CS599: Convex and Combinatorial Optimization
Fall 2013
Lecture 26: Maximizing Monotone Submodular Functions

Instructor: Shaddin Dughmi

Outline

1. Introduction
2. Cardinality Constraint
3. Matroid Constraint

Recall: Optimizing Submodular Functions

- As our examples suggest, optimization problems involving submodular functions are very common
- These can be classified on two axes: constrained/unconstrained and maximization/minimization

	Maximization	Minimization
Unconstrained	NP-hard $\frac{1}{2}$ approximation	Polynomial time via convex opt
Constrained	Usually NP-hard $1 - 1/e$ (mono, matroid) $O(1)$ ("nice" constriants)	Usually NP-hard to apx. Few easy special cases

Recall: Optimizing Submodular Functions

- As our examples suggest, optimization problems involving submodular functions are very common
- These can be classified on two axes: constrained/unconstrained and maximization/minimization

	Maximization	Minimization
Unconstrained	NP-hard $\frac{1}{2}$ approximation	Polynomial time via convex opt
Constrained	Usually NP-hard $1 - 1/e$ (mono, matroid) $O(1)$ ("nice" constraints)	Usually NP-hard to apx. Few easy special cases

Problem Definition

Given a non-decreasing and normalized submodular function $f : 2^X \to \mathbb{R}_+$ on a finite ground set X, and a matroid $M = (X, \mathcal{I})$

$$\begin{array}{ll} \text{maximize} & f(S) \\ \text{subject to} & S \in \mathcal{I} \end{array}$$

- Non-decreasing: $f(S) \leq f(T)$ for $S \subseteq T$
- Normalized: $f(\emptyset) = 0$.

Problem Definition

Given a non-decreasing and normalized submodular function $f : 2^X \to \mathbb{R}_+$ on a finite ground set X, and a matroid $M = (X, \mathcal{I})$

$$\begin{array}{ll} \text{maximize} & f(S) \\ \text{subject to} & S \in \mathcal{I} \end{array}$$

- Non-decreasing: $f(S) \leq f(T)$ for $S \subseteq T$
- Normalized: $f(\emptyset) = 0$.
- We denote $n = |X|$

Problem Definition

Given a non-decreasing and normalized submodular function $f : 2^X \to \mathbb{R}_+$ on a finite ground set X, and a matroid $M = (X, \mathcal{I})$

$$\begin{array}{ll} \text{maximize} & f(S) \\ \text{subject to} & S \in \mathcal{I} \end{array}$$

- Non-decreasing: $f(S) \leq f(T)$ for $S \subseteq T$
- Normalized: $f(\emptyset) = 0$.
- We denote $n = |X|$

Representation

As before, we work in the value oracle model. Namely, we only assume we have access to a subroutine evaluating $f(S)$ in constant time.

Examples

Maximum Coverage

X is the left hand side of a graph, and $f(S)$ is the total number of neighbors of S.

- Can think of $i \in X$ as a set, and $f(S)$ as the total “coverage” of S.

Goal is to cover as much of the RHS as possible with k LHS nodes.

Social Influence

- X is the family of nodes in a social network
- A meme, idea, or product is adopted at a set of nodes S
- $f(S)$ is the expected number of nodes in the network which end up adopting the idea.
- Goal is to obtain maximum influence subject to a constraint
  - Cardinality
  - Transversal
  - ...

Combinatorial Allocation

- G is a set of goods
- $f_i(B)$ is submodular utility of agent $i \in N$ for bundle $B \subseteq G$
- Allocation: A partition $(B_1, \ldots, B_n)$ of G among agents.
- Aggregate utility is $\sum_i f_i(B_i)$.

Combinatorial Allocation

- G is a set of goods
- $f_i(B)$ is submodular utility of agent $i \in N$ for bundle $B \subseteq G$
- Allocation: A partition $(B_1, \ldots, B_n)$ of G among agents.
- Aggregate utility is $\sum_i f_i(B_i)$.
- Let $X = G \times N$ be the set of good/agent pairs
- Allocations correspond to subsets S of X in which at most one "copy" of each good is chosen
    - Partition matroid constraint
- $f(S) = \sum_{i \in N} f_i(\{j \in G : (j, i) \in X\})$
    - Submodular

Complexity

Theorem

Maximizing a submodular function subject to a matroid constraint is NP-hard, and NP-hard to approximate to within any better than a factor of $1 - 1/e$.

- Holds even for max coverage

Complexity

Theorem

Maximizing a submodular function subject to a matroid constraint is NP-hard, and NP-hard to approximate to within any better than a factor of $1 - 1/e$.

- Holds even for max coverage

Goal

An algorithm in the value oracle model which

- Runs in time $\mathrm{poly}(n)$
- Returns a feasible set $S^* \in \mathcal{I}$ satisfying $f(S^*) \geq (1 - 1/e) \max_{S \in \mathcal{I}} f(S)$.

