CS675: Convex and Combinatorial Optimization
Spring 2018
Introduction to Matroid Theory

Instructor: Shaddin Dughmi

Optimization over Sets

- Most combinatorial optimization problems can be thought of as choosing the best set from a family of allowable sets
  - Shortest paths
  - Max-weight matching
  - Independent set
  - ...

Optimization over Sets

- Most combinatorial optimization problems can be thought of as choosing the best set from a family of allowable sets
  - Shortest paths
  - Max-weight matching
  - Independent set
  - ...
- Set system: Pair $(\mathcal{X}, \mathcal{I})$ where $\mathcal{X}$ is a finite ground set and $\mathcal{I} \subseteq 2^{\mathcal{X}}$ are the feasible sets

Optimization over Sets

- Most combinatorial optimization problems can be thought of as choosing the best set from a family of allowable sets
  - Shortest paths
  - Max-weight matching
  - Independent set
  - ...
- Set system: Pair $(\mathcal{X}, \mathcal{I})$ where $\mathcal{X}$ is a finite ground set and $\mathcal{I} \subseteq 2^{\mathcal{X}}$ are the feasible sets
- Objective: often "linear", referred to as modular

Optimization over Sets

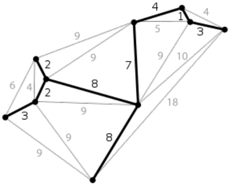
- Most combinatorial optimization problems can be thought of as choosing the best set from a family of allowable sets
  - Shortest paths
  - Max-weight matching
  - Independent set
  - ...
- Set system: Pair $(\mathcal{X}, \mathcal{I})$ where $\mathcal{X}$ is a finite ground set and $\mathcal{I} \subseteq 2^{\mathcal{X}}$ are the feasible sets
- Objective: often "linear", referred to as modular
- Analogues of concave and convex: submodular and supermodular (in no particular order!)

Optimization over Sets

- Most combinatorial optimization problems can be thought of as choosing the best set from a family of allowable sets
  - Shortest paths
  - Max-weight matching
  - Independent set
  - ...
- Set system: Pair $(\mathcal{X}, \mathcal{I})$ where $\mathcal{X}$ is a finite ground set and $\mathcal{I} \subseteq 2^{\mathcal{X}}$ are the feasible sets
- Objective: often "linear", referred to as modular
- Analogues of concave and convex: submodular and supermodular (in no particular order!)
- Today, we will look only at optimizing modular objectives over an extremely prolific family of set systems
  - Related, directly or indirectly, to a large fraction of optimization problems in P
  - Also pops up in submodular/supermodular optimization problems

Outline

Maximum Weight Forest Problem



Given a connected undirected graph $G = (V, E)$, and weights $w_e \in \mathbb{R}$ on edges e, find a maximum weight acyclic subgraph (aka forest) of G.

- Slight generalization of minimum weight spanning tree
- We use n and m to denote $|V|$ and $|E|$, respectively.

The Greedy Algorithm

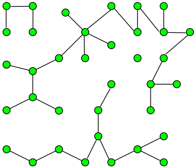
1. $B \leftarrow \emptyset$
2. Sort non-negative weight edges in decreasing order of weight
   - $e_1, \ldots, e_m$, with $w_1 \geq w_2 \geq \ldots \geq w_m \geq 0$
3. For $i = 1$ to m:
   - if $B \bigcup \{e_i\}$ is acyclic, add e_i to B.

The Greedy Algorithm

1. $B \leftarrow \emptyset$
2. Sort non-negative weight edges in decreasing order of weight
   - $e_1, \ldots, e_m$, with $w_1 \geq w_2 \geq \ldots \geq w_m \geq 0$
3. For $i = 1$ to m:
   - if $B \bigcup \{e_i\}$ is acyclic, add e_i to B.

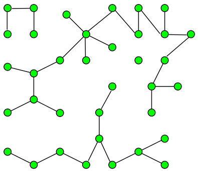
Theorem

The greedy algorithm outputs a maximum-weight forest.



Lemma

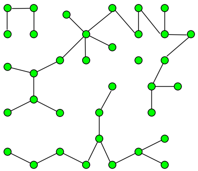
1. The empty set is acyclic
2. If A is an acyclic set of edges, and $B \subseteq A$, then B is also acyclic.
3. If A, B are acyclic, and $|B| > |A|$, then there is $e \in B \setminus A$ such that $A \bigcup \{e\}$ is acyclic



Lemma

1. The empty set is acyclic
2. If A is an acyclic set of edges, and $B \subseteq A$, then B is also acyclic.
3. If A, B are acyclic, and $|B| > |A|$, then there is $e \in B \setminus A$ such that $A \bigcup \{e\}$ is acyclic

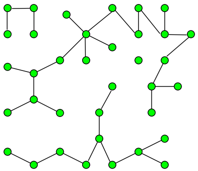
(1) and (2) are trivial, so let's prove (3)



Lemma

1. The empty set is acyclic
2. If A is an acyclic set of edges, and $B \subseteq A$, then B is also acyclic.
3. If A, B are acyclic, and $|B| > |A|$, then there is $e \in B \setminus A$ such that $A \bigcup \{e\}$ is acyclic

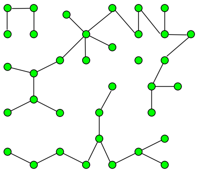
- Sub-lemma: if C is acyclic, then $|C| = n - \#components(C)$.
  - Induction



Lemma

1. The empty set is acyclic
2. If A is an acyclic set of edges, and $B \subseteq A$, then B is also acyclic.
3. If A, B are acyclic, and $|B| > |A|$, then there is $e \in B \setminus A$ such that $A \bigcup \{e\}$ is acyclic

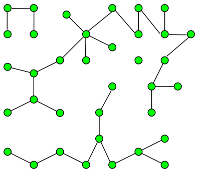
- Sub-lemma: if C is acyclic, then $|C| = n - \#components(C)$.
  - Induction
- When $|B| > |A|$, this means $\#components(B) < \#components(A)$



Lemma

1. The empty set is acyclic
2. If A is an acyclic set of edges, and $B \subseteq A$, then B is also acyclic.
3. If A, B are acyclic, and $|B| > |A|$, then there is $e \in B \setminus A$ such that $A \bigcup \{e\}$ is acyclic

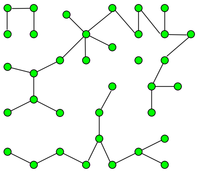
- Sub-lemma: if C is acyclic, then $|C| = n - \#components(C)$.
  - Induction
- When $|B| > |A|$, this means $\#components(B) < \#components(A)$
- Can't be that all $e \in B$ are "inside" connected components of A



Lemma

1. The empty set is acyclic
2. If A is an acyclic set of edges, and $B \subseteq A$, then B is also acyclic.
3. If A, B are acyclic, and $|B| > |A|$, then there is $e \in B \setminus A$ such that $A \bigcup \{e\}$ is acyclic

- Sub-lemma: if C is acyclic, then $|C| = n - \#components(C)$.
  - Induction
- When $|B| > |A|$, this means $\#components(B) < \#components(A)$
- Can't be that all $e \in B$ are "inside" connected components of A
- Some $e \in B$ must "go between" connected components of A.



Lemma

1. The empty set is acyclic
2. If A is an acyclic set of edges, and $B \subseteq A$, then B is also acyclic.
   - Contrapositive: if B cyclic then so is A
3. If A, B are acyclic, and $|B| > |A|$, then there is $e \in B \setminus A$ such that $A \bigcup \{e\}$ is acyclic
   - Inductively: can extend A by adding $|B| - |A|$ elements from $B \setminus A$

- Sub-lemma: if C is acyclic, then $|C| = n - \#components(C)$.
  - Induction
- When $|B| > |A|$, this means $\#components(B) < \#components(A)$
- Can't be that all $e \in B$ are "inside" connected components of A
- Some $e \in B$ must "go between" connected components of A.

