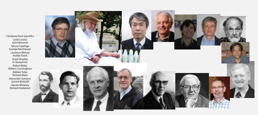
Submodular Functions, Optimization, and Applications to Machine Learning
— Spring Quarter, Lecture 3 —

http://www.ee.washington.edu/people/faculty/bilmes/classes/ee563_spring_2018/

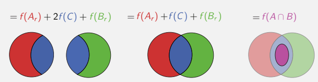
Prof. Jeff Bilmes

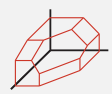
University of Washington, Seattle
Department of Electrical Engineering
http://melodi.ee.washington.edu/~bilmes

April 2nd, 2018



$f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$





Cumulative Outstanding Reading

- Read chapter 1 from Fujishige's book.

Class Road Map - EE563

- L1(3/26): Motivation, Applications, & Basic Definitions,
- L2(3/28): Machine Learning Apps (diversity, complexity, parameter, learning target, surrogate).
- L3(4/2): Info theory exs, more apps, definitions, graph/combinatorial examples
- L4(4/4):
- L5(4/9):
- L6(4/11):
- L7(4/16):
- L8(4/18):
- L9(4/23):
- L10(4/25):
- L11(4/30):
- L12(5/2):
- L13(5/7):
- L14(5/9):
- L15(5/14):
- L16(5/16):
- L17(5/21):
- L18(5/23):
- L–(5/28): Memorial Day (holiday)
- L19(5/30):
- L21(6/4): Final Presentations maximization.

Last day of instruction, June 1st. Finals Week: June 2-8, 2018.

Two Equivalent Submodular Definitions

Definition 3.2.1 (submodular concave)

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

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \quad (3.8)$$

An alternate and (as we will soon see) equivalent definition is:

Definition 3.2.2 (diminishing returns)

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

$$f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B) \quad (3.9)$$

The incremental "value", "gain", or "cost" of v decreases (diminishes) as the context in which v is considered grows from A to B.

Two Equivalent Supermodular Definitions

Definition 3.2.1 (supermodular)

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

$$f(A) + f(B) \leq f(A \cup B) + f(A \cap B) \quad (3.8)$$

Definition 3.2.2 (supermodular (improving returns))

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

$$f(A \cup \{v\}) - f(A) \leq f(B \cup \{v\}) - f(B) \quad (3.9)$$

- Incremental "value", "gain", or "cost" of v increases (improves) as the context in which v is considered grows from A to B.
- A function f is submodular iff $-f$ is supermodular.
- If f both submodular and supermodular, then f is said to be **modular**, and $f(A) = c + \sum_{a \in A} f(a)$ (often $c = 0$).

Submodularity's utility in ML

- A model of a physical process:
  - When maximizing, submodularity naturally models: diversity, coverage, span, and information.
  - When minimizing, submodularity naturally models: cooperative costs, complexity, roughness, and irregularity.
  - vice-versa for supermodularity.
- A submodular function can act as a parameter for a machine learning strategy (active/semi-supervised learning, discrete divergence, structured sparse convex norms for use in regularization).
- Itself, as an object or function to learn, based on data.
- A surrogate or relaxation strategy for optimization or analysis
  - An alternate to factorization, decomposition, or sum-product based simplification (as one typically finds in a graphical model). I.e., a means towards tractable surrogates for graphical models.
  - Also, we can "relax" a problem to a submodular one where it can be efficiently solved and offer a bounded quality solution.
  - Non-submodular problems can be analyzed via submodularity.

Learning Submodular Functions

- Learning submodular functions is hard
- *Goemans et al. (2009)*: "can one make only polynomial number of queries to an unknown submodular function f and constructs a $\hat{f}$ such that $\hat{f}(S) \leq f(S) \leq g(n)\hat{f}(S)$ where $g : \mathbb{N} \to \mathbb{R}$?" Many results, including that even with adaptive queries and monotone functions, can't do better than $\Omega(\sqrt{n}/\log n)$.
- *Balcan & Harvey (2011)*: submodular function learning problem from a learning theory perspective, given a distribution on subsets. Negative result is that can't approximate in this setting to within a constant factor.
- *Feldman, Kothari, Vondrák (2013)*, shows in some learning settings, things are more promising (PAC learning possible in $\tilde{O}(n^2) \cdot 2^{O(1/\epsilon^4)}$).
- One example: can we learn a subclass, perhaps non-negative weighted mixtures of submodular components?

Structured Learning of Submodular Mixtures

- Constraints specified in inference form:

$$\underset{\mathbf{w},\xi_t}{\text{minimize}} \quad \frac{1}{T}\sum_t \xi_t + \frac{\lambda}{2}\|\mathbf{w}\|^2 \tag{3.1}$$

$$\text{subject to} \quad \mathbf{w}^\top \mathbf{f}_t(\mathbf{y}^{(t)}) \geq \max_{\mathbf{y}\in\mathcal{Y}_t}\Big(\mathbf{w}^\top \mathbf{f}_t(\mathbf{y}) + \ell_t(\mathbf{y})\Big) - \xi_t, \forall t \tag{3.2}$$

$$\xi_t \geq 0, \forall t. \tag{3.3}$$

- Exponential set of constraints reduced to an embedded optimization problem, "loss-augmented inference."
- $\mathbf{w}^\top \mathbf{f}_t(\mathbf{y})$ is a mixture of submodular components.
- If loss is also submodular, then loss-augmented inference is submodular optimization.
- If loss is supermodular, this is a difference-of-submodular (DS) function optimization.

Structured Prediction: Subgradient Learning

- Solvable with simple sub-gradient descent algorithm using structured variant of hinge-loss (Taskar, 2004).
- Loss-augmented inference is either submodular optimization (Lin & B. 2012) or DS optimization (Tschiatschek, Iyer, & B. 2014).

Algorithm 1: Subgradient descent learning

Input : $S = \{(\mathbf{x}^{(t)}, \mathbf{y}^{(t)})\}_{t=1}^{T}$ and a learning rate sequence $\{\eta_t\}_{t=1}^{T}$.

