Submodular Functions, Optimization, and Applications to Machine Learning

— Spring Quarter, Lecture 17 —

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

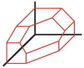
Prof. Jeff Bilmes

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

May 25th, 2016

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

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



Cumulative Outstanding Reading

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

Announcements, Assignments, and Reminders

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

Class Road Map - IT-I

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

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

Submodular Function Minimization (SFM) and Min-Norm

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

Min-Norm Point: Definition

- Consider the optimization:

$$\text{minimize} \quad \|x\|_2^2 \tag{17.1a}$$

$$\text{subject to} \quad x \in B_f \tag{17.1b}$$

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

Min-Norm Point: Definition

- Consider the optimization:

$$\text{minimize} \quad \|x\|_2^2 \tag{17.1a}$$

$$\text{subject to} \quad x \in B_f \tag{17.1b}$$

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

- Note, x^* is the unique optimal solution since we have a strictly convex objective over a set of convex constraints.

Min-Norm Point: Definition

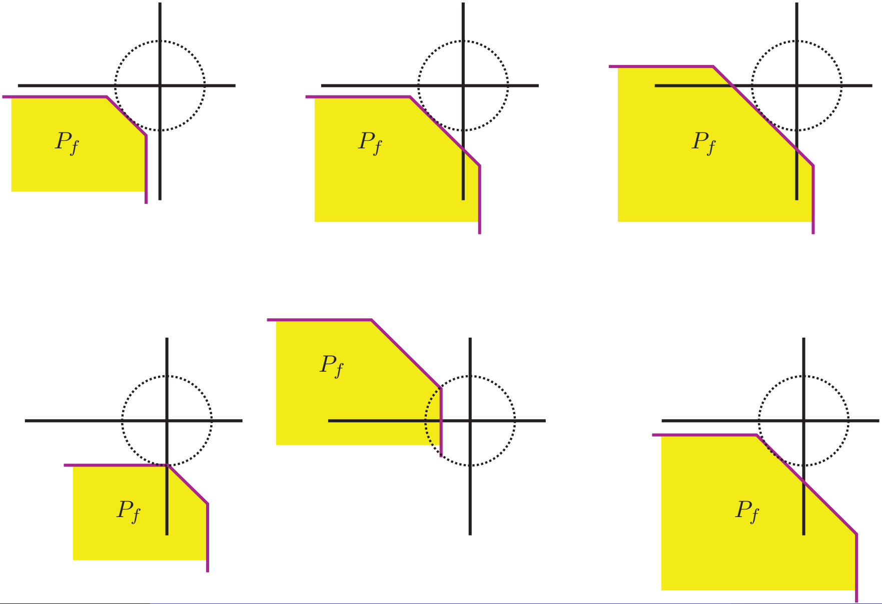
- Consider the optimization:

$$\begin{aligned} &\text{minimize} && \|x\|_2^2 && (17.1a) \\ &\text{subject to} && x \in B_f && (17.1b) \end{aligned}$$

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

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

Min-Norm Point: Examples

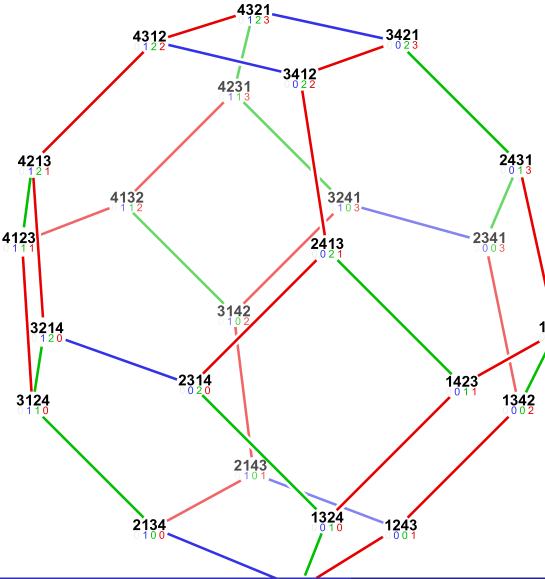


Ex: 3D base B_f: permutahedron

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

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

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



Min-Norm Point and Submodular Function Minimization

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

$$y^* = x^* \wedge 0 = (\min(x^*(e), 0) | e \in E) \tag{17.2}$$

$$A_- = \{e : x^*(e) < 0\} \tag{17.3}$$

$$A_0 = \{e : x^*(e) \leq 0\} \tag{17.4}$$

Min-Norm Point and Submodular Function Minimization

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

$$y^* = x^* \wedge 0 = (\min(x^*(e), 0) | e \in E) \tag{17.2}$$

$$A_- = \{e : x^*(e) < 0\} \tag{17.3}$$

$$A_0 = \{e : x^*(e) \leq 0\} \tag{17.4}$$

- Thus, we immediately have that:

$$A_- \subseteq A_0 \tag{17.5}$$

and that

$$x^*(A_-) = x^*(A_0) = y^*(A_-) = y^*(A_0) \tag{17.6}$$

Min-Norm Point and Submodular Function Minimization

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

$$y^* = x^* \wedge 0 = (\min(x^*(e), 0) | e \in E) \tag{17.2}$$
$$A_- = \{e : x^*(e) < 0\} \tag{17.3}$$
$$A_0 = \{e : x^*(e) \leq 0\} \tag{17.4}$$

- Thus, we immediately have that:

$$A_- \subseteq A_0 \tag{17.5}$$

and that

$$x^*(A_-) = x^*(A_0) = y^*(A_-) = y^*(A_0) \tag{17.6}$$

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

Min-Norm Point and Submodular Function Minimization

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

$$y^* = x^* \wedge 0 = (\min(x^*(e), 0) | e \in E) \tag{17.2}$$
$$A_- = \{e : x^*(e) < 0\} \tag{17.3}$$
$$A_0 = \{e : x^*(e) \leq 0\} \tag{17.4}$$

- Thus, we immediately have that:

$$A_- \subseteq A_0 \tag{17.5}$$

and that

$$x^*(A_-) = x^*(A_0) = y^*(A_-) = y^*(A_0) \tag{17.6}$$

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

More about the base B_f

Theorem 17.4.1

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

More about the base B_f

Theorem 17.4.1

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

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

Theorem 17.4.2

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

More about the base B_f

Theorem 17.4.1

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

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

Theorem 17.4.2

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

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

Review from Lecture 12

The follow slide repeates Theorem 12.5.2 from lecture 12 and which is also essentially the same as Theorem 13.4.2 from lecture 13, and is one of the most important theorems in submodular theory.

A polymatroid function's polyhedron is a polymatroid.

Theorem 17.4.1

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

$$\textit{rank}(x) = \max\left(y(E) : y \leq x, y \in P_f\right) = \min\left(x(A) + f(E \setminus A) : A \subseteq E\right) \tag{17.1}$$

Essentially the same theorem as Theorem 11.4.1, but note P_f rather than P_f^+. Taking $x = 0$ we get:

Corollary 17.4.2

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

$$\textit{rank}(0) = \max\left(y(E) : y \leq 0, y \in P_f\right) = \min\left(f(A) : A \subseteq E\right) \tag{17.2}$$

Modified max-min theorem

- Min-max theorem (Thm 12.5.2) restated for $x = 0$.

$$\max\{y(E) | y \in P_f, y \leq 0\} = \min\{f(X) | X \subseteq V\} \tag{17.7}$$

Modified max-min theorem

- Min-max theorem (Thm 12.5.2) restated for $x = 0$.

$$\max\{y(E) | y \in P_f, y \leq 0\} = \min\{f(X) | X \subseteq V\} \tag{17.7}$$

Theorem 17.4.3 (Edmonds-1970)

$$\min\{f(X) | X \subseteq E\} = \max\left\{x^-(E) | x \in B_f\right\} \tag{17.8}$$

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

Modified max-min theorem

- Min-max theorem (Thm 12.5.2) restated for $x = 0$.

$$\max\{y(E) | y \in P_f, y \leq 0\} = \min\{f(X) | X \subseteq V\} \tag{17.7}$$

Theorem 17.4.3 (Edmonds-1970)

$$\min\{f(X) | X \subseteq E\} = \max\left\{x^-(E) | x \in B_f\right\} \tag{17.8}$$

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

Proof via the Lovász ext.

$$\min\{f(X) | X \subseteq E\} = \min_{w \in [0,1]^E} \tilde{f}(w) = \min_{w \in [0,1]^E} \max_{x \in P_f} w^\intercal x \tag{17.9}$$

$$= \min_{w \in [0,1]^E} \max_{x \in B_f} w^\intercal x \tag{17.10}$$

$$= \max_{x \in B_f} \min_{w \in [0,1]^E} w^\intercal x \tag{17.11}$$

$$= \max_{x \in B_f} x^-(E) \tag{17.12}$$

Alternate proof of modified max-min theorem

We start directly from Theorem 12.5.2.

$$\max\left(y(E) : y \leq 0, y \in P_f\right) = \min\left(f(A) : A \subseteq E\right) \tag{17.16}$$

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

$$\begin{aligned}
\max\left(y(E) : y \leq 0, y \in P_f\right) &= \max\left(y^-(E) : y \leq 0, y \in P_f\right) && (17.17)\\
&= \max\left(y^-(E) : y \in P_f\right) && (17.18)\\
&= \max\left(y^-(E) : y \in B_f\right) && (17.19)
\end{aligned}$$

The first equality follows since $y \leq 0$. For the second equality will be shown on the following slide. The third equality follows since for any $x \in P_f$ there exists a $y \in B_f$ with $x \leq y$ (follows from Theorem 17.4.2).

