1 The Continuous Greedy Algorithm

In the previous lecture, we introduced the overall approach to the problem of maximizing submodular functions over matroids. Having defined a suitable fractional version of the problem, we now give the approximate continuous greedy algorithm for solving the fractional version.

Algorithm 1 Continuous Greedy Algorithm ($\vec{y}(t)$ is a point in $\mathcal{P}$ at time $t$)

1. Start with $\vec{y}(0) = \vec{0}$.
2. Continuously, for 1 unit of time total, move in the direction of maximum gradient $\frac{\partial \vec{y}}{\partial t} = \hat{v}(\vec{y}(t))$, where $\hat{v}(\vec{y}(t)) \in \arg\max_{\vec{v} \in \mathcal{P}} \vec{v} \cdot \nabla F(\vec{y}(t))$.

It is fairly easy to show that without loss of generality, at each time $t$, $\hat{v}(\vec{y}(t))$ is a combination of basis vectors $\vec{y}_S$ (as opposed to other independent sets). Otherwise, we could simply replace the $\alpha_S > 0$ for some non-basis set $S$ with $\alpha_{S'} > 0$ for some basis $S' \supseteq S$, achieving no smaller gradient.

Remark 1 To turn this into an actual algorithm, there are some technical issues that we will return to in the next lecture.

1. We need to discretize time so that we can run the algorithm in a finite number of steps. How does this affect the approximation?
2. How can we evaluate $F(\vec{y})$?
3. How do we find the arg max?

We first show the the algorithm returns a feasible solution, and then analyze its approximation quality.

Proposition 2 Algorithm 1 gives a feasible solution. That is, $\vec{y}(1) \in \mathcal{P}$.

Proof. The basic idea underlying the proof is that we construct a convex combination of points in the polytope, which is again in the polytope.

For each time $t$, the direction is a convex combination of bases: $\hat{v}(\vec{y}(t)) = \sum_S \alpha_{t,S} \vec{y}_S$, where $\sum_S \alpha_{t,S} = 1$, and $\alpha_{t,S} \geq 0$. Therefore, $\vec{y}(1) = \int_0^1 \hat{v}(\vec{y}(t)) \, dt = \sum_S \vec{y}_S \int_0^1 \alpha_{t,S} \, dt$.

Let $\int_0^1 \alpha_{t,S} \, dt = \beta_S \geq 0$. We have $\sum_S \beta_S = \int_0^1 \sum_S \alpha_{t,S} \, dt = \int_0^1 1 \, dt = 1$. Therefore, $\vec{y}(1)$ is a convex combination of $\vec{y}_S$, and $\vec{y}(1) \in \mathcal{P}$. 

Proposition 3 Algorithm 1 is a factor $(1 - \frac{1}{e})$ approximation algorithm.

Proof. Let $\vec{x}$ be an optimum vector. Thus, we have $\text{OPT} = F(\vec{x})$. Define $\vec{v} = \min(\vec{x} - \vec{y}(t), \vec{0})$, coordinate-wise. (The idea in this definition and the subsequent proof is similar to when we were looking at $f(S \cup \text{OPT}) - f(S)$ in the analysis of the Nemhauser-Wolsey algorithm.) By concavity (which we proved last class), $F(\vec{x}) - F(\vec{y}(t)) \leq 1 \cdot \vec{v} \cdot \nabla F(\vec{y}(t))$, where 1 denotes one unit of time. Because the matroid is downward closed, so is $\mathcal{P}$, i.e., if $\vec{z} \in \mathcal{P}$ and $\vec{z}' \leq \vec{z}$ coordinate-wise, then $\vec{z}' \in \mathcal{P}$. In our case, $\vec{v} \leq \vec{x}$ coordinate-wise, so $\vec{v} \in \mathcal{P}$, and was a candidate for $\hat{v}(\vec{y}(t))$. 

1
Write \( g(t) := F(\vec{y}(t)) \) for the algorithm’s function value at time \( t \). Then, \( \frac{dg}{dt} \geq \text{OPT} - g(t) \) by the choice of the algorithm. We have \( g(0) = 0 \), and we want \( g(1) \). Since \( \frac{dg}{dt} \geq \text{OPT} - g(t) \), \( g(t) \) is lower-bounded by the solution to \( \frac{dg}{dt} = \text{OPT} - \phi(t) \).

