Submodular Functions, Optimization, and Applications to Machine Learning

— Spring Quarter, Lecture 18 —

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

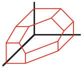
Prof. Jeff Bilmes

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

May 30th, 2018

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

$= f(A_r) + 2f(C) + f(B_r)$ $\quad = f(A_r) + f(C) + f(B_r)$ $\quad = f(A \cap B)$



Announcements, Assignments, and Reminders

- Take home final exam (like long homework). Due Friday, June 8th, 4:00pm via our assignment dropbox (https://canvas.uw.edu/courses/1216339/assignments).
- Get started now. At least read through everything and ask any questions you might have.
- As always, if you have any questions about anything, please ask then via our discussion board (https://canvas.uw.edu/courses/1216339/discussion_topics). Can meet at odd hours via zoom (send message on canvas to schedule time to chat).

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): Graph and Combinatorial Examples, Matrix Rank, Examples and Properties, visualizations
- L5(4/9): More Examples/Properties/ Other Submodular Defs., Independence,
- L6(4/11): Matroids, Matroid Examples, Matroid Rank, Partition/Laminar Matroids
- L7(4/16): Laminar Matroids, System of Distinct Reps, Transversals, Transversal Matroid, Matroid Representation, Dual Matroids
- L8(4/18): Dual Matroids, Other Matroid Properties, Combinatorial Geometries, Matroids and Greedy.
- L9(4/23): Polyhedra, Matroid Polytopes, Matroids $\rightarrow$ Polymatroids
- L10(4/29): Matroids $\rightarrow$ Polymatroids, Polymatroids, Polymatroids and Greedy,
- L11(4/30): Polymatroids, Polymatroids and Greedy
- L12(5/2): Polymatroids and Greedy, Extreme Points, Cardinality Constrained Maximization
- L13(5/7): Constrained Submodular Maximization
- L14(5/9): Submodular Max w. Other Constraints, Cont. Extensions, Lovasz Extension
- L15(5/14): Cont. Extensions, Lovasz Extension, Choquet Integration, Properties
- L16(5/16): More Lovasz extension, Choquet, defs/props, examples, multiliear extension
- L17(5/21): Finish L.E., Multilinear Extension, Submodular Max/polyhedral approaches, Most Violated inequality, Still More on Matroids, Closure/Sat
- L–(5/28): Memorial Day (holiday)
- L18(5/30): Closure/Sat, Fund. Circuit/Dep, Min-Norm Point Definitions, Proof that min-norm gives optimal Review & Support for Min-Norm, Computing Min-Norm Vector for B_f
- L21(6/4): Final Presentations maximization.

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

Most violated inequality problem in matroid polytope case

- Consider

$$P_r^+ = \left\{x \in \mathbb{R}^E : x \geq 0, x(A) \leq r_M(A), \forall A \subseteq E\right\} \tag{18.22}$$

- 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:

$$\max\left\{x(A) - r_M(A) : A \in \mathcal{W}\right\} = \max\left\{x(A) - r_M(A) : A \subseteq E\right\} \tag{18.23}$$

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

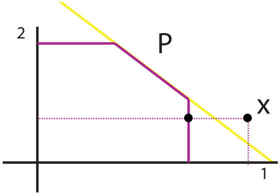
$$\min\left\{r_M(A) + x(E \setminus A) : A \subseteq E\right\} \tag{18.24}$$

Most violated inequality/polymatroid membership/SFM

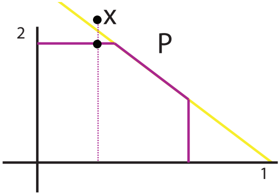
- Consider

$$P_f^+ = \{x \in \mathbb{R}^E : x \geq 0, x(A) \leq f(A), \forall A \subseteq E\} \tag{18.22}$$

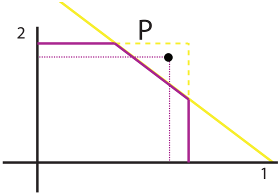
- Suppose we have any $x \in \mathbb{R}_+^E$ such that $x \notin P_f^+$.
- Hence, there must be a set of $\mathcal{W} \subseteq 2^V$, each member of which corresponds to a violated inequality, i.e., equations of the form $x(A) > r_M(A)$ for $A \in \mathcal{W}$.



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



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



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

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:

$$\max \{x(A) - f(A) : A \in \mathcal{W}\} = \max \{x(A) - f(A) : A \subseteq E\} \quad (18.22)$$

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

$$\min \{f(A) + x(E \setminus A) : A \subseteq E\} \quad (18.23)$$

- 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 18.2.5

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.

$\square$

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 $e \notin \text{span}(I)$ (i.e., when $I + e$ is independent), then we set $C(I, e) = \emptyset$.

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

$$\mathcal{D}(x) = \{A : A \subseteq E, x(A) = f(A)\} \tag{18.23}$$

Minimizers of a Submodular Function form a lattice

Theorem 18.2.6

For arbitrary submodular f, the minimizers are closed under union and intersection. That is, let $\mathcal{M} = \operatorname{argmin}_{X \subseteq E} f(X)$ 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

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \tag{18.25}$$

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:

$$\mathsf{cl}(x) \stackrel{\text{def}}{=} \mathrm{sat}(x) \stackrel{\text{def}}{=} \bigcup \{A : A \in \mathcal{D}(x)\} \tag{18.25}$$

$$= \bigcup \{A : A \subseteq E, x(A) = f(A)\} \tag{18.26}$$

$$= \{e : e \in E, \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f\} \tag{18.27}$$

- Hence, $\mathrm{sat}(x)$ is the maximal (zero-valued) minimizer of the submodular function $f_x(A) \triangleq f(A) - x(A)$.
- Eq. (??) says that sat consists of elements of E for point x that are 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 18.2.6 (Matroid $\text{sat} : \mathbb{R}_+^E \to 2^E$ is the same as closure.)

For $I \in \mathcal{I}$, we have $$\text{sat}(\mathbf{1}_I) = \text{span}(I) \tag{18.29}$$

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, span, and submodular function minimization

- Thus, for a matroid, $\mathrm{sat}(\mathbf{1}_I)$ is exactly the closure (or span) of I in the matroid. I.e., for matroid (E, r), we have $\mathrm{span}(I) = \mathrm{sat}(\mathbf{1}_B)$.

The sat function, span, and submodular function minimization

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

The sat function, span, and submodular function minimization

- Thus, for a matroid, $\mathrm{sat}(\mathbf{1}_I)$ is exactly the closure (or span) of I in the matroid. I.e., for matroid (E, r), we have $\mathrm{span}(I) = \mathrm{sat}(\mathbf{1}_B)$.
- Recall, for $x \in P_f$ and polymatroidal f, $\mathrm{sat}(x)$ is the maximal (by inclusion) minimizer of $f(A) - x(A)$, and thus in a matroid, 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.

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

$$\text{sat}(x) = \bigcup \{A : x(A) = f(A)\} \tag{18.1}$$

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

$$\text{sat}(x) = \bigcup \{A : x(A) = f(A)\} \tag{18.1}$$

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

$$\text{sat}(x) = \left\{e : \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f^+\right\} \tag{18.2}$$

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

$$\text{sat}(x) = \bigcup \{A : x(A) = f(A)\} \tag{18.1}$$

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

$$\text{sat}(x) = \left\{e : \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f^+\right\} \tag{18.2}$$

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

sat, as tight polymatroidal elements

- Lets start with one definition and derive the other.

$$\mathrm{sat}(x)$$

sat, as tight polymatroidal elements

- Lets start with one definition and derive the other.

$$\text{sat}(x) \stackrel{\text{def}}{=} \left\{ e : \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f^+ \right\} \tag{18.3}$$

sat, as tight polymatroidal elements

- Lets start with one definition and derive the other.

$$\operatorname{sat}(x) \stackrel{\text{def}}{=} \left\{e : \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f^+\right\} \tag{18.3}$$

$$= \{e : \forall \alpha > 0, \exists A \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A)\} \tag{18.4}$$

sat, as tight polymatroidal elements

- Lets start with one definition and derive the other.

$$\text{sat}(x) \stackrel{\text{def}}{=} \left\{e : \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f^+\right\} \tag{18.3}$$

$$= \{e : \forall \alpha > 0, \exists A \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A)\} \tag{18.4}$$

$$= \{e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A)\} \tag{18.5}$$

sat, as tight polymatroidal elements

- Lets start with one definition and derive the other.

$$\text{sat}(x) \stackrel{\text{def}}{=} \left\{ e : \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f^+ \right\} \tag{18.3}$$

$$= \{ e : \forall \alpha > 0, \exists A \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A) \} \tag{18.4}$$

$$= \{ e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A) \} \tag{18.5}$$

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

sat, as tight polymatroidal elements

- Lets start with one definition and derive the other.

$$\text{sat}(x) \stackrel{\text{def}}{=} \left\{e : \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f^+\right\} \tag{18.3}$$
$$= \{e : \forall \alpha > 0, \exists A \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A)\} \tag{18.4}$$
$$= \{e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A)\} \tag{18.5}$$

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

$$\text{sat}(x) = \{e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } x(A) + \alpha > f(A)\} \tag{18.6}$$

sat, as tight polymatroidal elements

- Lets start with one definition and derive the other.

$$\mathrm{sat}(x) \stackrel{\text{def}}{=} \left\{e : \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f^+ \right\} \tag{18.3}$$

$$= \{e : \forall \alpha > 0, \exists A \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A)\} \tag{18.4}$$

$$= \{e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A)\} \tag{18.5}$$

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

$$\mathrm{sat}(x) = \{e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } x(A) + \alpha > f(A)\} \tag{18.6}$$

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

$$\mathrm{sat}(x)$$

sat, as tight polymatroidal elements

- Lets start with one definition and derive the other.

$$\text{sat}(x) \stackrel{\text{def}}{=} \left\{e : \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f^+\right\} \tag{18.3}$$

$$= \{e : \forall \alpha > 0, \exists A \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A)\} \tag{18.4}$$

$$= \{e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A)\} \tag{18.5}$$

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

$$\text{sat}(x) = \{e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } x(A) + \alpha > f(A)\} \tag{18.6}$$

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

$$\text{sat}(x) = \{e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } x(A) = f(A)\} \tag{18.7}$$

sat, as tight polymatroidal elements

- Lets start with one definition and derive the other.

$$\begin{aligned}\mathrm{sat}(x) &\stackrel{\text{def}}{=} \left\{e : \forall\alpha > 0, x + \alpha\mathbf{1}_e \notin P_f^+\right\} && (18.3)\\ &= \{e : \forall\alpha > 0, \exists A \text{ s.t. } (x + \alpha\mathbf{1}_e)(A) > f(A)\} && (18.4)\\ &= \{e : \forall\alpha > 0, \exists A \ni e \text{ s.t. } (x + \alpha\mathbf{1}_e)(A) > f(A)\} && (18.5)\end{aligned}$$

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

$$\mathrm{sat}(x) = \{e : \forall\alpha > 0, \exists A \ni e \text{ s.t. } x(A) + \alpha > f(A)\} \qquad (18.6)$$

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

$$\begin{aligned}\mathrm{sat}(x) &= \{e : \forall\alpha > 0, \exists A \ni e \text{ s.t. } x(A) = f(A)\} && (18.7)\\ &= \{e : \exists A \ni e \text{ s.t. } x(A) = f(A)\} && (18.8)\end{aligned}$$

sat, as tight polymatroidal elements

- Lets start with one definition and derive the other.

$$\begin{aligned}\mathrm{sat}(x) &\stackrel{\mathrm{def}}{=} \left\{e : \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f^+\right\} && (18.3)\\ &= \{e : \forall \alpha > 0, \exists A \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A)\} && (18.4)\\ &= \{e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A)\} && (18.5)\end{aligned}$$

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

$$\mathrm{sat}(x) = \{e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } x(A) + \alpha > f(A)\} \qquad (18.6)$$

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

$$\begin{aligned}\mathrm{sat}(x) &= \{e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } x(A) = f(A)\} && (18.7)\\ &= \{e : \exists A \ni e \text{ s.t. } x(A) = f(A)\} && (18.8)\end{aligned}$$

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

(18.9)

sat, as tight polymatroidal elements

- Lets start with one definition and derive the other.

$$\mathrm{sat}(x) \stackrel{\text{def}}{=} \left\{ e : \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f^+ \right\} \tag{18.3}$$

$$= \{ e : \forall \alpha > 0, \exists A \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A) \} \tag{18.4}$$

$$= \{ e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A) \} \tag{18.5}$$

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

$$\mathrm{sat}(x) = \{ e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } x(A) + \alpha > f(A) \} \tag{18.6}$$

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

$$\mathrm{sat}(x) = \{ e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } x(A) = f(A) \} \tag{18.7}$$

$$= \{ e : \exists A \ni e \text{ s.t. } x(A) = f(A) \} \tag{18.8}$$

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

$$\forall e \in A, e \in \mathrm{sat}(x), \tag{18.9}$$

sat, as tight polymatroidal elements

- Lets start with one definition and derive the other.

$$\begin{aligned}
\mathrm{sat}(x) &\stackrel{\text{def}}{=} \left\{e : \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f^+\right\} && (18.3)\\
&= \{e : \forall \alpha > 0, \exists A \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A)\} && (18.4)\\
&= \{e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } (x + \alpha \mathbf{1}_e)(A) > f(A)\} && (18.5)
\end{aligned}$$

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

$$\mathrm{sat}(x) = \{e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } x(A) + \alpha > f(A)\} \qquad (18.6)$$

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

$$\begin{aligned}
\mathrm{sat}(x) &= \{e : \forall \alpha > 0, \exists A \ni e \text{ s.t. } x(A) = f(A)\} && (18.7)\\
&= \{e : \exists A \ni e \text{ s.t. } x(A) = f(A)\} && (18.8)
\end{aligned}$$

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

$$\forall e \in A, e \in \mathrm{sat}(x), \text{ and therefore that } \mathrm{sat}(x) \supseteq A \qquad (18.9)$$

sat, as tight polymatroidal elements

- . . . and therefore, with sat as defined in Eq. (17.35),

$$\mathrm{sat}(x) \supseteq \bigcup \{A : x(A) = f(A)\} \tag{18.10}$$

sat, as tight polymatroidal elements

- . . . and therefore, with sat as defined in Eq. (17.35),

$$\operatorname{sat}(x) \supseteq \bigcup \{A : x(A) = f(A)\} \tag{18.10}$$

- On the other hand, for any $e \in \operatorname{sat}(x)$ defined as in Eq. (18.8), since e is itself a member of a tight set, there is a set $A \ni e$ such that $x(A) = f(A)$, giving

$$\operatorname{sat}(x) \subseteq \bigcup \{A : x(A) = f(A)\} \tag{18.11}$$

sat, as tight polymatroidal elements

- . . . and therefore, with sat as defined in Eq. (17.35),

$$\mathrm{sat}(x) \supseteq \bigcup \{A : x(A) = f(A)\} \tag{18.10}$$

- On the other hand, for any $e \in \mathrm{sat}(x)$ defined as in Eq. (18.8), since e is itself a member of a tight set, there is a set $A \ni e$ such that $x(A) = f(A)$, giving

$$\mathrm{sat}(x) \subseteq \bigcup \{A : x(A) = f(A)\} \tag{18.11}$$

- Therefore, the two definitions of sat are identical.

Saturation Capacity

- Another useful concept is saturation capacity which we develop next.

