A Carathéodory result for combinatorial hull?

S. Thomas McCormick

Sauder School of Business, UBC
FND rump session, July 2013

Submodularity definitions

- If f is a set function on E, we say that f is submodular if

$$\forall S \subset T \subset T + e, \; f(T + e) - f(T) \leq f(S + e) - f(S). \quad (1)$$

Submodularity definitions

- If f is a set function on E, we say that f is submodular if

$$\forall S \subset T \subset T + e, \ f(T + e) - f(T) \leq f(S + e) - f(S). \quad (1)$$

- The classic definition of submodularity is that set function f is submodular if

$$\text{for all } S, T \subseteq E, \ f(S) + f(T) \geq f(S \cup T) + f(S \cap T). \quad (2)$$

Submodularity definitions

- If f is a set function on E, we say that f is submodular if

$$\forall S \subset T \subset T + e, \ f(T+e) - f(T) \leq f(S+e) - f(S). \quad (1)$$

- The classic definition of submodularity is that set function f is submodular if

$$\text{for all } S, T \subseteq E, \ f(S) + f(T) \geq f(S \cup T) + f(S \cap T). \quad (2)$$

Lemma

Definitions (1) and (2) are equivalent.

Submodular polyhedra

- Let's associate submodular functions with polyhedra.

Submodular polyhedra

- Let's associate submodular functions with polyhedra.
- It turns out that the right thing to do is to think about vectors $x \in \mathbb{R}^E$, and so polyhedra in $\mathbb{R}^E$.

Submodular polyhedra

- Let's associate submodular functions with polyhedra.
- It turns out that the right thing to do is to think about vectors $x \in \mathbb{R}^E$, and so polyhedra in $\mathbb{R}^E$.
- The key constraint for us is for some subset $S \subseteq E$

$$x(S) \leq f(S).$$

Submodular polyhedra

- Let's associate submodular functions with polyhedra.
- It turns out that the right thing to do is to think about vectors $x \in \mathbb{R}^E$, and so polyhedra in $\mathbb{R}^E$.
- The key constraint for us is for some subset $S \subseteq E$

$$x(S) \leq f(S).$$

- What about when $S = \emptyset$? To get this to make sense we will *normalize* all our submodular functions by assuming that $f(\emptyset) = 0$.

Submodular polyhedra

- Let's associate submodular functions with polyhedra.
- It turns out that the right thing to do is to think about vectors $x \in \mathbb{R}^E$, and so polyhedra in $\mathbb{R}^E$.
- The key constraint for us is for some subset $S \subseteq E$

$$x(S) \le f(S).$$

- What about when $S = \emptyset$? To get this to make sense we will *normalize* all our submodular functions by assuming that $f(\emptyset) = 0$.
- Now that we've normalized s.t. $f(\emptyset) = 0$, define the submodular polyhedron associated with set function f by

$$P(f) \equiv \{x \in \mathbb{R}^E \mid x(S) \le f(S)\ \forall S \subseteq E\}.$$

Submodular polyhedra

- Let's associate submodular functions with polyhedra.
- It turns out that the right thing to do is to think about vectors $x \in \mathbb{R}^E$, and so polyhedra in $\mathbb{R}^E$.
- The key constraint for us is for some subset $S \subseteq E$
$$x(S) \leq f(S).$$
- What about when $S = \emptyset$? To get this to make sense we will *normalize* all our submodular functions by assuming that $f(\emptyset) = 0$.
- Now that we've normalized s.t. $f(\emptyset) = 0$, define the submodular polyhedron associated with set function f by
$$P(f) \equiv \{x \in \mathbb{R}^E \mid x(S) \leq f(S)\ \forall S \subseteq E\}.$$
- It turns out to be convenient to also consider the face of $P(f)$ induced by the constraint $x(E) \leq f(E)$, called the base polyhedron of f:
$$B(f) \equiv \{x \in \mathbb{R}^E \mid x(S) \leq f(S) \forall S \subset E,\ x(E) = f(E)\}.$$

How do we know that $y \in B(f)$?

- Algorithms need to verify that $y \in B(f)$, but there are 2^n inequalities to check.

How do we know that $y \in B(f)$?

- Algorithms need to verify that $y \in B(f)$, but there are 2^n inequalities to check.
- Here is a clever way to do it (Cunningham):

How do we know that $y \in B(f)$?

- Algorithms need to verify that $y \in B(f)$, but there are 2^n inequalities to check.
- Here is a clever way to do it (Cunningham):
    1. $B(f)$ is bounded, and so it is the convex hull of its vertices, i.e., $y \in B(f)$ iff y is a convex combination of vertices of $B(f)$.

How do we know that $y \in B(f)$?

