; no algorithm requiring a *polynomial* number of oracle calls has a better performance [10].

Proof of greedy algorithm performance

We can assume w.l.o.g that $|S^\star| = k$, since by monotonicity of f adding elements can only increase the function value. Let $S^\star = \{e_1^\star, e_2^\star, \cdots, e_k^\star\}$, then for all $i < \ell$,

$$
\begin{aligned}
f(S^\star) &\leq f(S^\star \cup S_i) && \text{by monotonicity} \\
&= f(S_i) + \sum_{j=1}^{k} \Delta\left(e_j^\star | S_i \cup \{e_1^\star, e_2^\star, \cdots, e_{j-1}^\star\}\right) && \text{by telescoping sums} \\
&\leq f(S_i) + \sum_{j=1}^{k} \Delta\left(e_j^\star | S_i\right) && \text{by submodularity} \\
&\leq f(S_i) + \sum_{j=1}^{k} \Delta\left(e_i | S_i\right) && \text{by definition of the greedy updates} \\
&\leq f(S_i) + k\left(f(S_{i+1}) - f(S_i)\right) && |S^\star| = k
\end{aligned}
$$

Proof of greedy algorithm performance

Let $\Delta f_i := f(S^*) - f(S_i)$. Then, $\Delta f_{i+1} \leq \left(1 - \frac{1}{k}\right) \Delta f_i, \forall i < \ell$.

$$f(S^\star) - f(S_\ell) \leq \left(1 - \frac{1}{k}\right)^\ell f(S^\star) \qquad (f(\emptyset) = 0)$$

$$\leq e^{-l/k} f(S^*) \qquad (1 - x \leq e^{-x}, \forall x \in \mathbb{R})$$

Rearranging terms yields $f(S_l) \geq (1 - e^{-l/k}) f(S^*)$. □

- When $\ell = k$, the above approximation factor becomes $1 - 1/e \approx 0.63$
- Running the greedy algorithm for more than k iterations leads to a better approximation factor, but still with respect to the k-optimal solution. E.g., for $\ell = 5k$, $f(S_{5k}) \geq 0.9933 \max_{|S| \leq k} f(S)$

A related constrained submodular maximization problem

Two related questions:

- Given a fixed $k \in \mathbb{Z}$, if we choose k items greedily, how far away is $f(S_k)$ from the optimal function value $f(S_k^\star)$ for sets of size k?
- Given a fixed $z \in \mathbb{Z}, z \leq f(V)$, if we now run the greedy algorithm until we reach $f(S_\ell) \geq z$, how wasteful is the size of S_l, compared to the minimum sized set achieving $f(S) \geq z$.

$$k^\star = \min_{S \subseteq V} \{|S| : f(S) \geq z\} \qquad \text{(Minimum submodular set cover)}$$

Theorem ([15])

Given a normalized ($f(\emptyset) = 0$) monotone submodular integer-valued function f and a fixed $z \in \mathbb{Z}, z \leq f(V)$. Let ℓ be smallest integer such that $f(S_\ell) \geq z$. Then

$$\ell \leq \left(1 + \ln \max_{v \in V} f(\{v\})\right) k^\star$$

Unconstrained submodular minimization

We abuse notation by treating any set function $f : 2^V \to \mathbb{R}$ as a function over $\{0,1\}^n$ too, where $f(\mathbb{1}_S) = f(S)$. SFMin can then be equivalently written as:

$$\min_{s \in \{0,1\}^n} f(s)$$

Relax and round approach:

- Relax integer constraints $\{0,1\}^n$ to $[0,1]^n$.
- Relax discrete objective f defined over $\{0,1\}^n$ to a continuous *extension* f' defined over $[0,1]^n$; i.e., $f'(s) = f(s), \forall s \in \{0,1\}^n$.
- Choose a continuous function that can be minimized efficiently and is close to the original function f.

Definition (Convex closure)

Given any set function $f : \{0,1\}^n \to \mathbb{R}$, we define $\forall x \in [0,1]^n$ the convex closure of f as:

$$f^-(x) = \min_{\alpha \in [0,1]^{2^n}} \left\{ \sum_{S \subseteq V} \alpha_S f(S) : x = \sum_{S \subseteq V} \alpha_S \mathbb{1}_S, \sum_{S \subseteq V} \alpha_S = 1, \alpha_S \geq 0 \right\}$$

Lovász extension

The convex closure $f^-(x) = f_L(x)$ where f_L is the Lovász extension of f iff f is a *submodular* function [14].

