Supplementary material for "Online Submodular Minimization for Combinatorial Structures"

1. Problems when applying the algorithms in (Kakade et al., 2009) to the submodular-cost setting

Kakade et al. (2009) show online approximation algorithms that use an offline approximation algorithm as a black box. Their method generalizes online gradient descent (Zinkevich, 2003) to use the approximation algorithm in an approximate projection. Their cost function is of the form $c : 2^E \times \mathbb{R}^d \to \mathbb{R}$, $c(S, w) = \langle \phi(S), w \rangle$ and must be *linear* in w. That means, it is the dot product between some feature vector of S and a weight vector. (In the paper, they leave nonlinear costs as an open problem.)

To use this framework, we must express any non-decreasing submodular f via a cost vector w^f as $c(S, w^f) = f(S)$. The set of non-decreasing submodular functions on E is equivalent to a convex cone in $\mathbb{R}^{2^{|E|}}$. This set has a non-empty relative interior (e.g., $f(S) = \log(1 + |S|)$). As a result, simple linear algebra shows that a full basis is needed to represent all such f meaning that w has an exponential dimension d. But then the regret bound in (Kakade et al., 2009) is exponential in $|E|$, since it is linear in $\|w\|$, i.e., proportional to $\sqrt{d}$. Whilst the norm issue can possibly be resolved, the algorithm also assumes that, given any $w \in \mathbb{R}^d$, we can project it onto the set of those w for which $c(\cdot, w)$ is a nondecreasing submodular function. Given the results in (**?**), this too seems to be non-trivial.

2. Rounding scheme for cuts

We consider the problem

$$\min f(S) \quad \text{s.t. } S \text{ is an } (s,t) \text{ cut.}$$

The corresponding convex program uses the same constraints as the linear program for minimum (s,t) cut, and introduces additional variables π for the nodes:

$$\begin{aligned} \min \quad & \tilde{f}(x) \\ \text{s.t.} \quad & x(e) \geq \pi(v) - \pi(u) \ \ \forall (u,v) \in E \\ & \pi(t) - \pi(s) \geq 1 \\ & \pi \in [0,1]^V, \ x \in [0,1]^E. \end{aligned} \tag{1}$$

The additional node variables π essentially indicate the membership of a node in the s side (label 0) or t side (label 1) of the cut. The constraints demand that an edge e from a label-zero node to a label-one node should be selected, that is, $x(e) = 1$. These edges will eventually make up the cut.

Let x^* be the optimal solution of Program (1). We test the values of $x^*(e)$ as rounding thresholds in decreasing order. If the set C_i of edges e with $x^*(e)$ greater than the threshold contains a cut, we stop and prune C_i to a minimal cut. Pruning is one minimum cut computation, where edges in $E \setminus C_i$ have infinite weight.

Algorithm 1 rounding procedure given x^*

```
order E such that x*(e_1) ≥ x*(e_2) ≥ ... ≥ x*(e_m)
for i = 1, ..., m do
  let C_i = {e_j | x*(e_j) ≥ x*(e_i)}
  if C_i is a cut then
    prune C_i to Ĉ and return Ĉ
  end if
end for
```

Lemma 6. *Let $\widehat{C}$ be the rounded solution returned by Algorithm 1, and C^* the optimal cut. Then $f(\widehat{C}) \leq |P_{\max}| f(C) \leq (n-1) f(C)$, where $P_{\max}$ is the longest simple path in the graph.*

Proof. Summing up the constraints on $x(e)$ in Program (1) over any (s,t) path shows that the sum of $x(e)$ along any path must be at least $\pi(t) - \pi(s) \geq 1$. That means, at least one edge from every path must be included in the cut. (In the relaxation, the weight x can be distributed along the path.) Thus, the above program is equivalent to the following program:

$$\begin{aligned} \min \quad & \tilde{f}(x) \\ \text{s.t.} \quad & \textstyle\sum_{e \in P} x(e) \geq 1 \ \ \forall (s,t)\text{-paths } P \\ & x \in [0,1]^E. \end{aligned} \tag{2}$$

