Submodular Functions, Optimization, and Applications to Machine Learning

— Fall Quarter, Lecture 20 —

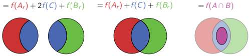
http://www.ee.washington.edu/people/faculty/bilmes/classes/ee563_spring_2018/

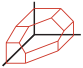
Prof. Jeff Bilmes

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

Dec 9th, 2020

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





Class Road Map - EE563

- L1(9/30): Motivation, Applications, Definitions, Properties
- L2(10/5): Sums concave(modular), uses (diversity/costs, feature selection), information theory
- L3(10/7): Monge, More Definitions, Graph and Combinatorial Examples,
- L4(10/12): Graph & Combinatorial Examples, Matrix Rank, Properties, Other Defs, Independence
- L5(10/14): Properties, Defs of Submodularity, Independence
- L6(10/19): Matroids, Matroid Examples, Matroid Rank,
- L7(10/21): Matroid Rank, More on Partition Matroid, Laminar Matroids, System of Distinct Reps, Transversals
- L8(10/26): Transversal Matroid, Matroid and representation, Dual Matroid
- L9(10/28): Other Matroid Properties, Combinatorial Geometries, Matroid and Greedy, Polyhedra, Matroid Polytopes
- L10(11/2): Matroid Polytopes, Matroids → Polymatroids
- L11(11/4): Matroids → Polymatroids, Polymatroids
- L12(11/9): Polymatroids, Polymatroids and Greedy
- L–(11/11): Veterans Day, Holiday
- L13(11/16): Polymatroids and Greedy, Possible Polytopes, Extreme Points, Cardinality Constrained Maximization
- L14(11/18): Cardinality Constrained Maximization, Curvature
- L15(11/23): Curvature, Submodular Max w. Other Constraints, Start Cont. Extensions
- L16(11/25): Submodular Max w. Other Constraints, Cont. Extensions, Lovász extension
- L17(11/30): Choquet Integration, Non-linear Measure/Aggregation, Definitions/Properties, Examples.
- L18(12/2): Multilinear Extension, Submodular Max/polyhedral, Most Violated Ineq., Matroids Closure/Sat
- L19(12/7): Fund. Circuit/Dep, SFM, L.E. primal, Start SFM via Min-Norm Point
- L20(12/9): support for min-norm, proof that min-norm gives optimal, computing min-norm vector in B_f, SFM
- L21(12/14): final meeting (presentations) maximization.

Last day of instruction, Fri. Dec 11th. Finals Week: Dec 12-18, 2020

Rest of class

- Homework 4 posted, due Thursday Dec 17th, 2020, 11:55pm.
- Final project 4-page paper and presentation slides, due Sunday Dec 13th, 11:59pm.
- Final project presentation, Monday Dec 14th, starting at 10:30am.
- Final project: Read and present a recent (past 5 years) paper on submodular/supermodular optimization. Paper should have both a theoretical and practical component. What is due: (1) 4-page paper summary, and (2) 10 minute presentation about the paper, will be giving presentations on Monday 12/14/2020. You must choose your paper before the 14th (this will be HW5), and you must turn in your slides and 4-page paper (this will be HW6).
- Recall, grades will be based on a combination of a final project (40%) and the four homeworks (60%).

Summary List 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

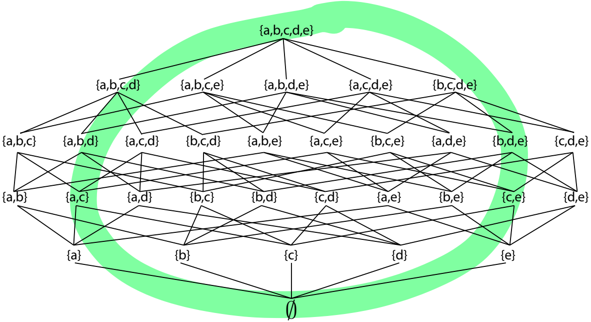
- x-tight sets: For $x \in P_f$, $\mathcal{D}(x) \triangleq \{A \subseteq E : x(A) = f(A)\}$.
- Polymatroid closure/maximal x-tight set: For $x \in P_f$, $\text{sat}(x) \triangleq \cup\{A : A \in \mathcal{D}(x)\} = \{e : e \in E, \forall\alpha > 0, x + \alpha\mathbf{1}_e \notin P_f\}$.
- Saturation capacity: 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 $x \in P_f$,

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

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

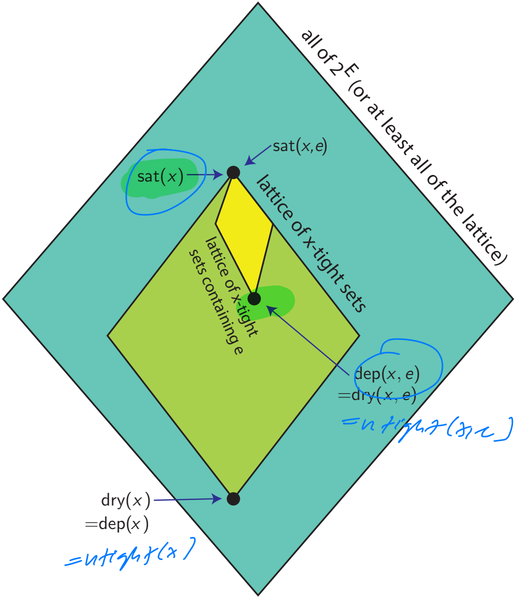
dep and sat in a lattice

- 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 Hasse diagram on 5 elements.



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?



Minimizing $\breve{f}$ vs. minimizing f

In fact, we have:

Theorem 20.2.4

Let f be submodular and $\breve{f}$ be its Lovász extension. Then $\min\{f(A)|A \subseteq E\} = \min_{w\in\{0,1\}^E} \breve{f}(w) = \min_{w\in[0,1]^E} \breve{f}(w)$.

Proof.

- First, since $\breve{f}(\mathbf{1}_A) = f(A), \forall A \subseteq V$, we clearly have $\min\{f(A)|A \subseteq V\} = \min_{w\in\{0,1\}^E} \breve{f}(w) \geq \min_{w\in[0,1]^E} \breve{f}(w)$.
- Next, consider any $w \in [0,1]^E$, sort elements $E = \{e_1, \dots, e_m\}$ as $w(e_1) \geq w(e_2) \geq \dots \geq w(e_m)$, define $E_i = \{e_1, \dots, e_i\}$, and define $\lambda_m = w(e_m)$ and $\lambda_i = w(e_i) - w(e_{i+1})$ for $i \in \{1, \dots, m-1\}$.
- Then, as we have seen, $w = \sum_i \lambda_i \mathbf{1}_{E_i}$ and $\lambda_i \geq 0$.
- Also, $\sum_i \lambda_i = w(e_1) \leq 1$.

...

Min-Norm Point: Definition

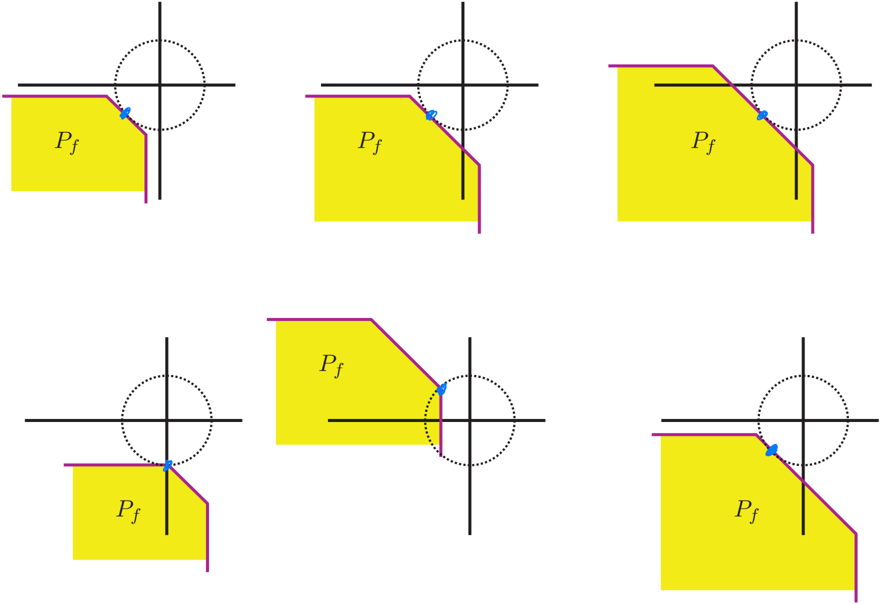
- Consider the optimization:

$$\begin{aligned} &\text{minimize} && \|x\|_2^2 && (20.25a)\\ &\text{subject to} && x \in B_f && (20.25b)\end{aligned}$$

  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



Min-Norm Point and Submodular Function Minimization

- Given optimal solution x^* to $[\min \|x\|_2^2 \text{ s.t. } x \in B_f]$, and consider:

$$y^* = x^* \wedge 0 = (\min(x^*(e), 0) | e \in E) \in P_f, \tag{20.25}$$

Since P_f is down closed. & sin B_f dominates.

$$A_- = \{e : x^*(e) < 0\}, \qquad A_0 = \{e : x^*(e) \leq 0\}. \tag{20.26}$$

- Thus, we immediately have that:

$$A_- \subseteq A_0 \tag{20.27}$$

and that

$$x^*(A_-) = x^*(A_0) = y^*(A_-) = y^*(A_0). \tag{20.28}$$

- These quantities will solve the SFM problem: we will see that $f(A_-) = f(A_0) = \min_{A \subseteq V} f(A)$ and that A_- is the unique minimal minimizer and A_0 is the unique maximal minimizer.
- The proof is nice since it uses recently developed tools (e.g., dep, sat).
- We'll also show both the Fujishige-Wolfe algorithm and the Frank-Wolfe algorithm (which are quite different from each other) can find the min-norm point relatively efficiently.

B_f dominates P_f

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

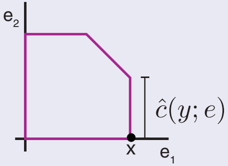
Theorem 20.2.6

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 $e \in T$.

Proof.

- We construct the y algorithmically: initially set $y \leftarrow x$.
- $y \in P_f$, T is tight for y so $y(T) = f(T)$.
- Recall saturation capacity: for $y \in P_f$, $\hat{c}(y;e) = \min\{f(A) - y(A) | \forall A \ni e\} = \max\{\alpha : \alpha \in \mathbb{R}, y + \alpha \mathbf{1}_e \in P_f\}$
- Consider following algorithm:

```
$T' \leftarrow T$ ;
for $e \in E \setminus T$ do
    $y \leftarrow y + c(y;e)\mathbf{1}_e$ ; $T' \leftarrow T' \cup \{e\}$;
```



...

Modified max-min theorem

- Min-max theorem (Thm 13.4.2) restated for $x = 0$.

$$\max\{y(E) | y \in P_f, y \leq 0\} = \min\{f(X) | X \subseteq V\} \tag{20.27}$$

Theorem 20.2.6 (Edmonds-1970)

$$\min\{f(X) | X \subseteq E\} = \max\left\{x^-(E) | x \in B_f\right\} \tag{20.28}$$

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

Proof via the Lovász ext.

$$\min\{f(X) | X \subseteq E\} = \min_{w \in [0,1]^E} \breve{f}(w) = \min_{w \in [0,1]^E} \max_{x \in P_f} w^\intercal x \tag{20.29}$$

$$= \min_{w \in [0,1]^E} \max_{x \in B_f} w^\intercal x \tag{20.30}$$

$$= \max_{x \in B_f} \min_{w \in [0,1]^E} w^\intercal x \tag{20.31}$$

$$= \max_{x \in B_f} x^-(E) \tag{20.32}$$

□

Max-min theorem, all forms

We start directly from Theorem 13.4.2.

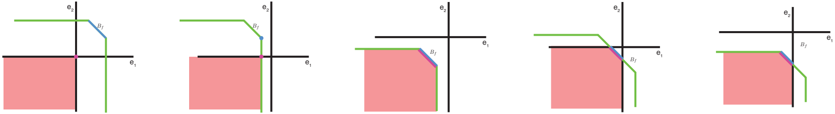
$$\max\left(y(E) : y \leq 0, y \in P_f\right) = \min\left(f(A) : A \subseteq E\right) \tag{20.1}$$

Theorem 20.3.1 (Edmond's Max-Min Theorem (restated))

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

$$\begin{aligned}\max\left(y(E) : y \leq 0, y \in P_f\right) &= \max\left(y^-(E) : y \leq 0, y \in P_f\right) && (20.2)\\ &= \max\left(y^-(E) : y \in P_f\right) && (20.3)\\ &= \max\left(y^-(E) : y \in B_f\right) && (20.4)\\ &= \min\left(f(A) : A \subseteq E\right) && (20.5)\end{aligned}$$

The first equality follows since $y \leq 0$. The second equality (together with the first) shown on 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 ~~??~~ 20.2.6).



