6.883 Learning with Combinatorial Structure
Note for Lecture 17
Author(s): Yeesian Ng

1 Maximizing a submodular function

We previously looked at methods for maximizing monotone submodular functions. In this lecture, we turn our attention to the maximization of non-monotone submodular functions $F(\cdot)$, but require that $F(S) \geq 0$ for all $S \subseteq \mathcal{V}$. Examples of such functions include graph-cuts and "coverage-minus-cost".

In particular, we will cover the following 3 algorithms for maximizing $F(\cdot)$:

- Random Selection
- Local Search
- Double/bidirectional Greedy

before looking at submodular maximization for two types of diffusion models.

2 Random Selection

Let F be a non-monotone non-negative submodular function, and pick set R uniformly at random from $\mathcal{V}$, where

$$\mathbb{P}(i \in R) = 0.5 \quad \forall\, i \in \mathcal{V}$$

Although the performance of such a set R can be arbitrarily bad, the expected value of such a set can be shown to be

$$\mathbb{E}[F(R)] \geq \frac{1}{4} F(S^*)$$

where $S^* \in \arg\max_{S \subseteq \mathcal{V}} F(S)$. Moreover, if $F(\cdot)$ is a symmetric function, then

$$\mathbb{E}[F(R)] \geq \frac{1}{2} F(S^*).$$

Remark: In practice, you might want to do better, e.g. by performing local updates to improve the set returned by "random selection".

3 Local Search

The algorithm for local search to solve the problem

$$\max_{S \subseteq \mathcal{V}} F(S)$$

is as follows

1. Let $S = \arg\max_{e \in \mathcal{V}} F(S)$
2. While there is an element $e \in \mathcal{V} \setminus S$ such that $F(S \cup e) > (1 + \frac{\epsilon}{n^2})F(S)$
   - add e to S
3. If there is an element $e \in S$ such that $F(S \setminus e) > (1 + \frac{\epsilon}{n^2})F(S)$
   - remove e from S
   - goto step (2)
4. return maximum of $F(S)$ and $F(\mathcal{V} \setminus S)$.

where ϵ is a user-defined parameter for the tolerance, and $n = |\mathcal{V}|$.

We state[1] the following result [1].

Theorem 1 ((Feige, Mirrokni, Vondrak 2007))**.** *Let S_{LS} be the set returned by local search. Then*

$$F(S_{LS}) \geq (\frac{1}{3} - \frac{\epsilon}{n})F(S^*)$$

In addition, if $F(\cdot)$ is symmetric, then the bound can be tightened to

$$F(S_{LS}) \geq (\frac{1}{2} - \frac{\epsilon}{n})F(S^*)$$

Remark: In fact, it has been shown that achieving a lower bound of $\frac{1}{2}$ from optimality is the best you can do in general for any algorithm that runs in polynomial time.

4 Double (bidirectional) Greedy

Instead of local search, the double (or bidirectional) greedy algorithm maintains 2 sets, as it makes a single pass through $\mathcal{V}$.

[1] stated without proof here, to focus on the proof for the double-greedy algorithm in section 4

4.1 Deterministic version

The (deterministic) algorithm is as follows:

1. Start with $A = \emptyset$ and $B = \mathcal{V}$.
2. For $i \in \{1, \ldots, n\}$
   a) Calculate

$$\alpha_i = F(A_{i-1} \cup e_i) - F(A_{i-1})$$
$$\beta_i = F(B_{i-1} \setminus e_i) - F(B_{i-1})$$

   b) if $\alpha_i \geq \beta_i$, update $A_i = A_{i-1} \cup \{e_i\}$ and $B_i = B_{i-1}$
   c) otherwise, update $A_i = A_{i-1}$ and $B_i = B_{i-1} \setminus \{e_i\}$
3. Return $A_n (= B_n)$

We first study the proof of a weaker result for the deterministic double greedy algorithm, before sketching out an analogous proof for the randomized version. We need the following two lemmas:

Lemma 1. *For all $i \in \{1, \ldots, n\}$, we have $\alpha_i + \beta_i \geq 0$.*

Proof. Note that

$$(A_{i-1} \cup e_i) \cup (B_i \setminus e_i) = B_{i-1}$$
$$(A_{i-1} \cup e_i) \cap (B_i \setminus e_i) = A_{i-1}$$

By the submodularity of $F(\cdot)$, we have

$$\begin{aligned}
\alpha_i + \beta_i &= [F(A_{i-1} \cup e_i) - F(A_{i-1})] + [F(B_{i-1} \setminus e_i) - F(B_{i-1})] \\
&= [F(S) - F(S \cap T)] + [F(T) - F(S \cup T)] \\
&\geq 0
\end{aligned}$$

where

$$S = A_{i-1} \cup e_i, \quad S \cap T = A_{i-1}$$
$$T = B_{i-1} \setminus e_i, \quad S \cup T = B_{i-1}$$

