Approximation Bounds for Inference using Cooperative Cuts

Stefanie Jegelka JEGELKA@TUEBINGEN.MPG.DE
Max Planck Institute for Intelligent Systems, Tübingen, Germany

Jeff Bilmes BILMES@U.WASHINGTON.EDU
University of Washington, Seattle, WA 98195, USA

Abstract

We analyze a family of probability distributions that are characterized by an embedded combinatorial structure. This family includes models having arbitrary treewidth and arbitrary sized factors. Unlike general models with such freedom, where the "most probable explanation" (MPE) problem is inapproximable, the combinatorial structure within our model, in particular the indirect use of submodularity, leads to several MPE algorithms that all have approximation guarantees.

1. Introduction

Our interest is in the "most probable explanation" (MPE) problem: given a probability distribution $p(x) = \frac{1}{Z}\exp(-E(x))$ where $x = (x_1, x_2, \ldots, x_n) \in \mathcal{D}^n$ for some discrete domain $\mathcal{D}$, find

$$x^* \in \operatorname*{argmax}_x p(x), \text{ or equivalently, } x^* \in \operatorname*{argmin}_x E(x),$$

where $E(x)$ is an "energy" function. In this work, we assume all variables are binary, i.e., $\mathcal{D} = \{0, 1\}$.

Without any restrictions placed on E, it is easy to see that there is not much hope for efficient inference, even if we consider bounded approximations. For example, assume that E is given by an oracle, and let $y \in \{0, 1\}^n$ be an unknown vector. Consider the energy $E(x) = 1$ if $x = y$, and $E(x) = \gamma(n)$ otherwise, where $\gamma(n) > 1$ could be any (polynomial-time) computable function of n. With only polynomially many queries to E, it is exponentially unlikely to identify y, and since $\gamma(n)$ is almost arbitrary, no approximation guarantee of any form is possible in polynomial time. The exponential difficulty of approximate inference in such unrestricted models, therefore, is worse than that implied by the well known fact that MPE is NP-hard and not constant-factor approximable (Abdelbar & Hedetniemi, 1998).

Thus, model restrictions are often applied to allow for exact or good approximate inference in polynomial time. These are either structural, such as treewidth or factor size, or functional, such as submodularity. There are often problems with such restrictions, however, such as the well known drawbacks of local pairwise random fields in computer vision. Our work herein is motivated by finding new combinatorial structures that go beyond the previous restrictions but still, as opposed to the introductory example, enable inference with a *bounded approximation factor*. Thus, we devote a major part of this paper to algorithms and approximation bounds. The model we address indeed includes non-local and rich energy functions, and consequently improves, e.g., image segmentation results substantially (Jegelka & Bilmes, 2011).

The common structural restrictions for tractability correspond to factorizations of p. Let p factor with respect to a graphical model $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ comprising $n = |\mathsf{V}|$ nodes and edge set E. The decisive parameter indicating the complexity of MPE in G is the *treewidth* (Chandrasekaran et al., 2008). The treewidth is one less than the size of the maximum clique in a minimum triangulation. Generally, finding the MPE takes time exponential in the treewidth when it is known.

In general, we write $E(x) = \sum_{\phi \in \Phi} E_\phi(x_\phi)$ where Φ corresponds to the set of factors comprising the distribution. Viewed as a bipartite (factor) graph, each $\phi \in \Phi$ is the subset of nodes $\phi \subseteq \mathsf{V}$ involved in a factor. Many approximate inference algorithms rely on $\max_{\phi \in \Phi} |\phi|$ being small. For example, the cost of sending messages even in loopy belief propagation is exponential in $|\phi|$. Therefore, $\max_{\phi \in \Phi} |\phi|$ (which we call the *factorwidth*) may also be seen as a complexity parameter for certain *approximate* inference algorithms.

