Submodular Functions, Optimization, and Applications to Machine Learning

— Spring Quarter, Lecture 18 —

http://j.ee.washington.edu/~bilmes/classes/ee596b_spring_2014/

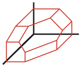
Prof. Jeff Bilmes

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

June 2nd, 2014

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

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



Cumulative Outstanding Reading

- Good references for today: Schrijver-2003, Oxley-1992/2011, Welsh-1973, Goemans-2010, Cunningham-1984, Edmonds-1969, Choquet-1955, Grabisch/Marichal/Mesiar/Pap "Aggregation Functions", Lovász-1983, Bach-2011.
- Read Tom McCormick's overview paper on SFM `http://people.commerce.ubc.ca/faculty/mccormick/sfmchap8a.pdf`
- Read chapters 1 - 4 from Fujishige book.
- Matroid properties `http://www-math.mit.edu/~goemans/18433S09/matroid-notes.pdf`
- Read lecture 14 slides on lattice theory at our web page (`http://j.ee.washington.edu/~bilmes/classes/ee596b_spring_2014/`)
- Wolfe "Finding the Nearest Point in a Polytope", 1976.
- Fujishige & Isotani, "A Submodular Function Minimization Algorithm Based on the Minimum-Norm Base", 2009.

Announcements, Assignments, and Reminders

- Weekly Office Hours: Wednesdays, 5:00-5:50, or by skype or google hangout (email me).

Class Road Map - IT-I

- L1 (3/31): Motivation, Applications, & Basic Definitions
- L2: (4/2): Applications, Basic Definitions, Properties
- L3: More examples and properties (e.g., closure properties), and examples, spanning trees
- L4: proofs of equivalent definitions, independence, start matroids
- L5: matroids, basic definitions and examples
- L6: More on matroids, System of Distinct Reps, Transversals, Transversal Matroid, Matroid and representation
- L7: Dual Matroids, other matroid properties, Combinatorial Geometries
- L8: Combinatorial Geometries, matroids and greedy, Polyhedra, Matroid Polytopes,
- L9: From Matroid Polytopes to Polymatroids.
- L10: Polymatroids and Submodularity
- L11: More properties of polymatroids, SFM special cases
- L12: polymatroid properties, extreme points polymatroids,
- L13: sat, dep, supp, exchange capacity, examples
- L14: Lattice theory: partially ordered sets; lattices; distributive, modular, submodular, and boolean lattices; ideals and join irreducibles.
- L15: Supp, Base polytope, polymatroids and entropic Venn diagrams, exchange capacity,
- L16: proof that minimum norm point yields min of submodular function, and the lattice of minimizers of a submodular function, Lovasz extension
- L17: Lovasz extension, Choquet Integration, more properties/examples of Lovasz extension, convex minimization and SFM.
- L18: Lovasz extension examples and structured convex norms, The Min-Norm Point Algorithm detailed.
- L19: symmetric submodular function minimization, maximizing monotone submodular function w. card constraints.
- L20: maximizing monotone submodular function w. other constraints, non-monotone maximization.

Finals Week: June 9th-13th, 2014.

Choquet integral

Definition 18.2.1

Let f be any capacity on E and $w \in \mathbb{R}_+^E$. The Choquet integral (1954) of w w.r.t. f is defined by

$$C_f(w) = \sum_{i=1}^{m} (w_{e_i} - w_{e_{i+1}}) f(E_i) \tag{18.12}$$

where in the sum, we have sorted and renamed the elements of E so that $w_{e_1} \geq w_{e_2} \geq \cdots \geq w_{e_m} \geq w_{e_{m+1}} \triangleq 0$, and where $E_i = \{e_1, e_2, \ldots, e_i\}$.

- We immediately see that an equivalent formula is as follows:

$$C_f(w) = \sum_{i=1}^{m} w(e_i)(f(E_i) - f(E_{i-1})) \tag{18.13}$$

where $E_0 \overset{\text{def}}{=} \emptyset$.

Lovász extension, as integral

- Additional ways we can define the Lovász extension for any (not necessarily submodular) but normalized function f include:

$$\tilde{f}(w) = \sum_{i=1}^{m} w(e_i) f(e_i | E_{i-1}) = \sum_{i=1}^{m} \lambda_i f(E_i) \tag{18.22}$$

$$= \sum_{i=1}^{m-1} f(E_i)(w(e_i) - w(e_{i+1})) + f(E) w(e_m) \tag{18.23}$$

$$= \int_{\min\{w_1,\ldots,w_m\}}^{+\infty} f(\{w \geq \alpha\}) d\alpha + f(E) \min\{w_1, \ldots, w_m\} \tag{18.24}$$

$$= \int_{0}^{+\infty} f(\{w \geq \alpha\}) d\alpha + \int_{-\infty}^{0} [f(\{w \geq \alpha\}) - f(E)] d\alpha \tag{18.25}$$

Lovász extension properties

- Using the above, have the following (some of which we've seen):

Theorem 18.2.2

Let $f, g : 2^E \to \mathbb{R}$ be normalized ($f(\emptyset) = g(\emptyset) = 0$). Then

1. *Superposition of LE operator: Given f and g with Lovász extensions $\tilde{f}$ and $\tilde{g}$ then $\tilde{f} + \tilde{g}$ is the Lovász extension of $f + g$ and $\lambda \tilde{f}$ is the Lovász extension of λf for $\lambda \in \mathbb{R}$.*
2. *If $w \in \mathbb{R}_+^E$ then $\tilde{f}(w) = \int_0^{+\infty} f(\{w \geq \alpha\}) d\alpha$.*
3. *For $w \in \mathbb{R}^E$, and $\alpha \in \mathbb{R}$, we have $\tilde{f}(w + \alpha \mathbf{1}_E) = \tilde{f}(w) + \alpha f(E)$.*
4. *Positive homogeneity: I.e., $\tilde{f}(\alpha w) = \alpha \tilde{f}(w)$ for $\alpha \geq 0$.*
5. *For all $A \subseteq E$, $\tilde{f}(\mathbf{1}_A) = f(A)$.*
6. *f symmetric as in $f(A) = f(E \setminus A), \forall A$, then $\tilde{f}(w) = \tilde{f}(-w)$ ($\tilde{f}$ is even).*
7. *Given partition $E^1 \cup E^2 \cup \dots \cup E^k$ of E and $w = \sum_{i=1}^k \gamma_i \mathbf{1}_{E_k}$ with $\gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_k$, and with $E^{1:i} = E^1 \cup E^2 \cup \dots \cup E^i$, then $\tilde{f}(w) = \sum_{i=1}^k \gamma_i f(E^i | E^{1:i-1}) = \sum_{i=1}^{k-1} f(E^{1:i})(\gamma_i - \gamma_{i+1}) + f(E)\gamma_k$.*

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

In fact, we have:

Theorem 18.2.5

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

Proof.

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

...

Simple expressions for Lovász E. with $m = 2$, $E = \{1, 2\}$

- If $w_1 \geq w_2$, then

$$\begin{aligned} \tilde{f}(w) &= w_1 f(\{1\}) + w_2 f(\{2\}|\{1\}) && (18.1) \\ &= (w_1 - w_2) f(\{1\}) + w_2 f(\{1, 2\}) && (18.2) \end{aligned}$$

Simple expressions for Lovász E. with $m = 2$, $E = \{1, 2\}$

- If $w_1 \geq w_2$, then

$$\tilde{f}(w) = w_1 f(\{1\}) + w_2 f(\{2\}|\{1\}) \tag{18.1}$$

$$= (w_1 - w_2) f(\{1\}) + w_2 f(\{1, 2\}) \tag{18.2}$$

- If $w_1 \leq w_2$, then

$$\tilde{f}(w) = w_2 f(\{2\}) + w_1 f(\{1\}|\{2\}) \tag{18.3}$$

$$= (w_2 - w_1) f(\{2\}) + w_1 f(\{1, 2\}) \tag{18.4}$$

Simple expressions for Lovász E. with $m = 2$, $E = \{1, 2\}$

- If $w_1 \geq w_2$, then

$$\begin{aligned}
\tilde{f}(w) &= w_1 f(\{1\}) + w_2 f(\{2\}|\{1\}) && (18.5)\\
&= (w_1 - w_2) f(\{1\}) + w_2 f(\{1,2\}) && (18.6)\\
&= \frac{1}{2} f(1)(w_1 - w_2) + \frac{1}{2} f(1)(w_1 - w_2) && (18.7)\\
&\quad + \frac{1}{2} f(\{1,2\})(w_1 + w_2) - \frac{1}{2} f(\{1,2\})(w_1 - w_2) && (18.8)\\
&\quad + \frac{1}{2} f(2)(w_1 - w_2) + \frac{1}{2} f(2)(w_2 - w_1) && (18.9)
\end{aligned}$$

Simple expressions for Lovász E. with $m = 2$, $E = \{1, 2\}$

- If $w_1 \geq w_2$, then

$$\begin{aligned}
\tilde{f}(w) &= w_1 f(\{1\}) + w_2 f(\{2\}|\{1\}) && (18.5)\\
&= (w_1 - w_2) f(\{1\}) + w_2 f(\{1,2\}) && (18.6)\\
&= \frac{1}{2} f(1)(w_1 - w_2) + \frac{1}{2} f(1)(w_1 - w_2) && (18.7)\\
&\quad + \frac{1}{2} f(\{1,2\})(w_1 + w_2) - \frac{1}{2} f(\{1,2\})(w_1 - w_2) && (18.8)\\
&\quad + \frac{1}{2} f(2)(w_1 - w_2) + \frac{1}{2} f(2)(w_2 - w_1) && (18.9)
\end{aligned}$$

- A similar (symmetric) expression holds when $w_1 \leq w_2$.

Simple expressions for Lovász E. with $m = 2$, $E = \{1, 2\}$

- This gives, for general w_1, w_2, that

$$\tilde{f}(w) = \frac{1}{2}\left(f(\{1\}) + f(\{2\}) - f(\{1,2\})\right)|w_1 - w_2| \tag{18.10}$$

$$+ \frac{1}{2}\left(f(\{1\}) - f(\{2\}) + f(\{1,2\})\right)w_1 \tag{18.11}$$

$$+ \frac{1}{2}\left(-f(\{1\}) + f(\{2\}) + f(\{1,2\})\right)w_2 \tag{18.12}$$

$$= -\left(f(\{1\}) + f(\{2\}) - f(\{1,2\})\right)\min\{w_1, w_2\} \tag{18.13}$$

$$+ f(\{1\})w_1 + f(\{2\})w_2 \tag{18.14}$$

Simple expressions for Lovász E. with $m = 2$, $E = \{1, 2\}$

- This gives, for general w_1, w_2, that

$$\tilde{f}(w) = \frac{1}{2}\left(f(\{1\}) + f(\{2\}) - f(\{1,2\})\right)|w_1 - w_2| \tag{18.10}$$
$$+ \frac{1}{2}\left(f(\{1\}) - f(\{2\}) + f(\{1,2\})\right)w_1 \tag{18.11}$$
$$+ \frac{1}{2}\left(-f(\{1\}) + f(\{2\}) + f(\{1,2\})\right)w_2 \tag{18.12}$$
$$= -\left(f(\{1\}) + f(\{2\}) - f(\{1,2\})\right)\min\{w_1, w_2\} \tag{18.13}$$
$$+ f(\{1\})w_1 + f(\{2\})w_2 \tag{18.14}$$

- Thus, if $f(A) = H(X_A)$ is the entropy function, we have $\tilde{f}(w) = H(e_1)w_1 + H(e_2)w_2 - I(e_1; e_2)\min\{w_1, w_2\}$ which must be convex in w, where $I(e_1; e_2)$ is the mutual information.

