Pipage Rounding–Matroid Constraints

Jean Pouget-Abadie

March 3, 2015

1 Notation

Let N be a ground set of n elements and let f be a non-decreasing submodular set function $f : 2^N \to \mathbb{R}^+$. Let $\mathcal{M}$ be a matroid. We recall the definition of a matroid below:

Definition 1. *A* **matroid** *M is a pair (N, I), where N is a ground set and I is a family of subsets of N, called the independent sets of the matroid, such that:*

- *$I \neq \emptyset$ (generally, we suppose $\emptyset \in I$)*
- *$A \in B$ and $B \in I \implies A \in I$*
- *$\forall (A, B) \in I^2, |A| < |B| \implies \exists c \in B : A \cup \{c\} \in I$*

When it is not ambiguous, we will use $A \in I$ and $A \in M$ indiscrimanately. For a vector $y \in [0,1]^N$, we define $y(S) := \sum_{i \in S} y_i$. We define the **polytope of a matroid** M as:

$$P(M) := \left\{y \in [0,1]^N : y(S) \leq r_M(S) \ \forall S \subseteq N\right\}$$

We define a **tight** set with respect to y as any set such that $y(A) = r_M(A)$. Finally, the **base polytope** $\mathcal{B}(M)$ is defined as:

$$\mathcal{B}(M) := \left\{y \in [0,1]^N : y(N) = r_M(N)\right\}$$

To have some geometric interpretation in mind, the vertices of $P(M)$ are the independent sets of M and the vertices of $B(M)$ are the bases of M. Note that it is easy to optimize linear functions over $P(M)$. In fact, for weight functions, the greedy algorithm is optimal. We are going to need two properties of tight sets and optimization on matroid polytopes. Since these properties are not proved in [2], we prove them below.

Proposition 1. *If A and B are tight sets with respect to y, $A \cup B$ and $A \cap B$ are tight sets with respect to y.*

Proof. Suppose that A and B are tight with respect to y:

$$\begin{aligned} y(A \cup B) + y(A \cap B) &= y(A) + y(B) \\ &= r_M(A) + r_M(B) \\ &\geq r_M(A \cup B) + r_M(A \cap B) \end{aligned}$$

The final inequality is by submodularity of the matroid rank function. Observing that

$$y(A \cup B) + y(A \cap B) \leq r_M(A \cup B) + r_M(A \cap B)$$

completes the proof. □

Proposition 2. *If $\tilde{f}$ is non-decreasing, then there exists an optimum solution y^* to $\max\left\{\tilde{f}(y) : y \in P(M)\right\}$ such that N is tight with respect to y^*.*

Proof. Suppose that N is not tight w.r.t. y^*. Suppose that there is a variable y_j which belongs to no tight set with respect to y^*, then we can increase y^* along this coordinate until we reach a tightness constraint and still remain at optimum, since $\tilde{f}$ is non-decreasing. Suppose no such variable exist, then the set of tight sets covers all variables, and therefore, their union is equal to N. From the previous proposition, N is also tight. □

2 Framework

Our objective is to maximize $f(S)$ for $S \in I$:

$$\max_{S \in M} f(S)$$

Suppose we are given a monotone function F defined on the polytope $P(M)$, which coincides with f on vectors $y \in \{0,1\}^N$ such that $\forall (i,j),\ t \mapsto F(y + t(e_i - e_j))$ is convex (F is said to be cross-convex). Then it can be shown that any fractional vector y can be rounded to an integral solution $\hat{y}$ in polynomial time such that $F(\hat{y}) \geq F(y)$. This is known as **pipage rounding**. We will show that for any submodular function, there is a canonical relaxation, known as the **multilinear relaxation**, which verifies these properties.