Complexity

Theorem

Maximizing a submodular function subject to a matroid constraint is NP-hard, and NP-hard to approximate to within any better than a factor of $1 - 1/e$.

- Holds even for max coverage

Goal

An algorithm in the value oracle model which

- Runs in time $\mathrm{poly}(n)$
- Returns a feasible set $S^* \in \mathcal{I}$ satisfying $f(S^*) \geq (1 - 1/e) \max_{S \in \mathcal{I}} f(S)$.

Holds for arbitrary matroid, but much simpler for uniform matroids.

Outline

1 Introduction

2 Cardinality Constraint

3 Matroid Constraint

Subject to a Cardinality Constraint

Problem Definition

Given a non-decreasing and normalized submodular function $f : 2^X \to \mathbb{R}_+$ on a finite ground set X with $|X| = n$, and an integer $k \le n$

$$\begin{array}{ll} \text{maximize} & f(S) \\ \text{subject to} & |S| \le k \end{array}$$

- k-uniform matroid constraint

The Greedy Algorithm

The following is the straightforward adaptation of the greedy algorithm for maximizing modular functions over a matroid.

The Greedy Algorithm

1. $S \leftarrow \emptyset$
2. While $|S| \leq k$
   - Choose $e \in X$ maximizing $f(S \bigcup \{e\})$
   - $S \leftarrow S \bigcup \{e\}$

The Greedy Algorithm

The following is the straightforward adaptation of the greedy algorithm for maximizing modular functions over a matroid.

The Greedy Algorithm

1. $S \leftarrow \emptyset$
2. While $|S| \leq k$
   - Choose $e \in X$ maximizing $f(S \bigcup \{e\})$
   - $S \leftarrow S \bigcup \{e\}$

Theorem

The greedy algorithm is a $(1 - 1/e)$ approximation algorithm for maximizing a monotone, normalized, and submodular function subject to a cardinality constraint.

Contraction/Conditioning

Let $f : 2^X \to \mathbb{R}$ and $A \subseteq X$. Define $f_A(S) = f(A \bigcup S) - f(A)$.

Lemma

If f is monotone and submodular, then f_A is monotone, submodular, and normalized for any A.

Contraction/Conditioning

Let $f : 2^X \to \mathbb{R}$ and $A \subseteq X$. Define $f_A(S) = f(A \bigcup S) - f(A)$.

Lemma

If f is monotone and submodular, then f_A is monotone, submodular, and normalized for any A.

Proof

- Normalized: trivial

Contraction/Conditioning

Let $f : 2^X \to \mathbb{R}$ and $A \subseteq X$. Define $f_A(S) = f(A \bigcup S) - f(A)$.

Lemma

If f is monotone and submodular, then f_A is monotone, submodular, and normalized for any A.

Proof

- Normalized: trivial
- Monotone:
  - Let $S \subseteq T$
  - $f_A(S) = f(S \cup A) - f(A) \leq f(T \cup A) - f(A) = f_A(T)$.

Contraction/Conditioning

Let $f : 2^X \to \mathbb{R}$ and $A \subseteq X$. Define $f_A(S) = f(A \bigcup S) - f(A)$.

Lemma

If f is monotone and submodular, then f_A is monotone, submodular, and normalized for any A.

Proof

- Normalized: trivial
- Monotone:
  - Let $S \subseteq T$
  - $f_A(S) = f(S \cup A) - f(A) \leq f(T \cup A) - f(A) = f_A(T)$.
- Submodular:

$$\begin{aligned} f_A(S) + f_A(T) &= f(S \cup A) - f(A) + f(T \cup A) - f(A) \\ &\geq f(S \cup T \cup A) - f(A) + f((S \cap T) \cup A) - f(A) \\ &= f_A(S \cup T) - f_A(S \cap T) \end{aligned}$$

Lemma

If f is normalized and submodular, and $A \subseteq X$, then there is $j \in A$ such that $f(\{j\}) \geq \frac{1}{|A|} f(A)$.

Lemma

If f is normalized and submodular, and $A \subseteq X$, then there is $j \in A$ such that $f(\{j\}) \geq \frac{1}{|A|} f(A)$.

Proof

- If A_1, A_2 partition A, then

$$f(A_1) + f(A_2) \geq f(A_1 \cup A_2) + f(A_1 \cap A_2) = f(A)$$

Lemma

If f is normalized and submodular, and $A \subseteq X$, then there is $j \in A$ such that $f(\{j\}) \geq \frac{1}{|A|} f(A)$.

Proof

- If A_1, A_2 partition A, then

$$f(A_1) + f(A_2) \geq f(A_1 \cup A_2) + f(A_1 \cap A_2) = f(A)$$

- Applying recursively, we get

$$\sum_{j \in A} f(\{j\}) \geq f(A)$$

Lemma

If f is normalized and submodular, and $A \subseteq X$, then there is $j \in A$ such that $f(\{j\}) \geq \frac{1}{|A|} f(A)$.