Definition (Lovász extension [7])

Given a normalized ($f(\emptyset) = 0$) set function $f : \{0,1\}^n \to \mathbb{R}$, its Lovász extension $f_L : [0,1]^n \to \mathbb{R}$ is defined $\forall s \in [0,1]^n$ as follows:

$$f_L(s) = \sum_{k=1}^{n} s_{j_k} \Big(f(\{j_1, \cdots, j_k\}) - f(\{j_1, \cdots, j_{k-1}\}) \Big)$$

where $s_{j_1} \geq s_{j_1} \geq \cdots \geq s_{j_n}$.

Observations:

1. f_L is an *extension* of f since $f_L(s) = f(s), \forall s \in \{0,1\}^n$.
2. f_L can be computed efficiently in $O(n \log n)$.
3. For a fixed ordering of s, f_L is a *linear* function.
4. For modular functions, f_L is a *linear* function.
5. f_L is *positively homogenous*, i.e., $f_L(\alpha s) = \alpha f_L(s), \forall \alpha > 0$.
6. Let $h = f + g$, then $h_L = f_L + g_L$.
7. f_L is a *non-decreasing* function if f is monotone.
8. The Lovász extension has several equivalent definitions (c.f., [1]).

Lovász extension

Theorem ([7])

Given a set function f and its Lovász extension f_L, f_L is convex iff f is submodular.

Proof:

- If f is submodular, $f_L = f^-$ is convex.
- If f_L is convex, then:
  - The vector $\mathbb{1}_{A\cup B} + \mathbb{1}_{A\cap B} = \mathbb{1}_A + \mathbb{1}_B$ have entries equal to 2 on $A \cap B$, 1 on $A \Delta B = A \setminus B \cup B \setminus A$, and 0 outside of $A \cup B$.
  - By definition of LE:

$$
\begin{aligned}
f_L(\mathbb{1}_{A\cup B} + \mathbb{1}_{A\cap B}) &= 2(f(A \cap B) - f(\emptyset)) + 1(f(A\Delta B \cup A \cap B) - f(A \cap B)) \\
&= f(A \cap B) + f(A \cup B)
\end{aligned}
$$

  - $f_L(\mathbb{1}_A) = f(A)$ and $f_L(\mathbb{1}_B) = f(B)$.
  - $f_L(\mathbb{1}_{A\cup B} + \mathbb{1}_{A\cap B}) = f_L(\mathbb{1}_A + \mathbb{1}_B) \leq f_L(\mathbb{1}_A) + f_L(\mathbb{1}_B)$.
  - It follows that $f(A \cap B) + f(A \cup B) \leq f(A) + f(B)$.

Examples of Lovász extensions

1. Let $f(S) = |S|$. f is a modular function. $f_L(s) = \mathbb{1}^T s$.
2. Let $f(S) = \mathbb{1}_{G \cap S \neq \emptyset}(S)$. f is a submodular function.

$$f_L(s) = s_{j_1} \mathbb{1}_{j_1 \in G} + s_{j_2} \mathbb{1}_{j_2 \in G, j_1 \notin G} + \cdots + s_{j_n} \mathbb{1}_{j_n \in G, V \setminus j_n \notin G} = \max_{k \in G} s_k,$$

   where $s_{j_1} \geq s_{j_1} \geq \cdots \geq s_{j_n}$.
3. $f(S) = \sum_{G \in \mathfrak{G}} d_G \mathbb{1}_{G \cap S \neq \emptyset}(S)$, where $d_G > 0$. f is a submodular function.

$$f_L(s) = \sum_{G \in \mathfrak{G}} d_G \max_{k \in G} s_k.$$

Submodular functions can be minimized in polynomial time

Theorem (SFMin is equivalent to a convex problem)

Given a normalized submodular function f and its Lovász extension f_L,

$$\min_{s \in \{0,1\}^n} f(s) = \min_{x \in [0,1]^n} f_L(x)$$

where any minimizer $s^\star$ of the LHS is a minimizer of the RHS, and any minimizer $x^\star$ of the RHS, has all its thresholded sets $\{x^\star \geq \theta\}, \forall \theta \in (0,1)$ as minimizers of the LHS.