Simple expressions for Lovász E. with $m = 2$, $E = \{1, 2\}$

- This gives, for general w_1, w_2, that

$$\begin{aligned}
\tilde{f}(w) &= \frac{1}{2}\left(f(\{1\}) + f(\{2\}) - f(\{1,2\})\right)|w_1 - w_2| && (18.10)\\
&\quad + \frac{1}{2}\left(f(\{1\}) - f(\{2\}) + f(\{1,2\})\right)w_1 && (18.11)\\
&\quad + \frac{1}{2}\left(-f(\{1\}) + f(\{2\}) + f(\{1,2\})\right)w_2 && (18.12)\\
&= -\left(f(\{1\}) + f(\{2\}) - f(\{1,2\})\right)\min\{w_1, w_2\} && (18.13)\\
&\quad + f(\{1\})w_1 + f(\{2\})w_2 && (18.14)
\end{aligned}$$

- Thus, if $f(A) = H(X_A)$ is the entropy function, we have $\tilde{f}(w) = H(e_1)w_1 + H(e_2)w_2 - I(e_1; e_2)\min\{w_1, w_2\}$ which must be convex in w, where $I(e_1; e_2)$ is the mutual information.
- This "simple" but general form of the Lovász extension with $m = 2$ can be useful.

Example: $m = 2$, $E = \{1, 2\}$, contours

- If $w_1 \geq w_2$, then

$$\tilde{f}(w) = w_1 f(\{1\}) + w_2 f(\{2\}|\{1\}) \tag{18.15}$$

Example: $m = 2$, $E = \{1, 2\}$, contours

- If $w_1 \geq w_2$, then

$$\tilde{f}(w) = w_1 f(\{1\}) + w_2 f(\{2\}|\{1\}) \tag{18.15}$$

  - If $w = (1, 0)/f(\{1\}) = \big(1/f(\{1\}), 0\big)$ then $\tilde{f}(w) = 1$.

Example: $m = 2$, $E = \{1, 2\}$, contours

- If $w_1 \geq w_2$, then

$$\tilde{f}(w) = w_1 f(\{1\}) + w_2 f(\{2\}|\{1\}) \tag{18.15}$$

  - If $w = (1, 0)/f(\{1\}) = \Big(1/f(\{1\}), 0\Big)$ then $\tilde{f}(w) = 1$.
  - If $w = (1, 1)/f(\{1, 2\})$ then $\tilde{f}(w) = 1$.

Example: $m = 2$, $E = \{1, 2\}$, contours

- If $w_1 \geq w_2$, then

$$\tilde{f}(w) = w_1 f(\{1\}) + w_2 f(\{2\}|\{1\}) \tag{18.15}$$

  - If $w = (1, 0)/f(\{1\}) = \Big(1/f(\{1\}), 0\Big)$ then $\tilde{f}(w) = 1$.
  - If $w = (1, 1)/f(\{1, 2\})$ then $\tilde{f}(w) = 1$.

- If $w_1 \leq w_2$, then

$$\tilde{f}(w) = w_2 f(\{2\}) + w_1 f(\{1\}|\{2\}) \tag{18.16}$$

Example: $m = 2$, $E = \{1, 2\}$, contours

- If $w_1 \geq w_2$, then

$$\tilde{f}(w) = w_1 f(\{1\}) + w_2 f(\{2\}|\{1\}) \tag{18.15}$$

  - If $w = (1, 0)/f(\{1\}) = \Big(1/f(\{1\}), 0\Big)$ then $\tilde{f}(w) = 1$.
  - If $w = (1, 1)/f(\{1, 2\})$ then $\tilde{f}(w) = 1$.

- If $w_1 \leq w_2$, then

$$\tilde{f}(w) = w_2 f(\{2\}) + w_1 f(\{1\}|\{2\}) \tag{18.16}$$

  - If $w = (0, 1)/f(\{2\}) = (0, 1/f(\{2\}))$ then $\tilde{f}(w) = 1$.

Example: $m = 2$, $E = \{1,2\}$, contours

- If $w_1 \geq w_2$, then

$$\tilde{f}(w) = w_1 f(\{1\}) + w_2 f(\{2\}|\{1\}) \tag{18.15}$$

  - If $w = (1,0)/f(\{1\}) = \Big(1/f(\{1\}), 0\Big)$ then $\tilde{f}(w) = 1$.
  - If $w = (1,1)/f(\{1,2\})$ then $\tilde{f}(w) = 1$.

- If $w_1 \leq w_2$, then

$$\tilde{f}(w) = w_2 f(\{2\}) + w_1 f(\{1\}|\{2\}) \tag{18.16}$$

  - If $w = (0,1)/f(\{2\}) = (0, 1/f(\{2\}))$ then $\tilde{f}(w) = 1$.
  - If $w = (1,1)/f(\{1,2\})$ then $\tilde{f}(w) = 1$.

Example: $m = 2$, $E = \{1, 2\}$, contours

- If $w_1 \geq w_2$, then

$$\tilde{f}(w) = w_1 f(\{1\}) + w_2 f(\{2\}|\{1\}) \tag{18.15}$$

  - If $w = (1, 0)/f(\{1\}) = \Big(1/f(\{1\}), 0\Big)$ then $\tilde{f}(w) = 1$.
  - If $w = (1, 1)/f(\{1, 2\})$ then $\tilde{f}(w) = 1$.

- If $w_1 \leq w_2$, then

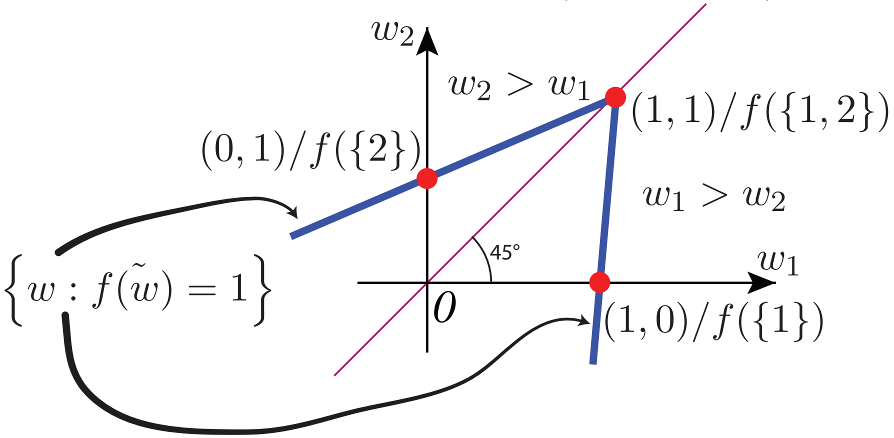
$$\tilde{f}(w) = w_2 f(\{2\}) + w_1 f(\{1\}|\{2\}) \tag{18.16}$$

  - If $w = (0, 1)/f(\{2\}) = (0, 1/f(\{2\}))$ then $\tilde{f}(w) = 1$.
  - If $w = (1, 1)/f(\{1, 2\})$ then $\tilde{f}(w) = 1$.

- Can plot contours of the form $\Big\{w \in \mathbb{R}^2 : \tilde{f}(w) = 1\Big\}$, particular marked points of form $w = \mathbf{1}_A \times \frac{1}{f(A)}$ for certain A, where $\tilde{f}(w) = 1$.

Example: $m = 2$, $E = \{1, 2\}$

- Contour plot of $m = 2$ Lovász extension (from Bach-2011).



Example: $m = 3$, $E = \{1, 2, 3\}$

- In order to visualize in 3D, we make a few simplifications.

Example: $m = 3$, $E = \{1, 2, 3\}$

- In order to visualize in 3D, we make a few simplifications.
- Consider any submodular f' and $x \in B_{f'}$. Then $f(A) = f'(A) - x(A)$ is submodular

Example: $m = 3$, $E = \{1, 2, 3\}$

- In order to visualize in 3D, we make a few simplifications.
- Consider any submodular f' and $x \in B_{f'}$. Then $f(A) = f'(A) - x(A)$ is submodular, and moreover $f(E) = f'(E) - x(E) = 0$.

Example: $m = 3$, $E = \{1, 2, 3\}$

- In order to visualize in 3D, we make a few simplifications.
- Consider any submodular f' and $x \in B_{f'}$. Then $f(A) = f'(A) - x(A)$ is submodular, and moreover $f(E) = f'(E) - x(E) = 0$.
- Hence, from $\tilde{f}(w + \alpha \mathbf{1}_E) = \tilde{f}(w) + \alpha f(E)$, we have that $\tilde{f}(w + \alpha \mathbf{1}_E) = \tilde{f}(w)$.

Example: $m = 3$, $E = \{1, 2, 3\}$

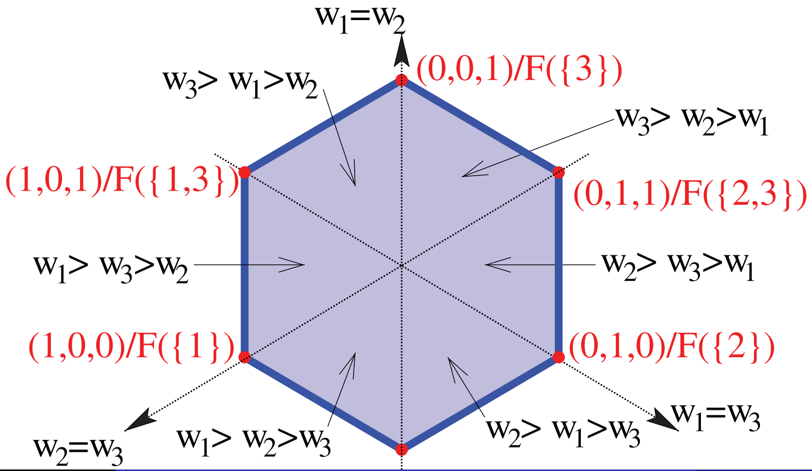
- In order to visualize in 3D, we make a few simplifications.
- Consider any submodular f' and $x \in B_{f'}$. Then $f(A) = f'(A) - x(A)$ is submodular, and moreover $f(E) = f'(E) - x(E) = 0$.
- Hence, from $\tilde{f}(w + \alpha \mathbf{1}_E) = \tilde{f}(w) + \alpha f(E)$, we have that $\tilde{f}(w + \alpha \mathbf{1}_E) = \tilde{f}(w)$.
- Thus, we can look "down" on the contour plot of the Lovász extension, $\left\{ w : \tilde{f}(w) = 1 \right\}$, from a vantage point right on the line $\{x : x = \alpha \mathbf{1}_E, \alpha > 0\}$ since moving in direction $\mathbf{1}_E$ changes nothing.

Example: $m = 3$, $E = \{1, 2, 3\}$

- Example 1 (from Bach-2011): $f(A) = \mathbf{1}_{|A| \in \{1,2\}}$ $= \min\{|A|, 1\} + \min\{|E \setminus A|, 1\} - 1$ is submodular, and $\tilde{f}(w) = \max_{k \in \{1,2,3\}} w_k - \min_{k \in \{1,2,3\}} w_k$.

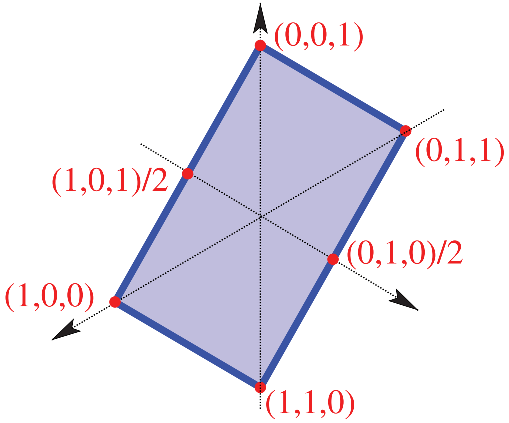
Example: $m = 3$, $E = \{1, 2, 3\}$

- Example 1 (from Bach-2011): $f(A) = \mathbf{1}_{|A| \in \{1,2\}}$ $= \min\{|A|, 1\} + \min\{|E \setminus A|, 1\} - 1$ is submodular, and $\tilde{f}(w) = \max_{k \in \{1,2,3\}} w_k - \min_{k \in \{1,2,3\}} w_k$.