Alternate proof of modified max-min theorem

Consider the following two problems:

$$\max \sum_{e \in E} y(e) \tag{17.20a}$$
$$\text{s.t. } y \leq x \tag{17.20b}$$
$$y \in P \tag{17.20c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{17.21a}$$
$$\text{s.t. } y \in P \tag{17.21b}$$

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

$$\sum_{e \in E} \min(y_1^*(e), x(e)) < \sum_{e \in E} \min(y_2^*(e), x(e)) \tag{17.22}$$

Alternate proof of modified max-min theorem

Consider the following two problems:

$$\max \sum_{e \in E} y(e) \tag{17.20a}$$
$$\text{s.t. } y \le x \tag{17.20b}$$
$$y \in P \tag{17.20c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{17.21a}$$
$$\text{s.t. } y \in P \tag{17.21b}$$

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

$$\sum_{e \in E} \min(y_1^*(e), x(e)) < \sum_{e \in E} \min(y_2^*(e), x(e)) \tag{17.22}$$

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

Alternate proof of modified max-min theorem

Consider the following two problems:

$$\max \sum_{e \in E} y(e) \tag{17.20a}$$
$$\text{s.t. } y \le x \tag{17.20b}$$
$$y \in P \tag{17.20c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{17.21a}$$
$$\text{s.t. } y \in P \tag{17.21b}$$

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

$$\sum_{e \in E} \min(y_1^*(e), x(e)) < \sum_{e \in E} \min(y_2^*(e), x(e)) \tag{17.22}$$

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

Alternate proof of modified max-min theorem

Consider the following two problems:

$$\max \sum_{e \in E} y(e) \tag{17.20a}$$
$$\text{s.t. } y \leq x \tag{17.20b}$$
$$y \in P \tag{17.20c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{17.21a}$$
$$\text{s.t. } y \in P \tag{17.21b}$$

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

$$\sum_{e \in E} y_2^*(e) < \sum_{e \in E} y_1^*(e) \tag{17.22}$$

Alternate proof of modified max-min theorem

Consider the following two problems:

$$\max \sum_{e \in E} y(e) \tag{17.20a}$$
$$\text{s.t. } y \le x \tag{17.20b}$$
$$y \in P \tag{17.20c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{17.21a}$$
$$\text{s.t. } y \in P \tag{17.21b}$$

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

$$\sum_{e \in E} y_2^*(e) < \sum_{e \in E} y_1^*(e) \tag{17.22}$$

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

Alternate proof of modified max-min theorem

Consider the following two problems:

$$\max \sum_{e \in E} y(e) \tag{17.20a}$$
$$\text{s.t. } y \leq x \tag{17.20b}$$
$$y \in P \tag{17.20c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{17.21a}$$
$$\text{s.t. } y \in P \tag{17.21b}$$

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

$$\sum_{e \in E} y_2^*(e) < \sum_{e \in E} y_1^*(e) \tag{17.22}$$

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

Alternate proof of modified max-min theorem

Consider the following two problems:

$$\max \sum_{e \in E} y(e) \tag{17.20a}$$
$$\text{s.t. } y \le x \tag{17.20b}$$
$$y \in P \tag{17.20c}$$

$$\max \sum_{e \in E} \min(y(e), x(e)) \tag{17.21a}$$
$$\text{s.t. } y \in P \tag{17.21b}$$

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

$$\sum_{e \in E} y_2^*(e) < \sum_{e \in E} y_1^*(e) \tag{17.22}$$

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

$$\max\left(y^-(E) : y \in B_f\right) = \max\left(y(E) : y \le 0, y \in P_f\right) \tag{17.23}$$

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

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

$$\max \{w^\intercal x | x \in P_f\} = \max \{w^\intercal x | x \in B_f\} \tag{17.24}$$

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

$$\max \{w^\intercal x | x \in P_f\} = \max \{w^\intercal x | x \in B_f\} \tag{17.24}$$

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

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

$$\max \{w^\intercal x | x \in B_f\} \tag{17.25}$$

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

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

$$\max \{w^\intercal x | x \in P_f\} = \max \{w^\intercal x | x \in B_f\} \tag{17.24}$$

  since for all $x \in P_f$, there exists $y \geq x$ with $y \in B_f$.
- For arbitrary $w \in \mathbb{R}^E$, greedy algorithm will also solve:

$$\max \{w^\intercal x | x \in B_f\} \tag{17.25}$$

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

$$\min \{w^\intercal x | x \in B_f\} = -\max \{-w^\intercal x | x \in B_f\} \tag{17.26}$$

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

$$w(e_1) \leq w(e_2) \leq \cdots \leq w(e_m) \tag{17.27}$$

  will solve l.h.s. of Equation (17.26).

$\max\{w^\intercal x | x \in B_f\}$ 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 12, 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.

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.,

$$\operatorname{conv} P \triangleq \left\{ \sum_{i=1}^{k} \lambda_i p_i : \sum_i \lambda_i = 1,\ \lambda_i \geq 0, i \in [k] \right\}. \tag{17.28}$$

Convex and affine hulls, affinely independent

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

$$\text{conv}\, P \triangleq \left\{ \sum_{i=1}^{k} \lambda_i p_i : \sum_i \lambda_i = 1,\ \lambda_i \geq 0, i \in [k] \right\}. \tag{17.28}$$

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

$$\text{aff}\, Q \triangleq \left\{ \sum_{i \in 1}^{k} \lambda_i q_i : \sum_{i=1}^{k} \lambda_i = 1 \right\} \tag{17.29}$$

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.,

$$\operatorname{conv} P \triangleq \left\{ \sum_{i=1}^{k} \lambda_i p_i : \sum_i \lambda_i = 1,\ \lambda_i \geq 0, i \in [k] \right\}. \tag{17.28}$$

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

$$\operatorname{aff} Q \triangleq \left\{ \sum_{i \in 1}^{k} \lambda_i q_i : \sum_{i=1}^{k} \lambda_i = 1 \right\} \supseteq \operatorname{conv} Q. \tag{17.29}$$

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.,

$$\operatorname{conv} P \triangleq \left\{ \sum_{i=1}^{k} \lambda_i p_i : \sum_i \lambda_i = 1,\ \lambda_i \geq 0, i \in [k] \right\}. \tag{17.28}$$

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

$$\operatorname{aff} Q \triangleq \left\{ \sum_{i \in 1}^{k} \lambda_i q_i : \sum_{i=1}^{k} \lambda_i = 1 \right\} \supseteq \operatorname{conv} Q. \tag{17.29}$$

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

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

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

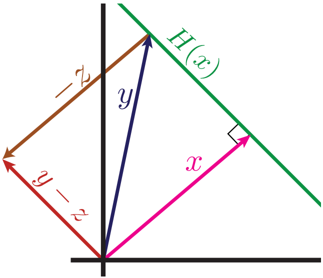
$$H(x) \triangleq \left\{ y \in \mathbb{R}^V \mid x^\intercal y = \|x\|_2^2 \right\} \tag{17.30}$$

$H(x)$: 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.

$$H(x) \triangleq \left\{ y \in \mathbb{R}^V \mid x^\intercal y = \|x\|_2^2 \right\} \tag{17.30}$$

- 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.

