Matroids and the Greedy Algorithm, Matroid Polytope

Presented By-
Jaya Bharti (23D0374)
Himanshi Singh (23D0363)
Greedy Algorithms

- Greedy algorithms are by far one of the easiest and most well-understood algorithmic techniques.
- A greedy algorithm solves an optimization problem by working in several phases.
- In each phase, a decision is made that is locally optimal given the information that has been obtained so far. This decision is made without regard for future consequences.
- This greedy “pick what looks best” strategy is explains the name for this class of algorithms.
- So a greedy routing algorithm would say to a routing problem: “You want to visit all these locations with minimum travel time? Let’s start by going to the closest one. And from there to the next closest one. And so on.”
Correct Greedy Algorithms

• When a greedy algorithm terminates, then the hope is that the greedy choices in each phase lead to a global optimum of the optimization problem. If a global optimum is always reached, then the algorithm is correct.

• *Can we characterize when greedy algorithms give an optimal solution to a problem?*

• The answer is yes, and the framework that enables us to do this is called a matroid. That is, *if we can phrase the problem we’re trying to solve as a matroid, then the greedy algorithm is guaranteed to be optimal*. Let’s start with an example when greedy is provably optimal: the minimum spanning tree problem.
Minimum Spanning Trees

- Given a graph $G = (V, E)$ and edge weights $w_e \geq 0$, our goal is to connect all vertices by a subset of edges $F$ while minimizing its cost $\sum_{e \in F} w_e$.
- Without loss of generality the optimal solution is a tree which is called the Minimum Spanning Tree (MST).
- This is perhaps the oldest combinatorial optimization problem; it was first solved by Boruvka in 1926 and Jarník in 1930.
- Both of the proposed algorithms were variants of the greedy algorithm.
Minimum Spanning Trees

**Algorithm 1** Borůvka’s Algorithm

\[ F \leftarrow \emptyset \]

while \( F \) is disconnected do
  for all components \( C_i \) do
    \[ F \leftarrow F \cup \{ e_i \} \text{ for } e_i = \text{ the min-weight edge leaving } C_i. \]
  end for
end while

**Algorithm 2** Jarník’s Algorithm

\[ T \leftarrow \emptyset \]

while \( T \) is not a spanning tree do
  \[ T \leftarrow T \cup \{ e \} \text{ for } e = \text{ the min-weight edge extending the tree } T \text{ to a new vertex.} \]
end while
Minimum Spanning Trees

In the 1950’s the problem was studied again and algorithms were proposed by Prim and Kruskal.

Algorithm 3 Kruskal’s Algorithm

\[
S \leftarrow E \\
F \leftarrow \emptyset \\
\text{while } S \neq \emptyset \text{ and } F \text{ is not spanning do} \\
\quad \text{Remove the min-weight edge } e \text{ from } S. \\
\quad \text{if } F \cup \{e\} \text{ does not create a cycle then} \\
\quad \quad F \leftarrow F \cup \{e\} \\
\quad \text{else} \\
\quad \quad \text{Discard } e \\
\quad \text{end if} \\
\text{end while}
\]

All of these algorithms work because spanning trees form a “matroid”
Matroids
History

- Matroids were first introduced by Hassler Whitney in 1935.
- Almost immediately after Whitney first wrote about matroids, an important article was written by Saunders Mac Lane (1936) on the relation of matroids to projective geometry. A year later, B. L. van der Waerden (1937) noted similarities between algebraic and linear dependence in his classic textbook on Modern Algebra.
- They were interested in devising a general description of “independence,” the properties of which are strikingly similar when specified in linear algebra and graph theory.
- Since then the study of matroids has blossomed into a large and beautiful theory, one part of which is the characterization of the greedy algorithm: greedy is optimal on a problem if and only if the problem can be represented as a matroid.
The Matroids

The theory of matroids is a rich mathematical theory, which is tightly connected to the greedy algorithm. Many, but not all, applications of the greedy algorithm involve a matroid.

Matroids are abstract combinatorial structures that generalize the notion of spanning trees (or spanning forests).

Definition: A matroid $M$ is a pair $(E, I)$, where $E$ is a finite set and $I$ a family of subsets of $E$, that satisfies the following two properties

1. Hereditary property (down-closed family): $\forall P \in I, Q \subset P \Rightarrow P \in I$.

2. Exchange property (extension axiom): $\forall P, Q \in I, |P| < |Q| \Rightarrow \exists x \in Q \setminus P$ such that $P \cup \{x\} \in I$.

