Constrained Interacting Submodular Groupings

Andrew Cotter [1] **Mahdi Milani Fard** [1] **Seungil You** [2] **Maya Gupta** [1] **Jeff Bilmes** [3]

Abstract

We introduce the problem of grouping a finite set V into m blocks where each block is a subset of V and where: (i) the blocks are individually highly valued by a submodular function f (both robustly and in the average case) while satisfying block-specific matroid constraints; and (ii) block scores interact where blocks are *jointly* scored highly via f, thus making the blocks mutually non-redundant. Submodular functions are good models of *information* and *diversity*; thus, the above can be seen as grouping V into matroid constrained blocks that are both intra- and inter-diverse. Potential applications include forming ensembles of classification/regression models, partitioning data for parallel processing, and summarization. In the non-robust case, we reduce the problem to non-monotone submodular maximization subject to multiple matroid constraints. In the mixed robust/average case, we offer a bicriterion guarantee for a polynomial time deterministic algorithm and a probabilistic guarantee for randomized algorithm, as long as the involved submodular functions (including the inter-block interaction terms) are monotone. We close with a case study in which we use these algorithms to find high quality diverse ensembles of classifiers, showing good results.

1. Introduction

In recent years, submodular functions (Fujishige, 2005) have been used to address an increasingly wide variety of problems in machine learning and artificial intelligence. This includes energy functions in probabilistic models (Kohli et al., 2007; Gotovos et al., 2015; Djolonga et al., 2016), influence in social network (Kempe et al., 2003; Mossel & Roch, 2007), crowd teaching (Singla et al., 2014), non-parametric Bayesian estimation (Reed & Ghahramani, 2013), document and speech summarization (Lin et al., 2009; Lin & Bilmes, 2011; Li et al., 2012), image summarization (Tschiatschek et al., 2014; Singla et al., 2016), and clustering (Narasimhan et al., 2005).

In this paper, we introduce and apply a new submodular optimization problem related to partitioning, covering, and packing (Schrijver, 2003). Given a set function $f : 2^V \to \mathbb{R}_+$, it may be *normalized* (i.e., $f(\emptyset) = 0$), *monotone non-decreasing* (i.e., $f(A) \leq f(B)$ whenever $A \subseteq B$), and/or *submodular* (i.e., $\forall A, B \subseteq V$, $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$). A function g is said to be *supermodular* if $-g$ is submodular. A function m is *modular* if it is both submodular and supermodular. An *m-partition* of V is a set of m subsets, called *blocks*, that are non-intersecting ($A_i^\pi \cap A_j^\pi = \emptyset$ for all $i \neq j$) and *covering* ($\cup_i A_i^\pi = V$). An *m-covering* is a set of blocks that is required only to be covering. In an m-covering, we might also have a multiplicity constraint which is expressed as a positive integer valued vector $\mathbf{k} = (k_v : v \in V)$ where $k_v \in \mathbb{Z}_+$. To be a $(\mathbf{k}, m)$-covering, we must have an m-covering with no multiplicity violations, i.e., $|\{i \in [m] : v \in A_i^\pi\}| \leq k_v, \forall v \in V$. A *packing* is a set of blocks that is required only to be non-intersecting. When we wish to refer collectively either to a partition, a covering, or a packing, we use the term *grouping*.

Given a finite set V of size $n = |V|$, a non-negative integer $m \in [1, n]$, and m monotone non-decreasing submodular functions $f_i : 2^V \to \mathbb{R}_+$ for $i \in [m]$, the problem we study finds a feasible m-partition, or m-covering, or m-packing π of V into m blocks $A_1^\pi, A_2^\pi, \ldots, A_m^\pi$ that are "good" in a way to be described below.

Feasibility of our groupings is expressed using matroids, which are powerful combinatorial objects that can express many useful constraints over sets. A matroid (Oxley, 2006) $\mathcal{M} = (V, \mathcal{I})$ consists of a finite countable set V and a non-empty set of "independent" subsets $\mathcal{I} = \{I_1, I_2, \ldots\}$, where $I_i \subseteq V$, that is down-closed ($A \subseteq B \in \mathcal{I} \Rightarrow A \in \mathcal{I}$) and where all maximally independent sets have the same size (i.e., $\forall A, B \in \mathcal{I}$ with $|A| < |B|$, $\exists v \in B \setminus A$ having $A + v \in \mathcal{I}$).

For a feasible grouping to be good, it must have several properties. First, the blocks in the grouping should both be individually highly diverse and also all be highly diverse

[1] Google AI [2] Kakao Mobility [3] University of Washington, Seattle, work done while at Google AI. Correspondence to: Andrew Cotter <acotter@google.com>, Jeff Bilmes <bilmes@uw.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

on average, where diversity is measured by the functions $\{f_i\}_{i\in[m]}$. For example, suppose the elements of the ground set include animal names: "goldfish" and "carp" have a similar meaning, as do "crow" and "raven". But fish are very different from birds, so each of the sets $\{\text{goldfish}, \text{crow}\}$ and $\{\text{carp}, \text{raven}\}$ are diverse. Second, and more importantly, different blocks should not be redundant w.r.t. each other. This avoids the case where two different blocks convey the same information but in different ways, something that might happen even if the two blocks are disjoint. For example, the sets $A_1 = \{\text{goldfish}, \text{crow}\}$ and $A_2 = \{\text{carp}, \text{raven}\}$ are similar and hence redundant, despite being disjoint, and thus would be undesirable blocks when chosen together. This distinction between disjointedness and non-redundancy is particularly relevant in the context of submodular scoring functions where, as measured by a submodular function f, redundancy between A_1 and A_2 would mean that $f(A_1 \cup A_2) \approx f(A_1)$ and/or $f(A_1 \cup A_2) \approx f(A_2)$. We show below that this idea has a number of natural applications, one of which we evaluate in our case study section.