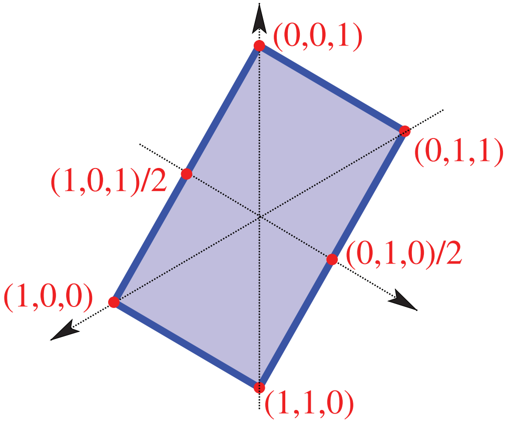
Example: $m = 3$, $E = \{1, 2, 3\}$

- Example 2 (from Bach-2011): $f(A) = |\mathbf{1}_{1\in A} - \mathbf{1}_{2\in A}| + |\mathbf{1}_{2\in A} - \mathbf{1}_{3\in A}|$



Example: $m = 3$, $E = \{1, 2, 3\}$

- Example 2 (from Bach-2011): $f(A) = |\mathbf{1}_{1 \in A} - \mathbf{1}_{2 \in A}| + |\mathbf{1}_{2 \in A} - \mathbf{1}_{3 \in A}|$
- This gives a "total variation" function for the Lovász extension, with $\tilde{f}(w) = |w_1 - w_2| + |w_2 - w_3|$, a prior to prefer piecewise-constant signals.



Total Variation Example

From "Nonlinear total variation based noise removal algorithms" Rudin, Osher, and Fatemi, 1992. Top left original, bottom right total variation.

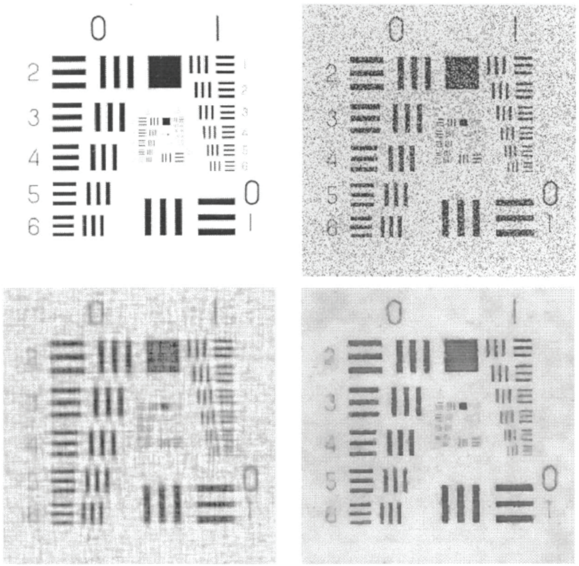


Fig. 3. (a) "Resolution Chart". (b) Noisy "Resolution Chart", SNR = 1.0. (c) Wiener filter reconstruction from (b). (d) TV reconstruction from (b).

Example: Lovász extension of concave over modular

- Let $m : E \to \mathbb{R}_+$ be a modular function and define $f(A) = g(m(A))$ where g is concave. Then f is submodular.

Example: Lovász extension of concave over modular

- Let $m : E \to \mathbb{R}_+$ be a modular function and define $f(A) = g(m(A))$ where g is concave. Then f is submodular.
- Let $M_j = \sum_{i=1}^{j} m(e_i)$

Example: Lovász extension of concave over modular

- Let $m : E \to \mathbb{R}_+$ be a modular function and define $f(A) = g(m(A))$ where g is concave. Then f is submodular.
- Let $M_j = \sum_{i=1}^{j} m(e_i)$
- $\tilde{f}(w)$ is given as

$$\tilde{f}(w) = \sum_{i=1}^{m} w(e_i)\big(g(M_i) - g(M_{i-1})\big) \tag{18.17}$$

Example: Lovász extension of concave over modular

- Let $m : E \to \mathbb{R}_+$ be a modular function and define $f(A) = g(m(A))$ where g is concave. Then f is submodular.
- Let $M_j = \sum_{i=1}^{j} m(e_i)$
- $\tilde{f}(w)$ is given as

$$\tilde{f}(w) = \sum_{i=1}^{m} w(e_i)\big(g(M_i) - g(M_{i-1})\big) \tag{18.17}$$

- And if $m(A) = |A|$, we get

$$\tilde{f}(w) = \sum_{i=1}^{m} w(e_i)\big(g(i) - g(i-1)\big) \tag{18.18}$$

Example: Lovász extension and cut functions

- Cut Function: Given a non-negative weighted graph $G = (V, E, m)$ where $m : E \to \mathbb{R}_+$ is a modular function over the edges, we know from Lecture 2 that $f : 2^V \to \mathbb{R}_+$ with $f(X) = m(\Gamma(X))$ where $\Gamma(X) = \{(u, v) | (u, v) \in E, u \in X, v \in V \setminus X\}$ is non-monotone submodular.

Example: Lovász extension and cut functions

- Cut Function: Given a non-negative weighted graph $G = (V, E, m)$ where $m : E \to \mathbb{R}_+$ is a modular function over the edges, we know from Lecture 2 that $f : 2^V \to \mathbb{R}_+$ with $f(X) = m(\Gamma(X))$ where $\Gamma(X) = \{(u, v) | (u, v) \in E, u \in X, v \in V \setminus X\}$ is non-monotone submodular.
- Simple way to write it, with $m_{ij} = m((i, j))$:

$$f(X) = \sum_{i \in X, j \in V \setminus X} m_{ij} \tag{18.19}$$

Example: Lovász extension and cut functions

- Cut Function: Given a non-negative weighted graph $G = (V, E, m)$ where $m : E \to \mathbb{R}_+$ is a modular function over the edges, we know from Lecture 2 that $f : 2^V \to \mathbb{R}_+$ with $f(X) = m(\Gamma(X))$ where $\Gamma(X) = \{(u, v) | (u, v) \in E, u \in X, v \in V \setminus X\}$ is non-monotone submodular.
- Simple way to write it, with $m_{ij} = m((i, j))$:

$$f(X) = \sum_{i \in X, j \in V \setminus X} m_{ij} \tag{18.19}$$

- Exercise: show that Lovász extension of graph cut may be written as:

$$\tilde{f}(w) = \sum_{i,j \in V} m_{ij} \max\{(w_i - w_j), 0\} \tag{18.20}$$

where elements are ordered as usual, $w_1 \geq w_2 \geq \cdots \geq w_n$.

Example: Lovász extension and cut functions

- Cut Function: Given a non-negative weighted graph $G = (V, E, m)$ where $m : E \to \mathbb{R}_+$ is a modular function over the edges, we know from Lecture 2 that $f : 2^V \to \mathbb{R}_+$ with $f(X) = m(\Gamma(X))$ where $\Gamma(X) = \{(u, v) | (u, v) \in E, u \in X, v \in V \setminus X\}$ is non-monotone submodular.
- Simple way to write it, with $m_{ij} = m((i, j))$:

$$f(X) = \sum_{i \in X, j \in V \setminus X} m_{ij} \tag{18.19}$$

- Exercise: show that Lovász extension of graph cut may be written as:

$$\tilde{f}(w) = \sum_{i,j \in V} m_{ij} \max \{(w_i - w_j), 0\} \tag{18.20}$$

  where elements are ordered as usual, $w_1 \geq w_2 \geq \cdots \geq w_n$.
- This is also a form of "total variation"

A few more Lovász extension examples

Some additional submodular functions and their Lovász extensions, where $w(e_1) \geq w(e_2) \geq \cdots \geq w(e_m) \geq 0$. Let $W_k \triangleq \sum_{i=1}^{k} w(e_i)$.

$f(A)$	$\tilde{f}(w)$
$\lvert A\rvert$	$\lVert w\rVert_1$
$\min(\lvert A\rvert, 1)$	$\lVert w\rVert_\infty$
$\min(\lvert A\rvert, 1) - \max(\lvert A\rvert - m + 1, 0)$	$\lVert w\rVert_\infty - \min_i w_i$
$\min(\lvert A\rvert, k)$	W_k
$\min(\lvert A\rvert, k) - \max(\lvert A\rvert - (n-k) + 1, 1)$	$2W_k - W_m$
$\min(\lvert A\rvert, \lvert E \setminus A\rvert)$	$2W_{\lfloor m/2 \rfloor} - W_m$

(thanks to K. Narayanan).

Supervised And Unsupervised Machine Learning

- Given training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$ with $(x_i, y_i) \in \mathbb{R}^n \times \mathbb{R}$, perform the following risk minimization problem:

$$\min_{w \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^m \ell(y_i, w^\intercal x_i) + \lambda \Omega(w), \tag{18.21}$$

  where $\ell(\cdot)$ is a loss function (e.g., squared error) and $\Omega(w)$ is a norm.
- When
  data has multiple responses $(x_i, y_i) \in \mathbb{R}^n \times \mathbb{R}^k$, learning becomes:

$$\min_{w^1, \dots, w^k \in \mathbb{R}^n} \sum_{j=1}^k \frac{1}{m} \sum_{i=1}^m \ell(y_i^k, (w^k)^\intercal x_i) + \lambda \Omega(w^k), \tag{18.22}$$

- When data has multiple responses only that are observed, $(y_i) \in R^k$ we get dictionary learning (Krause & Guestrin, Das & Kempe):

$$\min_{x_1, \dots, x_m} \min_{w^1, \dots, w^k \in \mathbb{R}^n} \sum_{j=1}^k \frac{1}{m} \sum_{i=1}^m \ell(y_i^k, (w^k)^\intercal x_i) + \lambda \Omega(w^k), \tag{18.23}$$

Norms, sparse norms, and computer vision

- Common norms include p-norm $\Omega(w) = \|w\|_p = \left(\sum_{i=1}^{p} w_i^p\right)^{1/p}$

Norms, sparse norms, and computer vision

- Common norms include p-norm $\Omega(w) = \|w\|_p = \left(\sum_{i=1}^{p} w_i^p\right)^{1/p}$
- 1-norm promotes sparsity (prefer solutions with zero entries).

Norms, sparse norms, and computer vision

- Common norms include p-norm $\Omega(w) = \|w\|_p = \left(\sum_{i=1}^p w_i^p\right)^{1/p}$
- 1-norm promotes sparsity (prefer solutions with zero entries).
- Image denoising, total variation is useful, norm takes form:

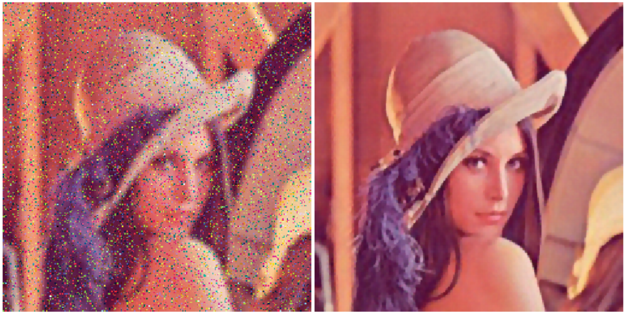
$$\Omega(w) = \sum_{i=2}^{N} |w_i - w_{i-1}| \tag{18.24}$$

Norms, sparse norms, and computer vision

- Common norms include p-norm $\Omega(w) = \|w\|_p = \left(\sum_{i=1}^p w_i^p\right)^{1/p}$
- 1-norm promotes sparsity (prefer solutions with zero entries).
- Image denoising, total variation is useful, norm takes form:

$$\Omega(w) = \sum_{i=2}^{N} |w_i - w_{i-1}| \tag{18.24}$$

- Points of difference should be "sparse" (frequently zero).