Saturation Capacity

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

$$\max \{\alpha : \alpha \in \mathbb{R}, x + \alpha \mathbf{1}_e \in P_f\} \tag{18.12}$$

Saturation Capacity

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

$$\max \{\alpha : \alpha \in \mathbb{R}, x + \alpha \mathbf{1}_e \in P_f\} \tag{18.12}$$

- This is identical to:

$$\max \{\alpha : (x + \alpha \mathbf{1}_e)(A) \leq f(A), \forall A \supseteq \{e\}\} \tag{18.13}$$

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

Saturation Capacity

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

$$\max\{\alpha : \alpha \in \mathbb{R}, x + \alpha \mathbf{1}_e \in P_f\} \tag{18.12}$$

- This is identical to:

$$\max\{\alpha : (x + \alpha \mathbf{1}_e)(A) \leq f(A), \forall A \supseteq \{e\}\} \tag{18.13}$$

  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:

$$\max\{\alpha : x(A) + \alpha \leq f(A), \forall A \supseteq \{e\}\} \tag{18.14}$$

Saturation Capacity

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

$$\max \{\alpha : \alpha \in \mathbb{R}, x + \alpha \mathbf{1}_e \in P_f\} \tag{18.12}$$

- This is identical to:

$$\max \{\alpha : (x + \alpha \mathbf{1}_e)(A) \leq f(A), \forall A \supseteq \{e\}\} \tag{18.13}$$

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:

$$\max \{\alpha : x(A) + \alpha \leq f(A), \forall A \supseteq \{e\}\} \tag{18.14}$$

or

$$\max \{\alpha : \alpha \leq f(A) - x(A), \forall A \supseteq \{e\}\} \tag{18.15}$$

Saturation Capacity

- The max is achieved when

$$\alpha = \hat{c}(x;e) \stackrel{\text{def}}{=} \min\{f(A) - x(A), \forall A \supseteq \{e\}\} \tag{18.16}$$

Saturation Capacity

- The max is achieved when

$$\alpha = \hat{c}(x; e) \stackrel{\text{def}}{=} \min \{ f(A) - x(A), \forall A \supseteq \{e\} \} \tag{18.16}$$

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

Saturation Capacity

- The max is achieved when

$$\alpha = \hat{c}(x;e) \stackrel{\text{def}}{=} \min\{f(A) - x(A), \forall A \supseteq \{e\}\} \tag{18.16}$$

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

$$\hat{c}(x;e) \stackrel{\text{def}}{=} \min\{f(A) - x(A), \forall A \ni e\} \tag{18.17}$$
$$= \max\{\alpha : \alpha \in \mathbb{R}, x + \alpha \mathbf{1}_e \in P_f\} \tag{18.18}$$

Saturation Capacity

- The max is achieved when

$$\alpha = \hat{c}(x;e) \stackrel{\text{def}}{=} \min\{f(A) - x(A), \forall A \supseteq \{e\}\} \tag{18.16}$$

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

$$\hat{c}(x;e) \stackrel{\text{def}}{=} \min\{f(A) - x(A), \forall A \ni e\} \tag{18.17}$$

$$= \max\{\alpha : \alpha \in \mathbb{R}, x + \alpha \mathbf{1}_e \in P_f\} \tag{18.18}$$

- We immediately see that for $e \in E \setminus \operatorname{sat}(x)$, we have that $\hat{c}(x;e) > 0$.

Saturation Capacity

- The max is achieved when

$$\alpha = \hat{c}(x; e) \stackrel{\text{def}}{=} \min \{ f(A) - x(A), \forall A \supseteq \{e\} \} \tag{18.16}$$

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

$$\hat{c}(x; e) \stackrel{\text{def}}{=} \min \{ f(A) - x(A), \forall A \ni e \} \tag{18.17}$$

$$= \max \{ \alpha : \alpha \in \mathbb{R}, x + \alpha \mathbf{1}_e \in P_f \} \tag{18.18}$$

- We immediately see that for $e \in E \setminus \text{sat}(x)$, we have that $\hat{c}(x; e) > 0$.
- Also, we have that: $e \in \text{sat}(x) \Leftrightarrow \hat{c}(x; e) = 0$.

Saturation Capacity

- The max is achieved when

$$\alpha = \hat{c}(x;e) \stackrel{\text{def}}{=} \min \{f(A) - x(A), \forall A \supseteq \{e\}\} \tag{18.16}$$

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

$$\hat{c}(x;e) \stackrel{\text{def}}{=} \min \{f(A) - x(A), \forall A \ni e\} \tag{18.17}$$

$$= \max \{\alpha : \alpha \in \mathbb{R}, x + \alpha \mathbf{1}_e \in P_f\} \tag{18.18}$$

- We immediately see that for $e \in E \setminus \text{sat}(x)$, we have that $\hat{c}(x;e) > 0$.
- Also, we have that: $e \in \text{sat}(x) \Leftrightarrow \hat{c}(x;e) = 0$.
- Note that any α with $0 \leq \alpha \leq \hat{c}(x;e)$ we have $x + \alpha \mathbf{1}_e \in P_f$.

Saturation Capacity

- The max is achieved when

$$\alpha = \hat{c}(x;e) \stackrel{\text{def}}{=} \min\{f(A) - x(A), \forall A \supseteq \{e\}\} \tag{18.16}$$

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

$$\hat{c}(x;e) \stackrel{\text{def}}{=} \min\{f(A) - x(A), \forall A \ni e\} \tag{18.17}$$

$$= \max\{\alpha : \alpha \in \mathbb{R}, x + \alpha \mathbf{1}_e \in P_f\} \tag{18.18}$$

- We immediately see that for $e \in E \setminus \text{sat}(x)$, we have that $\hat{c}(x;e) > 0$.
- Also, we have that: $e \in \text{sat}(x) \Leftrightarrow \hat{c}(x;e) = 0$.
- 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.

Dependence Function

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

$$\mathcal{D}(x, e) = \{A : e \in A \subseteq E, x(A) = f(A)\} \tag{18.19}$$

$$= \mathcal{D}(x) \cap \{A : A \subseteq E, e \in A\} \tag{18.20}$$

Dependence Function

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

$$\mathcal{D}(x, e) = \{A : e \in A \subseteq E, x(A) = f(A)\} \tag{18.19}$$
$$= \mathcal{D}(x) \cap \{A : A \subseteq E, e \in A\} \tag{18.20}$$

- Thus, $\mathcal{D}(x, e) \subseteq \mathcal{D}(x)$, and $\mathcal{D}(x, e)$ is a sublattice of $\mathcal{D}(x)$.

Dependence Function

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

$$\mathcal{D}(x,e) = \{A : e \in A \subseteq E, x(A) = f(A)\} \tag{18.19}$$
$$= \mathcal{D}(x) \cap \{A : A \subseteq E, e \in A\} \tag{18.20}$$

- 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:

$$\text{dep}(x,e) = \begin{cases} \bigcap\{A : e \in A \subseteq E, x(A) = f(A)\} & \text{if } e \in \text{sat}(x) \\ \emptyset & \text{else} \end{cases} \tag{18.21}$$

Dependence Function

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

$$\mathcal{D}(x,e) = \{A : e \in A \subseteq E, x(A) = f(A)\} \tag{18.19}$$

$$= \mathcal{D}(x) \cap \{A : A \subseteq E, e \in A\} \tag{18.20}$$

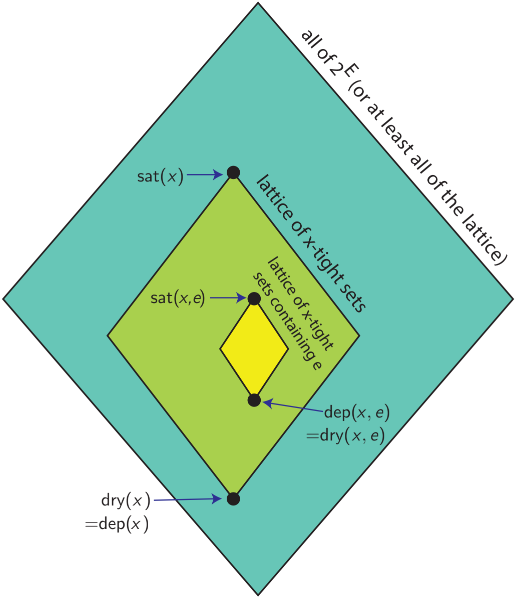
- Thus, $\mathcal{D}(x,e) \subseteq \mathcal{D}(x)$, and $\mathcal{D}(x,e)$ is a sublattice of $\mathcal{D}(x)$.
- Therefore, we can define a unique minimal element of $\mathcal{D}(x,e)$ denoted as follows:

$$\text{dep}(x,e) = \begin{cases} \bigcap \{A : e \in A \subseteq E, x(A) = f(A)\} & \text{if } e \in \text{sat}(x) \\ \emptyset & \text{else} \end{cases} \tag{18.21}$$

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

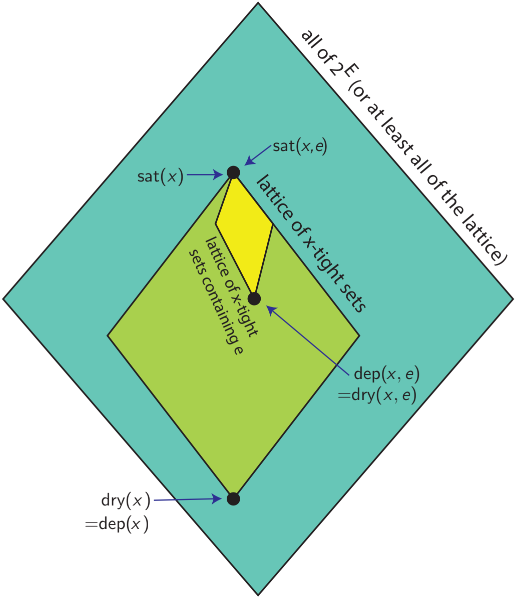
dep and sat in a lattice

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



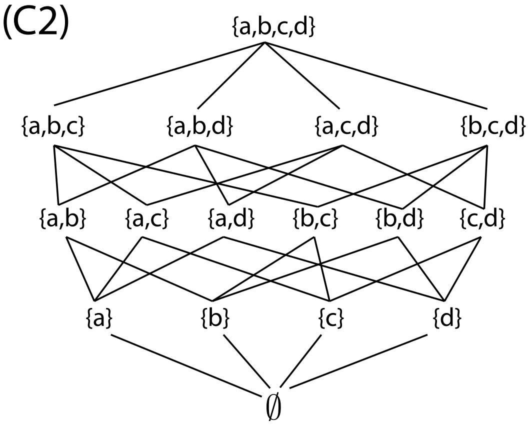
dep and sat in a lattice

- Given some $x \in P_f$,
- The picture on the right summarizes the relationships between the lattices and sublattices.
- Note, $\text{dep}(x, e) \supseteq \text{dep}(x) = \bigcap \{A : x(A) = f(A)\}$.
- In fact, $\text{sat}(x, e) = \text{sat}(x)$. Why?



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, $\text{dep}(x, e) \supseteq \text{dep}(x) = \bigcap \{A : x(A) = f(A)\}$.
- In fact, $\text{sat}(x, e) = \text{sat}(x)$. Why?
- Example lattice on 4 elements.



dep and sat in a lattice

- Given $x \in P_f$, recall distributive lattice of tight sets $\mathcal{D}(x) = \{A : x(A) = f(A)\}$

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.

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., $\text{dry}(x) \stackrel{\text{def}}{=} \bigcap \{A : A \in \mathcal{D}(x)\}$

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.

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

$$\text{dry}(x) = \{e' : x(A) < f(A), \forall A \not\ni e'\} \tag{18.22}$$

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

$$\text{dry}(x) = \left\{e' : x(A) < f(A), \forall A \not\ni e'\right\} \tag{18.22}$$

- 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).

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

$$\text{dry}(x) = \left\{e' : x(A) < f(A), \forall A \not\ni e'\right\} \tag{18.22}$$

- 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 ntight(x), for the necessary for tightness (but we'll actually use neither name).

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

$$\text{dry}(x) = \left\{e' : x(A) < f(A), \forall A \not\ni e'\right\} \tag{18.22}$$

- 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 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 $\mathrm{dep} = \mathrm{dry}$: restated

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

$$\mathrm{dry}(x,e) = \left\{e' : x(A) < f(A), \forall A \not\ni e', e \in A\right\} \tag{18.23}$$

- This can be read as, for any $e' \in \mathrm{dry}(x,e)$, any e-containing set that does not contain e' is not tight for x.
- But actually, $\mathrm{dry}(x,e) = \mathrm{dep}(x,e)$, so we have derived another expression for $\mathrm{dep}(x,e)$ in Eq. (18.23).

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 $\text{sat}(\mathbf{1}_I) = \text{span}(I) = \text{closure}(I)$.

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 $\mathrm{sat}(\mathbf{1}_I) = \mathrm{span}(I) = \mathrm{closure}(I)$.
- Given $e \in \mathrm{sat}(\mathbf{1}_I) \setminus I$ and then consider an $A \ni e$ with $|I \cap A| = r(A)$.

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).

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 $\mathrm{sat}(\mathbf{1}_I) = \text{span}(I) = \text{closure}(I)$.
- Given $e \in \mathrm{sat}(\mathbf{1}_I) \setminus I$ 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 $e \in \mathrm{sat}(\mathbf{1}_I) \setminus I$, and consider $\mathrm{dep}(\mathbf{1}_I, e)$, with

$$\mathrm{dep}(\mathbf{1}_I, e) = \bigcap \{A : e \in A \subseteq E, \mathbf{1}_I(A) = r(A)\} \tag{18.24}$$

$$= \bigcap \{A : e \in A \subseteq E, |I \cap A| = r(A)\} \tag{18.25}$$

$$= \bigcap \{A : e \in A \subseteq E, r(A) - |I \cap A| = 0\} \tag{18.26}$$

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 $\mathrm{sat}(\mathbf{1}_I) = \text{span}(I) = \text{closure}(I)$.
- Given $e \in \mathrm{sat}(\mathbf{1}_I) \setminus I$ 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 $e \in \mathrm{sat}(\mathbf{1}_I) \setminus I$, and consider $\mathrm{dep}(\mathbf{1}_I, e)$, with

$$\mathrm{dep}(\mathbf{1}_I, e) = \bigcap \{A : e \in A \subseteq E, \mathbf{1}_I(A) = r(A)\} \tag{18.24}$$

$$= \bigcap \{A : e \in A \subseteq E, |I \cap A| = r(A)\} \tag{18.25}$$

$$= \bigcap \{A : e \in A \subseteq E, r(A) - |I \cap A| = 0\} \tag{18.26}$$

- By SFM lattice, $\exists$ a unique minimal $A \ni e$ with $|I \cap A| = r(A)$.

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

$$\text{dep}(\mathbf{1}_I, e) = \bigcap \{A : e \in A \subseteq E, \mathbf{1}_I(A) = r(A)\} \tag{18.24}$$

$$= \bigcap \{A : e \in A \subseteq E, |I \cap A| = r(A)\} \tag{18.25}$$

$$= \bigcap \{A : e \in A \subseteq E, r(A) - |I \cap A| = 0\} \tag{18.26}$$

- 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).

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 $C(I, e) = \{e\}$

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., $|I \cap A| \geq |I \cap \{e\}| = r(e) = 1$.

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.

Dependence Function and Fundamental Matroid Circuit

- Therefore, when $e \in \mathrm{sat}(\mathbf{1}_I) \setminus I$, then $\mathrm{dep}(\mathbf{1}_I, e) = C(I, e)$ 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 $e \in \mathrm{sat}(\mathbf{1}_I) \cap I$ 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 $\mathrm{dep}(\mathbf{1}_I, e) = \{e\}$ 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 $e \in \mathrm{sat}(x)$, we have $\mathrm{dep}(x, e)$ is tight by definition (i.e., $x(\mathrm{dep}(x, e)) = f(\mathrm{dep}(x, e))$).

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,

$$\text{cl}(x) \stackrel{\text{def}}{=} \text{sat}(x) \triangleq \bigcup \{A : A \in \mathcal{D}(x)\} \tag{18.27}$$

$$= \bigcup \{A : A \subseteq E, x(A) = f(A)\} \tag{18.28}$$

$$= \{e : e \in E, \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f\} \tag{18.29}$$

- 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.,

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

$$= \{e' : \exists \alpha > 0, \text{ s.t. } x + \alpha(\mathbf{1}_e - \mathbf{1}_{e'}) \in P_f\} \tag{18.30}$$

Note, if $x + \alpha(\mathbf{1}_e - \mathbf{1}_{e'}) \in P_f$, then $x + \alpha'(\mathbf{1}_e - \mathbf{1}_{e'}) \in P_f$ for any $0 \le \alpha' < \alpha$.

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.

Dependence Function and exchange

- For $e \in \operatorname{span}(I) \setminus I$, we have that $I + e \notin \mathcal{I}$. This is a set addition restriction property.
- Analogously, for $e \in \operatorname{sat}(x)$, any $x + \alpha \mathbf{1}_e \notin P_f$ for $\alpha > 0$. This is a vector increase restriction property.

Dependence Function and exchange

- For $e \in \mathrm{span}(I) \setminus I$, we have that $I + e \notin \mathcal{I}$. This is a set addition restriction property.
- Analogously, for $e \in \mathrm{sat}(x)$, 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.

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

$$C(I, e) = \{a \in E : I + e - a \in \mathcal{I}\} \tag{18.31}$$

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

$$C(I, e) = \{a \in E : I + e - a \in \mathcal{I}\} \tag{18.31}$$

- I.e., an addition of e to I stays within $\mathcal{I}$ only if we simultaneously remove one of the elements of $C(I, e)$.

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

$$C(I, e) = \{a \in E : I + e - a \in \mathcal{I}\} \tag{18.31}$$

- 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?