Alt proof of $x^-(E)$ part of max-min theorem

Consider the following two problems for down-closed polyhedron P:

$$\max \sum_{e \in E} y(e) \quad (20.6a)$$
$$\text{s.t. } y \leq x \quad (20.6b)$$
$$y \in P \quad (20.6c)$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \quad (20.7a)$$
$$\text{s.t. } y \in P \quad (20.7b)$$

- Solutions identical cost. Let y_1^* be l.h.s. OPT and y_2^* be r.h.s. OPT.
- Consider l.h.s. OPT y_1^* in r.h.s. evaluation and suppose it is worse (lower) than r.h.s. OPT:

$$\sum_{e \in E} \min(y_1^*(e), x(e)) < \sum_{e \in E} \min(y_2^*(e), x(e)) \quad (20.8)$$

Alt proof of $x^-(E)$ part of max-min theorem

Consider the following two problems for down-closed polyhedron P:

$$\max \sum_{e\in E} y(e) \tag{20.6a}$$
$$\text{s.t. } y \le x \tag{20.6b}$$
$$y \in P \tag{20.6c}$$

$$\max \sum_{e\in E} \min(y(e), x(e)) \tag{20.7a}$$
$$\text{s.t. } y \in P \tag{20.7b}$$

- Solutions identical cost. Let y_1^* be l.h.s. OPT and y_2^* be r.h.s. OPT.
- Consider l.h.s. OPT y_1^* in r.h.s. evaluation and suppose it is worse (lower) than r.h.s. OPT:

l.h.s feasibh

$$\sum_{e\in E} \min(y_1^*(e), x(e)) < \sum_{e\in E} \min(y_2^*(e), x(e)) \tag{20.8}$$

- But the vector $\bar{y}_1^*$ with entries $\bar{y}_1^*(e) = \min(y_2^*(e), x(e))$ has $\bar{y}_1^*(e) \le x(e)$ and $\bar{y}_1^* \in P$ since $y_2^* \in P$, $\bar{y}_1^* \le y_2^*$, and P is down-closed.

Alt proof of $x^-(E)$ part of max-min theorem

Consider the following two problems for down-closed polyhedron P:

$$\max \sum_{e \in E} y(e) \tag{20.6a}$$

$$\text{s.t. } y \leq x \tag{20.6b}$$

$$y \in P \tag{20.6c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{20.7a}$$

$$\text{s.t. } y \in P \tag{20.7b}$$

- Solutions identical cost. Let y_1^* be l.h.s. OPT and y_2^* be r.h.s. OPT.
- Consider l.h.s. OPT y_1^* in r.h.s. evaluation and suppose it is worse (lower) than r.h.s. OPT:

$$\sum_{e \in E} \min(y_1^*(e), x(e)) \overset{\geq}{<} \sum_{e \in E} \min(y_2^*(e), x(e)) \tag{20.8}$$

- But the vector $\bar{y}_1^*$ with entries $\bar{y}_1^*(e) = \min(y_2^*(e), x(e))$ has $\bar{y}_1^*(e) \leq x(e)$ and $\bar{y}_1^* \in P$ since $y_2^* \in P$, $\bar{y}_1^* \leq y_2^*$, and P is down-closed.
- Thus, $\bar{y}_1^*$ is l.h.s. feasible but a better l.h.s. evaluation, a contradiction of the optimality of y_1^* for l.h.s.

Alt proof of $x^-(E)$ part of max-min theorem

Consider the following two problems for down-closed polyhedron P:

$$\max \sum_{e \in E} y(e) \tag{20.6a}$$

$$\text{s.t. } y \leq x \tag{20.6b}$$

$$y \in P \tag{20.6c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{20.7a}$$

$$\text{s.t. } y \in P \tag{20.7b}$$

- Solutions identical cost. Let y_1^* be l.h.s. OPT and y_2^* be r.h.s. OPT.
- Similarly, consider r.h.s. OPT y_2^* in l.h.s. evaluation and suppose it is worse (lower) than l.h.s. OPT

$$\sum_{e \in E} y_2^*(e) < \sum_{e \in E} y_1^*(e) \tag{20.8}$$

Alt proof of $x^-(E)$ part of max-min theorem

Consider the following two problems for down-closed polyhedron P:

$$\max \sum_{e \in E} y(e) \tag{20.6a}$$
$$\text{s.t. } y \leq x \tag{20.6b}$$
$$y \in P \tag{20.6c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{20.7a}$$
$$\text{s.t. } y \in P \tag{20.7b}$$

- Solutions identical cost. Let y_1^* be l.h.s. OPT and y_2^* be r.h.s. OPT.
- Similarly, consider r.h.s. OPT y_2^* in l.h.s. evaluation and suppose it is worse (lower) than l.h.s. OPT

$$\sum_{e \in E} y_2^*(e) < \sum_{e \in E} y_1^*(e) \tag{20.8}$$

- But the vector $\bar{y}_2^*$ with entries $\bar{y}_2^*(e) = y_1^*(e)$ has $\bar{y}_2^* \in P$ and since $\bar{y}_2^*(e) \leq x(e)$ for all e, we have

$$\sum_{e \in E} y_2^*(e) < \sum_{e \in E} y_1^*(e) = \sum_{e \in E} \bar{y}_2^*(e) = \sum_{e \in E} \min(\bar{y}_2^*(e), x(e)) \tag{20.9}$$

Alt proof of $x^-(E)$ part of max-min theorem

Consider the following two problems for down-closed polyhedron P:

$$\max \sum_{e \in E} y(e) \tag{20.6a}$$
$$\text{s.t. } y \leq x \tag{20.6b}$$
$$y \in P \tag{20.6c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{20.7a}$$
$$\text{s.t. } y \in P \tag{20.7b}$$

- Solutions identical cost. Let y_1^* be l.h.s. OPT and y_2^* be r.h.s. OPT.
- Similarly, consider r.h.s. OPT y_2^* in l.h.s. evaluation and suppose it is worse (lower) than l.h.s. OPT

$$\sum_{e \in E} y_2^*(e) < \sum_{e \in E} y_1^*(e) \tag{20.8}$$

- But the vector $\bar{y}_2^*$ with entries $\bar{y}_2^*(e) = y_1^*(e)$ has $\bar{y}_2^* \in P$ and since $\bar{y}_2^*(e) \leq x(e)$ for all e, we have

$$\sum_{e \in E} y_2^*(e) < \sum_{e \in E} y_1^*(e) = \sum_{e \in E} \bar{y}_2^*(e) = \sum_{e \in E} \min(\bar{y}_2^*(e), x(e)) \tag{20.9}$$

- Thus, we have r.h.s. feasible vector $\bar{y}_2^*$ strictly better than r.h.s. OPT contradicting the optimality of y_2^*.

Alt proof of $x^-(E)$ part of max-min theorem

Consider the following two problems for down-closed polyhedron P:

$$\max \sum_{e \in E} y(e) \tag{20.6a}$$
$$\text{s.t. } y \leq x \tag{20.6b}$$
$$y \in P \tag{20.6c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{20.7a}$$
$$\text{s.t. } y \in P \tag{20.7b}$$

- Solutions identical cost. Let y_1^* be l.h.s. OPT and y_2^* be r.h.s. OPT.
- Thus, l.h.s. and r.h.s. have identically valued solutions.

Alt proof of $x^-(E)$ part of max-min theorem

Consider the following two problems for down-closed polyhedron P:

$$\max \sum_{e \in E} y(e) \tag{20.6a}$$
$$\text{s.t. } y \leq x \tag{20.6b}$$
$$y \in P \tag{20.6c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{20.7a}$$
$$\text{s.t. } y \in P \tag{20.7b}$$

- Solutions identical cost. Let y_1^* be l.h.s. OPT and y_2^* be r.h.s. OPT.
- Thus, l.h.s. and r.h.s. have identically valued solutions.
- Hence, from previous slide, taking $x = 0$, $\max\left(y(E) : y \leq 0, y \in P_f\right) = \max\left(y^-(E) : y \in P_f\right) = \max\left(y^-(E) : y \in B_f\right)$

How to get a discrete SFM solution from the ~~dual~~ cont solution

- So we have $\max\{x^-(E) | x \in B_f\} = \underline{\min\{f(X) | X \subseteq E\}}$

Suppose x^* is a dual solution

How to get a discrete SFM solution from the dual solution

- So we have $\max \{x^-(E) | x \in B_f\} = \min \{f(X) | X \subseteq E\}$
- Suppose we have a solution x^* to l.h.s. Then $x^* \in P_f$, $x^*(E) = f(E)$, and $x^*(X) \leq f(X), \forall X \subseteq E$.

How to get a discrete SFM solution from the dual solution

- So we have $\max\{x^-(E) | x \in B_f\} = \min\{f(X) | X \subseteq E\}$
- Suppose we have a solution x^* to l.h.s. Then $x^* \in P_f$, $x^*(E) = f(E)$, and $x^*(X) \leq f(X), \forall X \subseteq E$.
- Define $A_- = \{e \in E : \overset{*}{x}(e) < 0\}$. Then $x^{*-}(A_-) = x^{*-}(E) = \min\{f(X) | X \subseteq E\}$.

$\forall e,$

$$\underline{x^{*-}(e) = \min(x^*(e), 0)}$$

How to get a discrete SFM solution from the dual solution

- So we have $\max\{x^-(E) | x \in B_f\} = \min\{f(X) | X \subseteq E\}$
- Suppose we have a solution x^* to l.h.s. Then $x^* \in P_f$, $x^*(E) = f(E)$, and $x^*(X) \leq f(X), \forall X \subseteq E$.
- Define $A_- = \{e \in E : x(e) < 0\}$. Then $x^{*-}(A_-) = x^{*-}(E) = \min\{f(X) | X \subseteq E\}$.
- That is, $x^*(A_-) = x^{*-}(A_-) = x^{*-}(E) \overset{\leq}{\cancel{\geq}} f(X), \forall X \subseteq E$ since x^* is optimal. In particular, $x^*(A_-) \underset{\leq}{\cancel{\geq}} f(A_-)$.

How to get a discrete SFM solution from the dual solution

- So we have $\max\{x^-(E) | x \in B_f\} = \min\{f(X) | X \subseteq E\}$
- Suppose we have a solution x^* to l.h.s. Then $x^* \in P_f$, $x^*(E) = f(E)$, and $x^*(X) \leq f(X), \forall X \subseteq E$.
- Define $A_- = \{e \in E : x(e) < 0\}$. Then $x^{*-}(A_-) = x^{*-}(E) = \min\{f(X) | X \subseteq E\}$.
- That is, $x^*(A_-) = x^{*-}(A_-) = x^{*-}(E) \geq f(X), \forall X \subseteq E$ since x^* is optimal. In particular, $x^*(A_-) \geq f(A_-)$.
- Since $x^* \in P_f$, we have $x^*(A_-) \leq f(A_-)$.

How to get a discrete SFM solution from the dual solution

- So we have $\max\{x^-(E) | x \in B_f\} = \min\{f(X) | X \subseteq E\}$
- Suppose we have a solution x^* to l.h.s. Then $x^* \in P_f$, $x^*(E) = f(E)$, and $x^*(X) \leq f(X), \forall X \subseteq E$.
- Define $A_- = \{e \in E : x(e) < 0\}$. Then $x^{*-}(A_-) = x^{*-}(E) = \min\{f(X) | X \subseteq E\}$.
- That is, $x^*(A_-) = x^{*-}(A_-) = x^{*-}(E) \geq f(X), \forall X \subseteq E$ since x^* is optimal. In particular, $x^*(A_-) \geq f(A_-)$.
- Since $x^* \in P_f$, we have $x^*(A_-) \leq f(A_-)$.
- Thus we have found an A_- such that $x^*(A_-) = f(A_-) = \min(f(X) : X \subseteq E)$ and A_- is a solution to the SFM problem.