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

Nevertheless, treewidth and factorwidth are not the only characterizations of tractability. In fact, exact polynomial-time MPE is possible even with maximum treewidth and factorwidth if E is restricted in other ways. A recent class of energy functions having received attention in the vision community is that of submodular functions. A function $g : 2^{\mathsf{V}} \to \mathbb{R}$ is submodular if for all $A, B \subseteq \mathsf{V}$, $g(A) + g(B) \geq g(A \cup B) + g(A \cap B)$. The following notation should cause no confusion: let $X(x) \subseteq \mathsf{V}$ be the set of nodes $v_i \in \mathsf{V}$ whose corresponding variable x_i is one, i.e., $X(x) = \{v_i \in \mathsf{V} : x_i = 1\}$. We can then define an energy function via g as $E(x) = g(X(x))$, and finding an assignment that minimizes the energy is equivalent to finding the subset $X \subseteq \mathsf{V}$ that minimizes g. When g is submodular, this can be done in polynomial time (Fujishige, 2005). As an example of a submodular g that places restrictions neither on treewidth nor factorwidth, consider the submodular function $g(S) = -\sum_i \prod_{v \in S} w_{i,v}$, where $0 \leq w_{i,v} \leq 1$ is a set of coefficients $\forall i$ and $v \in \mathsf{V}$.

Submodular function minimization is not currently a low-order polynomial time algorithm. In some cases, however, much faster inference is possible. For example, if g may be written as $g(S) = \sum_{(v,v') \in \mathcal{N}} g_{v,v'}(S \cap \{v, v'\})$, where each $g_{v,v'}(\cdot)$ is a submodular function over a size-2 ground set $\{v, v'\}$, and $\mathcal{N}$ is a set of node-pairs, then MPE reduces to a minimum (s,t)-cut (Boykov & Jolly, 2001; Kolmogorov & Zabih, 2004) on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. We call $\mathcal{G}$ the *structure graph* to clearly distinguish it from the graphical model G. In particular, $\mathcal{G}$ has terminal nodes s, t, and a node v_i for each variable x_i. For a set of nodes $X \subset \mathcal{V}$, we define its cut as $\delta(X) = \{(u, v) \in \mathcal{E} | u \in X \cup \{s\}, v \in \mathcal{V} \setminus X\}$. A labeling x induces a partition of V and thus an (s,t)-cut $\delta(X(x))$. The graph $\mathcal{G}$ has weights $w : \mathcal{E} \to \mathbb{R}_+$ and is designed such that its cut equals the energy:

$$E(x) = \sum_{e \in \delta(X(x))} w(e) \triangleq w(\delta(X(x))). \tag{1}$$

Let $C^* \subseteq \mathcal{E}$ be the optimal cut, and X^* the nodes reachable from s after removal of C^*; then $C^* = \delta(X^*)$ and $x_i^* = 1$ if and only if $v_i \in X^*$. To achieve such efficiency, the construction must be limited to pairwise energies (a factorwidth of 2). Higher order models may be obtained by adding variables, but at additional cost.

Even though submodular energies are widely applicable, there are still cases where submodularity can be limiting. For example, applications that traditionally have been well suited to submodular functions (such as information cascades) sometimes have exceptions to their submodularity (Sheldon, 2010).

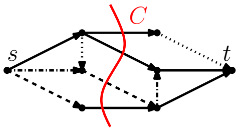


Figure 1. Illustration of a label cost function. Labels are indicated by line style. Cut C cuts four edges with two different labels, hence the cost is $f(C) = 2$.

In this paper, we define a class of energies that are neither submodular nor restricted in treewidth/factorwidth, but still have limited generality in order to retain approximate optimizability. While an application of these energies is addressed in (Jegelka & Bilmes, 2011), we here focus on the theoretical aspects of the problem and propose a set of approximation algorithms for finding MPE, by finding a minimum of the energy. The key feature for deriving approximation bounds is a new structural characterization that relies on a generalization of graph cuts. Although the energies are not submodular, our construction exploits submodularity indirectly. Theorem 7 backs up our approximation factors by giving a lower bound.

1.1. The generalized cut model

Similar to the graph cut analogy (1), we define a set $\mathcal{F}_{coop}$ of energies that are representable by generalized cuts, i.e., *cooperative cuts* in a structure graph $\mathcal{G}$.