Program (2) is a submodular covering program. Thus, thresholded rounding is possible analogous to other covering programs (Iwata & Nagano, 2009). Let θ be the rounding threshold that implied the final C_i. In the worst case, x^* is uniformly distributed along the longest path, and then θ must be $|P_{\max}|^{-1}$ to include at least one of the edges in $P_{\max}$. Since $\tilde{f}$ is nondecreasing like f and also positively homogeneous, it holds that

$$\begin{aligned} f(\widehat{C}) &\le f(C_i) = \tilde{f}(\chi_{C_i}) \\ &\le \tilde{f}(\theta^{-1}x^*) \le \theta^{-1}\tilde{f}(x^*) \le \theta^{-1}\tilde{f}(\chi_{C^*}) = \theta^{-1}f(C^*). \end{aligned}$$

The first inequality follows from monotonicity of f and the fact that $\widehat{C} \subseteq C_i$. Similarly, the relation between $\tilde{f}(\chi_{C_i})$ and $\tilde{f}(\theta^{-1}x^*)$ holds because $\tilde{f}$ is nondecreasing: by construction, $x^*(e) \ge \theta\chi_{C_i}(e)$ for all $e \in E$, and hence $\chi_{C_i}(e) \le \theta^{-1}x^*(e)$. Finally, we use the optimality of x^* to relate the cost to $f(C^*)$ (χ_{C^*} is also feasible, but x^* optimal). The lemma follows since $\theta^{-1} \le |P_{\max}|$. □

3. Detailed proof of Theorem 2

First, we re-state the theorem.

Theorem 2. *For an approximation $\hat{f}$ that satisfies (C1) and (C2), $M = \max_t f_t(E)$, and $\eta = T^{-1/2}$, Algorithm 2 achieves an expected α-regret $\mathbb{E}[R_\alpha(T)] \le 3\alpha m M/\sqrt{T} = O(\alpha m/\sqrt{T})$.*

Proof. Let

$$S_t = \operatorname*{argmin}_{S\in\mathcal{S}} \sum\nolimits_{\tau=1}^{t-1} \hat{f}_\tau(S) + \alpha r(S);$$

$$\widehat{S}_t = \operatorname*{argmin}_{S\in\mathcal{S}} \sum\nolimits_{\tau=1}^{t-1} \hat{f}_\tau(S); \quad S_t^* = \operatorname*{argmin}_{S\in\mathcal{S}} \sum\nolimits_{\tau=1}^{t} f_\tau(S).$$

First, we show a relation for $\sum_{t=1}^T \hat{f}_t(S_{t+1})$ and later relate it to the actual cost $\sum_{t=1}^T \hat{f}_t(S_t)$. The first inequality is

$$\sum\nolimits_{t=1}^T \hat{f}_t(\widehat{S}_{t+1}) \le \sum\nolimits_{t=1}^T \hat{f}_t(\widehat{S}_{T+1}). \tag{3}$$

It holds trivially for $T = 1$. The case $T+1$ follows by induction and the optimality of $\widehat{S}_{T+1}$:

$$\begin{aligned} \sum\nolimits_{t=1}^{T+1} \hat{f}_t(\widehat{S}_{t+1}) &\le \sum\nolimits_{t=1}^{T} \hat{f}_t(\widehat{S}_{T+1}) + \hat{f}_{T+1}(\widehat{S}_{T+2}) \\ &\le \sum\nolimits_{t=1}^{T} \hat{f}_t(\widehat{S}_{T+2}) + \hat{f}_{T+1}(\widehat{S}_{T+2}) \\ &= \sum\nolimits_{t=1}^{T+1} \hat{f}_t(\widehat{S}_{T+2}). \end{aligned}$$

We now replace $\hat{f}_1$ in Equation (3) by $\hat{f}_1 + \alpha r$:

$$\begin{aligned} \sum\nolimits_{t=1}^T \hat{f}_t(S_{t+1}) + \alpha r(S_1) &\le \sum\nolimits_{t=1}^T \hat{f}_t(S_{T+1}) + \alpha r(S_{T+1}) \\ &\le \sum\nolimits_{t=1}^T \hat{f}_t(\widehat{S}_{T+1}) + \alpha r(\widehat{S}_{T+1}). \end{aligned}$$

Rearranging the terms yields

$$\sum\nolimits_{t=1}^T \hat{f}_t(S_{t+1}) \le \sum\nolimits_{t=1}^T \hat{f}_t(\widehat{S}_{T+1}) + \alpha(r(\widehat{S}_{T+1}) - r(S_1)). \tag{4}$$