□

Lemma 2. *Let OPT_i be the set that agrees with A_i, B_i on $\{e_1, \ldots, e_i\}$ and with OPT on $\{e_{i+1}, \ldots, e_n\}$. We have*

$$OPT_0 = OPT, \quad OPT_n = A_n = B_n$$

and for all $i \in \{0, \ldots, n\}$,

$$F(OPT_{i-1}) - F(OPT_i) \leq F(A_i) - F(A_{i-1}) + F(B_i) - F(B_{i-1})$$

Proof. Assume, without loss of generality, that $\alpha_i \geq \beta_i$. Then

$$\begin{aligned} A_i &= A_{i-1} \cup e_i \\ B_i &= B_{i-1} \\ \mathrm{OPT}_i &= \mathrm{OPT}_{i-1} \cup e_i \end{aligned}$$

We have two cases.

Case 1: If $e_i \in \mathrm{OPT}$, then $0 \leq F(A_i) - F(A_{i-1}) = \alpha_i$.

Case 2: If $e_i \notin \mathrm{OPT}$, then by inverse marginal gains,

$$F(\mathrm{OPT}_{i-1}) - F(\mathrm{OPT}_i) \leq F(B_{i-1} \setminus e_i) - F(B_{i-1}) = \beta_i \leq \alpha_i$$

□

Putting together the two lemmas, we have

Theorem 2. *Let $F(\cdot)$ be a submodular set function, and S_g be the set returned by the double greedy algorithm. Then*

$$F(S_g) \geq \frac{1}{3} F(S^*)$$

Proof.

$$\begin{aligned} F(\mathrm{OPT}) - F(A_n) &= F(\mathrm{OPT}_0) - F(\mathrm{OPT}_n) \\ &= \sum_{i=1}^{n} F(\mathrm{OPT}_{i-1}) - F(\mathrm{OPT}_n) \\ &\leq \sum_{i=1}^{n} (F(A_i) - F(A_{i-1}) + F(B_i) - F(B_{i-1})) \\ &= F(A_n) - F(A_0) + F(B_n) - F(B_0) \\ &\leq F(A_n) + F(B_n) \end{aligned}$$

where the first inequality follows from lemma 2, the last equality from telescoping sums, and the last inequality from the non-negativity of $F(\cdot)$: $F(A_0), F(B_0) \geq 0$.

By re-arranging terms, we get $F(\mathrm{OPT}) \leq 3F(A_n)$. □

4.2 Randomized version

In the randomized variant of the algorithm, we perform step (2b) of the algorithm with probability $\frac{\alpha_i}{\alpha_i+\beta_i}$, and perform step (2c) otherwise. Then lemma 2 will be modified to

Lemma 3. *Let OPT_i be the set that agrees with A_i, B_i on $\{e_1, \ldots, e_i\}$ and with OPT on $\{e_i, \ldots, e_n\}$. We have*

$$OPT_0 = OPT, \quad OPT_n = A_n = B_n$$

and for all $i \in \{0, \ldots, n\}$,

$$\mathbb{E}[F(OPT_{i-1}) - F(OPT_i)] \leq \frac{1}{2}\mathbb{E}[F(A_i) - F(A_{i-1}) + F(B_i) - F(B_{i-1})]$$

By a similar argument to that for theorem 2, replacing lemma 2 with lemma 3, we have the following result [2]:

Theorem 3 ((Buchbinder, Feldman, Naor, Schwartz 2012))**.** *Let $F(\cdot)$ be a submodular set function, and S_g be the set returned by the double greedy algorithm. Then*

$$\mathbb{E}[F(S_g)] \geq \frac{1}{2}F(S^*)$$

4.3 Continuous version

Alternatively, we can look at the continuous relaxation, by using the multi-linear extension as the cost function, to solve the problem. Instead of sets, we start with the vector of all “0"s an “1"s, and look at the gains with respect to the multi-linear extension of the set function. The algorithm is as follows:

1. Start with $x_0 = 0_{\mathcal{V}}$ and $y_0 = 1_{\mathcal{V}}$.
2. For $i \in \{1, \ldots, n\}$
   a) Calculate
   $$\alpha_i = [g(x_{i-1} + 1_{e_i}) - g(x_{i-1})]_+$$
   $$\beta_i = [g(y_{i-1} - 1_{e_i}) - g(y_{i-1})]_+$$
   b) $x_i \leftarrow x_{i-1} + \frac{\alpha_i}{\alpha_i+\beta_i} 1_{e_i}$
   c) $y_i \leftarrow y_{i-1} - \frac{\alpha_i}{\alpha_i+\beta_i} 1_{e_i}$
3. Return a random set $R(x_n)$, where $e_i \in R(X_n)$ with probability x_i.

