Matroids

Jiashuai Lu
University of Texas at Dallas

Outline

What's a Matroid

1. Definition of Matroid
2. Independent Systems
3. Indepenent Set, Basis, Rank

We say $(S, \mathcal{I})$ is a **matroid** if S is a finite ***ground*** set and $\mathcal{I}$ is a collection of subsets of S such that the following properties are satisfied:

1. **Non-emptiness**: The empty set is in $\mathcal{I}$. (Thus, $\mathcal{I}$ is not itself empty.)
2. **Heredity**: If $\mathcal{I}$ contains a subset $X \subseteq S$, then $\forall Y \subset X$, $Y \in \mathcal{I}$.
3. **Exchange**: If X and Y are two sets in $\mathcal{I}$ where $|X| > |Y|$, then $\exists x \in X \setminus Y$, such that $Y \cup x \in \mathcal{I}$.

$(S, \mathcal{I})$ is called an **Independent System** if it satisfies the first two properties: non-emptiness and heredity properties.

A matroid is an independent system satisfying the exchange property.

Independent Set, Basis, Rank

1. Every $X \in \mathcal{I}$ is typically called an **independent set**.
2. An independent set $X \in \mathcal{I}$ is called a **basis** of the matroid if $\nexists Y \in \mathcal{I}$ such that $X \subset Y$, i.e., a basis is a maximal independent set of $\mathcal{I}$.
3. Every basis of a matroid $(S, \mathcal{I})$ has the same size, and we denote this size as the **rank** of this matroid.

Independent Set, Basis, Rank (cont'd)

It easy to use the exchange property to prove the uniqueness of the size of bases in a matroid by contradiction:

Suppose a matroid $(S, \mathcal{I})$ has two maximal independent sets X and Y with sizes $|X| > |Y|$, then there must exist some element $x \in X \backslash Y$ such that $Y \cup x \in \mathcal{I}$. Then Y is not a basis of $(S, \mathcal{I})$. A contradition.

Examples

1. Linear Matroid
2. Uniform Matroid
3. Graphic Matroid

Linear Matroid

Let M be an arbitrary $n \times m$ matrix, S be set of all m columns in M. Let $\mathcal{I}$ be a collection of all the linearly independent subsets of S. Then $(S, \mathcal{I})$ is a matroid.

(We say a subset $X \subseteq S$ is a linearly independent subset if all the columns in X are linearly independent.)

A subset $X \subseteq \{1, 2, \ldots, n\}$ is independent if and only if $|X| \leq k$. Any subset of $\{1, 2, \ldots, n\}$ of size k is a basis of $U_{k,n}$.

Graphic Matroid $M(G)$

Let $G = (V, E)$ is an undirected graph. Let $S = E$ and $\mathcal{I}$ be the collection of all sets of edges that induce a forest of G. $(S, \mathcal{I})$ is the graphic matroid $M(G)$ of G. Every spanning tree of G is a basis of $M(G)$.

Applications of Matroids

---

1. Matroid Optimization Problem
2. A Greedy Method Solving Matroid Optimization Problem
3. Find Minimum Spanning Tree with Greedy Method
4. Scheduling with Deadlines

A matroid $(S, \mathcal{I})$ is **weighted** if every element $e \in S$ is weighted non-negatively by a weight function $w : S \to \mathbb{R}_{\geq 0}$.

The **matroid optimization problem** is finding a basis with maximum total weight.

For example, if $M(G)$ is a graphic matroid for a undirected graph $G = (V, E)$, then the problem is finding a maximum spanning tree of G.

```
GreedyBasis(S[1..n], I, w[1..n]):
    sort S in decreasing order of weight w
    G ← ∅
    for i ← 1 to n
        if G ∪ {S[i]} ∈ I
            G ← G ∪ {S[i]}
    Return G
```

Theorem (3.1)

For any matroid $M = (S, \mathcal{I})$ *and any weight function* $w : S \to \mathbb{R}^+$, *GreedyBasis*$(S, \mathcal{I}, w)$ *returns a maximum-weight basis of* M.

Firstly, we introduce the greedy choice property:

Lemma (3.2)

Let x be the first element in the weight decreasing order list s.t. $\{x\} \in \mathcal{I}$, then there is a basis A containing x.