To transfer this result to the series of S_t, we use that $\hat{f}_t(S_t) \le \hat{f}_t(S_{t+1}) + (\hat{f}_t(S_t) - \hat{f}_t(S_{t+1}))$:

$$\begin{aligned} \sum_{t=1}^T \hat{f}_t(S_t) &\le \sum_{t=1}^T \hat{f}_t(\widehat{S}_{T+1}) \\ &+ \sum_{t=1}^T (\hat{f}_t(S_t) - \hat{f}_t(S_{t+1})) + \alpha(r(\widehat{S}_{T+1}) - r(S_1)). \end{aligned} \tag{5}$$

Condition (C1) implies that

$$\sum\nolimits_{t=1}^T \hat{f}_t(\widehat{S}_{T+1}) \le \sum\nolimits_{t=1}^T \hat{f}_t(S_T^*) \le \alpha \sum\nolimits_{t=1}^T f_t(S_T^*),$$

and that $\sum_{t=1}^T f_t(S_t) \le \sum_{t=1}^T \hat{f}_t(S_t)$. Together with Equation (5), this yields

$$\begin{aligned} &\sum\nolimits_{t=1}^T f_t(S_t) - \alpha \sum\nolimits_{t=1}^T f_t(S_T^*) \\ &\quad\le \sum\nolimits_{t=1}^T (\hat{f}_t(S_t) - \hat{f}_t(S_{t+1})) + \alpha(r(\widehat{S}_{T+1}) - r(S_1)). \end{aligned} \tag{6}$$

It remains to bound the two terms on the right hand side, and these bounds depend on $r \in [0, M/\eta]^E$.

We first address the random perturbation r in $[0, M/\eta]^E$. The last term can be bounded as

$$\alpha\mathbb{E}[r(\widehat{S}_{T+1}) - r(S_1)] \le \alpha m M/\eta. \tag{7}$$

To bound the expected sum of differences of the function values, we use a technique by Hazan & Kale (2009). For the analysis, one can assume that r is resampled in each round. We first bound $P(S_t \neq S_{t+1})$. A simple union bound holds:

$$\begin{aligned} P(S_t \neq S_{t+1}) &\le \sum\nolimits_{i=1}^m P(e_i \in S_t \text{ and } e_i \notin S_{t+1}) \\ &+ \sum\nolimits_{i=1}^m P(e_i \notin S_t \text{ and } e_i \in S_{t+1}). \end{aligned} \tag{8}$$

To bound the right hand side, we fix i and look at $P(e_i \in S_t \text{ and } e_i \notin S_{t+1})$. Denote the components of r by r_j and define $r' : 2^E \to \mathbb{R}$ as $r'(S) = \sum_{e_j \in S, j \neq i} r_j$, so $r'(e_j) = r(e_j) = r_j$ for all $j \neq i$, but $r'(e_i) = 0$; and define $\Phi'_t : 2^E \to \mathbb{R}$ as $\Phi'_t(S) = \sum_{\tau=1}^{t-1} \hat{f}_\tau + \alpha r'(S)$. Now let

$$S^1 = \operatorname*{argmin}_{S\in\mathcal{S}, e_i \in S} \Phi'_t(S); \qquad S^2 = \operatorname*{argmin}_{S\in\mathcal{S}, e_i \notin S} \Phi'_t(S).$$

The event $e_i \in S_t$ only happens if $\Phi'_t(S^1) + \alpha r_i < \Phi'_t(S^2)$ and $S_t = S^1$. On the other hand, to have $e_i \notin S_{t+1}$, it must be that $\Phi'_t(S^1)+\alpha r_i \geq \Phi'_t(S^2)-\alpha M$, since otherwise

$$\begin{aligned}\sum\nolimits_{\tau=1}^{t+1} \hat{f}_t(S^1) + \alpha r(S^1) &= \Phi'_t(S^1) + \alpha r_i + \hat{f}_t(S^1)\\ &< \Phi'_t(S^2)\\ &< \Phi'_t(B) + \hat{f}_t(B)\end{aligned}$$