Proof:

- Since $\{0,1\}^n \subseteq [0,1]^n$, then $f(s^\star) \geq f_L(x^\star)$.
- We use an equivalent definition of the Lovász extension [1]:

$$\begin{aligned}
f_L(x^\star) &= \sum_{k=1}^{n-1} (x^\star_{j_k} - x^\star_{j_{k+1}}) f(\{j_1, \cdots, j_k\}) + x^\star_{j_n} f(V) \\
&\geq \sum_{k=1}^{n-1} (x^\star_{j_k} - x^\star_{j_{k+1}}) f(s^\star) + x^\star_{j_n} f(s^\star) && (x_{j_k} - x_{j_{k+1}} \geq 0) \\
&\geq x_{j_1} f(s^\star) \\
&\geq f(s^\star) && (\text{since } f(s^\star) \leq f(\emptyset) = 0 \text{ and } x_{j_1} \leq 1)
\end{aligned}$$

where $x^\star_{j_1} \geq \cdots \geq x^\star_{j_n}$.

For the proof of the second claim, c.f., [1].

Submodular functions can be minimized in polynomial time

Theorem (SFMin is equivalent to a convex problem)

Given a normalized submodular function f and its Lovász extension f_L,

$$\min_{s\in\{0,1\}^n} f(s) = \min_{x\in[0,1]^n} f_L(x)$$

where any minimizer $s^\star$ of the LHS is a minimizer of the RHS, and any minimizer $x^\star$ of the RHS, has all its thresholded sets $\{x^\star \geq \theta\}, \forall\theta \in (0,1)$ as minimizers of the LHS.

The convex problem $\min_{x\in[0,1]^n} f_L(x)$ can be solved for e.g., by the following algorithms:

- Minimum norm point algorithm [3], $O((n^3 EO + n^3 \log n + n^4)F^2)$
- Frank Wolfe algorithm [1], $O((n^3 EO + n^3 \log n)F^2)$
- Combinatorial algorithms, e.g., [11], $O(n^5 EO + n^6)$
- Cutting plane method [6], $O(n^2 \log nF \cdot EO + n^3 log^{O(1)} nF)$ and $O(n^3 log^2 n \cdot EO + n^4 log^{O(1)} n)$

where $F := \max_i\{|f(\{i\})|, |f(V) - f(V \setminus i)|\}$ and EO is the running time of the evaluation oracle.

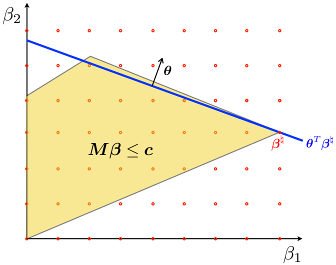
Integer linear programming

Integer linear program

Many important discrete optimization problems can be formulated as an integer linear program

$$\beta^\natural \in \arg\max_{\beta \in \mathbb{Z}^m} \{\theta^T \beta : M\beta \le c\} \qquad \text{(IP)}$$

NP-Hard (in general)



Convex Polyhedra

$$\mathcal{P} = \{\beta | M\beta \le c\}$$

$$(\beta \in \mathbb{R}^m, c \in \mathbb{R}^m)$$

Polytope: A bounded polyhedron

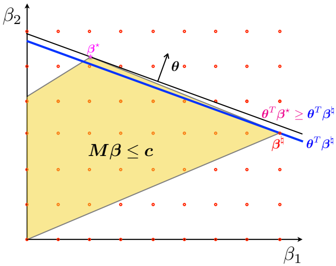
Integer linear programming

Integer linear program

Many important discrete optimization problems can be formulated as an integer linear program

$$\boldsymbol{\beta}^\natural \in \arg\max_{\boldsymbol{\beta} \in \mathbb{Z}^m} \{\boldsymbol{\theta}^T \boldsymbol{\beta} : \boldsymbol{M}\boldsymbol{\beta} \leq \boldsymbol{c}\} \qquad \text{(IP)}$$

NP-Hard (in general)



Relaxation: Linear program

$$\boldsymbol{\beta}^\star \in \arg\max_{\boldsymbol{\beta} \in \mathbb{R}^m} \{\boldsymbol{\theta}^T \boldsymbol{\beta} : \boldsymbol{M}\boldsymbol{\beta} \leq \boldsymbol{c}\} \qquad \text{(LP)}$$

Obtains an upperbound

Convex Polyhedra

$$\mathcal{P} = \{\boldsymbol{\beta} | \boldsymbol{M}\boldsymbol{\beta} \leq \boldsymbol{c}\}$$

$$(\boldsymbol{\beta} \in \mathbb{R}^m, \boldsymbol{c} \in \mathbb{R}^m)$$

Polytope: A bounded polyhedron

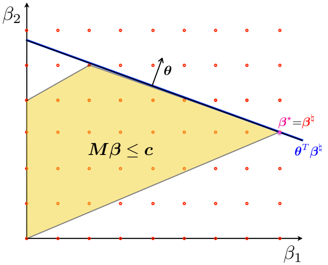
Integer linear programming

Integer linear program

Many important discrete optimization problems can be formulated as an integer linear program

$$\beta^\natural \in \arg\max_{\beta \in \mathbb{Z}^m} \{\theta^T \beta : M\beta \leq c\} \qquad \text{(IP)}$$

NP-Hard (in general)



Relaxation: Linear program

$$\beta^\star \in \arg\max_{\beta \in \mathbb{R}^m} \{\theta^T \beta : M\beta \leq c\} \qquad \text{(LP)}$$

Obtains an upperbound

Convex Polyhedra

$$\mathcal{P} = \{\beta | M\beta \leq c\}$$

Observation:

When every vertex of $\mathcal{P}$ is integral,

LP has integer optimal solutions.