The paper is organized as follows. In the remainder of this section, we formally define our contributions (Section 1.1) and outline their utility in practice (Section 1.2). This section also defines objectives that, when constrainedly optimized, achieve our stated grouping goals. Section 2 places this paper in the context of previous work, and demonstrates that our contributions are novel. Section 3 formally outlines our approach and Section 4 details how we achieve cross-block diversity. Section 5 provides algorithms for constrainedly optimizing our objectives. Notably, we show a bi-criterion multiplicative approximation ratio guarantee for a fast polynomial time deterministic algorithm (Theorem 1), and also provide a randomized version of the algorithm giving a guarantee with high probability (using Lemma 4). The guarantees themselves are rather complex, and they are best appreciated in context, so we refer the reader to Theorem 1 and Lemma 4 for their statement. Lastly, Section 6 explores a case study application where we show our approach can be used to produce ensembles of machine learning models. We demonstrate that our approach improves on previous state-of-the-art results and moreover the groupings achieve the aforementioned desired properties.

1.1. Contributions and Objectives

Our starting point is a recently introduced objective (Wei et al., 2015) that takes a convex combination of a robust and an average objective and finds a grouping π that scores highly w.r.t.:

$$F_{\text{ra}}(\pi) = \lambda_1 \min_{i\in[m]} f_i(A_i^\pi) + \frac{\lambda_2}{m} \sum_{i\in[m]} f_i(A_i^\pi), \qquad (1)$$

where the λ_is are non-negative coefficients and the f_is may be distinct submodular functions. Our **first contribution** is that, unlike Wei et al. (2015), which only handles partitions, we also handle coverings and packings. Our **second contribution** is the inclusion of more general block-specific constraints, expressed as intersections of matroids on an expanded ground set. For example, we may wish for blocks to not exceed a certain size, or for each block to correspond to a sub-tree of some graph (Section 3).

Our **third contribution**, and the most significant, is the introduction of cross-block interaction terms, enabling us to avoid groupings containing pairs of blocks that jointly score poorly. Our final objective is:

$$\begin{aligned} F(\pi) =& F_{\text{ra}}(\pi) + \lambda_3 \min_{i,j\in[m], i<j} F_{i,j}(A_i^\pi, A_j^\pi) \qquad (2) \\ &+ \lambda_4 \frac{1}{\binom{m}{2}} \sum_{i,j\in[m],i<j} F_{i,j}(A_i^\pi, A_j^\pi). \end{aligned}$$

While there are four λ_is in this objective, typically only two—one for scoring individual blocks, and the other for pairwise interactions—will be nonzero. We interpret the extra cross-block terms $F_{i,j}$ as rewarding *inter-block diversity*. For example, we could cause our objective function to prefer blocks with large pairwise symmetric differences by taking $F_{i,j}(A_i^\pi, A_j^\pi) = \left|A_i^\pi \triangle A_j^\pi\right|$. Alternatively, in the partitioning or packing setting, we could define $F_{i,j}(A_i^\pi, A_j^\pi) = f(A_i^\pi \cup A_j^\pi)$, in which case if there are two blocks A_i^π, A_j^π with either $f(A_i^\pi \cup A_j^\pi) \approx f(A_i^\pi)$ or $f(A_i^\pi \cup A_j^\pi) \approx f(A_j^\pi)$, then, under an interpretation of f as a diversity measure, the two blocks would be redundant, a situation we would prefer to avoid. We study several possible cross-block interactions, based on unions, intersections and symmetric differences, and $\{$sub,super$\}$modular functions thereof, and show that cross-block diversity *preserves submodularity* in an expanded ground set under various set-to-set mappings (Section 4).

Finally, we offer an approach that reduces the above problem to either non-monotone (without robust terms, i.e. $\lambda_1 = 0$ and $\lambda_3 = 0$) or iterative monotone (with one robust term, i.e. only one of λ_1 or λ_3 is nonzero) submodular maximization subject to multiple matroid constraints (Section 5).

1.2. Applications

There are several applications in machine learning and data science that fit naturally into this setting, two of which we outline here.

Constructing ensembles of machine learning models: Let V index a set of features, with subsets of V corresponding to subsets of features on which a model will be trained. The classical *feature selection* problem would be to choose a single set of features that result in a good model. We're interested instead in the problem of finding an ensemble of models, each trained on a different subset of features, that together achieve good performance (Canini et al., 2016).

This can be done by grouping V into $A_1^\pi, A_2^\pi, \ldots, A_m^\pi$, from which we form an ensemble of m models, the results of which are aggregated together e.g. by averaging, voting, or taking the minimum. Given a submodular function $f : V \to \mathbb{R}_+$ measuring the "quality" of a *single* feature subset, one natural goal would be to choose each A_i^π to be individually high-quality, according to f. However, since the ensemble outputs are being aggregated, it would be purposeless to have redundant models—many results (e.g. Kittler et al., 1998) suggest that when aggregating models, it is best for them to be as diverse as possible (so that the errors they make are independent, thereby improving accuracy and reducing variance). This motivates us to seek blocks that are as different from each other as possible. Individual model quality combined with diversity is exactly what maximizing Eq. (2) encourages. Our case study (Section 6) was performed in this setting and supports the benefits of aggregating diverse models.

Multiple mutually diverse summaries: Data summarization involves finding a small but representative subset of a large set. There are some cases where it is useful to have multiple mutually diverse summaries, each of which is representative of the whole. For example, in parallel machine learning, where training data might need to be partitioned onto multiple machines distributed across a network, it can be useful to ensure that each subset is representative (so that local computations are accurate) but also diverse (since if two subsets are redundant, than so will the work that each processor performs). As another example, consider the problem of document summarization. It can be useful to produce multiple representative but distinct summaries of a collection of documents, as this ensures all concepts are covered but different perspectives are preserved.