However, the multilinear relaxation is not necessarily optimizable. We are therefore going to construct extensions of f, $\tilde{f}$ on the matroid polytope $P(M)$, which are monotone, coincide with f if $y \in \{0,1\}^N$, can be optimized in polynomial time, and such that $\exists \alpha, \forall y, F(y) \geq \alpha \tilde{f}(y)$. The general framework is then:

1. Optimize function $\tilde{f}$ on the matroid polytope $P(M)$ which outputs fractional solution $\tilde{y} \in P(M)$

2. Round $\tilde{y}$ using pipage rounding and output integral solution $\hat{y} \in M$ such that $F(\hat{y}) \geq F(\tilde{y})$

3. Output $\hat{y}$. We have $f(\hat{y}) = F(\hat{y}) \geq F(\tilde{y}) \geq \alpha \tilde{f}(\tilde{y}) \geq \alpha f(y^*) = \alpha OPT$, where y^* is the optimal solution to $\max_{y \in M} f$.

3 Pipage Rounding

Observe that any tight set cannot contain exactly one non-integral variables because the rank constraints are integral. Let y_i and y_j be two fractional variables. Let $y^t_{i,j} : t \mapsto y + t(e_i - e_j)$. If a set contains both y_i and y_j, then for any value of t, its constraint is not violated. If a set contains only y_i (or y_j), then its constraint cannot be matched exactly. We can therefore define $\epsilon^+ > 0$ (resp. $\epsilon^- < 0$) to be the greatest (resp. smallest) value that can be added to y without violating any constraints. Note that both $y^+_{i,j} := y + \epsilon^+(e_i - e_j)$ and $y^-_{i,j} := y + \epsilon^-(e_i - e_j)$ are in the matroid polytope.

In such a way, as long as y is fractional, we can find two fractional variables and exchange mass from one to the other without decreasing the value for F. Unlike the [1] paper, we are not guaranteed to make one of the (i,j) integral at the end of a single iteration. However, we show that after n iterations, we are guaranteed to gain at least one integral variable.

Data: Fractional y
while *y is not integral* **do**
  Let A be minimal tight set containing fractional $(i,j) \in A$ **if** $F(Y^+_{i,j}) \geq F(Y^-_{i,j})$ **then**
    $y \leftarrow y^+_{i,j}$
  else
    $y \leftarrow y^-_{i,j}$
  end
end
Result: Output y, f(y)

Finding a minimal tight set The algorithm requires us to find a minimal tight set. This can in fact be done in polynomial time. We supposed "wlog" that N is a tight set for y^*. At every iteration, we can remove an element from the current set, starting at N, until we are no longer tight.

Proving that in n^2 iterations, y is integral At iteration h, let A_h be the minimal tight set chosen. The main idea is to prove that at every iteration, one of two things happen:

1. $|A_{h+1}| < |A_h|$

2. y_{h+1} has one more integral variable than y_h

Proof. (sketch) The author's provide a proof for a slightly modified algorithm, and state that the proof is also true in the original case. The modification is the following: A_h is not only a minimal tight set, but it is a minimal tight set of miminum cardinality among such minimal sets.

Suppose that you do not get one more integral variable at iteration $h+1$, then there is another tight set B which stopped us from increasing $|\epsilon|$ enough to gain an extra integral variable. B can only contain one of the two variables, otherwise it could not have become tight as we exchanged mass. Therefore $|A_h \cap B| < |A_h|$. Since A_h and B are tight sets, then $A_h \cap B$ is also tight and $|A_{h+1}| < |A_h|$. □

It follows that y_{n+h-1} has at least one more integral variable than y_h (since $|A_h| \leq n$) and that pipage rounding runs in at most $\mathcal{O}(n^2)$ iterations.

4 Relaxing f

Multilinear Extension For any submodular function f, there is a canonical extension which is cross-convex, defined as such

$$F(y) := \mathbb{E}_{\hat{y}\sim y} f(\hat{y})$$

where $\hat{y} \sim y$ is chosen by independently rounding up $y's$ value with probability y_i and rounding down with probability $1-y_i$. It is easy to see that $\forall y \in \{0,1\}^N, F(y) = f(y)$. It is also easy to see that if f is monotone, then F is also monotone.

