Briefly, what is a Matroid[1]

丛宇

UESTC

2024-08-13

[1] The title is borrowed from https://www.math.lsu.edu/~oxley/matroid_intro_summ.pdf

A *matroid* is an important structure in combinatorial optimization, which generalizes the notion of linear independence in vector spaces.

Definition (Matroid): A matroid M is a pair $(E, \mathcal{J})$, where E is a finite set (ground set) and $\mathcal{J}$ is a family of subsets of E (independence sets). The family $\mathcal{J}$ satisfies the following properties:

- $\emptyset \in \mathcal{J}$
- If $I \in \mathcal{J}$ and $J \subseteq I$, then $J \in \mathcal{J}$
- If $I, J \in \mathcal{J}$ and $|I| < |J|$, then there exists an element $x \in J - I$ such that $I \cup \{x\} \in \mathcal{J}$.

Matroid is a blueprint for what independence should mean.

combinatorial optimization ≈ optimize over discrete structures

The knapsack problem is a classic combinatorial optimization problem.
The goal is finding the most valuable combination of items that can be fit into a knapsack of limited capacity.

CO problems are often NP-hard.

However, some discrete structures allow efficient optimization algorithms.

Matroid is such a structure.

linearly independent sets

Matrix is just a rectangular array of numbers, e.g.,

$$\begin{bmatrix} 1 & 2 & \dots & 10 \\ 2 & 2 & \dots & 10 \\ \vdots & \vdots & \ddots & \vdots \\ 10 & 10 & \dots & 10 \end{bmatrix}$$

set of column vectors

$$\left[\begin{pmatrix} 1 \\ 2 \\ \vdots \\ 10 \end{pmatrix} \dots \begin{pmatrix} 10 \\ 10 \\ \vdots \\ 10 \end{pmatrix} \right]$$

The family of linearly independent columns forms a matroid.

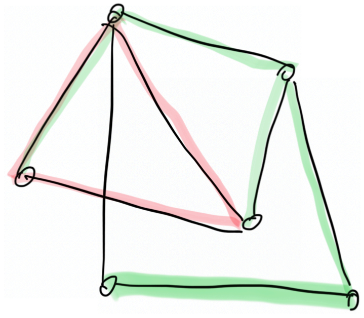
$\mathcal{I}$ = the family of linearly independent columns vectors

- $\emptyset \in \mathcal{I}$ (trivially true)
- If $I \in \mathcal{I}$ and $J \subseteq I$, then $J \in \mathcal{I}$ (subsets of linearly independent columns are linearly independent!)
- If $I, J \in \mathcal{I}$ and $|I| < |J|$, then there exists an element $x \in J - I$ such that $I \cup \{x\} \in \mathcal{I}$. (this is the Steinitz exchange lemma in linear algebra. 🔗)

This matroid is called the *vector matroid*.

Spanning Trees

A *spanning tree* of a graph $G = (V, E)$ is a subgraph of G and contains all the vertices of G but no cycle.



A graph with 6 vertices and 8 edges.
The 3 red edges form a cycle.
The 5 green edges form a spanning tree.

$E =$ the set of edges in G
$\mathcal{I} =$ the family of trees(subgraphs without cycles)

- $\emptyset \in \mathcal{I}$ (trivially true)
- If $I \in \mathcal{I}$ and $J \subseteq I$, then $J \in \mathcal{I}$ (subset of a tree is still a tree!)
- If $I, J \in \mathcal{I}$ and $|I| < |J|$, then there exists an element $x \in J - I$ such that $I \cup \{x\} \in \mathcal{I}$. (try to prove it)

This matroid is called the *graphic matroid*.

Greedy Algorithm

from the perspective of optimization

Problem (optimization over matroids): Given a matroid $M = (E, \mathcal{J})$ and a weight function $w : E \to R^{+}$, find a maximum-weight independent set.

Max-Weighted Independent Set

let $E = \{e_1, ..., e_n\}$ such that $w(e_1) \geq w(e_2) \geq ... \geq w(e_n)$
$x \leftarrow \emptyset$
for $i = 1$ to n **do**
    if $X \cup \{e_i\} \in \mathcal{J}$ **then**
        $X \leftarrow X \cup \{e_i\}$
return X

- For vector matroids, the greedy algorithm is conceptually the same as Gaussian elimination.
- For graphic matroids, the greedy algorithm is the same as Kruskal's algorithm.