Proof

Going back to proving the algorithm correct.

Inductive Hypothesis (i)

There is a maximum-weight acyclic forest B_i^* which "agrees" with the algorithm's choices on edges $e_1, \ldots, e_i$.

- i.e. if B_i denotes the algorithm's choice up to iteration i, then $B_i = B_i^* \bigcap \{e_1, \ldots, e_i\}$

Proof

Going back to proving the algorithm correct.

Inductive Hypothesis (i)

There is a maximum-weight acyclic forest B_i^* which "agrees" with the algorithm's choices on edges $e_1, \ldots, e_i$.

- i.e. if B_i denotes the algorithm's choice up to iteration i, then $B_i = B_i^* \bigcap \{e_1, \ldots, e_i\}$

- Assume true for step $i-1$, and consider step i

Proof

Going back to proving the algorithm correct.

Inductive Hypothesis (i)

There is a maximum-weight acyclic forest B_i^* which "agrees" with the algorithm's choices on edges $e_1, \ldots, e_i$.

- i.e. if B_i denotes the algorithm's choice up to iteration i, then $B_i = B_i^* \bigcap \{e_1, \ldots, e_i\}$

- Assume true for step $i-1$, and consider step i
- If $e_i \notin B_i$, then $B_{i-1} \bigcup \{e_i\}$ is cyclic. Since $B_{i-1} \subseteq B_{i-1}^*$, then $e_i \notin B_{i-1}^*$ (Property 2). So take $B_i^* = B_{i-1}^*$.

Proof

Going back to proving the algorithm correct.

Inductive Hypothesis (i)

There is a maximum-weight acyclic forest B_i^* which "agrees" with the algorithm's choices on edges $e_1, \ldots, e_i$.

- i.e. if B_i denotes the algorithm's choice up to iteration i, then $B_i = B_i^* \bigcap \{e_1, \ldots, e_i\}$

- Assume true for step $i-1$, and consider step i
- If $e_i \notin B_i$, then $B_{i-1} \bigcup \{e_i\}$ is cyclic. Since $B_{i-1} \subseteq B_{i-1}^*$, then $e_i \notin B_{i-1}^*$ (Property 2). So take $B_i^* = B_{i-1}^*$.
- If $e_i \in B_i$ and $e_i \notin B_{i-1}^*$, build B_i^* by repeatedly extending B_i using B_{i-1}^* (property 3)
  - Recall that $B_i = B_{i-1} \bigcup \{e_i\}$ agrees with B_{i-1}^* on $e_1, \ldots, e_{i-1}$.
  - $B_i^* = B_{i-1}^* \bigcup \{e_i\} \setminus \{e_k\}$ for some $k > i$
  - B_i^* has weight no less than B_{i-1}^*, so optimal.

To prove optimality of the greedy algorithm, all we needed was the following.

Matroids

A set system $M = (\mathcal{X}, \mathcal{I})$ is a matroid if

1. $\emptyset \in \mathcal{I}$
2. If $A \in \mathcal{I}$ and $B \subseteq A$, then $B \in \mathcal{I}$ (Downward Closure)
3. If $A, B \in \mathcal{I}$ and $|B| > |A|$, then $\exists\, x \in B \setminus A$ such that $A \bigcup \{x\} \in \mathcal{I}$ (Exchange Property)

To prove optimality of the greedy algorithm, all we needed was the following.

Matroids

A set system $M = (\mathcal{X}, \mathcal{I})$ is a matroid if

1. $\emptyset \in \mathcal{I}$
2. If $A \in \mathcal{I}$ and $B \subseteq A$, then $B \in \mathcal{I}$ (Downward Closure)
3. If $A, B \in \mathcal{I}$ and $|B| > |A|$, then $\exists\, x \in B \setminus A$ such that $A \bigcup \{x\} \in \mathcal{I}$ (Exchange Property)

- Three conditions above are called the matroid axioms
- $A \in \mathcal{I}$ is called an independent set of the matroid.

To prove optimality of the greedy algorithm, all we needed was the following.

Matroids

A set system $M = (\mathcal{X}, \mathcal{I})$ is a matroid if

1. $\emptyset \in \mathcal{I}$
2. If $A \in \mathcal{I}$ and $B \subseteq A$, then $B \in \mathcal{I}$ (Downward Closure)
3. If $A, B \in \mathcal{I}$ and $|B| > |A|$, then $\exists\, x \in B \setminus A$ such that $A \bigcup \{x\} \in \mathcal{I}$ (Exchange Property)

- Three conditions above are called the matroid axioms
- $A \in \mathcal{I}$ is called an independent set of the matroid.
- The matroid whose independent sets are acyclic subgraphs is called a graphic matroid
- Other examples abound!

Example: Linear Matroid

- $\mathcal{X}$ is a finite set of vectors $\{v_1, \ldots, v_m\} \subseteq \mathbb{R}^n$
- $S \in \mathcal{I}$ iff the vectors in S are linearly independent

- Downward closure: If a set of vectors is linearly independent, then every subset of it is also
- Exchange property: Can always extend a low-dimension independent set S by adding vectors from a higher dimension independent set T

Example: Uniform Matroid

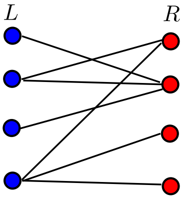
- $\mathcal{X}$ is an arbitrary finite set $\{1, \ldots, n\}$.
- $S \in \mathcal{I}$ iff $|S| \leq k$.

- Downward closure: If a set S has $|S| \leq k$ then the same holds for $T \subseteq S$.
- Exchange property: If $|S| < |T| \leq k$, then there is an element in $T \setminus S$, and we can add it to S while preserving independence.

Example: Partition Matroid

- $\mathcal{X}$ is the disjoint union of classes $X_1, \ldots, X_m$
- Each class X_j has an upperbound k_j.
- $S \in \mathcal{I}$ iff $|S \bigcap X_j| \leq k_j$ for all j

- This is the “disjoint union” of a number of uniform matroids



Example: Transversal Matroid

- Described by a bipartite graph $E \subseteq L \times R$
- $\mathcal{X} = L$
- $S \in \mathcal{I}$ iff there is a bipartite matching which matches S

- Downward closure: If we can match S, then we can match $T \subseteq S$.
- Exchange property: If $|T| > |S|$ is matchable, then an augmenting path/alternating path extends the matching of S to some $x \in T \setminus S$.

The Greedy Algorithm on Matroids

The Greedy Algorithm

1. $B \leftarrow \emptyset$
2. Sort nonnegative elements of $\mathcal{X}$ in decreasing order of weight
   - $\{1, \ldots, n\}$ with $w_1 \geq w_2, \geq \ldots \geq w_n \geq 0$.
3. For $i = 1$ to n:
   - if $B \bigcup \{i\} \in \mathcal{I}$, add i to B.

Theorem

The greedy algorithm returns the maximum weight set for every choice of weights if and only if the set system $(\mathcal{X}, \mathcal{I})$ is a matroid.

We already saw the "if" direction. We will skip "only if".

The Greedy Algorithm on Matroids

The Greedy Algorithm

1. $B \leftarrow \emptyset$
2. Sort nonnegative elements of $\mathcal{X}$ in decreasing order of weight
   - $\{1, \ldots, n\}$ with $w_1 \geq w_2, \geq \ldots \geq w_n \geq 0$.
3. For $i = 1$ to n:
   - if $B \bigcup \{i\} \in \mathcal{I}$, add i to B.

The Greedy Algorithm on Matroids

The Greedy Algorithm

1. $B \leftarrow \emptyset$
2. Sort nonnegative elements of $\mathcal{X}$ in decreasing order of weight
   - $\{1, \ldots, n\}$ with $w_1 \geq w_2, \geq \ldots \geq w_n \geq 0$.
3. For $i = 1$ to n:
   - if $B \bigcup \{i\} \in \mathcal{I}$, add i to B.