$w_0 = 0$;

for $t = 1, \cdots, T$ **do**

  Loss augmented inference: $\mathbf{y}_t^* \in \text{argmax}_{\mathbf{y} \in \mathcal{Y}_t} \mathbf{w}_{t-1}^\top \mathbf{f}_t(\mathbf{y}) + \ell_t(\mathbf{y})$;

  Compute the subgradient: $\mathbf{g}_t = \lambda \mathbf{w}_{t-1} + \mathbf{f}_t(\mathbf{y}^*) - \mathbf{f}_t(\mathbf{y}^{(t)})$;

  Update the weights: $\mathbf{w}_t = \mathbf{w}_{t-1} - \eta_t \mathbf{g}_t$;

Return : the averaged parameters $\frac{1}{T} \sum_t \mathbf{w}_t$.

Recall

The next page shows a slide from Lecture 1

Submodular-Supermodular Decomposition

- As an alternative to graphical decomposition, we can decompose a function without resorting sums of local terms.

Theorem 3.4.1 (Additive Decomposition (Narasimhan & Bilmes, 2005))

*Let $h : 2^V \to \mathbb{R}$ be **any** set function. Then there exists a submodular function $f : 2^V \to \mathbb{R}$ and a supermodular function $g : 2^V \to \mathbb{R}$ such that h may be additively decomposed as follows: For all $A \subseteq V$,*

$$h(A) = f(A) + g(A) \tag{3.8}$$

- For many applications (as we will see), either the submodular or supermodular component is naturally zero.
- Sometimes more natural than a graphical decomposition.
- Sometimes $h(A)$ has structure in terms of submodular functions but is non additively decomposed (one example is $h(A) = f(A)/g(A)$).
- Complementary: simultaneous graphical/submodular-supermodular decomposition (i.e., submodular + supermodular tree).

Applications of DS functions

Any function $h : 2^V \to \mathbb{R}$ can be expressed as a difference between two submodular (DS) functions, $h = f - g$.

- Sensor placement with submodular costs. I.e., let V be a set of possible sensor locations, $f(A) = I(X_A; X_{V \setminus A})$ measures the quality of a subset A of placed sensors, and $c(A)$ the submodular cost. We have $f(A) - \lambda c(A)$ as the overall objective to maximize.
- Discriminatively structured graphical models, EAR measure $I(X_A; X_{V \setminus A}) - I(X_A; X_{V \setminus A} | C)$, and synergy in neuroscience.
- Feature selection: a problem of maximizing $I(X_A; C) - \lambda c(A) = H(X_A) - [H(X_A|C) + \lambda c(A)]$, the difference between two submodular functions, where H is the entropy and c is a feature cost function.
- Graphical Model Inference. Finding x that maximizes $p(x) \propto \exp(-v(x))$ where $x \in \{0,1\}^n$ and v is a pseudo-Boolean function. When v is non-submodular, it can be represented as a difference between submodular functions.

Submodular Relaxation

- We often are unable to optimize an objective. E.g., high tree-width graphical models (as we saw).
- If potentials are submodular, we can solve them.
- When potentials are not, we might resort to factorization (e.g., the marginal polytope in variational inference, were we optimize over a tree-constrained polytope).
- An alternative is submodular relaxation. I.e., given

$$\Pr(x) = \frac{1}{Z}\exp(-E(x)) \tag{3.4}$$

where $E(x) = E_f(x) - E_g(x)$ and both of $E_f(x)$ and $E_g(x)$ are submodular.
- Any function can be expressed as the difference between two submodular functions.
- Hence, rather than minimize $E(x)$ (hard), we can minimize the easier $\tilde{E}(x) = E_f(x) - E_m(x) \geq E(x)$ where $E_m(x)$ is a modular lower bound on $E_g(x)$.

ML Target | Surrogate | Bit More Notation | Info Theory Examples | Monge | More Definitions | Graph & Combinatorial Examples

Submodular Analysis for Non-Submodular Problems

- Sometimes the quality of solutions to non-submodular problems can be analyzed via submodularity.
- For example, "deviation from submodularity" can be measured using the submodularity ratio (Das & Kempe):

$$\gamma_{U,k}(f) \triangleq \min_{L \subseteq U, S: |S| \leq k, S \cap L = \emptyset} \frac{\sum_{s \in S} f(x|L)}{f(S|L)} \tag{3.5}$$

- f is submodular if and only if $\gamma_{V,|V|} = 1$.
- For some variable selection problems, can get bounds of the form:

$$\text{Solution} \geq (1 - \frac{1}{e^{\gamma_{U^*,k}}})\text{OPT} \tag{3.6}$$

where U^* is the solution set of a variable selection algorithm.
- This gradually get worse as we move away from an objective being submodular (see Das & Kempe, 2011).
- Other analogous concepts: curvature of a submodular function, and also the submodular degree.

Ground set: E or V?

Submodular functions are functions defined on subsets of some finite set, called the ground set.

- It is common in the literature to use either E or V as the ground set — we will at different times use both (there should be no confusion).
- The terminology ground set comes from lattice theory, where V are the ground elements of a lattice (just above 0).

Notation $\mathbb{R}^E$, and modular functions as vectors

What does $x \in \mathbb{R}^E$ mean?

$$\mathbb{R}^E = \{x = (x_j \in \mathbb{R} : j \in E)\} \tag{3.7}$$

and

$$\mathbb{R}^E_+ = \{x = (x_j : j \in E) : x \geq 0\} \tag{3.8}$$

Any vector $x \in \mathbb{R}^E$ can be treated as a normalized modular function, and vice verse. That is, for $A \subseteq E$,

$$x(A) = \sum_{a \in A} x_a \tag{3.9}$$

Note that x is said to be normalized since $x(\emptyset) = 0$.

characteristic (incidence) vectors of sets & modular functions

- Given an $A \subseteq E$, define the incidence (or characteristic) vector $\mathbf{1}_A \in \{0,1\}^E$ on the unit hypercube to be

$$\mathbf{1}_A(j) = \begin{cases} 1 & \text{if } j \in A; \\ 0 & \text{if } j \notin A \end{cases} \tag{3.10}$$

or equivalently,

$$\mathbf{1}_A \stackrel{\text{def}}{=} \left\{ x \in \{0,1\}^E : x_i = 1 \text{ iff } i \in A \right\} \tag{3.11}$$

- Sometimes this is written as $\chi_A \equiv \mathbf{1}_A$.
- Thus, given modular function $x \in \mathbb{R}^E$, we can write $x(A)$ in a variety of ways, i.e.,

$$x(A) = x^\intercal \cdot \mathbf{1}_A = \sum_{i \in A} x(i) \tag{3.12}$$

Other Notation: singletons and sets

When A is a set and k is a singleton (i.e., a single item), the union is properly written as $A \cup \{k\}$, but sometimes we will write just $A + k$.