for all $B \in \mathcal{S}$ with $e_i \notin B$. Here, we used that $\hat{f}_t(S) \leq \alpha f_t(S) \leq \alpha M$ for all $S \subseteq E$. Let $v = \alpha^{-1}(\Phi'(S^2) - \Phi'(S^1))$, then $e_i \in S_t$ and $e_i \notin S_{t+1}$ only if $r_i \in [v - M, v]$. The number r_i is in this range with probability at most η since it is chosen uniformly at random from $[0, M/\eta]$, so $P(e_i \in S_t \text{ and } e_i \notin S_{t+1}) \leq \eta$. The bound on $P(e_i \notin S_t \text{ and } e_i \in S_{t+1})$ follows by an analogous argumentation. Together, those results bound (8):

$$\begin{aligned}P(S_t \neq S_{t+1}) &\leq \sum\nolimits_{i=1}^{m} P(e_i \in S_t \text{ and } e_i \notin S_{t+1})\\ &\quad + \sum\nolimits_{i=1}^{m} P(e_i \notin S_t \text{ and } e_i \in S_{t+1})\\ &\leq 2m\eta. \end{aligned} \tag{9}$$

Equation 9 helps to bound the sum of function values, using $\hat{f}(C) \leq \alpha M$ for all C:

$$\begin{aligned}&\sum\nolimits_{t=1}^{T} \mathbb{E}\big[\hat{f}_t(S_t) - \hat{f}_t(S_{t+1})\big]\\ &\leq \sum\nolimits_{t=1}^{T} P(S_t \neq S_{t+1}) \max_{B\in\mathcal{S}} \hat{f}(B)\\ &\leq 2\alpha m M T\eta. \end{aligned} \tag{10}$$

Combining Inequalities (6), (7) and (10) results in

$$\begin{aligned}\mathbb{E}\big[\sum\nolimits_{t=1}^{T} f_t(S_t)\big] - \alpha \sum\nolimits_{t=1}^{T} f_t(S^*_T)&\\ \leq \alpha M m/\eta + 2\alpha m M T\eta.&\end{aligned}$$

The final regret bound follows for $\eta = T^{-1/2}$. □

4. Proof of Lemma 3

Lemma 3. *Let $\hat{f}$ be either $\hat{f}_h$ or $\hat{f}_t$, with equal probabilities. Then $f(S) \leq \mathbb{E}[\hat{f}(S)] \leq (|V|/2)f(S)$ for all minimal (s,t)-cuts S.*

Proof. First, we bound $\hat{f}_h(S)$. Let $\Delta_t(S)$ be the set of head nodes of edges in S, i.e., at most all nodes on the t side of the cut.

$$\begin{aligned}\hat{f}_h(S) &= \sum\nolimits_{v\in\Delta_t(S)} f(S \cap E^h_v)\\ &\leq |\Delta_t(S)| \max\nolimits_{v\in\Delta_t(S)} f(S \cap E^h_v)\\ &\leq |\Delta_t(S)| f(S).\end{aligned}$$

Analogously, it follows that $\hat{f}_t(S) \leq |\Delta_s(S)| f(S)$, $|\Delta_s(S)|$ being the number of tail nodes of edges in S. We combine these bounds to

$$\begin{aligned}\mathbb{E}[\hat{f}(S)] &= (\hat{f}_h(S) + \hat{f}_t(S))/2\\ &\leq f(S)(|\Delta_s(S)| + |\Delta_t(S)|)/2\\ &\leq f(S)|V|/2.\end{aligned}$$

□

5. Proof of Lemma 4

Let $S^* = \operatorname{argmin}_{S\in\mathcal{S}} \sum_t f_t(S)$, and $\hat{S}^* = \operatorname{argmin}_{S\in\mathcal{S}} \sum_t \hat{f}^2_t(S)$. We play S_t as prescribed by the algorithm $\mathcal{A}$.