2. Previous Work

Special cases of our problem have previously been studied. For example, maximizing Eq. (1) with $1 = \lambda_1 = 1 - \lambda_2$ over the space of all otherwise unconstrained partitions corresponds to the submodular fair allocation (SFA) problem. It is possible to achieve a $O(1/(\sqrt{m}\log^3 m))$ approximation (Asadpour & Saberi, 2010) via iteratively rounding an LP solution when the f_i's are all modular, although the problem is NP-hard to $1/2 + \epsilon$ approximate for any $\epsilon > 0$ (Golovin, 2005). For submodular f_i's, (Golovin, 2005) also gives a matching-based algorithm with a factor $1/(n - m + 1)$ approximation. A binary search algorithm (Khot & Ponnuswami, 2007) has a better factor of $1/(2m - 1)$ that is independent of n. A less practical approach uses an ellipsoid approximation (Goemans et al., 2009) of each submodular function and reduces SFA to its modular version yielding an approximation factor of $1/(\sqrt{n}m^{1/4}\log n \log^{3/2} m)$. (Wei et al., 2015) shows that a greedy algorithm has a $1/m$ approximation when all f_i's are the same. Fair allocation problems are also studied with sometimes non-submodular objectives (Ghodsi et al., 2017). Maximizing Eq. (1) with $0 = \lambda_1 = 1 - \lambda_2$ over the space of all partitions corresponds to the submodular welfare problem (SW), which can be reduced to submodular maximization on an expanded ground set under a partition matroid constraint (Vondrák, 2008) using a greedy algorithm, an approach having a $1/2$ guarantee (Fisher et al., 1978). The multi-linear extension of a submodular function can be used in a continuous greedy approach that solves SW with a $(1 - 1/e)$ tight approximation factor (Vondrák, 2008). When $\lambda_1 > 0, \lambda_2 > 0$, (Wei et al., 2015) offers two approaches. The first takes the best of the two solutions computed under $\lambda_1 = 1 = 1 - \lambda_2$ and $\lambda_1 = 0 = 1 - \lambda_2$ to provide a $\max(\frac{\beta\alpha}{(\lambda_1)\beta+\alpha}, \lambda_2\beta)$ guarantee, where α is the approximation factor for the SFA problem and β the factor for the SW problem. A second binary-search approach, the inspiration for Algorithm 1, finds a partition whose block objective value is at least $\max(\frac{\delta}{1-\alpha+\delta}, \lambda_2\alpha)(\text{OPT} - \epsilon)$ for an $\alpha - \delta$ fraction of the blocks, where α is the approximation factor of a SW solver and $0 < \delta < \alpha$.

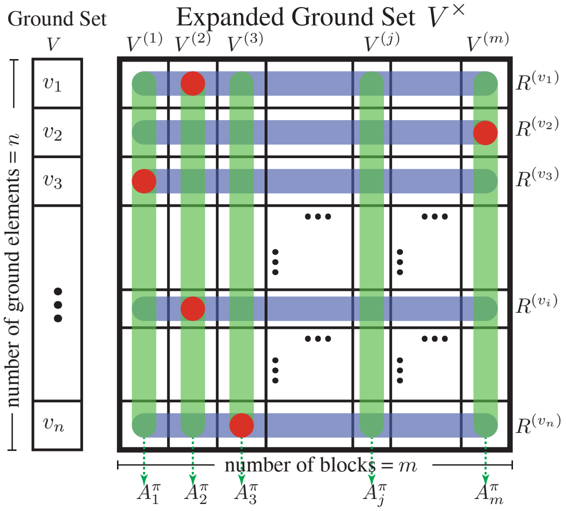

Figure 1. Illustration of the mapping from the original ground set V to the expanded ground set $V^\times$. A subset $S \subset V^\times$ is shown as red dots. The resulting blocks are shown at the bottom—e.g. $v_3 \in A_1^\pi$, $\{v_1, v_i\} \subseteq A_2^\pi$, $v_n \in A_3^\pi$, etc.

The novelty of our optimization problem is that: (i) we may form not only partitions but also $(\mathbf{k}, m)$-coverings and packings; (ii) we utilize a set of m matroids to define the feasibility of the individual blocks in a grouping; and (iii) we explicitly incorporate cross-block interaction terms.

3. Approach

As with the strategy for the submodular welfare problem (Vondrák, 2008), our approach to maximizing Eq. (2) starts by defining an *expanded ground set* $V^\times$ (Figure 1), consisting of m disjoint unions of the original ground set V, i.e., the product set, defined as:

$$V^\times \triangleq \biguplus_{i=1}^{m} V^{(i)} = \biguplus_{v\in V} R^{(v)} = \{(v,i) : v \in V, i \in [m]\}$$

where $|V^\times| = nm$ and where $\uplus$ is the disjoint union operator. $V^\times$ can be viewed as indexing into a $n \times m$ matrix with $V^{(i)}$ (isomorphic to V) being the i^{th} column, and $R^{(v)}$ (isomorphic to $[m]$) being the v^{th} row.

We also define a mapping from subsets $S \subseteq V^\times$ to the original ground set, and another mapping that selects the original ground set elements corresponding to those in the ith column as follows:

$$\begin{aligned}\operatorname{abs}(S) &\triangleq \{v \in V : \exists i \in [m] \text{ with } (v,i) \in S\}\\ \operatorname{col}(S,i) &\triangleq \operatorname{abs}\left(S \cap V^{(i)}\right)\end{aligned}$$

Given $S \subseteq V^\times$, a grouping π is obtained by setting $A_i^\pi = \operatorname{col}(S,i)$ for all $i \in [m]$.

Using these mappings, we will ultimately (Eq. (5)) define a new objective $F^\times : 2^{V^\times} \to \mathbb{R}$ on the expanded ground set that produces the valuation of a set $S \subseteq V^\times$ *indirectly*, via submodular functions defined on the original ground set, using $\operatorname{col}(S,i)$ to map subsets of $V^\times$ to subsets of V.

3.1. Partitionings, Packings, and Coverings

During optimization, we will take the feasible set $\mathcal{F}^\times \subseteq 2^{V^\times}$ to be the intersection of the independent sets of zero or more matroids over $V^\times$. Such matroid independence constraints can be used to ensure that any feasible solution maps back to a partitioning, packing, or covering over V. When we wish for a partition, we can maximize $F^\times$ subject to a partition matroid on $V^\times$ whose independent sets are defined based on the "rows". That is, the independent sets are defined as follows:

$$\mathcal{I}_{\mathbf{k}} = \left\{S \subseteq V^\times : \forall v \in V, \left|S \cap R^{(v)}\right| \leq k_v\right\}, \qquad (3)$$

where $\mathbf{k} = (k_{v_1}, k_{v_2}, \ldots, k_{v_n})$ and $\forall v, k_v = 1$. In words, at most one "copy" of each element of the original ground set may be present. To express $(\mathbf{k}, m)$-covering constraints on V, we allow $\forall v, k_v \geq 1$. A covering and partition is obtained when maximizing a monotone $F^\times$, since any candidate solution that is not yet a covering or partition can be made so by adding elements until all constraints are met with equality.