- To implement this, we need an independence oracle for step 3
  - A subroutine which checks whether $S \in \mathcal{I}$ or not.
- Runs in time $O(n \log n) + nT$, where T is runtime of the independence oracle.

The Greedy Algorithm on Matroids

The Greedy Algorithm

1. $B \leftarrow \emptyset$
2. Sort nonnegative elements of $\mathcal{X}$ in decreasing order of weight
   - $\{1, \ldots, n\}$ with $w_1 \geq w_2, \geq \ldots \geq w_n \geq 0$.
3. For $i = 1$ to n:
   - if $B \bigcup \{i\} \in \mathcal{I}$, add i to B.

- To implement this, we need an independence oracle for step 3
  - A subroutine which checks whether $S \in \mathcal{I}$ or not.
- Runs in time $O(n \log n) + nT$, where T is runtime of the independence oracle.
- For most "natural" matroids, independence oracle is easy to implement efficiently
  - Graphic matroid
  - Linear matroid
  - Uniform/partition matroid
  - Transversal matroid

Outline

Independent Sets, Bases, and Circuits

Consider a matroid $\mathcal{M} = (\mathcal{X}, \mathcal{I})$.

- An independent set is a set $A \in \mathcal{I}$.

Independent Sets, Bases, and Circuits

Consider a matroid $\mathcal{M} = (\mathcal{X}, \mathcal{I})$.

- An independent set is a set $A \in \mathcal{I}$.
- A base of $\mathcal{M}$ is a maximal independent set

Independent Sets, Bases, and Circuits

Consider a matroid $\mathcal{M} = (\mathcal{X}, \mathcal{I})$.

- An independent set is a set $A \in \mathcal{I}$.
- A base of $\mathcal{M}$ is a maximal independent set
- A base of $S \subseteq \mathcal{X}$ in $\mathcal{M}$ is maximal independent subset of S
  - I.e. a base of the matroid after deleting $\overline{S}$.

Independent Sets, Bases, and Circuits

Consider a matroid $\mathcal{M} = (\mathcal{X}, \mathcal{I})$.

- An independent set is a set $A \in \mathcal{I}$.
- A base of $\mathcal{M}$ is a maximal independent set
- A base of $S \subseteq \mathcal{X}$ in $\mathcal{M}$ is maximal independent subset of S
  - I.e. a base of the matroid after deleting $\overline{S}$.
- A circuit $S \subseteq \mathcal{X}$ is a minimal dependent subset of $\mathcal{X}$

Independent Sets, Bases, and Circuits

Consider a matroid $\mathcal{M} = (\mathcal{X}, \mathcal{I})$.

- An independent set is a set $A \in \mathcal{I}$.
- A base of $\mathcal{M}$ is a maximal independent set
- A base of $S \subseteq \mathcal{X}$ in $\mathcal{M}$ is maximal independent subset of S
  - I.e. a base of the matroid after deleting $\overline{S}$.
- A circuit $S \subseteq \mathcal{X}$ is a minimal dependent subset of $\mathcal{X}$

What are these for:

- Graphic matroid
- Linear matroid
- Uniform matroid
- Partition matroid
- Transversal matroid

Rank

Lemma

For every $S \subseteq \mathcal{X}$, all bases of S in $\mathcal{M}$ have the same cardinality.

- Special case of $S = \mathcal{X}$: all bases of $\mathcal{M}$ have the same cardinality.
- Should remind you of vector space dimension

Rank

Lemma

For every $S \subseteq \mathcal{X}$, all bases of S in $\mathcal{M}$ have the same cardinality.

- Special case of $S = \mathcal{X}$: all bases of $\mathcal{M}$ have the same cardinality.
- Should remind you of vector space dimension

- Follows directly from the exchange property.

Rank

Lemma

For every $S \subseteq \mathcal{X}$, all bases of S in $\mathcal{M}$ have the same cardinality.

- Special case of $S = \mathcal{X}$: all bases of $\mathcal{M}$ have the same cardinality.
- Should remind you of vector space dimension

- Follows directly from the exchange property.

The following analogue of vector space dimension is well-defined.

Rank

- The Rank of $S \subseteq \mathcal{X}$ in $\mathcal{M}$ is the size of the maximal independent subsets (i.e. bases) of S.
- The rank of $\mathcal{M}$ is the size of the bases of $\mathcal{M}$.
- The function $rank_{\mathcal{M}}(S) : 2^{\mathcal{X}} \rightarrow \mathbb{N}$ is called the rank function of $\mathcal{M}$.

Rank

Lemma

For every $S \subseteq \mathcal{X}$, all bases of S in $\mathcal{M}$ have the same cardinality.

- Special case of $S = \mathcal{X}$: all bases of $\mathcal{M}$ have the same cardinality.
- Should remind you of vector space dimension

- Follows directly from the exchange property.

The following analogue of vector space dimension is well-defined.

Rank

- The Rank of $S \subseteq \mathcal{X}$ in $\mathcal{M}$ is the size of the maximal independent subsets (i.e. bases) of S.
- The rank of $\mathcal{M}$ is the size of the bases of $\mathcal{M}$.
- The function $rank_{\mathcal{M}}(S) : 2^{\mathcal{X}} \to \mathbb{N}$ is called the rank function of $\mathcal{M}$.

E.g.: Graphic matroid, linear matroid, transversal matroid

Span

Span

Given $S \subseteq \mathcal{X}$, $span(S) = \{i \in \mathcal{X} : rank(S) = rank(S \bigcup \{i\})\}$

- i.e. S itself, plus the elements which would form a circuit if added to a base of S
- e.g.: Linear matroid, graphic matroid, uniform matroid.

Span

Span

Given $S \subseteq \mathcal{X}$, $span(S) = \{i \in \mathcal{X} : rank(S) = rank(S \bigcup \{i\})\}$

- i.e. S itself, plus the elements which would form a circuit if added to a base of S
- e.g.: Linear matroid, graphic matroid, uniform matroid.

Observation

$i \in \{1, \ldots, n\}$ is selected by the greedy algorithm iff
$i \notin span(\{1, \ldots, i-1\})$

Operations preserving Matroidness

Given $\mathcal{M} = (\mathcal{X}, \mathcal{I})$, consider the following operations:

- Deletion: For $B \subseteq \mathcal{X}$, we define $\mathcal{M} \setminus B = (\mathcal{X}', \mathcal{I}')$ with $\mathcal{X}' = \mathcal{X} \setminus B$,

$$\mathcal{I}' = \{S \subseteq \mathcal{X}' : S \in \mathcal{I}\}$$

  - Graphic: deleting edges from the graph

Operations preserving Matroidness

Given $\mathcal{M} = (\mathcal{X}, \mathcal{I})$, consider the following operations:

- Deletion: For $B \subseteq \mathcal{X}$, we define $\mathcal{M} \setminus B = (\mathcal{X}', \mathcal{I}')$ with $\mathcal{X}' = \mathcal{X} \setminus B$,
$$\mathcal{I}' = \left\{S \subseteq \mathcal{X}' : S \in \mathcal{I}\right\}$$
  - Graphic: deleting edges from the graph
- Disjoint union: Given $M_1 = (\mathcal{X}_1, \mathcal{I}_2)$ and $M_2 = (\mathcal{X}_2, \mathcal{I}_2)$ with $\mathcal{X}_1 \bigcap \mathcal{X}_2 = \emptyset$, we define
$$M_1 \oplus M_2 = (\mathcal{X}_1 \bigcup \mathcal{X}_2, \left\{A_1 \bigcup A_2 : A_1 \in \mathcal{I}_1, A_2 \in \mathcal{I}_2\right\})$$
  - Graphic: combining two node-disjoint graphs

Operations preserving Matroidness

Given $\mathcal{M} = (\mathcal{X}, \mathcal{I})$, consider the following operations:

- Deletion: For $B \subseteq \mathcal{X}$, we define $\mathcal{M} \setminus B = (\mathcal{X}', \mathcal{I}')$ with $\mathcal{X}' = \mathcal{X} \setminus B$,
$$\mathcal{I}' = \left\{ S \subseteq \mathcal{X}' : S \in \mathcal{I} \right\}$$
  - Graphic: deleting edges from the graph
- Disjoint union: Given $M_1 = (\mathcal{X}_1, \mathcal{I}_2)$ and $M_2 = (\mathcal{X}_2, \mathcal{I}_2)$ with $\mathcal{X}_1 \bigcap \mathcal{X}_2 = \emptyset$, we define
$$M_1 \oplus M_2 = (\mathcal{X}_1 \bigcup \mathcal{X}_2, \left\{ A_1 \bigcup A_2 : A_1 \in \mathcal{I}_1, A_2 \in \mathcal{I}_2 \right\})$$
  - Graphic: combining two node-disjoint graphs
- Contraction: Given $B \subseteq \mathcal{X}$, let $M/B = (\mathcal{X}', \mathcal{I}')$ with $\mathcal{X}' = \mathcal{X} \setminus B$,
$$\mathcal{I}' = \left\{ S \subseteq X' : \mathcal{B} \bigcup S \in \mathcal{I} \right\}$$
  - i.e. Think of B as always being included
  - Graphic: contract the connected components of B

Operations preserving Matroidness

Given $\mathcal{M} = (\mathcal{X}, \mathcal{I})$, consider the following operations:

- Deletion: For $B \subseteq \mathcal{X}$, we define $\mathcal{M} \setminus B = (\mathcal{X}', \mathcal{I}')$ with $\mathcal{X}' = \mathcal{X} \setminus B$,
$$\mathcal{I}' = \{S \subseteq \mathcal{X}' : S \in \mathcal{I}\}$$
  - Graphic: deleting edges from the graph
- Disjoint union: Given $M_1 = (\mathcal{X}_1, \mathcal{I}_2)$ and $M_2 = (\mathcal{X}_2, \mathcal{I}_2)$ with $\mathcal{X}_1 \bigcap \mathcal{X}_2 = \emptyset$, we define
$$M_1 \oplus M_2 = (\mathcal{X}_1 \bigcup \mathcal{X}_2, \left\{A_1 \bigcup A_2 : A_1 \in \mathcal{I}_1, A_2 \in \mathcal{I}_2\right\})$$
  - Graphic: combining two node-disjoint graphs
- Contraction: Given $B \subseteq \mathcal{X}$, let $M/B = (\mathcal{X}', \mathcal{I}')$ with $\mathcal{X}' = \mathcal{X} \setminus B$,
$$\mathcal{I}' = \left\{S \subseteq X' : \mathcal{B} \bigcup S \in \mathcal{I}\right\}$$
  - i.e. Think of B as always being included
  - Graphic: contract the connected components of B
- Others: truncation, dual, union...

Matroids as an Algebra of Tractable Discrete Problems

- Optimization over matroids is “easy”, in the same way that optimization over convex sets is “easy”

Matroids as an Algebra of Tractable Discrete Problems

- Optimization over matroids is “easy”, in the same way that optimization over convex sets is “easy”
- Operations preserving set convexity are analogous to operations preserving matroid structure

Matroids as an Algebra of Tractable Discrete Problems

- Optimization over matroids is “easy”, in the same way that optimization over convex sets is “easy”
- Operations preserving set convexity are analogous to operations preserving matroid structure
- Arguably, matroids and submodular functions are discrete analogues of convex sets and convex functions, respectively.
    - Less exhaustive

Outline

Viewing Matroids Polyhedrally

- As is often the case with tractable discrete problems, we can view their feasible set as a polyhedron

Viewing Matroids Polyhedrally

- As is often the case with tractable discrete problems, we can view their feasible set as a polyhedron
- For $\mathcal{M} = (\mathcal{X}, \mathcal{I})$, the convex hull of independent sets can be written as a polytope in a natural way
    - The polytope is “solvable”, and admits a polytime separation oracle

Viewing Matroids Polyhedrally

- As is often the case with tractable discrete problems, we can view their feasible set as a polyhedron
- For $\mathcal{M} = (\mathcal{X}, \mathcal{I})$, the convex hull of independent sets can be written as a polytope in a natural way
  - The polytope is “solvable”, and admits a polytime separation oracle
- This perspective will be crucial for more advanced applications of matroids
  - Optimization of linear functions over matroid intersections
  - Optimization of submodular functions over matroids

The Matroid Polytope

Polytope $\mathcal{P}(\mathcal{M})$ for $\mathcal{M} = (\mathcal{X}, \mathcal{I})$

$$\sum_{i \in S} x_i \leq rank_{\mathcal{M}}(S), \quad \text{for } S \subseteq \mathcal{X}.$$
$$x_i \geq 0, \quad \text{for } i \in \mathcal{X}.$$

- Assigns a variable x_i to every element i of the ground set
- Each feasible x is a fractional subset of $\mathcal{X}$
  - $0 \leq x_i \leq 1$ since the rank of a singleton is at most 1.

The Matroid Polytope

Polytope $\mathcal{P}(\mathcal{M})$ for $\mathcal{M} = (\mathcal{X}, \mathcal{I})$

$$\begin{aligned} \sum_{i \in S} x_i \leq rank_{\mathcal{M}}(S), & \quad \text{for } S \subseteq \mathcal{X}. \\ x_i \geq 0, & \quad \text{for } i \in \mathcal{X}. \end{aligned}$$

- Assigns a variable x_i to every element i of the ground set
- Each feasible x is a fractional subset of $\mathcal{X}$
  - $0 \leq x_i \leq 1$ since the rank of a singleton is at most 1.
- The 0-1 indicator vector x_I for independent set $I \in \mathcal{I}$ is in the above polytope

The Matroid Polytope

Polytope $\mathcal{P}(\mathcal{M})$ for $\mathcal{M} = (\mathcal{X}, \mathcal{I})$

$$\sum_{i \in S} x_i \leq rank_{\mathcal{M}}(S), \quad \text{for } S \subseteq \mathcal{X}.$$
$$x_i \geq 0, \quad \text{for } i \in \mathcal{X}.$$

- Assigns a variable x_i to every element i of the ground set
- Each feasible x is a fractional subset of $\mathcal{X}$
  - $0 \leq x_i \leq 1$ since the rank of a singleton is at most 1.
- The 0-1 indicator vector x_I for independent set $I \in \mathcal{I}$ is in the above polytope
- In fact, we will show that $\mathcal{P}(\mathcal{M})$ is precisely the convex hull of independent sets $\mathcal{I}$

The Matroid Polytope

Polytope $\mathcal{P}(\mathcal{M})$ for $\mathcal{M} = (\mathcal{X}, \mathcal{I})$

$$\begin{aligned} \sum_{i \in S} x_i \leq rank_{\mathcal{M}}(S), & \quad \text{for } S \subseteq \mathcal{X}. \\ x_i \geq 0, & \quad \text{for } i \in \mathcal{X}. \end{aligned}$$

- Assigns a variable x_i to every element i of the ground set
- Each feasible x is a fractional subset of $\mathcal{X}$
  - $0 \leq x_i \leq 1$ since the rank of a singleton is at most 1.
- The 0-1 indicator vector x_I for independent set $I \in \mathcal{I}$ is in the above polytope
- In fact, we will show that $\mathcal{P}(\mathcal{M})$ is precisely the convex hull of independent sets $\mathcal{I}$
- Note: polytope has $2^{|\mathcal{X}|}$ constraints.

