Submodular Functions, Optimization, and Applications to Machine Learning

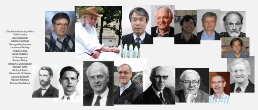
— Spring Quarter, Lecture 16 —

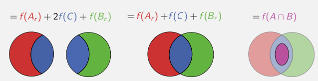
http://www.ee.washington.edu/people/faculty/bilmes/classes/ee596b_spring_2016/

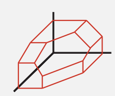
Prof. Jeff Bilmes

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

May 25th, 2016







Cumulative Outstanding Reading

- Read chapters 2 and 3 from Fujishige's book.
- Read chapter 1 from Fujishige's book.

Announcements, Assignments, and Reminders

- Homework 4, available at our assignment dropbox (https://canvas.uw.edu/courses/1039754/assignments), due (electronically) Wednesday (5/25) at 11:55pm.
- Homework 3, available at our assignment dropbox (https://canvas.uw.edu/courses/1039754/assignments), due (electronically) Monday (5/2) at 11:55pm.
- Homework 2, available at our assignment dropbox (https://canvas.uw.edu/courses/1039754/assignments), due (electronically) Monday (4/18) at 11:55pm.
- Homework 1, available at our assignment dropbox (https://canvas.uw.edu/courses/1039754/assignments), due (electronically) Friday (4/8) at 11:55pm.
- Weekly Office Hours: Mondays, 3:30-4:30, or by skype or google hangout (set up meeting via our our discussion board (https://canvas.uw.edu/courses/1039754/discussion_topics)).

Class Road Map - IT-I

- L1(3/28): Motivation, Applications, & Basic Definitions
- L2(3/30): Machine Learning Apps (diversity, complexity, parameter, learning target, surrogate).
- L3(4/4): Info theory exs, more apps, definitions, graph/combinatorial examples, matrix rank example, visualization
- L4(4/6): Graph and Combinatorial Examples, matrix rank, Venn diagrams, examples of proofs of submodularity, some useful properties
- L5(4/11): Examples & Properties, Other Defs., Independence
- L6(4/13): Independence, Matroids, Matroid Examples, matroid rank is submodular
- L7(4/18): Matroid Rank, More on Partition Matroid, System of Distinct Reps, Transversals, Transversal Matroid,
- L8(4/20): Transversals, Matroid and representation, Dual Matroids,
- L9(4/25): Dual Matroids, Properties, Combinatorial Geometries, Matroid and Greedy
- L10(4/27): Matroid and Greedy, Polyhedra, Matroid Polytopes,
- L11(5/2): From Matroids to Polymatroids, Polymatroids
- L12(5/4): Polymatroids, Polymatroids and Greedy
- L13(5/9): Polymatroids and Greedy; Possible Polytopes; Extreme Points; Polymatroids, Greedy, and Cardinality Constrained Maximization
- L14(5/11): Cardinality Constrained Maximization; Curvature; Submodular Max w. Other Constraints
- L15(5/16): Submodular Max w. Other Constraints, Most Violated $\leq$, Matroids cont., Closure/Sat,
- L16(5/18): Closure/Sat, Fund. Circuit/Dep, Min-Norm Point and SFM, Min-Norm Point Algorithm, Proof that min-norm gives optimal.
- L17(5/23):
- L18(5/25):
- L19(6/1):
- L20(6/6): Final Presentations maximization.

Finals Week: June 6th-10th, 2016.

Most violated inequality problem in matroid polytope case

- Consider

$$P_r^+ = \left\{x \in \mathbb{R}^E : x \geq 0, x(A) \leq r_M(A), \forall A \subseteq E\right\} \tag{16.7}$$

- Suppose we have any $x \in \mathbb{R}_+^E$ such that $x \notin P_r^+$.
- Hence, there must be a set of $\mathcal{W} \subseteq 2^V$, each member of which corresponds to a violated inequality, i.e., equations of the form $x(A) > r_M(A)$ for $A \in \mathcal{W}$.
- The most violated inequality when x is considered w.r.t. P_r^+ corresponds to the set A that maximizes $x(A) - r_M(A)$, i.e., the most violated inequality is valuated as:

$$\max\left\{x(A) - r_M(A) : A \in \mathcal{W}\right\} = \max\left\{x(A) - r_M(A) : A \subseteq E\right\} \tag{16.8}$$

- Since x is modular and $x(E \setminus A) = x(E) - x(A)$, we can express this via a min as in;:

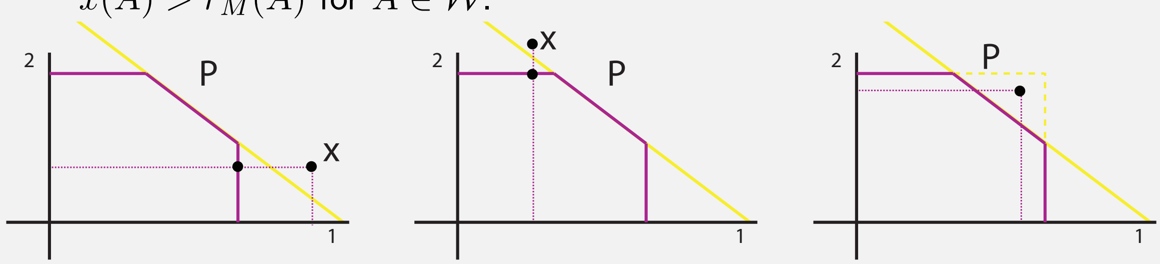
$$\min\left\{r_M(A) + x(E \setminus A) : A \subseteq E\right\} \tag{16.9}$$

Most violated inequality/polymatroid membership/SFM

- Consider

$$P_f^+ = \left\{x \in \mathbb{R}^E : x \geq 0, x(A) \leq f(A), \forall A \subseteq E\right\} \tag{16.7}$$

- Suppose we have any $x \in \mathbb{R}_+^E$ such that $x \notin P_f^+$.
- Hence, there must be a set of $\mathcal{W} \subseteq 2^V$, each member of which corresponds to a violated inequality, i.e., equations of the form $x(A) > r_M(A)$ for $A \in \mathcal{W}$.



$\mathcal{W} = \{\{1\}\{1,2\}\}$ $\mathcal{W} = \{\{2\},\{1,2\}\}$ $\mathcal{W} = \{\{1,2\}\}$

Most violated inequality/polymatroid membership/SFM

- The most violated inequality when x is considered w.r.t. P_f^+ corresponds to the set A that maximizes $x(A) - f(A)$, i.e., the most violated inequality is valuated as:

$$\max \{x(A) - f(A) : A \in \mathcal{W}\} = \max \{x(A) - f(A) : A \subseteq E\} \quad (16.7)$$

- Since x is modular and $x(E \setminus A) = x(E) - x(A)$, we can express this via a min as in;:

$$\min \{f(A) + x(E \setminus A) : A \subseteq E\} \quad (16.8)$$

- More importantly, $\min \{f(A) + x(E \setminus A) : A \subseteq E\}$ is a form of submodular function minimization, namely $\min \{f(A) - x(A) : A \subseteq E\}$ for a submodular f and $x \in \mathbb{R}_+^E$, consisting of a difference of polymatroid and modular function (so $f - x$ is no longer necessarily monotone, nor positive).
- We will ultimatley answer how general this form of SFM is.

Fundamental circuits in matroids

Lemma 16.2.5

Let $I \in \mathcal{I}(M)$, and $e \in E$, then $I \cup \{e\}$ contains at most one circuit in M.

Proof.

- Suppose, to the contrary, that there are two distinct circuits C_1, C_2 such that $C_1 \cup C_2 \subseteq I \cup \{e\}$.
- Then $e \in C_1 \cap C_2$, and by (C2), there is a circuit C_3 of M s.t. $C_3 \subseteq (C_1 \cup C_2) \setminus \{e\} \subseteq I$
- This contradicts the independence of I.

□

In general, let $C(I, e)$ be the unique circuit associated with $I \cup \{e\}$ (commonly called the fundamental circuit in M w.r.t. I and e).

Matroids: The Fundamental Circuit

- Define $C(I, e)$ be the unique circuit associated with $I \cup \{e\}$ (the fundamental circuit in M w.r.t. I and e, if it exists).
- If $e \in \text{span}(I) \setminus I$, then $C(I, e)$ is well defined ($I + e$ creates one circuit).
- If $e \in I$, then $I + e = I$ doesn't create a circuit. In such cases, $C(I, e)$ is not really defined.
- In such cases, we define $C(I, e) = \{e\}$, and we will soon see why.
- If $e \notin \text{span}(I)$, then $C(I, e) = \emptyset$, since no circuit is created in this case.

The sat function = Polymatroid Closure

- Thus, in a matroid, closure (span) of a set A are all items that A spans (eq. that depend on A).
- We wish to generalize closure to polymatroids.
- Consider $x \in P_f$ for polymatroid function f.
- Again, recall, tight sets are closed under union and intersection, and therefore form a distributive lattice.
- That is, we saw in Lecture 7 that for any $A, B \in \mathcal{D}(x)$, we have that $A \cup B \in \mathcal{D}(x)$ and $A \cap B \in \mathcal{D}(x)$, which can constitute a join and meet.
- Recall, for a given $x \in P_f$, we have defined this tight family as

$$\mathcal{D}(x) = \{A : A \subseteq E, x(A) = f(A)\} \tag{16.8}$$

Minimizers of a Submodular Function form a lattice

Theorem 16.2.6

For arbitrary submodular f, the minimizers are closed under union and intersection. That is, let $\mathcal{M} = \text{argmin}_{X \subseteq E} f(X)$ be the set of minimizers of f. Let $A, B \in \mathcal{M}$. Then $A \cup B \in \mathcal{M}$ and $A \cap B \in \mathcal{M}$.

Proof.

Since A and B are minimizers, we have $f(A) = f(B) \leq f(A \cap B)$ and $f(A) = f(B) \leq f(A \cup B)$.
By submodularity, we have

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \tag{16.10}$$

Hence, we must have $f(A) = f(B) = f(A \cup B) = f(A \cap B)$. □

Thus, the minimizers of a submodular function form a lattice, and there is a maximal and a minimal minimizer of every submodular function.

The sat function = Polymatroid Closure

- Matroid closure is generalized by the unique maximal element in $\mathcal{D}(x)$, also called the polymatroid closure or sat (saturation function).
- For some $x \in P_f$, we have defined:

$$\text{cl}(x) \stackrel{\text{def}}{=} \text{sat}(x) \stackrel{\text{def}}{=} \bigcup \{A : A \in \mathcal{D}(x)\} \tag{16.10}$$

$$= \bigcup \{A : A \subseteq E, x(A) = f(A)\} \tag{16.11}$$

$$= \{e : e \in E, \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f\} \tag{16.12}$$

- Hence, $\text{sat}(x)$ is the maximal (zero-valued) minimizer of the submodular function $f_x(A) \triangleq f(A) - x(A)$.
- Eq. (16.12) says that sat consists of any point x that is P_f saturated (any additional positive movement, in that dimension, leaves P_f). We'll revisit this in a few slides.
- First, we see how sat generalizes matroid closure.

The sat function = Polymatroid Closure

Lemma 16.3.1 (Matroid $\text{sat} : \mathbb{R}_+^E \to 2^E$ is the same as closure.)

For $I \in \mathcal{I}$, *we have* $\text{sat}(\mathbf{1}_I) = \text{span}(I)$ (16.1)

Proof.

- For $\mathbf{1}_I(I) = |I| = r(I)$, so $I \in \mathcal{D}(\mathbf{1}_I)$ and $I \subseteq \text{sat}(\mathbf{1}_I)$. Also, $I \subseteq \text{span}(I)$.
- Consider some $b \in \text{span}(I) \setminus I$.
- Then $I \cup \{b\} \in \mathcal{D}(\mathbf{1}_I)$ since $\mathbf{1}_I(I \cup \{b\}) = |I| = r(I \cup \{b\}) = r(I)$.
- Thus, $b \in \text{sat}(\mathbf{1}_I)$.
- Therefore, $\text{sat}(\mathbf{1}_I) \supseteq \text{span}(I)$.

...

The sat function = Polymatroid Closure

... proof continued.