The set $E$ is called the ground set and the elements of $I$ are called independent sets.
Notation and Terminology

- $I + j \equiv I \cup \{j\}$
- $I - j \equiv I \setminus \{j\}$
- $B$ is a base in $M \iff B \in I$ and $B$ cannot be extended to a larger independent set. More generally, $B$ is a base of $S \subseteq E$, if $B \in I$ and $\nexists x \in S \setminus B$ such that $B + x \in I$. 
Lemma: M = (E, I) is a matroid $\iff$ I is down-closed and $\forall S \subseteq E$ all bases of S have the same size.

Proof:

1. If M is a matroid and $B_1, B_2$ are bases of $S \subseteq E$, then if $|B_1| < |B_2|$ $\Rightarrow B_1$ can be extended by $x \in B_2 \setminus B_1$, hence $B_1$ was not a base.

2. Assume that $\forall S \subseteq E$ all bases have the same size. Then for $P, Q \subseteq E$ where $|P| < |Q|$ let $S = P \cup Q$. Then P is not a base of S, therefore $\exists j \in S \setminus P = Q \setminus P$ such that $P + j \in I$. 
Graphic Matroids

So why are spanning trees an example of a matroid?

**Graphic Matroids:** For a graph $G = (V, E)$ a forest is any set of edges $F \subseteq E$ that does not contain any cycles.

Lemma: $M = (E, F)$ where $F = \{F \subseteq E : F$ is a forest$\}$ is a matroid.

**Proof:**

- $F$ is clearly down-closed. We prove that for any $S \subseteq E$, all bases have the same size.
- Take any $S \subseteq E$ and look at the connected components of $(V, S)$. What are the bases of $S$, i.e. maximal subsets $F \subseteq S$ which are forests?
- In each connected component $C_i$ we have $|F \cap E[C_i]| = |C_i| - 1$, because if we have less than $|C_i| - 1$ edges, then $F$ does not contain a spanning tree on $C_i$ and we can add some edge without creating a cycle.
- Conversely, we cannot have more than $|C_i| - 1$ edges on $C_i$ because we would create a cycle. This implies that $|F| = \Sigma |C_i| - 1 = |V| - \#components(S)$, the same number for each base $F$. 

Weighted matroids

Given a matroid \((E, I)\), we can define a weighted matroid by associating a positive weight \(w(x)\) to each element \(x\) of the ground set \(E\).

The weighted matroid problem has

Input: A weighted matroid

Output: An independent set of maximum total weight
The greedy algorithm for matroids

The following algorithm finds the maximum weight base in a matroid $M = (E, I)$.

**Algorithm 1** Greedy algorithm for selecting the max-weight base of a matroid

**Input:** a matroid $M = (E, I)$, where $E = \{1, 2, \ldots, n\}$ is the ground set, and weight of $i$ is $w_i$.

**Output:** A base $B \in I$ such that $w(B) = \max_{B \in B} w(B)$.

1. Relabel the elements of the matroid so that $w_1 \geq w_2 \geq \ldots \geq w_n$.
2. $S \leftarrow \emptyset$.
3. for $i \leftarrow 1$ to $n$ do
4.   if $S + i \in I$ then
5.     $S \leftarrow S + i$.
6.   end if
7. end for
8. return $S$
The greedy algorithm for matroids

Theorem 1 (Rado/Gale) For any ground set \( E = \{1, 2, \ldots, n\} \), and a family of subsets \( I \subseteq 2^E \), Algorithm 1 returns the maximum-weight base for any set of weights \( w : E \to \mathbb{R} \) if and only if \( M = (E, I) \) is a matroid. We prove this theorem in two parts.

\( \Leftarrow \) part) Suppose that \((E, I)\) is a matroid. For any set of weights assigned to the elements of \( E \), Algorithm 1 returns the maximum-weight base.

\( \Rightarrow \) part) Suppose \((E, I)\) is not a matroid. There exists an assignment of weights to the elements of \( E \) such that algorithm 1 does not return a maximum-weight base.
The greedy algorithm for matroids

(⇒ part) Suppose that $(E, I)$ is a matroid. For any set of weights assigned to the elements of $E$, Algorithm 1 returns the maximum-weight base.

Wlog assume that $w_1 \geq w_2 \geq \ldots \geq w_n$. We prove that at any point of the execution of the algorithm, there exists an optimal base $B$ such that $S \subseteq B$ and $B \setminus S$ is among the remaining elements.

We use induction to show that for any $S_i$, there exists an optimal base $B_i$ such that $S \subseteq B_i$, and $B_i \setminus S_i \subseteq \{i + 1, \ldots, n\}$.
The greedy algorithm for matroids

(⇒ part) Suppose \((E, I)\) is not a matroid. There exists an assignment of weights to the elements of \(E\) such that algorithm 1 does not return a maximum-weight base.