Integrality of the Matroid Polytope

Polytope $\mathcal{P}(\mathcal{M})$ for $\mathcal{M} = (\mathcal{X}, \mathcal{I})$

$$\sum_{i \in S} x_i \leq rank_{\mathcal{M}}(S), \quad \text{for } S \subseteq \mathcal{X}.$$
$$x_i \geq 0, \quad \text{for } i \in \mathcal{X}.$$

Theorem

$\mathcal{P}(\mathcal{M}) = convexhull\,\{x_I : I \in \mathcal{I}\}$

Integrality of the Matroid Polytope

Polytope $\mathcal{P}(\mathcal{M})$ for $\mathcal{M} = (\mathcal{X}, \mathcal{I})$

$$\sum_{i \in S} x_i \leq rank_{\mathcal{M}}(S), \quad \text{for } S \subseteq \mathcal{X}.$$
$$x_i \geq 0, \quad \text{for } i \in \mathcal{X}.$$

Theorem

$\mathcal{P}(\mathcal{M}) = convexhull\{x_I : I \in \mathcal{I}\}$

- It is clear that $\mathcal{P}(\mathcal{M}) \supseteq convexhull\{x_I : I \in \mathcal{I}\}$

Integrality of the Matroid Polytope

Polytope $\mathcal{P}(\mathcal{M})$ for $\mathcal{M} = (\mathcal{X}, \mathcal{I})$

$$\begin{aligned} \sum_{i \in S} x_i &\leq rank_{\mathcal{M}}(S), && \text{for } S \subseteq \mathcal{X}. \\ x_i &\geq 0, && \text{for } i \in \mathcal{X}. \end{aligned}$$

Theorem

$\mathcal{P}(\mathcal{M}) = convexhull \{x_I : I \in \mathcal{I}\}$

- It is clear that $\mathcal{P}(\mathcal{M}) \supseteq convexhull \{x_I : I \in \mathcal{I}\}$
- To show that $\mathcal{P}(\mathcal{M}) \subseteq convexhull \{x_I : I \in \mathcal{I}\}$, we will show that every vertex of $\mathcal{P}(\mathcal{M})$ equals x_I for some $I \in \mathcal{I}$.

Integrality of the Matroid Polytope

Polytope $\mathcal{P}(\mathcal{M})$ for $\mathcal{M} = (\mathcal{X}, \mathcal{I})$

$$\begin{aligned} \sum_{i \in S} x_i \leq rank_{\mathcal{M}}(S), & \quad \text{for } S \subseteq \mathcal{X}. \\ x_i \geq 0, & \quad \text{for } i \in \mathcal{X}. \end{aligned}$$

Theorem

$\mathcal{P}(\mathcal{M}) = convexhull \{x_I : I \in \mathcal{I}\}$

- It is clear that $\mathcal{P}(\mathcal{M}) \supseteq convexhull \{x_I : I \in \mathcal{I}\}$
- To show that $\mathcal{P}(\mathcal{M}) \subseteq convexhull \{x_I : I \in \mathcal{I}\}$, we will show that every vertex of $\mathcal{P}(\mathcal{M})$ equals x_I for some $I \in \mathcal{I}$.
- Recall: suffices to show that every linear function $w^T x$ is maximized over $\mathcal{P}(\mathcal{M})$ at some x_I for $I \in \mathcal{I}$.

Recall:The Greedy Algorithm

1. $B \leftarrow \emptyset$
2. Sort nonnegative elements of $\mathcal{X}$ in decreasing order of weight
   - $\{1, \ldots, n\}$ with $w_1 \geq w_2, \geq \ldots \geq w_n \geq 0$.
3. For $i = 1$ to n:
   - if $B \bigcup \{i\} \in \mathcal{I}$, add i to B.

Theorem

The greedy algorithm returns the maximum weight set for every choice of weights if and only if the set system $(\mathcal{X}, \mathcal{I})$ is a matroid.

Recall:The Greedy Algorithm

1. $B \leftarrow \emptyset$
2. Sort nonnegative elements of $\mathcal{X}$ in decreasing order of weight
   - $\{1, \ldots, n\}$ with $w_1 \geq w_2, \geq \ldots \geq w_n \geq 0$.
3. For $i = 1$ to n:
   - if $B \bigcup \{i\} \in \mathcal{I}$, add i to B.

Theorem

The greedy algorithm returns the maximum weight set for every choice of weights if and only if the set system $(\mathcal{X}, \mathcal{I})$ is a matroid.

- We can think of the greedy algorithm as computing the indicator vector $x^* = x_B \in \mathcal{P}(\mathcal{M})$
- We will show that x^* maximizes $w^\intercal x$ over $x \in \mathcal{P}(\mathcal{M})$.

Recall: Observation

$i \in \{1, \ldots, n\}$ selected by greedy algorithm iff $i \notin span(\{1, \ldots, i-1\})$

- i.e. if $rank[1:i] - rank[1:i-1] = 1$.

Recall: Observation

$i \in \{1, \ldots, n\}$ selected by greedy algorithm iff $i \notin span(\{1, \ldots, i-1\})$

- i.e. if $rank[1:i] - rank[1:i-1] = 1$.

- Therefore, $x_i^* = rank[1:i] - rank[1:i-1]$ for nonneg-weight i, and $x_i^* = 0$ for negative-weight i.

$$\sum_{i \in \mathcal{X}} w_i x_i^* = \sum_{i=1}^{n} w_i x_i^* = \sum_{i=1}^{n} w_i (rank[1:i] - rank[1:i-1])$$

Recall: Observation

$i \in \{1, \ldots, n\}$ selected by greedy algorithm iff $i \notin span(\{1, \ldots, i-1\})$

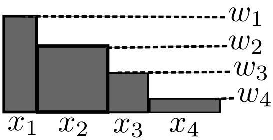
- i.e. if $rank[1:i] - rank[1:i-1] = 1$.

- Therefore, $x_i^* = rank[1:i] - rank[1:i-1]$ for nonneg-weight i, and $x_i^* = 0$ for negative-weight i.

$$\sum_{i \in \mathcal{X}} w_i x_i^* = \sum_{i=1}^{n} w_i x_i^* = \sum_{i=1}^{n} w_i (rank[1:i] - rank[1:i-1])$$

- Consider an arbitrary $x \in \mathcal{P}(\mathcal{M})$

$$\sum_{i \in \mathcal{X}} w_i x_i \leq \sum_{i=1}^{n} w_i x_i =$$



Recall: Observation

$i \in \{1, \ldots, n\}$ selected by greedy algorithm iff $i \notin span(\{1, \ldots, i-1\})$

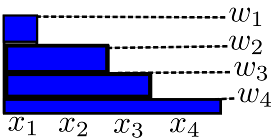
- i.e. if $rank[1:i] - rank[1:i-1] = 1$.

- Therefore, $x_i^* = rank[1:i] - rank[1:i-1]$ for nonneg-weight i, and $x_i^* = 0$ for negative-weight i.

$$\sum_{i \in \mathcal{X}} w_i x_i^* = \sum_{i=1}^{n} w_i x_i^* = \sum_{i=1}^{n} w_i (rank[1:i] - rank[1:i-1])$$

- Consider an arbitrary $x \in \mathcal{P}(\mathcal{M})$

$$\sum_{i \in \mathcal{X}} w_i x_i \leq \sum_{i=1}^{n} w_i x_i = \sum_{i=1}^{n} (w_i - w_{i+1}) x(1:i)$$



Recall: Observation

$i \in \{1, \ldots, n\}$ selected by greedy algorithm iff $i \notin span(\{1, \ldots, i-1\})$

- i.e. if $rank[1:i] - rank[1:i-1] = 1$.

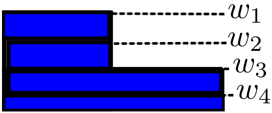
- Therefore, $x_i^* = rank[1:i] - rank[1:i-1]$ for nonneg-weight i, and $x_i^* = 0$ for negative-weight i.

$$\sum_{i \in \mathcal{X}} w_i x_i^* = \sum_{i=1}^{n} w_i x_i^* = \sum_{i=1}^{n} w_i (rank[1:i] - rank[1:i-1])$$

- Consider an arbitrary $x \in \mathcal{P}(\mathcal{M})$

$$\sum_{i \in \mathcal{X}} w_i x_i \leq \sum_{i=1}^{n} w_i x_i = \sum_{i=1}^{n} (w_i - w_{i+1}) x(1:i)$$

$$\leq \sum_{i=1}^{n} (w_i - w_{i+1}) rank(1:i)$$



Recall: Observation

$i \in \{1, \dots, n\}$ selected by greedy algorithm iff $i \notin span(\{1, \dots, i-1\})$

- i.e. if $rank[1:i] - rank[1:i-1] = 1$.