- Now, consider $b \in \text{sat}(\mathbf{1}_I) \setminus I$.
- Choose any $A \in \mathcal{D}(\mathbf{1}_I)$ with $b \in A$, thus $b \in A \setminus I$.
- Then $\mathbf{1}_I(A) = |A \cap I| = r(A)$.
- Now $r(A) = |A \cap I| \leq |I| = r(I)$.
- Also, $r(A \cap I) = |A \cap I|$ since $A \cap I \in \mathcal{I}$.
- Hence, $r(A \cap I) = r(A) = r((A \cap I) \cup (A \setminus I))$ meaning $(A \setminus I) \subseteq \text{span}(A \cap I) \subseteq \text{span}(I)$.
- Since $b \in A \setminus I$, we get $b \in \text{span}(I)$.
- Thus, $\text{sat}(\mathbf{1}_I) \subseteq \text{span}(I)$.
- Hence $\text{sat}(\mathbf{1}_I) = \text{span}(I)$

□

The sat function = Polymatroid Closure

- Now, consider a matroid (E, r) and some $C \subseteq E$ with $C \notin \mathcal{I}$, and consider $\mathbf{1}_C$. Is $\mathbf{1}_C \in P_r$? No, it might not be a vertex, or even a member, of P_r.
- $\text{span}(\cdot)$ operates on more than just independent sets, so $\text{span}(C)$ is perfectly sensible.
- Note $\text{span}(C) = \text{span}(B)$ where $\mathcal{I} \ni B \in \mathcal{B}(C)$ is a base of C.
- Then we have $\mathbf{1}_B \leq \mathbf{1}_C \leq \mathbf{1}_{\text{span}(C)}$, and that $\mathbf{1}_B \in P_r$. We can then make the definition:

$$\text{sat}(\mathbf{1}_C) \triangleq \text{sat}(\mathbf{1}_B) \text{ for } B \in \mathcal{B}(C) \tag{16.2}$$

  In which case, we also get $\text{sat}(\mathbf{1}_C) = \text{span}(C)$ (in general, could define $\text{sat}(y) = \text{sat}(\text{P-basis}(y))$).
- However, consider the following form

$$\text{sat}(\mathbf{1}_C) = \bigcup \{A : A \subseteq E, |A \cap C| = r(A)\} \tag{16.3}$$

  Exercise: is $\text{span}(C) = \text{sat}(\mathbf{1}_C)$? Prove or disprove it.

The sat function, span, and submodular function minimization

- Thus, for a matroid, $\text{sat}(\mathbf{1}_I)$ is exactly the closure (or span) of I in the matroid. I.e., for matroid (E, r), we have $\text{span}(I) = \text{sat}(\mathbf{1}_B)$.
- Recall, for $x \in P_f$ and polymatroidal f, $\text{sat}(x)$ is the maximal (by inclusion) minimizer of $f(A) - x(A)$, and thus in a matroid, $\text{span}(I)$ is the maximal minimizer of the submodular function formed by $r(A) - \mathbf{1}_I(A)$.
- Submodular function minimization can solve "span" queries in a matroid or "sat" queries in a polymatroid.

sat, as tight polymatroidal elements

- We are given an $x \in P_f^+$ for submodular function f.
- Recall that for such an x, $\text{sat}(x)$ is defined as

$$\text{sat}(x) = \bigcup \{A : x(A) = f(A)\} \tag{16.4}$$

- We also have stated that $\text{sat}(x)$ can be defined as:

$$\text{sat}(x) = \left\{e : \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f^+\right\} \tag{16.5}$$

- We next show more formally that these are the same.

sat, as tight polymatroidal elements

- Lets start with one definition and derive the other.

$$\text{sat}(x) \stackrel{\text{def}}{=} \left\{e : \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f^+\right\} \tag{16.6}$$

$$= \{e : \forall \alpha > 0, \exists A \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A)\} \tag{16.7}$$

$$= \{e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A)\} \tag{16.8}$$

- this last bit follows since $\mathbf{1}_e(A) = 1 \iff e \in A$. Continuing, we get

$$\text{sat}(x) = \{e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } x(A) + \alpha > f(A)\} \tag{16.9}$$

- given that $x \in P_f^+$, meaning $x(A) \leq f(A)$ for all A, we must have

$$\text{sat}(x) = \{e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } x(A) = f(A)\} \tag{16.10}$$

$$= \{e : \exists A \ni e \text{ s.t. } x(A) = f(A)\} \tag{16.11}$$

- So now, if A is any set such that $x(A) = f(A)$, then we clearly have

$$\forall e \in A, e \in \text{sat}(x), \text{ and therefore that } \text{sat}(x) \supseteq A \tag{16.12}$$

sat, as tight polymatroidal elements

- ...and therefore, with sat as defined in Eq. (**??**),

$$\text{sat}(x) \supseteq \bigcup \{A : x(A) = f(A)\} \tag{16.13}$$

- On the other hand, for any $e \in \text{sat}(x)$ defined as in Eq. (16.11), since e is itself a member of a tight set, there is a set $A \ni e$ such that $x(A) = f(A)$, giving

$$\text{sat}(x) \subseteq \bigcup \{A : x(A) = f(A)\} \tag{16.14}$$

- Therefore, the two definitions of sat are identical.

Saturation Capacity

- Another useful concept is saturation capacity which we develop next.
- For $x \in P_f$, and $e \in E$, consider finding

$$\max \{\alpha : \alpha \in \mathbb{R}, x + \alpha \mathbf{1}_e \in P_f\} \tag{16.15}$$

- This is identical to:

$$\max \{\alpha : (x + \alpha \mathbf{1}_e)(A) \leq f(A), \forall A \supseteq \{e\}\} \tag{16.16}$$

since any $B \subseteq E$ such that $e \notin B$ does not change in a $\mathbf{1}_e$ adjustment, meaning $(x + \alpha \mathbf{1}_e)(B) = x(B)$.
- Again, this is identical to:

$$\max \{\alpha : x(A) + \alpha \leq f(A), \forall A \supseteq \{e\}\} \tag{16.17}$$

or

$$\max \{\alpha : \alpha \leq f(A) - x(A), \forall A \supseteq \{e\}\} \tag{16.18}$$

Saturation Capacity

- The max is achieved when

$$\alpha = \hat{c}(x; e) \stackrel{\text{def}}{=} \min \{f(A) - x(A), \forall A \supseteq \{e\}\} \tag{16.19}$$

- $\hat{c}(x; e)$ is known as the saturation capacity associated with $x \in P_f$ and e.
- Thus we have for $x \in P_f$,

$$\hat{c}(x; e) \stackrel{\text{def}}{=} \min \{f(A) - x(A), \forall A \ni e\} \tag{16.20}$$
$$= \max \{\alpha : \alpha \in \mathbb{R}, x + \alpha \mathbf{1}_e \in P_f\} \tag{16.21}$$

- We immediately see that for $e \in E \setminus \text{sat}(x)$, we have that $\hat{c}(x; e) > 0$.
- Also, for $e \in \text{sat}(x)$, we have that $\hat{c}(x; e) = 0$.
- Note that any α with $0 \leq \alpha \leq \hat{c}(x; e)$ we have $x + \alpha \mathbf{1}_e \in P_f$.
- We also see that computing $\hat{c}(x; e)$ is a form of submodular function minimization.

Dependence Function

- Tight sets can be restricted to contain a particular element.
- Given $x \in P_f$, and $e \in \text{sat}(x)$, define

$$\mathcal{D}(x, e) = \{A : e \in A \subseteq E, x(A) = f(A)\} \tag{16.22}$$
$$= \mathcal{D}(x) \cap \{A : A \subseteq E, e \in A\} \tag{16.23}$$

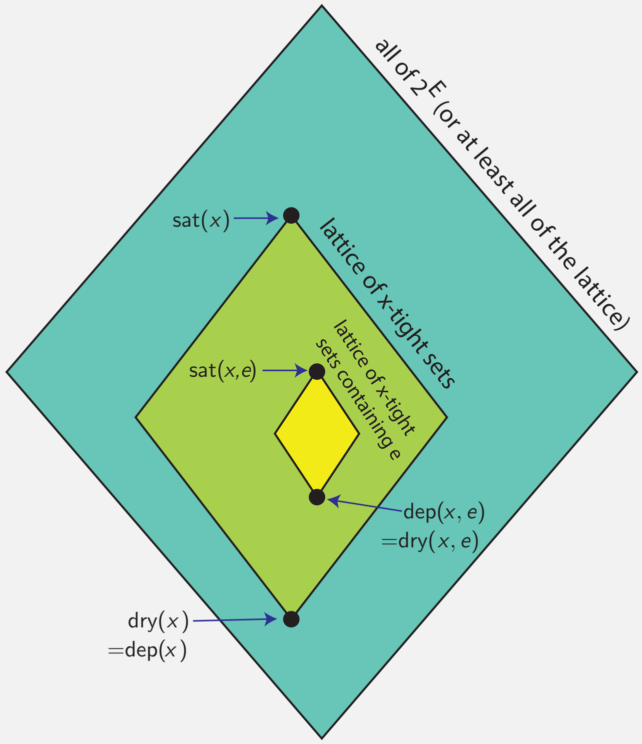
- Thus, $\mathcal{D}(x, e) \subseteq \mathcal{D}(x)$, and $\mathcal{D}(x, e)$ is a sublattice of $\mathcal{D}(x)$.
- Therefore, we can define a unique minimal element of $\mathcal{D}(x, e)$ denoted as follows:

$$\text{dep}(x, e) = \begin{cases} \bigcap \{A : e \in A \subseteq E, x(A) = f(A)\} & \text{if } e \in \text{sat}(x) \\ \emptyset & \text{else} \end{cases} \tag{16.24}$$

- I.e., $\text{dep}(x, e)$ is the minimal element in $\mathcal{D}(x)$ that contains e (the minimal x-tight set containing e).

dep and sat in a lattice

- Given some $x \in P_f$,
- The picture on the right summarizes the relationships between the lattices and sublattices.
- Note, $\bigcap_e \text{dep}(x, e) = \text{dep}(x)$.



dep and sat in a lattice

- Given $x \in P_f$, recall distributive lattice of tight sets $\mathcal{D}(x) = \{A : x(A) = f(A)\}$
- We had that $\text{sat}(x) = \bigcup \{A : A \in \mathcal{D}(x)\}$ is the "1" element of this lattice.
- Consider the "0" element of $\mathcal{D}(x)$, i.e., $\text{dry}(x) \stackrel{\text{def}}{=} \bigcap \{A : A \in \mathcal{D}(x)\}$
- We can see $\text{dry}(x)$ as the elements that are necessary for tightness.
- That is, we can equivalently define $\text{dry}(x)$ as

$$\text{dry}(x) = \left\{e' : x(A) < f(A), \forall A \not\ni e'\right\} \tag{16.25}$$

- This can be read as, for any $e' \in \text{dry}(x)$, any set that does not contain e' is not tight for x (any set A that is missing any element of $\text{dry}(x)$ is not tight).
- Perhaps, then, a better name for dry is $\text{ntight}(x)$, for the necessary for tightness (but we'll actually use neither name).
- Note that dry need not be the empty set. Exercise: give example.

An alternate expression for $\text{dep} = \text{dry}$

- Now, given $x \in P_f$, and $e \in \text{sat}(x)$, recall distributive sub-lattice of e-containing tight sets $\mathcal{D}(x, e) = \{A : e \in A, x(A) = f(A)\}$
- We can define the "1" element of this sub-lattice as $\text{sat}(x, e) \stackrel{\text{def}}{=} \bigcup \{A : A \in \mathcal{D}(x, e)\}$.
- Analogously, we can define the "0" element of this sub-lattice as $\text{dry}(x, e) \stackrel{\text{def}}{=} \bigcap \{A : A \in \mathcal{D}(x, e)\}$.
- We can see $\text{dry}(x, e)$ as the elements that are necessary for e-containing tightness, with $e \in \text{sat}(x)$.
- That is, we can view $\text{dry}(x, e)$ as

$$\text{dry}(x, e) = \left\{e' : x(A) < f(A), \forall A \not\ni e', e \in A\right\} \tag{16.26}$$

- This can be read as, for any $e' \in \text{dry}(x, e)$, any e-containing set that does not contain e' is not tight for x.
- But actually, $\text{dry}(x, e) = \text{dep}(x, e)$, so we have derived another expression for $\text{dep}(x, e)$ in Eq. (16.26).

Dependence Function and Fundamental Matroid Circuit

- Now, let $(E, \mathcal{I}) = (E, r)$ be a matroid, and let $I \in \mathcal{I}$ giving $\mathbf{1}_I \in P_r$. We have $\text{sat}(\mathbf{1}_I) = \text{span}(I) = \text{closure}(I)$.
- Given $e \in \text{sat}(\mathbf{1}_I) \setminus I$ and then consider an $A \ni e$ with $|I \cap A| = r(A)$.
- Then $I \cap A$ serves as a base for A (i.e., $I \cap A$ spans A) and any such A contains a circuit (i.e., we can add $e \in A \setminus I$ to $I \cap A$ w/o increasing rank).
- Given $e \in \text{sat}(\mathbf{1}_I) \setminus I$, and consider $\text{dep}(\mathbf{1}_I, e)$, with

$$\text{dep}(\mathbf{1}_I, e) = \bigcap \{A : e \in A \subseteq E, \mathbf{1}_I(A) = r(A)\} \tag{16.27}$$

$$= \bigcap \{A : e \in A \subseteq E, |I \cap A| = r(A)\} \tag{16.28}$$

$$= \bigcap \{A : e \in A \subseteq E, r(A) - |I \cap A| = 0\} \tag{16.29}$$

- By SFM lattice, $\exists$ a unique minimal $A \ni e$ with $|I \cap A| = r(A)$.
- Thus, $\text{dep}(\mathbf{1}_I, e)$ must be a circuit since if it included more than a circuit, it would not be minimal in this sense.