Proof

- If A_1, A_2 partition A, then

$$f(A_1) + f(A_2) \geq f(A_1 \cup A_2) + f(A_1 \cap A_2) = f(A)$$

- Applying recursively, we get

$$\sum_{j \in A} f(\{j\}) \geq f(A)$$

- Therefore, $\max_{j \in A} f(\{j\}) \geq \frac{1}{|A|} f(A)$

Theorem

The greedy algorithm is a $(1 - 1/e)$ approximation algorithm for maximizing a monotone, normalized, and submodular function subject to a cardinality constraint.

Proof

- Let S be the working set in the algorithm

Theorem

The greedy algorithm is a $(1 - 1/e)$ approximation algorithm for maximizing a monotone, normalized, and submodular function subject to a cardinality constraint.

Proof

- Let S be the working set in the algorithm
- Let S^* be optimal solution with $f(S^*) = OPT$.

Theorem

The greedy algorithm is a $(1 - 1/e)$ approximation algorithm for maximizing a monotone, normalized, and submodular function subject to a cardinality constraint.

Proof

- Let S be the working set in the algorithm
- Let S^* be optimal solution with $f(S^*) = OPT$.
- We will show that the suboptimality $OPT - f(S)$ shrinks by a factor of $(1 - 1/k)$ each iteration

Theorem

The greedy algorithm is a $(1 - 1/e)$ approximation algorithm for maximizing a monotone, normalized, and submodular function subject to a cardinality constraint.

Proof

- Let S be the working set in the algorithm
- Let S^* be optimal solution with $f(S^*) = OPT$.
- We will show that the suboptimality $OPT - f(S)$ shrinks by a factor of $(1 - 1/k)$ each iteration
- After k iterations, it has shrunk to $(1 - 1/k)^k \leq 1/e$ from its original value

$$OPT - f(S) \leq \frac{1}{e} OPT$$

$$(1 - 1/e) OPT \leq f(S)$$

Theorem

The greedy algorithm is a $(1 - 1/e)$ approximation algorithm for maximizing a monotone, normalized, and submodular function subject to a cardinality constraint.

Proof

- By definition, in each iteration $f(S)$ increases by $\max_j f_S(\{j\})$

Theorem

The greedy algorithm is a $(1 - 1/e)$ approximation algorithm for maximizing a monotone, normalized, and submodular function subject to a cardinality constraint.

Proof

- By definition, in each iteration $f(S)$ increases by $\max_j f_S(\{j\})$
- By our lemmas, there is $j \in S^*$ s.t.

$$f_S(\{j\}) \geq \frac{1}{|S^*|} f_S(S^*)$$

Theorem

The greedy algorithm is a $(1 - 1/e)$ approximation algorithm for maximizing a monotone, normalized, and submodular function subject to a cardinality constraint.

Proof

- By definition, in each iteration $f(S)$ increases by $\max_j f_S(\{j\})$
- By our lemmas, there is $j \in S^*$ s.t.

$$
\begin{aligned}
f_S(\{j\}) &\geq \frac{1}{|S^*|} f_S(S^*) \\
&= \frac{1}{k}(f(S \cup S^*) - f(S))
\end{aligned}
$$

Theorem

The greedy algorithm is a $(1 - 1/e)$ approximation algorithm for maximizing a monotone, normalized, and submodular function subject to a cardinality constraint.

Proof

- By definition, in each iteration $f(S)$ increases by $\max_j f_S(\{j\})$
- By our lemmas, there is $j \in S^*$ s.t.

$$
\begin{aligned}
f_S(\{j\}) &\geq \frac{1}{|S^*|} f_S(S^*) \\
&= \frac{1}{k}(f(S \cup S^*) - f(S)) \\
&\geq \frac{1}{k}(OPT - f(S))
\end{aligned}
$$

Theorem

The greedy algorithm is a $(1 - 1/e)$ approximation algorithm for maximizing a monotone, normalized, and submodular function subject to a cardinality constraint.

Proof

- By definition, in each iteration $f(S)$ increases by $\max_j f_S(\{j\})$
- By our lemmas, there is $j \in S^*$ s.t.

$$\begin{aligned} f_S(\{j\}) &\geq \frac{1}{|S^*|} f_S(S^*) \\ &= \frac{1}{k}(f(S \cup S^*) - f(S)) \\ &\geq \frac{1}{k}(OPT - f(S)) \end{aligned}$$

- Therefore, suboptimality decreases by factor of $1 - \frac{1}{k}$, as needed.