How to get a discrete SFM solution from the dual solution

- So we have $\max\{x^-(E) | x \in B_f\} = \min\{f(X) | X \subseteq E\}$
- Suppose we have a solution x^* to l.h.s. Then $x^* \in P_f$, $x^*(E) = f(E)$, and $x^*(X) \leq f(X), \forall X \subseteq E$.
- Define $A_- = \{e \in E : x(e) < 0\}$. Then $x^{*-}(A_-) = x^{*-}(E) = \min\{f(X) | X \subseteq E\}$.
- That is, $x^*(A_-) = x^{*-}(A_-) = x^{*-}(E) \geq f(X), \forall X \subseteq E$ since x^* is optimal. In particular, $x^*(A_-) \geq f(A_-)$.
- Since $x^* \in P_f$, we have $x^*(A_-) \leq f(A_-)$.
- Thus we have found an A_- such that $x^*(A_-) = f(A_-) = \min(f(X) : X \subseteq E)$ and A_- is a solution to the SFM problem.
- Define $A_0 = \{e \in E : x(e) \leq 0\}$. Then above analysis holds as well, leading to $x^*(A_0) = f(A_0) = \min(f(X) : X \subseteq E)$.

How to get a discrete SFM solution from the dual solution

- So we have $\max\{x^-(E)|x \in B_f\} = \min\{f(X)|X \subseteq E\}$
- Suppose we have a solution x^* to l.h.s. Then $x^* \in P_f$, $x^*(E) = f(E)$, and $x^*(X) \leq f(X), \forall X \subseteq E$.
- Define $A_- = \{e \in E : x(e) < 0\}$. Then $x^{*-}(A_-) = x^{*-}(E) = \min\{f(X)|X \subseteq E\}$.
- That is, $x^*(A_-) = x^{*-}(A_-) = x^{*-}(E) \geq f(X), \forall X \subseteq E$ since x^* is optimal. In particular, $x^*(A_-) \geq f(A_-)$.
- Since $x^* \in P_f$, we have $x^*(A_-) \leq f(A_-)$.
- Thus we have found an A_- such that $x^*(A_-) = f(A_-) = \min(f(X) : X \subseteq E)$ and A_- is a solution to the SFM problem.
- Define $A_0 = \{e \in E : x(e) \leq 0\}$. Then above analysis holds as well, leading to $x^*(A_0) = f(A_0) = \min(f(X) : X \subseteq E)$.
- Thus, if we can find a solution to $\max\{x^-(E)|x \in B_f\}$ we have solved SFM.

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

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

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

$y = x + m$

sort $e_1, e_2, \dots, e_m$ so that $w(e_1) \ge w(e_2) \ge \dots \ge w(e_m)$

$y^*(e_1) = f'(e_1), \dots \; y^*(e_i) = f'(e_i | e_1, \dots, e_{i-1})$.

$x^*(e_i) = f(e_i | e_1, \dots, e_{i-1}) - m(e_i)$

$= f'(e_i | e_1, \dots, e_{i-1}) - m(e_i)$.

$= y^*(e_i) - m(e_i)$.

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

So y^* uses greedy algorithm with positive orthant $B_{f'}$. To show, we use Theorem 12.4.1 in Lecture 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.

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

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

$$\max \{w^\intercal x | x \in P_f\} = \max \{w^\intercal x | x \in B_f\} \tag{20.8}$$

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 $w \in \mathbb{R}_+^E$

$$\max \{w^\intercal x | x \in P_f\} = \max \{w^\intercal x | x \in B_f\} \tag{20.8}$$

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

- For arbitrary $w \in \mathbb{R}^E$, we saw in Lecture 16 that the greedy algorithm will also solve:

$$\max \{w^\intercal x | x \in B_f\} \tag{20.9}$$

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

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

$$\max \{w^\intercal x | x \in P_f\} = \max \{w^\intercal x | x \in B_f\} \tag{20.8}$$

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

- For arbitrary $w \in \mathbb{R}^E$, we saw in Lecture 16 that the greedy algorithm will also solve:

$$\max \{w^\intercal x | x \in B_f\} \tag{20.9}$$

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

$$\min \{w^\intercal x | x \in B_f\} = -\max \{-w^\intercal x | x \in B_f\} \tag{20.10}$$

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

$$w(e_1) \leq w(e_2) \leq \dots \leq w(e_m) \tag{20.11}$$

will solve l.h.s. of Equation (20.10).

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

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

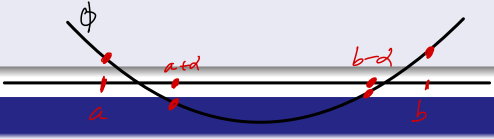
where $y = x + m$, so that $x^* = y^* - m$.
So y^* uses greedy algorithm with positive orthant $B_{f'}$. To show, we use Theorem 12.4.1 in Lecture 11, but we don't require $y \ge 0$, and don't stop when w goes negative to ensure $y^* \in B_{f'}$. Then when we subtract off m from y^*, we get solution to the original problem.

One last lemma

Lemma 20.3.2

Given function $\phi : \mathbb{R} \to \mathbb{R}$ and two points $a, b \in \mathbb{R}$ with $a < b$. Then ϕ is convex in the region $[a, b]$ if and only if

$$\phi(a) + \phi(b) \geq \phi(a + \alpha) + \phi(b - \alpha), \forall \alpha \in [0, b - a] \tag{20.12}$$



Proof.

This inequality is the same as

$$f(b) - f(b - \alpha) \geq f(a + \alpha) - f(a) \tag{20.13}$$

and the rest follows from Bilmes&Bai, "Deep Submodular Functions", Theorem 5.3 (which shows the corresponding theorem for concave functions). □

Min-Norm Point and Submodular Function Minimization

- Given optimal solution x^* to $[\min \|x\|_2^2$ s.t. $x \in B_f]$, and consider:

$$y^* = x^* \wedge 0 = (\min(x^*(e), 0) | e \in E) \in P_f, \tag{20.25}$$

$$A_- = \{e : x^*(e) < 0\}, \qquad A_0 = \{e : x^*(e) \leq 0\}. \tag{20.26}$$

- Thus, we immediately have that:

$$A_- \subseteq A_0 \tag{20.27}$$

and that

$$x^*(A_-) = x^*(A_0) = y^*(A_-) = y^*(A_0). \tag{20.28}$$

- These quantities will solve the SFM problem: we will see that $f(A_-) = f(A_0) = \min_{A \subseteq V} f(A)$ and that A_- is the unique minimal minimizer and A_0 is the unique maximal minimizer.
- The proof is nice since it uses recently developed tools (e.g., dep, sat).
- We'll also show both the Fujishige-Wolfe algorithm and the Frank-Wolfe algorithm (which are quite different from each other) can find the min-norm point relatively efficiently.

Min-Norm Point and SFM

Theorem 20.4.1

Let x^*, y^*, A_-, and A_0 be as given. Then y^* is a maximizer of $\max\{y(E) | y \in P_f, y \leq 0\}$ [redacted]*, 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 may consider any $e \in E$ within $\text{dep}(x^*, e)$.

. . .

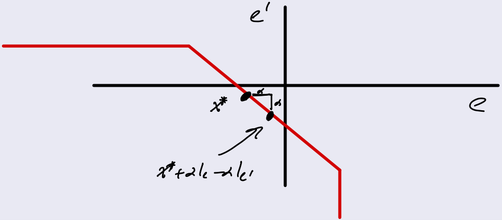
Min-Norm Point and SFM

Theorem 20.4.1

Let x^*, y^*, A_-, and A_0 be as given. Then y^* is a maximizer of $\max\{y(E) | y \in P_f, y \leq 0\} = \max(y^-(E) : y \in B_f)$, 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 may consider any $e \in E$ within $\text{dep}(x^*, e)$.
- Consider any pair (e, e') with $e \in A_-$ and $e' \in \text{dep}(x^*, e)$. 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 20.4.1

Let x^*, y^*, A_-, and A_0 be as given. Then y^* is a maximizer of $\max\{y(E)|y \in P_f, y \leq 0\} = \max(y^-(E) : y \in B_f)$, *$A_-$ 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 may consider any $e \in E$ within $\text{dep}(x^*, e)$.
- Consider any pair (e, e') with $e \in A_-$ and $e' \in \text{dep}(x^*, e)$. 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

$$(x^* + \alpha \mathbf{1}_e - \alpha \mathbf{1}_{e'})(E) = x^*(E) + \alpha - \alpha = f(E) \tag{20.14}$$

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

...

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 cont.

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

$$\boxed{x^*_{new}(a) = x^*(a) \quad \forall\ a \notin \{e, e'\}}$$

. . .

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 cont.

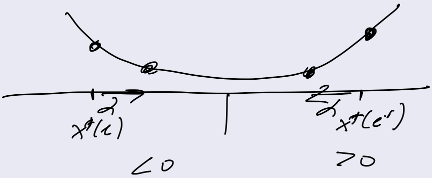
- Then $(x^* + \alpha \mathbf{1}_e - \alpha \mathbf{1}_{e'})(E)$ $= x^*(E \setminus \{e, e'\}) + \underbrace{(x^*(e) + \alpha)}_{x^*_{\text{new}}(e)} + \underbrace{(x^*(e') - \alpha)}_{x^*_{\text{new}}(e')} = f(E)$.
- Minimality of $x^* \in B_f$ in l2 sense requires that, with such an $\alpha > 0$, $\Big(x^*(e)\Big)^2 + \Big(x^*(e')\Big)^2 < \Big(x^*_{\text{new}}(e)\Big)^2 + \Big(x^*_{\text{new}}(e')\Big)^2$

. . .

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 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 would have $(x^*(e) + \alpha')^2 + (x^*(e') - \alpha')^2 < \underline{(x^*(e))^2 + (x^*(e'))^2}$, for some $0 < \alpha' \leq \alpha$, contradicting the optimality of x^*.

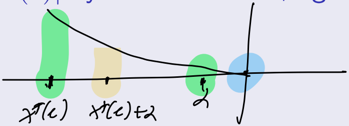


. . .

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 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 would have $(x^*(e) + \alpha')^2 + (x^*(e') - \alpha')^2 < (x^*(e))^2 + (x^*(e'))^2$, for some $0 < \alpha' \leq \alpha$, contradicting the optimality of x^*.
- If $x^*(e') = 0$, we would have $(x^*(e) + \alpha')^2 + (\alpha')^2 < (x^*(e))^2$ $+0$, for any $0 < \alpha' < |x^*(e)|$ by Lemma 20.3.2, again contradicting optimality of x^*.



. . .

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 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 would have $(x^*(e) + \alpha')^2 + (x^*(e') - \alpha')^2 < (x^*(e))^2 + (x^*(e'))^2$, for some $0 < \alpha' \leq \alpha$, contradicting the optimality of x^*.
- If $x^*(e') = 0$, we would have $(x^*(e) + \alpha')^2 + (\alpha')^2 < (x^*(e))^2$, for any $0 < \alpha' < |x^*(e)|$ by Lemma 20.3.2, again contradicting optimality of x^*.
- Thus, we must have $x^*(e') < 0$ (strict negativity).

. . .

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 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. 20.4.1 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. 20.4.1 cont.

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

. . .

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 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 $\text{dep}(x^*, e) \subseteq A_0$ (Exercise).
- Also, recall that $e \in \text{dep}(x^*, e)$.

. . .

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 cont.

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

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 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.

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 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. 20.4.1 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:

$$x^*(A_-) = f(A_-) \tag{20.15}$$

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 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:

$$x^*(A_-) = f(A_-) \tag{20.15}$$

$$x^*(A_0) = f(A_0) \tag{20.16}$$

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 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:

$$x^*(A_-) = f(A_-) \tag{20.15}$$

$$x^*(A_0) = f(A_0) \tag{20.16}$$

$$x^*(A_-) = x^*(A_0) = y^*(E) = y^*(A_0) + \underbrace{y^*(E \setminus A_0)}_{=0} \tag{20.17}$$

Min-Norm Point and SFM

... proof of Thm. 20.4.1 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:

$$x^*(A_-) = f(A_-) \tag{20.15}$$

$$x^*(A_0) = f(A_0) \tag{20.16}$$

$$x^*(A_-) = x^*(A_0) = y^*(E) = y^*(A_0) + \underbrace{y^*(E \setminus A_0)}_{=0} \tag{20.17}$$

and therefore, all together we have

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 cont.