Lemma 4. *Let $\widehat{R}_{\mathcal{A}}$ be the regret of an online algorithm $\mathcal{A}$ for cost functions $\hat{f}^2_t$. Using $\mathcal{A}$ with $\hat{f}^2_t$ when observing f_t leads to an α_g regret of $R_{\alpha_g}(T) \leq \widehat{R}_{\mathcal{A}}\alpha_g/\nu$.*

Proof. Since we use $\hat{f}^2_t$ in $\mathcal{A}$, the regret $\widehat{R}_{\mathcal{A}}$ bounds $\sum_t(\hat{f}^2_t(S_t) - \hat{f}^2_t(\widehat{S}^*))$. Therefore, we relate the actual regret, $\sum_t(f_t(S_t) - \alpha_g f_t(S^*))$, to the regret of $\mathcal{A}$. We use that $\hat{f}^2(S) \leq f^2(S) \leq \alpha_g^2 \hat{f}^2(S)$. We have that

$$\begin{aligned}\sum\nolimits_t (f_t(S_t) - \alpha_g f_t(S^*)) &= \sum\nolimits_t \frac{(f^2_t(S_t) - \alpha^2_g f^2_t(S^*))}{(f_t(S_t) + \alpha_g f_t(S^*))}\\ &\leq \sum\nolimits_t (f^2_t(S_t) - \alpha^2_g f^2_t(S^*))/(\alpha_g\nu)\\ &\leq \sum\nolimits_t \alpha^2_g(\hat{f}^2_t(S_t) - \hat{f}^2_t(S^*))/(\alpha_g\nu)\\ &\leq \sum\nolimits_t \alpha_g(\hat{f}^2_t(S_t) - \hat{f}^2_t(\widehat{S}^*))/(\nu)\\ &= \alpha_g \sum\nolimits_t \widehat{R}_{\mathcal{A}}/\nu,\end{aligned}$$

since $\hat{S}^* = \operatorname{argmin}_{S\in\mathcal{S}} \sum_t \hat{f}^2_t(S)$ is optimal for $\hat{f}^2$. □

6. Multiple labels for label costs in Algorithm 3

Here, we will outline how to simulate label costs when one edge can have more than one label. This simulation applies to the spanning tree example.

Let k be the maximum number of labels any edge can have. We assign k "slots" to each edge. Each label $\ell \in \pi(e)$ occupies $1 \leq \gamma_e(\ell) \leq k$ slots, such that $\sum_{\ell\in\pi(e)} \gamma_e(\ell) = k$. Define k copies $G_i = (V, E_i)$ of G. Edge e is contained in $E_i(L)$ if i of its slots are filled by labels in L. Then we use

$$g(L) = \sum\nolimits_{i=1}^{k} r(E_i(L)).$$

This sum is still submodular, and maximum only if $E(L)$ contains a tree of full edges. The approximation factor increases moderately to $O(\log(nk))$.

7. Lower bound for submodular minimum (s,t) cut

We prove a lower bound for the special case of Problem (1) in the main paper that $\mathcal{S}$ is the set of all (s,t)-cuts for given nodes s,t in a given graph. The ground set E is the set of edges.

$$\min f(S) \quad \text{s.t. } S \text{ is an } (s,t) \text{ cut} \tag{11}$$

The lower bound is information theoretic and assumes oracle access to the cost function.

Theorem 7. *No polynomial-time algorithm can solve Problem (11) with an approximation factor better than* $o(\sqrt{|E|/\log|E|})$.

The main idea of the proof is to construct two submodular cost functions f, h with different minima that are almost indistinguishable. In fact, with high probability they cannot be discriminated with a polynomial number of function queries. If the optima of h and f differ by a factor larger than α, then any solution for f within a factor α of the optimum would be enough evidence to discriminate f and h. As a result, a polynomial-time algorithm that guarantees an approximation factor α would lead to a contradiction. The proof technique is similar to that in (Goemans et al., 2009).

One of the functions, f, depends on a hidden random set $R \subseteq E$ that will be its optimal cut. We will use the following Lemma that assumes f to depend on a random set R.

Lemma 8 ((?), Lemma 2.1)**.** *If for any set $Q \subseteq E$, chosen without knowledge of R, the probability of $f(Q) \neq h(Q)$ over the random R is $m^{-\omega(1)}$, then any algorithm that makes a polynomial number of oracle queries has probability at most $m^{-\omega(1)}$ of distinguishing f and h.*

The Lemma holds by a union bound and computation path argument.