Dependence Function and Fundamental Matroid Circuit

- Therefore, when $e \in \text{sat}(\mathbf{1}_I) \setminus I$, then $\text{dep}(\mathbf{1}_I, e) = C(I, e)$ where $C(I, e)$ is the unique circuit contained in $I + e$ in a matroid (the fundamental circuit of e and I that we encountered before).
- Now, if $e \in \text{sat}(\mathbf{1}_I) \cap I$ with $I \in \mathcal{I}$, we said that $C(I, e)$ was undefined (since no circuit is created in this case) and so we defined it as $C(I, e) = \{e\}$
- In this case, for such an e, we have $\text{dep}(\mathbf{1}_I, e) = \{e\}$ since all such sets $A \ni e$ with $|I \cap A| = r(A)$ contain e, but in this case no cycle is created, i.e., $|I \cap A| \geq |I \cap \{e\}| = r(e) = 1$.
- We are thus free to take subsets of I as A, all of which must contain e, but all of which have rank equal to size, and min size is 1.
- Also note: in general for $x \in P_f$ and $e \in \text{sat}(x)$, we have $\text{dep}(x, e)$ is tight by definition.

Summary of sat, and dep

- For $x \in P_f$, $\text{sat}(x)$ (span, closure) is the maximal saturated (x-tight) set w.r.t. x. I.e., $\text{sat}(x) = \{e : e \in E, \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f\}$. That is,

$$\text{cl}(x) \stackrel{\text{def}}{=} \text{sat}(x) \triangleq \bigcup \{A : A \in \mathcal{D}(x)\} \tag{16.30}$$
$$= \bigcup \{A : A \subseteq E, x(A) = f(A)\} \tag{16.31}$$
$$= \{e : e \in E, \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f\} \tag{16.32}$$

- For $e \in \text{sat}(x)$, we have $\text{dep}(x, e) \subseteq \text{sat}(x)$ (fundamental circuit) is the minimal (common) saturated (x-tight) set w.r.t. x containing e. I.e.,

$$\text{dep}(x, e) = \begin{cases} \bigcap \{A : e \in A \subseteq E, x(A) = f(A)\} & \text{if } e \in \text{sat}(x) \\ \emptyset & \text{else} \end{cases}$$
$$= \{e' : \exists \alpha > 0, \text{ s.t. } x + \alpha(\mathbf{1}_e - \mathbf{1}_{e'}) \in P_f\} \tag{16.33}$$

Dependence Function and exchange

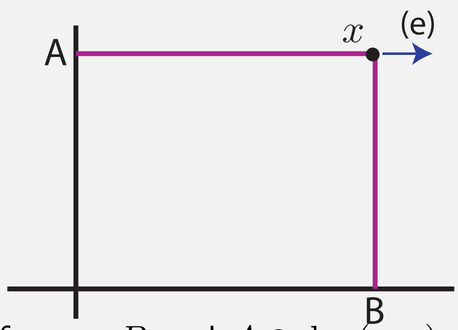
- For $e \in \text{span}(I) \setminus I$, we have that $I + e \notin \mathcal{I}$. This is a set addition restriction property.
- Analogously, for $e \in \text{sat}(x)$, any $x + \alpha \mathbf{1}_e \notin P_f$ for $\alpha > 0$. This is a vector increase restriction property.
- Recall, we have $C(I, e) \setminus e' \in \mathcal{I}$ for $e' \in C(I, e)$. I.e., $C(I, e)$ consists of elements that when removed recover independence.
- In other words, for $e \in \text{span}(I) \setminus I$, we have that

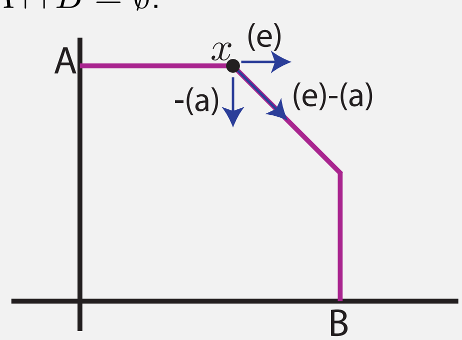
$$C(I, e) = \{a \in E : I + e - a \in \mathcal{I}\} \tag{16.34}$$

- I.e., an addition of e to I stays within $\mathcal{I}$ only if we simultaneously remove one of the elements of $C(I, e)$.
- But, analogous to the circuit case, is there an exchange property for $\text{dep}(x, e)$ in the form of vector movement restriction?
- We might expect the vector $\text{dep}(x, e)$ property to take the form: a positive move in the e-direction stays within P_f^+ only if we simultaneously take a negative move in one of the $\text{dep}(x, e)$ directions.

Dependence Function and exchange in 2D

- $\text{dep}(x, e)$ is set of neg. directions we must move if we want to move in pos. e direction, starting at x and staying within P_f.
- Viewable in 2D, we have for $A, B \subseteq E$, $A \cap B = \emptyset$:



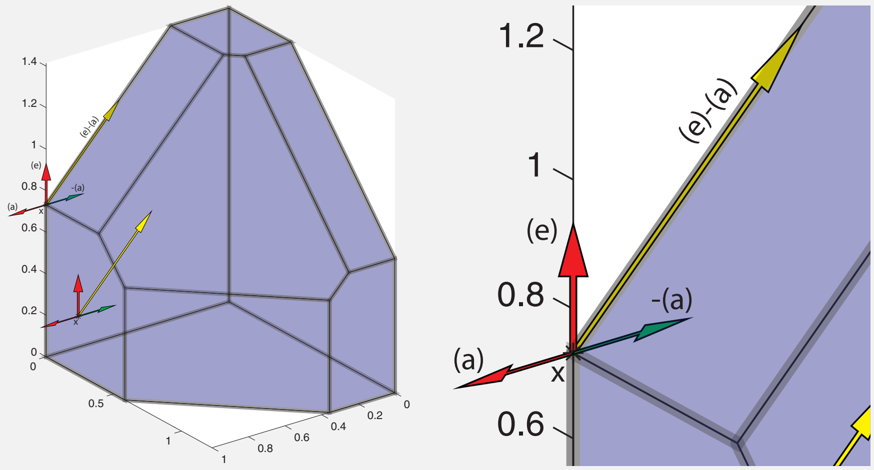


Left: $e \in B$ and $A \cap \text{dep}(x, e) = \emptyset$, and we can't move further in (e) direction, and moving in any negative $a \in A$ direction doesn't change that. **No dependence** between (e) and any element in A.

Right: $A \subseteq \text{dep}(x, e)$. We can't move further in the (e) direction, but we can move further in (e) direction by moving in some negative $a \in A$ direction. **Dependence** between (e) and elements in A.

Dependence Function and exchange in 3D

- We can move neither in the (e) nor the (a) direction, but we can move in the (e) direction if we simultaneously move in the -(a) direction.
- In 3D, we have:



- I.e., for $e \in \text{sat}(x)$, $a \in \text{sat}(x)$, $a \in \text{dep}(x, e)$, $e \notin \text{dep}(x, a)$, and
$$\text{dep}(x, e) = \{a : a \in E, \exists \alpha > 0 : x + \alpha(\mathbf{1}_e - \mathbf{1}_a) \in P_f\} \qquad (16.35)$$
- We next show this formally.

dep and exchange derived

- The derivation for $\text{dep}(x, e)$ involves turning a strict inequality into a non-strict one with a strict explicit slack variable α:

$$\text{dep}(x, e) = \text{ntight}(x, e) = \qquad (16.36)$$
$$= \left\{e' : x(A) < f(A), \forall A \not\ni e', e \in A\right\} \qquad (16.37)$$
$$= \left\{e' : \exists \alpha > 0, \text{ s.t. } \alpha \le f(A) - x(A), \forall A \not\ni e', e \in A\right\} \qquad (16.38)$$
$$= \left\{e' : \exists \alpha > 0, \text{ s.t. } \alpha \mathbf{1}_e(A) \le f(A) - x(A), \forall A \not\ni e', e \in A\right\} \qquad (16.39)$$
$$= \left\{e' : \exists \alpha > 0, \text{ s.t. } \alpha(\mathbf{1}_e(A) - \mathbf{1}_{e'}(A)) \le f(A) - x(A), \forall A \not\ni e', e \in A\right\} \qquad (16.40)$$
$$= \left\{e' : \exists \alpha > 0, \text{ s.t. } x(A) + \alpha(\mathbf{1}_e(A) - \mathbf{1}_{e'}(A)) \le f(A), \forall A \not\ni e', e \in A\right\} \qquad (16.41)$$

- Now, $1_e(A) - \mathbf{1}_{e'}(A) = 0$ if either $\{e, e'\} \subseteq A$, or $\{e, e'\} \cap A = \emptyset$.
- Also, if $e' \in A$ but $e \notin A$, then $x(A) + \alpha(\mathbf{1}_e(A) - \mathbf{1}_{e'}(A)) = x(A) - \alpha \le f(A)$ since $x \in P_f$.

dep and exchange derived

- thus, we get the same in the above if we remove the constraint $A \not\ni e', e \in A$, that is we get

$$\text{dep}(x, e) = \left\{e' : \exists \alpha > 0, \text{ s.t. } x(A) + \alpha(\mathbf{1}_e(A) - \mathbf{1}_{e'}(A)) \leq f(A), \forall A\right\} \quad (16.42)$$

- This is then identical to

$$\text{dep}(x, e) = \left\{e' : \exists \alpha > 0, \text{ s.t. } x + \alpha(\mathbf{1}_e - \mathbf{1}_{e'}) \in P_f\right\} \quad (16.43)$$

- Compare with original, the minimal element of $\mathcal{D}(x, e)$, with $e \in \text{sat}(x)$:

$$\text{dep}(x, e) = \begin{cases} \bigcap \{A : e \in A \subseteq E, x(A) = f(A)\} & \text{if } e \in \text{sat}(x) \\ \emptyset & \text{else} \end{cases} \quad (16.44)$$

Summary of Concepts

- Most violated inequality $\max\{x(A) - f(A) : A \subseteq E\}$
- Matroid by circuits, and the fundamental circuit $C(I, e) \subseteq I + e$.
- Minimizers of submodular functions form a lattice.
- Minimal and maximal element of a lattice.
- x-tight sets, maximal and minimal tight set.
- sat function & Closure
- Saturation Capacity
- e-containing tight sets
- dep function & fundamental circuit of a matroid

Summary important definitions so far: tight, dep, & sat

- x-tight sets: For $x \in P_f$, $\mathcal{D}(x) \triangleq \{A \subseteq E : x(A) = f(A)\}$.
- Polymatroid closure/maximal x-tight set: For $x \in P_f$, $\text{sat}(x) \triangleq \cup\{A : A \in \mathcal{D}(x)\} = \{e : e \in E, \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f\}$.
- Saturation capacity: for $x \in P_f$, $0 \leq \hat{c}(x;e) \triangleq \min\{f(A) - x(A) | \forall A \ni e\} = \max\{\alpha : \alpha \in \mathbb{R}, x + \alpha \mathbf{1}_e \in P_f\}$
- Recall: $\text{sat}(x) = \{e : \hat{c}(x;e) = 0\}$ and $E \setminus \text{sat}(x) = \{e : \hat{c}(x;e) > 0\}$.
- e-containing x-tight sets: For $x \in P_f$, $\mathcal{D}(x,e) = \{A : e \in A \subseteq E, x(A) = f(A)\} \subseteq \mathcal{D}(x)$.
- Minimal e-containing x-tight set/polymatroidal fundamental circuit/: For $x \in P_f$,

$$\text{dep}(x,e) = \begin{cases} \bigcap\{A : e \in A \subseteq E, x(A) = f(A)\} & \text{if } e \in \text{sat}(x) \\ \emptyset & \text{else} \end{cases}$$
$$= \{e' : \exists \alpha > 0, \text{ s.t. } x + \alpha(\mathbf{1}_e - \mathbf{1}_{e'}) \in P_f\}$$

A polymatroid function's polyhedron is a polymatroid.