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

$$x^*(A_-) = f(A_-) \tag{20.15}$$

$$x^*(A_0) = f(A_0) \tag{20.16}$$

$$x^*(A_-) = x^*(A_0) = y^*(E) = y^*(A_0) + \underbrace{y^*(E \setminus A_0)}_{=0} \tag{20.17}$$

and therefore, all together we have

$$f(A_-) = f(A_0) = x^*(A_-) = x^*(A_0) = y^*(E) \tag{20.18}$$

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 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:

$$x^*(A_-) = f(A_-) \tag{20.15}$$

$$x^*(A_0) = f(A_0) \tag{20.16}$$

$$x^*(A_-) = x^*(A_0) = y^*(E) = y^*(A_0) + \underbrace{y^*(E \setminus A_0)}_{=0} \tag{20.17}$$

and therefore, all together we have

$$f(A_-) = f(A_0) = x^*(A_-) = x^*(A_0) = y^*(E) \tag{20.18}$$

- 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. 20.4.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (20.1) (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. 20.4.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (20.1) (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. 20.4.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (20.1) (recall, which is $\max\{y(E) | y \in P_f, y \le 0\} = \min\{f(X) | X \subseteq V\}$). This follows since, we have $y^* = x^* \wedge 0 \le 0$, and since $x^* \in B_f \subset P_f$, and $y^* \le x^*$ and P_f is down-closed, we have that $y^* \in P_f$.
- Also, for any $y \in P_f$ with $y \le 0$ and for any $X \subseteq E$, we have $y(E) \le y(X) \le f(X)$.

$$\underbrace{y(E) \le y(X)}_{\text{since } y \le 0}\;\underbrace{\le f(X)}_{\text{since } y \in P_f}$$

. . .

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (20.1) (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. (20.1), $y^* \leq 0$, $y^* \in P_f$, so $y^*(E) \leq f(X)$ for all X.

. . .

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 cont.

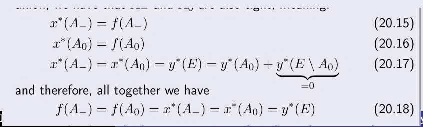
- Now, y^* is feasible for the l.h.s. of Eqn. (20.1) (recall, which is $\max\{y(E)|y \in P_f, y \leq 0\} = \min\{f(X)|X \subseteq V\}$). This follows since, we have $y^* = x^* \wedge 0 \leq 0$, and since $x^* \in B_f \subset P_f$, and $y^* \leq x^*$ and P_f is down-closed, we have that $y^* \in P_f$.
- 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. (20.1), $y^* \leq 0$, $y^* \in P_f$, 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. 20.4.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (20.1) (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. (20.1), $y^* \leq 0$, $y^* \in P_f$, so $y^*(E) \leq f(X)$ for all X.
- So $y^*(E) \leq \min\{f(X)|X \subseteq V\}$.
- Considering Eqn. (20.15), we have found sets A_- and A_0 with tightness in Eqn. (20.1), meaning $y^*(E) = f(A_-) = f(A_0)$.



union, we have that A_- and A_0 are also tight, meaning:

$$x^*(A_-) = f(A_-) \tag{20.15}$$

$$x^*(A_0) = f(A_0) \tag{20.16}$$

$$x^*(A_-) = x^*(A_0) = y^*(E) = y^*(A_0) + \underbrace{y^*(E \setminus A_0)}_{=0} \tag{20.17}$$

and therefore, all together we have

$$f(A_-) = f(A_0) = x^*(A_-) = x^*(A_0) = y^*(E) \tag{20.18}$$

. . .

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (20.1) (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. (20.1), $y^* \leq 0$, $y^* \in P_f$, so $y^*(E) \leq f(X)$ for all X.
- So $y^*(E) \leq \min\{f(X)|X \subseteq V\}$.
- Considering Eqn. (20.15), we have found sets A_- and A_0 with tightness in Eqn. (20.1), meaning $y^*(E) = f(A_-) = f(A_0)$.
- Hence, y^* is a maximizer of l.h.s. of Eqn. (20.1), and A_- and A_0 are minimizers of f.

. . .

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 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

□

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 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

$$f(X) \geq x^*(X) > x^*(A_-) = f(A_-) \tag{20.19}$$

□

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 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

$$f(X) \geq x^*(X) > x^*(A_-) = f(A_-) \tag{20.19}$$

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

$$f(X) \geq x^*(X) > x^*(A_0) = f(A_0) \tag{20.20}$$

□

Min-Norm Point and SFM

. . . proof of Thm. 20.4.1 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

$$f(X) \geq x^*(X) > x^*(A_-) = f(A_-) \tag{20.19}$$

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

$$f(X) \geq x^*(X) > x^*(A_0) = f(A_0) \tag{20.20}$$

- 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 will see, 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 will see, the algorithm (by [redacted] 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 still currently the best practical algorithm for general purpose submodular function minimization (although other algorithms have better asymptotic complexity).

much lower

Min-norm point and other minimizers of f

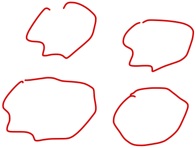
- Recall, that the set of minimizers of f forms a lattice.

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? No. Consider graph cut function with graph with multiple connected components, so $A_- = \emptyset$, $A_0 = V$ but not all $A : A_- \subset A \subset A_0$ is 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? No. Consider graph cut function with graph with multiple connected components, so $A_- = \emptyset$, $A_0 = V$ but not all $A : A_- \subset A \subset A_0$ is 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? No. Consider graph cut function with graph with multiple connected components, so $A_- = \emptyset$, $A_0 = V$ but not all $A : A_- \subset A \subset A_0$ is a minimizer.
- In fact, with x^* the min-norm point, and A_- and A_0 as defined above, we have the following theorem:

Theorem 20.4.2

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:

$$A = A_- \cup \bigcup_{a \in A_m} \operatorname{dep}(x^*, a) \tag{20.21}$$

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. 20.4.2.

- 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. 20.4.2.

- 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 $f(A) = x^*(A_-) \leq x^*(A)$.

. . .

Min-norm point and other minimizers of f

proof of Thm. 20.4.2.

- 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 $f(A) = x^*(A_-) \leq x^*(A)$.
- Also, since $A \subseteq A_0$ and $x^*(A_0 \setminus A) = 0$, $x^*(A_-) = x^*(A) = x^*(A_0)$

. . .

Min-norm point and other minimizers of f

proof of Thm. 20.4.2.

- 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 $f(A) = x^*(A_-) \leq x^*(A)$.
- Also, since $A \subseteq A_0$ and $x^*(A_0 \setminus A) = 0$, $x^*(A_-) = x^*(A) = x^*(A_0)$
- 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. 20.4.2.

- 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 $f(A) = x^*(A_-) \leq x^*(A)$.
- Also, since $A \subseteq A_0$ and $x^*(A_0 \setminus A) = 0$, $x^*(A_-) = x^*(A) = x^*(A_0)$
- 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. 20.4.2.

- 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 $f(A) = x^*(A_-) \leq x^*(A)$.
- Also, since $A \subseteq A_0$ and $x^*(A_0 \setminus A) = 0$, $x^*(A_-) = x^*(A) = x^*(A_0)$
- 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. 20.4.2.

- 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 $f(A) = x^*(A_-) \leq x^*(A)$.
- Also, since $A \subseteq A_0$ and $x^*(A_0 \setminus A) = 0$, $x^*(A_-) = x^*(A) = x^*(A_0)$
- 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 $\bigcup_{a \in A} \text{dep}(x^*, a) = A$. $a \in dep(x^*, a)$

...

Min-norm point and other minimizers of f

proof of Thm. 20.4.2.

- 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 $f(A) = x^*(A_-) \leq x^*(A)$.
- Also, since $A \subseteq A_0$ and $x^*(A_0 \setminus A) = 0$, $x^*(A_-) = x^*(A) = x^*(A_0)$
- 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. 20.4.2.

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

$$A = A_- \cup \bigcup_{a \in A_m} \operatorname{dep}(x^*, a) = \bigcup_{a \in A_-} \operatorname{dep}(x^*, a) \cup \bigcup_{a \in A_m} \operatorname{dep}(x^*, a) \qquad (20.22)$$

□

Min-norm point and other minimizers of f

proof of Thm. 20.4.2.

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

$$A = A_- \cup \bigcup_{a \in A_m} \operatorname{dep}(x^*, a) = \bigcup_{a \in A_-} \operatorname{dep}(x^*, a) \cup \bigcup_{a \in A_m} \operatorname{dep}(x^*, a) \tag{20.22}$$

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

□

Min-norm point and other minimizers of f

proof of Thm. 20.4.2.

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

$$A = A_- \cup \bigcup_{a \in A_m} \mathrm{dep}(x^*, a) = \bigcup_{a \in A_-} \mathrm{dep}(x^*, a) \cup \bigcup_{a \in A_m} \mathrm{dep}(x^*, a) \tag{20.22}$$

- Then since A is a union of tight sets, A is also a tight set, and we have $f(A) = x^*(A)$.
- But since for any $a \in A_0$, $\mathrm{dep}(x^*, a) \subseteq A_0$ then $A \subseteq A_0$ and we have that $x^*(A \setminus A_-) = 0$, so $f(A) = x^*(A) = x^*(A_-) = f(A_-)$ meaning A is also a minimizer of f.

□

Min-norm point and other minimizers of f

proof of Thm. 20.4.2.

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

$$A = A_- \cup \bigcup_{a \in A_m} \operatorname{dep}(x^*, a) = \bigcup_{a \in A_-} \operatorname{dep}(x^*, a) \cup \bigcup_{a \in A_m} \operatorname{dep}(x^*, a) \tag{20.22}$$

- Then since A is a union of tight sets, A is also a tight set, and we have $f(A) = x^*(A)$.
- But since for any $a \in A_0$, $\operatorname{dep}(x^*, a) \subseteq A_0$ then $A \subseteq A_0$ and we have that $x^*(A \setminus A_-) = 0$, so $f(A) = x^*(A) = x^*(A_-) = f(A_-)$ meaning A is also a minimizer of f.

□

Min-norm point and other minimizers of f

proof of Thm. 20.4.2.

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

$$A = A_- \cup \bigcup_{a \in A_m} \text{dep}(x^*, a) = \bigcup_{a \in A_-} \text{dep}(x^*, a) \cup \bigcup_{a \in A_m} \text{dep}(x^*, a) \tag{20.22}$$

- Then since A is a union of tight sets, A is also a tight set, and we have $f(A) = x^*(A)$.
- But since for any $a \in A_0$, $\text{dep}(x^*, a) \subseteq A_0$ then $A \subseteq A_0$ and we have that $x^*(A \setminus A_-) = 0$, 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 $\text{dep}(x^*, e)$.

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):

$$\underset{w \in \mathbb{R}^V}{\text{minimize}} \; \breve{f}(w) + \frac{1}{2}\|w\|_2^2 \qquad (20.23)$$

$$\text{maximize} \quad -\frac{1}{2}\|x\|_2^2 \qquad (20.24a)$$

$$\text{subject to} \quad x \in B_f \qquad (20.24b)$$

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

- Equation (20.23) is related to proximal methods to minimize the Lovász extension (see Parikh&Boyd, "Proximal Algorithms" 2013).
- Equation (20.24b) is solved by the minimum-norm point algorithm (Wolfe-1976, Fujishige-1984, Fujishige-2005, Fujishige-2011) is essentially an active-set procedure for quadratic programming, and uses Edmonds's greedy algorithm to make it efficient. ← or Frank-Wolfe.
- These algorithms usually perform quite well in practice, they can be made to perform about the same, given a properly tuned implementation (also, the FrankWolfe based algorithm is much simpler).

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 (again, not same as Frank-Wolfe’1956).

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 (again, not same as Frank-Wolfe'1956).
- 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:

$$\min_{x \in \text{conv } P} \|x\|_2 \tag{20.25}$$

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 (again, not same as Frank-Wolfe'1956).
- 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:

$$\min_{x \in \text{conv } P} \|x\|_2 \tag{20.25}$$

- 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 (again, not same as Frank-Wolfe’1956).
- 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:

$$\min_{x \in \text{conv } P} \|x\|_2 \tag{20.25}$$

- Wolfe’s algorithm is guaranteed terminating, and explicitly uses a representation of x as a convex combination of points in P
- Fujishige-1984 “Submodular Systems and Related Topics” realized this algorithm can find the the min. norm point of B_f thanks to Edmond’s greedy algorithm.