What does S^T mean when S and T are arbitrary sets?

- Let S and T be two arbitrary sets (either of which could be countable, or uncountable).
- We define the notation S^T to be the set of all functions that map from T to S. That is, if $f \in S^T$, then $f : T \to S$.
- Hence, given a finite set E, $\mathbb{R}^E$ is the set of all functions that map from elements of E to the reals $\mathbb{R}$, and such functions are identical to a vector in a vector space with axes labeled as elements of E (i.e., if $m \in \mathbb{R}^E$, then for all $e \in E$, $m(e) \in \mathbb{R}$).
- Often "2" is shorthand for the set $\{0, 1\}$. I.e., $\mathbb{R}^2$ where $2 \equiv \{0, 1\}$.
- Similarly, 2^E is the set of all functions from E to "two" — so 2^E is shorthand for $\{0, 1\}^E$ — hence, 2^E is the set of all functions that map from elements of E to $\{0, 1\}$, equivalent to all binary vectors with elements indexed by elements of E, equivalent to subsets of E. Hence, if $A \in 2^E$ then $A \subseteq E$.
- What might 3^E mean?

Example Submodular: Entropy from Information Theory

- Entropy is submodular. Let V be the index set of a set of random variables, then the function

$$f(A) = H(X_A) = -\sum_{x_A} p(x_A) \log p(x_A) \tag{3.13}$$

  is submodular.
- Proof: (further) conditioning reduces entropy. With $A \subseteq B$ and $v \notin B$,

$$H(X_v|X_B) = H(X_{B+v}) - H(X_B) \tag{3.14}$$
$$\leq H(X_{A+v}) - H(X_A) = H(X_v|X_A) \tag{3.15}$$

- We say "further" due to $B \setminus A$ not nec. empty.

Example Submodular: Entropy from Information Theory

- Alternate Proof: Conditional mutual Information is always non-negative.
- Given $A, B \subseteq V$, consider conditional mutual information quantity:

$$I(X_{A\setminus B}; X_{B\setminus A} | X_{A\cap B}) = \sum_{x_{A\cup B}} p(x_{A\cup B}) \log \frac{p(x_{A\setminus B}, x_{B\setminus A} | x_{A\cap B})}{p(x_{A\setminus B}|x_{A\cap B}) p(x_{B\setminus A}|x_{A\cap B})}$$
$$= \sum_{x_{A\cup B}} p(x_{A\cup B}) \log \frac{p(x_{A\cup B}) p(x_{A\cap B})}{p(x_A) p(x_B)} \geq 0 \quad (3.16)$$

then

$$I(X_{A\setminus B}; X_{B\setminus A} | X_{A\cap B})$$
$$= H(X_A) + H(X_B) - H(X_{A\cup B}) - H(X_{A\cap B}) \geq 0 \quad (3.17)$$

so entropy satisfies

$$H(X_A) + H(X_B) \geq H(X_{A\cup B}) + H(X_{A\cap B}) \quad (3.18)$$

Information Theory: Block Coding

- Given a set of random variables $\{X_i\}_{i\in V}$ indexed by set V, how do we partition them so that we can best block-code them within each block.
- I.e., how do we form $S \subseteq V$ such that $I(X_S; X_{V\setminus S})$ is as small as possible, where $I(X_A; X_B)$ is the mutual information between random variables X_A and X_B, i.e.,

$$I(X_A; X_B) = H(X_A) + H(X_B) - H(X_A, X_B) \quad (3.19)$$

and $H(X_A) = -\sum_{x_A} p(x_A) \log p(x_A)$ is the joint entropy of the set X_A of random variables.

Example Submodular: Mutual Information

- Also, symmetric mutual information is submodular,

$$f(A) = I(X_A; X_{V \setminus A}) = H(X_A) + H(X_{V \setminus A}) - H(X_V) \tag{3.20}$$

Note that $f(A) = H(X_A)$ and $\bar{f}(A) = H(X_{V \setminus A})$, and adding submodular functions preserves submodularity (which we will see quite soon).

Monge Matrices

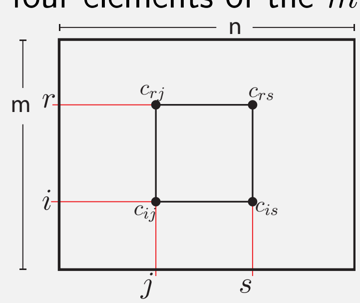
- $m \times n$ matrices $C = [c_{ij}]_{ij}$ are called Monge matrices if they satisfy the Monge property, namely:

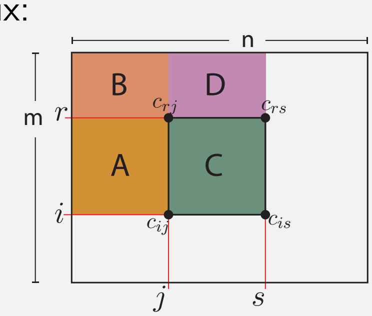
$$c_{ij} + c_{rs} \leq c_{is} + c_{rj} \tag{3.21}$$

for all $1 \leq i < r \leq m$ and $1 \leq j < s \leq n$.
- Equivalently, for all $1 \leq i, r \leq m$, $1 \leq j, s \leq n$,

$$c_{\min(i,r),\min(j,s)} + c_{\max(i,r),\max(j,s)} \leq c_{is} + c_{rj} \tag{3.22}$$

- Consider four elements of the $m \times n$ matrix:





$c_{ij} = A + B$, $c_{rj} = B$, $c_{rs} = B + D$, $c_{is} = A + B + C + D$.