Theorem 16.5.1

Let f be a submodular function defined on subsets of E. For any $x \in \mathbb{R}^E$, we have:

$$\textit{rank}(x) = \max(y(E) : y \leq x, y \in P_f) = \min(x(A) + f(E \setminus A) : A \subseteq E) \quad (16.1)$$

Essentially the same theorem as Theorem **??**, but note P_f rather than P_f^+. Taking $x = 0$ we get:

Corollary 16.5.2

Let f be a submodular function defined on subsets of E. We have:

$$\textit{rank}(0) = \max(y(E) : y \leq 0, y \in P_f) = \min(f(A) : A \subseteq E) \quad (16.2)$$

Min-Norm Point: Definition

- Restating what we saw before, we have:

$$\max \{y(E) | y \in P_f, y \leq 0\} = \min \{f(X) | X \subseteq V\} \qquad (16.45)$$

- Consider the optimization:

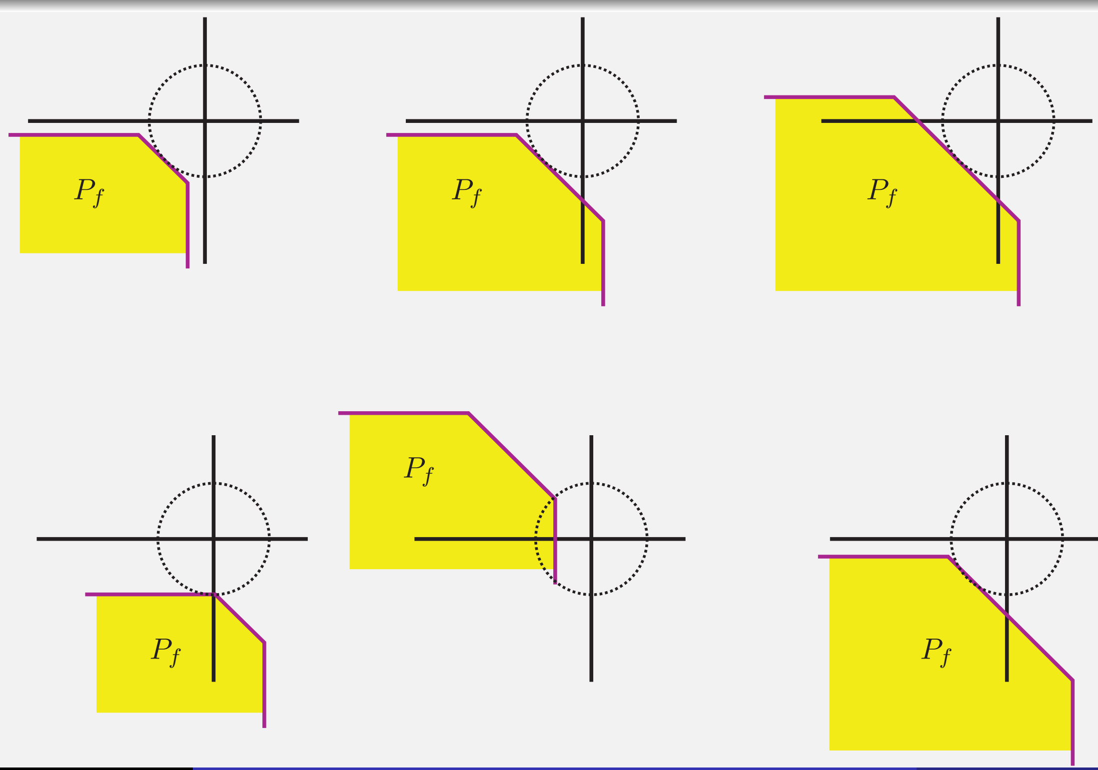
$$\text{minimize} \quad \|x\|_2^2 \qquad (16.46a)$$

$$\text{subject to} \quad x \in B_f \qquad (16.46b)$$

where B_f is the base polytope of submodular f, and $\|x\|_2^2 = \sum_{e \in E} x(e)^2$ is the squared 2-norm. Let x^* be the optimal solution.

- Note, x^* is the unique optimal solution since we have a strictly convex objective over a set of convex constraints.
- x^* is called the minimum norm point of the base polytope.

Min-Norm Point: Examples



Min-Norm Point and Submodular Function Minimization

- Given optimal solution x^* to the above, consider the quantities

$$y^* = x^* \wedge 0 = (\min(x^*(e), 0) | e \in E) \tag{16.47}$$
$$A_- = \{e : x^*(e) < 0\} \tag{16.48}$$
$$A_0 = \{e : x^*(e) \leq 0\} \tag{16.49}$$

- Thus, we immediately have that:

$$A_- \subseteq A_0 \tag{16.50}$$

and that

$$x^*(A_-) = x^*(A_0) = y^*(A_-) = y^*(A_0) \tag{16.51}$$

- It turns out, these quantities will solve the submodular function minimization problem, as we now show.
- The proof is nice since it uses the tools we've been recently developing.

Review

The following three slides are review, and are from Lectures 13, and 16.

A polymatroid function's polyhedron is a polymatroid.

Theorem 16.6.1

Let f be a submodular function defined on subsets of E. For any $x \in \mathbb{R}^E$, we have:

$$\text{rank}(x) = \max\left(y(E) : y \leq x, y \in P_f\right) = \min\left(x(A) + f(E \setminus A) : A \subseteq E\right) \tag{16.1}$$

Essentially the same theorem as Theorem **??**, but note P_f rather than P_f^+. Taking $x = 0$ we get:

Corollary 16.6.2

Let f be a submodular function defined on subsets of E. We have:

$$\text{rank}(0) = \max\left(y(E) : y \leq 0, y \in P_f\right) = \min\left(f(A) : A \subseteq E\right) \tag{16.2}$$

Min-Norm Point: Definition

- Restating what we saw before, we have:

$$\max\{y(E) | y \in P_f, y \leq 0\} = \min\{f(X) | X \subseteq V\} \tag{16.45}$$

- Consider the optimization:

$$\begin{aligned} &\text{minimize} && \|x\|_2^2 && (16.46a)\\ &\text{subject to} && x \in B_f && (16.46b) \end{aligned}$$

  where B_f is the base polytope of submodular f, and $\|x\|_2^2 = \sum_{e \in E} x(e)^2$ is the squared 2-norm. Let x^* be the optimal solution.
- Note, x^* is the unique optimal solution since we have a strictly convex objective over a set of convex constraints.
- x^* is called the minimum norm point of the base polytope.

Min-Norm Point and Submodular Function Minimization

- Given optimal solution x^* to the above, consider the quantities

$$y^* = x^* \wedge 0 = (\min(x^*(e), 0) | e \in E) \quad (16.47)$$

$$A_- = \{e : x^*(e) < 0\} \quad (16.48)$$

$$A_0 = \{e : x^*(e) \leq 0\} \quad (16.49)$$

- Thus, we immediately have that:

$$A_- \subseteq A_0 \quad (16.50)$$

and that

$$x^*(A_-) = x^*(A_0) = y^*(A_-) = y^*(A_0) \quad (16.51)$$

- It turns out, these quantities will solve the submodular function minimization problem, as we now show.
- The proof is nice since it uses the tools we've been recently developing.

Duality: convex minimization of L.E. and min-norm alg.

- Let f be a submodular function with $\tilde{f}$ it's Lovász extension. Then the following two problems are duals (Bach-2013):

$$\underset{w \in \mathbb{R}^V}{\text{minimize}} \ \tilde{f}(w) + \frac{1}{2}\|w\|_2^2 \quad (16.52)$$

$$\text{maximize} \quad -\|x\|_2^2 \quad (16.53a)$$

$$\text{subject to} \quad x \in B_f \quad (16.53b)$$

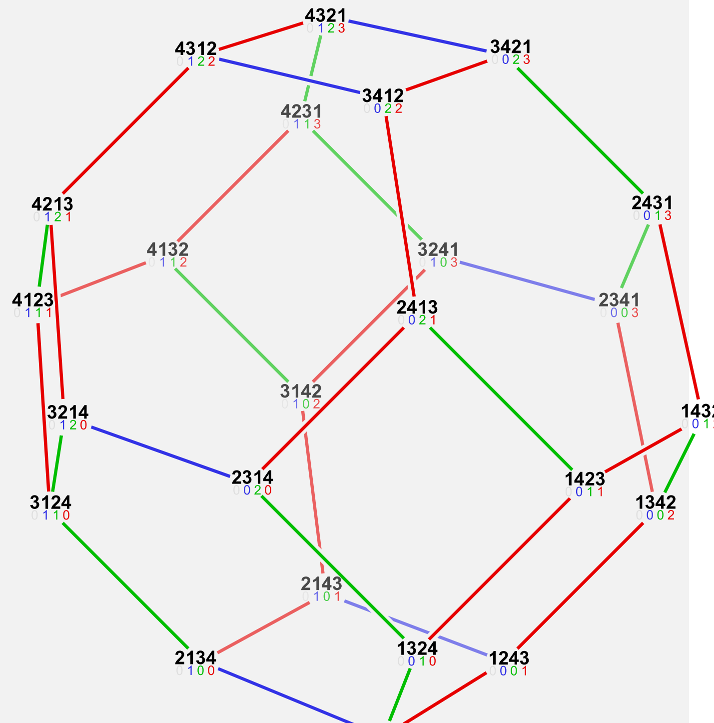
where $B_f = P_f \cap \{x \in \mathbb{R}^V : x(V) = f(V)\}$ is the base polytope of submodular function f, and $\|x\|_2^2 = \sum_{e \in V} x(e)^2$ is squared 2-norm.
- Equation (16.52) is related to proximal methods to minimize the Lovász extension (see Parikh&Boyd, "Proximal Algorithms" 2013).
- Equation (16.53b) is solved by the minimum-norm point algorithm (Wolfe-1976, Fujishige-1984, Fujishige-2005, Fujishige-2011) is (as we will see) essentially an active-set procedure for quadratic programming, and uses Edmonds's greedy algorithm to make it efficient.
- Unknown worst-case running time, although in practice it usually performs quite well (see below).

Ex: 3D base B_f: permutahedron

- Consider submodular function $f : 2^V \to \mathbb{R}$ with $|V| = 4$, and for $X \subseteq V$, concave g,

$$f(X) = g(|X|)$$
$$= \sum_{i=1}^{|X|} (4 - i + 1)$$

- Then B_f is a 3D polytope, and in this particular case gives us a permutahedron with 24 distinct extreme points, on the right (from wikipedia).



Closure/Sat Fund. Circuit/Dep Min-Norm Point and SFM Min-Norm Point Algorithm Proof that min-norm gives optimal

Modified max-min theorem

- We have a variant of Theorem 12.5.2, the min-max theorem, namely that:

Theorem 16.6.1 (Edmonds-1970)

$$\min \{f(X) | X \subseteq E\} = \max \left\{ x^-(E) | x \in B_f \right\} \quad (16.54)$$

where $x^-(e) = \min \{x(e), 0\}$ *for* $e \in E$.

Proof.

$$\min \{f(X) | X \subseteq E\} = \min_{w \in [0,1]^E} \tilde{f}(w) = \min_{w \in [0,1]^E} \max_{x \in P_f} w^\intercal x \quad (16.55)$$
$$= \min_{w \in [0,1]^E} \max_{x \in B_f} w^\intercal x \quad (16.56)$$
$$= \max_{x \in B_f} \min_{w \in [0,1]^E} w^\intercal x \quad (16.57)$$
$$= \max_{x \in B_f} x^-(E) \quad (16.58)$$

Convexity, Strong duality, and min/max swap

The min/max switch follows from strong duality. I.e., consider $g(w, x) = w^\intercal x$ and we have domains $w \in [0, 1]^E$ and $x \in B_f$. then for any $(w, x) \in [0, 1]^E \times B_f$, we have

$$\min_{w' \in [0,1]^E} g(w', x) \leq g(w, x) \leq \max_{x' \in B_f} g(w, x') \tag{16.59}$$

which means that we have weak duality

$$\max_{x \in B_f} \min_{w' \in [0,1]^E} g(w', x) \leq \min_{w \in [0,1]^E} \max_{x' \in B_f} g(w, x') \tag{16.60}$$

but since $g(w, x)$ is linear, we have strong duality, meaning

$$\max_{x \in B_f} \min_{w' \in [0,1]^E} g(w', x) = \min_{w \in [0,1]^E} \max_{x' \in B_f} g(w, x') \tag{16.61}$$

Alternate proof of modified max-min theorem

We start directly from Theorem 12.5.2.

$$\max\left(y(E) : y \leq 0, y \in P_f\right) = \min\left(f(A) : A \subseteq E\right) \tag{16.62}$$

Given $y \in \mathbb{R}^E$, define $y^- \in \mathbb{R}^E$ with $y^-(e) = \min\{y(e), 0\}$ for $e \in E$.

$$\max\left(y(E) : y \leq 0, y \in P_f\right) = \max\left(y^-(E) : y \leq 0, y \in P_f\right) \tag{16.63}$$

$$= \max\left(y^-(E) : y \in P_f\right) \tag{16.64}$$

$$= \max\left(y^-(E) : y \in B_f\right) \tag{16.65}$$

The first equality follows since $y \leq 0$. For the second equality, clearly l.h.s. $\leq$ r.h.s. Also, l.h.s. $\geq$ r.h.s. since the positive parts don't matter.

$$\max\left(y^-(E) : y \in P_f\right) = \max\left(y^-(E) : y(A) \leq f(A) \forall A\right) \tag{16.66}$$

$$= \max\left(y^-(E) : y^-(A) + y^+(A) \leq f(A) \forall A\right)$$

The third equality follows since for any $x \in P_f$ there exists a $y \in B_f$ with $x \leq y$ (follows from Theorem **??**).

$\min\{w^\intercal x : x \in B_f\}$

- Recall that the greedy algorithm solves, for $w \in \mathbb{R}_+^E$

$$\max\{w^\intercal x | x \in P_f\} = \max\{w^\intercal x | x \in B_f\} \tag{16.67}$$

since for all $x \in P_f$, there exists $y \geq x$ with $y \in B_f$.

- For arbitrary $w \in \mathbb{R}^E$, greedy algorithm will also solve:

$$\max\{w^\intercal x | x \in B_f\} \tag{16.68}$$

- Also, since

$$\min\{w^\intercal x | x \in B_f\} = -\max\{-w^\intercal x | x \in B_f\} \tag{16.69}$$

the greedy algorithm using ordering $(e_1, e_2, \dots, e_m)$ such that

$$w(e_1) \leq w(e_2) \leq \dots \leq w(e_m) \tag{16.70}$$

will solve Equation (16.69).