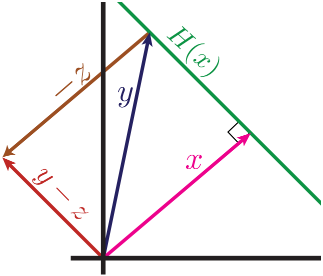


$H(x)$: Orthogonal x-containing hyperplane

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

$$H(x) \triangleq \left\{ y \in \mathbb{R}^V \mid x^\intercal y = \|x\|_2^2 \right\} \tag{17.30}$$

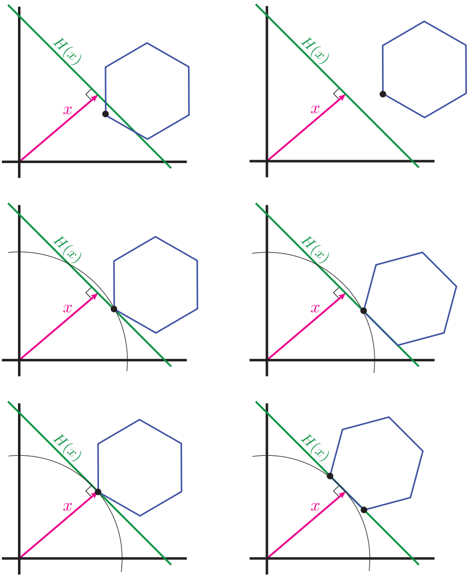
- Any set $\{y \in \mathbb{R}^V | x^\intercal y = c\}$ is orthogonal to the line from 0 to x. This follows since, for constant z, $\{y : (y - z)^\intercal x = 0\} = \{y : y^\intercal x = z^\intercal x\}$ is hyperplane orthogonal to x translated by z. Take $c = z^\intercal x$ for result, and $z = x$,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



- Note, $H(x)$ is translation of subspace of dimension $|V| - 1 = n - 1$ (i.e., $H(x) - \{x\}$ is a subspace, $H(x)$ is an affine set).

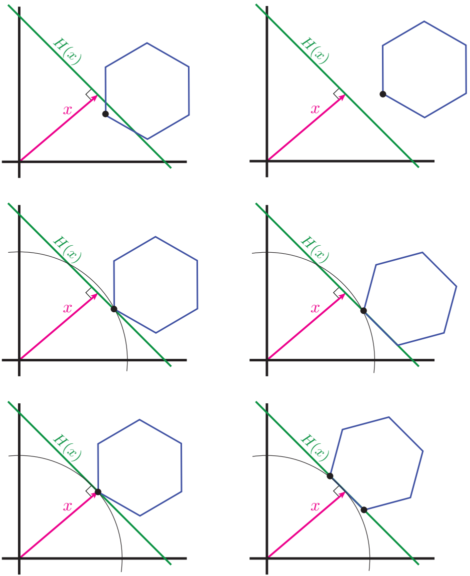
Ex: $H(x)$, polytopes, and supporting hyperplanes

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



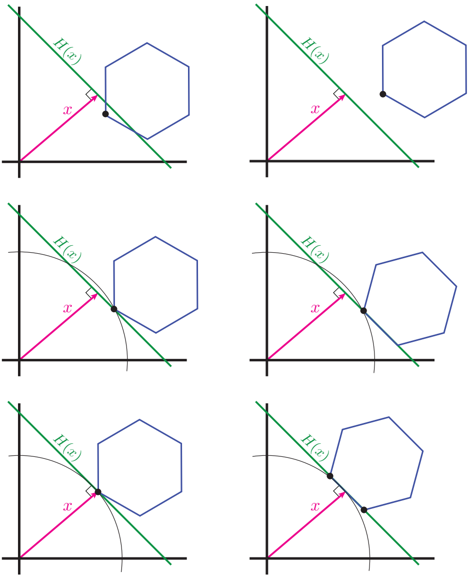
Ex: $H(x)$, polytopes, and supporting hyperplanes

- $H(x) = \left\{ y \in \mathbb{R}^V | x^\intercal y = \|x\|_2^2 \right\}$, 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: $x^\intercal p < x^\intercal x$; TR: $x^\intercal p > x^\intercal x$; middle row: $x^\intercal p = x^\intercal x$.



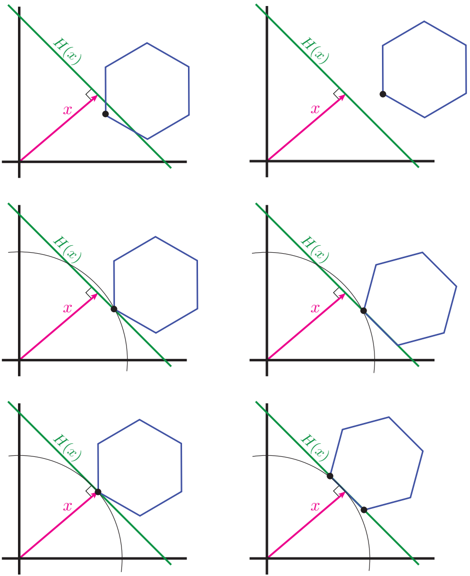
Ex: $H(x)$, polytopes, and supporting hyperplanes

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



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

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



Notation

- The line between x and y: given two points $x, y \in \mathbb{R}^V$, let $[x, y] \triangleq \{\lambda x + (1 - \lambda y) : \lambda \in [0, 1]\}$. Hence, $[x, y] = \text{conv}\{x, y\}$.

Notation

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

Fujishige-Wolfe Min-Norm Algorithm

- Wolfe-1976 ("Finding the Nearest Point in a Polytope") developed an algorithm to compute the minimum norm point of a polytope, specified as a set of vertices.

Fujishige-Wolfe Min-Norm Algorithm

- Wolfe-1976 ("Finding the Nearest Point in a Polytope") developed an algorithm to compute the minimum norm point of a polytope, specified as a set of vertices.
- Fujishige-1984 "Submodular Systems and Related Topics" realized this algorithm can find the the min. norm point of B_f.

Fujishige-Wolfe Min-Norm Algorithm

- Wolfe-1976 ("Finding the Nearest Point in a Polytope") developed an algorithm to compute the minimum norm point of a polytope, specified as a set of vertices.
- Fujishige-1984 "Submodular Systems and Related Topics" realized this algorithm can find the the min. norm point of B_f.
- Seems to be (among) the fastest general purpose SFM algo.

Fujishige-Wolfe Min-Norm Algorithm

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

$$\min_{x \in \text{conv } P} \|x\|_2 \tag{17.31}$$

Fujishige-Wolfe Min-Norm Algorithm

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

$$\min_{x \in \text{conv } P} \|x\|_2 \tag{17.31}$$

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

Fujishige-Wolfe Min-Norm Algorithm

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

$$\min_{x \in \operatorname{conv} P} \|x\|_2 \tag{17.31}$$

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

Fujishige-Wolfe Min-Norm Algorithm

- When Q are affinely independent, minimum norm point in the affine hull of Q can easily be found, as a closed form solution for $\min_{x \in \operatorname{aff} Q} \|x\|_2$ is available (see below).

Fujishige-Wolfe Min-Norm Algorithm

- When Q are affinely independent, minimum norm point in the affine hull of Q can easily be found, as a closed form solution for $\min_{x \in \text{aff } Q} \|x\|_2$ is available (see below).
- Algorithm repeatedly produces min. norm point x^* for selected set Q.

Fujishige-Wolfe Min-Norm Algorithm

- 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 $\text{conv}\, Q$.

Fujishige-Wolfe Min-Norm Algorithm

- When Q are affinely independent, minimum norm point in the affine hull of Q can easily be found, as a closed form solution for $\min_{x \in \operatorname{aff} Q} \|x\|_2$ is available (see below).
- Algorithm repeatedly produces min. norm point x^* for selected set Q.
- If we find $w_i \geq 0, i = 1, \cdots, m$ for the minimum norm point, then x^* also belongs to $\operatorname{conv} Q$ and also a minimum norm point over $\operatorname{conv} Q$.
- If $Q \subseteq P$ is suitably chosen, x^* may even be the minimum norm point over $\operatorname{conv} P$ solving the original problem.

Fujishige-Wolfe Min-Norm Algorithm

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

Fujishige-Wolfe Min-Norm Algorithm

- When Q are affinely independent, minimum norm point in the affine hull of Q can easily be found, as a closed form solution for $\min_{x \in \mathrm{aff}\, Q} \|x\|_2$ 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 $\mathrm{conv}\, Q$ and also a minimum norm point over $\mathrm{conv}\, Q$.
- If $Q \subseteq P$ is suitably chosen, x^* may even be the minimum norm point over $\mathrm{conv}\, P$ solving the original problem.
- One of the most expensive parts of Wolfe's algorithm is solving linear optimization problem over the polytope, doable by examining all the extreme points in the polytope.
- If number of extreme points is exponential, hard to do in general.