- Algorithms need to verify that $y \in B(f)$, but there are 2^n inequalities to check.
- Here is a clever way to do it (Cunningham):
  1. $B(f)$ is bounded, and so it is the convex hull of its vertices, i.e., $y \in B(f)$ iff y is a convex combination of vertices of $B(f)$.
  2. We know that all vertices of $B(f)$ come from Greedy applied to linear orders, which have succinct certificates.

How do we know that $y \in B(f)$?

- Algorithms need to verify that $y \in B(f)$, but there are 2^n inequalities to check.
- Here is a clever way to do it (Cunningham):
  1. $B(f)$ is bounded, and so it is the convex hull of its vertices, i.e., $y \in B(f)$ iff y is a convex combination of vertices of $B(f)$.
  2. We know that all vertices of $B(f)$ come from Greedy applied to linear orders, which have succinct certificates.
  3. Carathéodory's Theorem says that in fact there is always a convex hull representation of y using at most n vertices.

How do we know that $y \in B(f)$?

- Algorithms need to verify that $y \in B(f)$, but there are 2^n inequalities to check.
- Here is a clever way to do it (Cunningham):
  1. $B(f)$ is bounded, and so it is the convex hull of its vertices, i.e., $y \in B(f)$ iff y is a convex combination of vertices of $B(f)$.
  2. We know that all vertices of $B(f)$ come from Greedy applied to linear orders, which have succinct certificates.
  3. Carathéodory's Theorem says that in fact there is always a convex hull representation of y using at most n vertices.
- Therefore the algorithms will keep a convex hull representation of y like this:

How do we know that $y \in B(f)$?

- Algorithms need to verify that $y \in B(f)$, but there are 2^n inequalities to check.
- Here is a clever way to do it (Cunningham):
  1. $B(f)$ is bounded, and so it is the convex hull of its vertices, i.e., $y \in B(f)$ iff y is a convex combination of vertices of $B(f)$.
  2. We know that all vertices of $B(f)$ come from Greedy applied to linear orders, which have succinct certificates.
  3. Carathéodory's Theorem says that in fact there is always a convex hull representation of y using at most n vertices.
- Therefore the algorithms will keep a convex hull representation of y like this:
  - We have an index set $\mathcal{I}$ of size $O(n)$.

How do we know that $y \in B(f)$?

- Algorithms need to verify that $y \in B(f)$, but there are 2^n inequalities to check.
- Here is a clever way to do it (Cunningham):
  1. $B(f)$ is bounded, and so it is the convex hull of its vertices, i.e., $y \in B(f)$ iff y is a convex combination of vertices of $B(f)$.
  2. We know that all vertices of $B(f)$ come from Greedy applied to linear orders, which have succinct certificates.
  3. Carathéodory's Theorem says that in fact there is always a convex hull representation of y using at most n vertices.
- Therefore the algorithms will keep a convex hull representation of y like this:
  - We have an index set $\mathcal{I}$ of size $O(n)$.
  - For each $i \in \mathcal{I}$ we have a linear order $\prec_i$ with associated Greedy vertex v^i.

How do we know that $y \in B(f)$?

- Algorithms need to verify that $y \in B(f)$, but there are 2^n inequalities to check.
- Here is a clever way to do it (Cunningham):
  1. $B(f)$ is bounded, and so it is the convex hull of its vertices, i.e., $y \in B(f)$ iff y is a convex combination of vertices of $B(f)$.
  2. We know that all vertices of $B(f)$ come from Greedy applied to linear orders, which have succinct certificates.
  3. Carathéodory's Theorem says that in fact there is always a convex hull representation of y using at most n vertices.
- Therefore the algorithms will keep a convex hull representation of y like this:
  - We have an index set $\mathcal{I}$ of size $O(n)$.
  - For each $i \in \mathcal{I}$ we have a linear order $\prec_i$ with associated Greedy vertex v^i.
  - We keep multipliers $\lambda_i \geq 0$ for $i \in \mathcal{I}$ satisfying $\sum_{i \in \mathcal{I}} \lambda_i = 1$.

How do we know that $y \in B(f)$?

- Algorithms need to verify that $y \in B(f)$, but there are 2^n inequalities to check.
- Here is a clever way to do it (Cunningham):
  1. $B(f)$ is bounded, and so it is the convex hull of its vertices, i.e., $y \in B(f)$ iff y is a convex combination of vertices of $B(f)$.
  2. We know that all vertices of $B(f)$ come from Greedy applied to linear orders, which have succinct certificates.
  3. Carathéodory's Theorem says that in fact there is always a convex hull representation of y using at most n vertices.
- Therefore the algorithms will keep a convex hull representation of y like this:
  - We have an index set $\mathcal{I}$ of size $O(n)$.
  - For each $i \in \mathcal{I}$ we have a linear order $\prec_i$ with associated Greedy vertex v^i.
  - We keep multipliers $\lambda_i \geq 0$ for $i \in \mathcal{I}$ satisfying $\sum_{i \in \mathcal{I}} \lambda_i = 1$.
  - Then $y = \sum_{i \in \mathcal{I}} \lambda_i v^i$ is a succinct certificate proving that $y \in B(f)$.