(Rodriguez, 2009)

Submodular parameterization of a sparse convex norm

- Prefer convex norms since they can be solved.

Submodular parameterization of a sparse convex norm

- Prefer convex norms since they can be solved.
- For $w \in \mathbb{R}^V$, $\text{supp}(w) \in \{0,1\}^V$ has $\text{supp}(w)(v) = 1$ iff $w(v) > 0$

Submodular parameterization of a sparse convex norm

- Prefer convex norms since they can be solved.
- For $w \in \mathbb{R}^V$, $\text{supp}(w) \in \{0,1\}^V$ has $\text{supp}(w)(v) = 1$ iff $w(v) > 0$
- Desirable sparse norm: count the non-zeros, $\|w\|_0 = \mathbf{1}^\intercal \text{supp}(w)$.

Submodular parameterization of a sparse convex norm

- Prefer convex norms since they can be solved.
- For $w \in \mathbb{R}^V$, $\text{supp}(w) \in \{0,1\}^V$ has $\text{supp}(w)(v) = 1$ iff $w(v) > 0$
- Desirable sparse norm: count the non-zeros, $\|w\|_0 = \mathbf{1}^\intercal \text{supp}(w)$.
- Using $\Omega(w) = \|w\|_0$ is NP-hard, instead we often optimize tightest convex relaxation, $\|w\|_1$ which is the convex envelope.

Submodular parameterization of a sparse convex norm

- Prefer convex norms since they can be solved.
- For $w \in \mathbb{R}^V$, $\text{supp}(w) \in \{0,1\}^V$ has $\text{supp}(w)(v) = 1$ iff $w(v) > 0$
- Desirable sparse norm: count the non-zeros, $\|w\|_0 = \mathbf{1}^\intercal \text{supp}(w)$.
- Using $\Omega(w) = \|w\|_0$ is NP-hard, instead we often optimize tightest convex relaxation, $\|w\|_1$ which is the convex envelope.
- With $\|w\|_0$ or its relaxation, each non-zero element has equal degree of penalty. Penalties do not interact.

Submodular parameterization of a sparse convex norm

- Prefer convex norms since they can be solved.
- For $w \in \mathbb{R}^V$, $\text{supp}(w) \in \{0,1\}^V$ has $\text{supp}(w)(v) = 1$ iff $w(v) > 0$
- Desirable sparse norm: count the non-zeros, $\|w\|_0 = \mathbf{1}^\intercal \text{supp}(w)$.
- Using $\Omega(w) = \|w\|_0$ is NP-hard, instead we often optimize tightest convex relaxation, $\|w\|_1$ which is the convex envelope.
- With $\|w\|_0$ or its relaxation, each non-zero element has equal degree of penalty. Penalties do not interact.
- Given submodular function $f : 2^V \to \mathbb{R}_+$, $f(\text{supp}(w))$ measures the "complexity" of the non-zero pattern of w; can have more non-zero values if they cooperate (via f) with other non-zero values.

Submodular parameterization of a sparse convex norm

- Prefer convex norms since they can be solved.
- For $w \in \mathbb{R}^V$, $\text{supp}(w) \in \{0,1\}^V$ has $\text{supp}(w)(v) = 1$ iff $w(v) > 0$
- Desirable sparse norm: count the non-zeros, $\|w\|_0 = \mathbf{1}^\intercal \text{supp}(w)$.
- Using $\Omega(w) = \|w\|_0$ is NP-hard, instead we often optimize tightest convex relaxation, $\|w\|_1$ which is the convex envelope.
- With $\|w\|_0$ or its relaxation, each non-zero element has equal degree of penalty. Penalties do not interact.
- Given submodular function $f : 2^V \to \mathbb{R}_+$, $f(\text{supp}(w))$ measures the "complexity" of the non-zero pattern of w; can have more non-zero values if they cooperate (via f) with other non-zero values.
- $f(\text{supp}(w))$ is hard to optimize, but it's convex envelope $\tilde{f}(|w|)$ (i.e., largest convex under-estimator of $f(\text{supp}(w))$) is obtained via the Lovász-extension $\tilde{f}$ of f (Vondrák 2007, Bach 2010).

Submodular parameterization of a sparse convex norm

- Prefer convex norms since they can be solved.
- For $w \in \mathbb{R}^V$, $\text{supp}(w) \in \{0,1\}^V$ has $\text{supp}(w)(v) = 1$ iff $w(v) > 0$
- Desirable sparse norm: count the non-zeros, $\|w\|_0 = \mathbf{1}^\intercal \text{supp}(w)$.
- Using $\Omega(w) = \|w\|_0$ is NP-hard, instead we often optimize tightest convex relaxation, $\|w\|_1$ which is the convex envelope.
- With $\|w\|_0$ or its relaxation, each non-zero element has equal degree of penalty. Penalties do not interact.
- Given submodular function $f : 2^V \to \mathbb{R}_+$, $f(\text{supp}(w))$ measures the "complexity" of the non-zero pattern of w; can have more non-zero values if they cooperate (via f) with other non-zero values.
- $f(\text{supp}(w))$ is hard to optimize, but it's convex envelope $\tilde{f}(|w|)$ (i.e., largest convex under-estimator of $f(\text{supp}(w))$) is obtained via the Lovász-extension $\tilde{f}$ of f (Vondrák 2007, Bach 2010).
- Submodular functions thus parameterize structured convex sparse norms via the Lovász-extension!

Submodular parameterization of a sparse convex norm

- Prefer convex norms since they can be solved.
- For $w \in \mathbb{R}^V$, $\text{supp}(w) \in \{0,1\}^V$ has $\text{supp}(w)(v) = 1$ iff $w(v) > 0$
- Desirable sparse norm: count the non-zeros, $\|w\|_0 = \mathbf{1}^\intercal \text{supp}(w)$.
- Using $\Omega(w) = \|w\|_0$ is NP-hard, instead we often optimize tightest convex relaxation, $\|w\|_1$ which is the convex envelope.
- With $\|w\|_0$ or its relaxation, each non-zero element has equal degree of penalty. Penalties do not interact.
- Given submodular function $f : 2^V \to \mathbb{R}_+$, $f(\text{supp}(w))$ measures the "complexity" of the non-zero pattern of w; can have more non-zero values if they cooperate (via f) with other non-zero values.
- $f(\text{supp}(w))$ is hard to optimize, but it's convex envelope $\tilde{f}(|w|)$ (i.e., largest convex under-estimator of $f(\text{supp}(w))$) is obtained via the Lovász-extension $\tilde{f}$ of f (Vondrák 2007, Bach 2010).
- Submodular functions thus parameterize structured convex sparse norms via the Lovász-extension!
- Ex: total variation is Lovász-ext. of graph cut, but $\exists$ many more!

Lovász extension and norms

- Using Lovász extension to define various norms of the form $\|w\|_{\tilde{f}} = \tilde{f}(|w|)$, renders the function symmetric about all orthants (i.e., $\|w\|_{\tilde{f}} = \|b \odot w\|_{\tilde{f}}$ where $b \in \{-1, 1\}^m$ and $\odot$ is element-wise multiplication).

Lovász extension and norms

- Using Lovász extension to define various norms of the form $\|w\|_{\tilde{f}} = \tilde{f}(|w|)$, renders the function symmetric about all orthants (i.e., $\|w\|_{\tilde{f}} = \|b \odot w\|_{\tilde{f}}$ where $b \in \{-1, 1\}^m$ and $\odot$ is element-wise multiplication).
- Simple example. The Lovász extension of the modular function $f(A) = |A|$ is the ℓ_1 norm, and the Lovász extension of the modular function $f(A) = m(A)$ is the weighted ℓ_1 norm.

Lovász extension and norms

- Using Lovász extension to define various norms of the form $\|w\|_{\tilde{f}} = \tilde{f}(|w|)$, renders the function symmetric about all orthants (i.e., $\|w\|_{\tilde{f}} = \|b \odot w\|_{\tilde{f}}$ where $b \in \{-1, 1\}^m$ and $\odot$ is element-wise multiplication).
- Simple example. The Lovász extension of the modular function $f(A) = |A|$ is the ℓ_1 norm, and the Lovász extension of the modular function $f(A) = m(A)$ is the weighted ℓ_1 norm.
- With more general submodular functions, one can generate a large and interesting variety of norms, all of which have polyhedral contours (unlike, say, something like the ℓ_2 norm).

Lovász extension and norms

- Using Lovász extension to define various norms of the form $\|w\|_{\tilde{f}} = \tilde{f}(|w|)$, renders the function symmetric about all orthants (i.e., $\|w\|_{\tilde{f}} = \|b \odot w\|_{\tilde{f}}$ where $b \in \{-1, 1\}^m$ and $\odot$ is element-wise multiplication).
- Simple example. The Lovász extension of the modular function $f(A) = |A|$ is the ℓ_1 norm, and the Lovász extension of the modular function $f(A) = m(A)$ is the weighted ℓ_1 norm.
- With more general submodular functions, one can generate a large and interesting variety of norms, all of which have polyhedral contours (unlike, say, something like the ℓ_2 norm).
- Hence, not all norms come from the Lovász extension of some submodular function.

Lovász extension and norms

- Using Lovász extension to define various norms of the form $\|w\|_{\tilde{f}} = \tilde{f}(|w|)$, renders the function symmetric about all orthants (i.e., $\|w\|_{\tilde{f}} = \|b \odot w\|_{\tilde{f}}$ where $b \in \{-1, 1\}^m$ and $\odot$ is element-wise multiplication).
- Simple example. The Lovász extension of the modular function $f(A) = |A|$ is the ℓ_1 norm, and the Lovász extension of the modular function $f(A) = m(A)$ is the weighted ℓ_1 norm.
- With more general submodular functions, one can generate a large and interesting variety of norms, all of which have polyhedral contours (unlike, say, something like the ℓ_2 norm).
- Hence, not all norms come from the Lovász extension of some submodular function.
- Similarly, not all convex functions are the Lovász extension of some submodular function.

Lovász extension and norms

- Using Lovász extension to define various norms of the form $\|w\|_{\tilde{f}} = \tilde{f}(|w|)$, renders the function symmetric about all orthants (i.e., $\|w\|_{\tilde{f}} = \|b \odot w\|_{\tilde{f}}$ where $b \in \{-1, 1\}^m$ and $\odot$ is element-wise multiplication).
- Simple example. The Lovász extension of the modular function $f(A) = |A|$ is the ℓ_1 norm, and the Lovász extension of the modular function $f(A) = m(A)$ is the weighted ℓ_1 norm.
- With more general submodular functions, one can generate a large and interesting variety of norms, all of which have polyhedral contours (unlike, say, something like the ℓ_2 norm).
- Hence, not all norms come from the Lovász extension of some submodular function.
- Similarly, not all convex functions are the Lovász extension of some submodular function.
- Bach-2011 has a complete discussion of this.

Review

The following four slides are review, and are from Lectures 12, 15, and 16.

A polymatroid function's polyhedron is a polymatroid.