Fujishige-Wolfe Min-Norm Algorithm

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

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

Input : $P = \{p_1, \cdots, p_m\}, p_i \in \mathbb{R}^n, i = 1, \cdots, m$.

Output: x^*: the minimum-norm-point in $\operatorname{conv} P$.

$x^* \longleftarrow p_{i^*}$ where $p_{i^*} \in \operatorname{argmin}_{p \in P} \|p\|_2$ /* or choose it arbitrarily */ ;

$Q \longleftarrow \{x^*\}$;

while 1 **do** /* major loop */

    if $x^* = 0$ *or* $H(x^*)$ *separates* P *from origin* **then**

        return : x^*

    else

        Choose $\hat{x} \in P$ on the near (closer to 0) side of $H(x^*)$;

        $Q = Q \cup \{\hat{x}\}$;

    while 1 **do** /* minor loop */

        $x_0 \longleftarrow \operatorname{argmin}_{x \in \operatorname{aff} Q} \|x\|_2$;

        if $x_0 \in \operatorname{conv} Q$ **then**

            $x^* \longleftarrow x_0$;

            break;

        else

            $y \longleftarrow \operatorname{argmin}_{x \in \operatorname{conv} Q \cap [x^*, x_0]} \|x - x_0\|_2$;

            Delete from Q points not on the face of $\operatorname{conv} Q$ where y lies;

            $x^* \longleftarrow y$;

Fujishige-Wolfe Min-Norm algorithm: Geometric Example

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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example

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

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{17.32}$$

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:

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{17.32}$$

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

$$\min_{x \in \operatorname{aff} Q} \|x\|_2 \leq \min_{x \in \operatorname{conv} Q} \|x\|_2 \leq \|x^*\|_2 \tag{17.33}$$

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:

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{17.32}$$

  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

$$\min_{x \in \operatorname{aff} Q} \|x\|_2 \leq \min_{x \in \operatorname{conv} Q} \|x\|_2 \leq \|x^*\|_2 \tag{17.33}$$

- 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:

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{17.32}$$

  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

$$\min_{x \in \operatorname{aff} Q} \|x\|_2 \leq \min_{x \in \operatorname{conv} Q} \|x\|_2 \leq \|x^*\|_2 \tag{17.33}$$

- 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:

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{17.32}$$

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.

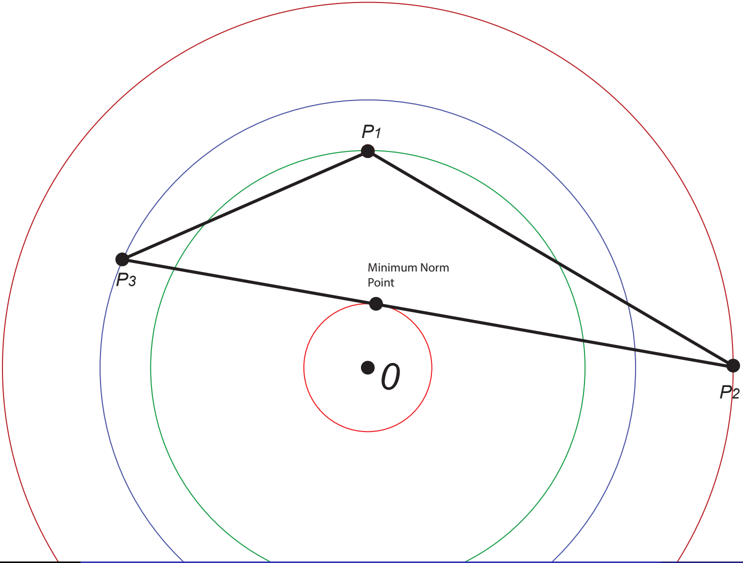
- Note also for any $x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P$, we have

$$\min_{x \in \operatorname{aff} Q} \|x\|_2 \leq \min_{x \in \operatorname{conv} Q} \|x\|_2 \leq \|x^*\|_2 \tag{17.33}$$

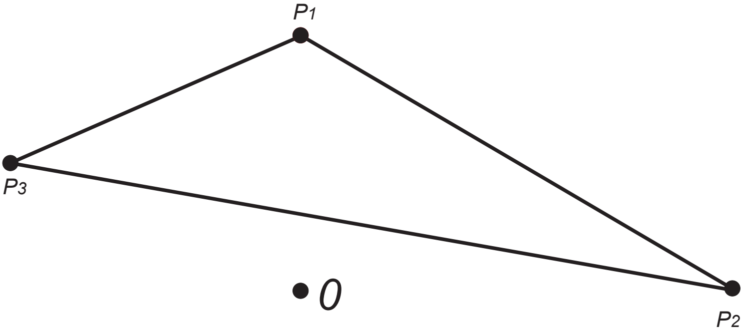
- Note, the input, P, consists of m points. In the case of the base polytope, $P = B_f$ could be exponential in $n = |V|$.
- There are six places that might be seemingly tricky or expensive: Line 4, Line 6, Line 9, Line 10, Line 14, and Line 15.
- We will consider each in turn, but first we do a geometric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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example

Polytope, and circles concentric at 0.

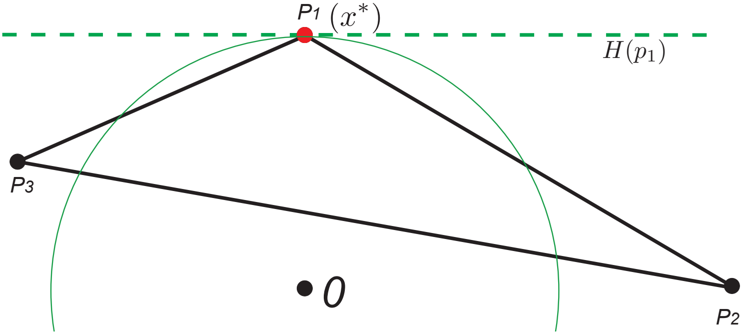


Fujishige-Wolfe Min-Norm algorithm: Geometric Example



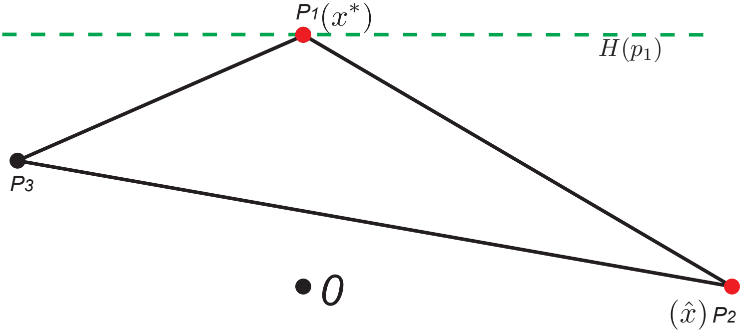
The initial polytope consisting of the convex hull of three points p_1, p_2, p_3, and the origin 0.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



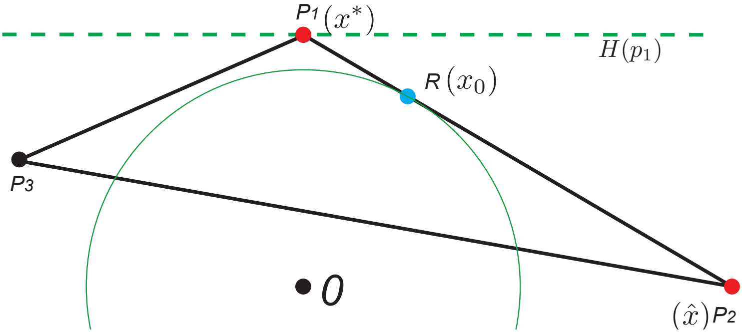
p_1 is the extreme point closest to 0 and so we choose it first, although we can choose any arbitrary extreme point as the initial point. We set $x^* \leftarrow p_1$ in Line 1, and $Q \leftarrow \{p_1\}$ in Line 2. $H(x^*) = H(p_1)$ (green dashed line) is not a supporting hyperplane of $\text{conv}(P)$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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