- Therefore, $x_i^* = rank[1:i] - rank[1:i-1]$ for nonneg-weight i, and $x_i^* = 0$ for negative-weight i.

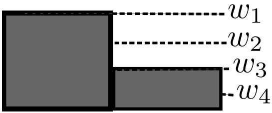
$$\sum_{i \in \mathcal{X}} w_i x_i^* = \sum_{i=1}^{n} w_i x_i^* = \sum_{i=1}^{n} w_i (rank[1:i] - rank[1:i-1])$$

- Consider an arbitrary $x \in \mathcal{P}(\mathcal{M})$

$$\sum_{i \in \mathcal{X}} w_i x_i \le \sum_{i=1}^{n} w_i x_i = \sum_{i=1}^{n} (w_i - w_{i+1}) x(1:i)$$

$$\le \sum_{i=1}^{n} (w_i - w_{i+1}) rank(1:i)$$

$$= \sum_{i=1}^{n} w_i (rank[1:i] - rank[1:i-1])$$



The Matroid Base Polytope

- The matroid polytope is the convex hull of independent sets
  - Graphic: convex hull of forests
- What if we wish to consider only “full-rank” sets?
  - Graphic: spanning trees

The Matroid Base Polytope

- The matroid polytope is the convex hull of independent sets
  - Graphic: convex hull of forests
- What if we wish to consider only "full-rank" sets?
  - Graphic: spanning trees

Polytope $\mathcal{P}_{base}(\mathcal{M})$ for $\mathcal{M} = (\mathcal{X}, \mathcal{I})$

$$\begin{aligned} \sum_{i \in S} x_i &\leq rank_{\mathcal{M}}(S), && \text{for } S \subseteq \mathcal{X}. \\ \sum_{i \in \mathcal{X}} x_i &= rank(\mathcal{M}) && \\ x_i &\geq 0, && \text{for } i \in \mathcal{X}. \end{aligned}$$

- The 0-1 indicator vector x_B for a base B of $\mathcal{M}$ is in above polytope

The Matroid Base Polytope

- The matroid polytope is the convex hull of independent sets
  - Graphic: convex hull of forests
- What if we wish to consider only “full-rank” sets?
  - Graphic: spanning trees

Polytope $\mathcal{P}_{base}(\mathcal{M})$ for $\mathcal{M} = (\mathcal{X}, \mathcal{I})$

$$\begin{aligned} \sum_{i \in S} x_i &\leq rank_{\mathcal{M}}(S), && \text{for } S \subseteq \mathcal{X}. \\ \sum_{i \in \mathcal{X}} x_i &= rank(\mathcal{M}) && \\ x_i &\geq 0, && \text{for } i \in \mathcal{X}. \end{aligned}$$

- The 0-1 indicator vector x_B for a base B of $\mathcal{M}$ is in above polytope
- In fact, we will show that $\mathcal{P}_{base}(\mathcal{M})$ is precisely the convex hull of bases of $\mathcal{M}$

Polytope $\mathcal{P}_{base}(\mathcal{M})$ for $\mathcal{M} = (\mathcal{X}, \mathcal{I})$

$$\sum_{i \in S} x_i \le rank_{\mathcal{M}}(S), \quad \text{for } S \subseteq \mathcal{X}.$$
$$\sum_{i \in \mathcal{X}} x_i = rank(\mathcal{M})$$
$$x_i \ge 0, \quad \text{for } i \in \mathcal{X}.$$

Theorem

$\mathcal{P}_{base}(\mathcal{M}) = convexhull\,\{x_B : B$ *is a base of* $\mathcal{M}\}$

Polytope $\mathcal{P}_{base}(\mathcal{M})$ for $\mathcal{M} = (\mathcal{X}, \mathcal{I})$

$$\begin{aligned} \sum_{i \in S} x_i &\leq rank_{\mathcal{M}}(S), && \text{for } S \subseteq \mathcal{X}. \\ \sum_{i \in \mathcal{X}} x_i &= rank(\mathcal{M}) && \\ x_i &\geq 0, && \text{for } i \in \mathcal{X}. \end{aligned}$$

Theorem

$\mathcal{P}_{base}(\mathcal{M}) = convexhull \{x_B : B$ *is a base of* $\mathcal{M}\}$

- As before, one direction is obvious:
  $\mathcal{P}_{base}(\mathcal{M}) \supseteq convexhull \{x_B : B$ is a base of $\mathcal{M}\}$

Polytope $\mathcal{P}_{base}(\mathcal{M})$ for $\mathcal{M} = (\mathcal{X}, \mathcal{I})$

$$\sum_{i \in S} x_i \leq rank_{\mathcal{M}}(S), \quad \text{for } S \subseteq \mathcal{X}.$$
$$\sum_{i \in \mathcal{X}} x_i = rank(\mathcal{M})$$
$$x_i \geq 0, \quad \text{for } i \in \mathcal{X}.$$

Theorem

$\mathcal{P}_{base}(\mathcal{M}) = convexhull\,\{x_B : B$ *is a base of* $\mathcal{M}\}$

- As before, one direction is obvious: $\mathcal{P}_{base}(\mathcal{M}) \supseteq convexhull\,\{x_B : B \text{ is a base of } \mathcal{M}\}$
- For the other direction, take $x \in \mathcal{P}_{base}(\mathcal{M})$

Polytope $\mathcal{P}_{base}(\mathcal{M})$ for $\mathcal{M} = (\mathcal{X}, \mathcal{I})$

$$\sum_{i \in S} x_i \leq rank_{\mathcal{M}}(S), \quad \text{for } S \subseteq \mathcal{X}.$$
$$\sum_{i \in \mathcal{X}} x_i = rank(\mathcal{M})$$
$$x_i \geq 0, \quad \text{for } i \in \mathcal{X}.$$

Theorem

$\mathcal{P}_{base}(\mathcal{M}) = convexhull\,\{x_B : B$ *is a base of* $\mathcal{M}\}$

- As before, one direction is obvious: $\mathcal{P}_{base}(\mathcal{M}) \supseteq convexhull\,\{x_B : B \text{ is a base of } \mathcal{M}\}$
- For the other direction, take $x \in \mathcal{P}_{base}(\mathcal{M})$
- Since $x \in \mathcal{P}(\mathcal{M})$, x is a convex combination of independent sets $I_1, \ldots, I_k$ of $\mathcal{M}$.

Polytope $\mathcal{P}_{base}(\mathcal{M})$ for $\mathcal{M} = (\mathcal{X}, \mathcal{I})$

$$\sum_{i \in S} x_i \leq rank_{\mathcal{M}}(S), \quad \text{for } S \subseteq \mathcal{X}.$$
$$\sum_{i \in \mathcal{X}} x_i = rank(\mathcal{M})$$
$$x_i \geq 0, \quad \text{for } i \in \mathcal{X}.$$

Theorem

$\mathcal{P}_{base}(\mathcal{M}) = convexhull \{x_B : B$ *is a base of* $\mathcal{M}\}$

- As before, one direction is obvious: $\mathcal{P}_{base}(\mathcal{M}) \supseteq convexhull \{x_B : B$ is a base of $\mathcal{M}\}$
- For the other direction, take $x \in \mathcal{P}_{base}(\mathcal{M})$
- Since $x \in \mathcal{P}(\mathcal{M})$, x is a convex combination of independent sets $I_1, \ldots, I_k$ of $\mathcal{M}$.
- Since $||x||_1 = rank(\mathcal{M})$, and $||x_{I_\ell}||_1 \leq rank(\mathcal{M})$ for all ℓ, it must be that $||x_{I_1}||_1 = ||x_{I_2}||_1 = \ldots = ||x_{I_k}||_1 = rank(\mathcal{M})$

Solvability of Matroid Polytopes

Polytope $\mathcal{P}(\mathcal{M})$ for $\mathcal{M} = (\mathcal{X}, \mathcal{I})$

$$\begin{aligned} \sum_{i \in S} x_i \leq rank_{\mathcal{M}}(S), & \quad \text{for } S \subseteq \mathcal{X}. \\ x_i \geq 0, & \quad \text{for } i \in \mathcal{X}. \end{aligned}$$

- When given an independence oracle for $\mathcal{M}$, we can maximize linear functions over $\mathcal{P}(\mathcal{M})$ in $O(n \log n + nT)$ time, where T is runtime of independence oracle
  - By integrality, same as finding max-weight independent set of $\mathcal{M}$.