Outline

1. Introduction
2. Cardinality Constraint
3. Matroid Constraint

From Uniform to Arbitrary Matroid

Problem Definition

Given a non-decreasing and normalized submodular function $f : 2^X \to \mathbb{R}_+$ on a finite ground set X, and a matroid $M = (X, \mathcal{I})$

$$\begin{array}{ll} \text{maximize} & f(S) \\ \text{subject to} & S \in \mathcal{I} \end{array}$$

From Uniform to Arbitrary Matroid

Problem Definition

Given a non-decreasing and normalized submodular function $f : 2^X \to \mathbb{R}_+$ on a finite ground set X, and a matroid $M = (X, \mathcal{I})$

$$\begin{array}{ll} \text{maximize} & f(S) \\ \text{subject to} & S \in \mathcal{I} \end{array}$$

- The discrete greedy algorithm is no longer a $1 - 1/e$ approximation
  - It is, however, a $1/2$ approximation

From Uniform to Arbitrary Matroid

Problem Definition

Given a non-decreasing and normalized submodular function $f : 2^X \to \mathbb{R}_+$ on a finite ground set X, and a matroid $M = (X, \mathcal{I})$

$$\begin{array}{ll} \text{maximize} & f(S) \\ \text{subject to} & S \in \mathcal{I} \end{array}$$

- The discrete greedy algorithm is no longer a $1 - 1/e$ approximation
  - It is, however, a $1/2$ approximation
- Nevertheless, a continuous greedy algorithm gives $1 - 1/e$

From Uniform to Arbitrary Matroid

Problem Definition

Given a non-decreasing and normalized submodular function $f : 2^X \to \mathbb{R}_+$ on a finite ground set X, and a matroid $M = (X, \mathcal{I})$

$$\begin{aligned} &\text{maximize} && f(S) \\ &\text{subject to} && S \in \mathcal{I} \end{aligned}$$

- The discrete greedy algorithm is no longer a $1 - 1/e$ approximation
  - It is, however, a $1/2$ approximation
- Nevertheless, a continuous greedy algorithm gives $1 - 1/e$
- Approach resembles that for minimization
  - Define a continous extension of f
  - Optimize continuous extension over matroid polytope
  - Extract an integer point

The Multilinear Extension

Multilinear Extension

Given a set function $f : \{0,1\}^n \to \mathbb{R}$, its multilinear extension $F : [0,1]^n \to \mathbb{R}$ evaluated at $x \in [0,1]^n$ gives the expected value of $f(S)$ for the random set S which includes each i independently with probability x_i.

$$F(x) = \sum_{S \subseteq X} f(S) \prod_{i \in S} x_i \prod_{i \neq S} (1 - x_i)$$

The Multilinear Extension

Multilinear Extension

Given a set function $f : \{0,1\}^n \to \mathbb{R}$, its multilinear extension $F : [0,1]^n \to \mathbb{R}$ evaluated at $x \in [0,1]^n$ gives the expected value of $f(S)$ for the random set S which includes each i independently with probability x_i.

$$F(x) = \sum_{S \subseteq X} f(S) \prod_{i \in S} x_i \prod_{i \neq S} (1 - x_i)$$

- For each point x, evaluates f on the independent distribution $D(x)$

The Multilinear Extension

Multilinear Extension

Given a set function $f : \{0,1\}^n \to \mathbb{R}$, its multilinear extension $F : [0,1]^n \to \mathbb{R}$ evaluated at $x \in [0,1]^n$ gives the expected value of $f(S)$ for the random set S which includes each i independently with probability x_i.

$$F(x) = \sum_{S \subseteq X} f(S) \prod_{i \in S} x_i \prod_{i \neq S} (1 - x_i)$$

- For each point x, evaluates f on the independent distribution $D(x)$
- Clearly an extension of f

The Multilinear Extension

Multilinear Extension

Given a set function $f : \{0,1\}^n \to \mathbb{R}$, its multilinear extension $F : [0,1]^n \to \mathbb{R}$ evaluated at $x \in [0,1]^n$ gives the expected value of $f(S)$ for the random set S which includes each i independently with probability x_i.

$$F(x) = \sum_{S \subseteq X} f(S) \prod_{i \in S} x_i \prod_{i \neq S} (1 - x_i)$$

- For each point x, evaluates f on the independent distribution $D(x)$
- Clearly an extension of f
- Not concave (or convex) in general
  - Recall f with $f(\emptyset) = 0$ and $f(\{1\}) = f(\{2\}) = f(\{1,2\}) = 1$
  - $F(x) = 1 - (1 - x_1)(1 - x_2)$

Easy Properties of the Multilinear Extension

Normalized

When f is normalized, $F(0) = 0$

Follows from the fact that F is an extension of f

Easy Properties of the Multilinear Extension

Normalized

When f is normalized, $F(0) = 0$

Follows from the fact that F is an extension of f

Nondecreasing

When f is monotone non-decreasing, $F(x) \leq F(y)$ whenever $x \preceq y$ component-wise.

Increasing the probability of selecting each element increases the expected value.

Up-concavity

Even though F is not concave, it is concave in "upwards" directions.

Up-concavity

Assume f is submodular. For every $\vec{a} \in [0,1]^n$ and $\vec{d} \in [0,1]^n$ satisfying $d \succeq 0$, the function $F(\vec{a} + \vec{d}\,t)$ is a concave function of $t \in \mathbb{R}$.