Likewise, a *packing* constraint can be expressed using a ℓ-uniform matroid with independent sets:

$$\mathcal{I}_\ell = \big\{S \subseteq V^\times : |S| \leq \ell\big\}. \qquad (4)$$

If we set $\mathcal{F}^\times = \mathcal{I}_\ell \cap \mathcal{I}_{\mathbf{k}}$, where $\mathbf{k} = \mathbf{1}$ is the vector of all 1s, this expresses a packing constraint, and is the intersection of two matroids defined on $V^\times$. In fact, the intersection of a matroid and a ℓ-uniform matroid is still a single matroid, called its ℓ-truncation (Schrijver, 2003), and hence $\mathcal{I}_\ell \cap \mathcal{I}_{\mathbf{k}}$ constitutes only a single matroid.

3.2. Block Constraints

Having discussed how matroid constraints on the rows of $V^\times$ can be used to express the partitioning, packing and covering problems, we now turn our attention to how matroid constraints on the columns can be used to represent more general constraints on the blocks. Imagine that each block is required to satisfy its own matroid independence constraint: we are given m matroids $\{(V, \mathcal{I}_i)\}_{i\in[m]}$ with independent sets $\mathcal{I}_i$ for $i \in [m]$, where each matroid is defined over the *original* ground set V. Using the expanded ground set and taking $S \subseteq V^\times$, we have that $A_i^\pi \in \mathcal{I}_i$ if and only if $\operatorname{col}(S,i) \in \mathcal{I}_i$.

Given a size-m set of matroids $\{\mathcal{M}_i\}_{i\in[m]}$ where $\mathcal{M}_i = (V, \mathcal{I}_i)$, the matroid union theorem (Schrijver, 2003, Thm. 42.1a) states that a new matroid can be defined on $V^\times$ with independent sets $\mathcal{I}_{\mathrm{b}} = \{I_1 \uplus I_2 \uplus \ldots \uplus I_m : I_i \in \mathcal{I}_i, \forall i \in [m]\}$. Despite there being a matroid for each block, the disjoint union of these matroids is a *single* matroid on $V^\times$.

One of the simplest examples of such constraints, and the one that we use in our case study (Section 6), simply places an upper bound on the number of elements within each block. The resulting matroid is a column-based analogue of Eq. (3).

4. Cross-block Interaction

In order to define the expanded objective $F^\times$ in terms of submodular functions on the original ground set V, we will define set functions via mappings from subsets of an expanded ground set to subsets of the original ground set. The next result shows that in some cases, composition and set-to-set mappings preserve submodularity or supermodularity.

Lemma 1. *Let V', V be two ground sets and define a set-to-set mapping function $G : 2^{V'} \to 2^V$. Also, let $f : 2^V \to \mathbb{R}_+$ be monotone non-decreasing and submodular, and let $g : 2^V \to \mathbb{R}_+$ be monotone non-decreasing and supermodular. Then:*

1. *If G is monotone non-decreasing (i.e. $G(S) \subseteq G(T)$ whenever $S \subseteq T$), then $f \circ G$ and $g \circ G$ are both*

monotone non-decreasing.[1]

2. *If* $\forall S, T \subseteq V'$, $G(S \cup T) = G(S) \cup G(T)$ *and* $G(S \cap T) \subseteq G(S) \cap G(T)$, *then* $f \circ G : 2^{V'} \to \mathbb{R}_+$ *is submodular.*

3. *If* $\forall S, T \subseteq V'$, $G(S \cup T) \supseteq G(S) \cup G(T)$ *and* $G(S \cap T) = G(S) \cap G(T)$, *then* $g \circ G : 2^{V'} \to \mathbb{R}_+$ *is supermodular.*

Proof. In Appendix C. □

The objective defined in Eq. (2) involves cross block interaction terms via $F_{i,j}(A_i^\pi, A_j^\pi)$ for all $i, j \in [m]$. The ground set expansion defined in Section 3, combined with the above lemma, surprisingly allows many such interaction terms to be handled in a way that preserves submodularity. To this end, we define three additional set-to-set (i.e. $2^{V^\times} \to 2^V$) mappings corresponding to union, intersection, or symmetric difference of the ith and jth mapped subsets:

$$G_{i,j}^{\square}(S) \triangleq \operatorname{col}(S, i) \, \square \, \operatorname{col}(S, j)$$

where $\square = \cup, \cap$ or $\triangle$. The following lemma, which largely follows from Lemma 1, shows that these can be used in a way that preserves sub/supermodularity:

Lemma 2. *Let* $f : 2^V \to \mathbb{R}$ *be monotone non-decreasing submodular,* $m : 2^V \to \mathbb{R}$ *be non-negative modular, and* $g : 2^V \to \mathbb{R}$ *be monotone non-decreasing supermodular. Then* $f \circ G_{i,j}^{\cup} : 2^{V^\times} \to \mathbb{R}$ *is monotone non-decreasing submodular,* $g \circ G_{i,j}^{\cap} : 2^{V^\times} \to \mathbb{R}$ *is monotone non-decreasing supermodular, and* $m \circ G_{i,j}^{\triangle} : 2^{V^\times} \to \mathbb{R}$ *is non-negative submodular.*

Proof. In Appendix C. □

If one wishes to reduce the number of common elements in pairs of blocks, a useful cross-block interaction term is $|V| - \left|G_{i,j}^{\cap}(S)\right|$. Because the cardinality function is non-decreasing and modular, $\left|G_{i,j}^{\cap}(S)\right|$ is non-decreasing supermodular by Lemma 2. A submodular function (including the constant $|V|$) minus a supermodular function is submodular, and thus, the above interaction function is submodular and non-increasing.