Monge Matrices, where useful

- Useful for speeding up many transportation, dynamic programming, flow, search, lot-sizing and many other problems.
- Example, Hitchcock transportation problem: Given $m \times n$ cost matrix $C = [c_{ij}]_{ij}$, a non-negative supply vector $a \in \mathbb{R}^m_+$, a non-negative demand vector $b \in \mathbb{R}^n_+$ with $\sum_{i=1}^m a(i) = \sum_{j=1}^n b_j$, we wish to optimally solve the following linear program:

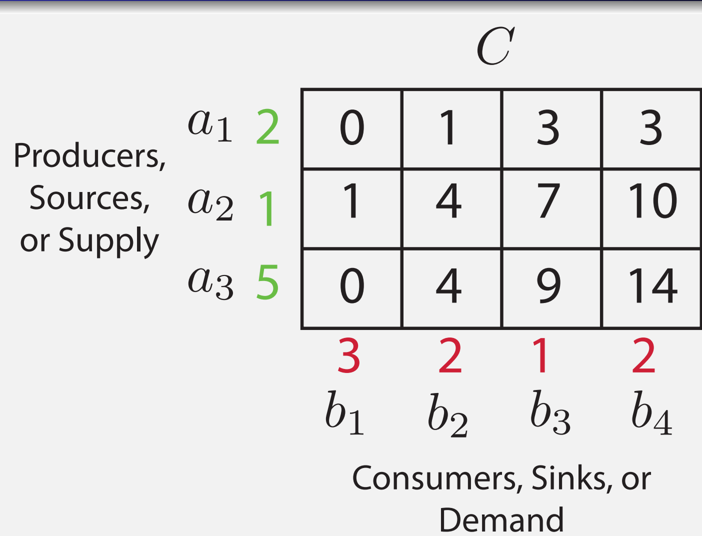
$$\underset{X \in \mathbb{R}^{m \times n}}{\text{minimize}} \quad \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \tag{3.23}$$

$$\text{subject to} \quad \sum_{i=1}^m x_{ij} = b_j \quad \forall j = 1, \ldots, n \tag{3.24}$$

$$\sum_{j=1}^n x_{ij} = a_i \quad \forall i = 1, \ldots, m \tag{3.25}$$

$$x_{i,j} \geq 0 \quad \forall i, j \tag{3.26}$$

Monge Matrices, Hitchcock transportation

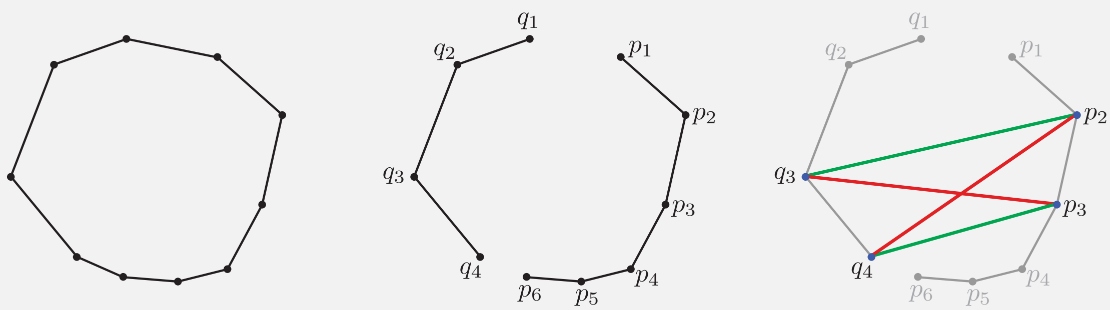


- Solving the linear program can be done easily and optimally using the "North West Corner Rule" (a 2D greedy-like approach starting at top-left and moving down-right) in only $O(m + n)$ if the matrix C is Monge!

Monge Matrices and Convex Polygons

- Can generate a Monge matrix from a convex polygon - delete two segments, then separately number vertices on each chain. Distances c_{ij} satisfy Monge property (or quadrangle inequality).

$$d(p_2, q_3) + d(p_3, q_4) \leq d(p_2, q_4) + d(p_3, q_3) \tag{3.27}$$



Monge Matrices and Submodularity

- A submodular function has the form: $f : 2^V \to \mathbb{R}$ which can be seen as $f : \{0,1\}^V \to \mathbb{R}$
- We can generalize this to $f : \{0,K\}^V \to \mathbb{R}$ for some constant $K \in \mathbb{Z}_+$.
- We may define submodularity as: for all $x, y \in \{0,K\}^V$, we have

$$f(x) + f(y) \geq f(x \vee y) + f(x \wedge y) \tag{3.28}$$

- $x \vee y$ is the (join) element-wise min of each element, that is $(x \vee y)(v) = \min(x(v), y(v))$ for $v \in V$.
- $x \wedge y$ is the (meet) element-wise min of each element, that is, $(x \wedge y)(v) = \max(x(v), y(v))$ for $v \in V$.
- With $K = 1$, then this is the standard definition of submodularity.
- With $|V| = 2$, and $K + 1$ the side-dimension of the matrix, we get a Monge property (on square matrices).
- Not-necessarily-square would be $f : \{0,K_1\} \times \{0,K_2\} \to \mathbb{R}$.

Submodular Motivation Recap

- Given a set of objects $V = \{v_1, \ldots, v_n\}$ and a function $f : 2^V \to \mathbb{R}$ that returns a real value for any subset $S \subseteq V$.
- Suppose we are interested in finding the subset that either maximizes or minimizes the function, e.g., $\text{argmax}_{S \subseteq V} f(S)$, possibly subject to some constraints.
- In general, this problem has exponential time complexity.
- Example: f might correspond to the value (e.g., information gain) of a set of sensor locations in an environment, and we wish to find the best set $S \subseteq V$ of sensors locations given a fixed upper limit on the number of sensors $|S|$.
- In many cases (such as above) f has properties that make its optimization tractable to either exactly or approximately compute.
- One such property is *submodularity*.

Two Equivalent Submodular Definitions

Definition 3.8.1 (submodular concave)

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

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \tag{3.8}$$

An alternate and (as we will soon see) equivalent definition is:

Definition 3.8.2 (diminishing returns)

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

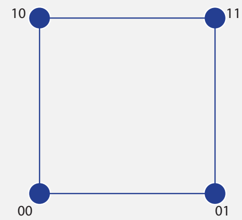
$$f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B) \tag{3.9}$$

The incremental "value", "gain", or "cost" of v decreases (diminishes) as the context in which v is considered grows from A to B.