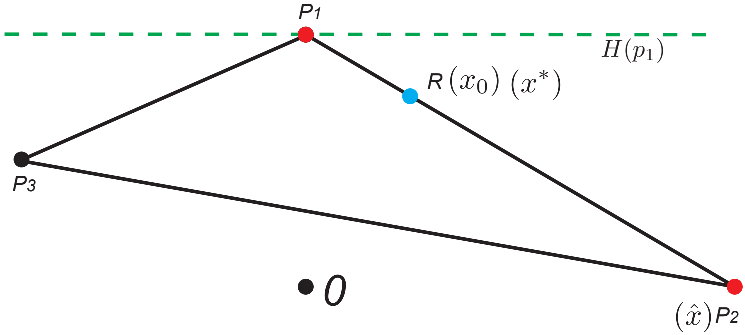
We need to add some extreme point $\hat{x}$ on the "near" side of $H(p_1)$ in Line 6, we choose $\hat{x} = p_2$. In Line 7, we set $Q \leftarrow Q \cup \{p_2\}$, so $Q = \{p_1, p_2\}$.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



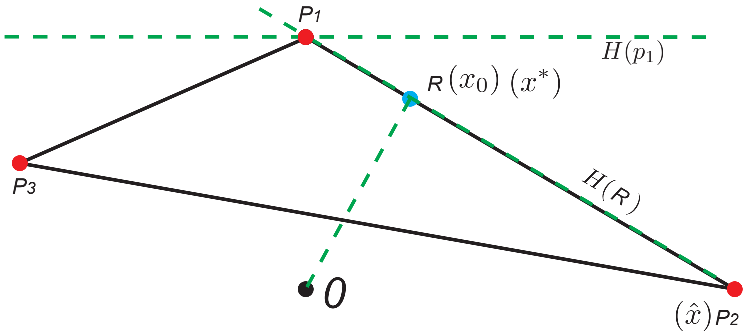
$x_0 = R$ is the min-norm point in $\text{aff}\,\{p_1, p_2\}$ computed in Line 9.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



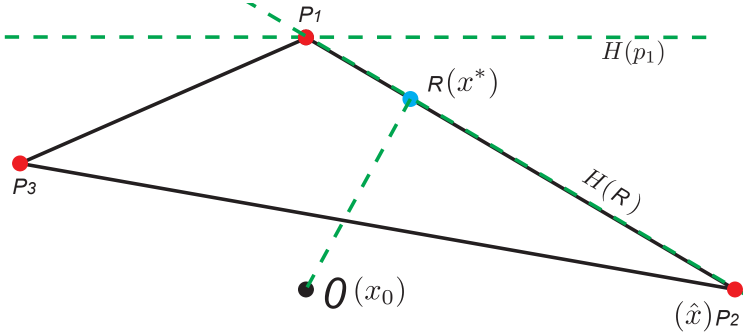
$x_0 = R$ is the min-norm point in $\text{aff}\,\{p_1, p_2\}$ computed in Line 9. Also, with $Q = \{p_1, p_2\}$, since $R \in \text{conv}\,Q$, we set $x^* \leftarrow x_0 = R$ in Line 11, not violating the invariant $x^* \in \text{conv}\,Q$. Note, after Line 11, we still have $x^* \in P$ and $\|x^*\|_2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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



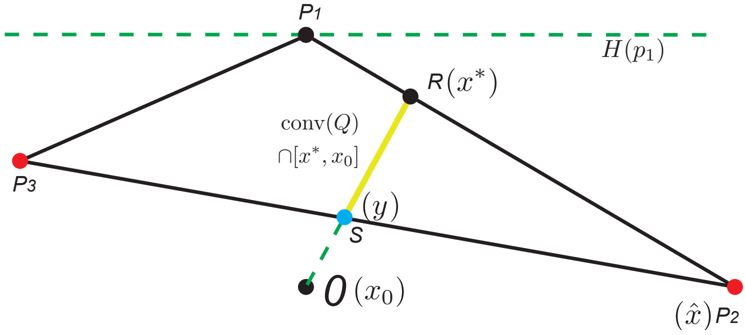
$R = x_0 = x^*$. We consider next $H(R) = H(x^*)$ in Line 4. $H(x^*)$ is not a supporting hyperplane of $\operatorname{conv} P$. So we choose p_3 on the "near" side of $H(x^*)$ in Line 6. Add $Q \leftarrow Q \cup \{p_3\}$ in Line 7. Now $Q = P = \{p_1, p_2, p_3\}$.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



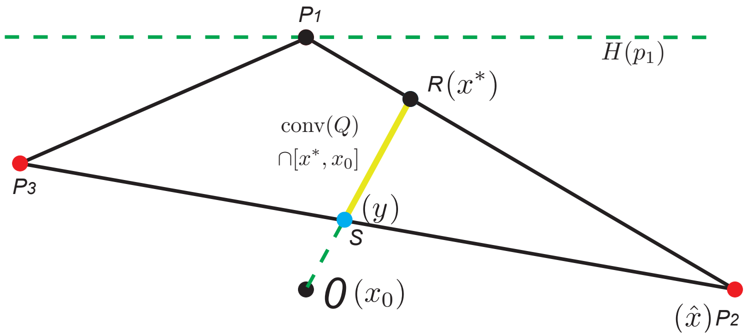
$R = x_0 = x^*$. We consider next $H(R) = H(x^*)$ in Line 4. $H(x^*)$ is not a supporting hyperplane of $\operatorname{conv} P$. So we choose p_3 on the "near" side of $H(x^*)$ in Line 6. Add $Q \leftarrow Q \cup \{p_3\}$ in Line 7. Now $Q = P = \{p_1, p_2, p_3\}$. The origin $x_0 = 0$ is the min-norm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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



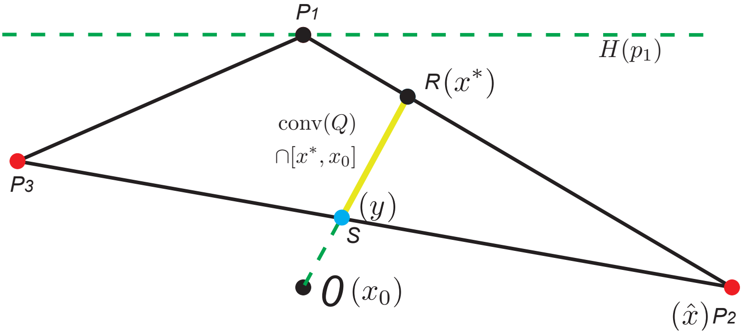
$Q = P = \{p_1, p_2, p_3\}$. Line 14: $S = y = \mathrm{argmin}_{x \in \mathrm{conv}\, Q \cap [x^*, x_0]} \|x - x_0\|_2$ where x_0 is 0 and x^* is R here. Thus, y lies on the boundary of $\mathrm{conv}\, Q$. Note, $\|y\|_2 < \|x^*\|_2$ since $x^* \in \mathrm{conv}\, Q$, $\|x_0\|_2 < \|x^*\|_2$.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



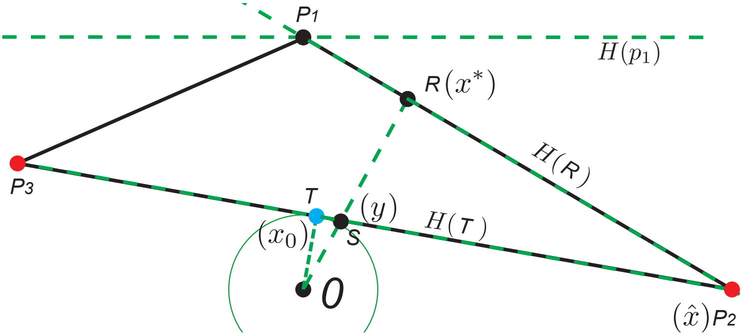
$Q = P = \{p_1, p_2, p_3\}$. Line 14: $S = y = \mathrm{argmin}_{x \in \mathrm{conv}\, Q \cap [x^*, x_0]} \|x - x_0\|_2$ where x_0 is 0 and x^* is R here. Thus, y lies on the boundary of $\mathrm{conv}\, Q$. Note, $\|y\|_2 < \|x^*\|_2$ since $x^* \in \mathrm{conv}\, Q$, $\|x_0\|_2 < \|x^*\|_2$. Line 15: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