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 (again, not same as Frank-Wolfe'1956).
- 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:

$$\min_{x \in \text{conv } P} \|x\|_2 \tag{20.25}$$

- Wolfe's algorithm is guaranteed terminating, and explicitly uses a representation of x as a convex combination of points in P
- Fujishige-1984 "Submodular Systems and Related Topics" realized this algorithm can find the the min. norm point of B_f thanks to Edmond's greedy algorithm.
- Seems to still be (among) the fastest general purpose SFM algorithms in practice.

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.,

$$\operatorname{conv} P \triangleq \left\{ \sum_{i=1}^{k} \lambda_i p_i : \sum_i \lambda_i = 1,\ \lambda_i \geq 0, i \in [k] \right\}. \tag{20.26}$$

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.,

$$\operatorname{conv} P \triangleq \left\{ \sum_{i=1}^{k} \lambda_i p_i : \sum_{i} \lambda_i = 1,\ \lambda_i \geq 0, i \in [k] \right\}. \tag{20.26}$$

- 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.:

$$\operatorname{aff} Q \triangleq \left\{ \sum_{i \in 1}^{k} \lambda_i q_i : \sum_{i=1}^{k} \lambda_i = 1 \right\} \tag{20.27}$$

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 $\text{conv}\, P$ be the convex hull of P, i.e.,

$$\text{conv}\, P \triangleq \left\{ \sum_{i=1}^{k} \lambda_i p_i : \sum_i \lambda_i = 1,\ \lambda_i \geq 0, i \in [k] \right\}. \tag{20.26}$$

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

$$\text{aff}\, Q \triangleq \left\{ \sum_{i\in 1}^{k} \lambda_i q_i : \sum_{i=1}^{k} \lambda_i = 1 \right\} \supseteq \text{conv}\, Q. \tag{20.27}$$

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.,

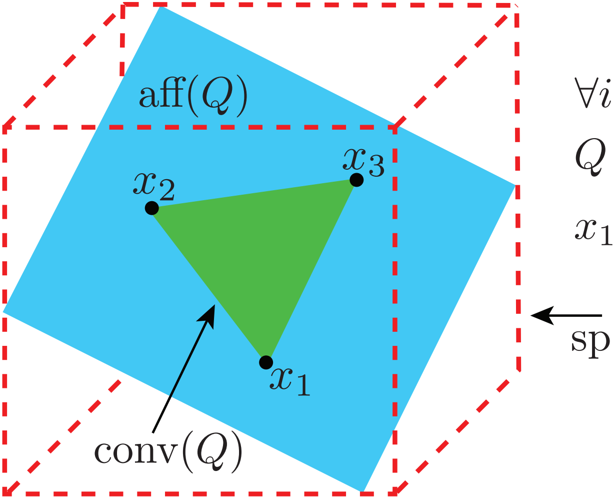
$$\operatorname{conv} P \triangleq \left\{ \sum_{i=1}^{k} \lambda_i p_i : \sum_{i} \lambda_i = 1,\ \lambda_i \geq 0, i \in [k] \right\}. \qquad (20.26)$$

- 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.:

$$\operatorname{aff} Q \triangleq \left\{ \sum_{i\in 1}^{k} \lambda_i q_i : \sum_{i=1}^{k} \lambda_i = 1 \right\} \supseteq \operatorname{conv} Q. \qquad (20.27)$$

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

Convex vs. Affine hull, geometry



$\forall i, x_i \in \mathbb{R}^3$

$Q = \{x_1, x_2, x_3\}$

x_1, x_2, x_3 coplanar

$H(x)$: 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.

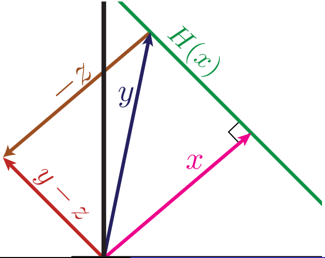
$$H(x) \triangleq \left\{ y \in \mathbb{R}^V \mid x^\intercal y = \|x\|_2^2 \right\} \tag{20.28}$$

$H(x)$: 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.

$$H(x) \triangleq \left\{ y \in \mathbb{R}^V \mid x^\intercal y = \|x\|_2^2 \right\} \tag{20.28}$$

- 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.

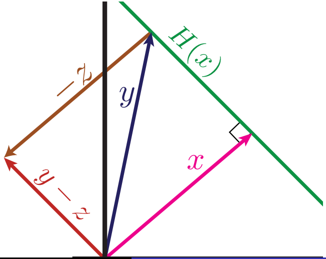


$H(x)$: Orthogonal x-containing hyperplane

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

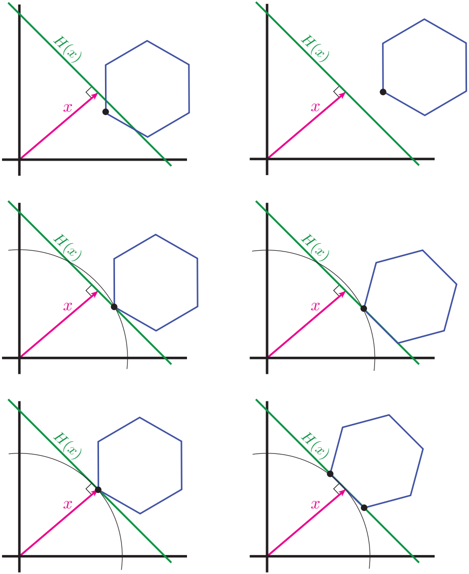
$$H(x) \triangleq \left\{ y \in \mathbb{R}^V \mid x^\intercal y = \|x\|_2^2 \right\} \tag{20.28}$$

- 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.



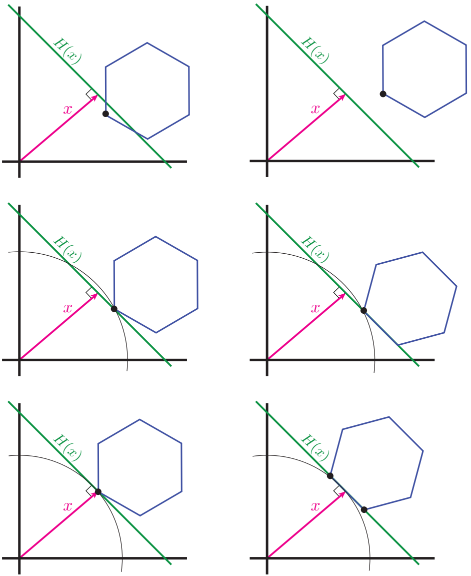
Ex: $H(x)$, polytopes, and supporting hyperplanes

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



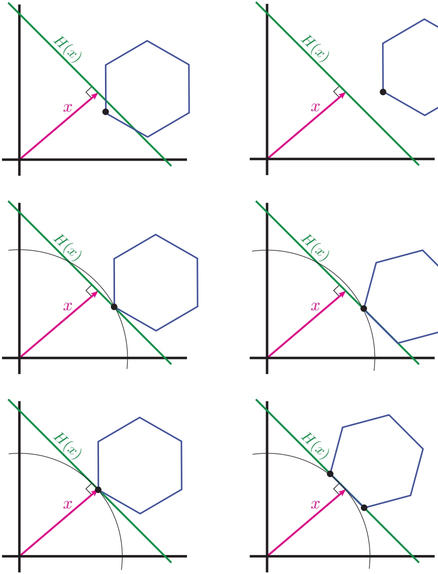
Ex: $H(x)$, polytopes, and supporting hyperplanes

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



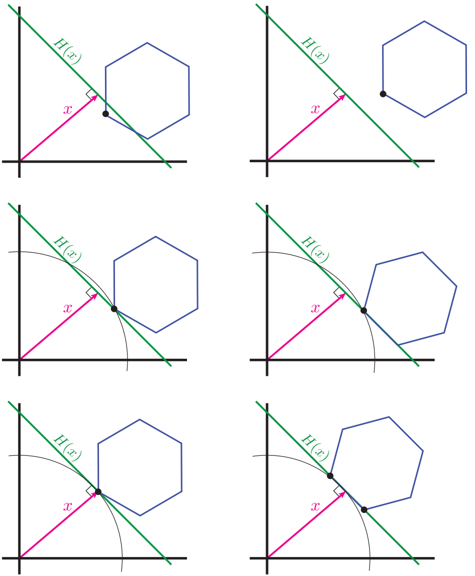
Ex: $H(x)$, polytopes, and supporting hyperplanes

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

- Algorithm maintains a set of points $Q \subseteq P$, which is always assuredly *affinely independent*, and also $|Q|$ doesn't grow large even if $|P|$ is large.

Fujishige-Wolfe Min-Norm Algorithm

- Algorithm maintains a set of points $Q \subseteq P$, which is always assuredly *affinely independent*, and also $|Q|$ doesn't grow large even if $|P|$ is large.
- 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

- Algorithm maintains a set of points $Q \subseteq P$, which is always assuredly *affinely independent*, and also $|Q|$ doesn't grow large even if $|P|$ is large.
- 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.

Fujishige-Wolfe Min-Norm Algorithm

- Algorithm maintains a set of points $Q \subseteq P$, which is always assuredly *affinely independent*, and also $|Q|$ doesn't grow large even if $|P|$ is large.
- 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

- Algorithm maintains a set of points $Q \subseteq P$, which is always assuredly *affinely independent*, and also $|Q|$ doesn't grow large even if $|P|$ is large.
- 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

- Algorithm maintains a set of points $Q \subseteq P$, which is always assuredly *affinely independent*, and also $|Q|$ doesn't grow large even if $|P|$ is large.
- 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 original 1976 algorithm is solving linear optimization problem over the polytope, doable by examining all the extreme points in the polytope.

Fujishige-Wolfe Min-Norm Algorithm

- Algorithm maintains a set of points $Q \subseteq P$, which is always assuredly *affinely independent*, and also $|Q|$ doesn't grow large even if $|P|$ is large.
- 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 original 1976 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

- Algorithm maintains a set of points $Q \subseteq P$, which is always assuredly *affinely independent*, and also $|Q|$ doesn't grow large even if $|P|$ is large.
- 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 $\operatorname{conv} Q$ 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 original 1976 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;
            $x^* \longleftarrow y$;
```

N: Pseudocode 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 $\operatorname{conv} P$.
$p \longleftarrow p_{i^*}$ where $p_{i^*} \in \operatorname{argmin}_{p \in P} \|p\|_2$, or choose $p \in P$ arbitrarily or
 heuristically ;
$Q \longleftarrow \{x^*\}$;
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:

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{20.29}$$

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:

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{20.29}$$

  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

$$\min_{x \in \operatorname{aff} Q} \|x\|_2 \leq \min_{x \in \operatorname{conv} Q} \|x\|_2 \leq \|x^*\|_2 \tag{20.30}$$

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:

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{20.29}$$

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

$$\min_{x \in \operatorname{aff} Q} \|x\|_2 \leq \min_{x \in \operatorname{conv} Q} \|x\|_2 \leq \|x^*\|_2 \tag{20.30}$$

- 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:

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{20.29}$$

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

$$\min_{x \in \operatorname{aff} Q} \|x\|_2 \leq \min_{x \in \operatorname{conv} Q} \|x\|_2 \leq \|x^*\|_2 \tag{20.30}$$

- 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:

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{20.29}$$

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

$$\min_{x \in \operatorname{aff} Q} \|x\|_2 \leq \min_{x \in \operatorname{conv} Q} \|x\|_2 \leq \|x^*\|_2 \tag{20.30}$$

- 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.

N: Pseudocode 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 $\mathrm{conv}\, P$.
$p \longleftarrow p_{i^*}$ where $p_{i^*} \in \mathrm{argmin}_{p \in P} \|p\|_2$, or choose $p \in P$ arbitrarily or
  heuristically ;
$Q \longleftarrow \{x^*\}$;
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 */
        $x_0 \longleftarrow \mathrm{argmin}_{x \in \mathrm{aff}\, Q} \|x\|_2$;
        if $x_0 \in \mathrm{conv}\, Q$ then
            $x^* \longleftarrow x_0$;
            break;
        else
            $y \longleftarrow \mathrm{argmin}_{x \in \mathrm{conv}\, Q \cap [x^*, x_0]} \|x - x_0\|_2$;
            Delete from $Q$ points not on the face of $\mathrm{conv}\, Q$ where $y$ lies;
            $x^* \longleftarrow y$;
```

Solved by Edmond's greedy procedure.