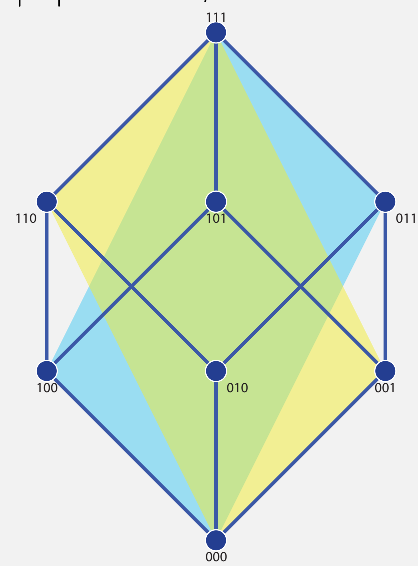
Submodular on Hypercube Vertices

- Test submodularity via values on verticies of hypercube.

Example: with $|V| = n = 2$, this is easy:



With $|V| = n = 3$, a bit harder.



How many inequalities?

Subadditive Definitions

Definition 3.8.1 (subadditive)

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

$$f(A) + f(B) \geq f(A \cup B) \tag{3.29}$$

This means that the "whole" is less than the sum of the parts.

Two Equivalent Supermodular Definitions

Definition 3.8.1 (supermodular)

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

$$f(A) + f(B) \leq f(A \cup B) + f(A \cap B) \quad (3.8)$$

Definition 3.8.2 (supermodular (improving returns))

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

$$f(A \cup \{v\}) - f(A) \leq f(B \cup \{v\}) - f(B) \quad (3.9)$$

- Incremental "value", "gain", or "cost" of v increases (improves) as the context in which v is considered grows from A to B.
- A function f is submodular iff $-f$ is supermodular.
- If f both submodular and supermodular, then f is said to be **modular**, and $f(A) = c + \sum_{a \in A} f(a)$ (often $c = 0$).

Superadditive Definitions

Definition 3.8.2 (superadditive)

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

$$f(A) + f(B) \leq f(A \cup B) \quad (3.30)$$

- This means that the "whole" is greater than the sum of the parts.
- In general, submodular and subadditive (and supermodular and superadditive) are different properties.
- Ex: Let $0 < k < |V|$, and consider $f : 2^V \to \mathbb{R}_+$ where:

$$f(A) = \begin{cases} 1 & \text{if } |A| \leq k \\ 0 & \text{else} \end{cases} \quad (3.31)$$

- This function is subadditive but not submodular.

Modular Definitions

Definition 3.8.3 (modular)

A function that is both submodular and supermodular is called **modular**

If f is a modular function, than for any $A, B \subseteq V$, we have

$$f(A) + f(B) = f(A \cap B) + f(A \cup B) \tag{3.32}$$

In modular functions, elements do not interact (or cooperate, or compete, or influence each other), and have value based only on singleton values.

Proposition 3.8.4

If f is modular, it may be written as

$$f(A) = f(\emptyset) + \sum_{a \in A} \Big(f(\{a\}) - f(\emptyset) \Big) = c + \sum_{a \in A} f'(a) \tag{3.33}$$

which has only $|V| + 1$ parameters.

ML Target Surrogate Bit More Notation Info Theory Examples Monge More Definitions Graph & Combinatorial Examples

Modular Definitions

Proof.

We inductively construct the value for $A = \{a_1, a_2, \dots, a_k\}$.
For $k = 2$,

$$f(a_1) + f(a_2) = f(a_1, a_2) + f(\emptyset) \tag{3.34}$$

$$\text{implies } f(a_1, a_2) = f(a_1) - f(\emptyset) + f(a_2) - f(\emptyset) + f(\emptyset) \tag{3.35}$$

then for $k = 3$,

$$f(a_1, a_2) + f(a_3) = f(a_1, a_2, a_3) + f(\emptyset) \tag{3.36}$$

$$\text{implies } f(a_1, a_2, a_3) = f(a_1, a_2) - f(\emptyset) + f(a_3) - f(\emptyset) + f(\emptyset) \tag{3.37}$$

$$= f(\emptyset) + \sum_{i=1}^{3} \big(f(a_i) - f(\emptyset) \big) \tag{3.38}$$

and so on . . . □

Complement function

Given a function $f : 2^V \to \mathbb{R}$, we can find a complement function $\bar{f} : 2^V \to \mathbb{R}$ as $\bar{f}(A) = f(V \setminus A)$ for any A.

Proposition 3.8.5

$\bar{f}$ is submodular iff f is submodular.

Proof.

$$\bar{f}(A) + \bar{f}(B) \geq \bar{f}(A \cup B) + \bar{f}(A \cap B) \tag{3.39}$$

follows from

$$f(V \setminus A) + f(V \setminus B) \geq f(V \setminus (A \cup B)) + f(V \setminus (A \cap B)) \tag{3.40}$$

which is true because $V \setminus (A \cup B) = (V \setminus A) \cap (V \setminus B)$ and $V \setminus (A \cap B) = (V \setminus A) \cup (V \setminus B)$ (De Morgan's laws for sets). □

Undirected Graphs

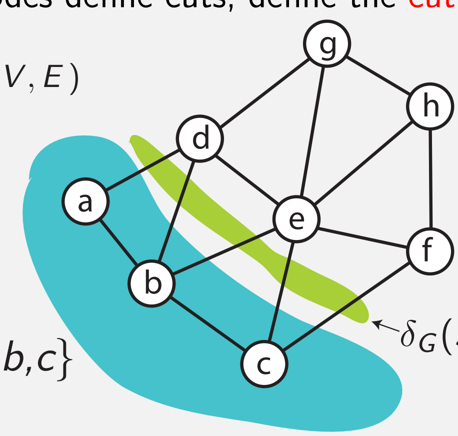
- Let $G = (V, E)$ be a graph with vertices $V = V(G)$ and edges $E = E(G) \subseteq V \times V$.
- If G is undirected, define

$$E(X, Y) = \{\{x, y\} \in E(G) : x \in X \setminus Y, y \in Y \setminus X\} \tag{3.41}$$

as the edges strictly between X and Y.

- Nodes define cuts, define the cut function $\delta(X) = E(X, V \setminus X)$.