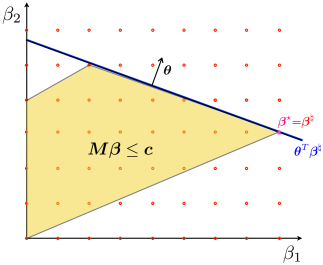
Integer linear programming

Integer linear program

Many important discrete optimization problems can be formulated as an integer linear program

$$\beta^\natural \in \arg\max_{\beta \in \mathbb{Z}^m} \{\theta^T \beta : M\beta \le c\} \qquad \text{(IP)}$$

NP-Hard (in general)



Relaxation: Linear program

$$\beta^\star \in \arg\max_{\beta \in \mathbb{R}^m} \{\theta^T \beta : M\beta \le c\} \qquad \text{(LP)}$$

Obtains an upperbound

Convex Polyhedra

$$\mathcal{P} = \{\beta | M\beta \le c\}$$

Observation:

When every vertex of $\mathcal{P}$ is integral,

LP has integer optimal solutions.

- Interior point method performs $\mathcal{O}\left(\sqrt{l}\log\frac{l}{\epsilon}\right)$ iterations ($l > m$) with up to $\mathcal{O}(m^2 l)$ operations, where ϵ is the absolute solution accuracy, and l is the number of constraints.

A sufficient condition: Total unimodularity

Definition (Total unimodularity)

A matrix $\boldsymbol{M} \in \mathbb{R}^{l \times m}$ is totally unimodular (TU) iff the determinant of every square submatrix of $\boldsymbol{M}$ is 0, or ± 1.

- Note that the entries of M are necessarily $0, \pm 1$.
- Verifying if a matrix is TU can be done in polynomial time [13].

Theorem (TU Polyhedron is integral)

The polyhedron $\mathcal{P} = \{\boldsymbol{M}\boldsymbol{\beta} \leq \boldsymbol{c}\}$ has integer vertices when $\boldsymbol{M}$ is TU and $\boldsymbol{c}$ is an integer vector.

Proof:

- Every vertex z is determined by a subsystem $M'z = c'$ where M' is matrix of full row rank.
- We can write $M' = [U, V]$ (up to some permutation of the coordinatres), where $\det U = \pm 1$.
- z is then given by $z = \begin{bmatrix} U^{-1}c' \\ 0 \end{bmatrix}$
- Cramer's rule: $U^{-1} = \frac{\mathrm{Adj}\, U}{\det U}$ where $\mathrm{Adj}\, U$ is the adjugate matrix of U.

A sufficient condition: Total unimodularity

Definition (Equivalent definition of TU [12, 8])

A matrix M is TU iff for every subset J of its columns, there exits a partition J_1, J_2 of J such that

$$|\sum_{j\in J_1} M_{ij} - \sum_{j\in J_2} M_{ij}| \leq 1, \forall i = 1, \cdots, l$$

Operations that preserves TU [8]: If M is TU then,

1. The tranpose of M is TU.
2. (M,I) is TU.
3. Matrix after deleting a row (column) with at most one non-zero entry from M is TU.
4. Matrix after interchanging two rows (columns) in M is TU.
5. Matrix after multiplying a row (column) from M by (-1) is TU.
6. Matrix after duplicating rows (column) of M is TU.
7. Matrix after applying a pivot operation on M is TU.

Examples of TU matrices

1. Given a directed graph, let T be its edge-node incidence matrix, i.e., $T_{\ell i} = 1$ and $T_{\ell j} = -1$ iff $e_\ell = (i, j)$ is an edge in the graph. Then, T is TU.

2. Given an undirected bipartite graph, let E be its edge-node incidence matrix, i.e., $T_{\ell i} = 1$ and $T_{\ell j} = 1$ iff $e_\ell = (i, j)$ is an edge in the graph. Then, E is TU.