In the partitioning and packing settings, if we have a non-decreasing submodular function f that measures the diversity of a set, one could use $f\left(G_{i,j}^{\cup}(S)\right)$, which is both submodular and non-decreasing, as the cross-block interaction term, to encourage pairwise diversity. If one wants blocks to have large pairwise differences, then a natural cross-block interaction term is $|G_{i,j}^{\triangle}(S)|$. This function is again submodular, but is non-monotone (it is neither non-increasing nor non-decreasing).

4.1. Submodular Approximation of $f \circ G_{i,j}^{\triangle}$

Unfortunately, $f \circ G_{i,j}^{\triangle}$ is not necessarily submodular, even for a submodular f, and has no obvious difference representation (Iyer & Bilmes, 2012). However, we can derive (non-monotone) submodular bounds based on the curvature. A normalized monotone submodular function $f : 2^V \to \mathbb{R}$ has *curvature* c if $f(v|S) \geq (1-c)f(v)$ for all $S \subseteq V$ and $V \ni v \notin S$, where $f(v|S) \triangleq f(S \cup \{v\}) - f(S)$ is the *gain*. Curvature is easily computed in $O(n)$ time since $c = 1 - \min_{v \in V} f(v|V \setminus v)/f(v)$, where $c \in [0, 1]$. Modular functions have $c = 0$, fully curved functions have $c = 1$, and submodular function classes can be restricted to those having a particular c, since many useful submodular functions have non-extreme curvature, e.g. sums of non-asymptoting concave functions composed with non-negative modular functions (Stobbe & Krause, 2010).

Given a set $X \subseteq V$, a normalized monotone submodular function f with curvature c, and any ordering $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ of V such that $X = \{\sigma_1, \sigma_2, \ldots, \sigma_{|X|}\}$, a modular subgradient m_f^X of f can be obtained (Fujishige, 2005) where $m_f^X(X) = f(X)$, $\forall Y, m_f^X(Y) \leq f(Y)$, and where $m_f^X(\sigma_i) = f(\sigma_i|\sigma_1, \sigma_2, \ldots, \sigma_{i-1})$. Since f is monotone, the subgradient is non-negative. Also, for any Y, submodularity ensures that $m_f^X(v) \geq (1-c)f(v)$ and hence $f(Y) \geq m_f^X(Y) \geq (1-c)f(Y)$ for any X, Y, where the second inequality follows since f is normalized submodular. This enables us to obtain a non-monotone submodular lower bound via $f \circ G_{i,j}^{\triangle}(S) \geq m_f^X \circ G_{i,j}^{\triangle}(S) \geq (1-c) f \circ G_{i,j}^{\triangle}(S)$ for any $X \in V$ that approximates the original problem:

Lemma 3. *Given any algorithm that produces a solution* $\hat{S}$ *having the property that* $F(\hat{S}) + m_f^X \circ G_{i,j}^{\triangle}(\hat{S}) \geq \alpha \max_{S \in \mathcal{F}^\times} \left(F(S) + m_f^X \circ G_{i,j}^{\triangle}(S)\right)$ *for* $\alpha > 0$, *then* $\hat{S}$ *also has the property that* $F(\hat{S}) + f \circ G_{i,j}^{\triangle}(\hat{S}) \geq \alpha(1 - c) \max_{S \in \mathcal{F}^\times} \left(F(S) + f \circ G_{i,j}^{\triangle}(S)\right) = \alpha(1-c)OPT.$

Proof. In Appendix C. □

5. Algorithms and Optimization

Eq. (2) can now be written in terms of the expanded ground set as $F^\times : 2^{V^\times} \to \mathbb{R}$:

$$\begin{aligned} F^\times(S) = & \qquad\qquad (5) \\ & \lambda_1 \min_i f_i(\operatorname{col}(S, i)) + \frac{\lambda_2}{m} \sum_{i \in [m]} f_i(\operatorname{col}(S, i)) \\ & + \lambda_3 \min_{i,j \in [m], i<j} F_{i,j}^\times(S) + \frac{\lambda_4}{\binom{m}{2}} \sum_{i,j \in [m], i<j} F_{i,j}^\times(S) \end{aligned}$$

Our approach in maximizing Eq. (2) subject to grouping constraints is to optimize Eq. (5) subject to the intersection

[1] Note that "$\circ$" denotes function composition.

Algorithm 1 Adaptation of the GeneralGreedSAT algorithm of Wei et al. (2015) to handle matroid constraints, cross-block interactions, and coverings/packings as well as partitions. The functions $f_1^\times, \ldots, f_m^\times : 2^{V^\times} \to \mathbb{R}$ are monotone non-decreasing, non-negative and submodular, $\eta_{f^\times}$ is a uniform upper bound on the $f_i^\times$'s, each $\mathcal{I}_1, \ldots, \mathcal{I}_k \subseteq 2^{V^\times}$ is the set of independent sets of a matroid on $V^\times$, and CALLBACK is a helper function that returns an α-approximation to the submodular maximization problem $\operatorname{argmax}_{S \in \bigcap_{i=1}^k \mathcal{I}_i} F_c^\times(S)$ in polynomial time.

```
Bisection $(\epsilon, V^\times, m, f_1^\times, \ldots, f_m^\times, \eta_{f^\times}, k, \mathcal{I}_1, \ldots, \mathcal{I}_k, \textsc{Callback}, \alpha)$:
    Define $F_c^\times(S) = \left(\sum_{i=1}^m \min\left\{c, f_i^\times(S)\right\}\right)/m$
    Initialize $c_{\min} = 0, c_{\max} = \eta_{f^\times}, S = \emptyset$
    While $c_{\max} - c_{\min} > \epsilon$:
        Let $c = (c_{\max} - c_{\min})/2$ and $S_c = \textsc{Callback}(V^\times, F_c^\times, k, \mathcal{I}_1, \ldots, \mathcal{I}_k)$
        If $F_c^\times(S_c) < \alpha c$, then let $c_{\max} = c$, else let $c_{\min} = c$ and $S = S_c$
    Return $S$
```

of multiple matroid constraints defined on the expanded ground set. The matroid constraints ensure that the solution can be transformed back to the original ground set, while preserving the approximation ratio. The algorithm used depends both on which terms are present (i.e. have nonzero associated λs), and whether the submodular functions are monotone.