★ If \((E, I)\) is not a matroid, it does not satisfy at least one of the two properties of the matroid. Suppose \(I\) is not a downward-closed family of sets. Therefore, there exist two sets \(S \subset T, T \in I\), but \(S \notin I\). Suppose we assign the weights as follows:

\[
\forall 1 \leq i \leq n, \ w_i = \begin{cases} 
2 & i \in S \\
1 & i \in T \setminus S \\
0 & \text{otherwise}
\end{cases}
\]
The greedy algorithm for matroids

Suppose the algorithm selects a subset \( S_1 \subset S \) after observing the elements of \( S \). Since \( S \not\in I \), we have \( S_i \neq S \). Out of the remaining elements, the algorithm can get value at most \( |T \setminus S| \). If \( S_2 \) is the final set chosen by the algorithm, we have

\[
    w(S_2) = 2|S_1| + w(S_2 \setminus S) < 2|S| + |T \setminus S| = w(T).
\]

Now suppose \((E, I)\) is not a matroid because the extension axiom is violated (assume the downward closed property). In particular, let \( S, T \in I \) be two independent sets such that \( |S| < |T| \), and for all \( i \in T \setminus S \), \( S + i \not\in I \). We use the following weights:

\[
    \forall 1 \leq i \leq n, \quad w_i = \begin{cases} 
        1 + \frac{1}{2|S|} & i \in S \\
        1 & i \in T \setminus S \\
        0 & \text{otherwise}
    \end{cases}
\]
The greedy algorithm for matroids

Note that S is not necessarily a subset of T here. This time, because of the downward closedness property the algorithm would select all of the elements of S. But this means that it can not add any element in $T \setminus S$, as this would violate independence. Further elements do not bring any value anymore, so if $S_2$ is the solution returned by the algorithm,

$$w(S_2) = w(S) = |S| \left( 1 + \frac{1}{2|S|} \right) = |S| + \frac{1}{2}$$

while the value of T is

$$w(T) \geq |T| \geq |S| + 1.$$
The greedy algorithm for matroids

The following properties can be shown using the above theorem.

1. Let $S_i$ be the set of elements chosen by the algorithm after observing the first $i$ elements. Then $S_i$ is always a base of those $i$ elements.

2. Finding the maximum weight base in a matroid is in fact equivalent to finding the minimum weight base. Let $w_{\text{max}} = \max_{1 \leq i \leq n} w_i$ be the maximum weight assigned to the elements, to find the minimum weight base it is sufficient to replace $w_i := w_{\text{max}} - w_i$, for all $i \in E$.

3. Also by considering the same proof, it is straightforward that if the weights are non-negative, then the weight of the maximum weight independent set will be the same as the weight of the maximum weight base. In general, we can say the weight of the maximum weight independent set among the elements with non-negative weights is the same as the weight of the maximum weight base of those elements.
Proof of Correctness of MST with Matroids

- Kruskal’s algorithm is equivalent to finding the base with the lowest weight. Thus, the matroid equivalent of Kruskal’s algorithm is sorting the groundset by weight, and adding the lightest weight elements to I, only if the independence property is satisfied (since it’s a graphic matroid, the independence property is that there are not going to be any cycles).
- Keep doing this until we get a basis.
- To prove this is the correct algorithm, assume that our algorithm gives us the minimal base as \( \{x_1, x_2, ..., x_{|E|-1}\} \) with \( x_1 \leq x_2 \leq ... \leq x_{|E|-1} \) but the correct minimal base is \( \{y_1, y_2, ..., y_{|E|-1}\} \) with \( y_1 \leq y_2 \leq ... \leq y_{|E|-1} \). Then there is a \( k \) such that \( x_k > y_k \).
Proof of Correctness of MST with Matroids

- Consider the set \{x_1, ..., x_{k-1}\} and the set \{y_1, ..., y_k\}. Note that these are both independent sets, and that |\{x_1, ..., x_{k-1}\}| < |\{y_1, ..., y_k\}|. Then from the definition, we know that there is some element \(y_j\) in \{y_1, ..., y_k\} that we can add to \{x_1, ..., x_{k-1}\} while still maintaining the independence of the latter set. Note that

\[ w(y_j) \leq w(y_k) < w(x_k) \]

- but if this is the case, then we would have chosen \(y_j\) before \(x_k\) in our algorithm! This is because \(y_j\) weighs less, and maintains the independence when added! This is a contradiction, so it must be that the matroid analog of Kruskal’s algorithm gives us the optimal basis, as desired.
The rank function of a matroid \( M = (E, I) \) is

\[
r(S) = \max \{|A|: A \in I, A \subseteq S\}.
\]

Properties of the rank function

- \( 0 \leq r(X) \leq |X| \) and is integer valued for all \( X \subseteq E \)
- \( X \subseteq Y \Rightarrow r(X) \leq r(Y) \),
- \( r(X) + r(Y) \geq r(X \cap Y) + r(X \cup Y) \). (Submodularity)