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

$$C(I,e) = \{a \in E : I + e - a \in \mathcal{I}\} \tag{18.31}$$

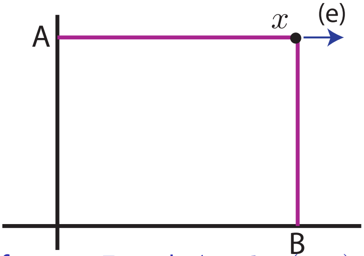
- I.e., an addition of e to I stays within $\mathcal{I}$ only if we simultaneously remove one of the elements of $C(I,e)$.
- But, analogous to the circuit case, is there an exchange property for $\text{dep}(x,e)$ in the form of vector movement restriction?
- We might expect the vector $\text{dep}(x,e)$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

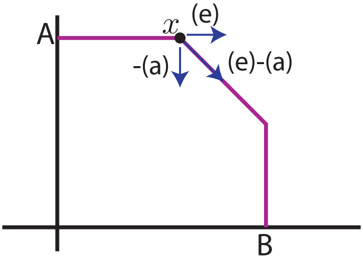
Dependence Function and exchange in 2D

- $\mathrm{dep}(x, e)$ 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.

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.

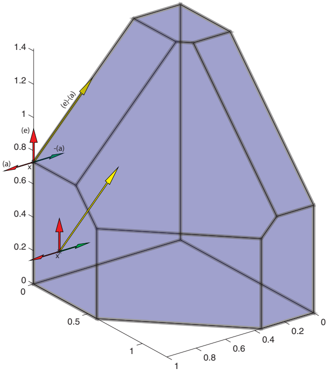
Right: $A \subseteq \text{dep}(x, e)$. We can't move further in the (e) direction, but we can move further in (e) direction by moving in some negative $a \in A$ direction. **Dependence** between (e) and elements in A.

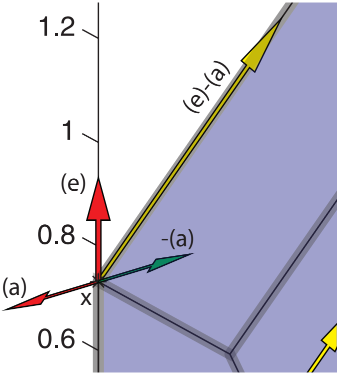
Dependence Function and exchange in 3D

- 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.

Dependence Function and exchange in 3D

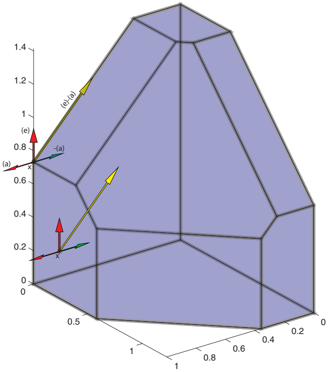
- We can move neither in the (e) nor the (a) direction, but we can move in the (e) direction if we simultaneously move in the -(a) direction.
- In 3D, we have:

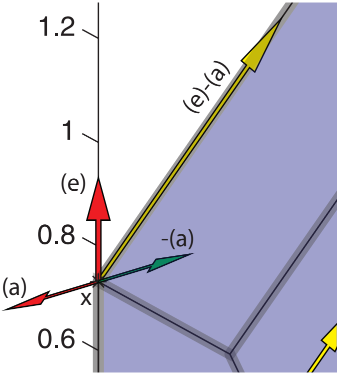




Dependence Function and exchange in 3D

- 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:

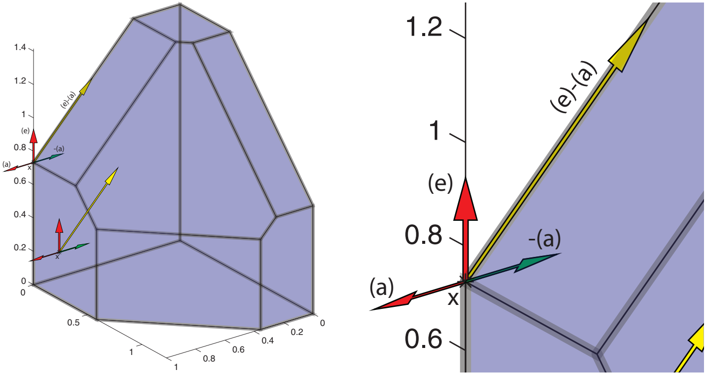




- I.e., for $e \in \text{sat}(x)$, $a \in \text{sat}(x)$, $a \in \text{dep}(x, e)$, $e \notin \text{dep}(x, a)$,

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:

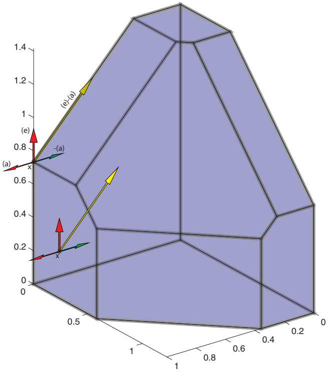


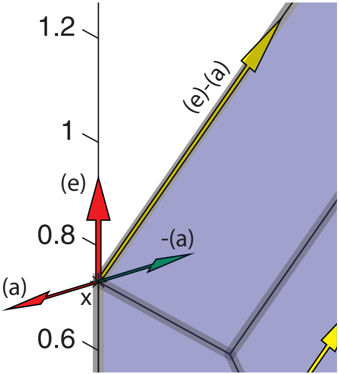
- I.e., for $e \in \text{sat}(x)$, $a \in \text{sat}(x)$, $a \in \text{dep}(x, e)$, $e \notin \text{dep}(x, a)$, and

$$\text{dep}(x, e) = \{a : a \in E, \exists \alpha > 0 : x + \alpha(\mathbf{1}_e - \mathbf{1}_a) \in P_f\} \tag{18.32}$$

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

$$\text{dep}(x, e) = \{a : a \in E, \exists \alpha > 0 : x + \alpha(\mathbf{1}_e - \mathbf{1}_a) \in P_f\} \quad (18.32)$$

- 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:

$$\text{dep}(x, e) = \text{ntight}(x, e) = \tag{18.33}$$

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:

$$\text{dep}(x, e) = \mathsf{ntight}(x, e) = \tag{18.33}$$

$$= \{e' : x(A) < f(A), \forall A \not\ni e', e \in A\} \tag{18.34}$$

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:

$$\text{dep}(x, e) = \text{ntight}(x, e) = \tag{18.33}$$

$$= \{e' : x(A) < f(A), \forall A \not\ni e', e \in A\} \tag{18.34}$$

$$= \{e' : \exists \alpha > 0, \text{ s.t. } \alpha \le f(A) - x(A), \forall A \not\ni e', e \in A\} \tag{18.35}$$

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:

$$\text{dep}(x, e) = \text{ntight}(x, e) = \tag{18.33}$$

$$= \{e' : x(A) < f(A), \forall A \not\ni e', e \in A\} \tag{18.34}$$

$$= \{e' : \exists \alpha > 0, \text{ s.t. } \alpha \leq f(A) - x(A), \forall A \not\ni e', e \in A\} \tag{18.35}$$

$$= \{e' : \exists \alpha > 0, \text{ s.t. } \alpha \mathbf{1}_e(A) \leq f(A) - x(A), \forall A \not\ni e', e \in A\} \tag{18.36}$$

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:

$$\text{dep}(x, e) = \text{ntight}(x, e) = \tag{18.33}$$

$$= \{e' : x(A) < f(A), \forall A \not\ni e', e \in A\} \tag{18.34}$$

$$= \{e' : \exists \alpha > 0, \text{ s.t. } \alpha \leq f(A) - x(A), \forall A \not\ni e', e \in A\} \tag{18.35}$$

$$= \{e' : \exists \alpha > 0, \text{ s.t. } \alpha \mathbf{1}_e(A) \leq f(A) - x(A), \forall A \not\ni e', e \in A\} \tag{18.36}$$

$$= \{e' : \exists \alpha > 0, \text{ s.t. } \alpha(\mathbf{1}_e(A) - \mathbf{1}_{e'}(A)) \leq f(A) - x(A), \forall A \not\ni e', e \in A\} \tag{18.37}$$

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:

$$\text{dep}(x, e) = \text{ntight}(x, e) = \tag{18.33}$$

$$= \{e' : x(A) < f(A), \forall A \not\ni e', e \in A\} \tag{18.34}$$

$$= \{e' : \exists \alpha > 0, \text{ s.t. } \alpha \leq f(A) - x(A), \forall A \not\ni e', e \in A\} \tag{18.35}$$

$$= \{e' : \exists \alpha > 0, \text{ s.t. } \alpha \mathbf{1}_e(A) \leq f(A) - x(A), \forall A \not\ni e', e \in A\} \tag{18.36}$$

$$= \{e' : \exists \alpha > 0, \text{ s.t. } \alpha(\mathbf{1}_e(A) - \mathbf{1}_{e'}(A)) \leq f(A) - x(A), \forall A \not\ni e', e \in A\} \tag{18.37}$$

$$= \{e' : \exists \alpha > 0, \text{ s.t. } x(A) + \alpha(\mathbf{1}_e(A) - \mathbf{1}_{e'}(A)) \leq f(A), \forall A \not\ni e', e \in A\} \tag{18.38}$$

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:

$$\text{dep}(x, e) = \text{ntight}(x, e) = \tag{18.33}$$

$$= \{e' : x(A) < f(A), \forall A \not\ni e', e \in A\} \tag{18.34}$$

$$= \{e' : \exists \alpha > 0, \text{ s.t. } \alpha \leq f(A) - x(A), \forall A \not\ni e', e \in A\} \tag{18.35}$$

$$= \{e' : \exists \alpha > 0, \text{ s.t. } \alpha \mathbf{1}_e(A) \leq f(A) - x(A), \forall A \not\ni e', e \in A\} \tag{18.36}$$

$$= \{e' : \exists \alpha > 0, \text{ s.t. } \alpha(\mathbf{1}_e(A) - \mathbf{1}_{e'}(A)) \leq f(A) - x(A), \forall A \not\ni e', e \in A\} \tag{18.37}$$

$$= \{e' : \exists \alpha > 0, \text{ s.t. } x(A) + \alpha(\mathbf{1}_e(A) - \mathbf{1}_{e'}(A)) \leq f(A), \forall A \not\ni e', e \in A\} \tag{18.38}$$

- Now, $1_e(A) - \mathbf{1}_{e'}(A) = 0$ if either $\{e, e'\} \subseteq A$, or $\{e, e'\} \cap A = \emptyset$.

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:

$$\text{dep}(x, e) = \text{ntight}(x, e) = \tag{18.33}$$

$$= \{e' : x(A) < f(A), \forall A \not\ni e', e \in A\} \tag{18.34}$$

$$= \{e' : \exists \alpha > 0, \text{ s.t. } \alpha \leq f(A) - x(A), \forall A \not\ni e', e \in A\} \tag{18.35}$$

$$= \{e' : \exists \alpha > 0, \text{ s.t. } \alpha \mathbf{1}_e(A) \leq f(A) - x(A), \forall A \not\ni e', e \in A\} \tag{18.36}$$

$$= \{e' : \exists \alpha > 0, \text{ s.t. } \alpha(\mathbf{1}_e(A) - \mathbf{1}_{e'}(A)) \leq f(A) - x(A), \forall A \not\ni e', e \in A\} \tag{18.37}$$

$$= \{e' : \exists \alpha > 0, \text{ s.t. } x(A) + \alpha(\mathbf{1}_e(A) - \mathbf{1}_{e'}(A)) \leq f(A), \forall A \not\ni e', e \in A\} \tag{18.38}$$

- 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 $x(A) + \alpha(\mathbf{1}_e(A) - \mathbf{1}_{e'}(A)) = x(A) - \alpha \leq f(A)$ 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

$$\text{dep}(x, e) = \{e' : \exists \alpha > 0, \text{ s.t. } x(A) + \alpha(\mathbf{1}_e(A) - \mathbf{1}_{e'}(A)) \leq f(A), \forall A\} \tag{18.39}$$

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

$$\text{dep}(x, e) = \{e' : \exists \alpha > 0, \text{ s.t. } x(A) + \alpha(\mathbf{1}_e(A) - \mathbf{1}_{e'}(A)) \leq f(A), \forall A\} \tag{18.39}$$

- This is then identical to

$$\text{dep}(x, e) = \{e' : \exists \alpha > 0, \text{ s.t. } x + \alpha(\mathbf{1}_e - \mathbf{1}_{e'}) \in P_f\} \tag{18.40}$$

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

$$\text{dep}(x, e) = \{e' : \exists \alpha > 0, \text{ s.t. } x(A) + \alpha(\mathbf{1}_e(A) - \mathbf{1}_{e'}(A)) \leq f(A), \forall A\} \tag{18.39}$$

- This is then identical to

$$\text{dep}(x, e) = \{e' : \exists \alpha > 0, \text{ s.t. } x + \alpha(\mathbf{1}_e - \mathbf{1}_{e'}) \in P_f\} \tag{18.40}$$

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

$$\text{dep}(x, e) = \begin{cases} \bigcap \{A : e \in A \subseteq E, x(A) = f(A)\} & \text{if } e \in \text{sat}(x) \\ \emptyset & \text{else} \end{cases} \tag{18.41}$$

Summary of Concepts

- Most violated inequality $\max\{x(A) - f(A) : A \subseteq E\}$

Summary of Concepts

- Most violated inequality $\max\{x(A) - f(A) : A \subseteq E\}$
- Matroid by circuits, and the fundamental circuit $C(I, e) \subseteq I + e$.

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.

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.

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.

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

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

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

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 $x \in P_f$, $\mathcal{D}(x) \triangleq \{A \subseteq E : x(A) = f(A)\}$.

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 $x \in P_f$, $\text{sat}(x) \triangleq \cup\{A : A \in \mathcal{D}(x)\} = \{e : e \in E, \forall \alpha > 0, x + \alpha \mathbf{1}_e \notin P_f\}$.

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 $x \in P_f$, $0 \leq \hat{c}(x; e) \triangleq \min\{f(A) - x(A) | \forall A \ni e\} = \max\{\alpha : \alpha \in \mathbb{R}, x + \alpha \mathbf{1}_e \in P_f\}$

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 $E \setminus \text{sat}(x) = \{e : \hat{c}(x; e) > 0\}$.

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 $x \in P_f$, $\mathcal{D}(x, e) = \{A : e \in A \subseteq E, x(A) = f(A)\} \subseteq \mathcal{D}(x)$.

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 $x \in P_f$,

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

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

Submodular Function Minimization (SFM) and Min-Norm

- We saw that SFM can be used to solve most violated inequality problems for a given $x \in P_f$ and, in general, SFM can solve the question "Is $x \in P_f$" by seeing if x violates any inequality (if the most violated one is negative, solution to SFM, then $x \in P_f$).
- Unconstrained SFM, $\min_{A \subseteq V} f(A)$ solves many other problems as well in combinatorial optimization, machine learning, and other fields.
- We next study an algorithm, the "Fujishige-Wolfe Algorithm", or what is known as the "Minimum Norm Point" algorithm, which is an active set method to do this, and one that in practice works about as well as anything else people (so far) have tried for general purpose SFM.
- Note special case SFM can be much faster.

Min-Norm Point: Definition

- Consider the optimization:

$$\begin{aligned} &\text{minimize} && \|x\|_2^2 && \text{(18.42a)} \\ &\text{subject to} && x \in B_f && \text{(18.42b)} \end{aligned}$$

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.

Min-Norm Point: Definition

- Consider the optimization:

$$\text{minimize} \quad \|x\|_2^2 \tag{18.42a}$$

$$\text{subject to} \quad x \in B_f \tag{18.42b}$$

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.