Definition 1 (Cooperative Cut)**.** *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes $s, t \in \mathcal{V}$, the cost of an (s,t)-cut $C \subseteq \mathcal{E}$ is measured by a nondecreasing submodular function $f : 2^{\mathcal{E}} \to \mathbb{R}_+$ defined on sets of edges.*

Note that f is defined on **edges**, not nodes. Here, a cut is a set of edges whose removal disconnects s and t. A *submodular* function satisfies diminishing marginal costs: for all $A \subseteq B \subseteq E \setminus \{e\}$, it holds that $f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B)$. In common minimum cut, f would be the sum of edge weights. This is a *modular* function, that is, it satisfies diminishing costs with equality. A cooperative cut replaces the usual sum of weights by a more general submodular cost function f that *couples* the edge weights.

A simple illustrative example of a submodular function is the following: each edge has a label, and the cost of a set of edges is the number of distinct edge labels in the set. Figure 1 illustrates this cost. Other submodular functions include entropy, matroid rank functions and concave functions of sums.

The family $\mathcal{F}_{coop}$ contains all energy functions that can be represented as a cooperative cut in an appropriate structure graph $\mathcal{G}$ with a submodular f such that

$$E_f(x) = f(\delta(X(x))). \tag{2}$$

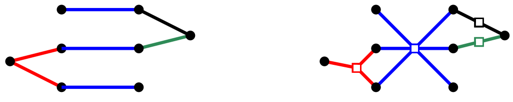

(a) structure graph (b) factor graph

Figure 2. Effect of coupling edges. (a) structure graph $\mathcal{G}$, where coupled edges have the same color; (b) factor graph corresponding to $\mathcal{G}$. Coupled edges couple their incident nodes, and lead to large factors.

Eq. 1 is a special case of Eq. 2, because sums of weights are also submodular. In general, the submodular cost f *couples* sets of edges, such that $f(A)+f(B) > f(A \cup B)$ (in Fig. 1, this happens if A and B contain edges with the same label). As a result, E_f usually has neither of the common simplifying properties mentioned in the introduction, and thus imposes neither of those restrictions on models: (1) all nodes incident to coupled edges are coupled, too, and hence also their corresponding random variables (Fig. 2). Since f can couple up to all edges, the corresponding graphical model can have *arbitrarily large treewidth and factorwidth.* (2) The energy E_f in Eq. 2 is in general *not submodular.* It is subadditive, but subadditivity alone is not enough for tractability: our introductory intractable example is in fact subadditive. In applications, coupling occurs if variables belong to the same greater structure, e.g., in images, to the same object boundary.

In view of (1) and (2), it becomes decisive that $\mathcal{G}$ endows the potentials in $\mathcal{F}_{coop}$ with structure. Before we explain how to exploit the structure, we remark that MPE inference for distributions in $\mathcal{F}_{coop}$ is equivalent to cooperative cut in $\mathcal{G}$, thanks to Eq. 2. Thus, we strive to solve the following problem:

$$\min f(C) \quad \text{s.t. } C \subseteq \mathcal{E} \text{ is an } (s,t)\text{-cut in } \mathcal{G}. \qquad (3)$$

For ease of notation, we proceed using the cut formulation (3). Since the cut cost $f(C)$ is equivalent to the potential, all guarantees transfer to the potential.

1.2. Preliminaries and notation

We are given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $m = |\mathcal{E}|$ edges. We assume the submodular cost function $f : 2^{\mathcal{E}} \to \mathbb{R}_+$ to be normalized, $f(\emptyset) = 0$, and nondecreasing: if $A \subseteq B$, then $f(A) \leq f(B)$. We note that a nonnegative submodular set function is also *subadditive*: $f(A) + f(B) \geq f(A \cup B)$. A *matroid rank function* is an integral submodular function with $f(e) \in \{0, 1\}$ for all $e \in \mathcal{E}$. The convolution of two submodular functions f, g is defined as $(f * g)(B) = \min_{A \subseteq B} f(A) + g(B \setminus A)$. More details about submodular functions can be found in (Fujishige, 2005). We denote the feasible set of all cuts by $\mathcal{C} \subset 2^{\mathcal{E}}$. Sometimes, we consider a set $A \subseteq \mathcal{E}$ by its indicator $\chi_A \in \{0,1\}^{\mathcal{E}}$, where $\chi_A(e) = 1$ if and only if $e \in A$. In the sequel, $C^* = \operatorname{argmin}_{C \in \mathcal{C}} f(C)$ is the optimal solution of Problem (3). An algorithm with approximation factor $\alpha \geq 1$ finds a solution $\widehat{C}$ that satisfies $f(\widehat{C}) \leq \alpha f(C^*)$. For simplicity, we state the results mostly in terms of directed graphs – they do extend to undirected graphs as well, however.