$\max\{w^\intercal x | x \in B_f\}$ for arbitrary $w \in \mathbb{R}^E$

Let $f(A)$ be arbitrary submodular function, and $f(A) = f'(A) - m(A)$ where f' is polymatroidal, and $w \in \mathbb{R}^E$.

$$\begin{aligned}
\max\{w^\intercal x | x \in B_f\} &= \max\{w^\intercal x | x(A) \leq f(A)\,\forall A, x(E) = f(E)\} \\
&= \max\left\{w^\intercal x | x(A) \leq f'(A) - m(A)\,\forall A, x(E) = f'(E) - m(E)\right\} \\
&= \max\left\{w^\intercal x | x(A) + m(A) \leq f'(A)\,\forall A, x(E) + m(E) = f'(E)\right\} \\
&= \max\Big\{w^\intercal x + w^\intercal m | \\
&\qquad x(A) + m(A) \leq f'(A)\,\forall A, x(E) + m(E) = f'(E)\Big\} - w^\intercal m \\
&= \max\left\{w^\intercal y | y \in B_{f'}\right\} - w^\intercal m \\
&= w^\intercal y^* - w^\intercal m = w^\intercal (y^* - m)
\end{aligned}$$

where $y = x + m$, so that $x^* = y^* - m$.

So y^* uses greedy algorithm with positive orthant $B_{f'}$. To show, we use Theorem 12.4.1 in Lecture 12, but we don't require $y \geq 0$, and don't stop when w goes negative to ensure $y^* \in B_{f'}$. Then when we subtract off m from y^*, we get solution to the original problem.

Orthogonal x-containing hyperplane & convex/affine hulls

- Define $H(x)$ as the hyperplane that is orthogonal to the line from 0 to x, while also containing x, i.e.

$$H(x) \triangleq \left\{ y \in \mathbb{R}^V \mid x^\intercal y = \|x\|_2^2 \right\} \tag{16.71}$$

  Any set $\left\{ y \in \mathbb{R}^V | x^\intercal y = c \right\}$ is orthogonal to the line from 0 to x. To also contain x, we need $\|x\|_2 \|x\|_2 \cos 0 = c$ giving $c = \|x\|_2^2$.
- Given a set of points $P = \{p_1, p_2, \ldots, p_k\}$ with $p_i \in \mathbb{R}^V$, let $\operatorname{conv} P$ be the convex hull of P, i.e.,

$$\operatorname{conv} P \triangleq \left\{ \sum_{i=1}^{k} \lambda_i p_i : \sum_i \lambda_i = 1,\ \lambda_i \geq 0, i \in [k] \right\}. \tag{16.72}$$

  and for $Q = \{q_1, q_2, \ldots, q_k\}$, with $q_i \in \mathbb{R}^V$, let $\operatorname{aff} Q$ be the affine hull of Q, i.e.,

$$\operatorname{aff} Q \triangleq \left\{ \sum_{i \in 1}^{k} \lambda_i q_i : \sum_{i=1}^{k} \lambda_i = 1 \right\} \supseteq \operatorname{conv} Q. \tag{16.73}$$

Notation

- The line between x and y: given two points $x, y \in \mathbb{R}^V$, let $[x, y] \triangleq \{\lambda x + (1 - \lambda y) : \lambda \in [0, 1]\}$. Hence, $[x, y] = \operatorname{conv}\{x, y\}$.
- Note, if we wish to minimize the 2-norm of a vector $\|x\|_2$, we can equivalently minimize its square $\|x\|_2^2 = \sum_i x_i^2$, and vice verse.

Fujishige-Wolfe Min-Norm Algorithm

- Wolfe-1976 developed an algorithm to compute the minimum norm point of a polytope, specified as a set of vertices.
- Fujishige-1984 "Submodular Systems and Related Topics" realized this algorithm can find the the min. norm point of B_f.
- Seems to be (among) the fastest general purpose SFM algo.
- Given set of points $P = \{p_1, \cdots, p_m\}$ where $p_i \in \mathbb{R}^n$: find the minimum norm point in convex hull of P:

$$\min_{x \in \text{conv } P} \|x\|_2 \tag{16.74}$$

- Wolfe's algorithm is guaranteed terminating, and explicitly uses a representation of x as a convex combination of points in P
- Algorithm maintains a set of points $Q \subseteq P$, which is always assuredly *affinely independent*.

Fujishige-Wolfe Min-Norm Algorithm

- When Q are affinely independent, minimum norm point in the affine hull of Q can easily be found, as a closed form solution for $\min_{x \in \text{aff } Q} \|x\|_2$ is available (see below).
- Algorithm repeatedly produces min. norm point x^* for selected set Q.
- If we find $w_i \geq 0, i = 1, \cdots, m$ for the minimum norm point, then x^* also belongs to $\text{conv } Q$ and also a minimum norm point over $\text{conv } Q$.
- If $Q \subseteq P$ is suitably chosen, x^* may even be the minimum norm point over $\text{conv } P$ solving the original problem.
- One of the most expensive parts of Wolfe's algorithm is solving linear optimization problem over the polytope, doable by examining all the extreme points in the polytope.
- If number of extreme points is exponential, hard to do in general.
- Number of extreme points of submodular base polytope is exponentially large, but linear optimization over the base polytope B_f doable $O(n \log n)$ time via Edmonds's greedy algorithm.

Pseudocode of Fujishige-Wolfe Min-Norm (MN) algorithm

Input : $P = \{p_1, \cdots, p_m\}, p_i \in \mathbb{R}^n, i = 1, \cdots, m$.
Output: x^*: the minimum-norm-point in $\operatorname{conv} P$.

1 $x^* \longleftarrow p_{i^*}$ where $p_{i^*} \in \operatorname{argmin}_{p \in P} \|p\|_2$ /* or choose it arbitrarily */ ;
2 $Q \longleftarrow \{x^*\}$;
3 **while** 1 **do** /* major loop */
4   **if** $x^* = 0$ *or* $H(x^*)$ *separates* P *from origin* **then**
        return : x^*
5   **else**
6   Choose $\hat{x} \in P$ on the near (closer to 0) side of $H(x^*)$;
7         $Q = Q \cup \{\hat{x}\}$;
8   **while** 1 **do** /* minor loop */
9         $x_0 \longleftarrow \min_{x \in \operatorname{aff} Q} \|x\|_2$;
10   **if** $x_0 \in \operatorname{conv} Q$ **then**
11             $x^* \longleftarrow x_0$;
12   **break**;
13   **else**
14             $y \longleftarrow \min_{x \in \operatorname{conv} Q \cap [x^*, x_0]} \|x - x_0\|_2$;
15   Delete from Q points not on the face of $\operatorname{conv} Q$ where y lies;
16             $x^* \longleftarrow y$;

Fujishige-Wolfe Min-Norm algorithm: Geometric Example

- It is advised that for the next set of slides, you have a print out of the previous MN algorithm available on display/paper somewhere.
- Algorithm maintains an invariant, namely that:

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{16.75}$$

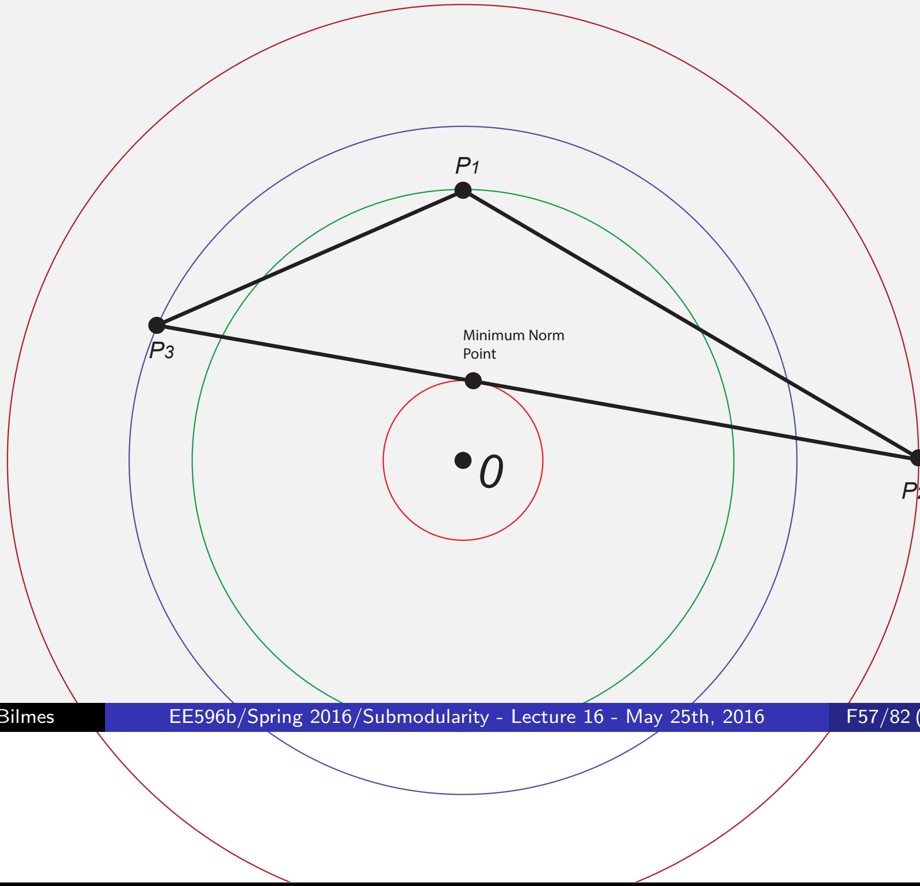
  must hold at every possible assignment of x^* (Lines 1, 11, and 16):
  1. True after Line 1 since $Q = \{x^*\}$,
  2. True after Line 11 since $x_0 \in \operatorname{conv} Q$,
  3. and true after Line 16 since $y \in \operatorname{conv} Q$ even after deleting points.
- Note also for any $x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P$, we have

$$\min_{x \in \operatorname{aff} Q} \|x\|_2 \le \min_{x \in \operatorname{conv} Q} \|x\|_2 \le \|x^*\|_2 \tag{16.76}$$

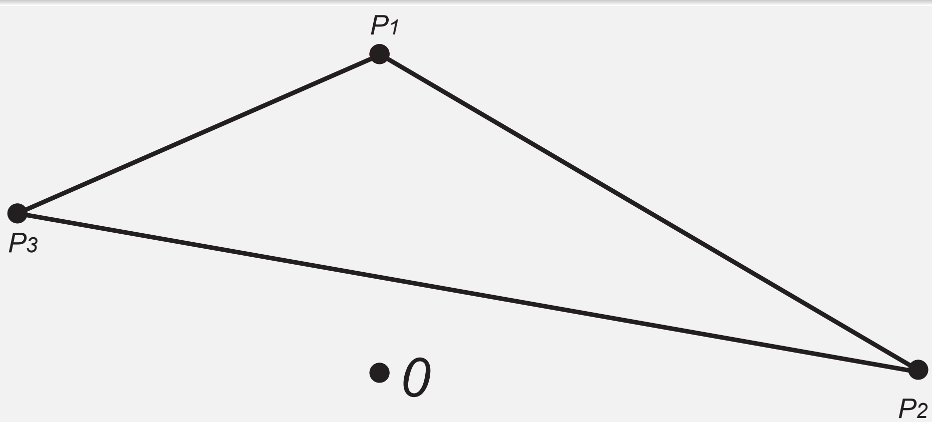
- Note, the input, P, consists of m points. In the case of the base polytope, $P = B_f$ could be exponential in $n = |V|$.
- There are six places that might be seemingly tricky or expensive: Line 4, Line 6, Line 9, Line 10, Line 14, and Line 15.
- We will consider each in turn, but first we do a geometric example.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example

Polytope, and circles concentric at 0.