Solved via linear equation solver.

Linear equation solver represents x_0 as affine coefs, so this just checks >= 0.

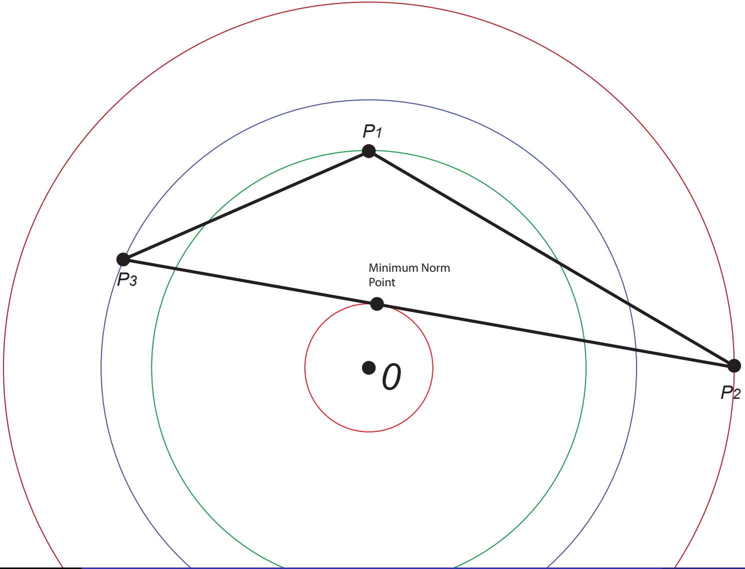
Doable since we're representing points as convex combinations of points within Q

Fujishige-Wolfe Min-Norm algorithm: Geometric Example

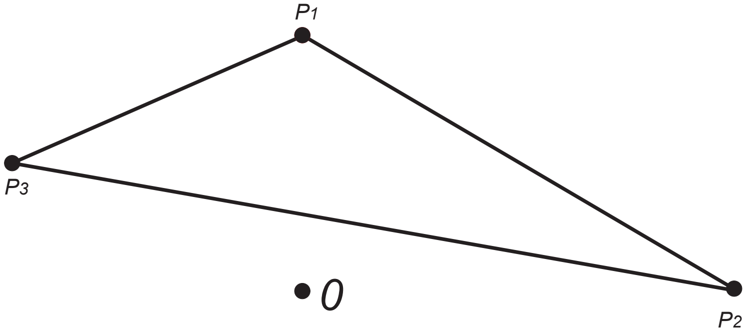
- In the following series of images, permanent (non-changing) named points on the polytope will be indicated by capital letters (i.e., P_1, P_2, P_3, R, S, T) while variables in the algorithm that are changing will use lower case letters (i.e., x^*, x_0, $\hat{x}$, y).
- Also, example is in 2D, so polytope given can't be a real base B_f for any f. Example meant to show only the geometry of the algorithm.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example

Polytope, and circles concentric at 0.

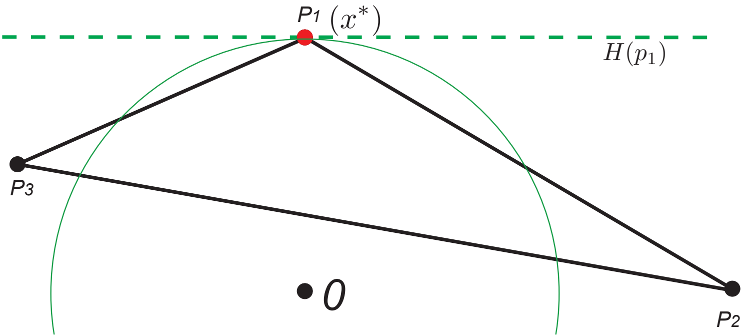


Fujishige-Wolfe Min-Norm algorithm: Geometric Example



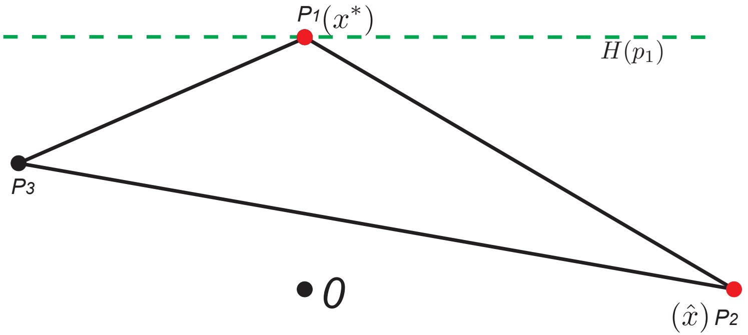
The initial polytope consisting of the convex hull of three points p_1, p_2, p_3, and the origin 0.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



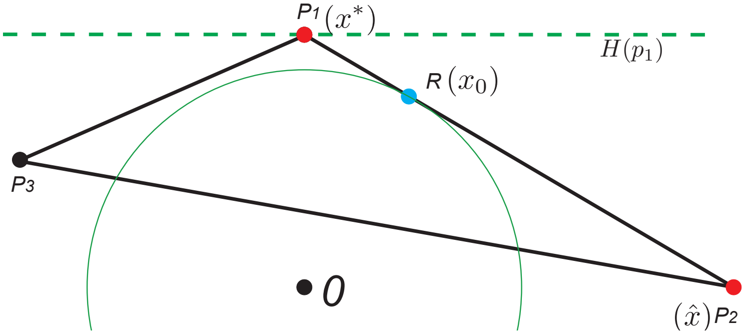
p_1 is the extreme point closest to 0 and so we choose it first, although we can choose any arbitrary extreme point as the initial point. We set $x^* \leftarrow p_1$ in Line 1, and $Q \leftarrow \{p_1\}$ in Line 2. $H(x^*) = H(p_1)$ (green dashed line) is not a supporting hyperplane of $\text{conv}(P)$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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



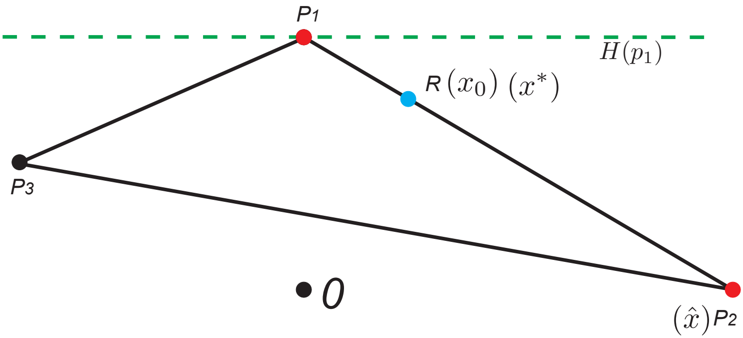
We need to add some extreme point $\hat{x}$ on the "near" side of $H(p_1)$ in Line 6, we choose $\hat{x} = p_2$. In Line 7, we set $Q \leftarrow Q \cup \{p_2\}$, so $Q = \{p_1, p_2\}$.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



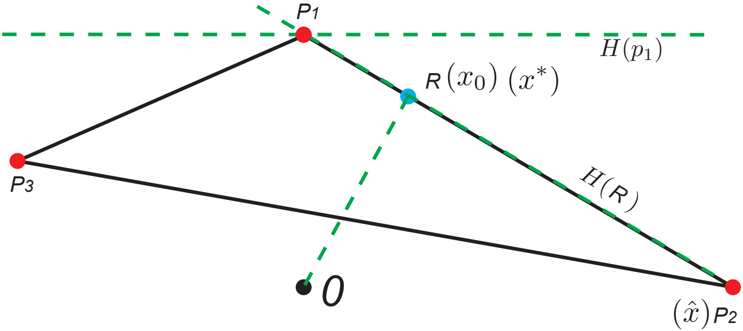
$x_0 = R$ is the min-norm point in $\mathrm{aff}\,\{p_1, p_2\}$ computed in Line 9.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



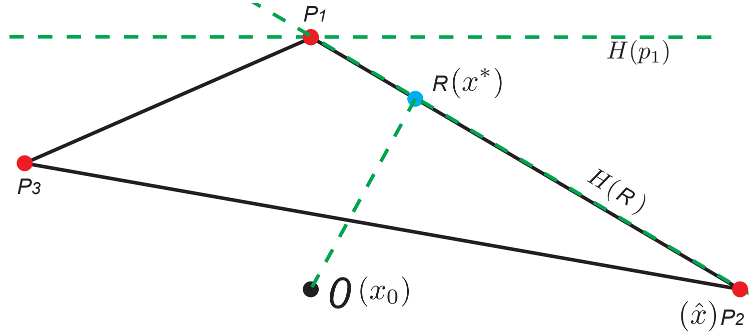
$x_0 = R$ is the min-norm point in $\text{aff}\{p_1, p_2\}$ computed in Line 9. Also, with $Q = \{p_1, p_2\}$, since $R \in \text{conv}\, Q$, we set $x^* \leftarrow x_0 = R$ in Line 11, not violating the invariant $x^* \in \text{conv}\, Q$. Note, after Line 11, we still have $x^* \in \text{conv}\, P$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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



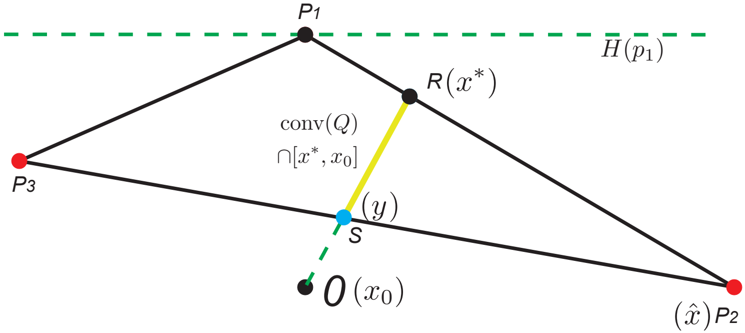
$R = x_0 = x^*$. We consider next $H(R) = H(x^*)$ in Line 4. $H(x^*)$ is not a supporting hyperplane of $\text{conv}\, P$. So we choose p_3 on the "near" side of $H(x^*)$ in Line 6. Add $Q \leftarrow Q \cup \{p_3\}$ in Line 7. Now $Q = P = \{p_1, p_2, p_3\}$.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



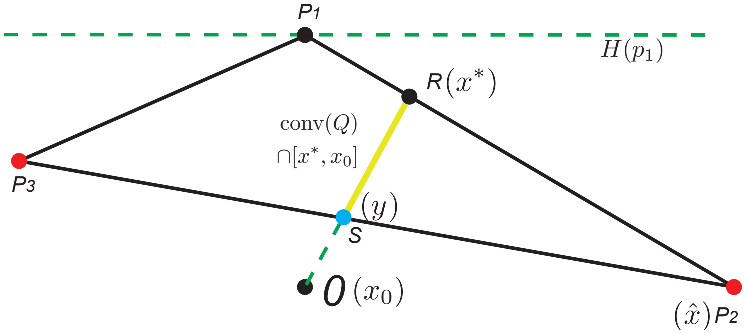
$R = x_0 = x^*$. We consider next $H(R) = H(x^*)$ in Line 4. $H(x^*)$ is not a supporting hyperplane of $\operatorname{conv} P$. So we choose p_3 on the "near" side of $H(x^*)$ in Line 6. Add $Q \leftarrow Q \cup \{p_3\}$ in Line 7. Now $Q = P = \{p_1, p_2, p_3\}$. The origin $x_0 = 0$ is the min-norm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



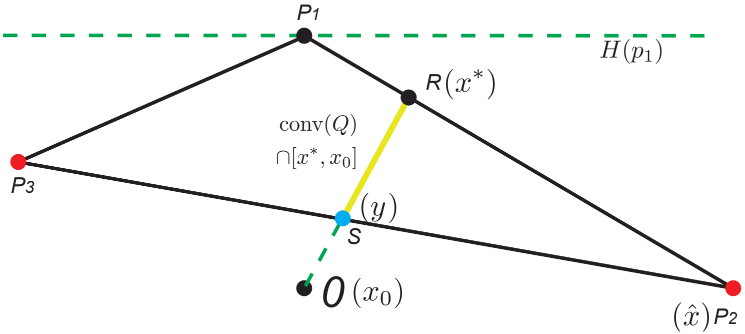
$Q = P = \{p_1, p_2, p_3\}$. Line 14: $S = y = \operatorname{argmin}_{x \in \operatorname{conv} Q \cap [x^*, x_0]} \|x - x_0\|_2$ where x_0 is 0 and x^* is R here. Thus, y lies on the boundary of $\operatorname{conv} Q$. Note, $\|y\|_2 < \|x^*\|_2$ since $x^* \in \operatorname{conv} Q$, $\|x_0\|_2 <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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