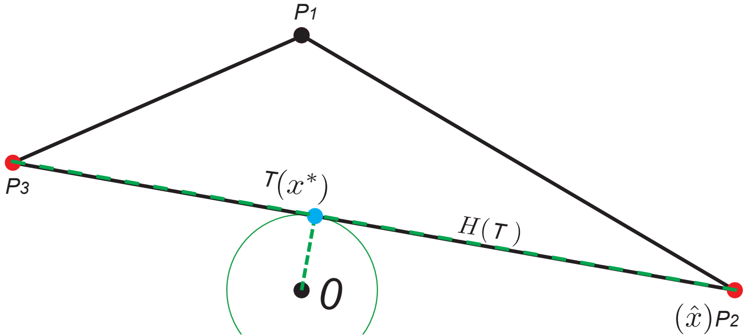
$Q = P = \{p_1, p_2, p_3\}$. Line 14: $S = y = \mathrm{argmin}_{x \in \mathrm{conv}\, Q \cap [x^*, x_0]} \|x - x_0\|_2$ where x_0 is 0 and x^* is R here. Thus, y lies on the boundary of $\mathrm{conv}\, Q$. Note, $\|y\|_2 < \|x^*\|_2$ since $x^* \in \mathrm{conv}\, Q$, $\|x_0\|_2 < \|x^*\|_2$. Line 15: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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



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

Condition for Min-Norm Point

Theorem 17.5.1

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

$$p_i^\intercal x^* \geq \|x^*\|_2^2 \quad \forall i = 1, \cdots, m. \tag{17.34}$$

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 17.5.1

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

$$p_i^\intercal x^* \geq \|x^*\|_2^2 \quad \forall i = 1, \cdots, m. \tag{17.34}$$

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

$$\|z\|_2^2 = \|x^* + \theta(y - x^*)\|_2^2 \tag{17.35}$$

$$= \|x^*\|_2^2 + 2\theta({x^*}^\intercal y - {x^*}^\intercal x^*) + \theta^2 \|y - x^*\|_2^2 \tag{17.36}$$

Condition for Min-Norm Point

Theorem 17.5.1

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

$$p_i^\intercal x^* \geq \|x^*\|_2^2 \quad \forall i = 1, \cdots, m. \tag{17.34}$$

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

$$\|z\|_2^2 = \|x^* + \theta(y - x^*)\|_2^2 \tag{17.35}$$

$$= \|x^*\|_2^2 + 2\theta(x^{*\intercal} y - x^{*\intercal} x^*) + \theta^2 \|y - x^*\|_2^2 \tag{17.36}$$

- 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 (17.34).

□

Condition for Min-Norm Point

Theorem 17.5.1

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

$$p_i^\intercal x^* \geq \|x^*\|_2^2 \quad \forall i = 1, \cdots, m. \tag{17.34}$$

Proof.

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

$$\|z\|_2^2 = \|x^* + \theta(y - x^*)\|_2^2 \tag{17.35}$$

$$= \|x^*\|_2^2 + 2\theta(x^{*\intercal} y - x^{*\intercal} x^*) + \theta^2 \|y - x^*\|_2^2 \tag{17.36}$$

- It is possible for $\|z\|_2^2 < \|x^*\|_2^2$ for small θ, unless $x^{*\intercal} y \geq x^{*\intercal} x^*$ for all $y \in \text{conv}\, P \Rightarrow$ Equation (17.34).
- Conversely, given Eq (17.34), and given that $y = \sum_i \lambda_i p_i \in \text{conv}\, P$,

$$y^\intercal x^* = \sum_i \lambda_i p_i^\intercal x^* \geq \sum_i \lambda_i x^{*\intercal} x^* = x^{*\intercal} x^* \tag{17.37}$$

implying that $\|z\|_2^2 > \|x^*\|_2^2$ in Equation 17.36 for arbitrary $z \in \text{conv}\, P$. $\square$

The set Q is always affinely independent

Lemma 17.5.2

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

The set Q is always affinely independent

Lemma 17.5.2

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

Proof.

$\square$

The set Q is always affinely independent

Lemma 17.5.2

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 17.5.2

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 17.5.2

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

□

The set Q is always affinely independent

Lemma 17.5.2

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 17.5.2

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 17.5.2

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 17.5.2

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

Thus, by Lemma 17.5.2, we have for any $x \in \text{aff}\, Q$ such that $x = \sum_i w_i q_i$ with $\sum_i w_i = 1$, the weights w_i are uniquely determined.

Minimum Norm in an affine set

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

Minimum Norm in an affine set

- Line 9 of the algorithm requires $x_0 \leftarrow \min_{x \in \text{aff } Q} \|x\|_2$.
- When Q is affinely independent, this is relatively easy.

Minimum Norm in an affine set

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

$$\text{minimize} \quad \|x\|_2^2 = w^\intercal Q^\intercal Q w \tag{17.38}$$

$$\text{subject to} \quad \mathbf{1}^\intercal w = 1 \tag{17.39}$$

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.
- Let Q also represent the $n \times k$ matrix with points as columns $q \in Q$. We get the following, solvable with matrix inversion/linear solver:

$$\text{minimize} \qquad \|x\|_2^2 = w^\intercal Q^\intercal Q w \tag{17.38}$$

$$\text{subject to} \qquad \mathbf{1}^\intercal w = 1 \tag{17.39}$$

- 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

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

$$\text{minimize} \qquad \|x\|_2^2 = w^\intercal Q^\intercal Q w \tag{17.38}$$

$$\text{subject to} \qquad \mathbf{1}^\intercal w = 1 \tag{17.39}$$

- 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

- 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.
- Let Q also represent the $n \times k$ matrix with points as columns $q \in Q$. We get the following, solvable with matrix inversion/linear solver:

$$\text{minimize} \qquad \|x\|_2^2 = w^\intercal Q^\intercal Q w \tag{17.38}$$

$$\text{subject to} \qquad \mathbf{1}^\intercal w = 1 \tag{17.39}$$

- 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 17.5.3

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 17.5.3

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 17.5.3

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 17.5.3

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 17.5.3 continued.

- Moreover, there can be no more iterations within a minor loop than the dimension of $\text{conv}\, Q$ for the initial Q given to the minor loop initially at Line 8 (dimension of $\text{conv}\, Q$ is $|Q| - 1$ since Q is affinely independent).

. . .

MN Algorithm finds the MN point in finite time.

. . . proof of Theorem 17.5.3 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 17.5.3 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 17.5.3 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 17.5.3 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 17.5.3 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 17.5.3 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 17.5.3 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

$$\begin{aligned}\|x^* + \theta(\hat{x} - x^*)\|_2^2 &= \|x^*\|_2^2 + 2\theta\left((x^*)^\top \hat{x} - \|x^*\|_2^2\right) + \theta^2 \|\hat{x} - x^*\|_2^2 \\ &\geq \|x_0\|_2^2 \end{aligned} \tag{17.40}$$

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 17.5.3 continued.

- Therefore, for sufficiently small θ, specifically for

$$\theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2} \tag{17.41}$$

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

$\square$

MN Algorithm finds the MN point in finite time.

... proof of Theorem 17.5.3 continued.

- Therefore, for sufficiently small θ, specifically for

$$\theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2} \tag{17.41}$$

  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 17.5.3 continued.

- Therefore, for sufficiently small θ, specifically for

$$\theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2} \tag{17.41}$$

  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. 17.40, reduction on norm is lower-bounded:

$$\Delta = \|x^*\|_2^2 - \|x_0\|_2^2 \geq 2\theta \left(\|x^*\|_2^2 - (x^*)^\top \hat{x} \right) - \theta^2 \|\hat{x} - x^*\|_2^2 \triangleq \underline{\Delta} \tag{17.42}$$

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. 17.40, reduction on norm is lower-bounded:

$$\Delta = \|x^*\|_2^2 - \|x_0\|_2^2 \geq 2\theta \left(\|x^*\|_2^2 - (x^*)^\top \hat{x} \right) - \theta^2 \|\hat{x} - x^*\|_2^2 \triangleq \underline{\Delta} \tag{17.42}$$

- 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:

$$\max_{0 \leq \theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2}} \underline{\Delta} = \left(\frac{\|x^*\|_2^2 - (x^*)^\top \hat{x}}{\|\hat{x} - x^*\|_2} \right)^2 \tag{17.43}$$

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 17.43) 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 17.43) is maximized.
- That is, we want to find

$$\hat{x} \in \underset{x \in P}{\operatorname{argmax}} \left(\frac{\|x^*\|_2^2 - (x^*)^\top x}{\|x - x^*\|_2} \right)^2 \tag{17.44}$$

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

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \left(\frac{\|x^*\|_2^2 - (x^*)^\top x}{\|x - x^*\|_2} \right)^2 \tag{17.44}$$

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. 17.44, i.e., find

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \|x^*\|_2^2 - (x^*)^\top x = \operatorname*{argmin}_{x \in P} (x^*)^\top x, \tag{17.45}$$

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

- As a surrogate, we maximize numerator in Eqn. 17.44, i.e., find

$$\hat{x} \in \underset{x \in P}{\operatorname{argmax}} \|x^*\|_2^2 - (x^*)^\top x = \underset{x \in P}{\operatorname{argmin}} (x^*)^\top x, \tag{17.45}$$

- 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. 17.44, i.e., find

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \|x^*\|_2^2 - (x^*)^\top x = \operatorname*{argmin}_{x \in P} (x^*)^\top x, \tag{17.45}$$

- 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. 17.44, i.e., find

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \|x^*\|_2^2 - (x^*)^\top x = \operatorname*{argmin}_{x \in P} (x^*)^\top x, \tag{17.45}$$

- 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.
- Mathematically and theoretically, we terminate the algorithm if

$$(x^*)^\top \hat{x} \geq \|x^*\|_2^2\,, \tag{17.46}$$

where $\hat{x}$ is the solution of Eq. 17.45.