Proof. Construct a graph $G = (V,E)$ with ℓ parallel disjoint paths from s to t; each path has k edges. Let the random set $R \subset E$ be a cut consisting of $|R| = \ell$ edges. The cut contains one edge from each path uniformly at random. We define $\beta = 8\ell/k < \ell$ (for $k > 8$), and, for any $Q \subseteq E$,

$$h(Q) = \min\{|Q|, \ell\} \tag{12}$$

$$f(Q) = \min\{|Q \setminus R| + \min\{|Q \cap R|,\ \beta\},\ \ell\}. \tag{13}$$

The functions differ only for the relatively few sets Q with $|Q \cap R| > \beta$ and $|Q \setminus R| < \ell - \beta$. Define ε such that $\varepsilon^2 = \omega(\log m)$, and set $k = 8\sqrt{m}/\varepsilon$ and $\ell = \varepsilon\sqrt{m}$.

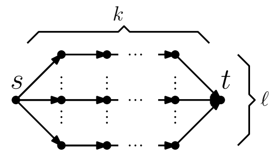

Figure 1. Graph for the proof of Theorem 7.

We compute the probability that f and h differ for a given query set Q. Probabilities are over the unknown R. Since $f \leq h$, the probability of difference is $P(f(Q) < h(Q))$. If $|Q| \leq \ell$, then $f(Q) < h(Q)$ only if $\beta < |Q \cap R|$, and the probability $P(f(Q) < h(Q)) = P(|Q \cap R| > \beta)$ increases as Q grows. If, on the other hand, $|Q| \geq \ell$, then the probability

$$P(f(Q) < h(Q)) = P(|Q \setminus R| + \min\{|Q \cap R|,\ \beta\} < \ell)$$

decreases as Q grows. Hence, the probability of difference is largest when $|Q| = \ell$.

So let $|Q| = \ell$. If Q spreads over $b \leq k$ edges of a path P, then the probability that Q includes the edge in $P \cap R$ is b/k. The expected overlap is the sum of hits on all paths, $\mathbb{E}[\ |Q \cap R|\ ] = |Q|/k = \ell/k$. Since the edges in R are independent across different paths, we bound the probability of a large intersection by a Chernoff bound, and Lemma 8 holds:

$$\begin{aligned} P\big(f(C) \neq h(C)\big) &\leq P\big(|C \cap R| \geq 8\ell/k\ \big) \\ &\leq 2^{-8\ell/k} = 2^{-\varepsilon^2} = 2^{-\omega(\log m)} = m^{-\omega(1)}. \end{aligned}$$

With this result, Lemma 8 applies. No polynomial-time algorithm can guarantee to distinguish f and h with high probability. A polynomial algorithm with approximation factor better than the ratio of optima $h(R)/f(R)$ would discriminate the two functions and thus lead to a contradiction. As a result, the lower bound will be the ratio of optima of h and f. The optimum of f is $f(R) = \beta$, and h has uniform cost ℓ for all minimal cuts. Hence, the ratio is $h(R)/f(R) = \ell/\beta = \sqrt{m}/\varepsilon = o(\sqrt{m/\log m})$.

For contradiction, assume there was an algorithm with approximation factor $\alpha = o(\sqrt{m/\log m})$. Set $\varepsilon = \sqrt{m}/(2\alpha)$, so $\varepsilon^2 = \omega(\log m)$ is satisfied. Given f for this ε, the algorithm would return a solution with cost at most $\alpha f(R) = \alpha\varepsilon^2 \leq \varepsilon\sqrt{m}/2 < \varepsilon\sqrt{m}$. For h, it can only return a solution with strictly larger cost $\ell = \varepsilon\sqrt{m}$ and could thus distinguish f and h, contradicting Lemma 8. $\square$

References

Goemans, M. X., Harvey, N. J. A., Iwata, A., and Mirrokni, V. S. Approximating submodular functions everywhere. In *Proc. of the ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2009.

Hazan, E. and Kale, S. Online submodular minimization. In *Proc. of the Ann. Conf. on Neural Info. Processing Systems (NIPS)*, 2009.

Iwata, S. and Nagano, K. Submodular function minimization under covering constraints. In *Proc. of the Ann. Symp. on Foundations of Comp. Science (FOCS)*, 2009.

Kakade, S., Kalai, A. T., and Ligett, K. Playing games with approximation algorithms. *SIAM Journal on Computing*, 39(3):1088–1106, 2009.

Zinkevich, M. Online convex programming and infinitesimal gradient ascent. In *Proc. of the Int. Conf. on Machine Learning (ICML)*, 2003.