3. Given an acyclic bipartite graph $G(U, V, E)$, where the degree of any node in U is at most 2. Let B be its biadjacency matrix, i.e., $B_{ij} = 1$ iff $i \in U, j \in V$ and $(i, j) \in E$. Then, B is TU.

Example of an integer program with TU constraints

Problem (k-sparse projection)

Recall $\|x\|_0 = |\mathrm{supp}(x)|$ *where* $\mathrm{supp}(x) = \{i : x_i \neq 0\}$. *The projection of a vector* $y \in \mathbb{R}^n$ *over the set of* k*-sparse vectors is given by:*

$$\min_{x \in \mathbb{R}^n} \{\|x - y\|_2^2 : \|x\|_0 \leq k\}$$

$$\begin{aligned}
\min_{x \in \mathbb{R}^n} \{\|x - y\|_2^2 : \|x\|_0 \leq k\} &= \min_{|S| \leq k} \min_{x \in \mathbb{R}^n} \{\|x - y\|_2^2 : \mathrm{supp}(x) = S\} \\
&= \min_{|S| \leq k} \min_{x \in \mathbb{R}^n} \{\sum_{i \in S} (x_i - y_i)^2 + \sum_{i \notin S} y_i^2 : \mathrm{supp}(x) = S\} \\
&= \min_{|S| \leq k} \{\sum_{i \notin S} y_i^2\} \\
&= \|y\|_2^2 - \max_{|S| \leq k} \{\|y\|_2^2 - \sum_{i \notin S} y_i^2\} \\
&= \|y\|_2^2 - \max_{|S| \leq k} \{\sum_{i \in S} y_i^2\}
\end{aligned}$$

Example of an integer program with TU constraints

We can view $\max_{|S|\leq k} \sum_{i\in S} y_i^2$ as an integer program over TU constraints:

$$\max_{s\in\{0,1\}} \{\sum_{i=1}^{n} s_i y_i^2 : \mathbb{1}^T s \leq k\} = \max_{s\in[0,1]} \{\sum_{i=1}^{n} s_i y_i^2 : \mathbb{1}^T s \leq k\}$$

Since matrix $\mathbb{1}^T$ is TU and $k \in \mathbb{Z}$.

References I

[1] Francis Bach.
Learning with submodular functions: A convex optimization perspective.
arXiv preprint arXiv:1111.6453, 2011.

[2] Niv Buchbinder, Moran Feldman, Joseph Seffi, and Roy Schwartz.
A tight linear time (1/2)-approximation for unconstrained submodular maximization.
SIAM Journal on Computing, 44(5):1384–1402, 2015.

[3] Deeparnab Chakrabarty, Prateek Jain, and Pravesh Kothari.
Provable submodular minimization via fujishige-wolfe's algorithm.

[4] Thomas H Cormen.
Introduction to algorithms.
MIT press, 2009.

[5] Uriel Feige, Vahab S Mirrokni, and Jan Vondrak.
Maximizing non-monotone submodular functions.
SIAM Journal on Computing, 40(4):1133–1153, 2011.

References II

[6] Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong.
A faster cutting plane method and its implications for combinatorial and convex optimization.
In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 1049–1065. IEEE, 2015.

[7] László Lovász.
Submodular functions and convexity.
In *Mathematical Programming The State of the Art*, pages 235–257. Springer, 1983.

[8] George L Nemhauser and Laurence A Wolsey.
Integer and combinatorial optimization, volume 18.
Wiley New York, 1988.

[9] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher.
An analysis of approximations for maximizing submodular set functions – i.
Mathematical Programming, 14(1):265–294, 1978.

[10] George L Nemhauser and Leonard A Wolsey.
Best algorithms for approximating the maximum of a submodular set function.
Mathematics of operations research, 3(3):177–188, 1978.

References III

[11] James B Orlin.
A faster strongly polynomial time algorithm for submodular function minimization.
Mathematical Programming, 118(2):237–251, 2009.

[12] Alexander Schrijver.
Theory of linear and integer programming.
John Wiley & Sons, 1998.

[13] Klaus Truemper.
Alpha-balanced graphs and matrices and gf (3)-representability of matroids.
Journal of Combinatorial Theory, Series B, 32(2):112–139, 1982.

[14] Jan Vondrák.
Continuous extensions of submodular functionscontinuous extensions of submodular functions.
CS 369P: Polyhedral techniques in combinatorial optimization,
`http://theory.stanford.edu/~jvondrak/CS369P-files/lec17.pdf`, 2010.

[15] Laurence A Wolsey.
An analysis of the greedy algorithm for the submodular set covering problem.
Combinatorica, 2(4):385–393, 1982.