Min-Norm Point: Definition

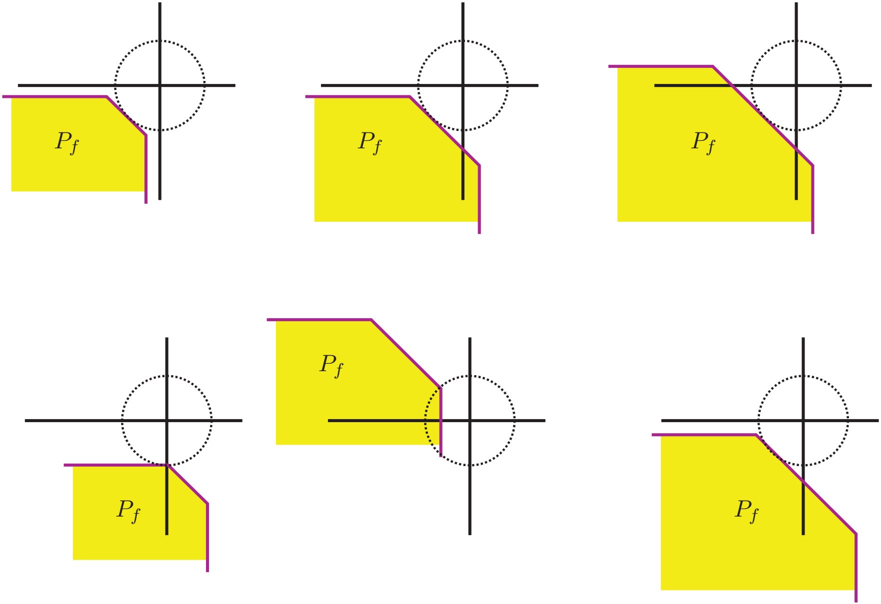
- Consider the optimization:

$$\begin{aligned} &\text{minimize} && \|x\|_2^2 && (18.42a) \\ &\text{subject to} && x \in B_f && (18.42b) \end{aligned}$$

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: Examples

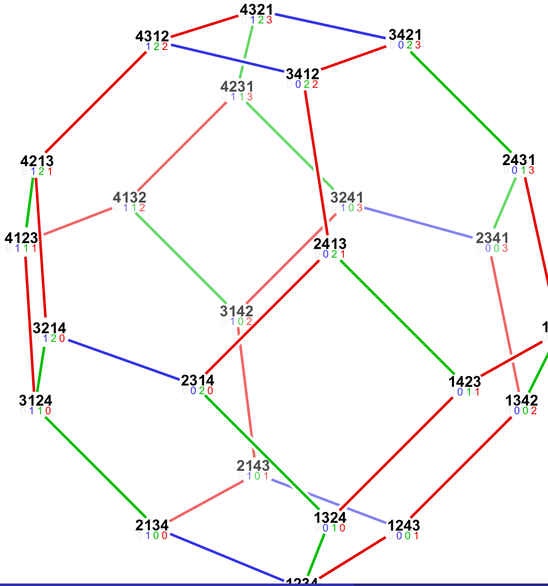


Ex: 3D base B_f: permutahedron

- 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).



Min-Norm Point and Submodular Function Minimization

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

$$y^* = x^* \wedge 0 = (\min(x^*(e), 0) | e \in E) \tag{18.43}$$

$$A_- = \{e : x^*(e) < 0\} \tag{18.44}$$

$$A_0 = \{e : x^*(e) \leq 0\} \tag{18.45}$$

Min-Norm Point and Submodular Function Minimization

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

$$y^* = x^* \wedge 0 = (\min(x^*(e), 0) | e \in E) \tag{18.43}$$

$$A_- = \{e : x^*(e) < 0\} \tag{18.44}$$

$$A_0 = \{e : x^*(e) \leq 0\} \tag{18.45}$$

- Thus, we immediately have that:

$$A_- \subseteq A_0 \tag{18.46}$$

and that

$$x^*(A_-) = x^*(A_0) = y^*(A_-) = y^*(A_0) \tag{18.47}$$

Min-Norm Point and Submodular Function Minimization

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

$$y^* = x^* \wedge 0 = (\min(x^*(e), 0) | e \in E) \tag{18.43}$$

$$A_- = \{e : x^*(e) < 0\} \tag{18.44}$$

$$A_0 = \{e : x^*(e) \leq 0\} \tag{18.45}$$

- Thus, we immediately have that:

$$A_- \subseteq A_0 \tag{18.46}$$

and that

$$x^*(A_-) = x^*(A_0) = y^*(A_-) = y^*(A_0) \tag{18.47}$$

- It turns out, these quantities will solve the submodular function minimization problem, as we now show.

Min-Norm Point and Submodular Function Minimization

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

$$y^* = x^* \wedge 0 = (\min(x^*(e), 0) | e \in E) \tag{18.43}$$
$$A_- = \{e : x^*(e) < 0\} \tag{18.44}$$
$$A_0 = \{e : x^*(e) \leq 0\} \tag{18.45}$$

- Thus, we immediately have that:

$$A_- \subseteq A_0 \tag{18.46}$$

and that

$$x^*(A_-) = x^*(A_0) = y^*(A_-) = y^*(A_0) \tag{18.47}$$

- 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.

More about the base B_f

Theorem 18.6.1

Let f be a polymatroid function and suppose that E can be partitioned into $(E_1, E_2, \ldots, E_k)$ such that $f(A) = \sum_{i=1}^{k} f(A \cap E_i)$ for all $A \subseteq E$, and k is maximum. Then the base polytope $B_f = \{x \in P_f : x(E) = f(E)\}$ (the E-tight subset of P_f) has dimension $|E| - k$.

More about the base B_f

Theorem 18.6.1

Let f be a polymatroid function and suppose that E can be partitioned into $(E_1, E_2, \dots, E_k)$ such that $f(A) = \sum_{i=1}^k f(A \cap E_i)$ for all $A \subseteq E$, and k is maximum. Then the base polytope $B_f = \{x \in P_f : x(E) = f(E)\}$ (the E-tight subset of P_f) has dimension $|E| - k$.

- In fact, every $x \in P_f$ is dominated by $x \leq y \in B_f$.

Theorem 18.6.2

If $x \in P_f$ and T is tight for x (meaning $x(T) = f(T)$), then there exists $y \in B_f$ with $x \leq y$ and $y(e) = x(e)$ for $e \in T$.

More about the base B_f

Theorem 18.6.1

Let f be a polymatroid function and suppose that E can be partitioned into $(E_1, E_2, \dots, E_k)$ such that $f(A) = \sum_{i=1}^{k} f(A \cap E_i)$ for all $A \subseteq E$, and k is maximum. Then the base polytope $B_f = \{x \in P_f : x(E) = f(E)\}$ (the E-tight subset of P_f) has dimension $|E| - k$.

- In fact, every $x \in P_f$ is dominated by $x \leq y \in B_f$.

Theorem 18.6.2

If $x \in P_f$ and T is tight for x (meaning $x(T) = f(T)$), then there exists $y \in B_f$ with $x \leq y$ and $y(e) = x(e)$ for $e \in T$.

- We will prove these after we describe min-norm algorithm.

Review from Lecture 12

The following slide repeats Theorem 12.3.2 from lecture 12 and is one of the most important theorems in submodular theory.

A polymatroid function's polyhedron is a polymatroid.

Theorem 18.6.1

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

$$\textit{rank}(x) = \max\left(y(E) : y \leq x, y \in P_f\right) = \min\left(x(A) + f(E \setminus A) : A \subseteq E\right) \tag{18.1}$$

Essentially the same theorem as Theorem 10.4.1, but note P_f rather than P_f^+. Taking $x = 0$ we get:

Corollary 18.6.2

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

$$\textit{rank}(0) = \max\left(y(E) : y \leq 0, y \in P_f\right) = \min\left(f(A) : A \subseteq E\right) \tag{18.2}$$

Modified max-min theorem

- Min-max theorem (Thm 12.3.2) restated for $x = 0$.

$$\max \{y(E) | y \in P_f, y \leq 0\} = \min \{f(X) | X \subseteq V\} \tag{18.48}$$

Modified max-min theorem

- Min-max theorem (Thm 12.3.2) restated for $x = 0$.

$$\max \{y(E) | y \in P_f, y \leq 0\} = \min \{f(X) | X \subseteq V\} \tag{18.48}$$

Theorem 18.6.3 (Edmonds-1970)

$$\min \{f(X) | X \subseteq E\} = \max \left\{x^-(E) | x \in B_f\right\} \tag{18.49}$$

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

Modified max-min theorem

- Min-max theorem (Thm 12.3.2) restated for $x = 0$.

$$\max\{y(E) | y \in P_f, y \leq 0\} = \min\{f(X) | X \subseteq V\} \tag{18.48}$$

Theorem 18.6.3 (Edmonds-1970)

$$\min\{f(X) | X \subseteq E\} = \max\left\{x^-(E) | x \in B_f\right\} \tag{18.49}$$

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

Proof via the Lovász ext.

$$\min\{f(X) | X \subseteq E\} = \min_{w \in [0,1]^E} \tilde{f}(w) = \min_{w \in [0,1]^E} \max_{x \in P_f} w^\intercal x \tag{18.50}$$

$$= \min_{w \in [0,1]^E} \max_{x \in B_f} w^\intercal x \tag{18.51}$$

$$= \max_{x \in B_f} \min_{w \in [0,1]^E} w^\intercal x \tag{18.52}$$

$$= \max_{x \in B_f} x^-(E) \tag{18.53}$$

$\square$

Alternate proof of modified max-min theorem

We start directly from Theorem 12.3.2.

$$\max\left(y(E) : y \leq 0, y \in P_f\right) = \min\left(f(A) : A \subseteq E\right) \tag{18.57}$$

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

$$\begin{aligned}
\max\left(y(E) : y \leq 0, y \in P_f\right) &= \max\left(y^-(E) : y \leq 0, y \in P_f\right) && (18.58)\\
&= \max\left(y^-(E) : y \in P_f\right) && (18.59)\\
&= \max\left(y^-(E) : y \in B_f\right) && (18.60)
\end{aligned}$$

The first equality follows since $y \leq 0$. For the second equality will be shown on the following slide. 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 18.6.2).

Alternate proof of modified max-min theorem

Consider the following two problems:

$$\max \sum_{e \in E} y(e) \tag{18.61a}$$
$$\text{s.t. } y \leq x \tag{18.61b}$$
$$y \in P \tag{18.61c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{18.62a}$$
$$\text{s.t. } y \in P \tag{18.62b}$$

- Solutions identical cost. Let y_1^* be l.h.s. OPT and y_2^* be r.h.s. OPT.
- Consider y_1^* as r.h.s. solution and suppose it is worse than r.h.s. OPT:

$$\sum_{e \in E} \min(y_1^*(e), x(e)) < \sum_{e \in E} \min(y_2^*(e), x(e)) \tag{18.63}$$

Alternate proof of modified max-min theorem

Consider the following two problems:

$$\max \sum_{e \in E} y(e) \tag{18.61a}$$
$$\text{s.t. } y \leq x \tag{18.61b}$$
$$y \in P \tag{18.61c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{18.62a}$$
$$\text{s.t. } y \in P \tag{18.62b}$$

- Solutions identical cost. Let y_1^* be l.h.s. OPT and y_2^* be r.h.s. OPT.
- Consider y_1^* as r.h.s. solution and suppose it is worse than r.h.s. OPT:

$$\sum_{e \in E} \min(y_1^*(e), x(e)) < \sum_{e \in E} \min(y_2^*(e), x(e)) \tag{18.63}$$

- Hence, $\exists e'$ s.t. $y_1^*(e') < \min(y_2^*(e'), x(e'))$. Recall $y_1^*, y_2^* \in P$.

Alternate proof of modified max-min theorem

Consider the following two problems:

$$\max \sum_{e \in E} y(e) \tag{18.61a}$$
$$\text{s.t. } y \leq x \tag{18.61b}$$
$$y \in P \tag{18.61c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{18.62a}$$
$$\text{s.t. } y \in P \tag{18.62b}$$

- Solutions identical cost. Let y_1^* be l.h.s. OPT and y_2^* be r.h.s. OPT.
- Consider y_1^* as r.h.s. solution and suppose it is worse than r.h.s. OPT:

$$\sum_{e \in E} \min(y_1^*(e), x(e)) < \sum_{e \in E} \min(y_2^*(e), x(e)) \tag{18.63}$$

- Hence, $\exists e'$ s.t. $y_1^*(e') < \min(y_2^*(e'), x(e'))$. Recall $y_1^*, y_2^* \in P$.
- This implies $\sum_{e \neq e'} y_1^*(e) + y_1^*(e') < \sum_{e \neq e'} y_1^*(e) + \min(y_2^*(e'), x(e'))$, better feasible solution to l.h.s., contradicting y_1^*'s optimality for l.h.s.

Alternate proof of modified max-min theorem

Consider the following two problems:

$$\max \sum_{e \in E} y(e) \tag{18.61a}$$
$$\text{s.t. } y \le x \tag{18.61b}$$
$$y \in P \tag{18.61c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{18.62a}$$
$$\text{s.t. } y \in P \tag{18.62b}$$

- Solutions identical cost. Let y_1^* be l.h.s. OPT and y_2^* be r.h.s. OPT.
- Similarly, consider y_2^* as l.h.s. solution, suppose worse than l.h.s. OPT

$$\sum_{e \in E} y_2^*(e) < \sum_{e \in E} y_1^*(e) \tag{18.63}$$

Alternate proof of modified max-min theorem

Consider the following two problems:

$$\max \sum_{e \in E} y(e) \tag{18.61a}$$
$$\text{s.t. } y \leq x \tag{18.61b}$$
$$y \in P \tag{18.61c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{18.62a}$$
$$\text{s.t. } y \in P \tag{18.62b}$$

- Solutions identical cost. Let y_1^* be l.h.s. OPT and y_2^* be r.h.s. OPT.
- Similarly, consider y_2^* as l.h.s. solution, suppose worse than l.h.s. OPT

$$\sum_{e \in E} y_2^*(e) < \sum_{e \in E} y_1^*(e) \tag{18.63}$$

- Then $\exists e'$ such that $y_2^*(e') < y_1^*(e') \leq x(e')$.

Alternate proof of modified max-min theorem

Consider the following two problems:

$$\max \sum_{e \in E} y(e) \tag{18.61a}$$
$$\text{s.t. } y \leq x \tag{18.61b}$$
$$y \in P \tag{18.61c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{18.62a}$$
$$\text{s.t. } y \in P \tag{18.62b}$$

- Solutions identical cost. Let y_1^* be l.h.s. OPT and y_2^* be r.h.s. OPT.
- Similarly, consider y_2^* as l.h.s. solution, suppose worse than l.h.s. OPT

$$\sum_{e \in E} y_2^*(e) < \sum_{e \in E} y_1^*(e) \tag{18.63}$$

- Then $\exists e'$ such that $y_2^*(e') < y_1^*(e') \leq x(e')$.
- This implies that replacing $y_2^*(e')$'s value with $y_1^*(e')$ is still feasible for r.h.s. but better, contradicting y_2^*'s optimality.

Alternate proof of modified max-min theorem

Consider the following two problems:

$$\max \sum_{e \in E} y(e) \tag{18.61a}$$
$$\text{s.t. } y \leq x \tag{18.61b}$$
$$y \in P \tag{18.61c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{18.62a}$$
$$\text{s.t. } y \in P \tag{18.62b}$$

- Solutions identical cost. Let y_1^* be l.h.s. OPT and y_2^* be r.h.s. OPT.
- Similarly, consider y_2^* as l.h.s. solution, suppose worse than l.h.s. OPT

$$\sum_{e \in E} y_2^*(e) < \sum_{e \in E} y_1^*(e) \tag{18.63}$$

- Then $\exists e'$ such that $y_2^*(e') < y_1^*(e') \leq x(e')$.
- This implies that replacing $y_2^*(e')$'s value with $y_1^*(e')$ is still feasible for r.h.s. but better, contradicting y_2^*'s optimality.
- Hence, from previous slide, taking $x = 0$:

$$\max \left(y^-(E) : y \in B_f\right) = \max \left(y(E) : y \leq 0, y \in P_f\right) \tag{18.64}$$

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

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

$$\max \{w^\intercal x | x \in P_f\} = \max \{w^\intercal x | x \in B_f\} \tag{18.65}$$

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

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

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

$$\max \{w^\intercal x | x \in P_f\} = \max \{w^\intercal x | x \in B_f\} \tag{18.65}$$

  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:

$$\max \{w^\intercal x | x \in B_f\} \tag{18.66}$$

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

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

$$\max\{w^\intercal x | x \in P_f\} = \max\{w^\intercal x | x \in B_f\} \tag{18.65}$$

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:

$$\max\{w^\intercal x | x \in B_f\} \tag{18.66}$$

- Also, since $w \in \mathbb{R}^E$ is arbitrary, and since

$$\min\{w^\intercal x | x \in B_f\} = -\max\{-w^\intercal x | x \in B_f\} \tag{18.67}$$

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

$$w(e_1) \leq w(e_2) \leq \cdots \leq w(e_m) \tag{18.68}$$

will solve l.h.s. of Equation (18.67).

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

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

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

So y^* uses greedy algorithm with positive orthant $B_{f'}$. To show, we use Theorem 11.4.1 in Lecture 11, but we don't require $y \ge 0$, 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.

Min-Norm Point and SFM

Theorem 18.7.1

Let y^*, A_-, and A_0 be as given. Then y^* is a maximizer of the l.h.s. of Eqn. (18.48). 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 $\text{dep}(x^*, e)$.

...

Min-Norm Point and SFM

Theorem 18.7.1

Let y^*, A_-, and A_0 be as given. Then y^* is a maximizer of the l.h.s. of Eqn. (18.48). 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 $\mathrm{sat}(x^*) = E$. Thus, we can consider any $e \in E$ within $\mathrm{dep}(x^*, e)$.
- Consider any pair (e, e') with $e' \in \mathrm{dep}(x^*, e)$ and $e \in A_-$. Then $x^*(e) < 0$, and $\exists \alpha > 0$ s.t. $x^* + \alpha \mathbf{1}_e - \alpha \mathbf{1}_{e'} \in P_f$.