$G = (V, E)$



$S = \{a,b,c\}$

$\delta_G(S) = \{\{u, v\} \in E : u \in S, v \in V \setminus S\}.$
$= \{\{a,d\},\{b,d\},\{b,e\},\{c,e\},\{c,f\}\}$

Directed graphs, and cuts and flows

- If G is directed, define

$$E^+(X,Y) \triangleq \{(x,y) \in E(G) : x \in X \setminus Y, y \in Y \setminus X\} \quad (3.42)$$

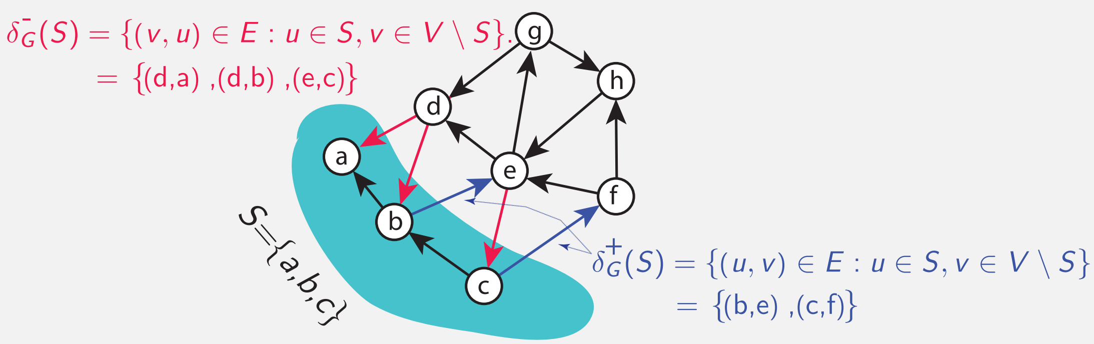
as the edges directed strictly from X towards Y.

- Nodes define cuts and flows. Define edges leaving X (out-flow) as

$$\delta^+(X) \triangleq E^+(X, V \setminus X) \quad (3.43)$$

and edges entering X (in-flow) as

$$\delta^-(X) \triangleq E^+(V \setminus X, X) \quad (3.44)$$

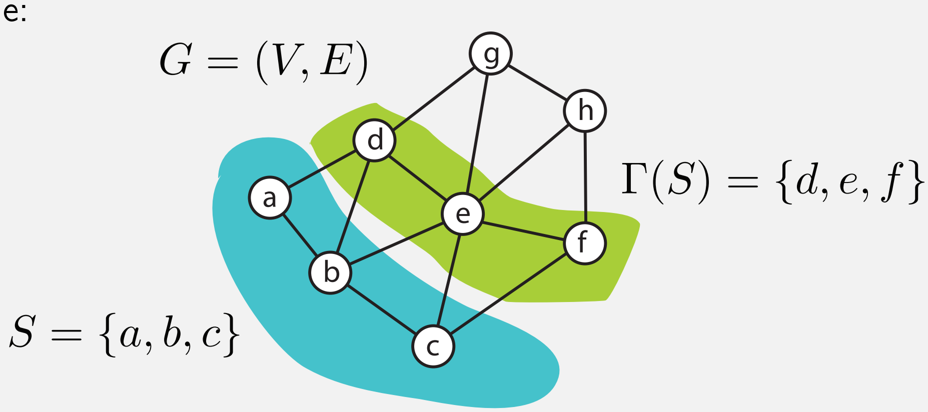


The Neighbor function in undirected graphs

- Given a set $X \subseteq V$, the neighbor function of X is defined as

$$\Gamma(X) \triangleq \{v \in V(G) \setminus X : E(X, \{v\}) \neq \emptyset\} \quad (3.45)$$

- Example:



Directed Cut function: property

Lemma 3.9.1

For a digraph $G = (V, E)$ and any $X, Y \subseteq V$: we have

$$|\delta^+(X)| + |\delta^+(Y)| = |\delta^+(X \cap Y)| + |\delta^+(X \cup Y)| + |E^+(X, Y)| + |E^+(Y, X)| \quad (3.46)$$

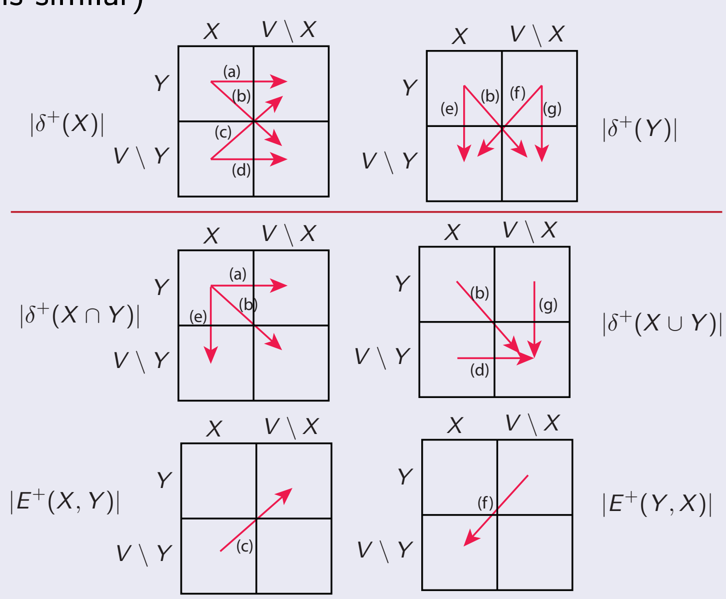
and

$$|\delta^-(X)| + |\delta^-(Y)| = |\delta^-(X \cap Y)| + |\delta^-(X \cup Y)| + |E^-(X, Y)| + |E^-(Y, X)| \quad (3.47)$$

Directed Cut function: proof of property

Proof.

We can prove Eq. (3.46) using a geometric counting argument (proof for $|\delta^-(X)|$ case is similar)



□

Directed cut/flow functions: submodular

Lemma 3.9.2

For a digraph $G = (V, E)$ and any $X, Y \subseteq V$: both functions $|\delta^+(X)|$ and $|\delta^-(X)|$ are submodular.

Proof.

$|E^+(X, Y)| \geq 0$ and $|E^-(X, Y)| \geq 0$. □

More generally, in the non-negative edge weighted case, both in-flow and out-flow are submodular on subsets of the vertices.