In the next several sections, we give a variety of algorithms that approximately solve Problem (3) and also provide approximation bounds. In Section 3, which gives a lower bound for approximability, we see that there can be no constant factor approximation for this problem. On the other hand, the class of problems is still within the realm of approximability, unlike the more general case mentioned in Section 1.

2. Algorithms

We aim for approximation algorithms for Problem (3). First, we build on a generalized maxflow-mincut duality, and later show two alternative techniques. The first algorithm differs from any of the algorithms for related submodular-cost problems. The main idea is to replace f by a tractable approximation $\hat{f}$ whose deviation from f is limited. We will use the following lemma.

Lemma 1. *Let* $\widehat{C} \in \operatorname{argmin}_{C \in \mathcal{C}} \hat{f}(C)$ *for an approximation* $\hat{f}$ *of* f*, with* $f(A) \leq \hat{f}(A)$ *for all* $A \subseteq \mathcal{E}$*, and* $\hat{f}(C^*) \leq \alpha f(C^*)$ *for* $C^* \in \operatorname{argmin}_{C \in \mathcal{C}} f(C)$*. Then* $f(\widehat{C}) \leq \alpha f(C^*)$*.*

Proof. Since $\hat{f}(\widehat{C}) \leq \hat{f}(C^*)$, we have that $f(\widehat{C}) \leq \hat{f}(\widehat{C}) \leq \hat{f}(C^*) \leq \alpha f(C^*)$. □

Lemma 1 immediately gives a bound for the simple linearization $\hat{f}_{\text{add}}(A) = \sum_{e \in A} f(e)$. Thanks to the subadditivity of f, $f(A) \leq \hat{f}_{\text{add}}(A)$. To derive α, consider the extreme case of a label cost where all edges have the same label. Then $f(A) = 1$ for all $A \subseteq \mathcal{E}$, but $\hat{f}_{\text{add}}(A) = |A|$. Thus, α can be as large as $|C^*| = O(m)$.

2.1. Approximation with polymatroidal network flows

We now find a tractable approximation $\hat{f}$ of f that is better than $\hat{f}_{\text{add}}$. Note that Problem (3) is hard because f is globally non-separable: the cost of remote edges e_1, e_2 can interact, so that $f(\{e_1, e_2\}) \ll f(e_1) + f(e_2)$. In contrast, the standard minimum cut with a *separable* sum of edge weights is solvable efficiently.

Therefore, we design $\hat{f}$ to be globally separable, but a locally tight approximation. To measure the cost

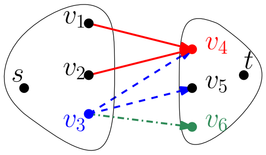


Figure 3. Approximation of a cut cost. Red edges are in $C^\Pi_{v_4}$ (head), blue dashed edges in $C^\Pi_{v_3}$ (tail), and the green dash-dotted edge in $C^\Pi_{v_6}$ (head).

of an edge set $C \subseteq \mathcal{E}$, we partition C into groups $\Pi(C) = \{C^\Pi_v\}_{v\in V}$, where the edges in C^Π_v must be incident to v. That is, we assign each edge either to its head or to its tail node (Fig. 3). Let $\mathcal{P}_C$ be the family of all such partitions (which vary in the head or tail assignment of each edge). We define an approximation

$$\hat{f}(C) = \min_{\Pi(C)\in\mathcal{P}_C} \sum\nolimits_{v\in V} f(C^\Pi_v) \tag{4}$$

that decomposes across node neighborhoods, but is accurate within a group C^Π_v. Thanks to the subadditivity of f, $\hat{f}$ is an upper bound on f, and we use the tightest possible such approximation. Instead of Problem (3), we now solve a different optimization problem:

$$\min \hat{f}(C) \;\text{ s.t. } C \subseteq \mathcal{E} \;\text{ is an } (s,t)\text{-cut}. \tag{5}$$

To solve Problem (5) exactly, we use its analogy to a generalized maxflow problem. This analogy only holds for cuts, but that suffices here. We first introduce the flow problem.