Solvability of Matroid Polytopes

Polytope $\mathcal{P}(\mathcal{M})$ for $\mathcal{M} = (\mathcal{X}, \mathcal{I})$

$$\begin{aligned} \sum_{i \in S} x_i \leq rank_{\mathcal{M}}(S), & \quad \text{for } S \subseteq \mathcal{X}. \\ x_i \geq 0, & \quad \text{for } i \in \mathcal{X}. \end{aligned}$$

- When given an independence oracle for $\mathcal{M}$, we can maximize linear functions over $\mathcal{P}(\mathcal{M})$ in $O(n \log n + nT)$ time, where T is runtime of independence oracle
  - By integrality, same as finding max-weight independent set of $\mathcal{M}$.
- Therefore, by equivalence of separation and optimization, can also implement a separation oracle for $\mathcal{P}(\mathcal{M})$ in $\mathrm{poly}(n, T)$ time.

Solvability of Matroid Polytopes

Polytope $\mathcal{P}(\mathcal{M})$ for $\mathcal{M} = (\mathcal{X}, \mathcal{I})$

$$\begin{aligned} \sum_{i \in S} x_i &\leq rank_{\mathcal{M}}(S), && \text{for } S \subseteq \mathcal{X}. \\ x_i &\geq 0, && \text{for } i \in \mathcal{X}. \end{aligned}$$

- When given an independence oracle for $\mathcal{M}$, we can maximize linear functions over $\mathcal{P}(\mathcal{M})$ in $O(n \log n + nT)$ time, where T is runtime of independence oracle
  - By integrality, same as finding max-weight independent set of $\mathcal{M}$.
- Therefore, by equivalence of separation and optimization, can also implement a separation oracle for $\mathcal{P}(\mathcal{M})$ in $\mathrm{poly}(n, T)$ time.
- A more direct proof: reduces to submodular function minimization
  - $rank_{\mathcal{M}}$ is a submodular set function.

Outline

Matroid Intersection

- Optimization of linear functions over matroids is tractable
- Matroid operations provide an algebra for constructing new matroids from old
- We will look at one operation on matroids which does not produce a matroid, but nevertheless produces a solvable problem.

Matroid Intersection

- Optimization of linear functions over matroids is tractable
- Matroid operations provide an algebra for constructing new matroids from old
- We will look at one operation on matroids which does not produce a matroid, but nevertheless produces a solvable problem.

Matroid Intersection

Given matroids $\mathcal{M}_1 = (\mathcal{X}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_2)$ on the same ground set, we define the set system $\mathcal{M}_1 \bigcap \mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_1 \bigcap \mathcal{I}_2)$.

Matroid Intersection

- Optimization of linear functions over matroids is tractable
- Matroid operations provide an algebra for constructing new matroids from old
- We will look at one operation on matroids which does not produce a matroid, but nevertheless produces a solvable problem.

Matroid Intersection

Given matroids $\mathcal{M}_1 = (\mathcal{X}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_2)$ on the same ground set, we define the set system $\mathcal{M}_1 \bigcap \mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_1 \bigcap \mathcal{I}_2)$.

- i.e. a set is feasible if it is independent in both matroids

Matroid Intersection

- Optimization of linear functions over matroids is tractable
- Matroid operations provide an algebra for constructing new matroids from old
- We will look at one operation on matroids which does not produce a matroid, but nevertheless produces a solvable problem.

Matroid Intersection

Given matroids $\mathcal{M}_1 = (\mathcal{X}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_2)$ on the same ground set, we define the set system $\mathcal{M}_1 \bigcap \mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_1 \bigcap \mathcal{I}_2)$.

- i.e. a set is feasible if it is independent in both matroids
- In general, does not produce a matroid

Matroid Intersection

- Optimization of linear functions over matroids is tractable
- Matroid operations provide an algebra for constructing new matroids from old
- We will look at one operation on matroids which does not produce a matroid, but nevertheless produces a solvable problem.

Matroid Intersection

Given matroids $\mathcal{M}_1 = (\mathcal{X}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_2)$ on the same ground set, we define the set system $\mathcal{M}_1 \bigcap \mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_1 \bigcap \mathcal{I}_2)$.

- i.e. a set is feasible if it is independent in both matroids
- In general, does not produce a matroid
- Nevertheless, it will turn out that maximizing linear functions over a matroid intersection is tractable

Matroid Intersection

- Optimization of linear functions over matroids is tractable
- Matroid operations provide an algebra for constructing new matroids from old
- We will look at one operation on matroids which does not produce a matroid, but nevertheless produces a solvable problem.

Matroid Intersection

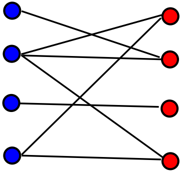
Given matroids $\mathcal{M}_1 = (\mathcal{X}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_2)$ on the same ground set, we define the set system $\mathcal{M}_1 \bigcap \mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_1 \bigcap \mathcal{I}_2)$.

- i.e. a set is feasible if it is independent in both matroids
- In general, does not produce a matroid
- Nevertheless, it will turn out that maximizing linear functions over a matroid intersection is tractable
- However, maximizing linear functions over the intersection of 3 or more matroids is NP-hard

Examples

Bipartite Matching

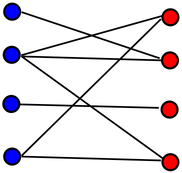
Given a bipartite graph G, a set of edges F is a bipartite matching if and only if each node is incident on at most one edge in F.



Examples

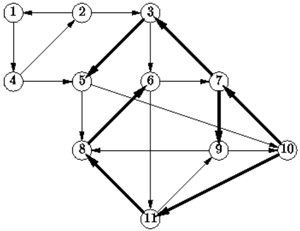
Bipartite Matching

Given a bipartite graph G, a set of edges F is a bipartite matching if and only if each node is incident on at most one edge in F.



Arborescence

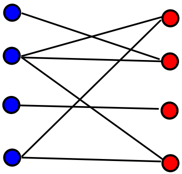
Given a directed graph G, a set of edges is an r-arborescence if it is a tree directed away from the root r.



Examples

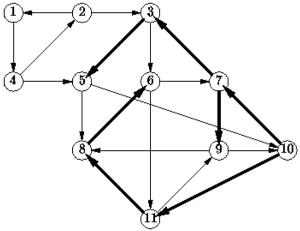
Bipartite Matching

Given a bipartite graph G, a set of edges F is a bipartite matching if and only if each node is incident on at most one edge in F.



Arborescence

Given a directed graph G, a set of edges is an r-arborescence if it is a tree directed away from the root r.



- Others: colorful spanning trees, orientations, . . .

The Matroid Intersection Polytope

Matroid Intersection

Given matroids $\mathcal{M}_1 = (\mathcal{X}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_2)$ on the same ground set, we define the set system $\mathcal{M}_1 \bigcap \mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_1 \bigcap \mathcal{I}_2)$.