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

$$(x^*)^\top \hat{x} > \|x^*\|_2^2 - \epsilon \max_{x \in Q} \|x\|_2^2 \tag{17.47}$$

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

$$(x^*)^\top \hat{x} > \|x^*\|_2^2 - \epsilon \max_{x \in Q} \|x\|_2^2 \tag{17.47}$$

- 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 17.45 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

$$(x^*)^\top \hat{x} > \|x^*\|_2^2 - \epsilon \max_{x \in Q} \|x\|_2^2 \tag{17.47}$$

- 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 17.45 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 $H(x^*)$

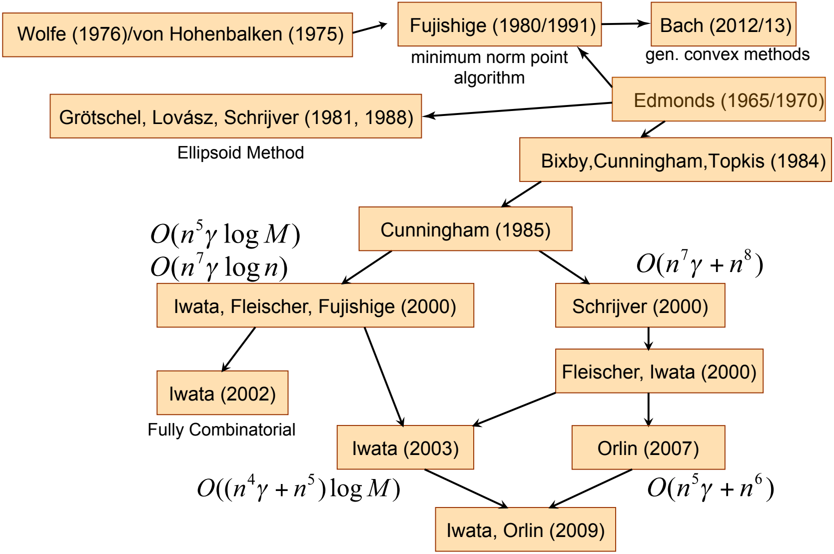
- In practice,the above optimality test might never hold numerically. Hence, as suggested by Wolfe, we introduce a tolerance parameter $\epsilon > 0$, and terminates the algorithm if

$$(x^*)^\top \hat{x} > \|x^*\|_2^2 - \epsilon \max_{x \in Q} \|x\|_2^2 \tag{17.47}$$

- 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 17.45 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.

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

General Submodular Function Minimization



MN Algorithm Complexity

- The currently fastest strongly polynomial combinatorial algorithm for SFM achieves a running time of $O(n^5 T + n^6)$ (Orlin'09) where T is the time for function evaluation, far from practical for large problem instances.

MN Algorithm Complexity

- The currently fastest strongly polynomial combinatorial algorithm for SFM achieves a running time of $O(n^5 T + n^6)$ (Orlin'09) where T is the time for function evaluation, far from practical for large problem instances.
- Fujishige & Isotani report that MN algorithm is fast in practice, but they use only a limited set of submodular functions.

MN Algorithm Complexity

- The currently fastest strongly polynomial combinatorial algorithm for SFM achieves a running time of $O(n^5T + n^6)$ (Orlin'09) where T is the time for function evaluation, far from practical for large problem instances.
- Fujishige & Isotani report that MN algorithm is fast in practice, but they use only a limited set of submodular functions.
- Complexity of MN Algorithm is still an unsolved problem.

MN Algorithm Complexity

- The currently fastest strongly polynomial combinatorial algorithm for SFM achieves a running time of $O(n^5 T + n^6)$ (Orlin'09) where T is the time for function evaluation, far from practical for large problem instances.
- Fujishige & Isotani report that MN algorithm is fast in practice, but they use only a limited set of submodular functions.
- Complexity of MN Algorithm is still an unsolved problem.
- Obvious facts:

MN Algorithm Complexity

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

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

    where each function oracle call requires $O(n^p)$ time.

MN Algorithm Complexity

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

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

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

MN Algorithm Empirical Complexity

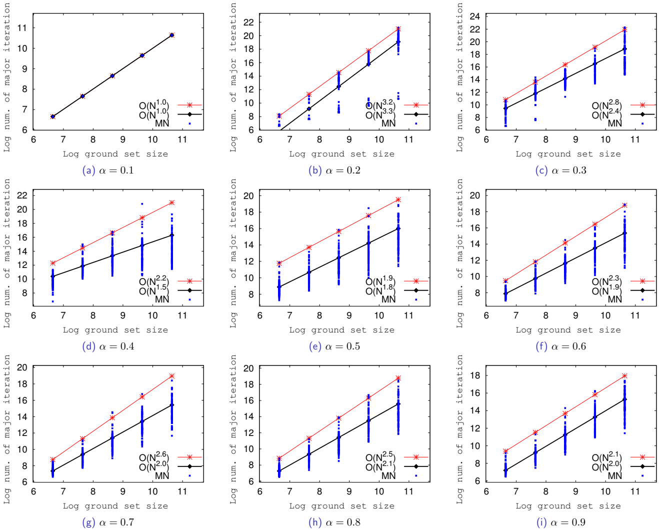


Figure: The number of major iteration for $f(S) = -m_1(S) + 100 \cdot (w_1(\mathcal{N}(S)))^{\alpha}$. The red lines are the linear interpolations of the worst case points, and the black lines are the linear interpolations of the average case points. From Lin&Bilmes 2014 (unpublished)

MN Algorithm Complexity

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

MN Algorithm Complexity

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

MN Algorithm Complexity

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

MN Algorithm Complexity

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

Min-Norm Point and SFM

Theorem 17.6.1

Let y^*, A_-, and A_0 be as given. Then y^* is a maximizer of the l.h.s. of Eqn. (17.7). Moreover, A_- is the unique minimal minimizer of f and A_0 is the unique maximal minimizer of f.

Proof.

- First note, since $x^* \in B_f$, we have $x^*(E) = f(E)$, meaning $\text{sat}(x^*) = E$. Thus, we can consider any $e \in E$ within $\text{dep}(x^*, e)$.

...

Min-Norm Point and SFM

Theorem 17.6.1

Let y^*, A_-, and A_0 be as given. Then y^* is a maximizer of the l.h.s. of Eqn. (17.7). Moreover, A_- is the unique minimal minimizer of f and A_0 is the unique maximal minimizer of f.

Proof.

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

...

Min-Norm Point and SFM

Theorem 17.6.1

Let y^*, A_-, and A_0 be as given. Then y^* is a maximizer of the l.h.s. of Eqn. (17.7). Moreover, A_- is the unique minimal minimizer of f and A_0 is the unique maximal minimizer of f.

Proof.

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

$$(x^* + \alpha \mathbf{1}_e - \alpha \mathbf{1}_{e'})(E) = x^*(E) + \alpha - \alpha = f(E) \tag{17.48}$$

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

...

Min-Norm Point and SFM

. . . proof of Thm. 17.6.1 cont.

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

. . .

Min-Norm Point and SFM

... proof of Thm. 17.6.1 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. 17.6.1 cont.

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

. . .

Min-Norm Point and SFM

. . . proof of Thm. 17.6.1 cont.

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

. . .

Min-Norm Point and SFM

. . . proof of Thm. 17.6.1 cont.

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

. . .

Min-Norm Point and SFM

. . . proof of Thm. 17.6.1 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. 17.6.1 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. 17.6.1 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 $\mathrm{dep}(x^*, e) \subseteq A_0$.