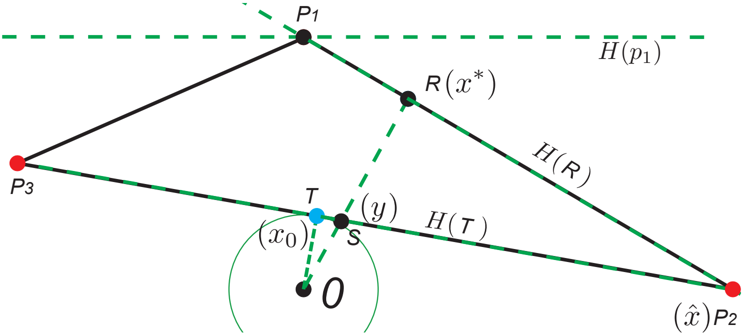
$Q = P = \{p_1, p_2, p_3\}$. Line 14: $S = y = \mathrm{argmin}_{x \in \mathrm{conv}\, Q \cap [x^*, x_0]} \|x - x_0\|_2$ where x_0 is 0 and x^* is R here. Thus, y lies on the boundary of $\mathrm{conv}\, Q$. Note, $\|y\|_2 < \|x^*\|_2$ since $x^* \in \mathrm{conv}\, Q$, $\|x_0\|_2 < \|x^*\|_2$.
Line 15: 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.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



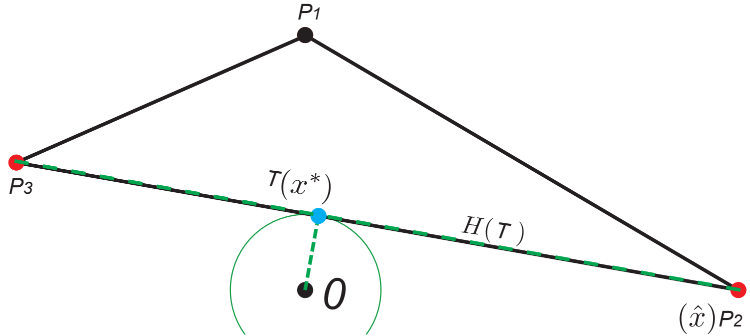
$Q = P = \{p_1, p_2, p_3\}$. Line 14: $S = y = \operatorname{argmin}_{x \in \mathrm{conv}\, Q \cap [x^*, x_0]} \|x - x_0\|_2$ where x_0 is 0 and x^* is R here. Thus, y lies on the boundary of $\mathrm{conv}\, Q$. Note, $\|y\|_2 < \|x^*\|_2$ since $x^* \in \mathrm{conv}\, Q$, $\|x_0\|_2 < \|x^*\|_2$.
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. Line 16: $x^* \leftarrow y$, retain invariant $x^* \in \mathrm{conv}\, Q$, and again have $\|x^*\|_2 = \|x^*\|_2 < \|x^*\|$ 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 20.5.1

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

$$p_i^\intercal x^* \geq \|x^*\|_2^2 \quad \forall i = 1, \cdots, m. \tag{20.31}$$

Proof.

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

□

Condition for Min-Norm Point

Theorem 20.5.1

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

$$p_i^\intercal x^* \geq \|x^*\|_2^2 \quad \forall i = 1, \cdots, m. \tag{20.31}$$

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

$$\|z\|_2^2 = \|x^* + \theta(y - x^*)\|_2^2 \tag{20.32}$$

$$= \|x^*\|_2^2 + 2\theta(x^{*\intercal} y - x^{*\intercal} x^*) + \theta^2 \|y - x^*\|_2^2 \tag{20.33}$$

□

Condition for Min-Norm Point

Theorem 20.5.1

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

$$p_i^{\mathsf{T}} x^* \geq \|x^*\|_2^2 \quad \forall i = 1, \cdots, m. \tag{20.31}$$

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

$$\|z\|_2^2 = \|x^* + \theta(y - x^*)\|_2^2 \tag{20.32}$$

$$= \|x^*\|_2^2 + 2\theta({x^*}^{\mathsf{T}} y - {x^*}^{\mathsf{T}} x^*) + \theta^2 \|y - x^*\|_2^2 \tag{20.33}$$

- It is possible for $\|z\|_2^2 < \|x^*\|_2^2$ for small θ, unless ${x^*}^{\mathsf{T}} y \geq {x^*}^{\mathsf{T}} x^*$ for all $y \in \operatorname{conv} P \Rightarrow$ Equation (20.31).

□

Condition for Min-Norm Point

Theorem 20.5.1

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

$$p_i^\intercal x^* \geq \|x^*\|_2^2 \quad \forall i = 1, \cdots, m. \tag{20.31}$$

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

$$\|z\|_2^2 = \|x^* + \theta(y - x^*)\|_2^2 \tag{20.32}$$

$$= \|x^*\|_2^2 + 2\theta({x^*}^\intercal y - {x^*}^\intercal x^*) + \theta^2 \|y - x^*\|_2^2 \tag{20.33}$$

- 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 (20.31).
- Conversely, given Eq (20.31), and given that $y = \sum_i \lambda_i p_i \in \operatorname{conv} P$,

$$y^\intercal x^* = \sum_i \lambda_i p_i^\intercal x^* \geq \sum_i \lambda_i {x^*}^\intercal x^* = {x^*}^\intercal x^* \tag{20.34}$$

implying that $\|z\|_2^2 > \|x^*\|_2^2$ in Equation 20.33 for arbitrary $z \in \operatorname{conv} P$. □

The set Q is always affinely independent

Lemma 20.5.2

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

The set Q is always affinely independent

Lemma 20.5.2

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

Proof.

□

The set Q is always affinely independent

Lemma 20.5.2

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 20.5.2

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 20.5.2

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 20.5.2

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 $\operatorname{aff} Q \subseteq H(x^*)$.

□

The set Q is always affinely independent

Lemma 20.5.2

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 20.5.2

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. □

The set Q is always affinely independent

Lemma 20.5.2

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. □

Thus, by Lemma 20.5.2, 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 20.5.3

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

The set Q is never too large

Lemma 20.5.3

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

Proof.

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

Minimum Norm in an affine set

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

Minimum Norm in an affine set

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

$$\text{minimize} \qquad \|x\|_2^2 = w^\intercal Q^\intercal Q w \tag{20.35}$$

$$\text{subject to} \qquad \mathbf{1}^\intercal w = 1 \tag{20.36}$$

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 $x = Qw$:

$$\text{minimize} \quad \|x\|_2^2 = w^\intercal Q^\intercal Q w \tag{20.35}$$

$$\text{subject to} \quad \mathbf{1}^\intercal w = 1 \tag{20.36}$$

- 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:

$$\mathbf{1}^\intercal w = 1 \tag{20.37}$$

$$Q^\intercal Q w + \lambda \mathbf{1} = 0 \tag{20.38}$$

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 $x = Qw$:

$$\text{minimize} \qquad \|x\|_2^2 = w^\intercal Q^\intercal Q w \tag{20.35}$$

$$\text{subject to} \qquad \mathbf{1}^\intercal w = 1 \tag{20.36}$$

- 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:

$$\mathbf{1}^\intercal w = 1 \tag{20.37}$$

$$Q^\intercal Q w + \lambda \mathbf{1} = 0 \tag{20.38}$$

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

$$\begin{bmatrix} 0 & \mathbf{1}^\intercal \\ \mathbf{1} & Q^\intercal Q \end{bmatrix} \begin{bmatrix} \lambda \\ w \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \tag{20.39}$$

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 $x = Qw$:

$$\text{minimize} \qquad \|x\|_2^2 = w^\intercal Q^\intercal Q w \tag{20.35}$$

$$\text{subject to} \qquad \mathbf{1}^\intercal w = 1 \tag{20.36}$$

- 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:

$$\mathbf{1}^\intercal w = 1 \tag{20.37}$$

$$Q^\intercal Q w + \lambda \mathbf{1} = 0 \tag{20.38}$$

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

$$\begin{bmatrix} 0 & \mathbf{1}^\intercal \\ \mathbf{1} & Q^\intercal Q \end{bmatrix} \begin{bmatrix} \lambda \\ w \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \tag{20.39}$$

- 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 20.5.4

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 20.5.4

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 20.5.4

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 20.5.4

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 20.5.4 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 20.5.4 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 20.5.4 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.

. . .

MN Algorithm finds the MN point in finite time.

. . . proof of Theorem 20.5.4 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 20.5.4 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 20.5.4 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 20.5.4 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 20.5.4 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

$$\begin{aligned}\|x^* + \theta(\hat{x} - x^*)\|_2^2 &= \|x^*\|_2^2 + 2\theta\left((x^*)^\top \hat{x} - \|x^*\|_2^2\right) + \theta^2 \|\hat{x} - x^*\|_2^2 \\ &\geq \|x_0\|_2^2 \end{aligned} \tag{20.40}$$

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 20.5.4 continued.

- Therefore, for sufficiently small θ, specifically for

$$\theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2} \tag{20.41}$$

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

□

MN Algorithm finds the MN point in finite time.

. . . proof of Theorem 20.5.4 continued.

- Therefore, for sufficiently small θ, specifically for

$$\theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2} \tag{20.41}$$

  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 20.5.4 continued.

- Therefore, for sufficiently small θ, specifically for

$$\theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2} \tag{20.41}$$

  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. 20.40, reduction on norm is lower-bounded:

$$\Delta = \|x^*\|_2^2 - \|x_0\|_2^2 \geq 2\theta \left(\|x^*\|_2^2 - (x^*)^\top \hat{x} \right) - \theta^2 \|\hat{x} - x^*\|_2^2 \triangleq \underline{\Delta} \tag{20.42}$$

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. 20.40, reduction on norm is lower-bounded:

$$\Delta = \|x^*\|_2^2 - \|x_0\|_2^2 \geq 2\theta\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right) - \theta^2 \|\hat{x} - x^*\|_2^2 \triangleq \underline{\Delta} \tag{20.42}$$

- 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:

$$\max_{0 \leq \theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2}} \underline{\Delta} = \left(\frac{\|x^*\|_2^2 - (x^*)^\top \hat{x}}{\|\hat{x} - x^*\|_2}\right)^2 \tag{20.43}$$

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 20.43) 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 20.43) is maximized.
- That is, we want to find

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \left(\frac{\|x^*\|_2^2 - (x^*)^\top x}{\|x - x^*\|_2} \right)^2 \tag{20.44}$$

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

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \left(\frac{\|x^*\|_2^2 - (x^*)^\top x}{\|x - x^*\|_2} \right)^2 \tag{20.44}$$

  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. 20.44, i.e., find

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \|x^*\|_2^2 - (x^*)^\top x = \operatorname*{argmin}_{x \in P} (x^*)^\top x, \qquad (20.45)$$

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

- As a surrogate, we maximize numerator in Eqn. 20.44, i.e., find

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \|x^*\|_2^2 - (x^*)^\top x = \operatorname*{argmin}_{x \in P} (x^*)^\top x, \tag{20.45}$$

- 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. 20.44, i.e., find

$$\hat{x} \in \underset{x \in P}{\operatorname{argmax}} \|x^*\|_2^2 - (x^*)^\top x = \underset{x \in P}{\operatorname{argmin}} (x^*)^\top x, \tag{20.45}$$

- 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. 20.44, i.e., find

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \|x^*\|_2^2 - (x^*)^\top x = \operatorname*{argmin}_{x \in P} (x^*)^\top x, \tag{20.45}$$

- 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 $\mathrm{conv}\, P$ 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 $\mathrm{conv}\, P$ from the origin.
- Mathematically and theoretically, we terminate the algorithm if

$$(x^*)^\top \hat{x} \geq \|x^*\|_2^2, \tag{20.46}$$

where $\hat{x}$ is the solution of Eq. 20.45.