When the overall objective is non-monotone submodular (so $\lambda_1 = 0$ and $\lambda_3 = 0$), the problem of maximizing Eq. (5) becomes one of non-monotone submodular maximization subject to matroid constraints. One can use algorithms such as Lee et al. (2010); Ward (2012), or the more recent, faster, and scalable approach given in Feldman et al. (2017). When using $f \circ G_{i,j}^{\triangle}(S)$ as a non-submodular block pair reward for non-robust coverings, a modular approximation adjusts any guarantees by $1 - c$ (Lemma 3). The non-robust packing or partitioning problem reduces to monotone submodular maximization subject to two matroid constraints, for which there are a variety of good solutions. For example, the efficient greedy algorithm (Nemhauser & Wolsey, 1978) solves this problem with a $1/3$ guarantee while more recent but also more complicated approaches, such as Ward (2012), can solve this with an approximation ratio of $(k+3)/2 + \epsilon$ for ℓ matroids (here $k = 2$).

When all of the involved submodular functions are *monotone non-decreasing*, a single robust term can be handled (one of λ_1 or λ_3 may be nonzero). Here, we are inspired by an approach originally used for robust submodular optimization (Krause et al., 2008) where the goal is to find the $\max$ of the $\min$ over a set of submodular functions subject to a cardinality constraint. In Wei et al. (2015), this was extended to apply to a mixed robust/average objective over partitions. It turns out that essentially the same idea—this is Algorithm 1—applies to the more general case of coverings and packings, as well as when there are matroids constraining each block individually, and also when the blocks' scores interact.

In brief, this algorithm proceeds by iteratively optimizing inner submodular optimization problems using a provided CALLBACK function, which is assumed to run in polynomial time, and return an α-approximation. The final algorithm achieves a nearly constant-factor approximation for a constant fraction of the blocks. As the fraction of the blocks shrinks, the guarantee for those blocks grows, and the guarantee holds simultaneously for a range over the fractions.

Theorem 1. *A call to Algorithm 1 will perform $\lceil \log_2(\eta_f/\epsilon) \rceil$ calls to* CALLBACK*, each of which will perform polynomially many evaluations of $F_c^\times$, each of which evaluates all m $f_i^\times$s once.*

The resulting set S will satisfy the constraints, and for any $\gamma \in (0, \alpha)$ there will exist at least $\lceil m(\alpha - \gamma)/(1 - \gamma) \rceil$ indices $i \in [m]$ for which $f_i^\times(S) > \gamma(OPT - \epsilon)$, where $OPT = \max_{S \in \bigcap_{i=1}^k \mathcal{I}_i} F^\times(S)$ is the optimum.

Proof. The proof technique follows that of Wei et al. (2015, Theorem 11), but we include it in Appendix C for completeness, and to show that it applies to our more general case (i.e., covers, packings, block-specific matroid constraints, and cross-block interactions). □

This algorithm depends on a CALLBACK that *deterministically* returns an α-approximation to an inner submodular optimization problem. However, many submodular maximization algorithms are randomized, and have approximation guarantees that hold only in expectation. With a bit of extra work, such an algorithm can be used as well. First, we must convert the in-expectation guarantee into a high-probability guarantee using the following lemma that requires only an approximation bound:

Lemma 4. *Let A be a randomized algorithm for submodular maximization that has an α-approximation guarantee* in expectation*, i.e. for which $\mathbb{E}[f(S)] \geq \alpha f(S^*)$, where f is the submodular function we wish to maximize, S is the result of algorithm A, and S^* is the maximizer of f. For* *parameters $\beta, \delta \in (0,1)$, suppose that we run algorithm A k times, where $k = \left\lceil \left(\ln \frac{1}{\delta}\right) / \left(\ln \frac{1-\alpha\beta}{1-\alpha}\right) \right\rceil$, yielding results $S_1, S_2, \ldots, S_k$. Take $S = \operatorname{argmax}_{S_i : i \in [k]} f(S_i)$ to be the best of these results. Then S will have an approximation ratio of $\alpha\beta$, i.e. $f(S) \geq \alpha\beta f(S^*)$, with probability $1 - \delta$.*

Proof. In Appendix C. □

As was shown in Theorem 1, CALLBACK will be called at most $\lceil \log_2 (\eta_f / \epsilon) \rceil$ times, so it follows from the union bound that if we use the procedure of Lemma 4, then, with probability $1 - \delta \lceil \log_2 (\eta_f / \epsilon) \rceil$, every call to CALLBACK will return an $\alpha\beta$-approximation, and the result of Theorem 1 will hold (with $\alpha\beta$ substituted for α).

The ability to handle a mixed robust/average objective, however, comes at a cost. Because Theorem 1 only applies when all of the involved submodular functions are monotone non-decreasing, we cannot use the intersection and symmetric-difference-based cross-block interaction terms—only the union-based terms are possible. Non-monotone interaction terms can only be used with a non-robust objective. Finding an algorithm that can handle *both* robustness *and* non-monotone interactions is therefore an interesting open problem that we leave to future work.

6. Case Study

We validate our proposed approach with a case study in the setting of Canini et al. (2016), in which the task is to construct an ensemble-of-lattices machine learning model. Each lattice model (Gupta et al., 2016) in the ensemble is defined on a subset of the features—intuitively, two features interact non-linearly if they are included in the same lattice, and interact only linearly if they are not—so our primary goal is to choose subsets of features that interact well with each other, with our secondary goal being to reduce redundancy in pairs of subsets. Notice that this is *not* a feature selection problem—typically, every feature will be included in at least one lattice—the task is to determine which features should interact non-linearly. For more details, please see Appendix A.

Our goals are to demonstrate that (i) we can successfully find good approximate maximizers of the proposed objective function, and (ii) the inclusion of pairwise diversity terms results in improved diversity.

We compare to two baselines, the "Crystals" and "Random Tiny Lattice (RTL)" algorithms of Canini et al. (2016). The first of these—the current state-of-the-art—is essentially a heuristic for choosing diverse ensembles, while the second simply chooses each lattice's features uniformly at random.