Theorem 18.4.1

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

$$\textit{rank}(x) = \max\left(y(E) : y \leq x, y \in P_f\right) = \min\left(x(A) + f(E \setminus A) : A \subseteq E\right) \tag{18.5}$$

Essentially the same theorem as Theorem **??**. Taking $x = 0$ we get:

Corollary 18.4.2

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

$$\textit{rank}(0) = \max\left(y(E) : y \leq 0, y \in P_f\right) = \min\left(f(A) : A \subseteq E\right) \tag{18.6}$$

Min-Norm Point: Definition

- Restating what we saw before, we have:

$$\max\{y(E) | y \in P_f, y \leq 0\} = \min\{f(X) | X \subseteq V\} \tag{18.12}$$

- Consider the optimization:

$$\begin{aligned} &\text{minimize} && \|x\|_2^2 && (18.13a)\\ &\text{subject to} && x \in B_f && (18.13b)\end{aligned}$$

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

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

Min-Norm Point and Submodular Function Minimization

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

$$\begin{aligned} y^* &= x^* \wedge 0 = (\min(x^*(e), 0) | e \in E) && (18.1)\\ A_- &= \{e : x^*(e) < 0\} && (18.2)\\ A_0 &= \{e : x^*(e) \leq 0\} && (18.3) \end{aligned}$$

- Thus, we immediately have that:

$$A_- \subseteq A_0 \qquad (18.4)$$

and that

$$x^*(A_-) = x^*(A_0) = y^*(A_-) = y^*(A_0) \qquad (18.5)$$

- 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.

Min-Norm Point and SFM

Theorem 18.4.1

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

Proof.

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

$$(x^* + \alpha \mathbf{1}_e - \alpha \mathbf{1}_{e'})(E) = x^*(E) + \alpha - \alpha = f(E) \qquad (18.1)$$

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

...

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

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

$$\underset{w \in \mathbb{R}^V}{\text{minimize}} \; \tilde{f}(w) + \frac{1}{2}\|w\|_2^2 \tag{18.25}$$

$$\begin{aligned} &\text{maximize} && -\|x\|_2^2 && (18.26a) \\ &\text{subject to} && x \in B_f && (18.26b) \end{aligned}$$

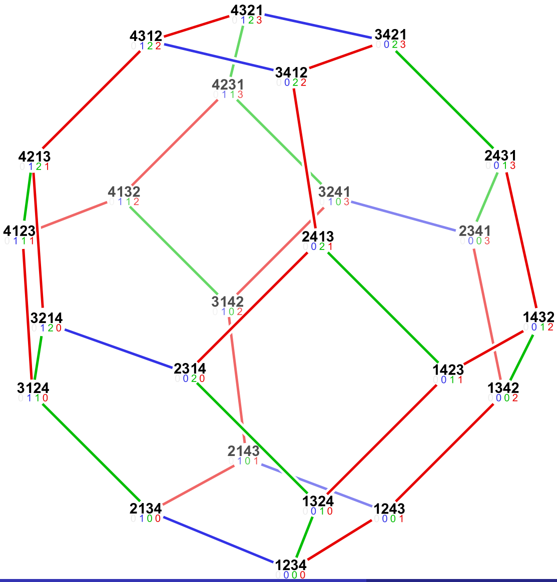
  where $B_f = P_f \cap \left\{x \in \mathbb{R}^V : x(V) = f(V)\right\}$ is the base polytope of submodular function f, and $\|x\|_2^2 = \sum_{e \in V} x(e)^2$ is squared 2-norm.
- Equation (18.25) is related to proximal methods to minimize the Lovász extension (see Parikh&Boyd, "Proximal Algorithms" 2013).
- Equation (18.26b) is solved by the minimum-norm point algorithm (Wolfe-1976, Fujishige-1984, Fujishige-2005, Fujishige-2011) is (as we will see) essentially an active-set procedure for quadratic programming, and uses Edmonds's greedy algorithm to make it efficient.
- Unknown worst-case running time, although in practice it usually performs quite well (see below).

Ex: 3D base B_f: permutahedron

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

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

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



Modified max-min theorem

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

Modified max-min theorem

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

Theorem 18.4.1 (Edmonds-1970)

$$\min\{f(X) | X \subseteq E\} = \max\left\{x^-(E) | x \in B_f\right\} \tag{18.27}$$

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

Modified max-min theorem

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

Theorem 18.4.1 (Edmonds-1970)

$$\min \{f(X) | X \subseteq E\} = \max \left\{x^-(E) | x \in B_f\right\} \tag{18.27}$$

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

Proof.

$$\begin{aligned}
\min \{f(X) | X \subseteq E\} &= \min_{w \in [0,1]^E} \tilde{f}(w) = \min_{w \in [0,1]^E} \max_{x \in P_f} w^\intercal x && (18.28)\\
&= \min_{w \in [0,1]^E} \max_{x \in B_f} w^\intercal x && (18.29)\\
&= \max_{x \in B_f} \min_{w \in [0,1]^E} w^\intercal x && (18.30)\\
&= \max_{x \in B_f} x^-(E) && (18.31)
\end{aligned}$$

Convexity, Strong duality, and min/max swap

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

$$\min_{w' \in [0,1]^E} g(w', x) \leq g(w, x) \leq \max_{x' \in B_f} g(w, x') \tag{18.32}$$

which means that we have weak duality

$$\max_{x \in B_f} \min_{w' \in [0,1]^E} g(w', x) \leq \min_{w \in [0,1]^E} \max_{x' \in B_f} g(w, x') \tag{18.33}$$

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

$$\max_{x \in B_f} \min_{w' \in [0,1]^E} g(w', x) = \min_{w \in [0,1]^E} \max_{x' \in B_f} g(w, x') \tag{18.34}$$

Alternate proof of modified max-min theorem

We start directly from Theorem 12.5.2.

$$\max\left(y(E) : y \le 0, y \in P_f\right) = \min\left(f(A) : A \subseteq E\right) \tag{18.35}$$

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

$$\begin{aligned}\max\left(y(E) : y \le 0, y \in P_f\right) &= \max\left(y^-(E) : y \le 0, y \in P_f\right) && (18.36)\\ &= \max\left(y^-(E) : y \in P_f\right) && (18.37)\\ &= \max\left(y^-(E) : y \in B_f\right) && (18.38)\end{aligned}$$

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

$$\begin{aligned}\max\left(y^-(E) : y \in P_f\right) &= \max\left(y^-(E) : y(A) \le f(A) \forall A\right) && (18.39)\\ &= \max\left(y^-(E) : y^-(A) + y^+(A) \le f(A) \forall A\right)\end{aligned}$$

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

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

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

$$\max\{w^\intercal x | x \in P_f\} = \max\{w^\intercal x | x \in B_f\} \tag{18.40}$$

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

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

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

$$\max\{w^\intercal x | x \in P_f\} = \max\{w^\intercal x | x \in B_f\} \tag{18.40}$$

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

$$\max\{w^\intercal x | x \in B_f\} \tag{18.41}$$

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

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

$$\max \{w^\intercal x | x \in P_f\} = \max \{w^\intercal x | x \in B_f\} \tag{18.40}$$

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

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

$$\max \{w^\intercal x | x \in B_f\} \tag{18.41}$$

- Also, since

$$\min \{w^\intercal x | x \in B_f\} = -\max \{-w^\intercal x | x \in B_f\} \tag{18.42}$$

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

$$w(e_1) \leq w(e_2) \leq \dots \leq w(e_m) \tag{18.43}$$

will solve Equation (18.42).

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

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

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

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

Orthogonal x-containing hyperplane & convex/affine hulls

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

$$H(x) \triangleq \left\{ y \in \mathbb{R}^V \mid x^\intercal y = \|x\|_2^2 \right\} \tag{18.44}$$

Any set $\{y \in \mathbb{R}^V | x^\intercal y = c\}$ is orthogonal to the line from 0 to x. To also contain x, we need $\|x\|_2 \|x\|_2 \cos 0 = c$ giving $c = \|x\|_2^2$.

Orthogonal x-containing hyperplane & convex/affine hulls

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

$$H(x) \triangleq \left\{ y \in \mathbb{R}^V \mid x^\intercal y = \|x\|_2^2 \right\} \tag{18.44}$$

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

$$\operatorname{conv} P \triangleq \left\{ \sum_{i=1}^k \lambda_i p_i : \sum_i \lambda_i = 1, \ \lambda_i \geq 0, i \in [k] \right\}. \tag{18.45}$$

Orthogonal x-containing hyperplane & convex/affine hulls

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

$$H(x) \triangleq \left\{ y \in \mathbb{R}^V \mid x^\intercal y = \|x\|_2^2 \right\} \tag{18.44}$$

  Any set $\left\{ y \in \mathbb{R}^V | x^\intercal y = c \right\}$ is orthogonal to the line from 0 to x. To also contain x, we need $\|x\|_2 \|x\|_2 \cos 0 = c$ giving $c = \|x\|_2^2$.

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

$$\text{conv}\, P \triangleq \left\{ \sum_{i=1}^k \lambda_i p_i : \sum_i \lambda_i = 1,\ \lambda_i \geq 0, i \in [k] \right\}. \tag{18.45}$$

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

$$\text{aff}\, Q \triangleq \left\{ \sum_{i \in 1}^k \lambda_i q_i : \sum_{i=1}^k \lambda_i = 1 \right\} \tag{18.46}$$

Orthogonal x-containing hyperplane & convex/affine hulls

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

$$H(x) \triangleq \left\{ y \in \mathbb{R}^V \mid x^\intercal y = \|x\|_2^2 \right\} \tag{18.44}$$

  Any set $\left\{ y \in \mathbb{R}^V | x^\intercal y = c \right\}$ is orthogonal to the line from 0 to x. To also contain x, we need $\|x\|_2 \|x\|_2 \cos 0 = c$ giving $c = \|x\|_2^2$.

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

$$\text{conv}\, P \triangleq \left\{ \sum_{i=1}^k \lambda_i p_i : \sum_i \lambda_i = 1,\ \lambda_i \geq 0, i \in [k] \right\}. \tag{18.45}$$

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

$$\text{aff}\, Q \triangleq \left\{ \sum_{i \in 1}^k \lambda_i q_i : \sum_{i=1}^k \lambda_i = 1 \right\} \supseteq \text{conv}\, Q. \tag{18.46}$$

Notation

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

Notation

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

Fujishige-Wolfe Min-Norm Algorithm

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

$$\min_{x \in \text{conv } P} \|x\|_2 \tag{18.47}$$

Fujishige-Wolfe Min-Norm Algorithm

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

$$\min_{x \in \text{conv}\, P} \|x\|_2 \tag{18.47}$$

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

$$\min_{x \in \text{conv } P} \|x\|_2 \tag{18.47}$$

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

Fujishige-Wolfe Min-Norm Algorithm

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

Fujishige-Wolfe Min-Norm Algorithm

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

Fujishige-Wolfe Min-Norm Algorithm

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

Fujishige-Wolfe Min-Norm Algorithm

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

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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example

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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example

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

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{18.48}$$

must hold at every possible assignment of x^* (Lines 1, 11, and 16):

1. True after Line 1 since $Q = \{x^*\}$,
2. True after Line 11 since $x_0 \in \operatorname{conv} Q$,
3. and true after Line 16 since $y \in \operatorname{conv} Q$ even after deleting points.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example

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

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{18.48}$$

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

$$\min_{x \in \operatorname{aff} Q} \|x\|_2 \leq \min_{x \in \operatorname{conv} Q} \|x\|_2 \leq \|x^*\|_2 \tag{18.49}$$

Fujishige-Wolfe Min-Norm algorithm: Geometric Example

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

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{18.48}$$

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

$$\min_{x \in \operatorname{aff} Q} \|x\|_2 \leq \min_{x \in \operatorname{conv} Q} \|x\|_2 \leq \|x^*\|_2 \tag{18.49}$$

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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example

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

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{18.48}$$

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

$$\min_{x \in \operatorname{aff} Q} \|x\|_2 \leq \min_{x \in \operatorname{conv} Q} \|x\|_2 \leq \|x^*\|_2 \tag{18.49}$$

- Note, the input, P, consists of m points. In the case of the base polytope, $P = B_f$ could be exponential in $n = |V|$.
- There are six places that might be seemingly tricky or expensive: Line 4, Line 6, Line 9, Line 10, Line 14, and Line 15.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example

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

$$x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P, \tag{18.48}$$

must hold at every possible assignment of x^* (Lines 1, 11, and 16):

1. True after Line 1 since $Q = \{x^*\}$,
2. True after Line 11 since $x_0 \in \operatorname{conv} Q$,
3. and true after Line 16 since $y \in \operatorname{conv} Q$ even after deleting points.