- Optimizing a modular function over $\mathcal{M}_1 \bigcap \mathcal{M}_2$ is equivalent to optimizing a linear function over $convexhull\,\{x_I : I \in \mathcal{I}_1 \bigcap \mathcal{I}_2\}$.
- As it turns out, this is a solvable polytope.

Theorem

$\mathcal{P}(M_1) \bigcap \mathcal{P}(M_2) = convexhull\,\{x_I : I \in \mathcal{I}_1 \bigcap \mathcal{I}_2\}$

The Matroid Intersection Polytope

Matroid Intersection

Given matroids $\mathcal{M}_1 = (\mathcal{X}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_2)$ on the same ground set, we define the set system $\mathcal{M}_1 \bigcap \mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_1 \bigcap \mathcal{I}_2)$.

- Optimizing a modular function over $\mathcal{M}_1 \bigcap \mathcal{M}_2$ is equivalent to optimizing a linear function over $convexhull\, \{x_I : I \in \mathcal{I}_1 \bigcap \mathcal{I}_2\}$.
- As it turns out, this is a solvable polytope.

Theorem

$\mathcal{P}(M_1) \bigcap \mathcal{P}(M_2) = convexhull\, \{x_I : I \in \mathcal{I}_1 \bigcap \mathcal{I}_2\}$

- One direction is obvious: $\mathcal{P}(M_1) \bigcap \mathcal{P}(M_2) \supseteq convexhull\, \{x_I : I \in \mathcal{I}_1 \bigcap \mathcal{I}_2\}$

The Matroid Intersection Polytope

Matroid Intersection

Given matroids $\mathcal{M}_1 = (\mathcal{X}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_2)$ on the same ground set, we define the set system $\mathcal{M}_1 \bigcap \mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_1 \bigcap \mathcal{I}_2)$.

- Optimizing a modular function over $\mathcal{M}_1 \bigcap \mathcal{M}_2$ is equivalent to optimizing a linear function over $convexhull\,\{x_I : I \in \mathcal{I}_1 \bigcap \mathcal{I}_2\}$.
- As it turns out, this is a solvable polytope.

Theorem

$\mathcal{P}(M_1) \bigcap \mathcal{P}(M_2) = convexhull\,\{x_I : I \in \mathcal{I}_1 \bigcap \mathcal{I}_2\}$

- One direction is obvious: $\mathcal{P}(M_1) \bigcap \mathcal{P}(M_2) \supseteq convexhull\,\{x_I : I \in \mathcal{I}_1 \bigcap \mathcal{I}_2\}$
- The other direction is not so obvious

The Matroid Intersection Polytope

Matroid Intersection

Given matroids $\mathcal{M}_1 = (\mathcal{X}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_2)$ on the same ground set, we define the set system $\mathcal{M}_1 \bigcap \mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_1 \bigcap \mathcal{I}_2)$.

- Optimizing a modular function over $\mathcal{M}_1 \bigcap \mathcal{M}_2$ is equivalent to optimizing a linear function over $convexhull\left\{x_I : I \in \mathcal{I}_1 \bigcap \mathcal{I}_2\right\}$.
- As it turns out, this is a solvable polytope.

Theorem

$\mathcal{P}(M_1) \bigcap \mathcal{P}(M_2) = convexhull\left\{x_I : I \in \mathcal{I}_1 \bigcap \mathcal{I}_2\right\}$

- One direction is obvious:
  $\mathcal{P}(M_1) \bigcap \mathcal{P}(M_2) \supseteq convexhull\left\{x_I : I \in \mathcal{I}_1 \bigcap \mathcal{I}_2\right\}$
- The other direction is not so obvious
  - It is conceivable that $\mathcal{P}(\mathcal{M}_1) \bigcap \mathcal{P}(\mathcal{M}_2)$ has fractional vertices

The Matroid Intersection Polytope

Matroid Intersection

Given matroids $\mathcal{M}_1 = (\mathcal{X}, \mathcal{I}_1)$ and $\mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_2)$ on the same ground set, we define the set system $\mathcal{M}_1 \bigcap \mathcal{M}_2 = (\mathcal{X}, \mathcal{I}_1 \bigcap \mathcal{I}_2)$.

- Optimizing a modular function over $\mathcal{M}_1 \bigcap \mathcal{M}_2$ is equivalent to optimizing a linear function over $convexhull \{x_I : I \in \mathcal{I}_1 \bigcap \mathcal{I}_2\}$.
- As it turns out, this is a solvable polytope.

Theorem

$\mathcal{P}(M_1) \bigcap \mathcal{P}(M_2) = convexhull \{x_I : I \in \mathcal{I}_1 \bigcap \mathcal{I}_2\}$

- One direction is obvious:
  $\mathcal{P}(M_1) \bigcap \mathcal{P}(M_2) \supseteq convexhull \{x_I : I \in \mathcal{I}_1 \bigcap \mathcal{I}_2\}$
- The other direction is not so obvious
  - It is conceivable that $\mathcal{P}(\mathcal{M}_1) \bigcap \mathcal{P}(\mathcal{M}_2)$ has fractional vertices
- Nevertheless, it is true but hard to prove, so we will skip it.

Optimization over Matroid Intersections

Optimization over Matroid Intersection $\mathcal{M}_1 \bigcap \mathcal{M}_2$

$$
\begin{array}{lll}
\text{maximize} & \sum_{i \in \mathcal{X}} w_i x_i & \\
\text{subject to} & & \\
& \displaystyle\sum_{i \in S} x_i \leq rank_{\mathcal{M}_1}(S), & \text{for } S \subseteq \mathcal{X}. \\
& \displaystyle\sum_{i \in S} x_i \leq rank_{\mathcal{M}_2}(S), & \text{for } S \subseteq \mathcal{X}. \\
& x_i \geq 0, & \text{for } i \in \mathcal{X}.
\end{array}
$$

Optimization over Matroid Intersections

Optimization over Matroid Intersection $\mathcal{M}_1 \bigcap \mathcal{M}_2$

maximize $\sum_{i \in \mathcal{X}} w_i x_i$

subject to

$$\begin{aligned} \sum_{i \in S} x_i \leq rank_{\mathcal{M}_1}(S), & \quad \text{for } S \subseteq \mathcal{X}. \\ \sum_{i \in S} x_i \leq rank_{\mathcal{M}_2}(S), & \quad \text{for } S \subseteq \mathcal{X}. \\ x_i \geq 0, & \quad \text{for } i \in \mathcal{X}. \end{aligned}$$

Theorem

Given independence oracles to both matroids $\mathcal{M}_1$ and $\mathcal{M}_2$, there is an algorithm for finding the maximum weight set in $\mathcal{M}_1 \bigcap \mathcal{M}_2$ which runs in $\mathrm{poly}(n)$ time.

Optimization over Matroid Intersections

Optimization over Matroid Intersection $\mathcal{M}_1 \bigcap \mathcal{M}_2$

$$
\begin{array}{ll}
\text{maximize} & \sum_{i \in \mathcal{X}} w_i x_i \\
\text{subject to} & \\
& \displaystyle\sum_{i \in S} x_i \leq rank_{\mathcal{M}_1}(S), \quad \text{for } S \subseteq \mathcal{X}. \\
& \displaystyle\sum_{i \in S} x_i \leq rank_{\mathcal{M}_2}(S), \quad \text{for } S \subseteq \mathcal{X}. \\
& x_i \geq 0, \qquad \text{for } i \in \mathcal{X}.
\end{array}
$$

Theorem

Given independence oracles to both matroids $\mathcal{M}_1$ and $\mathcal{M}_2$, there is an algorithm for finding the maximum weight set in $\mathcal{M}_1 \bigcap \mathcal{M}_2$ which runs in $\mathrm{poly}(n)$ time.

Proof: Using equivalence of separation and optimization, and the fact that all coefficients in the LP have $\mathrm{poly}(n)$ bits.

NP-hardness of 3-way Matroid Intersection

By a reduction from Hamiltonian Path in directed graphs