...

Min-Norm Point and SFM

Theorem 18.7.1

Let y^*, A_-, and A_0 be as given. Then y^* is a maximizer of the l.h.s. of Eqn. (18.48). 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

$$(x^* + \alpha \mathbf{1}_e - \alpha \mathbf{1}_{e'})(E) = x^*(E) + \alpha - \alpha = f(E) \tag{18.69}$$

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

...

Min-Norm Point and SFM

...proof of Thm. 18.7.1 cont.

- Then $(x^* + \alpha \mathbf{1}_e - \alpha \mathbf{1}_{e'})(E)$
$= x^*(E \setminus \{e, e'\}) + \underbrace{(x^*(e) + \alpha)}_{x^*_{\text{new}}(e)} + \underbrace{(x^*(e') - \alpha)}_{x^*_{\text{new}}(e')} = f(E)$.

...

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 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 $\alpha > 0$,
$\Big(x^*(e)\Big)^2 + \Big(x^*(e')\Big)^2 < \Big(x^*_{\text{new}}(e)\Big)^2 + \Big(x^*_{\text{new}}(e')\Big)^2$

. . .

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 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 $\alpha > 0$,
$\Big(x^*(e)\Big)^2 + \Big(x^*(e')\Big)^2 < \Big(x^*_{\text{new}}(e)\Big)^2 + \Big(x^*_{\text{new}}(e')\Big)^2$
- 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^*.

. . .

Min-Norm Point and SFM

…proof of Thm. 18.7.1 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 $\alpha > 0$,
$\Big(x^*(e)\Big)^2 + \Big(x^*(e')\Big)^2 < \Big(x^*_{\text{new}}(e)\Big)^2 + \Big(x^*_{\text{new}}(e')\Big)^2$
- 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^*.

…

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 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 $\alpha > 0$,
$\Big(x^*(e)\Big)^2 + \Big(x^*(e')\Big)^2 < \Big(x^*_{\text{new}}(e)\Big)^2 + \Big(x^*_{\text{new}}(e')\Big)^2$
- 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. 18.7.1 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 $e' \in A_-$.

. . .

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 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 $\text{dep}(x^*, e) \subseteq A_-$.

. . .

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 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. 18.7.1 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 $\text{dep}(x^*, e) \subseteq A_0$.
- Also, recall that $e \in \text{dep}(x^*, e)$.

. . .

Min-Norm Point and SFM

... proof of Thm. 18.7.1 cont.

- Therefore, we have $\cup_{e \in A_-} \operatorname{dep}(x^*, e) = A_-$ and $\cup_{e \in A_0} \operatorname{dep}(x^*, e) = A_0$

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 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.

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 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:

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 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:

$$x^*(A_-) = f(A_-) \tag{18.70}$$

Min-Norm Point and SFM

... proof of Thm. 18.7.1 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:

$$x^*(A_-) = f(A_-) \tag{18.70}$$

$$x^*(A_0) = f(A_0) \tag{18.71}$$

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 cont.

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

$$x^*(A_-) = f(A_-) \tag{18.70}$$

$$x^*(A_0) = f(A_0) \tag{18.71}$$

$$x^*(A_-) = x^*(A_0) = y^*(E) = y^*(A_0) + \underbrace{y^*(E \setminus A_0)}_{=0} \tag{18.72}$$

Min-Norm Point and SFM

… proof of Thm. 18.7.1 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:

$$x^*(A_-) = f(A_-) \tag{18.70}$$

$$x^*(A_0) = f(A_0) \tag{18.71}$$

$$x^*(A_-) = x^*(A_0) = y^*(E) = y^*(A_0) + \underbrace{y^*(E \setminus A_0)}_{=0} \tag{18.72}$$

and therefore, all together we have

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 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:

$$x^*(A_-) = f(A_-) \tag{18.70}$$

$$x^*(A_0) = f(A_0) \tag{18.71}$$

$$x^*(A_-) = x^*(A_0) = y^*(E) = y^*(A_0) + \underbrace{y^*(E \setminus A_0)}_{=0} \tag{18.72}$$

and therefore, all together we have

$$f(A_-) = f(A_0) = x^*(A_-) = x^*(A_0) = y^*(E) \tag{18.73}$$

Min-Norm Point and SFM

... proof of Thm. 18.7.1 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:

$$x^*(A_-) = f(A_-) \tag{18.70}$$

$$x^*(A_0) = f(A_0) \tag{18.71}$$

$$x^*(A_-) = x^*(A_0) = y^*(E) = y^*(A_0) + \underbrace{y^*(E \setminus A_0)}_{=0} \tag{18.72}$$

and therefore, all together we have

$$f(A_-) = f(A_0) = x^*(A_-) = x^*(A_0) = y^*(E) \tag{18.73}$$

- Hence, $f(A_-) = f(A_0)$, meaning A_- and A_0 have the same valuation, but we have not yet shown they are the minimizers of the submodular function, nor that they are, resp. the maximal and minimal minimizers.

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (18.48) (recall, which is $\max\{y(E)|y \in P_f, y \leq 0\} = \min\{f(X)|X \subseteq V\}$).

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (18.48) (recall, which is $\max\{y(E)|y \in P_f, y \leq 0\} = \min\{f(X)|X \subseteq V\}$). 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 $y^* \in P_f$.

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (18.48) (recall, which is $\max\{y(E) | y \in P_f, y \leq 0\} = \min\{f(X) | X \subseteq V\}$). 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 $y(E) \leq y(X) \leq f(X)$.

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (18.48) (recall, which is $\max\{y(E)|y \in P_f, y \leq 0\} = \min\{f(X)|X \subseteq V\}$). 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. (18.48), $y^* \leq 0$, $y^* \in P_f$, so $y^*(E) \leq f(X)$ for all X.

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (18.48) (recall, which is $\max\{y(E) | y \in P_f, y \leq 0\} = \min\{f(X) | X \subseteq V\}$). 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. (18.48), $y^* \leq 0$, $y^* \in P_f$, so $y^*(E) \leq f(X)$ for all X.
- So $y^*(E) \leq \min\{f(X) | X \subseteq V\}$.

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (18.48) (recall, which is $\max\{y(E)|y \in P_f, y \leq 0\} = \min\{f(X)|X \subseteq V\}$). 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. (18.48), $y^* \leq 0$, $y^* \in P_f$, so $y^*(E) \leq f(X)$ for all X.
- So $y^*(E) \leq \min\{f(X)|X \subseteq V\}$.
- Considering Eqn. (18.70), we have found sets A_- and A_0 with tightness in Eqn. (18.48), meaning $y^*(E) = f(A_-) = f(A_0)$.

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (18.48) (recall, which is $\max\{y(E) | y \in P_f, y \leq 0\} = \min\{f(X) | X \subseteq V\}$). 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. (18.48), $y^* \leq 0$, $y^* \in P_f$, so $y^*(E) \leq f(X)$ for all X.
- So $y^*(E) \leq \min\{f(X) | X \subseteq V\}$.
- Considering Eqn. (18.70), we have found sets A_- and A_0 with tightness in Eqn. (18.48), meaning $y^*(E) = f(A_-) = f(A_0)$.
- Hence, y^* is a maximizer of l.h.s. of Eqn. (18.48), and A_- and A_0 are minimizers of f.

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 cont.

- We next show that, not only are they minimizers, but A_- is the unique minimal and A_0 is the unique maximal minimizer of f

$\square$

Min-Norm Point and SFM

... proof of Thm. 18.7.1 cont.

- We next show that, not only are they minimizers, but A_- is the unique minimal and A_0 is the unique maximal minimizer of f
- Now, for any $X \subset A_-$, we have

$$f(X) \geq x^*(X) > x^*(A_-) = f(A_-) \tag{18.74}$$

□

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 cont.

- We next show that, not only are they minimizers, but A_- is the unique minimal and A_0 is the unique maximal minimizer of f
- Now, for any $X \subset A_-$, we have

$$f(X) \geq x^*(X) > x^*(A_-) = f(A_-) \tag{18.74}$$

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

$$f(X) \geq x^*(X) > x^*(A_0) = f(A_0) \tag{18.75}$$

□

Min-Norm Point and SFM

. . . proof of Thm. 18.7.1 cont.

- We next show that, not only are they minimizers, but A_- is the unique minimal and A_0 is the unique maximal minimizer of f
- Now, for any $X \subset A_-$, we have

$$f(X) \geq x^*(X) > x^*(A_-) = f(A_-) \tag{18.74}$$

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

$$f(X) \geq x^*(X) > x^*(A_0) = f(A_0) \tag{18.75}$$

- 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.

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.

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).

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).
- As we saw last time, the 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.

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).
- As we saw last time, the 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.

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).
- As we saw last time, the 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 recall, its underlying lower-bound complexity is unknown.

Min-norm point and other minimizers of f

- Recall, that the set of minimizers of f forms a lattice.

Min-norm point and other minimizers of f

- Recall, that the set of minimizers of f forms a lattice.
- Q: If we take any A with $A_- \subset A \subset A_0$, is A also a minimizer?

Min-norm point and other minimizers of f

- Recall, that the set of minimizers of f forms a lattice.
- Q: If we take any A with $A_- \subset A \subset A_0$, is A also a minimizer?
- In fact, with x^* the min-norm point, and A_- and A_0 as defined above, we have the following theorem:

Min-norm point and other minimizers of f

- Recall, that the set of minimizers of f forms a lattice.
- Q: If we take any A with $A_- \subset A \subset A_0$, is A also a minimizer?
- In fact, with x^* the min-norm point, and A_- and A_0 as defined above, we have the following theorem:

Theorem 18.7.2

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

$$A = A_- \cup \bigcup_{a \in A_m} \operatorname{dep}(x^*, a) \tag{18.76}$$

for some set $A_m \subseteq A_0 \setminus A_-$. Conversely, for any set $A_m \subseteq A_0 \setminus A_-$, then $A \triangleq A_- \cup \bigcup_{a \in A_m} \operatorname{dep}(x^*, a)$ is a minimizer.

Min-norm point and other minimizers of f

proof of Thm. 18.7.2.

- If A is a minimizer, then $A_- \subseteq A \subseteq A_0$, and $f(A) = y^*(E)$ is the minimum valuation of f.

...

Min-norm point and other minimizers of f

proof of Thm. 18.7.2.

- 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 $x^*(A_0 \setminus A) = 0$).

...

Min-norm point and other minimizers of f

proof of Thm. 18.7.2.

- 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^*.

...

Min-norm point and other minimizers of f

proof of Thm. 18.7.2.

- 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.

...

Min-norm point and other minimizers of f

proof of Thm. 18.7.2.

- 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 $a \in A$, $\text{dep}(x^*, a) \subseteq A$.

...

Min-norm point and other minimizers of f

proof of Thm. 18.7.2.

- 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 $\mathrm{dep}(x^*, a)$ is the minimal tight containing a.
- Hence, for any $a \in A$, $\mathrm{dep}(x^*, a) \subseteq A$.
- This means that $\bigcup_{a \in A} \mathrm{dep}(x^*, a) = A$.

...

Min-norm point and other minimizers of f

proof of Thm. 18.7.2.

- 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

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

...

Min-norm point and other minimizers of f

proof of Thm. 18.7.2.

- Conversely, consider any set $A_m \subseteq A_0 \setminus A_-$, and define A as

$$A = A_- \cup \bigcup_{a \in A_m} \operatorname{dep}(x^*, a) = \bigcup_{a \in A_-} \operatorname{dep}(x^*, a) \cup \bigcup_{a \in A_m} \operatorname{dep}(x^*, a) \tag{18.77}$$

□

Therefore, we can generate the entire lattice of minimizers of f starting from A_- and A_0 given access to $\operatorname{dep}(x^*, e)$.

Min-norm point and other minimizers of f

proof of Thm. 18.7.2.

- Conversely, consider any set $A_m \subseteq A_0 \setminus A_-$, and define A as

$$A = A_- \cup \bigcup_{a \in A_m} \text{dep}(x^*, a) = \bigcup_{a \in A_-} \text{dep}(x^*, a) \cup \bigcup_{a \in A_m} \text{dep}(x^*, a) \tag{18.77}$$

- Then since A is a union of tight sets, A is also a tight set, and we have $f(A) = x^*(A)$.

□

Therefore, we can generate the entire lattice of minimizers of f starting from A_- and A_0 given access to $\text{dep}(x^*, e)$.

Min-norm point and other minimizers of f

proof of Thm. 18.7.2.

- Conversely, consider any set $A_m \subseteq A_0 \setminus A_-$, and define A as

$$A = A_- \cup \bigcup_{a \in A_m} \mathrm{dep}(x^*, a) = \bigcup_{a \in A_-} \mathrm{dep}(x^*, a) \cup \bigcup_{a \in A_m} \mathrm{dep}(x^*, a) \tag{18.77}$$

- Then since A is a union of tight sets, A is also a tight set, and we have $f(A) = x^*(A)$.
- But $x^*(A \setminus A_-) = 0$, so $f(A) = x^*(A) = x^*(A_-) = f(A_-)$ meaning A is also a minimizer of f.

□

Therefore, we can generate the entire lattice of minimizers of f starting from A_- and A_0 given access to $\mathrm{dep}(x^*, e)$.

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 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 $A \subseteq E \setminus \{e\}$.

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 $e \in A_0$.

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):

$$\underset{w \in \mathbb{R}^V}{\text{minimize}} \; \tilde{f}(w) + \frac{1}{2}\|w\|_2^2 \qquad (18.78)$$

$$\begin{aligned} &\text{maximize} && -\|x\|_2^2 && (18.79a) \\ &\text{subject to} && x \in B_f && (18.79b) \end{aligned}$$

where $B_f = P_f \cap \left\{x \in \mathbb{R}^V : x(V) = f(V)\right\}$ is the base polytope of submodular function f, and $\|x\|_2^2 = \sum_{e \in V} x(e)^2$ is squared 2-norm.

- Equation (18.78) is related to proximal methods to minimize the Lovász extension (see Parikh&Boyd, "Proximal Algorithms" 2013).
- Equation (18.79b) 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).

Convex and affine hulls, affinely independent

- Given points set $P = \{p_1, p_2, \ldots, p_k\}$ with $p_i \in \mathbb{R}^V$, let $\text{conv}\, P$ be the convex hull of P, i.e.,

$$\text{conv}\, P \triangleq \left\{ \sum_{i=1}^{k} \lambda_i p_i : \sum_i \lambda_i = 1,\ \lambda_i \geq 0, i \in [k] \right\}. \tag{18.80}$$

Convex and affine hulls, affinely independent

- Given points set $P = \{p_1, p_2, \dots, p_k\}$ with $p_i \in \mathbb{R}^V$, let $\operatorname{conv} P$ be the convex hull of P, i.e.,

$$\operatorname{conv} P \triangleq \left\{ \sum_{i=1}^{k} \lambda_i p_i : \sum_i \lambda_i = 1,\ \lambda_i \geq 0, i \in [k] \right\}. \tag{18.80}$$

- For a set of points $Q = \{q_1, q_2, \dots, q_k\}$, with $q_i \in \mathbb{R}^V$, we define $\operatorname{aff} Q$ to be the affine hull of Q, i.e.:

$$\operatorname{aff} Q \triangleq \left\{ \sum_{i \in 1}^{k} \lambda_i q_i : \sum_{i=1}^{k} \lambda_i = 1 \right\} \tag{18.81}$$

Convex and affine hulls, affinely independent

- Given points set $P = \{p_1, p_2, \dots, p_k\}$ with $p_i \in \mathbb{R}^V$, let $\operatorname{conv} P$ be the convex hull of P, i.e.,

$$\operatorname{conv} P \triangleq \left\{ \sum_{i=1}^{k} \lambda_i p_i : \sum_i \lambda_i = 1,\ \lambda_i \geq 0, i \in [k] \right\}. \tag{18.80}$$

- For a set of points $Q = \{q_1, q_2, \dots, q_k\}$, with $q_i \in \mathbb{R}^V$, we define $\operatorname{aff} Q$ to be the affine hull of Q, i.e.:

$$\operatorname{aff} Q \triangleq \left\{ \sum_{i \in 1}^{k} \lambda_i q_i : \sum_{i=1}^{k} \lambda_i = 1 \right\} \supseteq \operatorname{conv} Q. \tag{18.81}$$

Convex and affine hulls, affinely independent

- Given points set $P = \{p_1, p_2, \dots, p_k\}$ with $p_i \in \mathbb{R}^V$, let $\operatorname{conv} P$ be the convex hull of P, i.e.,

$$\operatorname{conv} P \triangleq \left\{ \sum_{i=1}^{k} \lambda_i p_i : \sum_i \lambda_i = 1,\ \lambda_i \geq 0, i \in [k] \right\}. \tag{18.80}$$

- For a set of points $Q = \{q_1, q_2, \dots, q_k\}$, with $q_i \in \mathbb{R}^V$, we define $\operatorname{aff} Q$ to be the affine hull of Q, i.e.:

$$\operatorname{aff} Q \triangleq \left\{ \sum_{i \in 1}^{k} \lambda_i q_i : \sum_{i=1}^{k} \lambda_i = 1 \right\} \supseteq \operatorname{conv} Q. \tag{18.81}$$

- A set of points Q is affinely independent if no point in Q belows to the affine hull of the remaining points.