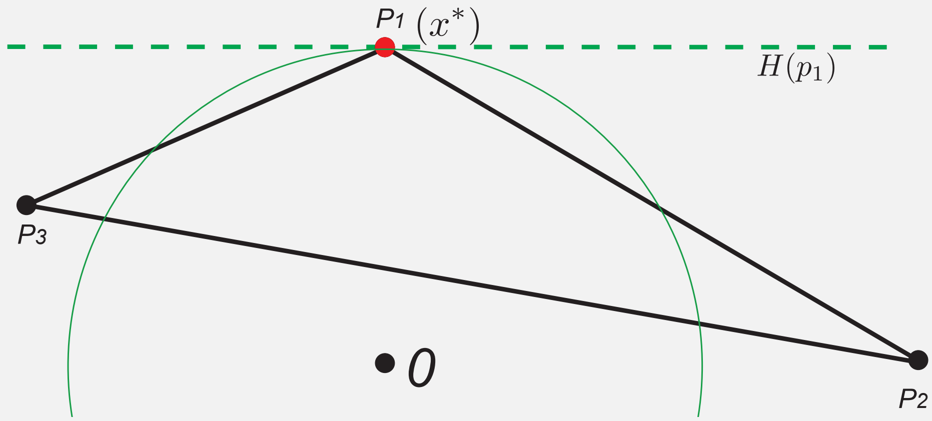


Fujishige-Wolfe Min-Norm algorithm: Geometric Example



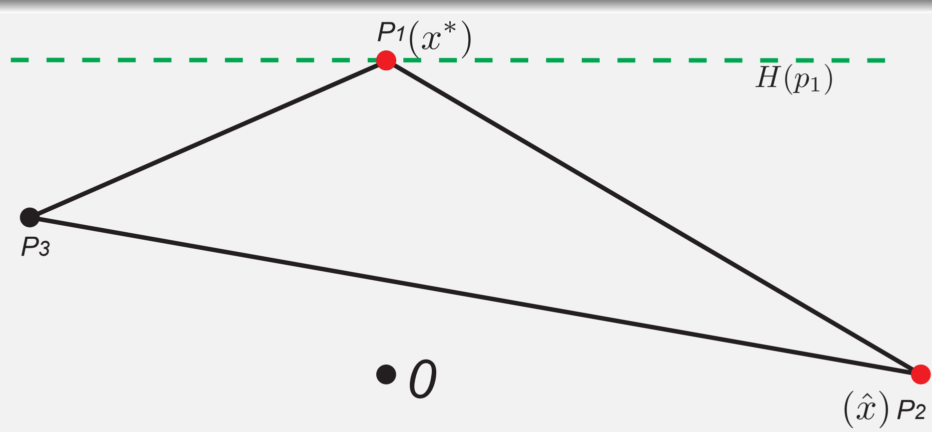
The initial polytope consisting of the convex hull of three points p_1, p_2, p_3, and the origin 0.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



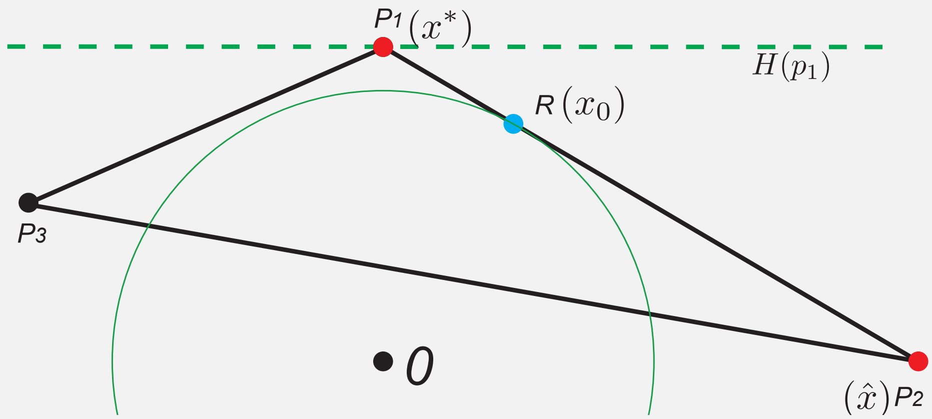
p_1 is the extreme point closest to 0 and so we choose it first, although we can choose any arbitrary extreme point as the initial point. We set $x^* \leftarrow p_1$ in Line 1, and $Q \leftarrow \{p_1\}$ in Line 2. $H(x^*) = H(p_1)$ (green dashed line) is not a supporting hyperplane of $\mathrm{conv}(P)$ in Line 4, so we move on to the else condition in Line 5.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



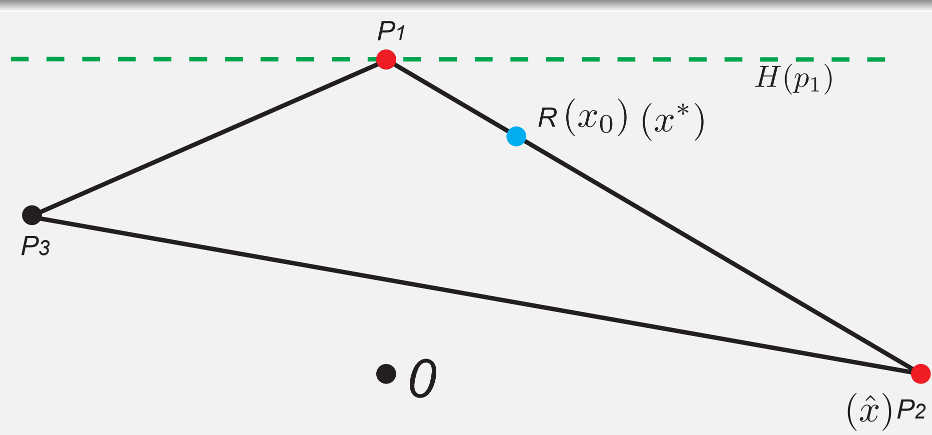
We need to add some extreme point $\hat{x}$ on the "near" side of $H(p_1)$ in Line 6, we choose $\hat{x} = p_2$. In Line 7, we set $Q \leftarrow Q \cup \{p_2\}$, so $Q = \{p_1, p_2\}$.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



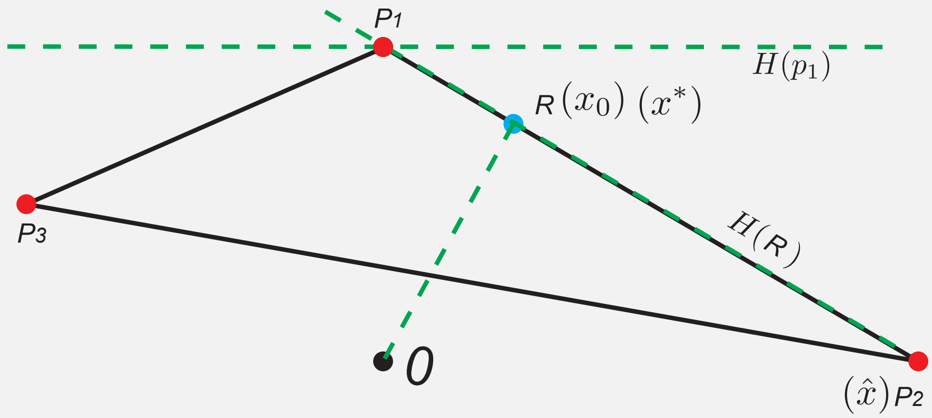
$x_0 = R$ is the min-norm point in $\text{aff}\{p_1, p_2\}$ computed in Line 9. Also, with $Q = \{p_1, p_2\}$, since $R \in \text{conv}\, Q$, we set $x^* \leftarrow x_0 = R$ in Line 11. Note, after Line 11, we still have $x^* \in P$ and $\|x^*\|_2 = \|x^*_{\text{new}}\|_2 < \|x^*_{\text{old}}\|_2$ strictly.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



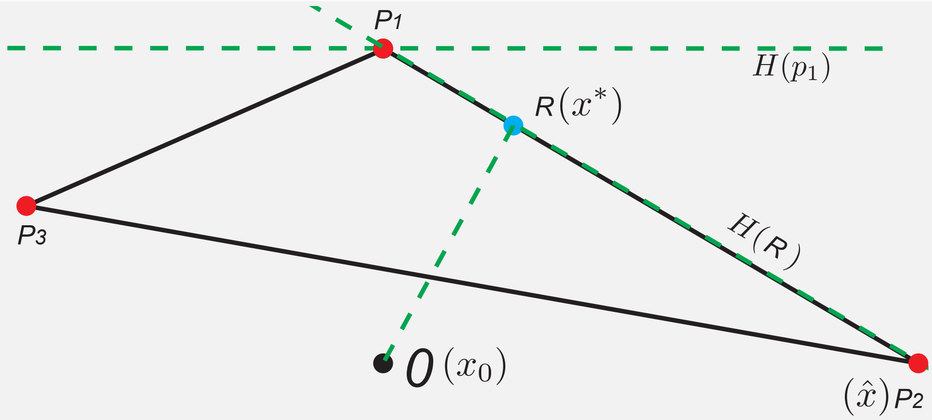
$x_0 = R$ is the min-norm point in $\text{aff}\{p_1, p_2\}$ computed in Line 9. Also, with $Q = \{p_1, p_2\}$, since $R \in \text{conv}\, Q$, we set $x^* \leftarrow x_0 = R$ in Line 11. Note, after Line 11, we still have $x^* \in P$ and $\|x^*\|_2 = \|x^*_{\text{new}}\|_2 < \|x^*_{\text{old}}\|_2$ strictly.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



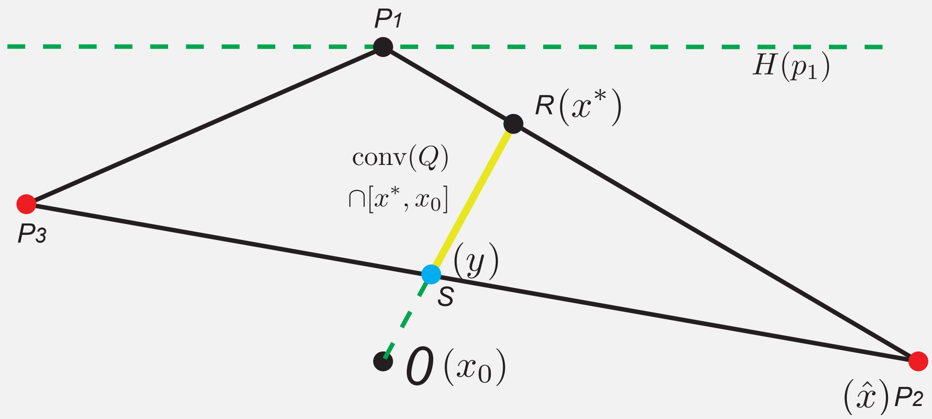
$R = x_0 = x^*$. We consider next $H(R) = H(x^*)$ in Line 4. $H(x^*)$ is not a supporting hyperplane of $\text{conv}\, P$. So we choose p_3 on the "near" side of $H(x^*)$ in Line 6. Add $Q \leftarrow Q \cup \{p_3\}$ in Line 7. Now $Q = P = \{p_1, p_2, p_3\}$. The origin $x_0 = 0$ is the min-norm point in $\text{aff}\, Q$ (Line 9), and it is not in the interior of $\text{conv}\, Q$ (condition in Line 10 is false).

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



$R = x_0 = x^*$. We consider next $H(R) = H(x^*)$ in Line 4. $H(x^*)$ is not a supporting hyperplane of $\text{conv}\, P$. So we choose p_3 on the "near" side of $H(x^*)$ in Line 6. Add $Q \leftarrow Q \cup \{p_3\}$ in Line 7. Now $Q = P = \{p_1, p_2, p_3\}$. The origin $x_0 = 0$ is the min-norm point in $\text{aff}\, Q$ (Line 9), and it is not in the interior of $\text{conv}\, Q$ (condition in Line 10 is false).

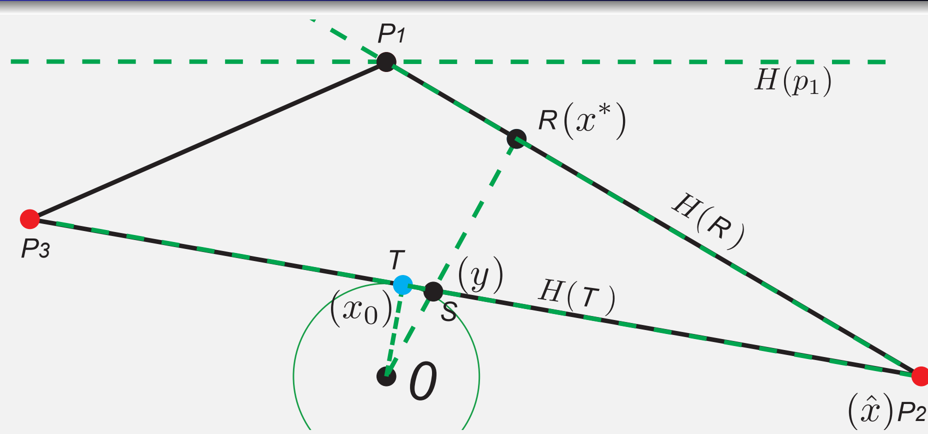
Fujishige-Wolfe Min-Norm algorithm: Geometric Example



$Q = P = \{p_1, p_2, p_3\}$. Line 14: $S = y = \min_{x \in \text{conv}\, Q \cap [x^*, x_0]} \|x - x_0\|_2$ where x_0 is 0 and x^* is R here. Thus, y lies on the boundary of $\text{conv}\, Q$. Note, $\|y\|_2 < \|x^*\|_2$ since $x^* \in \text{conv}\, Q$, $\|x_0\|_2 < \|x^*\|_2$. Line 15: Delete p_1 from Q since it is not on the face where S lies. $Q = \{p_2, p_3\}$ after Line 15. Note, we still have $y = S \in \text{conv}\, Q$ for the updated Q. Line 16: $x^* \leftarrow y$, hence we again have $\|x^*\|_2 = \|x^*_{\text{new}}\|_2 < \|x^*_{\text{old}}\|_2$ strictly.

