Modeling project work 2

Optimization on valuated matroids

Student: Nihad Guliyev Supervisor: Prof András Frank

2020

Content

- Introduction
- Valuated matroid
- Examples of the valuation
- Local improvement algorithm
- Dress-Wenzel algorithm
- Application of Dress-Wenzel algorithm

Introduction

Matroids: H. Whitney (1935).

Definition 1a. A matroid is a pair $M = (S, \mathcal{F})$, where S is a ground set and $\mathcal{F}$ is a collection of independent sets,satisfying the following axioms:

Introduction

Matroids: H. Whitney (1935).

Definition 1a. A matroid is a pair $M = (S, \mathcal{F})$, where S is a ground set and $\mathcal{F}$ is a collection of independent sets,satisfying the following axioms:
(1) Empty set is an independent set;
(2) Any subset of an independent set is independent;
(3) The maximal independent sets of each subset of S are equicardinal.
And this maximal number is called the rank of a matroid and denoted by $r(X)$.

Definition 1b. A matroid is a pair $M = (S, \mathcal{B})$, where $\mathcal{B}$ is a collection of bases, satisfying the axioms:
(a) No proper subset of a basis is basis
(b) If B_1 and B_2 are two bases of M and $s_1 \in B_1 - B_2$, then there exists an element $s_2 \in B_2 - B_1$ such that $B_1 - s_1 + s_2$ is a basis.

Dress and Wenzel defined a new notion called valuated matroids.

Valuated matroid

Let $M = (S, \mathcal{B})$ be a matroid and ω be a function on 2^S which is finite on the bases $\mathcal{B}$. The function ω is called a **valuation** of M if it satisfies the following condition:

$$\omega(B_1) + \omega(B_2) \leq \omega(B_1 - s_1 + s_2) + \omega(B_2 - s_2 + s_1). \tag{1}$$

A matroid which is equipped with a function ω and satisfies the exchange property is called a valuated matroid.

Examples of the valuation

Example 1. The starting example for valuation is a linear valuation when we are given a weight function $c : \mathcal{S} \to \mathbb{R}$. The valuation by definition is the sum $\sum_{i=1}^{n} c_i$. If the set is not a basis, then the valuation is $-\infty$.

For such a linear valuation there is a version of greedy algorithm which results in the max-weight basis.

Example 2. We are given the matrix matroid when the terms of the matrix are polynomials in variable x. Assume that rows of this matrix are linearly independent. Column set is a ground set. The rank of this matrix is equal to m. Now, we have to calculate determinant of this matrix. The largest degree of this determinant is the valuation of the given matroid.

Let us consider the following matrix:

Example 2. We are given the matrix matroid when the terms of the matrix are polynomials in variable x. Assume that rows of this matrix are linearly independent. Column set is a ground set. The rank of this matrix is equal to m. Now, we have to calculate determinant of this matrix. The largest degree of this determinant is the valuation of the given matroid.

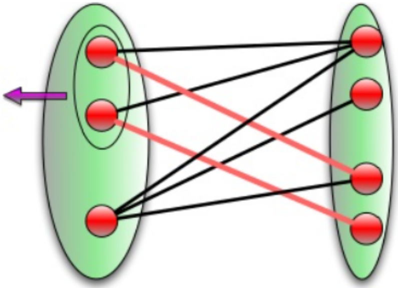
Let us consider the following matrix:

$$P = \begin{pmatrix} x & x^2 & 2x \\ 1+x & x^3 & 2+2x \end{pmatrix}$$

Let us denote the submatrix of this matrix containing first and second columns by P'.

$det(P') = x^4 - x^2 - x^3$. The largest degree of this polynomial is 4. Therefore, the valuation of this basis is 4.

Example 3. $G = (S, T; E)$- edge-weighted bipartite graph. This defines a matroid M on S (this is the so-called transversal matroid) in which a subset $I \subseteq S$ is independent if there is a matching of G covering I. The valuation of a basis $B \subseteq S$, by definition, is the maximum weight matching covering B. Then this is a valuation satisfying the axioms.