Since the multi-linear extension is defined as exactly the expectation over the random sets for $R(x_n)$, we know that in expectation, we obtain a set of value $g(x)$.

5 Diffusion Models

As an example of an application in the data-mining community, we'll look at diffusion models, defined on graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$. In particular, we are interested in finding a small set S of most influential nodes in $\mathcal{G}$ with maximal influence, defined as

$F(S) =$ Expected number of active nodes after T steps, if cascades started at points in S

5.1 Linear Threshold Model (LTM)

In the linear threshold model, for each outgoing edge $(w, v) \in \mathcal{E}$ from a node $w \in \mathcal{V}$, we have an "influence value" of $b_{vw} \geq 0$ satisfying $\sum_{v:(w,v)\in\mathcal{E}} b_{vw} \leq 1$.

1. We start with an initial set of nodes $S_0 \subseteq \mathcal{V}$.
2. For $i = 1, \ldots, n-1$
   - Sample a threshold $\theta_v \sim U[0, 1]$ independently for each node $v \in \mathcal{V}$.
   - Each vertex $v \in \mathcal{V} \setminus \cup_{0\leq j\leq i-1} S_j$ is "activated" if

     $$\sum_{w \in \cup_{0\leq j\leq i-1} S_j} b_{vw} \geq \theta_v$$

     We define S_i to be the set of all vertices activated in iteration i.
   - The process stops if $S_i = \emptyset$.

Since the length of a path from any vertex in S_0 to any other vertex in the graph is at most $n-1$, the model terminates in at most $n-1$ iterations.

5.2 Independent Cascade Model (ICM)

In the linear threshold model, for each outgoing edge $(w, v) \in \mathcal{E}$ from a node $w \in \mathcal{V}$, we have a "transition probability" $p_{vw} \in [0, 1]$.

1. We start with an initial set of nodes $S_0 \subseteq \mathcal{V}$.
2. For $i = 1, \ldots, n-1$
   - For each vertex $w \in S_{i-1}$
     - For each outgoing edge $(w, v) \in \mathcal{E}$, where $v \in \mathcal{V} \setminus \cup_{0\leq j\leq i-1} S_j$, "activate" v with probability p_{vw}.

We define S_i to be the set of all vertices activated in iteration i.

- The process stops if $S_i = \emptyset$.

Since the length of a path from any vertex in S_0 to any other vertex in the graph is at most $n-1$, the model terminates in at most $n-1$ iterations.

5.3 Submodularity for Diffusion Models

Defining $\sigma(S_0) = \mathbb{E}[|S_{n-1}|]$, we can show that $\sigma(\cdot)$ is monotone and submodular for both the LTM and ICM diffusion models. The key idea is to define an equivalent process which generates a distribution $\mathbb{P}$ over graphs. Introducing

$$R(u, \mathcal{G}) = \{v \mid v \text{ is reachable from } u \text{ in } \mathcal{G}\}$$

we observe that $\sigma_{\mathcal{G}}(S) = |\cup_{u \in S} R(u, \mathcal{G})|$ is a coverage function, which is a monotone increasing submodular function. Therefore, as an expectation over coverage functions,

$$\sigma(\cdot) = \sum_{\mathcal{G}} \mathbb{P}(\mathcal{G}) \sigma_{\mathcal{G}}(\cdot)$$

is also monotone and submodular.

The equivalent process for both the LTM and ICM can be described as follows: We generate a graph $\tilde{\mathcal{G}}$ with the same set of vertices $\mathcal{V}$, and introduce edges to $\tilde{\mathcal{G}}$ where

- **LTM**: For each node $v \in \mathcal{V}$, we pick an incoming edge (v, w) with probability b_{vw}, and no edge with probability $1 - \sum_{w \in \mathcal{V}} b_{vw}$. Add all edges that were picked to $\mathcal{G}$.
- **ICM**: For each node v, pick one incoming activating edge (w, v) with probability p_{vw}.Add all edges that were picked to $\mathcal{G}$.

The full construction may be found in [3]. For a more general result, we refer the interested reader to [4].

References

[1] Uriel Feige, Vahab S Mirrokni, and Jan Vondrak. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011.

[2] Niv Buchbinder, Michael Feldman, Joseph Naor, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 649–658. IEEE, 2012.

[3] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. 2003.

[4] Elchanan Mossel and Sebastien Roch. Submodularity of influence in social networks: From local to global. *SIAM Journal on Computing*, 39(6):2176–2188, 2010.