Undirected Cut/Flow & the Neighbor function: submodular

Lemma 3.9.3

For an undirected graph $G = (V, E)$ and any $X, Y \subseteq V$: we have that both the undirected cut (or flow) function $|\delta(X)|$ and the neighbor function $|\Gamma(X)|$ are submodular. I.e.,

$$|\delta(X)| + |\delta(Y)| = |\delta(X \cap Y)| + |\delta(X \cup Y)| + 2|E(X, Y)| \tag{3.48}$$

and

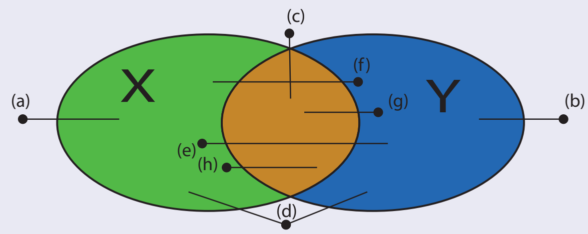
$$|\Gamma(X)| + |\Gamma(Y)| \geq |\Gamma(X \cap Y)| + |\Gamma(X \cup Y)| \tag{3.49}$$

Proof.

- Eq. (3.48) follows from Eq. (3.46): we replace each undirected edge $\{u, v\}$ with two oppositely-directed directed edges (u, v) and (v, u). Then we use same counting argument.
- Eq. (3.49) follows as shown in the following page.

...

cont.



Graphically, we can count and see that

$$\Gamma(X) = (a) + (c) + (f) + (g) + (d) \tag{3.50}$$

$$\Gamma(Y) = (b) + (c) + (e) + (h) + (d) \tag{3.51}$$

$$\Gamma(X \cup Y) = (a) + (b) + (c) + (d) \tag{3.52}$$

$$\Gamma(X \cap Y) = (c) + (g) + (h) \tag{3.53}$$

so

$$|\Gamma(X)| + |\Gamma(Y)| = (a) + (b) + 2(c) + 2(d) + (e) + (f) + (g) + (h)$$
$$\geq (a) + (b) + 2(c) + (d) + (g) + (h) = |\Gamma(X \cup Y)| + |\Gamma(X \cap Y)| \tag{3.54}$$

Undirected Neighbor functions

Therefore, the undirected cut function $|\delta(A)|$ and the neighbor function $|\Gamma(A)|$ of a graph G are both submodular.

Undirected cut/flow is submodular: alternate proof

- Another simple proof shows that $|\delta(X)|$ is submodular.
- Define a graph $G_{uv} = (\{u,v\}, \{e\}, w)$ with two nodes u, v and one edge $e = \{u,v\}$ with non-negative weight $w(e) \in \mathbb{R}_+$.
- Cut weight function over those two nodes: $w(\delta_{u,v}(\cdot))$ has valuation:

$$w(\delta_{u,v}(\emptyset)) = w(\delta_{u,v}(\{u,v\})) = 0 \tag{3.55}$$

  and

$$w(\delta_{u,v}(\{u\})) = w(\delta_{u,v}(\{v\})) = w \geq 0 \tag{3.56}$$

- Thus, $w(\delta_{u,v}(\cdot))$ is submodular since

$$w(\delta_{u,v}(\{u\})) + w(\delta_{u,v}(\{v\})) \geq w(\delta_{u,v}(\{u,v\})) + w(\delta_{u,v}(\emptyset)) \tag{3.57}$$

- General non-negative weighted graph $G = (V, E, w)$, define $w(\delta(\cdot))$:

$$f(X) = w(\delta(X)) = \sum_{(u,v) \in E(G)} w(\delta_{u,v}(X \cap \{u,v\})) \tag{3.58}$$

- This is easily shown to be submodular using properties we will soon see (namely, submodularity closed under summation and restriction).

Other graph functions that are submodular/supermodular

These come from Narayanan's book 1997. Let G be an undirected graph.

- Let $V(X)$ be the vertices adjacent to some edge in $X \subseteq E(G)$, then $|V(X)|$ (the <u>vertex function</u>) is submodular.
- Let $E(S)$ be the edges with both vertices in $S \subseteq V(G)$. Then $|E(S)|$ (the <u>interior edge function</u>) is supermodular.
- Let $I(S)$ be the edges with at least one vertex in $S \subseteq V(G)$. Then $|I(S)|$ (the <u>incidence function</u>) is submodular.
- Recall $|\delta(S)|$, is the set size of edges with exactly one vertex in $S \subseteq V(G)$ is submodular (<u>cut size function</u>). Thus, we have $I(S) = E(S) \cup \delta(S)$ and $E(S) \cap \delta(S) = \emptyset$, and thus that $|I(S)| = |E(S)| + |\delta(S)|$. So we can get a submodular function by summing a submodular and a supermodular function. If you had to guess, is this always the case?
- Consider $f(A) = |\delta^+(A)| - |\delta^+(V \setminus A)|$. Guess, submodular, supermodular, modular, or neither? Exercise: determine which one and prove it.

Number of connected components in a graph via edges

- Recall, $f : 2^V \to \mathbb{R}$ is submodular, then so is $\bar{f} : 2^V \to \mathbb{R}$ defined as $\bar{f}(S) = f(V \setminus S)$.
- Hence, if $g : 2^V \to \mathbb{R}$ is supermodular, then so is $\bar{g} : 2^V \to \mathbb{R}$ defined as $\bar{g}(S) = g(V \setminus S)$.
- Given a graph $G = (V, E)$, for each $A \subseteq E(G)$, let $c(A)$ denote the number of connected components of the (spanning) subgraph $(V(G), A)$, with $c : 2^E \to \mathbb{R}_+$.
- $c(A)$ is monotone non-increasing, $c(A + a) - c(A) \leq 0$.
- Then $c(A)$ is supermodular, i.e.,
$$c(A + a) - c(A) \leq c(B + a) - c(B) \tag{3.59}$$
with $A \subseteq B \subseteq E \setminus \{a\}$.
- Intuition: an edge is "more" (no less) able to bridge separate components (and reduce the number of conected components) when edge is added in a smaller context than when added in a larger context.
- $\bar{c}(A) = c(E \setminus A)$ is number of connected components in G when we remove A; supermodular monotone non-decreasing but not normalized.