The dataset contains $463\,154$ samples with 29 informative features plus a binary label indicating whether a particular visual element should be displayed on a web page. The dataset was randomly partitioned into training, validation and testing subsets containing 80%, 10% and 10% of the data, respectively (the validation set was only used for hyperparameter optimization of the baseline Crystals algorithm).

6.1. Choice of f

Our ground set V consists of the $n = 29$ features. We began by finding a submodular function $f : 2^V \to \mathbb{R}_+$ for which $f(S)$ represents roughly how well a single lattice model on the features in S would perform. To this end, we chose f to have the form $f(S) = \beta + \sum_{A \in \mathcal{A}} \alpha_A \sqrt{|A \cap S|}$, where $\mathcal{A}$ consists of all 1- and 2-element subsets of V. Observe that $\sqrt{|A \cap S|}$ is submodular, non-negative, and monotone non-decreasing, so if β and α_A are non-negative, then f will likewise be submodular, non-negative, and monotone.

Based on $9\,191$ random subsets of sizes between two and ten, we learned the β and α_A parameters to minimize the squared error between $f(S)$ and the training accuracy of a lattice model trained on the features contained in S. The result is the f that we use throughout.

6.2. Covering

Our goal here is essentially identical to Canini et al. (2016)—we seek to choose $m = 8$ lattices, each containing up to 8 features (via a matroid constraint), by finding the $S \subseteq V^\times$ maximizing:

$$\sum_{i \in [m]} f(\operatorname{col}(S, i)) + \frac{\lambda_4}{|V|} \sum_{i,j \in [m] \wedge i \neq j} \left| G^{\triangle}_{i,j}(S) \right| \tag{6}$$

The 8 lattices together should have good performance (the first term), and the feature subsets should be relatively pairwise distinct, i.e. have large symmetric differences (this is the second term). We henceforth refer to the first (intra-block diversity) term, representing the individual quality of the lattices, as the "quality" term, and the second (not including the λ_4-scaling), representing the inter-block diversity, as the "diversity" term.

We optimized Eq. (6) for various choices of λ_4 using the randomized algorithm of Feldman et al. (2017) combined with the procedure of Lemma 4, with $\beta = 0.5$ and $\delta = 0.1$. Each optimization took between 2 and 30 seconds on a Xeon E5-2690. The results are shown in Figure 2. The left-hand plot shows that, as the trade-off parameter λ_4 increases, the relative magnitude of the quality term decreases, and of the diversity term increases, as expected. The right-hand plot shows that, when the λ_4 parameter is sufficiently large, the diversity term is "balanced" with the quality term (or is larger), and the ensembles found by our approach outperform those of both the state-of-the-

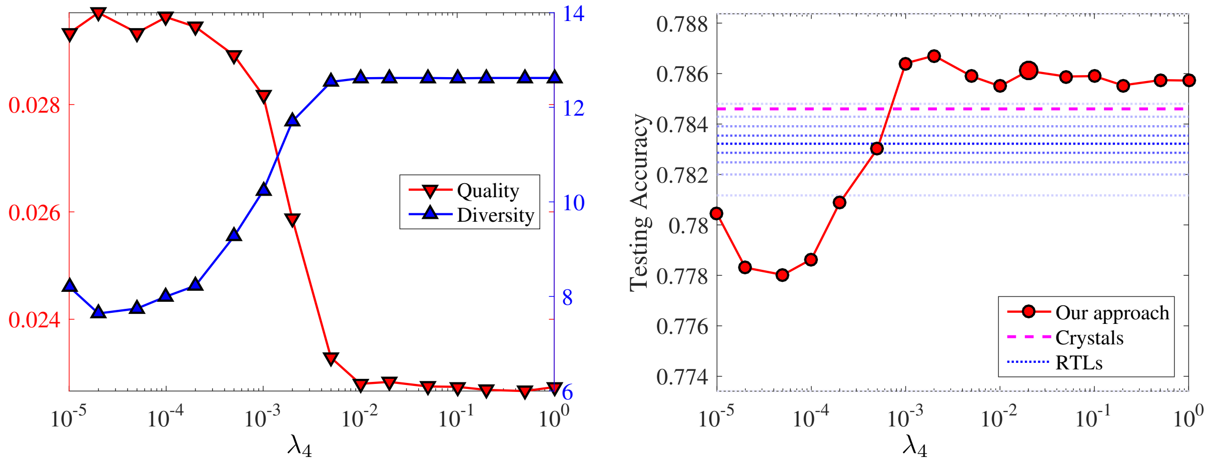

Figure 2. **(Left)** The magnitudes of the "quality" and "diversity" terms of Eq. (6) as functions of the trade-off parameter λ_4, averaged over ten independent runs. **(Right)** The testing accuracies of our proposed approach, as well as the "Crystals" and "Random Tiny Lattices" (RTL) algorithms of Canini et al. (2016), again as functions of λ_4. The plotted results of our algorithm are averaged over ten independent runs, with the larger point being that for which the validation accuracy was maximized. For the RTL results, we trained 1000 random models. The lower and upper bounds of the plot are the worst and best testing accuracies (respectively) over all models, and the dotted horizontal lines are the testing accuracies of the RTL model at the 10th, 20th, . . . , 90th percentiles.

art Crystals algorithm (which uses heuristics to encourage diversity) and the 90th percentile of RTLs, albeit by a small amount. More importantly, the leftmost points in the right-hand plot of Figure 2, in which the diversity portion of the objective is essentially zero, have significantly worse testing accuracies than those with larger λ_4s. This indicates that the use of pairwise diversity terms may be broadly beneficial to submodular grouping problems.

6.3. Partitioning and Packing

Appendix B contains additional case studies exploring the efficacy of the mixed robust/average objective for partitioning and packing, where the task is to maximize:

$$\sum_{i\in[m]} f\left(\mathrm{col}\left(S,i\right)\right) + \lambda_3 \min_{i,j\in[m]\wedge i\neq j} f\left(G^{\cup}_{i,j}(S)\right) \tag{7}$$

Unlike in Eq. (6), the "diversity" term is a minimum over monotone non-decreasing submodular functions, instead of a sum over non-monotone submodular functions. The results demonstrate that Algorithm 1 is effective at optimizing this objective, but also reveal that, for this problem and data set, the above diversity term is not helpful—and can be *harmful* if the quality term is overpowered. The lesson is that cross-block diversity is not a magic bullet—it must be chosen appropriately for the problem.

