A Primal-Dual Algorithm for Higher-Order Multilabel Markov Random Fields –Supplementary Material

1. Submodular Upper Bound

Recall in the PRE-EDIT-DUALS of our SoSPD algorithm, when the given clique function $f_C(S)$ is not submodular, we must find a submodular upper bound $\widetilde{f}_C(S)$ such that 1) $\widetilde{f}_C(S) \geq f_C(S), \forall S \subseteq C$, 2) $\widetilde{f}_C(\emptyset) = f_C(\emptyset)$, 3) $\widetilde{f}_C(C) = f_C(C)$, 4) $\widetilde{f}_C(\{i\}) \leq \max_S f_C(S), \forall i \in C$.

There is another equivalent definition of submodular function that $f : 2^{|V|} \mapsto \Re$ is submodular if and only if $f(S) + f(S \cup \{i,j\}) \leq f(S \cup \{i\}) + f(S \cup \{j\})$ for $\forall S \subseteq V, \forall i,j \in V \backslash S, i \neq j$. We will use this property to check whether a given function is submodular and restore the submodularity if it's not.

We will use tuple (S,i,j) to represent the constraint $\widetilde{f}_C(S) + \widetilde{f}_C(S \cup \{i,j\}) \leq \widetilde{f}_C(S \cup \{i\}) + \widetilde{f}_C(S \cup \{j\})$ thereafter. Meanwhile, we call this constraint (S,i,j) at level $|S|$. For a particular function value $f_C(S)$, we call it level $|S|$ entry.

Given a constraint (S,i,j),we can define the violation value $\delta(S,i,j) = \max\{0, \widetilde{f}_C(S) + \widetilde{f}_C(S \cup \{i,j\}) - \widetilde{f}_C(S \cup \{i\}) - \widetilde{f}_C(S \cup \{j\})\}$. The intuition to restore submodularity for this constraint is to raise its two level $|S|+1$ entries $\widetilde{f}_C(S \cup \{i\})$ and $\widetilde{f}_C(S \cup \{j\})$ by $\delta(S,i,j)/2$. However, these increments may break other constraints at level $|S|+1$ and $|S|-1$ involving $\widetilde{f}_C(S \cup \{i\})$ or $\widetilde{f}_C(S \cup \{j\})$[1]. What's worse, we cannot hope to fix all the violations in a single traversal of all (S,i,j) tuple[2]. Fortunately, when the energy is integral and we try to fix the constraint level by level, this procedure can terminate in finite number of iterations.

Lemma 1. *Algorithm 1 can terminate in finite number of iterations when the input f_C is integral. The output $\widetilde{f}_C$ satisfies: 1) $\widetilde{f}_C$ is submodular and $\widetilde{f}_C \geq f_C$, 2) $\widetilde{f}_C(\emptyset) = f_C(\emptyset)$, 3) $\widetilde{f}_C(C) = f_C(C)$.*

[1] E.g., $\widetilde{f}_C(S \cup \{i\}) + \widetilde{f}_C(S \cup \{i,j,k\}) \leq \widetilde{f}_C(S \cup \{i,j\}) + \widetilde{f}_C(S \cup \{i,k\})$ for $k \in V \backslash \{i,j\}$ might be violated due to the increment of $\widetilde{f}_C(S \cup \{i\})$.

[2] In the previous example, when we fix the violation of $\widetilde{f}_C(S \cup \{i\}) + \widetilde{f}_C(S \cup \{i,j,k\}) \leq \widetilde{f}_C(S \cup \{i,j\}) + \widetilde{f}_C(S \cup \{i,k\})$, the increment of $\widetilde{f}_C(S \cup \{i,j\})$ may cause $\widetilde{f}_C(S) + \widetilde{f}_C(S \cup \{i,j\}) \leq \widetilde{f}_C(S \cup \{i\}) + \widetilde{f}_C(S \cup \{j\})$ fail again.