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

$$(x^*)^\top \hat{x} > \|x^*\|_2^2 - \epsilon \max_{x \in Q} \|x\|_2^2 \qquad (20.47)$$

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

$$(x^*)^\top \hat{x} > \|x^*\|_2^2 - \epsilon \max_{x \in Q} \|x\|_2^2 \tag{20.47}$$

- 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 20.45 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

$$(x^*)^\top \hat{x} > \|x^*\|_2^2 - \epsilon \max_{x \in Q} \|x\|_2^2 \tag{20.47}$$

- 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 20.45 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

$$(x^*)^\top \hat{x} > \|x^*\|_2^2 - \epsilon \max_{x \in Q} \|x\|_2^2 \tag{20.47}$$

- 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 20.45 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.

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.

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 $O(n^5T + n^6)$ (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:

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

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 $O(n^5T + n^6)$ (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.

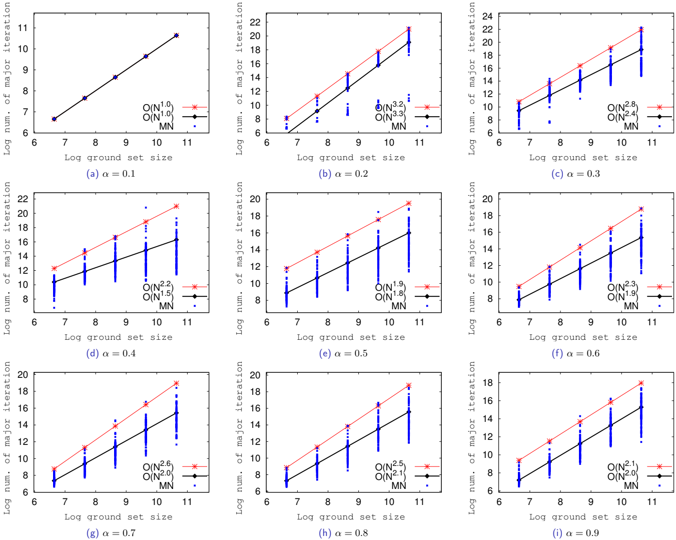
MN Algorithm Complexity

- 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



(a) $\alpha = 0.1$ (b) $\alpha = 0.2$ (c) $\alpha = 0.3$

(d) $\alpha = 0.4$ (e) $\alpha = 0.5$ (f) $\alpha = 0.6$

(g) $\alpha = 0.7$ (h) $\alpha = 0.8$ (i) $\alpha = 0.9$

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 Fujishige-Wolfe min-norm procedure has not yet been established.

MN Algorithm Complexity

- A lower bound complexity of the Fujishige-Wolfe min-norm procedure has not yet 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 Fujishige-Wolfe min-norm procedure has not yet 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 Fujishige-Wolfe min-norm procedure has not yet 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.
- In 2020, in De Loera et. al. "The Minimum Euclidean-Norm Point in a Convex Polytope: Wolfe's Combinatorial Algorithm is Exponential", 2020, SIAM J. Computing, gave an example where the Wolfe procedure can run in exponential time, although this is not for the submodular polytope B_f that applies here, this is left as an open question. Hence, the lower bound complexity of the Fujishige-Wolfe procedure is still unknown.

Frank-Wolfe vs. Fujishige-Wolfe

Another algorithm we could use to find the min-norm is M. Frank & P. Wolfe "An algorithm for quadratic programming", 1956 (conditional gradient descent) for constrained convex minimization of convex function $f : \mathcal{D} \to \mathbb{R}$.

```
Input  : Convex set $\mathcal{D} \subseteq \mathbb{R}^n$, convex $f : \mathcal{D} \to \mathbb{R}$, $x_0 \in \mathcal{D}$, $\tau > 0$
Output: $x^* \in \mathcal{D}$, the minimizer of $f$ on $\mathcal{D}$.
1 $k \leftarrow 0$ and start with $x_0 \in \mathcal{D}$ ;
2 Let $s_k$ solve $\min \langle s, \nabla f(x_k) \rangle$ s.t. $s \in \mathcal{D}$ ;
3 Let $\lambda_k \in [0, 1]$ minimize $f(\lambda s_k + (1 - \lambda) x_k)$ ;
4 $x_{k+1} \leftarrow \lambda_k s_k + (1 - \lambda_k) x_k$, $k \leftarrow k + 1$ ;
5 Goto line 2 if $\|x_{k+1} - x_k\| > \tau$ ;
6 $x^* \leftarrow x_{k+1}$
```

can be done using Edmonds

For min norm point we find

$\min \|x\|_2^2$

s.t. $x \in B_f$

- Above can also be used minimize Lovász extension, primal approach to SFM.
- The Frank-Wolfe and Fujishige-Wolfe are distinct procedures although Wolfe is the same person.

Other algorithms for approximate and/or pseudo-polynomial SFM

- In 2015, Lee, Sidford, and Wong, gave pseudo-poly algorithms for SFM that run in $O(n^2 \log nM\text{EO} + n^2 \log^{O(1)} nM)$ time nad $O(n^3 \log^2 n\text{EO} + n^4 \log^{O(1)} n)$ time respectively.

Other algorithms for approximate and/or pseudo-polynomial SFM

- In 2015, Lee, Sidford, and Wong, gave pseudo-poly algorithms for SFM that run in $O(n^2 \log nM\text{EO} + n^2 \log^{O(1)} nM)$ time nad $O(n^3 \log^2 n\text{EO} + n^4 \log^{O(1)} n)$ time respectively.
- In 2017, in Chakrabarty, Lee, Sidford, and Wong "Subquadratic Submodular Function Minimization", this was improved. I.e., for real-valued submodular functions, it runs in $\tilde{O}(n^{5/3}\text{EO}/\epsilon^3)$ giving an ϵ-additive approximate solution.

Other algorithms for approximate and/or pseudo-polynomial SFM

- In 2015, Lee, Sidford, and Wong, gave pseudo-poly algorithms for SFM that run in $O(n^2 \log nM\text{EO} + n^2 \log^{O(1)} nM)$ time nad $O(n^3 \log^2 n\text{EO} + n^4 \log^{O(1)} n)$ time respectively.
- In 2017, in Chakrabarty, Lee, Sidford, and Wong "Subquadratic Submodular Function Minimization", this was improved. I.e., for real-valued submodular functions, it runs in $\tilde{O}(n^{5/3}\text{EO}/\epsilon^3)$ giving an ϵ-additive approximate solution.
- In 2020, Axelrod, Liu, and Sidford "Near-optimal Approximate Discrete and Continuous Submodular Function Minimization" give a randomized algorithm that for a submodular function in the range $[-1, 1]$ runs in $\tilde{O}(n/\epsilon^2)$ for an ϵ-additive approximation to SFM. This can also be used to approximately minimize smooth DR-submodular (and not necessarily convex) functions.

Other algorithms for approximate and/or pseudo-polynomial SFM

- In 2015, Lee, Sidford, and Wong, gave pseudo-poly algorithms for SFM that run in $O(n^2 \log nM\text{EO} + n^2 \log^{O(1)} nM)$ time nad $O(n^3 \log^2 n\text{EO} + n^4 \log^{O(1)} n)$ time respectively.
- In 2017, in Chakrabarty, Lee, Sidford, and Wong "Subquadratic Submodular Function Minimization", this was improved. I.e., for real-valued submodular functions, it runs in $\tilde{O}(n^{5/3}\text{EO}/\epsilon^3)$ giving an ϵ-additive approximate solution.
- In 2020, Axelrod, Liu, and Sidford "Near-optimal Approximate Discrete and Continuous Submodular Function Minimization" give a randomized algorithm that for a submodular function in the range $[-1, 1]$ runs in $\tilde{O}(n/\epsilon^2)$ for an ϵ-additive approximation to SFM. This can also be used to approximately minimize smooth DR-submodular (and not necessarily convex) functions.
- In 2020, Balkanski and Singer, "A Lower Bound for Parallel Submodular Minimization", give "adaptivity lower bounds" (see the paper for what this is) for the parallel complexity of SFM.

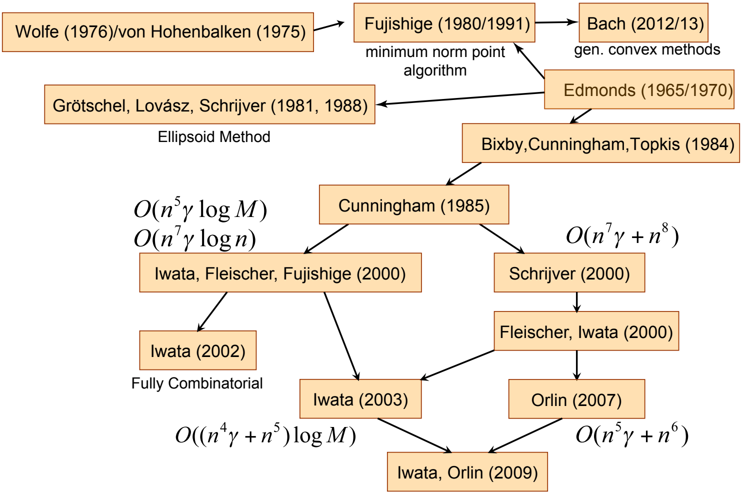
Other algorithms for approximate and/or pseudo-polynomial SFM

- In 2015, Lee, Sidford, and Wong, gave pseudo-poly algorithms for SFM that run in $O(n^2 \log nM\text{EO} + n^2 \log^{O(1)} nM)$ time nad $O(n^3 \log^2 n\text{EO} + n^4 \log^{O(1)} n)$ time respectively.
- In 2017, in Chakrabarty, Lee, Sidford, and Wong "Subquadratic Submodular Function Minimization", this was improved. I.e., for real-valued submodular functions, it runs in $\tilde{O}(n^{5/3}\text{EO}/\epsilon^3)$ giving an ϵ-additive approximate solution.
- In 2020, Axelrod, Liu, and Sidford "Near-optimal Approximate Discrete and Continuous Submodular Function Minimization" give a randomized algorithm that for a submodular function in the range $[-1, 1]$ runs in $\tilde{O}(n/\epsilon^2)$ for an ϵ-additive approximation to SFM. This can also be used to approximately minimize smooth DR-submodular (and not necessarily convex) functions.
- In 2020, Balkanski and Singer, "A Lower Bound for Parallel Submodular Minimization", give "adaptivity lower bounds" (see the paper for what this is) for the parallel complexity of SFM.

SFM Summary

modified from S. Iwata's slides

General Submodular Function Minimization



Wolfe (1976)/von Hohenbalken (1975)

Fujishige (1980/1991)

minimum norm point algorithm

Bach (2012/13)

gen. convex methods

Edmonds (1965/1970)

Grötschel, Lovász, Schrijver (1981, 1988)

Ellipsoid Method

Bixby,Cunningham,Topkis (1984)

Cunningham (1985)

$O(n^5\gamma \log M)$

$O(n^7\gamma \log n)$

Iwata, Fleischer, Fujishige (2000)

$O(n^7\gamma + n^8)$

Schrijver (2000)

Fleischer, Iwata (2000)

Iwata (2002)

Fully Combinatorial

Iwata (2003)

Orlin (2007)

$O((n^4\gamma + n^5)\log M)$

$O(n^5\gamma + n^6)$

Iwata, Orlin (2009)

Recent SFM Strongly Polynomial Summary

Table taken from Haotian Jiang's 2020 paper

Authors	Year	Oracle Complexity	Remarks
Grötschel, Lovász, Schrijver [GLS81, GLS88]	1981,88	$O(n^5)$ [McC05]	first strongly
Schrijver [Sch00]	2000	$O(n^8)$	first comb. strongly
Iwata, Fleischer, Fujishige [IFF01]	2000	$O(n^7 \log(n))$	first comb. strongly
Fleischer, Iwata [FI03]	2000	$O(n^7)$	
Iwata [Iwa03]	2002	$O(n^6 \log(n))$	
Vygen [Vyg03]	2003	$O(n^7)$	
Orlin [Orl09]	2007	$O(n^5)$	
Iwata, Orlin [IO09]	2009	$O(n^5 \log(n))$	
Lee, Sidford, Wong [LSW15]	2015	$O(n^3 \log^2(n))$	~~current best~~ strongly
Lee, Sidford, Wong [LSW15]	2015	$O(n^3 \log(n))$	exponential time
Dadush, Végh, Zambelli [DVZ18]	2018	$O(n^3 \log^2(n))$	close to best
Haotain Jiang	2020	$\boxed{O(n^3)}$	currently best strongly

Submodularity

This is only the beginning. Submodularity is still gaining in popularity in machine learning and data science, it has both a rich and long past and a promising future.