Local improvement algorithm

Lemma. A basis B is an ω-maximizer if and only if

$$\omega(B) \geq \omega(B - s + t) \text{ holds for every } s \in B, t \in S - B. \quad (2)$$

Local improvement algorithm

Lemma. A basis B is an ω-maximizer if and only if

$$\omega(B) \geq \omega(B - s + t) \text{ holds for every } s \in B, t \in S - B. \tag{2}$$

Algorithm: Check every s and t whether they satisfy the inequality $\omega(B) < \omega(B - s + t)$. If yes, then replace basis B with $B' := B - s + t$, and this is called a local improvement. If no local improvement exists, basis B is the final output of the algorithm and called ω-maximizer.

It is important to note that this algorithm is polynomial if the number of distinct ω-values is small.

Steepest ascent method

The local improvement algorithm can be slightly modified, and this modification will lead to the decrease of the total number of steps. Namely at each step, choose the pair $\{s, t\}$ in such a way that the increment $\omega(B - s + t) - \omega(B)$ is as large as possible. For an intermediate basis B, the algorithm checks all pairs $\{s, t\}$ with $s \in B, t \in S - B$ and find that one which maximizes the increase of ω. The total number of ω-evaluations is $r^2(n - r) + 1$.

Dress-Wenzel algorithm

This algorithm is more efficient than the previous ones. We are given a basis $B = \{s_1, s_2, \ldots, s_r\}$ in this order of elements. The algorithm consists of r phases. In Phase i we consider element s_i and find out whether s_i should remain in the current basis or it is replaced by an element t_i from outside of the current basis. If s_i remains in the basis, then this is a final decision, if s_i is replaced by t_i then it is final that t_i remains in the basis (but the element s_i falling out from the basis may return at a later phase).

Phase 1. Define $B_0 := B$.
Check whether there is an element t_1 in $S - B_0$ for which $\omega(B_0) < \omega(B_0 - s_1 + t_1)$. If no , then we define $B_1 := B_0$, and switch to Phase 2. Otherwise, we choose t_1, for which $\omega(B_0 - s_1 + t_1)$ is as large as possible and replace s_1 by t_1, that is, $B_1 := B_0 - s_1 + t_1$.

In Phase k we consider that a basis B_{k-1} is constructed in the previous Phase k-1 and check whether there is an element t in $S - B_{k-1}$ for which $\omega(B_{k-1}) < \omega(B_{k-1} - s_k + t_k)$.

If no, then we define $B_k := B_{k-1}$ and switch to Phase $k+1$ (if $k < r$). If yes, then we choose t_k, for which $\omega(B_{k-1} - s_k + t_k)$ is as large as possible, and replace s_k by t_k, that is, $B_k := B_{k-1} - s_k + t_k$.

In case when $k = r$, the greedy algorithm stops and we get B_r as an output.

Application of Dress-Wenzel algorithm

Let us consider the following example:
We are given a graphic matroid (complete graph K_4) with edges denoted by 1,2,3,4,5,6.

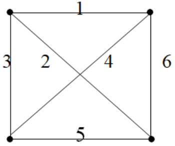


Figure:

K_4 matroid

Application of Dress-Wenzel algorithm

Let us consider the following example:
We are given a graphic matroid (complete graph K_4) with edges denoted by 1,2,3,4,5,6.

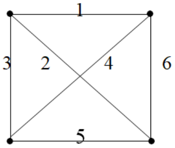


Figure:

K_4 matroid

The weight of edge i is i. In this case the unique max-weight basis (spanning tree) is $\{6, 5, 3\}$.
Suppose that the starting basis of the algorithm is $\{3, 2, 1\}$ in this order.
Then the greedy algorithm of Dress-Wenzel replaces edge 3 by an edge 5.

This means that the edge 3 leaves the basis. But later edge 3 must come back since our basis is $\{6, 5, 3\}$.
This basically means that an element removed at a certain moment from the current basis may come back later. This example simply shows an assymmetry between the leaving and coming elements: an element that leaves the current basis at a certain moment may come back, while an element entering the basis at a certain moment will never leave the basis later.

Thank you for your attention!