$H(x)$: Orthogonal x-containing hyperplane

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

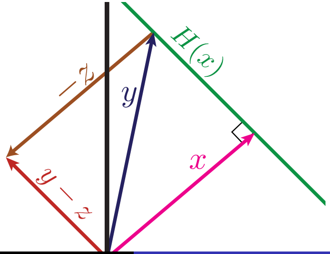
$$H(x) \triangleq \left\{ y \in \mathbb{R}^V \mid x^\intercal y = \|x\|_2^2 \right\} \tag{18.82}$$

$H(x)$: Orthogonal x-containing hyperplane

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

$$H(x) \triangleq \left\{ y \in \mathbb{R}^V \mid x^\intercal y = \|x\|_2^2 \right\} \tag{18.82}$$

- Any set $\{y \in \mathbb{R}^V | x^\intercal y = c\}$ is orthogonal to the line from 0 to x. This follows since, for constant z, $\{y : (y - z)^\intercal x = 0\} = \{y : y^\intercal x = z^\intercal x\}$ is hyperplane orthogonal to x translated by z. Take $c = z^\intercal x$ for result, and $z = x$, giving $c = \|x\|^2$, to contain x.

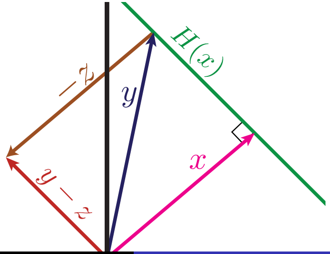


$H(x)$: Orthogonal x-containing hyperplane

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

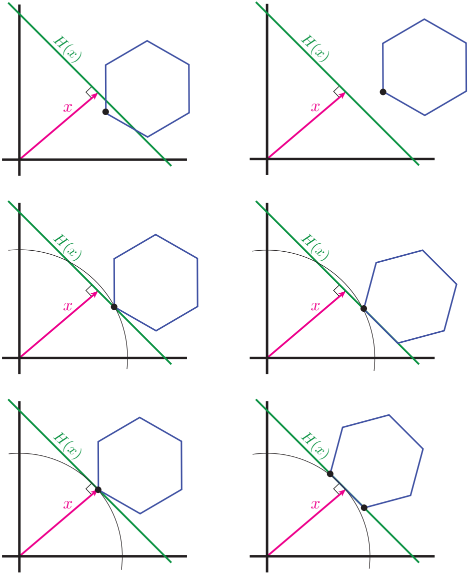
$$H(x) \triangleq \left\{ y \in \mathbb{R}^V \mid x^\intercal y = \|x\|_2^2 \right\} \tag{18.82}$$

- Any set $\{y \in \mathbb{R}^V | x^\intercal y = c\}$ is orthogonal to the line from 0 to x. This follows since, for constant z, $\{y : (y - z)^\intercal x = 0\} = \{y : y^\intercal x = z^\intercal x\}$ is hyperplane orthogonal to x translated by z. Take $c = z^\intercal x$ for result, and $z = x$, giving $c = \|x\|^2$, to contain x.



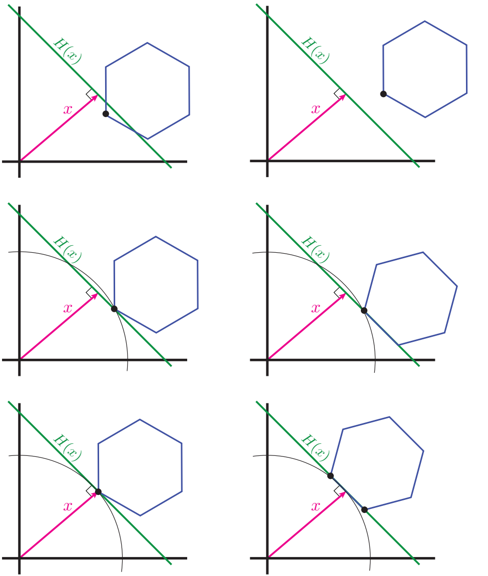
Ex: $H(x)$, polytopes, and supporting hyperplanes

- $H(x) = \left\{ y \in \mathbb{R}^V \,|\, x^\intercal y = \|x\|_2^2 \right\}$, any $z \in H(x)$ has $x^\intercal z = x^\intercal x$.



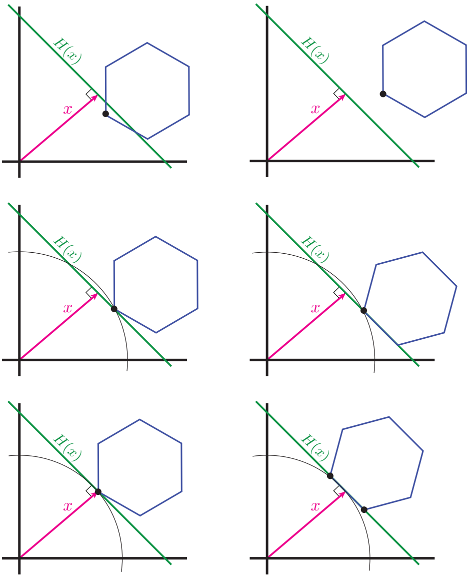
Ex: $H(x)$, polytopes, and supporting hyperplanes

- $H(x) = \left\{ y \in \mathbb{R}^V | x^\intercal y = \|x\|_2^2 \right\}$, any $z \in H(x)$ has $x^\intercal z = x^\intercal x$.
- Consider $\operatorname{conv} P$ polytope for points $P = \{p_1, p_2, \ldots\}$, and $\hat{p} \in \operatorname{argmin}_{p \in P} x^\intercal p$. TL: $x^\intercal p < x^\intercal x$; TR: $x^\intercal p > x^\intercal x$; middle row: $x^\intercal p = x^\intercal x$.



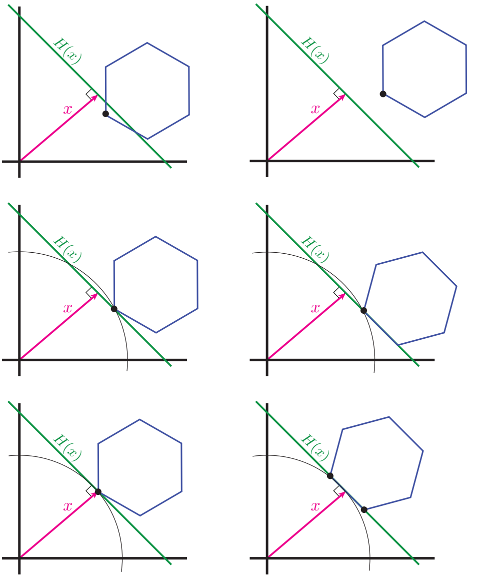
Ex: $H(x)$, polytopes, and supporting hyperplanes

- $H(x) = \left\{ y \in \mathbb{R}^V | x^\intercal y = \|x\|_2^2 \right\}$, any $z \in H(x)$ has $x^\intercal z = x^\intercal x$.
- Consider $\operatorname{conv} P$ polytope for points $P = \{p_1, p_2, \ldots\}$, and $\hat{p} \in \operatorname{argmin}_{p \in P} x^\intercal p$. TL: $x^\intercal p < x^\intercal x$; TR: $x^\intercal p > x^\intercal x$; middle row: $x^\intercal p = x^\intercal x$.
- Bottom Row: In Algo, x is chosen so that if $x^\intercal \hat{p} = x^\intercal x$ then $H(x)$ separates P from the origin, and x is the min 2-norm point. Notice that $x^\intercal p \geq x^\intercal x$ for all $p \in P$.



Ex: $H(x)$, polytopes, and supporting hyperplanes

- $H(x) = \left\{ y \in \mathbb{R}^V | x^\intercal y = \|x\|_2^2 \right\}$, any $z \in H(x)$ has $x^\intercal z = x^\intercal x$.
- Consider $\operatorname{conv} P$ polytope for points $P = \{p_1, p_2, \ldots\}$, and $\hat{p} \in \operatorname{argmin}_{p \in P} x^\intercal p$. TL: $x^\intercal p < x^\intercal x$; TR: $x^\intercal p > x^\intercal x$; middle row: $x^\intercal p = x^\intercal x$.
- Bottom Row: In Algo, x is chosen so that if $x^\intercal \hat{p} = x^\intercal x$ then $H(x)$ separates P from the origin, and x is the min 2-norm point. Notice that $x^\intercal p \geq x^\intercal x$ for all $p \in P$.
- Middle/bottom row: $H(x)$ is a supporting hyperplane of $\operatorname{conv} P$ (contained, touching).



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, $[x, y] = \text{conv}\,\{x, y\}$.

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 ("Finding the Nearest Point in a Polytope") developed an algorithm to compute the minimum norm point of a polytope, specified as a set of vertices.

Fujishige-Wolfe Min-Norm Algorithm

- Wolfe-1976 ("Finding the Nearest Point in a Polytope") 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.

Fujishige-Wolfe Min-Norm Algorithm

- Wolfe-1976 ("Finding the Nearest Point in a Polytope") 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.

Fujishige-Wolfe Min-Norm Algorithm

- Wolfe-1976 ("Finding the Nearest Point in a Polytope") 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:

$$\min_{x \in \text{conv } P} \|x\|_2 \tag{18.83}$$

Fujishige-Wolfe Min-Norm Algorithm

- Wolfe-1976 ("Finding the Nearest Point in a Polytope") 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:

$$\min_{x \in \text{conv } P} \|x\|_2 \tag{18.83}$$

- Wolfe's algorithm is guaranteed terminating, and explicitly uses a representation of x as a convex combination of points in P

Fujishige-Wolfe Min-Norm Algorithm

- Wolfe-1976 ("Finding the Nearest Point in a Polytope") 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:

$$\min_{x \in \text{conv } P} \|x\|_2 \tag{18.83}$$

- 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).

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 $\min_{x \in \operatorname{aff} Q} \|x\|_2$ is available (see below).
- Algorithm repeatedly produces min. norm point x^* for selected set Q.

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 $\min_{x \in \operatorname{aff} Q} \|x\|_2$ 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 $\operatorname{conv} Q$ and also a minimum norm point over $\operatorname{conv} Q$.

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.

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 $\min_{x \in \operatorname{aff} Q} \|x\|_2$ 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 $\operatorname{conv} Q$ and also a minimum norm point over $\operatorname{conv} Q$.
- If $Q \subseteq P$ is suitably chosen, x^* may even be the minimum norm point over $\operatorname{conv} P$ 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.

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 $\min_{x \in \operatorname{aff} Q} \|x\|_2$ 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 $\operatorname{conv} Q$ and also a minimum norm point over $\operatorname{conv} Q$.
- If $Q \subseteq P$ is suitably chosen, x^* may even be the minimum norm point over $\operatorname{conv} P$ 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.

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 $\operatorname{conv} P$.

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

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.

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:

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{18.84}$$

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.

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:

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{18.84}$$

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

$$\min_{x \in \operatorname{aff} Q} \|x\|_2 \leq \min_{x \in \operatorname{conv} Q} \|x\|_2 \leq \|x^*\|_2 \tag{18.85}$$

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:

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{18.84}$$

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

$$\min_{x \in \operatorname{aff} Q} \|x\|_2 \leq \min_{x \in \operatorname{conv} Q} \|x\|_2 \leq \|x^*\|_2 \tag{18.85}$$

- Note, the input, P, consists of m points. In the case of the base polytope, $P = B_f$ could be exponential in $n = |V|$.

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:

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{18.84}$$

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

$$\min_{x \in \operatorname{aff} Q} \|x\|_2 \leq \min_{x \in \operatorname{conv} Q} \|x\|_2 \leq \|x^*\|_2 \tag{18.85}$$

- 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.

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:

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{18.84}$$

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.

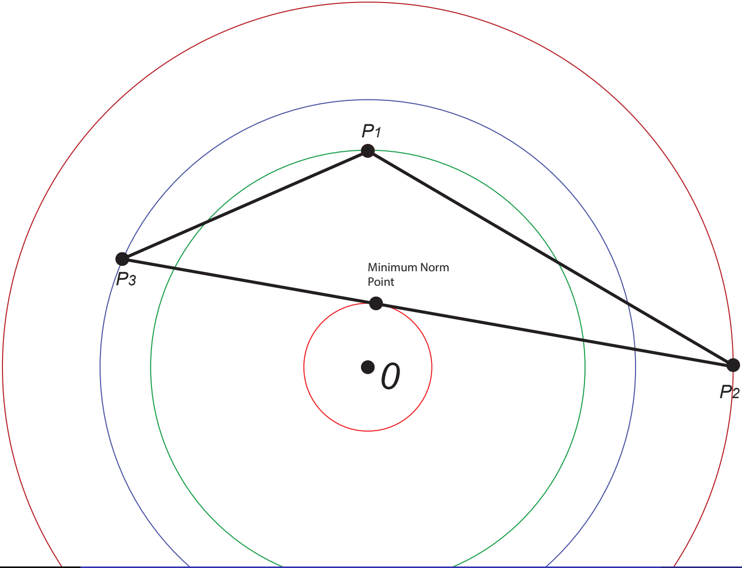
- Note also for any $x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P$, we have

$$\min_{x \in \operatorname{aff} Q} \|x\|_2 \leq \min_{x \in \operatorname{conv} Q} \|x\|_2 \leq \|x^*\|_2 \tag{18.85}$$

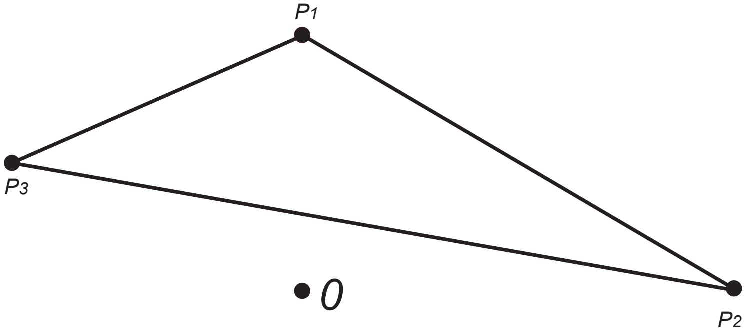
- Note, the input, P, consists of m points. In the case of the base polytope, $P = B_f$ could be exponential in $n = |V|$.
- There are six places that might be seemingly tricky or expensive: Line 4, Line 6, Line 9, Line 10, Line 14, and Line 15.
- We will consider each in turn, but first we do a geometric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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example

Polytope, and circles concentric at 0.