- This follows almost directly from diminishing marginal returns interpretation of submodularity.
- Proof sketch:
  - Up concave $\equiv$ mixed derivatives $\frac{\partial^2 F}{\partial x_i \partial x_j}$ negative everywhere
  - Negative mixed derivatives follow from diminishing marginal returns

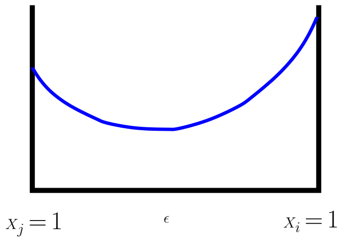
Cross-convexity

Nevertheless, F is convex in "cross" directions.

Cross-convexity

Assume f is submodular. For every $a \in [0,1]^n$ and $\vec{d} = e_i - e_j$ for some $i, j \in X$, the function $F(\vec{a} + \vec{d}\, t)$ is a convex function of $t \in \mathbb{R}$.

- i.e. trading off one item's probability for anothers gives a convex curve
- Follows from submodularity: as we "remove" j, the marginal benefit of "adding" i increases



Algorithm Outline

Step A: Continuous Greedy Algorithm

Computes a $1 - 1/e$ approximation to the following continuous (non-convex) optimization problem.

$$\begin{array}{ll} \text{maximize} & F(x) \\ \text{subject to} & x \in \mathcal{P}(\mathcal{M}) \end{array}$$

- i.e. Computes x^* s.t. $F(x^*) \geq (1 - 1/e) \max \{F(x) : x \in \mathcal{P}(\mathcal{M})\}$

Algorithm Outline

Step A: Continuous Greedy Algorithm

Computes a $1 - 1/e$ approximation to the following continuous (non-convex) optimization problem.

$$\begin{array}{ll} \text{maximize} & F(x) \\ \text{subject to} & x \in \mathcal{P}(\mathcal{M}) \end{array}$$

- i.e. Computes x^* s.t. $F(x^*) \geq (1 - 1/e) \max \{F(x) : x \in \mathcal{P}(\mathcal{M})\}$
- Note: $\max \{F(x) : x \in \mathcal{P}(\mathcal{M})\} \geq \max \{f(S) : S \in \mathcal{I}\}$

Algorithm Outline

Step A: Continuous Greedy Algorithm

Computes a $1 - 1/e$ approximation to the following continuous (non-convex) optimization problem.

$$\begin{array}{ll} \text{maximize} & F(x) \\ \text{subject to} & x \in \mathcal{P}(\mathcal{M}) \end{array}$$

- i.e. Computes x^* s.t. $F(x^*) \geq (1 - 1/e) \max \{F(x) : x \in \mathcal{P}(\mathcal{M})\}$
- Note: $\max \{F(x) : x \in \mathcal{P}(\mathcal{M})\} \geq \max \{f(S) : S \in \mathcal{I}\}$
- $D(x^*)$ is a distribution over sets with expected value at least $(1 - 1/e)$ of our target
- Are we done?

Algorithm Outline

Step A: Continuous Greedy Algorithm

Computes a $1 - 1/e$ approximation to the following continuous (non-convex) optimization problem.

$$\begin{array}{ll} \text{maximize} & F(x) \\ \text{subject to} & x \in \mathcal{P}(\mathcal{M}) \end{array}$$

- i.e. Computes x^* s.t. $F(x^*) \geq (1 - 1/e) \max \{F(x) : x \in \mathcal{P}(\mathcal{M})\}$
- Note: $\max \{F(x) : x \in \mathcal{P}(\mathcal{M})\} \geq \max \{f(S) : S \in \mathcal{I}\}$
- $D(x^*)$ is a distribution over sets with expected value at least $(1 - 1/e)$ of our target
- Are we done?

No! $D(x^*)$ may be mostly supported on infeasible sets (i.e. not independent in matroid $\mathcal{M}$).

Algorithm Outline

Step B: Pipage Rounding

"Rounds" x^* to some vertex y^* of the matroid polytope (i.e. an independent set) satisfying

$$F(y^*) \geq F(x^*)$$

Algorithm Outline

Step B: Pipage Rounding

"Rounds" x^* to some vertex y^* of the matroid polytope (i.e. an independent set) satisfying

$$F(y^*) \geq F(x^*)$$

- A-priori, not obvious that such a y^* exists

- The following "continuous" descent algorithm works for an arbitrary nondecreasing and up-concave function F, and solvable downwards-closed polytope $\mathcal{P} \subseteq \mathbb{R}^n_+$.
- Continuously moves a particle inside the matroid polytope, starting at 0, for a total of 1 time unit.
  - Position at time t given by $x(t)$.
- Discretized to time steps of ϵ, which we will assume to be arbitrarily small for convenience of analysis, but may be taken to be $1/\operatorname{poly}(n)$ in the actual implementation.