6.4. Discussion

We have introduced a new class of submodular optimization problems involving grouping ground elements together into multiple sets and the first, as far as we know, to involve block-block interaction terms as well as general (matroid intersection) block constraints.

Another potential application of our method is sensitivity analysis of machine learning systems (i.e., does an ML model vary greatly when trained on different representative but mutually diverse subsets of the training data?) and also a form of robustness analysis (i.e., how does an ML system perform when tested on different representative but mutually diverse subsets of test data?). These are important questions, considering for example the recent interest in adversarial examples in ML.

Also, since convex combinations of submodular components preserve submodularity, it might also in the future be interesting to consider some form of Pareto frontier of solution sets for different convex mixtures.

Acknowledgments: This work was done in part during a time when author Bilmes was visiting Google AI research and also visiting the Simons Institute for the Theory of Computing. This material is based upon work supported by the National Science Foundation under Grant No. IIS-1162606, the National Institutes of Health under award R01GM103544, and by a Google, a Microsoft, a Facebook, and an Intel research award. This work was supported in part by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA

References

Asadpour, A. and Saberi, A. An approximation algorithm for max-min fair allocation of indivisible goods. In *SICOMP*, 2010.

Canini, K., Cotter, A., Gupta, M., Milani Fard, M., and Pfeifer, J. Fast and flexible monotonic functions with ensembles of lattices. In *NIPS*, pp. 2919–2927, 2016.

Djolonga, J., Tschiatschek, S., and Krause, A. Variational inference in mixed probabilistic submodular models. In *Neural Information Processing Systems (NIPS)*, December 2016.

Feldman, M., Harshaw, C., and Karbasi, A. Greed is good: Near-optimal submodular maximization via greedy optimization. *arXiv preprint arXiv:1704.01652*, 2017.

Fisher, M., Nemhauser, G., and Wolsey, L. An analysis of approximations for maximizing submodular set functions—II. In *Polyhedral combinatorics*, 1978.

Fujishige, S. *Submodular functions and optimization*, volume 58. Elsevier, 2005.

Ghodsi, M., HajiAghayi, M., Seddighin, M., Seddighin, S., and Yami, H. Fair allocation of indivisible goods: Improvement and generalization. *arXiv preprint arXiv:1704.00222*, 2017.

Goemans, M., Harvey, N., Iwata, S., and Mirrokni, V. Approximating submodular functions everywhere. In *SODA*, 2009.

Golovin, D. Max-min fair allocation of indivisible goods. *Technical Report CMU-CS-05-144*, 2005.

Gotovos, A., Hassani, S. H., and Krause, A. Sampling from probabilistic submodular models. In *Neural Information Processing Systems (NIPS)*, December 2015.

Gupta, M. R., Cotter, A., Pfeifer, J., Voevodski, K., Canini, K., Mangylov, A., Moczydlowski, W., and van Esbroeck, A. Monotonic calibrated interpolated look-up tables. *JMLR*, 17(109):1–47, 2016.

Iyer, R. K. and Bilmes, J. A. Algorithms for approximate minimization of the difference between submodular functions, with applications. *CoRR*, abs/1207.0560, 2012. URL `http://arxiv.org/abs/1207.0560`.

Kempe, D., Kleinberg, J., and Tardos, É. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137–146. ACM, 2003.

Khot, S. and Ponnuswami, A. Approximation algorithms for the max-min allocation problem. In *APPROX*, 2007.

Kittler, J., Hatef, M., Duin, R. P. W., and Matas, J. On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(3):226–239, March 1998.

Kohli, P., Kumar, M. P., and Torr, P. H. P3 & beyond: Solving energies with higher order cliques. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8. IEEE, 2007.

Krause, A., McMahan, B., Guestrin, C., and Gupta, A. Robust submodular observation selection. In *JMLR*, 2008.

Lee, J., Sviridenko, M., and Vondrák, J. Submodular maximization over multiple matroids via generalized exchange properties. *Math. Oper. Res.*, 35(4):795–806, 2010.

Li, J., Li, L., and Li, T. Multi-document summarization via submodularity. *Applied Intelligence*, 37(3):420–430, 2012.

Lin, H. and Bilmes, J. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 510–520. Association for Computational Linguistics, 2011.

Lin, H., Bilmes, J., and Xie, S. Graph-based submodular selection for extractive summarization. In *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*, pp. 381–386. IEEE, 2009.

Mossel, E. and Roch, S. On the submodularity of influence in social networks. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pp. 128–134. ACM, 2007.

Narasimhan, M., Jojic, N., and Bilmes, J. Q-clustering. In *NIPS*, volume 5, pp. 5, 2005.

Nemhauser, G. and Wolsey, L. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, 3(3):177–188, 1978.

Oxley, J. *Matroid theory*. Oxford University Press, USA, 2006.

Reed, C. and Ghahramani, Z. Scaling the Indian buffet process via submodular maximization. *ICML*, 2013.

Schrijver, A. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Verlag, 2003.

Singla, A., Bogunovic, I., Bartok, G., Karbasi, A., and Krause, A. Near-optimally teaching the crowd to classify. In *ICML*, pp. 154–162, 2014.

Singla, A., Tschiatschek, S., and Krause, A. Noisy submodular maximization via adaptive sampling with applications to crowdsourced image collection summarization. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Stobbe, P. and Krause, A. Efficient minimization of decomposable submodular functions. In *NIPS*, 2010.

Tschiatschek, S., Iyer, R. K., Wei, H., and Bilmes, J. A. Learning mixtures of submodular functions for image collection summarization. In *Advances in neural information processing systems*, pp. 1413–1421, 2014.

Vondrák, J. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*, 2008.

Ward, J. A (k+ 3)/2-approximation algorithm for monotone submodular k-set packing and general k-exchange systems. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 14. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012.

Wei, K., Iyer, R. K., Wang, S., Bai, W., and Bilmes, J. A. Mixed robust/average submodular partitioning: Fast algorithms, guarantees, and applications. In *NIPS*, 2015.