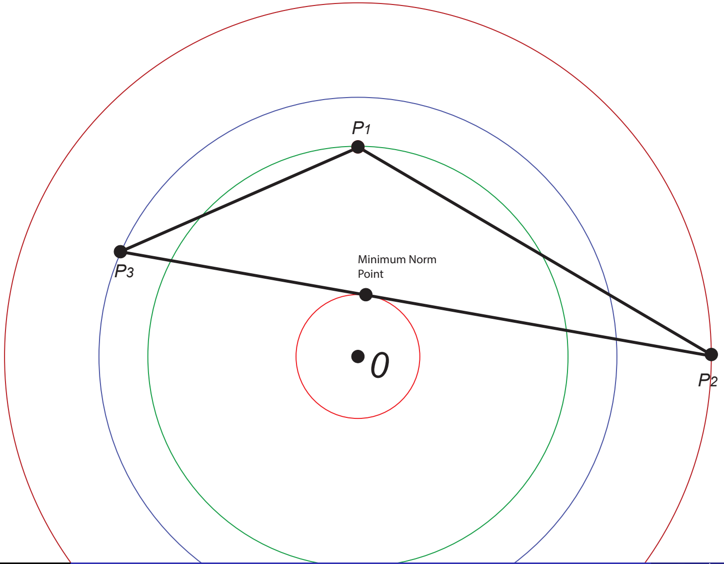
- Note also for any $x^* \in \operatorname{conv} Q \subseteq \operatorname{conv} P$, we have

$$\min_{x \in \operatorname{aff} Q} \|x\|_2 \leq \min_{x \in \operatorname{conv} Q} \|x\|_2 \leq \|x^*\|_2 \tag{18.49}$$

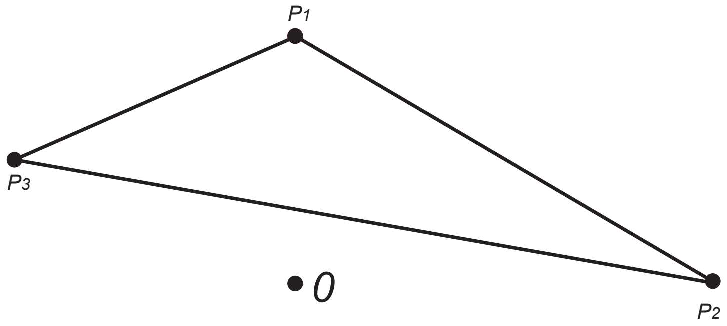
- Note, the input, P, consists of m points. In the case of the base polytope, $P = B_f$ could be exponential in $n = |V|$.
- There are six places that might be seemingly tricky or expensive: Line 4, Line 6, Line 9, Line 10, Line 14, and Line 15.
- We will consider each in turn, but first we do a geometric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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example

Polytope, and circles concentric at 0.

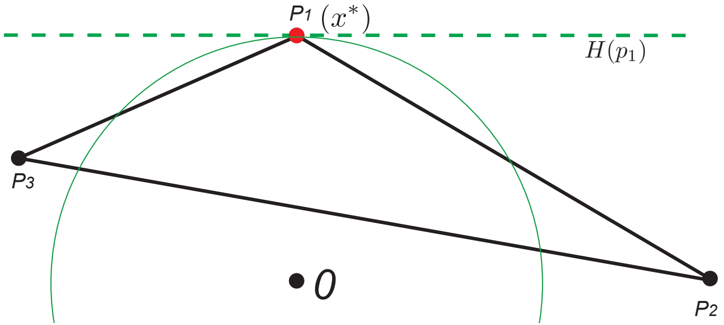


Fujishige-Wolfe Min-Norm algorithm: Geometric Example



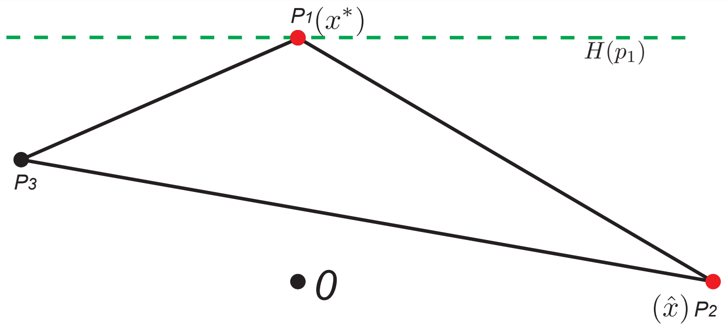
The initial polytope consisting of the convex hull of three points p_1, p_2, p_3, and the origin 0.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



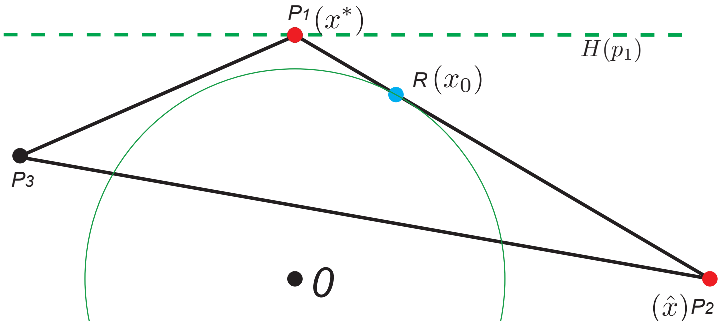
p_1 is the extreme point closest to 0 and so we choose it first, although we can choose any arbitrary extreme point as the initial point. We set $x^* \leftarrow p_1$ in Line 1, and $Q \leftarrow \{p_1\}$ in Line 2. $H(x^*) = H(p_1)$ (green dashed line) is not a supporting hyperplane of $\text{conv}(P)$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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



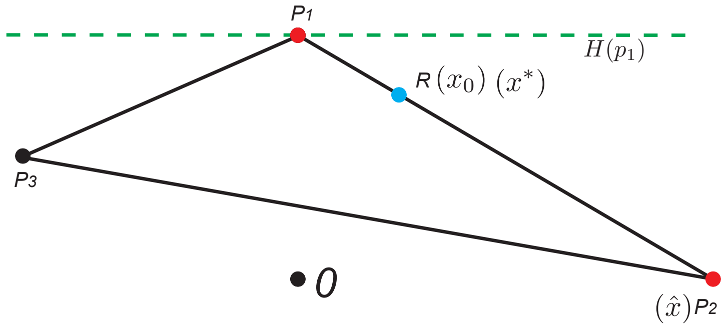
We need to add some extreme point $\hat{x}$ on the "near" side of $H(p_1)$ in Line 6, we choose $\hat{x} = p_2$. In Line 7, we set $Q \leftarrow Q \cup \{p_2\}$, so $Q = \{p_1, p_2\}$.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



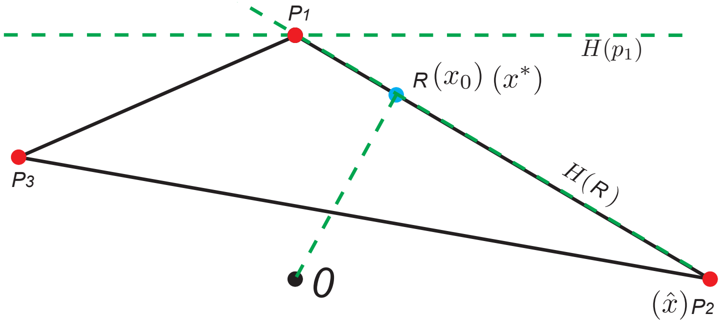
$x_0 = R$ is the min-norm point in $\text{aff}\,\{p_1, p_2\}$ computed in Line 9.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



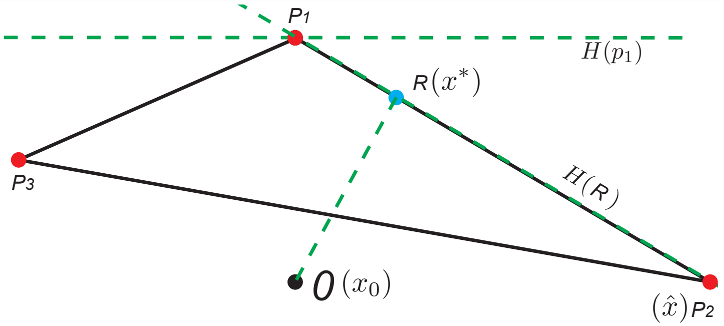
$x_0 = R$ is the min-norm point in $\text{aff}\,\{p_1, p_2\}$ computed in Line 9. Also, with $Q = \{p_1, p_2\}$, since $R \in \text{conv}\, Q$, we set $x^* \leftarrow x_0 = R$ in Line 11. Note, after Line 11, we still have $x^* \in P$ and $\|x^*\|_2 = \|x^*_{\text{new}}\|_2 < \|x^*_{\text{old}}\|_2$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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



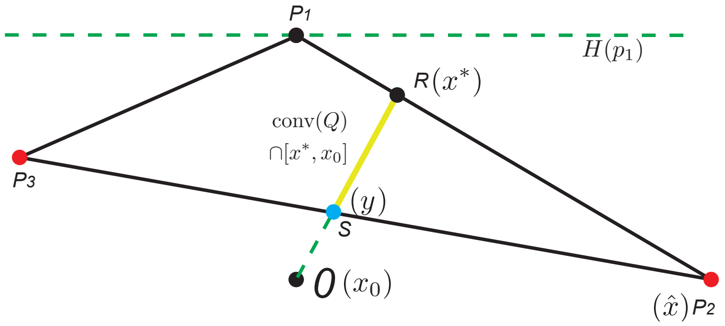
$R = x_0 = x^*$. We consider next $H(R) = H(x^*)$ in Line 4. $H(x^*)$ is not a supporting hyperplane of $\operatorname{conv} P$. So we choose p_3 on the "near" side of $H(x^*)$ in Line 6. Add $Q \leftarrow Q \cup \{p_3\}$ in Line 7. Now $Q = P = \{p_1, p_2, p_3\}$.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



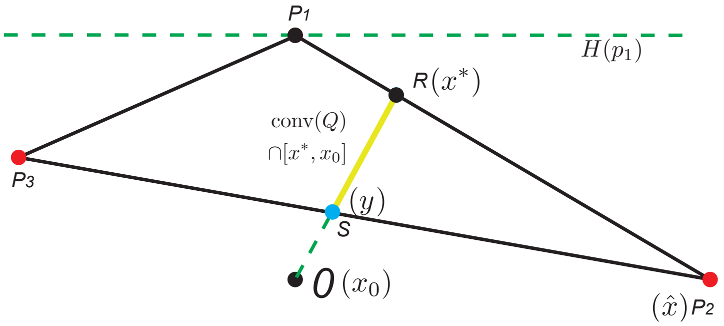
$R = x_0 = x^*$. We consider next $H(R) = H(x^*)$ in Line 4. $H(x^*)$ is not a supporting hyperplane of $\operatorname{conv} P$. So we choose p_3 on the "near" side of $H(x^*)$ in Line 6. Add $Q \leftarrow Q \cup \{p_3\}$ in Line 7. Now $Q = P = \{p_1, p_2, p_3\}$. The origin $x_0 = 0$ is the min-norm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



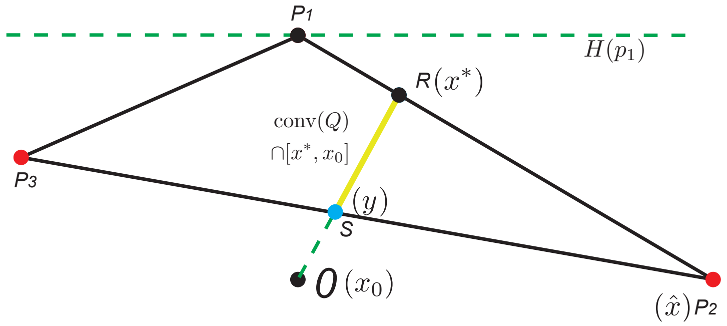
$Q = P = \{p_1, p_2, p_3\}$. Line 14: $S = y = \min_{x \in \text{conv}\, Q \cap [x^*, x_0]} \|x - x_0\|_2$ where x_0 is 0 and x^* is R here. Thus, y lies on the boundary of $\text{conv}\, Q$. Note, $\|y\|_2 < \|x^*\|_2$ since $x^* \in \text{conv}\, Q$, $\|x_0\|_2 < \|x^*\|_2$.