Proposition 3. *F is cross-convex: $t \mapsto F(y + t(e_i - e_j))$ is convex on its domain.*

Proof. Let $(i,j) \in N^2$ and let $p_y(S)$ be the probability that S is the set obtained by randomized rounding on $N \backslash i, j$.

$$\begin{aligned} F(y) = \sum_{S \subseteq N\backslash\{i,j\}} & p_y(S)[(1-y_i)(1-y_j)f(S) \\ & + (1-y_i)y_j f(S+j) \\ & + y_i(1-y_j)f(S+j) \\ & + y_i y_j f(S+i+j)] \end{aligned}$$

$t \mapsto F(y + t(e_i - e_j))$ is a degree two polynomial with leading coefficient $-f(S) + f(S+j) + f(S+i) - f(S+i+j) \geq 0$ by submodularity. □

We are now going to consider several interesting extensions of f, which in certain cases, can be shown to be optimizable in polynomial time and can be proven to verify $F(y) \geq \left(1 - \frac{1}{e}\right)\tilde{f}(y)$.

[Open Question]. *The constant-factor approximation only needs to be verified at the optimal y in $P(M)$ for $\tilde{f}$. Can we show that if it is verified for the optimal y, then under certain properties of $\tilde{f}$, it is verified for all y?*

Extension f^+

$$f^+(y) = \max\{\sum_{S\subseteq N} \alpha_S f(S) : \sum_S \alpha_S \leq 1, \alpha_S \geq 0, \forall j, \sum_{S:j\in S} \alpha_S \leq y_j\}$$

Let us provide a small interpretation of this function. Notice that it is an LP. However, in the general case, it cannot be optimized in polynomial time since it has exponentially many variables. For this function, instead of picking items independently with probability y_j (as we did in F), we pick sets. We can therefore think of f^+ as a correlated distribution for y, whereas F is an independent distribution.

[Open Question]. *What is the dual of f^+? The dual of f^+ will have exponentially many constraints, but there are cases where we can still optimize such an LP. Can the dual of f^+ be optimized in polynomial time?*

Extension f^*

$$f^*(y) = \min\left\{ f(S) + \sum_{j\in N} f_S(j) y_j : S \subseteq N \right\}$$

This extension is easy to analyse. Note that here we are minimizing! Consider a binary vector $y \in \{0,1\}^N$. If $y_j = 1$, then by not including element j in S, you pay $f_S(j)$ and $f_S(j')$ for any other element j', which would be more than if you included j in S. If $y_j = 0$, then if you include element j in S, you increase the value of $f(S)$, whereas if you do not, you pay no additional penalty. It is easy to see then that for any integral vector y, $f^*(y) = f(S)$ where $y = \mathbb{1}_S$

[Open Question]. *We can think of f^* to be a first-order extension of f. What does extension mean? What would a second-order extension look like?*

Lemma 1. *If f is monotone submodular:*

$$F(y) \leq f^+(y) \leq f^*(y)$$

Proof. $f^+(y) \geq F(y)$ since taking α_S equal to the probability of obtaining set S by randomized rounding of y is a feasible vector for f^+. To see that $f^*(y) \geq f^+(y)$, observe that for any set $T \subseteq S$ and any feasible vector α_S:

$$\begin{aligned}\sum_S \alpha_S f(S) \leq & \sum_S \alpha_s \left(f(T) + \sum_{j \in S} f_T(j) \right) \\ \leq & f(T) + \sum_{j \in N} y_j f_T(j)\end{aligned}$$

□

Lemma 2. *For any monotone submodular f, $F(y) \geq \left(1 - \frac{1}{e}\right) f^*(y)$*

Proof. The general idea of the proof is to build a set $S(t)$ incrementally and randomly, such that $(1 - \frac{1}{e}) f^*(y) \leq \mathbb{E}(f(S(1))) \leq F(y)$.

How do we construct the set $S(t)$? For each element, include it in $S(t)$ with an independent poisson clock C_j of parameter y_j. In other words, as soon as the poisson clock C_j "rings" for element j, include it in $S(t)$ and for all future time steps.