. . .

Min-Norm Point and SFM

. . . proof of Thm. 17.6.1 cont.

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

. . .

Min-Norm Point and SFM

... proof of Thm. 17.6.1 cont.

- Therefore, we have $\cup_{e \in A_-} \text{dep}(x^*, e) = A_-$ and $\cup_{e \in A_0} \text{dep}(x^*, e) = A_0$
- Ie., $\{\text{dep}(x^*, e)\}_{e \in A_-}$ is cover for A_-, as is $\{\text{dep}(x^*, e)\}_{e \in A_0}$ for A_0.

...

Min-Norm Point and SFM

...proof of Thm. 17.6.1 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. 17.6.1 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:

$$x^*(A_-) = f(A_-) \tag{17.49}$$

. . .

Min-Norm Point and SFM

. . . proof of Thm. 17.6.1 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:

$$x^*(A_-) = f(A_-) \tag{17.49}$$

$$x^*(A_0) = f(A_0) \tag{17.50}$$

. . .

Min-Norm Point and SFM

. . . proof of Thm. 17.6.1 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:

$$x^*(A_-) = f(A_-) \tag{17.49}$$

$$x^*(A_0) = f(A_0) \tag{17.50}$$

$$x^*(A_-) = x^*(A_0) = y^*(E) = y^*(A_0) + \underbrace{y^*(E \setminus A_0)}_{=0} \tag{17.51}$$

. . .

Min-Norm Point and SFM

. . . proof of Thm. 17.6.1 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:

$$x^*(A_-) = f(A_-) \tag{17.49}$$

$$x^*(A_0) = f(A_0) \tag{17.50}$$

$$x^*(A_-) = x^*(A_0) = y^*(E) = y^*(A_0) + \underbrace{y^*(E \setminus A_0)}_{=0} \tag{17.51}$$

and therefore, all together we have

. . .

Min-Norm Point and SFM

. . . proof of Thm. 17.6.1 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:

$$x^*(A_-) = f(A_-) \tag{17.49}$$

$$x^*(A_0) = f(A_0) \tag{17.50}$$

$$x^*(A_-) = x^*(A_0) = y^*(E) = y^*(A_0) + \underbrace{y^*(E \setminus A_0)}_{=0} \tag{17.51}$$

and therefore, all together we have

$$f(A_-) = f(A_0) = x^*(A_-) = x^*(A_0) = y^*(E) \tag{17.52}$$

. . .

Min-Norm Point and SFM

... proof of Thm. 17.6.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (17.7).

...

Min-Norm Point and SFM

. . . proof of Thm. 17.6.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (17.7). 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. 17.6.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (17.7). This follows since, we have $y^* = x^* \wedge 0 \leq 0$, and since $x^* \in B_f \subset P_f$, and $y^* \leq x^*$ and P_f is down-closed, we have that $y^* \in P_f$.
- Also, for any $y \in P_f$ with $y \leq 0$ and for any $X \subseteq E$, we have $y(E) \leq y(X) \leq f(X)$.

. . .

Min-Norm Point and SFM

. . . proof of Thm. 17.6.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (17.7). 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. (17.7), $y^* \leq 0$, $y^* \in P_f$, so $y^*(E) \leq f(X)$ for all X.

. . .

Min-Norm Point and SFM

... proof of Thm. 17.6.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (17.7). 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. (17.7), $y^* \leq 0$, $y^* \in P_f$, 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. 17.6.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (17.7). 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. (17.7), $y^* \leq 0$, $y^* \in P_f$, so $y^*(E) \leq f(X)$ for all X.
- So $y^*(E) \leq \min \{f(X) | X \subseteq V\}$.
- Considering Eqn. (17.49), we have found sets A_- and A_0 with tightness in Eqn. (17.7), meaning $y^*(E) = f(A_-) = f(A_0)$.

...

Min-Norm Point and SFM

. . . proof of Thm. 17.6.1 cont.

- Now, y^* is feasible for the l.h.s. of Eqn. (17.7). 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. (17.7), $y^* \leq 0$, $y^* \in P_f$, so $y^*(E) \leq f(X)$ for all X.
- So $y^*(E) \leq \min\{f(X)|X \subseteq V\}$.
- Considering Eqn. (17.49), we have found sets A_- and A_0 with tightness in Eqn. (17.7), meaning $y^*(E) = f(A_-) = f(A_0)$.
- Hence, y^* is a maximizer of l.h.s. of Eqn. (17.7), and A_- and A_0 are minimizers of f.

. . .

Min-Norm Point and SFM

. . . proof of Thm. 17.6.1 cont.

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

$$f(X) \geq x^*(X) > x^*(A_-) = f(A_-) \tag{17.53}$$

□

Min-Norm Point and SFM

... proof of Thm. 17.6.1 cont.

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

$$f(X) \geq x^*(X) > x^*(A_-) = f(A_-) \tag{17.53}$$

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

$$f(X) \geq x^*(X) > x^*(A_0) = f(A_0) \tag{17.54}$$

$\square$

Min-Norm Point and SFM

. . . proof of Thm. 17.6.1 cont.

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

$$f(X) \geq x^*(X) > x^*(A_-) = f(A_-) \tag{17.53}$$

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

$$f(X) \geq x^*(X) > x^*(A_0) = f(A_0) \tag{17.54}$$

- 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).
- An algorithm (by F. Wolfe) can find this min-norm point, essentially an active-set procedure for quadratic programming. It uses Edmonds's greedy algorithm to make it efficient.

Min-Norm Point and SFM

- So, if we have a procedure to compute the min-norm point computation, we can solve SFM.
- Nice thing about previous proof is that it uses both expressions for dep for different purposes.
- This was discovered by Fujishige (in fact the proof above is an expanded version of the one found in the book).
- An algorithm (by F. Wolfe) can find this min-norm point, essentially an active-set procedure for quadratic programming. It uses Edmonds's greedy algorithm to make it efficient.
- This is currently the best practical algorithm for general purpose submodular function minimization.

Min-Norm Point and SFM

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

Min-norm point and other minimizers of f

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

Min-norm point and other minimizers of f

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

Min-norm point and other minimizers of f

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

Theorem 17.6.2

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

$$A = A_- \cup \bigcup_{a \in A_m} \mathrm{dep}(x^*, a) \tag{17.55}$$

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

Min-norm point and other minimizers of f

proof of Thm. 17.6.2.

- 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. 17.6.2.

- If A is a minimizer, then $A_- \subseteq A \subseteq A_0$, and $f(A) = y^*(E)$ is the minimum valuation of f.
- But $x^* \in P_f$, so $x^*(A) \leq f(A)$ and $f(A) = x^*(A_-) \leq x^*(A)$ (or alternatively, just note that $x^*(A_0 \setminus A) = 0$).

□

Min-norm point and other minimizers of f

proof of Thm. 17.6.2.

- If A is a minimizer, then $A_- \subseteq A \subseteq A_0$, and $f(A) = y^*(E)$ is the minimum valuation of f.
- But $x^* \in P_f$, so $x^*(A) \leq f(A)$ and $f(A) = x^*(A_-) \leq x^*(A)$ (or alternatively, just note that $x^*(A_0 \setminus A) = 0$).
- Hence, $x^*(A) = x^*(A_-) = f(A)$ so that A is also a tight set for x^*.

$\square$

Min-norm point and other minimizers of f

proof of Thm. 17.6.2.

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

□

Min-norm point and other minimizers of f

proof of Thm. 17.6.2.

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

$\square$

Min-norm point and other minimizers of f

proof of Thm. 17.6.2.

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

$\square$

Min-norm point and other minimizers of f

proof of Thm. 17.6.2.

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

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

$\square$

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):

$$\underset{w \in \mathbb{R}^V}{\text{minimize}} \quad \tilde{f}(w) + \frac{1}{2}\|w\|_2^2 \tag{17.56}$$

$$\text{maximize} \quad -\|x\|_2^2 \tag{17.57a}$$

$$\text{subject to} \quad x \in B_f \tag{17.57b}$$

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 (17.56) is related to proximal methods to minimize the Lovász extension (see Parikh&Boyd, "Proximal Algorithms" 2013).
- Equation (17.57b) is solved by the minimum-norm point algorithm (Wolfe-1976, Fujishige-1984, Fujishige-2005, Fujishige-2011) is (as we will see) essentially an active-set procedure for quadratic programming, and uses Edmonds's greedy algorithm to make it efficient.
- Unknown worst-case running time, although in practice it usually performs quite well (see below).