Algorithm 1 Submodular Upper Bound

1: $\widetilde{f}_C(S) \leftarrow f_C(S), \forall S \subseteq C$
2: **while** $\widetilde{f}_C$ is not submodular **do**
3: $\quad \psi(S) \leftarrow 0, \forall S \subseteq C$
4: $\quad$ **for** $k = |C| - 2$ **to** 0 **do**
5: $\quad\quad$ **for all** constraint (S,i,j) s.t. $|S| = k$ **do**
6: $\quad\quad\quad \delta(S,i,j) = \max\{0, \widetilde{f}_C(S) + \widetilde{f}_C(S \cup \{i,j\}) - \widetilde{f}_C(S \cup \{i\}) - \widetilde{f}_C(S \cup \{j\})\}$
7: $\quad\quad\quad \psi(S \cup \{i\}) \leftarrow \max\{\psi(S \cup \{i\}), \lceil \delta(S,i,j)/2 \rceil\}$
8: $\quad\quad\quad \psi(S \cup \{j\}) \leftarrow \max\{\psi(S \cup \{j\}), \lceil \delta(S,i,j)/2 \rceil\}$
9: $\quad\quad$ **end for**
10: $\quad\quad \widetilde{f}_C(S') \leftarrow \widetilde{f}_C(S') + \psi(S'), \forall S' \subseteq C, |S'| = k+1$
11: $\quad$ **end for**
12: **end while**
13: **return** $\widetilde{f}_C$

Proof. Suppose the algorithm terminates, it clear that $\widetilde{f}_C$ is a submodular function. Since we never decrease the entry of $\widetilde{f}_C$, we have $\widetilde{f}_C \geq f_C$. Furthermore, we only update the entries between level 1 and $|C| - 1$ (line 10), i.e., we never update the value $\widetilde{f}_C(\emptyset)$ and $\widetilde{f}_C(C)$, so we have $\widetilde{f}_C(\emptyset) = f_C(\emptyset)$ and $\widetilde{f}_C(C) = f_C(C)$.

The strategy to prove termination is by defining $\Delta_k = \max_{|S|=k} \delta(S,i,j)$ as the maximum violation among all constraints at level k. We will argue $\sum_{k=0}^{|C|-2} \Delta_k$ decreases by a fixed fraction in each iteration. Therefore, when the given function f_C is integral (i.e., $\widetilde{f}_C$, δ and Δ are also integral), $\sum_{k=0}^{|C|-2} \Delta_k$ will drop to 0 after finite number of iterations. At that point, all the constrains are satisfied and $\widetilde{f}_C$ is submodular.

Clearly, due to the our settings of $\psi(S')$ on line 7 and 8, after updating the level $k+1$ entries on line 10, we will satisfy all the constraints at level k, i.e., $\Delta_k = 0$ at this point. However, just as we pointed out before, the increments of level $k+1$ entries may caused the constraints at level $k+1$ and $k-1$ violated, i.e., increasing Δ_{k+1} and Δ_{k-1}. At level k, the maximum increment ψ at level k is $\Delta_k/2$, so it may cause Δ_{k+1} and Δ_{k-1} increase by at most $\Delta_k/2$ respectively.

Now, let's use Δ_k^t to represent Δ_k before the t-th iteration of the outer while loop.

When $k = |C| - 2$, after we update $k+1$ level entries, $\Delta_{|C|-3} \leq \Delta^t_{|C|-3} + \Delta^t_{|C|-2}/2$. Note we always push part of the violation values down to the low level similarly, i.e., $\Delta_{k-1} \leq \Delta^t_{k-1} + \Delta_k/2$. So by induction, we can show when $0 \leq k < |C| - 2$, $\Delta_k \leq \sum_{i=k}^{|C|-2} \Delta^t_k / 2^{i-k}$ after we update the $k+1$ level entries. In addition, before we update the level k entries, we know $\Delta_{k+1} = 0$. So after we update them, we have $\Delta_{k+1} \leq \Delta_k/2 \leq \sum_{i=k}^{|C|-2} \Delta^t_k / 2^{i-k+1}$, which will not be changed later in the same iteration. Furthermore, we have $\Delta_0 = 0$ at the end of each iteration. Adding them up, we have $\sum_{k=0}^{|C|-2} \Delta_k \leq (1 - \frac{1}{2^{|C|-2}}) \sum_{k=0}^{|C|-2} \Delta^t_k$. This is what we desired, $\sum_{k=0}^{|C|-2} \Delta_k$ decreases by a fixed fraction in each iteration. So when the given f_C is integral, it will finally reach 0 after $O(2^{|C|} \cdot |C| \cdot \log_2 U)$ iterations, where $U = \max_S f_C(S)$. $\square$

Despite the poor asymptotic bound of the number of iterations, in practice Algorithm 1 can terminate pretty fast on real data. However, we need to point out Algorithm 1 cannot guarantee $\widetilde{f}_C(\{i\}) \leq \max_S f_C(S), \forall i \in C$. It rarely happens in our experiment. In that case, we can use another trivial constant upper bound $\widetilde{f}_C = \max_S f_C(S)$ instead.