In graphic matroids \( r(S) = |V| - \#\text{components}(S) \).
Proof of Submodularity

Consider any two sets $X, Y \subseteq$. Let $J$ be a maximal independent subset of $X \cap Y$; thus, $|J| = r(X \cap Y)$. By extension axiom, $J$ can be extended to a maximal (thus maximum) independent subset of $X$, call it $J_X$. We have that $J \subseteq J_X \subseteq X$ and $|J_X| = r(X)$. Furthermore, by maximality of $J$ within $X \cap Y$, we know

$$J_X \setminus Y = J_X \setminus J. \quad \ldots (1)$$

Now extend $J_X$ to a maximal independent set $J_{XY}$ of $X \cup Y$. Thus, $|J_{XY}| = r(X \cup Y)$. In order to be able to prove that

$$r(X) + r(Y) \geq r(X \cap Y) + r(X \cup Y)$$

or equivalently $|J_X| + r(Y) \geq |J| + |J_{XY}|$,

we need to show that

$$r(Y) \geq |J| + |J_{XY}| - |J_X|$$
Proof of Submodularity Contd.

Observe that $J_{XY} \cap Y$ is independent and a subset of $Y$, and thus $r(Y) \geq |J_{XY} \cap Y|$. Observe now that

$$J_{XY} \cap Y = J_{XY} \setminus (J_X \setminus Y) = J_{XY} \setminus (J_X \setminus J),$$

the first equality following from the fact that $J_X$ is a maximal independent subset of $X$ and the second equality by (1). Therefore,

$$r(Y) \geq |J_{XY} \cap Y| = |J_{XY} \setminus (J_X \setminus J)| = |J_{XY}| - |J_X| + |J|,$$
Span Function of Matroid

- Given a matroid $M = (E, I)$ and given $S \subseteq E$,
  
  \[ \text{let span}(S) = \{ e \in E : r(S \cup \{e\}) = r(S) \}. \]

- CLAIM: $r(S) = r(\text{span}(S))$ in other words, if adding an element to $S$ does not increase the rank, adding many such elements also does not increase the rank. Let's say $J$ is a maximal independent subset of $S$. If $r(\text{span}(S)) > |J|$ then there exists $e \in \text{span}(S) \setminus J$ such that $J + e \in I$. Thus $r(S + e) \geq r(J + e) = |J| + 1 > |J| = r(S)$ contradicting the fact that $e \in \text{span}(S)$. 

The matroid polytope

Let $X$ be the incidence vectors of all independent sets of a matroid $M = (E, I)$,

Then, $X = \{\chi(S) \in \{0, 1\}^{|E|} : S \in I\}$

$\text{conv}(X) = \text{Matroid Polytope}$

Let $P = \{x \in \mathbb{R}^{|E|} : x(S) \leq r(S) \quad \forall S \subseteq E$

$x_e \geq 0 \quad \forall e \in E\}$

where $x(S) := \sum_{e \in S} x_e$, then $\text{conv}(X) = P$. 
Primal and Dual LP

\[ \text{max } w^T x : \]
\[ x(S) \leq r(S) \quad \forall S \subseteq E \]
\[ x_e \geq 0 \quad \forall e \in E \]

\[ \text{min } \sum_S r(S)y_S : \]
\[ \sum_{S : e \in S} y_S \geq w_e \quad \forall e \in E \]
\[ y_S \geq 0 \quad S \subseteq E \]
Proof : conv(X) = P

- We know that the maximum cost independent set can be obtained by the greedy algorithm. More precisely, it is the last set $S_k$ returned by the greedy algorithm.
- For any index $j \leq k$, we have $S_j = \{s_1, s_2, \cdots, s_j\}$, define $U_j$ as all elements in the ordering up to and excluding $s_{j+1}$, i.e.

$$U_j = \{e_1, e_2, \cdots, e_l\} \text{ where } e_{l+1} = s_{j+1}.$$  

Then, $r(U_j) = r(S_j) = j$
Proof Contd.

- For $j = 1, \cdots, k$, let $y_{U_j} = c(s_j) - c(s_{j+1})$, $c(s_{k+1}) = 0$.
- By the ordering of the $c(.)$, we have that $y_S \geq 0$ for all $S$. In addition, for any $e \in E$, we have that

$$\sum_{S: e \in S} y_S = \sum_{j=t}^k y_{U_j} = c(s_t) \geq c(e),$$

- This shows $y$ is a feasible solution to the dual. Moreover, its dual value is:

$$\sum_S r(S)y_S = \sum_{j=1}^k r(U_j)y_{U_j} = \sum_{j=1}^k j(c(s_j) - c(s_{j+1})) = \sum_{j=1}^k (j-(j-1))c(s_j) = \sum_{j=1}^k c(s_j) = c(S_k).$$