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



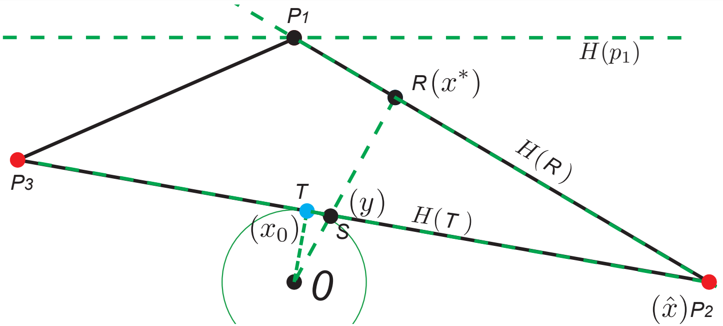
$Q = P = \{p_1, p_2, p_3\}$. Line 14: $S = y = \min_{x \in \mathrm{conv}\, Q \cap [x^*, x_0]} \|x - x_0\|_2$ where x_0 is 0 and x^* is R here. Thus, y lies on the boundary of $\mathrm{conv}\, Q$. Note, $\|y\|_2 < \|x^*\|_2$ since $x^* \in \mathrm{conv}\, Q$, $\|x_0\|_2 < \|x^*\|_2$. Line 15: Delete p_1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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



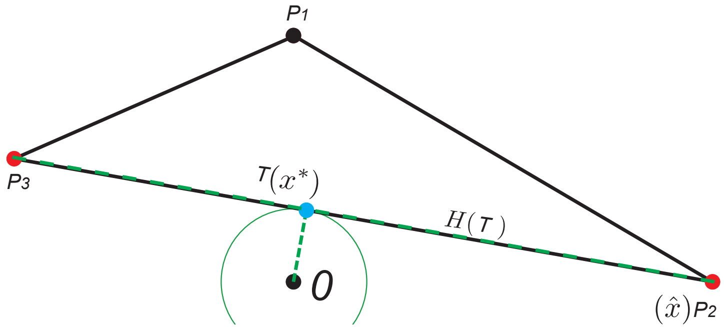
$Q = P = \{p_1, p_2, p_3\}$. Line 14: $S = y = \min_{x \in \mathrm{conv}\, Q \cap [x^*, x_0]} \|x - x_0\|_2$ where x_0 is 0 and x^* is R here. Thus, y lies on the boundary of $\mathrm{conv}\, Q$. Note, $\|y\|_2 < \|x^*\|_2$ since $x^* \in \mathrm{conv}\, Q$, $\|x_0\|_2 < \|x^*\|_2$. Line 15: Delete p_1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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



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

Fujishige-Wolfe Min-Norm algorithm: Geometric Example



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

Condition for Min-Norm Point

Theorem 18.4.2

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

$$p_i^\intercal x^* \geq \|x^*\|_2^2 \quad \forall i = 1, \cdots, m. \tag{18.50}$$

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 18.4.2

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

$$p_i^\intercal x^* \geq \|x^*\|_2^2 \quad \forall i = 1, \cdots, m. \tag{18.50}$$

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

$$\|z\|_2^2 = \|x^* + \theta(y - x^*)\|_2^2 \tag{18.51}$$

$$= \|x^*\|_2^2 + 2\theta(x^{*\intercal} y - x^{*\intercal} x^*) + \theta^2 \|y - x^*\|_2^2 \tag{18.52}$$

Condition for Min-Norm Point

Theorem 18.4.2

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

$$p_i^\intercal x^* \geq \|x^*\|_2^2 \quad \forall i = 1, \cdots, m. \tag{18.50}$$

Proof.

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

$$\|z\|_2^2 = \|x^* + \theta(y - x^*)\|_2^2 \tag{18.51}$$

$$= \|x^*\|_2^2 + 2\theta({x^*}^\intercal y - {x^*}^\intercal x^*) + \theta^2 \|y - x^*\|_2^2 \tag{18.52}$$

- It is possible for $\|z\|_2^2 < \|x^*\|_2^2$ for small θ, unless ${x^*}^\intercal y \geq {x^*}^\intercal x^*$ for all $y \in \operatorname{conv} P \Rightarrow$ Equation (18.50).

Condition for Min-Norm Point

Theorem 18.4.2

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

$$p_i^\intercal x^* \geq \|x^*\|_2^2 \quad \forall i = 1, \cdots, m. \tag{18.50}$$

Proof.

- Assume x^* is the min-norm point, let $y \in \operatorname{conv} P$, and $0 \leq \theta \leq 1$.
- Then $z \triangleq x^* + \theta(y - x^*) = (1-\theta)x^* + \theta y \in \operatorname{conv} P$, and
$$\|z\|_2^2 = \|x^* + \theta(y - x^*)\|_2^2 \tag{18.51}$$
$$= \|x^*\|_2^2 + 2\theta(x^{*\intercal} y - x^{*\intercal} x^*) + \theta^2 \|y - x^*\|_2^2 \tag{18.52}$$
- It is possible for $\|z\|_2^2 < \|x^*\|_2^2$ for small θ, unless $x^{*\intercal} y \geq x^{*\intercal} x^*$ for all $y \in \operatorname{conv} P \Rightarrow$ Equation (18.50).
- Conversely, given Eq (18.50), and given that $y = \sum_i \lambda_i p_i \in \operatorname{conv} P$,
$$y^\intercal x^* = \sum_i \lambda_i p_i^\intercal x^* \geq \sum_i \lambda_i x^{*\intercal} x^* = x^{*\intercal} x^* \tag{18.53}$$
implying that $\|z\|_2^2 > \|x^*\|_2^2$ in Equation 18.52 for arbitrary $z \in \operatorname{conv} P$. □

The set Q is always affinely independent

Lemma 18.4.3

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

The set Q is always affinely independent

Lemma 18.4.3

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

Proof.

$\square$

The set Q is always affinely independent

Lemma 18.4.3

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

Proof.

- Q is of course affinely independent when there is at most one point in it (e.g., after Line 2).

$\square$

The set Q is always affinely independent

Lemma 18.4.3

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

Proof.

- Q is of course affinely independent when there is at most one point in it (e.g., after Line 2).
- After the initialization, it changes only by deletion of points, or adding a single point. Deletion does not change the independence.

$\square$

The set Q is always affinely independent

Lemma 18.4.3

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

Proof.

- Q is of course affinely independent when there is at most one point in it (e.g., after Line 2).
- After the initialization, it changes only by deletion of points, or adding a single point. Deletion does not change the independence.
- Before adding $\hat{x}$ at Line 7, we know x^* is the minimum norm point in $\operatorname{aff} Q$ (since we break only at Line 11).

$\square$

The set Q is always affinely independent

Lemma 18.4.3

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

Proof.

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

$\square$

The set Q is always affinely independent

Lemma 18.4.3

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

Proof.

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

□

The set Q is always affinely independent

Lemma 18.4.3

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

Proof.

- Q is of course affinely independent when there is at most one point in it (e.g., after Line 2).
- After the initialization, it changes only by deletion of points, or adding a single point. Deletion does not change the independence.
- Before adding $\hat{x}$ at Line 7, we know x^* is the minimum norm point in $\operatorname{aff} Q$ (since we break only at Line 11).
- 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 18.4.3

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

Proof.

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

Thus, by Lemma 18.4.3, we have for any $x \in \operatorname{aff} Q$ such that $x = \sum_i w_i q_i$ with $\sum_i w_i = 1$, the weights w_i are uniquely determined.

Minimum Norm in an affine set

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

Minimum Norm in an affine set

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

$$\begin{aligned} &\text{minimize} && \|x\|_2^2 = w^\intercal Q^\intercal Q w && (18.54) \\ &\text{subject to} && \mathbf{1}^\intercal w = 1 && (18.55) \end{aligned}$$

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:

$$\text{minimize} \qquad \|x\|_2^2 = w^\intercal Q^\intercal Q w \tag{18.54}$$

$$\text{subject to} \qquad \mathbf{1}^\intercal w = 1 \tag{18.55}$$

- 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 $x_0 \leftarrow \min_{x \in \operatorname{aff} Q} \|x\|_2$.
- 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:

$$\text{minimize} \quad \|x\|_2^2 = w^\intercal Q^\intercal Q w \tag{18.54}$$

$$\text{subject to} \quad \mathbf{1}^\intercal w = 1 \tag{18.55}$$

- 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 $x^* = \sum_i \lambda_i p_i$ of x^* and from this vector. We also keep v such that $x_0 = \sum_i v_i q_i$ for points $q_i \in Q$, from Line 9.
  Given w and v, we can also easily solve Lines 14 and 15 (see "Step 3" on page 133 of Wolfe-1976, which also defines numerical tolerances).

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:

$$\text{minimize} \quad \|x\|_2^2 = w^\intercal Q^\intercal Q w \tag{18.54}$$

$$\text{subject to} \quad \mathbf{1}^\intercal w = 1 \tag{18.55}$$

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

MN Algorithm finds the MN point in finite time.

Theorem 18.4.4

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

Proof.

- In minor loop, we always have $x^* \in \operatorname{conv} Q$, since whenever Q is modified, x^* is updated as well (Line 16) such that the updated x^* remains in new $\operatorname{conv} Q$.

. . .

MN Algorithm finds the MN point in finite time.

Theorem 18.4.4

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

Proof.

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

...

MN Algorithm finds the MN point in finite time.

Theorem 18.4.4

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

Proof.

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

...

MN Algorithm finds the MN point in finite time.

. . . proof of Theorem 18.4.4 continued.

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

. . .

MN Algorithm finds the MN point in finite time.

. . . proof of Theorem 18.4.4 continued.

- Moreover, there can be no more iterations within a minor loop than the dimension of $\operatorname{conv} Q$ for the initial Q given to the minor loop initially at Line 8 (dimension of $\operatorname{conv} Q$ is $|Q| - 1$ since Q is affinely independent).
- Each iteration of the minor loop removes at least one point from Q in Line 15.

. . .

MN Algorithm finds the MN point in finite time.

. . . proof of Theorem 18.4.4 continued.