- The following “continuous” descent algorithm works for an arbitrary nondecreasing and up-concave function F, and solvable downwards-closed polytope $\mathcal{P} \subseteq \mathbb{R}_+^n$.
- Continuously moves a particle inside the matroid polytope, starting at 0, for a total of 1 time unit.
  - Position at time t given by $x(t)$.
- Discretized to time steps of ϵ, which we will assume to be arbitrarily small for convenience of analysis, but may be taken to be $1/\operatorname{poly}(n)$ in the actual implementation.

Continuous Greedy Algorithm (F,$\mathcal{P}$, ϵ)

1. $x(0) \leftarrow \vec{0}$
2. For $t \in [0, \epsilon, 2\epsilon, \ldots, 1-\epsilon]$
   - $x(t+\epsilon) \leftarrow x(t) + \epsilon \operatorname{argmax}_{y \in \mathcal{P}} \{\triangledown F(x(t)) \cdot y\}$
3. Return $x(1)$

Continuous Greedy Algorithm (F,$\mathcal{P}$, ϵ)

1. $x(0) \leftarrow \vec{0}$
2. For $t \in [0, \epsilon, 2\epsilon, \ldots, 1-\epsilon]$
   - $x(t+\epsilon) \leftarrow x(t) + \epsilon \operatorname{argmax}_{y \in \mathcal{P}} \{\triangledown F(x(t)) \cdot y\}$
3. Return $x(1)$

- I.e. When the particle is at x, it moves in direction y maximizing the linear function $\triangledown F(x) \cdot y$ over $y \in \mathcal{P}$
  - The direction is actually a vertex of our matroid polytope
  - This is NOT gradient descent

Continuous Greedy Algorithm (F,$\mathcal{P}$, ϵ)

1. $x(0) \leftarrow \vec{0}$
2. For $t \in [0, \epsilon, 2\epsilon, \ldots, 1-\epsilon]$
   - $x(t+\epsilon) \leftarrow x(t) + \epsilon \operatorname{argmax}_{y \in \mathcal{P}} \{\bigtriangledown F(x(t)) \cdot y\}$
3. Return $x(1)$

- I.e. When the particle is at x, it moves in direction y maximizing the linear function $\bigtriangledown F(x) \cdot y$ over $y \in \mathcal{P}$
  - The direction is actually a vertex of our matroid polytope
  - This is NOT gradient descent
- Observe: Algorithm forms a convex combination of $\frac{1}{\epsilon}$ vertices of the polytope $\mathcal{P}$, each with weight ϵ.
  - $x(1) \in \mathcal{P}$.

Theorem

Let F be nondecreasing and up-concave, and $\mathcal{P}$ be a downwards closed polytope. In the limit as $\epsilon \to 0$, the continuous greedy algorithm outputs a $1 - 1/e$ approximation to maximizing $F(x)$ over $\mathcal{P}$.

Theorem

Let F be nondecreasing and up-concave, and $\mathcal{P}$ be a downwards closed polytope. In the limit as $\epsilon \to 0$, the continuous greedy algorithm outputs a $1 - 1/e$ approximation to maximizing $F(x)$ over $\mathcal{P}$.

Proof Sketch

- Denote $y(t) = \text{argmax}_{y \in \mathcal{P}} \bigtriangledown F(x(t)) \cdot y$
  - $\frac{d\vec{x}}{dt} = y(t)$

Theorem

Let F be nondecreasing and up-concave, and $\mathcal{P}$ be a downwards closed polytope. In the limit as $\epsilon \to 0$, the continuous greedy algorithm outputs a $1 - 1/e$ approximation to maximizing $F(x)$ over $\mathcal{P}$.

Proof Sketch

- Denote $y(t) = \text{argmax}_{y \in \mathcal{P}} \bigtriangledown F(x(t)) \cdot y$
  - $\frac{d\vec{x}}{dt} = y(t)$
- Let x_{opt} be the vertex of $\mathcal{P}(\mathcal{M})$ maximizing $F(x)$.
  - $F(x_{opt}) = f(x_{opt}) = OPT$

Theorem

Let F be nondecreasing and up-concave, and $\mathcal{P}$ be a downwards closed polytope. In the limit as $\epsilon \to 0$, the continuous greedy algorithm outputs a $1 - 1/e$ approximation to maximizing $F(x)$ over $\mathcal{P}$.

Proof Sketch

- Denote $y(t) = \operatorname{argmax}_{y \in \mathcal{P}} \triangledown F(x(t)) \cdot y$
  - $\frac{d\vec{x}}{dt} = y(t)$
- Let x_{opt} be the vertex of $\mathcal{P}(\mathcal{M})$ maximizing $F(x)$.
  - $F(x_{opt}) = f(x_{opt}) = OPT$

$$\frac{dF(x(t))}{dt}$$

$$\geq OPT - F(x(t))$$

Theorem

Let F be nondecreasing and up-concave, and $\mathcal{P}$ be a downwards closed polytope. In the limit as $\epsilon \to 0$, the continuous greedy algorithm outputs a $1 - 1/e$ approximation to maximizing $F(x)$ over $\mathcal{P}$.