Why is this convex hull representation bad?

- There is no reason why we need multiplication and division in dealing with submodular functions — after all, Greedy operates with only addition, subtraction, and comparison.

Why is this convex hull representation bad?

- There is no reason why we need multiplication and division in dealing with submodular functions — after all, Greedy operates with only addition, subtraction, and comparison.
- As algorithms proceed, new vertices get added to $\mathcal{I}$, and then we need to do some linear algebra to reduce $|\mathcal{I}|$ to $O(n)$. This linear algebra associated with maintaining the convex hull representation is a computational bottleneck in empirical testing.

Why is this convex hull representation bad?

- There is no reason why we need multiplication and division in dealing with submodular functions — after all, Greedy operates with only addition, subtraction, and comparison.
- As algorithms proceed, new vertices get added to $\mathcal{I}$, and then we need to do some linear algebra to reduce $|\mathcal{I}|$ to $O(n)$. This linear algebra associated with maintaining the convex hull representation is a computational bottleneck in empirical testing.
- So let's look for a better method of proving that $y \in B(f)$ that involves only addition, subtraction, and comparison.

Combinatorial hull

- Suppose that we know (somehow) that $x, z \in B(f)$ and we want to prove that $y \in B(f)$.

Combinatorial hull

- Suppose that we know (somehow) that $x, z \in B(f)$ and we want to prove that $y \in B(f)$.
- Define $\tilde{x}$, $\tilde{y}$, and $\tilde{z}$ to be x, y, z with one fixed coordinate projected out.

Combinatorial hull

- Suppose that we know (somehow) that $x, z \in B(f)$ and we want to prove that $y \in B(f)$.
- Define $\tilde{x}$, $\tilde{y}$, and $\tilde{z}$ to be x, y, z with one fixed coordinate projected out.

Lemma

(Fujishige): If $\tilde{x} \leq \tilde{y} \leq \tilde{z}$ *then* $y \in B(f)$.

Combinatorial hull

- Suppose that we know (somehow) that $x, z \in B(f)$ and we want to prove that $y \in B(f)$.
- Define $\tilde{x}$, $\tilde{y}$, and $\tilde{z}$ to be x, y, z with one fixed coordinate projected out.

Lemma

(Fujishige): If $\tilde{x} \leq \tilde{y} \leq \tilde{z}$ then $y \in B(f)$.

- For example, if x, z are Greedy vertices, then the *level-0* "projected box" between them is contained in $B(f)$.

Combinatorial hull

- Suppose that we know (somehow) that $x, z \in B(f)$ and we want to prove that $y \in B(f)$.
- Define $\tilde{x}$, $\tilde{y}$, and $\tilde{z}$ to be x, y, z with one fixed coordinate projected out.

Lemma

(Fujishige): If $\tilde{x} \leq \tilde{y} \leq \tilde{z}$ then $y \in B(f)$.

- For example, if x, z are Greedy vertices, then the *level-0* "projected box" between them is contained in $B(f)$.
- We can iterate this operation: We could again take a projected box between two level-0 points (maybe coming from projecting out different coordinates) to get new points.

Combinatorial hull

- Suppose that we know (somehow) that $x, z \in B(f)$ and we want to prove that $y \in B(f)$.
- Define $\tilde{x}$, $\tilde{y}$, and $\tilde{z}$ to be x, y, z with one fixed coordinate projected out.

Lemma

(Fujishige): If $\tilde{x} \le \tilde{y} \le \tilde{z}$ then $y \in B(f)$.

- For example, if x, z are Greedy vertices, then the *level-0* "projected box" between them is contained in $B(f)$.
- We can iterate this operation: We could again take a projected box between two level-0 points (maybe coming from projecting out different coordinates) to get new points.
- It's easy to prove that 2^{n-1} such combinatorial hull operations suffice to cover all of $B(f)$.

Combinatorial hull

- Suppose that we know (somehow) that $x, z \in B(f)$ and we want to prove that $y \in B(f)$.
- Define $\tilde{x}$, $\tilde{y}$, and $\tilde{z}$ to be x, y, z with one fixed coordinate projected out.

Lemma

(Fujishige): If $\tilde{x} \leq \tilde{y} \leq \tilde{z}$ then $y \in B(f)$.

- For example, if x, z are Greedy vertices, then the *level-0* "projected box" between them is contained in $B(f)$.
- We can iterate this operation: We could again take a projected box between two level-0 points (maybe coming from projecting out different coordinates) to get new points.
- It's easy to prove that 2^{n-1} such combinatorial hull operations suffice to cover all of $B(f)$.
- **Open Question:** Can we algorithmically get a polynomial ("Carathéodory-like") bound on the size of such a combinatorial hull representation?