Let \( \psi(t) = \text{OPT} - \phi(t) \). Then \( \phi(t) = \text{OPT} - \psi(t) \). With the substitution, we obtain \( \frac{d\psi}{dt} = -\frac{d\phi}{dt} \), so the differential equation becomes \( -\frac{d\psi}{dt} = \psi(t) \). Integrating as \( \int_{0}^{T} -\frac{d\psi}{\psi} = \int_{0}^{T} dt \) gives us a solution of \( -\log \frac{\psi(t)}{\psi(0)} = t \), or \( \psi(t) = \text{OPT} \cdot e^{-t} \). Substituting back, we find that \( \phi(1) = \text{OPT} \cdot (1 - \frac{1}{e}) \).

## 2 Pipage Rounding

Having found a fractional solution \( \vec{y} = \vec{y}(1) \), we next want to round it to an integral solution (a basis). We will introduce a technique called Pipage Rounding for this. The idea is as follows. While there are fractional entries \( y_i, y_j \in (0, 1) \), replace the pair by \( y_i + \delta, y_j - \delta \) or \( y_i - \delta, y_j + \delta \) such that at least one of \( y_i, y_j \) becomes integral, and the quality of the solution does not decrease (too much). In trying to apply this general technique in our case, we have two issues to deal with:

1. We have to make sure that the objective does not decrease at all in our case. (In other applications of Pipage Rounding, it might decrease.)

2. While ideally, we would like to make at least one variable integral, this may not always work in our case: there may be matroid polytope constraints that prevent us from making variables integral. In that case, we need a systematic way to find a pair \((i,j)\) of variables for which making them integral works. We will achieve this with a more careful progress measure than just the number of integral entries.

But first, we want to show that there is always (at least) one direction in which the solution does not get any worse.

**Proposition 4** While there are fractional entries \( y_i, y_j \in (0, 1) \) in the fractional solution, at least one of replacing the pair by \( y_i + \delta, y_j - \delta \) or \( y_i - \delta, y_j + \delta \) does not decrease the objective.

**Proof.** For fixed \( i, j \), define \( F^y_{ij}(t) := F(\vec{y} + t \cdot \vec{d}_{ij}) \), where

\[
d_{ij,k} = \begin{cases} 1, & \text{if } k = i \\ -1, & \text{if } k = j \\ 0, & \text{otherwise} \end{cases}
\]

That is, we increase \( y_i \) at the same rate at which we decrease \( y_j \), and vice versa.

To see whether \( F^y_{ij}(t) \) decreases or not, we calculate the derivative and the second derivative.

\[
\frac{dF^y_{ij}}{dt} = \frac{\partial F}{\partial y_i} - \frac{\partial F}{\partial y_j},
\]

\[
\frac{d^2 F^y_{ij}}{dt^2} = \frac{\partial^2 F}{\partial y_i^2} - \frac{\partial^2 F}{\partial y_i \partial y_j} - \frac{\partial^2 F}{\partial y_j \partial y_i} + \frac{\partial^2 F}{\partial y_j^2}.
\]

Since \( \frac{\partial^2 F}{\partial y_i^2} = \frac{\partial^2 F}{\partial y_j^2} = 0 \), and \( \frac{\partial^2 F}{\partial y_i \partial y_j} \leq 0 \) (which we saw in the last lecture), we have

\[
\frac{d^2 F^y_{ij}}{dt^2} = -2 \cdot \frac{\partial^2 F}{\partial y_i \partial y_j} \geq 0
\]

Since the second derivative is non-negative, it follows that \( F^y_{ij} \) is convex. The convexity of \( F^y_{ij}(t) \) implies that it cannot have a local maximum. That is, in at least one direction of positive/negative \( t \), \( F \) does not decrease.
Since the objective does not decrease at all, we would be doing great if we could always make one of \( y_i, y_j \) integral, but there may be other faces of \( P \) preventing it. We identify such faces with the sets of variables they correspond to, and call these sets of variables \textit{tight}. We define the notion of tightness in terms of the rank of a set.

\textbf{Definition 5}  
1. The \textit{rank} of set \( A \) is \( \text{rank}_M(A) = \max_{S \subseteq \mathcal{I}, S \subseteq A} |S| \).

2. Writing \( y(A) := \sum_{i \in A} y_i \), we define a set \( A \) as \textit{tight} for \( \vec{y} \) if and only if \( y(A) = \text{rank}_M(A) \).

Notice that \( y(A) \leq \text{rank}_M(A) \) for every feasible vector \( \vec{y} \) and set \( A \), because each \( y \in P \) is a convex combination of \( \vec{y}_S, S \in \mathcal{I} \), and \( \vec{y}_S(A) \leq \text{rank}_M(A) \), for all \( S \in \mathcal{I} \). The name “rank” is motivated by the linear matroid: the rank of a set of vectors (a matrix) is the maximum number of linearly independent vectors in the set.