Proof Sketch

- Denote $y(t) = \text{argmax}_{y \in \mathcal{P}} \bigtriangledown F(x(t)) \cdot y$
  - $\frac{d\vec{x}}{dt} = y(t)$
- Let x_{opt} be the vertex of $\mathcal{P}(\mathcal{M})$ maximizing $F(x)$.
  - $F(x_{opt}) = f(x_{opt}) = OPT$

$$\frac{dF(x(t))}{dt} = \bigtriangledown F(x(t)) \cdot \frac{d\vec{x}}{dt}$$

$$\geq OPT - F(x(t))$$

Theorem

Let F be nondecreasing and up-concave, and $\mathcal{P}$ be a downwards closed polytope. In the limit as $\epsilon \to 0$, the continuous greedy algorithm outputs a $1 - 1/e$ approximation to maximizing $F(x)$ over $\mathcal{P}$.

Proof Sketch

- Denote $y(t) = \text{argmax}_{y \in \mathcal{P}} \bigtriangledown F(x(t)) \cdot y$
  - $\frac{d\vec{x}}{dt} = y(t)$
- Let x_{opt} be the vertex of $\mathcal{P}(\mathcal{M})$ maximizing $F(x)$.
  - $F(x_{opt}) = f(x_{opt}) = OPT$

$$\begin{aligned} \frac{dF(x(t))}{dt} &= \bigtriangledown F(x(t)) \cdot \frac{d\vec{x}}{dt} \\ &= \bigtriangledown F(x(t)) \cdot y(t) \end{aligned}$$

$$\geq OPT - F(x(t))$$

Theorem

Let F be nondecreasing and up-concave, and $\mathcal{P}$ be a downwards closed polytope. In the limit as $\epsilon \to 0$, the continuous greedy algorithm outputs a $1 - 1/e$ approximation to maximizing $F(x)$ over $\mathcal{P}$.

Proof Sketch

- Denote $y(t) = \text{argmax}_{y \in \mathcal{P}} \bigtriangledown F(x(t)) \cdot y$
  - $\frac{d\vec{x}}{dt} = y(t)$
- Let x_{opt} be the vertex of $\mathcal{P}(\mathcal{M})$ maximizing $F(x)$.
  - $F(x_{opt}) = f(x_{opt}) = OPT$

$$\begin{aligned} \frac{dF(x(t))}{dt} &= \bigtriangledown F(x(t)) \cdot \frac{d\vec{x}}{dt} \\ &= \bigtriangledown F(x(t)) \cdot y(t) \\ &\geq \bigtriangledown F(x(t)) \cdot [x_{opt} - x(t)]^{+} \\ &\geq OPT - F(x(t)) \end{aligned}$$

Theorem

Let F be nondecreasing and up-concave, and $\mathcal{P}$ be a downwards closed polytope. In the limit as $\epsilon \to 0$, the continuous greedy algorithm outputs a $1 - 1/e$ approximation to maximizing $F(x)$ over $\mathcal{P}$.

Proof Sketch

- Denote $y(t) = \text{argmax}_{y \in \mathcal{P}} \bigtriangledown F(x(t)) \cdot y$
  - $\frac{d\vec{x}}{dt} = y(t)$
- Let x_{opt} be the vertex of $\mathcal{P}(\mathcal{M})$ maximizing $F(x)$.
  - $F(x_{opt}) = f(x_{opt}) = OPT$

$$\begin{aligned}\frac{dF(x(t))}{dt} &= \bigtriangledown F(x(t)) \cdot \frac{d\vec{x}}{dt} \\ &= \bigtriangledown F(x(t)) \cdot y(t) \\ &\geq \bigtriangledown F(x(t)) \cdot [x_{opt} - x(t)]^+ \\ &= \bigtriangledown F(x(t)) \cdot [\max(x_{opt}, x(t)) - x(t)] \\ \\ &\geq OPT - F(x(t))\end{aligned}$$

Theorem

Let F be nondecreasing and up-concave, and $\mathcal{P}$ be a downwards closed polytope. In the limit as $\epsilon \to 0$, the continuous greedy algorithm outputs a $1 - 1/e$ approximation to maximizing $F(x)$ over $\mathcal{P}$.

Proof Sketch

- Denote $y(t) = \text{argmax}_{y \in \mathcal{P}} \bigtriangledown F(x(t)) \cdot y$
  - $\frac{d\vec{x}}{dt} = y(t)$
- Let x_{opt} be the vertex of $\mathcal{P}(\mathcal{M})$ maximizing $F(x)$.
  - $F(x_{opt}) = f(x_{opt}) = OPT$

$$\begin{aligned} \frac{dF(x(t))}{dt} &= \bigtriangledown F(x(t)) \cdot \frac{d\vec{x}}{dt} \\ &= \bigtriangledown F(x(t)) \cdot y(t) \\ &\geq \bigtriangledown F(x(t)) \cdot [x_{opt} - x(t)]^+ \\ &= \bigtriangledown F(x(t)) \cdot [\max(x_{opt}, x(t)) - x(t)] \\ &\geq F(\max(x_{opt}, x(t))) - F(x(t)) \\ &\geq OPT - F(x(t)) \end{aligned}$$

Theorem

Let F be nondecreasing and up-concave, and $\mathcal{P}$ be a downwards closed polytope. In the limit as $\epsilon \to 0$, the continuous greedy algorithm outputs a $1 - 1/e$ approximation to maximizing $F(x)$ over $\mathcal{P}$.