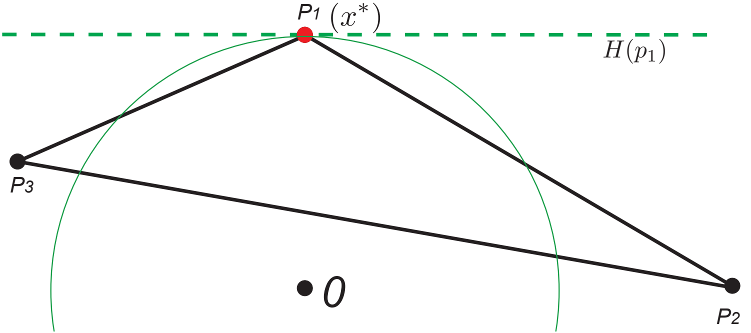


Fujishige-Wolfe Min-Norm algorithm: Geometric Example



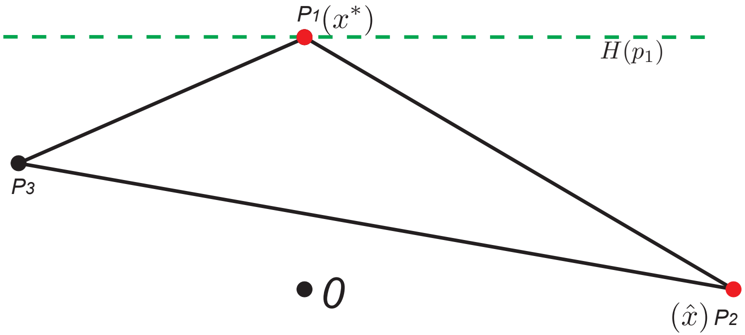
The initial polytope consisting of the convex hull of three points p_1, p_2, p_3, and the origin 0.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



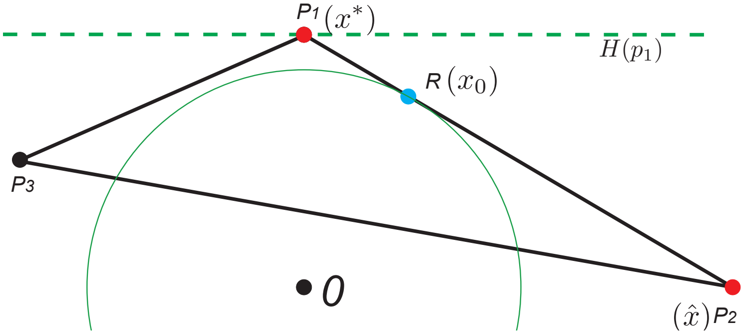
p_1 is the extreme point closest to 0 and so we choose it first, although we can choose any arbitrary extreme point as the initial point. We set $x^* \leftarrow p_1$ in Line 1, and $Q \leftarrow \{p_1\}$ in Line 2. $H(x^*) = H(p_1)$ (green dashed line) is not a supporting hyperplane of $\mathrm{conv}(P)$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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



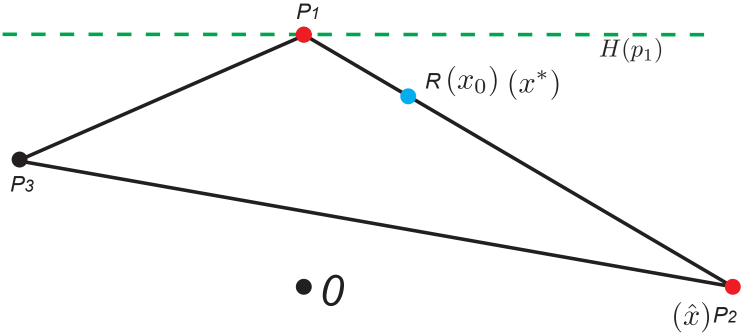
We need to add some extreme point $\hat{x}$ on the "near" side of $H(p_1)$ in Line 6, we choose $\hat{x} = p_2$. In Line 7, we set $Q \leftarrow Q \cup \{p_2\}$, so $Q = \{p_1, p_2\}$.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



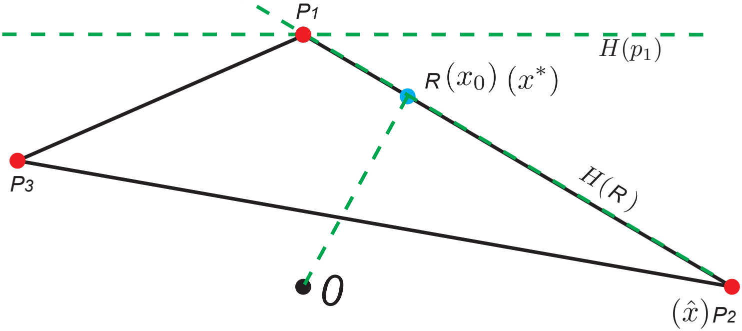
$x_0 = R$ is the min-norm point in $\text{aff}\,\{p_1, p_2\}$ computed in Line 9.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



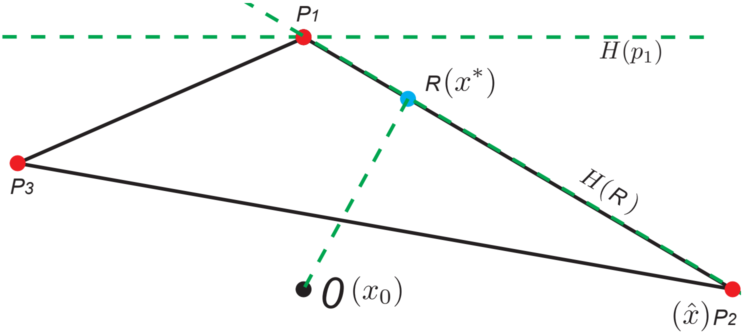
$x_0 = R$ is the min-norm point in $\text{aff}\{p_1, p_2\}$ computed in Line 9. Also, with $Q = \{p_1, p_2\}$, since $R \in \text{conv}\, Q$, we set $x^* \leftarrow x_0 = R$ in Line 11, not violating the invariant $x^* \in \text{conv}\, Q$. Note, after Line 11, we still have $x^* \in \text{conv}\, P$ and $\|x^*\|_2 = \|x^*_{\text{new}}\|_2 < \left\|x^*_{\text{old}}\right\|_2$ strictly.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



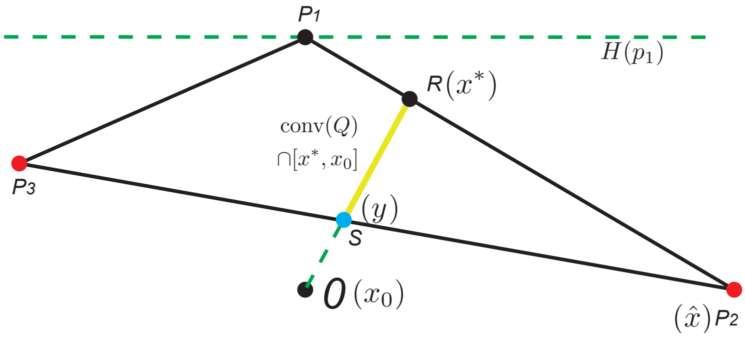
$R = x_0 = x^*$. We consider next $H(R) = H(x^*)$ in Line 4. $H(x^*)$ is not a supporting hyperplane of $\operatorname{conv} P$. So we choose p_3 on the "near" side of $H(x^*)$ in Line 6. Add $Q \leftarrow Q \cup \{p_3\}$ in Line 7. Now $Q = P = \{p_1, p_2, p_3\}$.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



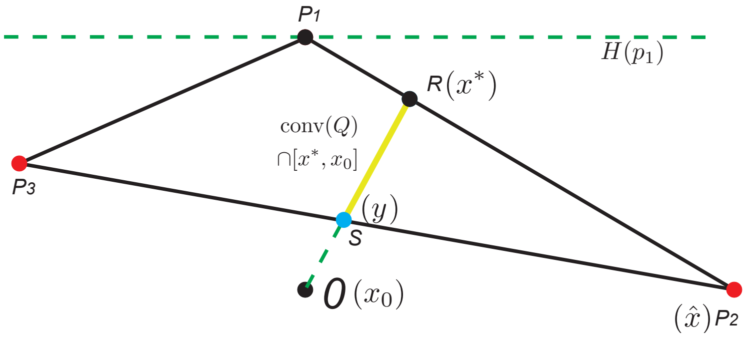
$R = x_0 = x^*$. We consider next $H(R) = H(x^*)$ in Line 4. $H(x^*)$ is not a supporting hyperplane of $\operatorname{conv} P$. So we choose p_3 on the "near" side of $H(x^*)$ in Line 6. Add $Q \leftarrow Q \cup \{p_3\}$ in Line 7. Now $Q = P = \{p_1, p_2, p_3\}$. The origin $x_0 = 0$ is the min-norm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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



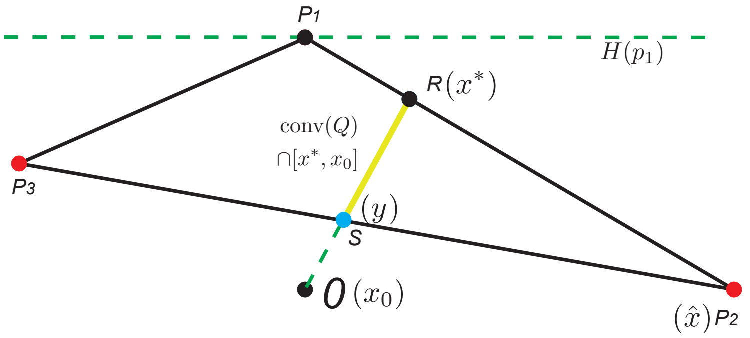
$Q = P = \{p_1, p_2, p_3\}$. Line 14: $S = y = \mathrm{argmin}_{x \in \mathrm{conv}\, Q \cap [x^*, x_0]} \|x - x_0\|_2$ where x_0 is 0 and x^* is R here. Thus, y lies on the boundary of $\mathrm{conv}\, Q$. Note, $\|y\|_2 < \|x^*\|_2$ since $x^* \in \mathrm{conv}\, Q$, $\|x_0\|_2 < \|x^*\|_2$.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



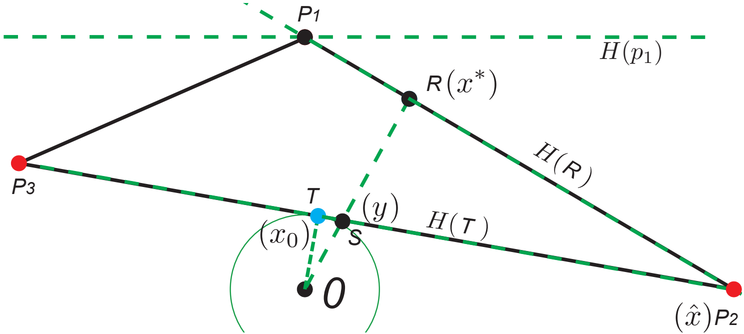
$Q = P = \{p_1, p_2, p_3\}$. Line 14: $S = y = \operatorname{argmin}_{x \in \mathrm{conv}\, Q \cap [x^*, x_0]} \|x - x_0\|_2$ where x_0 is 0 and x^* is R here. Thus, y lies on the boundary of $\mathrm{conv}\, Q$. Note, $\|y\|_2 < \|x^*\|_2$ since $x^* \in \mathrm{conv}\, Q$, $\|x_0\|_2 < \|x^*\|_2$. Line 15: Delete p_1 from Q since not on face where $y = S$ lies. $Q = \{p_2, p_3\}$ after Line 15. We still have $y = S \in \mathrm{conv}\, Q$ for the updated Q.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



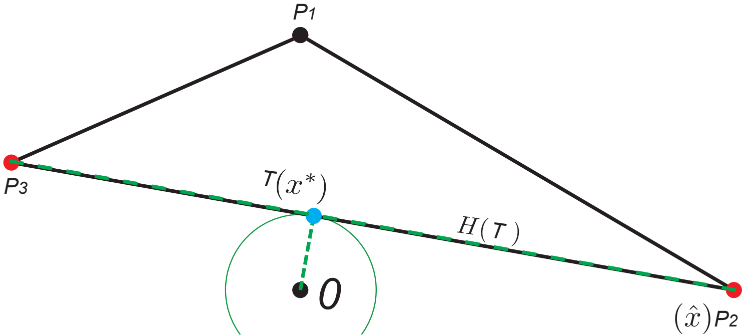
$Q = P = \{p_1, p_2, p_3\}$. Line 14: $S = y = \mathrm{argmin}_{x \in \mathrm{conv}\, Q \cap [x^*, x_0]} \|x - x_0\|_2$ where x_0 is 0 and x^* is R here. Thus, y lies on the boundary of $\mathrm{conv}\, Q$. Note, $\|y\|_2 < \|x^*\|_2$ since $x^* \in \mathrm{conv}\, Q$, $\|x_0\|_2 < \|x^*\|_2$. Line 15: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

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 $\operatorname{aff} Q$. We also have $x_0 \in \operatorname{conv} Q$ 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 18.8.1

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

$$p_i^{\intercal} x^* \geq \|x^*\|_2^2 \quad \forall i = 1, \cdots, m. \tag{18.86}$$

Proof.

- Assume x^* is the min-norm point, let $y \in \text{conv}\, P$, and $0 \leq \theta \leq 1$.

Condition for Min-Norm Point

Theorem 18.8.1

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

$$p_i^\intercal x^* \geq \|x^*\|_2^2 \quad \forall i = 1, \cdots, m. \tag{18.86}$$

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

$$\|z\|_2^2 = \|x^* + \theta(y - x^*)\|_2^2 \tag{18.87}$$

$$= \|x^*\|_2^2 + 2\theta({x^*}^\intercal y - {x^*}^\intercal x^*) + \theta^2 \|y - x^*\|_2^2 \tag{18.88}$$

Condition for Min-Norm Point

Theorem 18.8.1

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

$$p_i^\intercal x^* \geq \|x^*\|_2^2 \quad \forall i = 1, \cdots, m. \tag{18.86}$$

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

$$\|z\|_2^2 = \|x^* + \theta(y - x^*)\|_2^2 \tag{18.87}$$

$$= \|x^*\|_2^2 + 2\theta(x^{*\intercal} y - x^{*\intercal} x^*) + \theta^2 \|y - x^*\|_2^2 \tag{18.88}$$

- 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 (18.86).

Condition for Min-Norm Point

Theorem 18.8.1

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

$$p_i^\intercal x^* \geq \|x^*\|_2^2 \quad \forall i = 1, \cdots, m. \tag{18.86}$$

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

$$\|z\|_2^2 = \|x^* + \theta(y - x^*)\|_2^2 \tag{18.87}$$

$$= \|x^*\|_2^2 + 2\theta(x^{*\intercal} y - x^{*\intercal} x^*) + \theta^2 \|y - x^*\|_2^2 \tag{18.88}$$

- 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 (18.86).
- Conversely, given Eq (18.86), and given that $y = \sum_i \lambda_i p_i \in \operatorname{conv} P$,

$$y^\intercal x^* = \sum_i \lambda_i p_i^\intercal x^* \geq \sum_i \lambda_i x^{*\intercal} x^* = x^{*\intercal} x^* \tag{18.89}$$

implying that $\|z\|_2^2 > \|x^*\|_2^2$ in Equation 18.88 for arbitrary $z \in \operatorname{conv} P$. □

The set Q is always affinely independent

Lemma 18.8.2

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

The set Q is always affinely independent

Lemma 18.8.2

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

Proof.

$\square$

The set Q is always affinely independent

Lemma 18.8.2

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).

□

The set Q is always affinely independent

Lemma 18.8.2

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.

□

The set Q is always affinely independent

Lemma 18.8.2

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 $\operatorname{aff} Q$ (since we break only at Line 12).

□

The set Q is always affinely independent

Lemma 18.8.2

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 $\operatorname{aff} Q$ (since we break only at Line 12).
- Therefore, x^* is normal to $\operatorname{aff} Q$, which implies $\operatorname{aff} Q \subseteq H(x^*)$.

$\square$

The set Q is always affinely independent

Lemma 18.8.2

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 $\operatorname{aff} Q$ (since we break only at Line 12).
- Therefore, x^* is normal to $\operatorname{aff} Q$, which implies $\operatorname{aff} Q \subseteq H(x^*)$.
- Since $\hat{x} \notin H(x^*)$ chosen at Line 6, we have $\hat{x} \notin \operatorname{aff} Q$.

□

The set Q is always affinely independent

Lemma 18.8.2

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 12).
- 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. $\square$

The set Q is always affinely independent

Lemma 18.8.2

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 $\operatorname{aff} Q$ (since we break only at Line 12).
- Therefore, x^* is normal to $\operatorname{aff} Q$, which implies $\operatorname{aff} Q \subseteq H(x^*)$.
- Since $\hat{x} \notin H(x^*)$ chosen at Line 6, we have $\hat{x} \notin \operatorname{aff} Q$.
- $\therefore$ update $Q \cup \{\hat{x}\}$ at Line 7 is affinely independent as long as Q is. $\square$

Thus, by Lemma 18.8.2, we have for any $x \in \operatorname{aff} Q$ such that $x = \sum_i w_i q_i$ with $\sum_i w_i = 1$, the weights w_i are uniquely determined.

The set Q is never too large

Lemma 18.8.3

The set Q in the MN Algorithm has size never more than $n+1$.

The set Q is never too large

Lemma 18.8.3

The set Q in the MN Algorithm has size never more than $n + 1$.

Proof.

This is immediate, since Q is always affinely independnet, and in $\mathbb{R}^V$, an affinely independnet set can have at most $n + 1$ entries, with $|V| = n$. □

Minimum Norm in an affine set

- Line 9 of the algorithm requires $x_0 \leftarrow \min_{x \in \text{aff } Q} \|x\|_2$.

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.

