Fast Semi-differential based Submodular function Optimization

Rishabh Iyer\textsuperscript{1}, Stefanie Jegelka\textsuperscript{2}, and Jeff Bilmes\textsuperscript{1}
\textsuperscript{1}University of Washington, Seattle; \textsuperscript{2}University of California, Berkeley

Overview
- A generic sub-gradient ascent [super-gradient descent] framework for submodular maximization [minimization].
- New theoretical results for submodular minimization.
- A novel view as a framework for submodular maximization.
- Empirical experimental validation.

Submodular Optimization in Machine Learning

Maximization: $A^* \in \text{argmax}_A f(A)$

Minimization: $A^* \in \text{argmin}_A f(A)$

Sensor Placement Summarization

MAP Inference Clustering

Convexity, Concavity & Submodular Semigradients

Subgradients:
- Akin to convexity.
- Denote a permutation $\pi_Y$:

Supergradients:
- Akin to concavity.
- Three specific supergradients:

Modular Lower bound:
$m_p(X) = f(Y) + g_r(X) - g_r(Y) \leq f(I)$

Modular Upper bound:
$m_p(X) = f(Y) + g_r(X) - g_r(Y) \leq f(I)$. (1)

Semigradient Descent Algorithmic Framework

Algorithm 1 Subgradient ascent [descent] algorithm for submodular maximization [minimization].

1. Start with an arbitrary $X^0$.
2. repeat
3. Pick a semigradient $h_Y[g_r]$, $\sum_{\gamma \in Y}$ at $X^t$.
4. $X^{t+1} \leftarrow \text{argmax}_{X \in \mathcal{X}} m_{p}(X)[X^{t+1} \leftarrow \text{argmin}_{X \in \mathcal{X}} m_{p}(X)]$
5. $t \leftarrow t + 1$
6. until we have converged ($X^{t+1} = X^t$) or $i \leq T$

Lemma 1
Algorithm 1 monotonically improves the objective function value at every iteration.

Define the curvature of a monotone submodular function $k_f$ as:

$$ k_f = 1 - \min_{j \in V} \frac{f(I \setminus j)}{f(I)} \quad (1) $$

Submodular Function Minimization (MMin)

- Use supergradients $\tilde{g}^*, \tilde{g}^*$ and $\tilde{g}$ in algorithm 1.

Unconstrained minimization ($C = 2^V$)

- Define $A = \{j : f(j) < 0\}$ and $B = \{j : f(j \setminus \{j\}) > 0\}$.
- Known that for every optimizer $X^* : A \subseteq X^* \subseteq V$.

Theorem 1: MMin-IIa and MMin-IIb obtain sets $X^* = A$ and $X^* = B$ respectively. Furthermore, in $O(t^2)$ oracle calls, MMin-I and II obtain sets $A$ and $B$, which are local minimizers of $f$ and satisfy $A \subseteq X^* \subseteq B$.

Empirical results tested on Concave over modular: $\sqrt{w_1(X)} + \lambda w_2(V(X))$ and Bipartite neighborhoods: $\sqrt{w_1(X)} + \lambda w_2(V(X))$.

Constrained minimization (monotone)

Theorem 2: The set $A_c$ obtained through MMin-I under constraints $\mathcal{C}$ obtains:

$$ f(A_c) \leq \left[ \frac{|X^*|}{1 + (|X^*| - 1)(1 - \kappa_f)} \right] f(X^*) \leq \frac{1}{1 - \kappa_f} f(X^*) $$

Greedy subgradient:

$$ \frac{1}{1 - e^{-\alpha \kappa_f}} \leq 1 - 1/e \text{ approximation!} $$

Algorithm 1 using the subgradient $h^*$ exactly corresponds to the greedy algorithm!

Similar analysis extends to Knapsack constraints.

Generality of Algorithm 1 for submodular Maximization

The subgradient algorithm for maximization is more general:

$$ \text{Theorem 3: For every } \alpha \text{-approximation algorithm, there exists a schedule of subgradients obtainable in poly-time, such that Algorithm 1 achieves an approximation factor of at least } \alpha. $$