Proof Sketch

- $v(t) = F(x(t))$ satisfies $\frac{dv}{dt} \geq OPT - v$.
- Differential equation $\frac{dv}{dt} = OPT - v$ with boundary condition $v(0) = 0$ has a unique solution

$$v(t) = OPT(1 - e^{-t})$$

- $v(1) \geq OPT(1 - 1/e)$

Implementation Details

Continuous Greedy Algorithm (F,$\mathcal{P}$, ϵ)

1. $x(0) \leftarrow \vec{0}$
2. For $t \in [0, \epsilon, 2\epsilon, \ldots, 1-\epsilon]$
   - $x(t+\epsilon) \leftarrow x(t) + \epsilon \operatorname{argmax}_{y \in \mathcal{P}} \{\bigtriangledown F(x(t)) \cdot y\}$
3. Return $x(1)$

- $\bigtriangledown F(x)$ is not readily available, but can be estimated “accurately enough” using $\mathrm{poly}(n)$ random samples from $D(x)$, w.h.p.
- Step 2 can be implemented because $\mathcal{P}$ is solvable
- Discretization: Taking $\epsilon = 1/O(n^2)$ is “fine enough”
- Both the above introduce error into the approximation guarantee, yielding $1 - 1/e - 1/O(n)$ w.h.p
- This can be shaved off to $1 - 1/e$ with some additional “tricks”.

- The following algorithm takes x in matroid base polytope $\mathcal{P}_{base}(\mathcal{M})$, and non-decreasing cross-convex function F, and outputs integral y with $F(y) \geq F(x)$

PipageRounding ($\mathcal{M}$,x, F)

Whle x contains a fractional entry

1. Let T be the minimum-size tight set containing some fractional x_i
   - i.e. $x(T) = rank_{\mathcal{M}}(T)$, and $i \in T$.
2. Let $j \in T$ be such that $j \neq i$ and x_j is fractional.
3. Let $x(\mu) = x + \mu(e_i - e_j)$, and maximize $F(x(\mu))$ subject to $x(\mu) \in \mathcal{P}(\mathcal{M})$.
4. $x \leftarrow x(\mu)$.

PipageRounding ($\mathcal{M}$,x, F)

Whle x contains a fractional entry

1. Let T be the minimum-size tight set containing some fractional x_i
   - i.e. $x(T) = rank_{\mathcal{M}}(T)$, and $i \in T$.
2. Let $j \in T$ be such that $j \neq i$ and x_j is fractional.
3. Let $x(\mu) = x + \mu(e_i - e_j)$, and maximize $F(x(\mu))$ subject to $x(\mu) \in \mathcal{P}(\mathcal{M})$.
4. $x \leftarrow x(\mu)$.

Step 1

- T exists because tight sets with respect to $x \in \mathcal{P}(\mathcal{M})$ form a lattice
- Proof:
  - Minimizers of a submodular function form a lattice (implied by submodular inequality).
  - Tight sets in x are the minimizers of the set function $rank_{\mathcal{M}}(S) - x(S)$
  - This set function is submodular.

PipageRounding ($\mathcal{M}$,x, F)

Whle x contains a fractional entry

1. Let T be the minimum-size tight set containing some fractional x_i
   - i.e. $x(T) = rank_{\mathcal{M}}(T)$, and $i \in T$.
2. Let $j \in T$ be such that $j \neq i$ and x_j is fractional.
3. Let $x(\mu) = x + \mu(e_i - e_j)$, and maximize $F(x(\mu))$ subject to $x(\mu) \in \mathcal{P}(\mathcal{M})$.
4. $x \leftarrow x(\mu)$.

Step 2

- Since rank is integer valued, any tight set containing fractional variable should have another.

PipageRounding ($\mathcal{M}$,x, F)

Whle x contains a fractional entry

1. Let T be the minimum-size tight set containing some fractional x_i
   - i.e. $x(T) = rank_{\mathcal{M}}(T)$, and $i \in T$.
2. Let $j \in T$ be such that $j \neq i$ and x_j is fractional.
3. Let $x(\mu) = x + \mu(e_i - e_j)$, and maximize $F(x(\mu))$ subject to $x(\mu) \in \mathcal{P}(\mathcal{M})$.
4. $x \leftarrow x(\mu)$.

Step 3

- Either the number of fractional variables decreases, or a smaller tight set containing x_i or x_j is created.
- This leads to termination after $O(n^2)$ iterations