Recall that the probability that

$$\mathbb{P}\left(C(t + dt) - C(t) = k\right) = \frac{e^{-\lambda dt}(\lambda dt)^k}{k!}$$

Note that the probability we include an element j in $S(1)$:

$$\begin{aligned}\mathbb{P}(y_j \in S(1)) = 1 - e^{-y_j} \leq y_j \\ \implies \mathbb{E}(f(S(1))) \leq F(y)\end{aligned}$$

We can now write the following differential equation from the independence of the poisson clocks and "memoryless-ness" of a poisson process.

$$\begin{aligned}\mathbb{E}(f(S(t + dt)) - f(S(t)) | S(t) = S) & \\ &= \sum_{j \in N} f_S(j) y_j dt \\ &\geq (f^*(y) - f(S)) dt\end{aligned}$$

The first equality is obtained by observing that, conditioned on $S(t)$, the marginal increase to $f(S(t))$ for very small dt is the sum of the disjoint event that clock C_j "rings" between t and $t + dt$. It is easy to see that this probability is $y_j dt$, by observing the expression of the poisson process. The second inequality is the fundamental lemma which holds for submodular functions. Dividing by dt on both sides, and by taking the expectation with respect to $S(t)$, we get the following differential equation from the property of iterated expectations:

$$\begin{aligned}\frac{1}{dt}\big(\mathbb{E}(f(S(t + dt))) - f(S(t))\big) & \\ &\geq f^*(y) - \mathbb{E}(f(S(t)))\end{aligned}$$

Solving the differential equation in $\phi : t \mapsto \mathbb{E}(f(S(t))) : \phi'(t) + \phi(t) \geq f^*(y)$ we get $\phi(t) \geq (1 - e^{-t}) f^*(y)$, which concludes the proof by taking $t = 1$. □

5 Weighted Rank Functions

Note that in general f^* and f^+ cannot be optimized in polynomial time. In [2], the authors cover a special case of submodular functions for which f^+ is optimizable in polynomial time: sums of weighted rank functions.

We simply the presentation of the paper by considering a single matroid rank function, the extension is straight-forward. Let

$$g(S) := \max\{w(I) : I \subset S, I \in X\}$$

where X are the independent sets of a matroid.

$$\begin{aligned} f^+(y) = \max\{ \sum_{S \subseteq N} \alpha_S g(S) : \sum_S \alpha_S \leq 1, \\ \alpha_S \geq 0, \forall j, \sum_{S: j \in S} \alpha_S \leq y_j \} \\ = \max\{ \sum_{I \in X} \alpha_I \sum_{j \in I} w_j : \sum_{I \in X} \alpha_S \leq 1, \\ \alpha_S \geq 0, \forall j, \sum_{I \in X : j \in I} \alpha_S \leq y_j \} \end{aligned}$$

by assuming without loss of generality that α_S is nonzero only for $S \in X$. Indeed, any S can be substituted for the value of the maximal independent set of S in X without changing the value of $g(S)$.

We can inverse the summation:

$$\sum_{I \in X} \alpha_I \sum_{j \in I} w_j = \sum_{j \in N} \sum_{I \in X : j \in I} w_j \alpha_I$$

Let $x_j = \sum_{I \in X : j \in I} \alpha_I$. Notice that $x_j \in P(X)$, and that any vector of $P(X)$ can be written in this way. Indeed, $\sum_{I \in X} \alpha_I = 1$. The optimisation problem becomes:

$$g^+(y) = \max\{ \sum_{j \in N} w_j x_j : x \in P(X), \forall j,\ x_j \leq y_j \}$$

This problem can be solved using the ellipsoid method since a separation oracle can be implemented for each matroid polytope and therefore also for this LP.

6 Conclusion

In other words, for any monotone submodular function, if you can optimize either of its extension f^* or f^+ in polynomial time, then this framework allows you to get a $(1 - \frac{1}{e})$-solution under matroid constraint. As noted in the paper, in general, f^* and f^+ are not computable or optimizable in polynomial time unless $P = NP$. The authors show that for any sum of weighted rank functions of matroids, f^+ can be formulated as an LP which can be optimized in polynomial time.

References

[1] Alexander A Ageev and Maxim I Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8(3):307–328, 2004.

[2] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. In *Integer programming and combinatorial optimization*, pages 182–196. Springer, 2007.