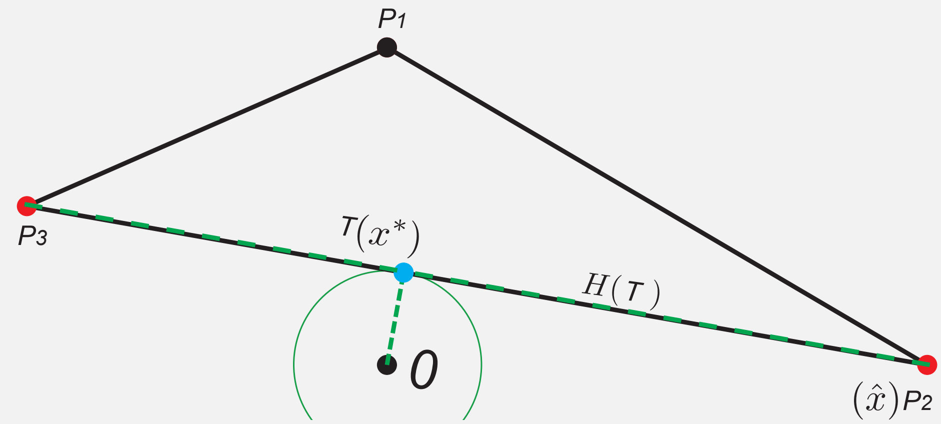
Closure/Sat | Fund. Circuit/Dep | Min-Norm Point and SFM | Min-Norm Point Algorithm | Proof that min-norm gives optimal

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



$Q = \{p_2, p_3\}$, and so $x_0 = T$ computed in Line 9 is the min-norm point in $\text{aff}\, Q$. We also have $x_0 \in \text{conv}\, Q$ in Line 10 so we assign $x^* \leftarrow x_0$ in Line 11 and break.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



$H(T)$ separates P from the origin in Line 4, and therefore is a supporting hyperplane, and therefore x^* is the min-norm point in $\operatorname{conv} P$, so we return with x^*.

Condition for Min-Norm Point

Theorem 16.6.2

$P = \{p_1, p_2, \ldots, p_m\}$, $x^* \in \operatorname{conv} P$ *is the min. norm point in* $\operatorname{conv} P$ *iff*

$$p_i^{\intercal} x^* \geq \|x^*\|_2^2 \quad \forall i = 1, \cdots, m. \tag{16.77}$$

Proof.

- Assume x^* is the min-norm point, let $y \in \operatorname{conv} P$, and $0 \leq \theta \leq 1$.
- Then $z \triangleq x^* + \theta(y - x^*) = (1-\theta)x^* + \theta y \in \operatorname{conv} P$, and

$$\|z\|_2^2 = \|x^* + \theta(y - x^*)\|_2^2 \tag{16.78}$$

$$= \|x^*\|_2^2 + 2\theta(x^{*\intercal} y - x^{*\intercal} x^*) + \theta^2 \|y - x^*\|_2^2 \tag{16.79}$$

- It is possible for $\|z\|_2^2 < \|x^*\|_2^2$ for small θ, unless $x^{*\intercal} y \geq x^{*\intercal} x^*$ for all $y \in \operatorname{conv} P \Rightarrow$ Equation (16.77).
- Conversely, given Eq (16.77), and given that $y = \sum_i \lambda_i p_i \in \operatorname{conv} P$,

$$y^{\intercal} x^* = \sum_i \lambda_i p_i^{\intercal} x^* \geq \sum_i \lambda_i x^{*\intercal} x^* = x^{*\intercal} x^* \tag{16.80}$$

implying that $\|z\|_2^2 > \|x^*\|_2^2$ in Equation 16.79 for arbitrary $z \in \operatorname{conv} P$. □

The set Q is always affinely independent

Lemma 16.6.3

The set Q in the MN Algorithm is always affinely independent.

Proof.

- Q is of course affinely independent when there is at most one point in it (e.g., after Line 2).
- After the initialization, it changes only by deletion of points, or adding a single point. Deletion does not change the independence.
- Before adding $\hat{x}$ at Line 7, we know x^* is the minimum norm point in $\text{aff}\, Q$ (since we break only at Line 11).
- Therefore, x^* is normal to $\text{aff}\, Q$, which implies $\text{aff}\, Q \subseteq H(x^*)$.
- Since $\hat{x} \notin H(x^*)$ chosen at Line 6, we have $\hat{x} \notin \text{aff}\, Q$.
- $\therefore$ update $Q \cup \{\hat{x}\}$ at Line 7 is affinely independent as long as Q is. □

Thus, by Lemma 16.6.3, we have for any $x \in \text{aff}\, Q$ such that $x = \sum_i w_i q_i$ with $\sum_i w_i = 1$, the weights w_i are uniquely determined.

Minimum Norm in an affine set

- Line 9 of the algorithm requires $x_0 \leftarrow \min_{x \in \text{aff}\, Q} \|x\|_2$.
- When Q is affinely independent, this is relatively easy.
- Let Q also represent the $n \times k$ matrix with points as columns $q \in Q$. We get the following, solvable with matrix inversion/linear solver:

$$\text{minimize} \quad \|x\|_2^2 = w^\intercal Q^\intercal Q w \tag{16.81}$$

$$\text{subject to} \quad \mathbf{1}^\intercal w = 1 \tag{16.82}$$

- Note, this also solves Line 10, since feasibility requires $\sum_i w_i = 1$, we need only check $w \geq 0$ to ensure $x_0 = \sum_i w_i q_i \in \text{conv}\, Q$.
- In fact, a feature of the algorithm (in Wolfe's 1976 paper) is that we keep the convex coefficients $\{w_i\}_i$ where $x^* = \sum_i \lambda_i p_i$ of x^* and from this vector. We also keep v such that $x_0 = \sum_i v_i q_i$ for points $q_i \in Q$, from Line 9.
  Given w and v, we can also easily solve Lines 14 and 15 (see "Step 3" on page 133 of Wolfe-1976, which also defines numerical tolerances).
- We have yet to see how to efficiently solve Lines 4 and 6, however.

MN Algorithm finds the MN point in finite time.

Theorem 16.6.4

The MN Algorithm finds the minimum norm point in $\operatorname{conv} P$ *after a finite number of iterations of the major loop.*

Proof.

- In minor loop, we always have $x^* \in \operatorname{conv} Q$, since whenever Q is modified, x^* is updated as well (Line 16) such that the updated x^* remains in new $\operatorname{conv} Q$.
- Hence, every time x^* is updated (in minor loop), its norm never increases i.e., before Line 11, $\|x_0\|_2 \leq \|x^*\|_2$ since $x^* \in \operatorname{aff} Q$ and $x_0 = \min_{x \in \operatorname{aff} Q} \|x\|_2$. Similarly, before Line 16, $\|y\|_2 \leq \|x^*\|_2$, since invariant $x^* \in \operatorname{conv} Q$ but while $x_0 \in \operatorname{aff} Q$, we have $x_0 \notin \operatorname{conv} Q$, and $\|x_0\|_2 < \|x^*\|_2$.

...

MN Algorithm finds the MN point in finite time.

... proof of Theorem 16.6.4 continued.

- Moreover, there can be no more iterations within a minor loop than the dimension of $\operatorname{conv} Q$ for the initial Q given to the minor loop initially at Line 8 (dimension of $\operatorname{conv} Q$ is $|Q| - 1$ since Q is affinely independent).
- Each iteration of the minor loop removes at least one point from Q in Line 15.
- When Q reduces to a singleton, the minor loop always terminates.
- Thus, the minor loop terminates in finite number of iterations, at most dimension of Q.
- In fact, total number of iterations of minor loop in entire algorithm is at most number of points in P since we never add back in points to Q that have been removed.

...

MN Algorithm finds the MN point in finite time.

... proof of Theorem 16.6.4 continued.

- Each time Q is augmented with $\hat{x}$ at Line 7, followed by updating x^* with x_0 at Line 11, (i.e., when the minor loop returns with only one iteration), $\|x^*\|_2$ strictly decreases from what it was before.
- To see this, consider $x^* + \theta(\hat{x} - x^*)$ where $0 \le \theta \le 1$. Since both $\hat{x}, x^* \in \operatorname{conv} Q$, we have $x^* + \theta(\hat{x} - x^*) \in \operatorname{conv} Q$.
- Therefore, we have $\|x^* + \theta(\hat{x} - x^*)\|_2 \ge \|x_0\|_2$, which implies

$$\begin{aligned}\|x^* + \theta(\hat{x} - x^*)\|_2^2 &= \|x^*\|_2^2 + 2\theta\left((x^*)^\top \hat{x} - \|x^*\|_2^2\right) + \theta^2 \|\hat{x} - x^*\|_2^2 \\ &\ge \|x_0\|_2^2 \end{aligned} \tag{16.83}$$

$\hat{x}$ is on the same side of $H(x^*)$ as the origin, i.e. $(x^*)^\top \hat{x} < \|x^*\|_2^2$.

...

MN Algorithm finds the MN point in finite time.

... proof of Theorem 16.6.4 continued.

- Therefore, for sufficiently small θ, specifically for

$$\theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2} \tag{16.84}$$

  we have that $\|x^*\|_2^2 > \|x_0\|_2^2$.
- For a similar reason, we have $\|x^*\|_2$ strictly decreases each time Q is updated at Line 7 and followed by updating x^* with y at Line 16.
- Therefore, in each iteration of major loop, $\|x^*\|_2$ strictly decreases, and the MN Algorithm must terminate and it can only do so when the optimal is found.

□

Line: 6: Finding $\hat{x} \in P$ on the near side of $H(x^*)$

- The "near" side means the side that contains the origin.
- Ideally, find $\hat{x}$ such that the reduction of $\|x^*\|_2$ is maximized to reduce number of major iterations.
- From Eqn. 16.83, reduction on norm is lower-bounded:

$$\Delta = \|x^*\|_2^2 - \|x_0\|_2^2 \geq 2\theta\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right) - \theta^2 \|\hat{x} - x^*\|_2^2 \triangleq \underline{\Delta} \tag{16.85}$$

- When $0 \leq \theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2}$, we can get the maximal value of the lower bound, over θ, as follows:

$$\max_{0 \leq \theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2}} \underline{\Delta} = \left(\frac{\|x^*\|_2^2 - (x^*)^\top \hat{x}}{\|\hat{x} - x^*\|_2}\right)^2 \tag{16.86}$$

Line: 6: Finding $\hat{x} \in P$ on the near side of $H(x^*)$

- To maximize lower bound of norm reduction at each major iteration, want to find an $\hat{x}$ such that the above lower bound (Equation 16.86) is maximized.
- That is, we want to find

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \left(\frac{\|x^*\|_2^2 - (x^*)^\top x}{\|x - x^*\|_2}\right)^2 \tag{16.87}$$

  to ensure that a large norm reduction is assured.
- This problem, however, is at least as hard as the MN problem itself as we have a quadratic term in the denominator.

Line: 6: Finding $\hat{x} \in P$ on the near side of $H(x^*)$

- As a surrogate, we maximize numerator in Eqn. 16.87, i.e., find

$$\hat{x} \in \underset{x \in P}{\operatorname{argmax}} \|x^*\|_2^2 - (x^*)^\top x = \underset{x \in P}{\operatorname{argmin}} (x^*)^\top x, \tag{16.88}$$

- Intuitively, by solving the above, we find $\hat{x}$ such that it has the largest distance to the hyperplane $H(x^*)$, and this is exactly the strategy used in the Wolfe-1976 algorithm.
- Also, solution $\hat{x}$ can be used to determine if hyperplane $H(x^*)$ separates $\operatorname{conv} P$ from the origin (Line 4): if the point in P having greatest distance to $H(x^*)$ is not on the side where origin lies, then $H(x^*)$ separates $\operatorname{conv} P$ from the origin.
- Mathematically, we terminate the algorithm if

$$(x^*)^\top \hat{x} \geq \|x^*\|_2^2, \tag{16.89}$$

where $\hat{x}$ is the solution of Eq. 16.88.

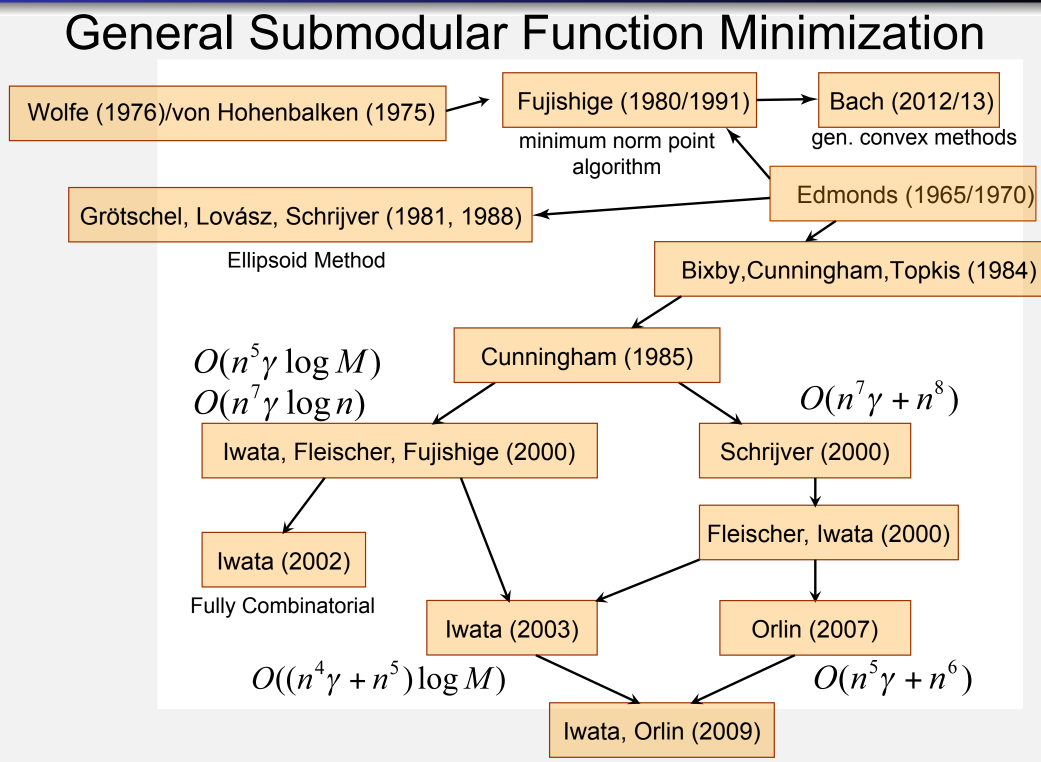
Line: 6: Finding $\hat{x} \in P$ on the near side of $H(x^*)$

- In practice,the above optimality test might never hold numerically. Hence, as suggested by Wolfe, we introduce a tolerance parameter $\epsilon > 0$, and terminates the algorithm if

$$(x^*)^\top \hat{x} > \|x^*\|_2^2 - \epsilon \max_{x \in Q} \|x\|_2^2 \tag{16.90}$$

- When $\operatorname{conv} P$ is a submodular base polytope (i.e., $\operatorname{conv} P = B_f$ for a submodular function f), then the problem in Eqn 16.88 can be solved efficiently by Edmonds's greedy algorithm (even though there may be an exponential number of extreme points).
- Hence, Edmonds's discovery is one of the main reasons that the MN algorithm is applicable to submodular function minimization.