Graph Strength

- So $\bar{c}(A) = c(E \setminus A)$ is the number of connected components in G when we remove A, is supermodular.
- Maximizing $\bar{c}(A)$ might seem as a goal for a network attacker — many connected components means that many points in the network have lost connectivity to many other points (unprotected network).
- If we can remove a small set A and shatter the graph into many connected components, then the graph is weak.
- An attacker wishes to choose a small number of edges (since it is cheap) to shatter the graph into as many components as possible.
- Let $G = (V, E, w)$ with $w : E \to \mathbb{R}+$ be a weighted graph with non-negative weights.
- For $(u, v) = e \in E$, let $w(e)$ be a measure of the strength of the connection between vertices u and v (strength meaning the difficulty of cutting the edge e).

Graph Strength

- Then $w(A)$ for $A \subseteq E$ is a modular function

$$w(A) = \sum_{e \in A} w_e \tag{3.60}$$

so that $w(E(G[S]))$ is the "internal strength" of the vertex set S.
- Suppose removing A shatters G into a graph with $\bar{c}(A) > 1$ components — then $w(A)/(\bar{c}(A) - 1)$ is like the "effort per achieved/additional component" for a network attacker.
- A form of graph strength can then be defined as the following:

$$\textit{strength}(G, w) = \min_{A \subseteq E(G): \bar{c}(A) > 1} \frac{w(A)}{\bar{c}(A) - 1} \tag{3.61}$$

- Graph strength is like the minimum effort per component. An attacker would use the argument of the min to choose which edges to attack. A network designer would maximize, over G and/or w, the graph strength, *strength*(G, w).
- Since submodularity, problems have strongly-poly-time solutions.

Submodularity, Quadratic Structures, and Cuts

Lemma 3.9.4

Let $\mathbf{M} \in \mathbb{R}^{n \times n}$ be a symmetric matrix and $m \in \mathbb{R}^n$ be a vector. Then $f : 2^V \to \mathbb{R}$ defined as

$$f(X) = m^\intercal \mathbf{1}_X + \frac{1}{2} \mathbf{1}_X^\intercal \mathbf{M} \mathbf{1}_X \tag{3.62}$$

is submodular iff the off-diagonal elements of M are non-positive.

Proof.

- Given a complete graph $G = (V, E)$, recall that $E(X)$ is the edge set with both vertices in $X \subseteq V(G)$, and that $|E(X)|$ is supermodular.
- Non-negative modular weights $w^+ : E \to \mathbb{R}_+$, $w(E(X))$ is also supermodular, so $-w(E(X))$ is submodular.
- f is a modular function $m^\intercal \mathbf{1}_A = m(A)$ added to a weighted submodular function, hence f is submodular.

Submodularity, Quadratic Structures, and Cuts

Proof of Lemma 3.9.4 cont.

- Conversely, suppose f is submodular.
- Then $\forall u, v \in V$, $f(\{u\}) + f(\{v\}) \geq f(\{u, v\}) + f(\emptyset)$ while $f(\emptyset) = 0$.
- This requires:

$$0 \leq f(\{u\}) + f(\{v\}) - f(\{u, v\}) \quad (3.63)$$
$$= m(u) + \frac{1}{2}M_{u,u} + m(v) + \frac{1}{2}M_{v,v} \quad (3.64)$$
$$- \Big(m(u) + m(v) + \frac{1}{2}M_{u,u} + M_{u,v} + \frac{1}{2}M_{v,v}\Big) \quad (3.65)$$
$$= -M_{u,v} \quad (3.66)$$

So that $\forall u, v \in V$, $M_{u,v} \leq 0$.

□

Set Cover and Maximum Coverage
just Special cases of Submodular Optimization

- We are given a finite set U of m elements and a set of subsets $\mathcal{U} = \{U_1, U_2, \ldots, U_n\}$ of n subsets of U, so that $U_i \subseteq U$ and $\bigcup_i U_i = U$.
- The goal of minimum set cover is to choose the smallest subset $A \subseteq [n] \triangleq \{1, \ldots, n\}$ such that $\bigcup_{a \in A} U_a = U$.
- Maximum k cover: The goal in maximum coverage is, given an integer $k \leq n$, select k subsets, say $\{a_1, a_2, \ldots, a_k\}$ with $a_i \in [n]$ such that $|\bigcup_{i=1}^k U_{a_i}|$ is maximized.
- $f : 2^{[n]} \to \mathbb{Z}_+$ where for $A \subseteq [n]$, $f(A) = |\bigcup_{a \in A} V_a|$ is the set cover function and is submodular.
- Weighted set cover: $f(A) = w(\bigcup_{a \in A} V_a)$ where $w : U \to \mathbb{R}_+$.
- Both Set cover and maximum coverage are well known to be NP-hard, but have a fast greedy approximation algorithm, and hence are instances of submodular optimization.

Vertex and Edge Covers

Also instances of submodular optimization

Definition 3.9.5 (vertex cover)

A *vertex cover* (a "vertex-based cover of edges") in graph $G = (V, E)$ is a set $S \subseteq V(G)$ of vertices such that every edge in G is incident to at least one vertex in S.

- Let $I(S)$ be the number of edges incident to vertex set S. Then we wish to find the smallest set $S \subseteq V$ subject to $I(S) = |E|$.

Definition 3.9.6 (edge cover)

A *edge cover* (an "edge-based cover of vertices") in graph $G = (V, E)$ is a set $F \subseteq E(G)$ of edges such that every vertex in G is incident to at least one edge in F.

- Let $|V|(F)$ be the number of vertices incident to edge set F. Then we wish to find the smallest set $F \subseteq E$ subject to $|V|(F) = |V|$.

Graph Cut Problems

Also submodular optimization

- Minimum cut: Given a graph $G = (V, E)$, find a set of vertices $S \subseteq V$ that minimize the cut (set of edges) between S and $V \setminus S$.
- Maximum cut: Given a graph $G = (V, E)$, find a set of vertices $S \subseteq V$ that minimize the cut (set of edges) between S and $V \setminus S$.
- Let $\delta : 2^V \to \mathbb{R}_+$ be the cut function, namely for any given set of nodes $X \subseteq V$, $|\delta(X)|$ measures the number of edges between nodes X and $V \setminus X$ — i.e., $\delta(x) = E(X, V \setminus X)$.
- Weighted versions, where rather than count, we sum the (non-negative) weights of the edges of a cut, $f(X) = w(\delta(X))$.
- Hence, Minimum cut and Maximum cut are also special cases of submodular optimization.