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 represent $n \times k$ matrix with points as columns $q \in Q$. The following is solvable with matrix inversion/linear solver, where $x = Qw$:

$$\text{minimize} \quad \|x\|_2^2 = w^\intercal Q^\intercal Q w \tag{18.90}$$

$$\text{subject to} \quad \mathbf{1}^\intercal w = 1 \tag{18.91}$$

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 represent $n \times k$ matrix with points as columns $q \in Q$. The following is solvable with matrix inversion/linear solver, where $x = Qw$:

$$\text{minimize} \quad \|x\|_2^2 = w^\intercal Q^\intercal Q w \tag{18.90}$$

$$\text{subject to} \quad \mathbf{1}^\intercal w = 1 \tag{18.91}$$

- Form Lagrangian $w^\intercal Q^\intercal Q w + 2\lambda(\mathbf{1}^\intercal w - 1)$, and differentiating w.r.t. λ and w, and setting to zero, we get:

$$\mathbf{1}^\intercal w = 1 \tag{18.92}$$

$$Q^\intercal Q w + \lambda \mathbf{1} = 0 \tag{18.93}$$

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 represent $n \times k$ matrix with points as columns $q \in Q$. The following is solvable with matrix inversion/linear solver, where $x = Qw$:

$$\text{minimize} \qquad \|x\|_2^2 = w^\intercal Q^\intercal Q w \tag{18.90}$$

$$\text{subject to} \qquad \mathbf{1}^\intercal w = 1 \tag{18.91}$$

- Form Lagrangian $w^\intercal Q^\intercal Q w + 2\lambda(\mathbf{1}^\intercal w - 1)$, and differentiating w.r.t. λ and w, and setting to zero, we get:

$$\mathbf{1}^\intercal w = 1 \tag{18.92}$$

$$Q^\intercal Q w + \lambda \mathbf{1} = 0 \tag{18.93}$$

- $k+1$ variables and k unknowns, solvable with linear solver with matrices

$$\begin{bmatrix} 0 & \mathbf{1}^\intercal \\ \mathbf{1} & Q^\intercal Q \end{bmatrix} \begin{bmatrix} \lambda \\ w \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \tag{18.94}$$

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 represent $n \times k$ matrix with points as columns $q \in Q$. The following is solvable with matrix inversion/linear solver, where $x = Qw$:

$$\text{minimize} \qquad \|x\|_2^2 = w^\intercal Q^\intercal Q w \tag{18.90}$$

$$\text{subject to} \qquad \mathbf{1}^\intercal w = 1 \tag{18.91}$$

- Form Lagrangian $w^\intercal Q^\intercal Q w + 2\lambda(\mathbf{1}^\intercal w - 1)$, and differentiating w.r.t. λ and w, and setting to zero, we get:

$$\mathbf{1}^\intercal w = 1 \tag{18.92}$$

$$Q^\intercal Q w + \lambda \mathbf{1} = 0 \tag{18.93}$$

- $k+1$ variables and k unknowns, solvable with linear solver with matrices

$$\begin{bmatrix} 0 & \mathbf{1}^\intercal \\ \mathbf{1} & Q^\intercal Q \end{bmatrix} \begin{bmatrix} \lambda \\ w \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \tag{18.94}$$

- Thanks to Q being affine, matrix on l.h.s. is invertable.

Minimum Norm in an affine set

- Note, this also solves Line 10, since feasibility requires $\sum_i w_i = 1$, we need only check $w \geq 0$ to ensure $x_0 = \sum_i w_i q_i \in \operatorname{conv} Q$.

Minimum Norm in an affine set

- Note, this also solves Line 10, since feasibility requires $\sum_i w_i = 1$, we need only check $w \geq 0$ to ensure $x_0 = \sum_i w_i q_i \in \operatorname{conv} Q$.
- In fact, a feature of the algorithm (in Wolfe's 1976 paper) is that we keep the convex coefficients $\{w_i\}_i$ where $x^* = \sum_i w_i p_i$ 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.

Minimum Norm in an affine set

- Note, this also solves Line 10, since feasibility requires $\sum_i w_i = 1$, we need only check $w \geq 0$ to ensure $x_0 = \sum_i w_i q_i \in \operatorname{conv} Q$.
- In fact, a feature of the algorithm (in Wolfe's 1976 paper) is that we keep the convex coefficients $\{w_i\}_i$ where $x^* = \sum_i w_i p_i$ 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).

Minimum Norm in an affine set

- Note, this also solves Line 10, since feasibility requires $\sum_i w_i = 1$, we need only check $w \geq 0$ to ensure $x_0 = \sum_i w_i q_i \in \operatorname{conv} Q$.
- In fact, a feature of the algorithm (in Wolfe's 1976 paper) is that we keep the convex coefficients $\{w_i\}_i$ where $x^* = \sum_i w_i p_i$ 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 18.8.4

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 $\operatorname{conv} Q$.

...

MN Algorithm finds the MN point in finite time.

Theorem 18.8.4

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,

...

MN Algorithm finds the MN point in finite time.

Theorem 18.8.4

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 $x_0 = \min_{x \in \operatorname{aff} Q} \|x\|_2$.

...

MN Algorithm finds the MN point in finite time.

Theorem 18.8.4

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 18.8.4 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).

. . .

MN Algorithm finds the MN point in finite time.

. . . proof of Theorem 18.8.4 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.

. . .

MN Algorithm finds the MN point in finite time.

. . . proof of Theorem 18.8.4 continued.

- Moreover, there can be no more iterations within a minor loop than the dimension of $\text{conv}\, Q$ for the initial Q given to the minor loop initially at Line 8 (dimension of $\text{conv}\, Q$ 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.

. . .

MN Algorithm finds the MN point in finite time.

. . . proof of Theorem 18.8.4 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.

. . .

MN Algorithm finds the MN point in finite time.

. . . proof of Theorem 18.8.4 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 18.8.4 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.

. . .

MN Algorithm finds the MN point in finite time.

. . . proof of Theorem 18.8.4 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 $0 \leq \theta \leq 1$. Since both $\hat{x}, x^* \in \operatorname{conv} Q$, we have $x^* + \theta(\hat{x} - x^*) \in \operatorname{conv} Q$.

. . .

MN Algorithm finds the MN point in finite time.

...proof of Theorem 18.8.4 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 $0 \leq \theta \leq 1$. Since both $\hat{x}, x^* \in \operatorname{conv} Q$, we have $x^* + \theta(\hat{x} - x^*) \in \operatorname{conv} Q$.
- Therefore, we have $\|x^* + \theta(\hat{x} - x^*)\|_2 \geq \|x_0\|_2$, which implies

$$\begin{aligned} \|x^* + \theta(\hat{x} - x^*)\|_2^2 &= \|x^*\|_2^2 + 2\theta\left((x^*)^\top \hat{x} - \|x^*\|_2^2\right) + \theta^2 \|\hat{x} - x^*\|_2^2 \\ &\geq \|x_0\|_2^2 \end{aligned} \tag{18.95}$$

and from Line 6, $\hat{x}$ is on the same side of $H(x^*)$ as the origin, i.e. $(x^*)^\top \hat{x} < \|x^*\|_2^2$, so middle term of r.h.s. of equality is negative.

...

MN Algorithm finds the MN point in finite time.

. . . proof of Theorem 18.8.4 continued.

- Therefore, for sufficiently small θ, specifically for

$$\theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2} \tag{18.96}$$

we have that $\|x^*\|_2^2 > \|x_0\|_2^2$.

□

MN Algorithm finds the MN point in finite time.

... proof of Theorem 18.8.4 continued.

- Therefore, for sufficiently small θ, specifically for

$$\theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2} \tag{18.96}$$

  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.

□

MN Algorithm finds the MN point in finite time.

... proof of Theorem 18.8.4 continued.

- Therefore, for sufficiently small θ, specifically for

$$\theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2} \tag{18.96}$$

  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.

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.

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. 18.95, reduction on norm is lower-bounded:

$$\Delta = \|x^*\|_2^2 - \|x_0\|_2^2 \geq 2\theta\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right) - \theta^2 \|\hat{x} - x^*\|_2^2 \triangleq \underline{\Delta} \tag{18.97}$$

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. 18.95, reduction on norm is lower-bounded:

$$\Delta = \|x^*\|_2^2 - \|x_0\|_2^2 \geq 2\theta \left(\|x^*\|_2^2 - (x^*)^\top \hat{x} \right) - \theta^2 \|\hat{x} - x^*\|_2^2 \triangleq \underline{\Delta} \tag{18.97}$$

- 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:

$$\max_{0 \leq \theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2}} \underline{\Delta} = \left(\frac{\|x^*\|_2^2 - (x^*)^\top \hat{x}}{\|\hat{x} - x^*\|_2} \right)^2 \tag{18.98}$$

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 18.98) is maximized.

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 18.98) is maximized.
- That is, we want to find

$$\hat{x} \in \underset{x \in P}{\operatorname{argmax}} \left(\frac{\|x^*\|_2^2 - (x^*)^\top x}{\|x - x^*\|_2} \right)^2 \tag{18.99}$$

to ensure that a large norm reduction is assured.

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 18.98) is maximized.
- That is, we want to find

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \left(\frac{\|x^*\|_2^2 - (x^*)^\top x}{\|x - x^*\|_2} \right)^2 \tag{18.99}$$

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. 18.99, i.e., find

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \|x^*\|_2^2 - (x^*)^\top x = \operatorname*{argmin}_{x \in P} (x^*)^\top x, \tag{18.100}$$

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

- As a surrogate, we maximize numerator in Eqn. 18.99, i.e., find

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \|x^*\|_2^2 - (x^*)^\top x = \operatorname*{argmin}_{x \in P} (x^*)^\top x, \qquad (18.100)$$

- 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.

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

- As a surrogate, we maximize numerator in Eqn. 18.99, i.e., find

$$\hat{x} \in \underset{x \in P}{\operatorname{argmax}} \|x^*\|_2^2 - (x^*)^\top x = \underset{x \in P}{\operatorname{argmin}} (x^*)^\top x, \tag{18.100}$$

- 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 $\hat{x}$ in Line 6 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.

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

- As a surrogate, we maximize numerator in Eqn. 18.99, i.e., find

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \|x^*\|_2^2 - (x^*)^\top x = \operatorname*{argmin}_{x \in P} (x^*)^\top x, \tag{18.100}$$

- 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 $\hat{x}$ in Line 6 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 and theoretically, we terminate the algorithm if

$$(x^*)^\top \hat{x} \geq \|x^*\|_2^2, \tag{18.101}$$

where $\hat{x}$ is the solution of Eq. 18.100.

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

$$(x^*)^\top \hat{x} > \|x^*\|_2^2 - \epsilon \max_{x \in Q} \|x\|_2^2 \tag{18.102}$$

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

$$(x^*)^\top \hat{x} > \|x^*\|_2^2 - \epsilon \max_{x \in Q} \|x\|_2^2 \tag{18.102}$$

- When $\mathrm{conv}\, P$ is a submodular base polytope (i.e., $\mathrm{conv}\, P = B_f$ for a submodular function f), then the problem in Eqn 18.100 can be solved efficiently by Edmonds's greedy algorithm (even though there may be an exponential number of extreme points).

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

$$(x^*)^\top \hat{x} > \|x^*\|_2^2 - \epsilon \max_{x \in Q} \|x\|_2^2 \tag{18.102}$$

- When $\mathrm{conv}\, P$ is a submodular base polytope (i.e., $\mathrm{conv}\, P = B_f$ for a submodular function f), then the problem in Eqn 18.100 can be solved efficiently by Edmonds's greedy algorithm (even though there may be an exponential number of extreme points).
- Edmond's greedy algorithm, therefore, solves both Line 4 and Line 6 simultaneously.

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

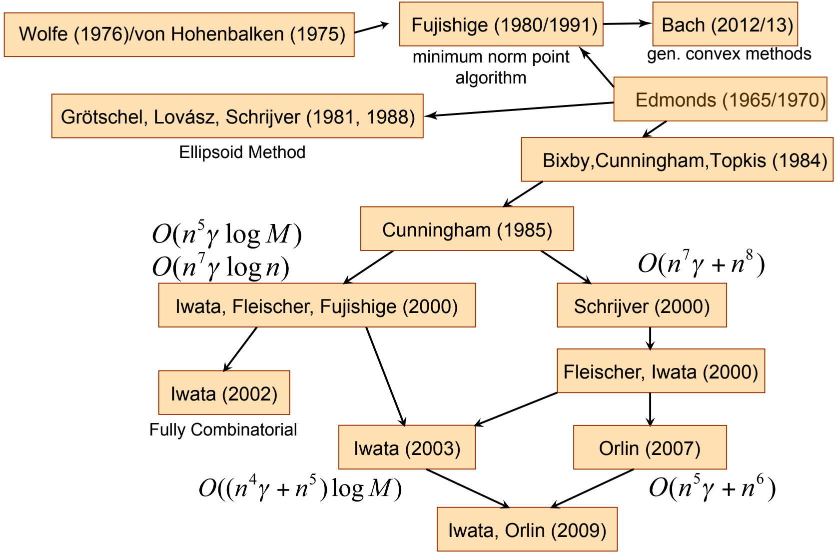
- In practice,the above optimality test might never hold numerically. Hence, as suggested by Wolfe, we introduce a tolerance parameter $\epsilon > 0$, and terminates the algorithm if

$$(x^*)^\top \hat{x} > \|x^*\|_2^2 - \epsilon \max_{x \in Q} \|x\|_2^2 \tag{18.102}$$

- 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 18.100 can be solved efficiently by Edmonds's greedy algorithm (even though there may be an exponential number of extreme points).
- Edmond's greedy algorithm, therefore, solves both Line 4 and Line 6 simultaneously.
- 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)

General Submodular Function Minimization



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.

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.

MN Algorithm Complexity

- The currently fastest strongly polynomial combinatorial algorithm for SFM achieves a running time of $O(n^5T + n^6)$ (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.

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:

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

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).

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.

MN Algorithm Complexity

- The currently fastest strongly polynomial combinatorial algorithm for SFM achieves a running time of $O(n^5T + n^6)$ (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

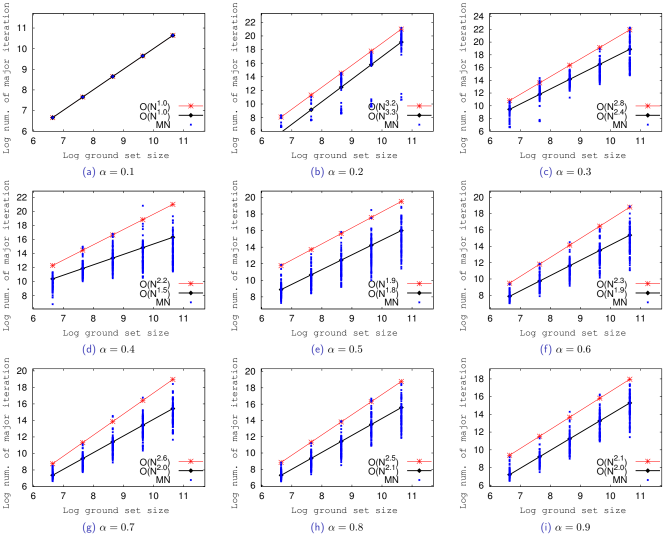


Figure: The number of major iteration for $f(S) = -m_1(S) + 100 \cdot (w_1(\mathcal{N}(S)))^{\alpha}$. The red lines are the linear interpolations of the worst case points, and the black lines are the linear interpolations of the average case points. From Lin&Bilmes 2014 (unpublished)

MN Algorithm Complexity

- A lower bound complexity of the min-norm has not been established.

MN Algorithm Complexity

- A lower bound complexity of the min-norm has not been established.
- In 2014, Chakrabarty, Jain, and Kothari in their NIPS 2014 paper "Provable Submodular Minimization using Wolfe's Algorithm" showed a pseudo-polynomial time bound of $O(n^7 g_f^2)$ where $n = |V|$ is the ground set, and g_f is the maximum gain of a particular function f.

MN Algorithm Complexity

- A lower bound complexity of the min-norm has not been established.
- In 2014, Chakrabarty, Jain, and Kothari in their NIPS 2014 paper "Provable Submodular Minimization using Wolfe's Algorithm" showed a pseudo-polynomial time bound of $O(n^7 g_f^2)$ where $n = |V|$ is the ground set, and g_f is the maximum gain of a particular function f.
- This is pseudo-polynomial since it depends on the function values.

MN Algorithm Complexity

- A lower bound complexity of the min-norm has not been established.
- In 2014, Chakrabarty, Jain, and Kothari in their NIPS 2014 paper "Provable Submodular Minimization using Wolfe's Algorithm" showed a pseudo-polynomial time bound of $O(n^7 g_f^2)$ where $n = |V|$ is the ground set, and g_f is the maximum gain of a particular function f.
- This is pseudo-polynomial since it depends on the function values.
- Therecurrently is no known polynomial time complexity analysis for this algorithm.