- Moreover, there can be no more iterations within a minor loop than the dimension of $\operatorname{conv} Q$ for the initial Q given to the minor loop initially at Line 8 (dimension of $\operatorname{conv} Q$ is $|Q| - 1$ since Q is affinely independent).
- Each iteration of the minor loop removes at least one point from Q in Line 15.
- When Q reduces to a singleton, the minor loop always terminates.

. . .

MN Algorithm finds the MN point in finite time.

. . . proof of Theorem 18.4.4 continued.

- Moreover, there can be no more iterations within a minor loop than the dimension of $\operatorname{conv} Q$ for the initial Q given to the minor loop initially at Line 8 (dimension of $\operatorname{conv} Q$ is $|Q| - 1$ since Q is affinely independent).
- Each iteration of the minor loop removes at least one point from Q in Line 15.
- When Q reduces to a singleton, the minor loop always terminates.
- Thus, the minor loop terminates in finite number of iterations, at most dimension of Q.

. . .

MN Algorithm finds the MN point in finite time.

. . . proof of Theorem 18.4.4 continued.

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

. . .

MN Algorithm finds the MN point in finite time.

. . . proof of Theorem 18.4.4 continued.

- Each time Q is augmented with $\hat{x}$ at Line 7, followed by updating x^* with x_0 at Line 11, (i.e., when the minor loop returns with only one iteration), $\|x^*\|_2$ strictly decreases from what it was before.

. . .

MN Algorithm finds the MN point in finite time.

. . . proof of Theorem 18.4.4 continued.

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

. . .

MN Algorithm finds the MN point in finite time.

…proof of Theorem 18.4.4 continued.

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

$$\begin{aligned}\|x^* + \theta(\hat{x} - x^*)\|_2^2 &= \|x^*\|_2^2 + 2\theta\left((x^*)^\top \hat{x} - \|x^*\|_2^2\right) + \theta^2 \|\hat{x} - x^*\|_2^2 \\ &\geq \|x_0\|_2^2 \end{aligned} \tag{18.56}$$

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

…

MN Algorithm finds the MN point in finite time.

... proof of Theorem 18.4.4 continued.

- Therefore, for sufficiently small θ, specifically for

$$\theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2} \tag{18.57}$$

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

$\square$

MN Algorithm finds the MN point in finite time.

... proof of Theorem 18.4.4 continued.

- Therefore, for sufficiently small θ, specifically for

$$\theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2} \tag{18.57}$$

  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.

$\square$

MN Algorithm finds the MN point in finite time.

. . . proof of Theorem 18.4.4 continued.

- Therefore, for sufficiently small θ, specifically for

$$\theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2} \tag{18.57}$$

  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.

$\square$

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

- The “near” side means the side that contains the origin.

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

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

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

- The "near" side means the side that contains the origin.
- Ideally, find $\hat{x}$ such that the reduction of $\|x^*\|_2$ is maximized to reduce number of major iterations.
- From Eqn. 18.56, reduction on norm is lower-bounded:

$$\Delta = \|x^*\|_2^2 - \|x_0\|_2^2 \geq 2\theta \left(\|x^*\|_2^2 - (x^*)^\top \hat{x} \right) - \theta^2 \|\hat{x} - x^*\|_2^2 \triangleq \underline{\Delta} \tag{18.58}$$

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

- The "near" side means the side that contains the origin.
- Ideally, find $\hat{x}$ such that the reduction of $\|x^*\|_2$ is maximized to reduce number of major iterations.
- From Eqn. 18.56, reduction on norm is lower-bounded:

$$\Delta = \|x^*\|_2^2 - \|x_0\|_2^2 \geq 2\theta \left(\|x^*\|_2^2 - (x^*)^\top \hat{x} \right) - \theta^2 \|\hat{x} - x^*\|_2^2 \triangleq \underline{\Delta} \tag{18.58}$$

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

$$\max_{0 \leq \theta < \frac{2\left(\|x^*\|_2^2 - (x^*)^\top \hat{x}\right)}{\|\hat{x} - x^*\|_2^2}} \underline{\Delta} = \left(\frac{\|x^*\|_2^2 - (x^*)^\top \hat{x}}{\|\hat{x} - x^*\|_2} \right)^2 \tag{18.59}$$

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

- To maximize lower bound of norm reduction at each major iteration, want to find an $\hat{x}$ such that the above lower bound (Equation 18.59) is maximized.

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

- To maximize lower bound of norm reduction at each major iteration, want to find an $\hat{x}$ such that the above lower bound (Equation 18.59) is maximized.
- That is, we want to find

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \left(\frac{\|x^*\|_2^2 - (x^*)^\top x}{\|x - x^*\|_2} \right)^2 \tag{18.60}$$

to ensure that a large norm reduction is assured.

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

- To maximize lower bound of norm reduction at each major iteration, want to find an $\hat{x}$ such that the above lower bound (Equation 18.59) is maximized.
- That is, we want to find

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \left(\frac{\|x^*\|_2^2 - (x^*)^\top x}{\|x - x^*\|_2} \right)^2 \tag{18.60}$$

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

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

- As a surrogate, we maximize numerator in Eqn. 18.60, i.e., find

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \|x^*\|_2^2 - (x^*)^\top x = \operatorname*{argmin}_{x \in P} (x^*)^\top x, \tag{18.61}$$

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

- As a surrogate, we maximize numerator in Eqn. 18.60, i.e., find

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \|x^*\|_2^2 - (x^*)^\top x = \operatorname*{argmin}_{x \in P} (x^*)^\top x, \tag{18.61}$$

- Intuitively, by solving the above, we find $\hat{x}$ such that it has the largest distance to the hyperplane $H(x^*)$, and this is exactly the strategy used in the Wolfe-1976 algorithm.

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

- As a surrogate, we maximize numerator in Eqn. 18.60, i.e., find

$$\hat{x} \in \underset{x \in P}{\operatorname{argmax}} \|x^*\|_2^2 - (x^*)^\top x = \underset{x \in P}{\operatorname{argmin}} (x^*)^\top x, \qquad (18.61)$$

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

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

- As a surrogate, we maximize numerator in Eqn. 18.60, i.e., find

$$\hat{x} \in \operatorname*{argmax}_{x \in P} \|x^*\|_2^2 - (x^*)^\top x = \operatorname*{argmin}_{x \in P}(x^*)^\top x, \tag{18.61}$$

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

$$(x^*)^\top \hat{x} \geq \|x^*\|_2^2\,, \tag{18.62}$$

where $\hat{x}$ is the solution of Eq. 18.61.

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

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

$$(x^*)^\top \hat{x} > \|x^*\|_2^2 - \epsilon \max_{x \in Q} \|x\|_2^2 \tag{18.63}$$

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

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

$$(x^*)^\top \hat{x} > \|x^*\|_2^2 - \epsilon \max_{x \in Q} \|x\|_2^2 \tag{18.63}$$

- When $\operatorname{conv} P$ is a submodular base polytope (i.e., $\operatorname{conv} P = B_f$ for a submodular function f), then the problem in Eqn 18.61 can be solved efficiently by Edmonds's greedy algorithm (even though there may be an exponential number of extreme points).

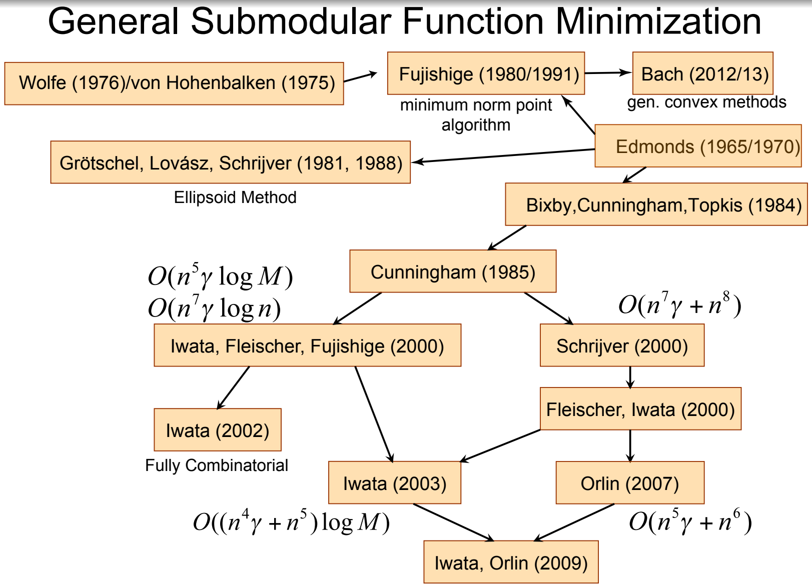
Line: 6: Finding $\hat{x} \in P$ on the near side of $H(x^*)$

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

$$(x^*)^\top \hat{x} > \|x^*\|_2^2 - \epsilon \max_{x \in Q} \|x\|_2^2 \tag{18.63}$$

- When $\operatorname{conv} P$ is a submodular base polytope (i.e., $\operatorname{conv} P = B_f$ for a submodular function f), then the problem in Eqn 18.61 can be solved efficiently by Edmonds's greedy algorithm (even though there may be an exponential number of extreme points).
- Hence, Edmonds's discovery is one of the main reasons that the MN algorithm is applicable to submodular function minimization.

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



MN Algorithm Complexity

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

MN Algorithm Complexity

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

MN Algorithm Complexity

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

MN Algorithm Complexity

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

MN Algorithm Complexity

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

MN Algorithm Complexity

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

MN Algorithm Complexity

- The currently fastest strongly polynomial combinatorial algorithm for SFM achieves a running time of $O(n^5 T + n^6)$ (Orlin'09) where T is the time for function evaluation, far from practical for large problem instances.
- Fujishige & Isotani report that MN algorithm is fast in practice, but they use only a limited set of submodular functions.
- Complexity of MN Algorithm is still an unsolved problem.
- Obvious facts:
  - each major iteration requires $O(n)$ function oracle calls
  - complexity of each major iteration could be at least $O(n^3)$ due to the affine projection step (solving a linear system).
  - Therefore, the complexity of each major iteration is
    $$O(n^3 + n^{1+p})$$
    where each function oracle call requires $O(n^p)$ time.

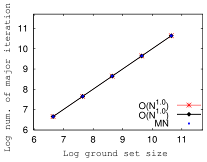
MN Algorithm Complexity

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

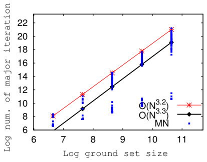
    $$O(n^3 + n^{1+p})$$

    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.

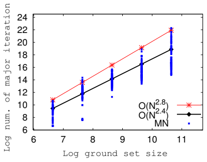
MN Algorithm Empirical Complexity



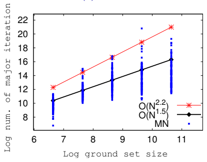
(a) $\alpha = 0.1$



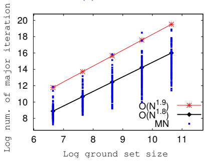
(b) $\alpha = 0.2$



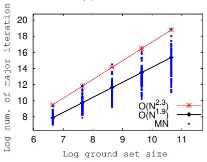
(c) $\alpha = 0.3$



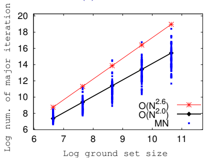
(d) $\alpha = 0.4$



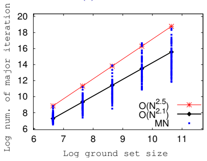
(e) $\alpha = 0.5$



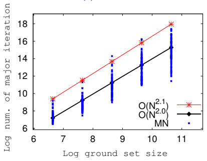
(f) $\alpha = 0.6$



(g) $\alpha = 0.7$



(h) $\alpha = 0.8$



(i) $\alpha = 0.9$

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