Proof: Suppose otherwise there exists another basis B with maximum total weight not containing x. We construct a subset A starting from $A = \{x\}$, according to the exchange property, when $|A| < |B|$, we could always find an element $y \in B - A$ make $A \cup \{y\} \in \mathcal{I}$. Let $A = A \cup \{y\}$ and repeat this step until we finally get a basis $|A| = |B|$. Since $w(x) \geq w(y), \forall y \in B$, we have $w(A) \geq w(B)$. □

Now we show the optimal substructure property:

Lemma (3.3)

Let x be the first element chosen by GreedyBasis. We define a matroid $M' = (S', \mathcal{I}')$ by:

$$S' = \{y \in S | \{x, y\} \in \mathcal{I}\}$$
$$I' = \{B \subseteq S' | B \cup \{x\} \in \mathcal{I}\}$$

Then finding a maximal total weight basis containing x in M reduces finding a maximal total weight basis in M'.

Theorem (3.4)

For any subset system $\mathcal{S}$ that is not a matroid, there is a weight function w such that GreedyBasis$(\mathcal{S}, w)$ does not return a maximum-weight set in $\mathcal{S}$.

Find MST with GreedyBasis

We already know that every basis of a graphic matroid $M(G)$ is a spanning tree of $G = (V, E)$, and GreedyBasis$(S, \mathcal{I}, w)$ gives us a basis with maximum total weight w.r.t. weight function w.

Let $w' : E \to \mathbb{R}$ be the original edge weight function for G. Firstly, we make every edge has non-negative weight by setting $w''(e) = w'(e) - min_{e' \in E}\{w'(e')\}$. After that, we construct $w : E \to \mathbb{R}$ by assigning $w(e) = (\sum_{e \in E} w''(e)) - w''(e)$.

GreedyBasis$(S, \mathcal{I}, w)$ gives us a minimum spanning tree of G. (Now GreedyBasis actually works as Kruskal's algorithm.)

Suppose you have a list T of n tasks $[t_1, \ldots, t_n]$ to do, each of them needs a full day to finish, a list $D = [d_1, \ldots, d_n]$ of deadlines for those tasks, and a list $P = [p_1, \ldots, p_n]$ of overdue penalties for T.

A schedule is a permutation π of $\{1, 2, \ldots, n\}$ which is an order you will work on the tasks. The question is how to find a schedule π such that the total penalty is minimized, i.e.

$$min_\pi \sum_{i=1}^{n} P[i] \cdot [\pi[i] > D[i]]$$

The cost of a schedule is determined by the subset of tasks that are on time.

Let π be a schedule.

Task t_i is late in π if it finishes after its deadline d_i, otherwise, it is early.

π is in early-first form if all early tasks precede late tasks.

π is in canonical form if it is early-first and all early tasks are scheduled in non-decreasing order of deadlines.

Every schedule π can be put into canonical form.

Definition: A set A of tasks is independent if there exists a schedule in which no task $t \in A$ is late.

Let $N_j(A) := |\{t_i \in A | d_i \leq j\}|$ for $j = 0, \ldots, n$.

Lemma (3.5)

A is independent $\iff$ $N_j(A) \leq j$ *for* $j = 0, \ldots, n$.

Theorem (3.6)

Let S be a set of all tasks in T, $\mathcal{I}$ be the collection of all independent sets of tasks. Then $(S, \mathcal{I})$ is a matroid.

Proof:

1. Hereditary: from the definition.

2. Exchange Property: consider two independent sets of tasks A and B with $|A| < |B|$.

   Let k be largest j s.t. $N_j(A) \geq N_j(B)$, then $k < n$ and for $j = k+1, \ldots, n$: $N_j(A) < N_j(B)$.

   Choose $x \in \{t_i \in B - A | d_i = k+1\}$, then $N_j(A \cup \{x\}) = N_j(A)$ for $j = 0, \ldots, k$ and $N_j(A \cup \{x\}) \leq N_j(A) + 1 \leq N_j(B) \leq j$ for $j = k+1, \ldots, n$. Thus, $A \cup \{x\}$ is an independent set. $\square$

1. Construct the weight function w by setting $w(t_i) = p_i$ for all $i = 1, \ldots, n$. Let $M = (S, \mathcal{I})$ as we defined in Theorem 3.6.
2. Use GreedyBasis to find a maximum-weight basis A of tasks.
3. Create an optimal schedule by first scheduling tasks in A by order of non-decreasing deadlines.

Time complexity: $O(n^2)$ (Sorting uses $O(n \log n)$. Checking independence costs $O(n)$ for each iteration. In total $O(n^2)$.)