SFM Summary (modified from S. Iwata's slides)



MN Algorithm Complexity

- The currently fastest strongly polynomial combinatorial algorithm for SFM achieves a running time of $O(n^5 T + n^6)$ (Orlin'09) where T is the time for function evaluation, far from practical for large problem instances.
- Fujishige & Isotani report that MN algorithm is fast in practice, but they use only a limited set of submodular functions.
- Complexity of MN Algorithm is still an unsolved problem.
- Obvious facts:
  - each major iteration requires $O(n)$ function oracle calls
  - complexity of each major iteration could be at least $O(n^3)$ due to the affine projection step (solving a linear system).
  - Therefore, the complexity of each major iteration is

    $$O(n^3 + n^{1+p})$$

    where each function oracle call requires $O(n^p)$ time.
- Since the number of major iterations required is unknown, the complexity of MN is also unknown.

MN Algorithm Empirical Complexity

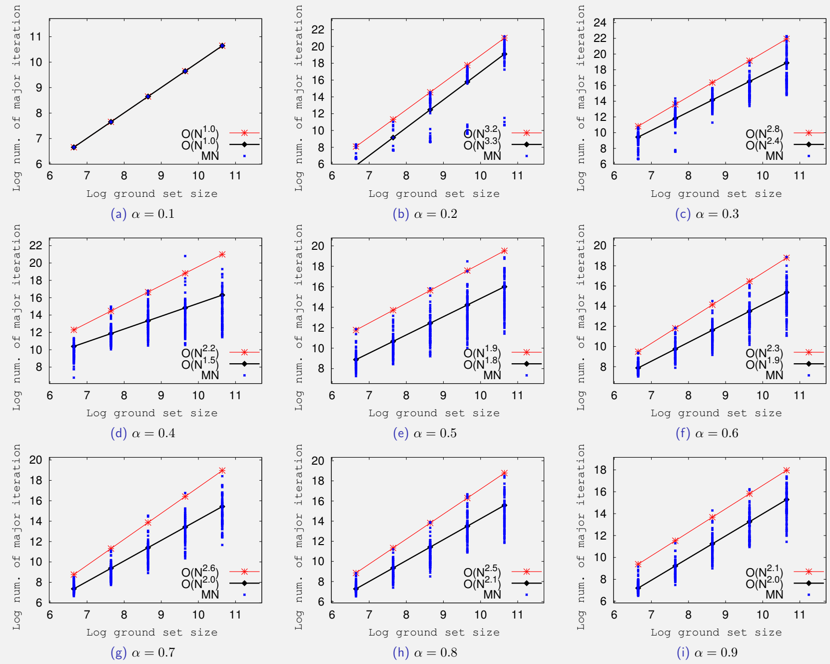


Figure: The number of major iteration for $f(S) = -m_1(S) + 100 \cdot (w_1(\mathcal{N}(S)))^\alpha$. The red lines are the linear interpolations of the worst case points, and the black lines are the linear interpolations of the average case points. From Lin&Bilmes 2014 (unpublished)

Min-Norm Point and SFM

Theorem 16.7.1

*Let y^*, A_-, and A_0 be as given. Then y^* is a maximizer of the l.h.s. of Eqn. (**??**). Moreover, A_- is the unique minimal minimizer of f and A_0 is the unique maximal minimizer of f.*

Proof.

- First note, since $x^* \in B_f$, we have $x^*(E) = f(E)$, meaning $\text{sat}(x^*) = E$. Thus, we can consider any $e \in E$ within $\text{dep}(x^*, e)$.
- Consider any pair (e, e') with $e' \in \text{dep}(x^*, e)$ and $e \in A_-$. Then $x^*(e) < 0$, and $\exists \alpha > 0$ s.t. $x^* + \alpha \mathbf{1}_e - \alpha \mathbf{1}_{e'} \in P_f$.
- We have $x^*(E) = f(E)$ and x^* is minimum in l2 sense. We have $(x^* + \alpha \mathbf{1}_e - \alpha \mathbf{1}_{e'}) \in P_f$, and in fact

$$(x^* + \alpha \mathbf{1}_e - \alpha \mathbf{1}_{e'})(E) = x^*(E) + \alpha - \alpha = f(E) \tag{16.91}$$

so $x^* + \alpha \mathbf{1}_e - \alpha \mathbf{1}_{e'} \in B_f$ also.

...

Min-Norm Point and SFM

. . . proof of Thm. 16.7.1 cont.

- Then $(x^* + \alpha \mathbf{1}_e - \alpha \mathbf{1}_{e'})(E)$
  $= x^*(E \setminus \{e, e'\}) + \underbrace{(x^*(e) + \alpha)}_{x^*_{\text{new}}(e)} + \underbrace{(x^*(e') - \alpha)}_{x^*_{\text{new}}(e')} = f(E).$
- Minimality of $x^* \in B_f$ in l2 sense requires that, with such an $\alpha > 0$,
  $\left(x^*(e)\right)^2 + \left(x^*(e')\right)^2 < \left(x^*_{\text{new}}(e)\right)^2 + \left(x^*_{\text{new}}(e')\right)^2$
- Given that $e \in A_-$, $x^*(e) < 0$. Thus, if $x^*(e') > 0$, we could have $(x^*(e) + \alpha)^2 + (x^*(e') - \alpha)^2 < (x^*(e))^2 + (x^*(e'))^2$, contradicting the optimality of x^*.
- If $x^*(e') = 0$, we would have $(x^*(e) + \alpha)^2 + (\alpha)^2 < (x^*(e))^2$, for any $0 < \alpha < |x^*(e)|$ (Exercise:), again contradicting the optimality of x^*.
- Thus, we must have $x^*(e') < 0$ (strict negativity).

. . .

Min-Norm Point and SFM

. . . proof of Thm. 16.7.1 cont.

- Thus, for a pair (e, e') with $e' \in \text{dep}(x^*, e)$ and $e \in A_-$, we have $x(e') < 0$ and hence $e' \in A_-$.
- Hence, $\forall e \in A_-$, we have $\text{dep}(x^*, e) \subseteq A_-$.
- A very similar argument can show that, $\forall e \in A_0$, we have $\text{dep}(x^*, e) \subseteq A_0$.

. . .

Min-Norm Point and SFM

. . . proof of Thm. 16.7.1 cont.

- Therefore, we have $\cup_{e \in A_-} \text{dep}(x^*, e) = A_-$ and $\cup_{e \in A_0} \text{dep}(x^*, e) = A_0$
- Ie., $\{\text{dep}(x^*, e)\}_{e \in A_-}$ is cover for A_-, as is $\{\text{dep}(x^*, e)\}_{e \in A_0}$ for A_0.
- $\text{dep}(x^*, e)$ is minimal tight set containing e, meaning $x^*(\text{dep}(x^*, e)) = f(\text{dep}(x^*, e))$, and since tight sets are closed under union, we have that A_- and A_0 are also tight, meaning:

$$x^*(A_-) = f(A_-) \tag{16.92}$$
$$x^*(A_0) = f(A_0) \tag{16.93}$$
$$x^*(A_-) = x^*(A_0) = y^*(E) = y^*(A_0) + \underbrace{y^*(E \setminus A_0)}_{=0} \tag{16.94}$$

and therefore, all together we have

$$f(A_-) = f(A_0) = x^*(A_-) = x^*(A_0) = y^*(E) \tag{16.95}$$

. . .

Min-Norm Point and SFM

. . . proof of Thm. 16.7.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (**??**). This follows since, we have $y^* = x^* \wedge 0 \leq 0$, and since $x^* \in B_f \subset P_f$, and $y^* \leq x^*$ and P_f is down-closed, we have that $y^* \in P_f$.
- Also, for any $y \in P_f$ with $y \leq 0$ and for any $X \subseteq E$, we have $y(E) \leq y(X) \leq f(X)$.
- Hence, we have found a feasible for l.h.s. of Eqn. (**??**), $y^* \leq 0$, $y^* \in P_f$, so $y^*(E) \leq f(X)$ for all X.
- So $y^*(E) \leq \min\{f(X) | X \subseteq V\}$.
- Considering Eqn. (16.96), we have found sets A_- and A_0 with tightness in Eqn. (**??**), meaning $y^*(E) = f(A_-) = f(A_0)$.
- Hence, y^* is a maximizer of l.h.s. of Eqn. (**??**), and A_- and A_0 are minimizers of f.

. . .

Min-Norm Point and SFM

. . . proof of Thm. 16.7.1 cont.

- Now, for any $X \subset A_-$, we have

$$f(X) \geq x^*(X) > x^*(A_-) = f(A_-) \tag{16.96}$$

- And for any $X \supset A_0$, we have

$$f(X) \geq x^*(X) > x^*(A_0) = f(A_0) \tag{16.97}$$

- Hence, A_- must be the unique minimal minimizer of f, and A_0 is the unique maximal minimizer of f.

□

Min-Norm Point and SFM

- So, if we have a procedure to compute the min-norm point computation, we can solve SFM.
- Nice thing about previous proof is that it uses both expressions for dep for different purposes.
- This was discovered by Fujishige (in fact the proof above is an expanded version of the one found in the book).
- An algorithm (by F. Wolfe) can find this min-norm point, essentially an active-set procedure for quadratic programming. It uses Edmonds's greedy algorithm to make it efficient.
- This is currently the best practical algorithm for general purpose submodular function minimization.
- But its underlying lower-bound complexity is unknown, although in practice its estimated empirical complexity runs anywhere from $O(n^3)$ to $O(n^{4.5})$ or so (see Jegelka, Lin, Bilmes (NIPS 2011)).

Min-norm point and other minimizers of f

- Recall, that the set of minimizers of f forms a lattice.
- In fact, with x^* the min-norm point, and A_- and A_0 as defined above, we have the following theorem:

Theorem 16.7.2

Let $A \subseteq E$ be any minimizer of submodular f, and let x^* be the minimum-norm point. Then A has the form:

$$A = A_- \cup \bigcup_{a \in A_m} \text{dep}(x^*, a) \tag{16.98}$$

for some set $A_m \subseteq A_0 \setminus A_-$.

Min-norm point and other minimizers of f

proof of Thm. 16.7.2.

- If A is a minimizer, then $A_- \subseteq A \subseteq A_0$, and $f(A) = y^*(E)$ is the minimum valuation of f.
- But $x^* \in P_f$, so $x^*(A) \leq f(A)$ and $f(A) = x^*(A_-) \leq x^*(A)$ (or alternatively, just note that $x^*(A_0 \setminus A) = 0$).
- Hence, $x^*(A) = x^*(A_-) = f(A)$ so that A is also a tight set for x^*.
- For any $a \in A$, A is a tight set containing a, and $\text{dep}(x^*, a)$ is the minimal tight containing a.
- Hence, for any $a \in A$, $\text{dep}(x^*, a) \subseteq A$.
- This means that $\bigcup_{a \in A} \text{dep}(x^*, a) = A$.
- Since $A_- \subseteq A \subseteq A_0$, then $\exists A_m \subseteq A \setminus A_-$ such that

$$A = \bigcup_{a \in A_-} \text{dep}(x^*, a) \cup \bigcup_{a \in A_m} \text{dep}(x^*, a) = A_- \cup \bigcup_{a \in A_m} \text{dep}(x^*, a)$$

□

On a unique minimizer f

- Note that if $f(e|A) > 0, \forall A \subseteq E$ and $e \in E \setminus A$, then we have $A_- = A_0$ (there is one unique minimizer).
- On the other hand, if $A_- = A_0$, it does not imply $f(e|A) > 0$ for all $A \subseteq E \setminus \{e\}$.
- If $A_- = A_0$ then certainly $f(e|A_0) > 0$ for $e \in E \setminus A_0$ and $-f(e|A_0 \setminus \{e\}) > 0$ for all $e \in A_0$.