2.1.1. Polymatroidal network flows

Polymatroidal network flows (Lawler & Martel, 1982) generalize the capacity of traditional flow problems. A function $\varphi : E \to \mathbb{R}_+$ is a flow if the inflow at each node $v \in \mathcal{V} \setminus \{s,t\}$ equals the outflow, and if the flow on an edge does not exceed its capacity: $\varphi(e) \leq \mathrm{cap}(e)$ for all $e \in \mathcal{E}$, given a capacity function $\mathrm{cap} : \mathcal{E} \to \mathbb{R}_+$. Polymatroidal flows replace the usual additive capacities by submodular ones at each node v: $\mathrm{cap}^{\mathrm{in}}_v$ for incoming edges, and $\mathrm{cap}^{\mathrm{out}}_v$ for outgoing edges. Let $\delta^- v$ be the incoming edges of v, and $\delta^+ v$ its outgoing edges. Then the capacity constraints are, at each $v \in \mathcal{V}$:

$$\varphi(A) \leq \mathrm{cap}^{\mathrm{in}}_v(A) \qquad \text{for all } A \subseteq \delta^- v,$$

and equivalently for $\mathrm{cap}^{\mathrm{out}}_v$ on $\delta^+ v$. The maximum flow with such constraints is solved exactly in polynomial time by a layered augmenting paths algorithm (Tardos et al., 1986). The algorithm involves submodular function minimization (SFM) only on the sets $\delta^+ v$, $\delta^- v$ that are much smaller than $\mathcal{E}$. It takes time $O(m^4 T)$, where T is the time for SFM on any $\delta^+ v$, $\delta^- v$.

2.1.2. Analogy

The next lemma relates Problem (5) to polymatroidal flows. For ease of notation, we explicitly write restrictions here, but drop them later.

Lemma 2. *Minimum (s,t)-cut with cost function $\hat{f}$ is dual to a polymatroidal network flow with capacities $\mathrm{cap}^{in}_v = f|_{\delta^- v}$ and $\mathrm{cap}^{out}_v = f|_{\delta^+ v}$ at each node $v \in \mathcal{V}$.*

Proof. First, we restate the dual of a polymatroidal flow. Let $\mathrm{cap}^{\mathrm{in}} : 2^{\mathcal{E}} \to \mathbb{R}_+$ be the joint incoming capacity, $\mathrm{cap}^{\mathrm{in}}(C) = \sum_{v\in V} \mathrm{cap}^{\mathrm{in}}_v(C \cap \delta^- v)$, and equivalently $\mathrm{cap}^{\mathrm{out}}$ the joint outgoing capacity. The dual of the polymatroidal maxflow is a mincut problem whose cost is a convolution of edge capacities: $\mathrm{cap}(C) = (\mathrm{cap}^{\mathrm{in}} * \mathrm{cap}^{\mathrm{out}})(C) \triangleq \min_{A\subseteq C} \mathrm{cap}^{\mathrm{in}}(A) + \mathrm{cap}^{\mathrm{out}}(C \setminus A)$ (Lovász, 1983).

We relate this dual to our approximation $\hat{f}$. Given a minimal[1] (s,t)-cut C, let $\Pi(C)$ be a partition of C, and $C^{\mathrm{in}}_v = C^\Pi_v \cap \delta^-_v$, $C^{\mathrm{out}}_v = C^\Pi_v \cap \delta^+_v$. Since C is a minimal directed cut, it contains only edges from the s side to the t side of the graph. In consequence, $C^{\mathrm{in}}_v = \emptyset$ if v is on the s side, and $C^{\mathrm{out}}_v = \emptyset$ otherwise. Hence, $f(C^{\mathrm{in}}_v \cup C^{\mathrm{out}}_v) = f(C^{\mathrm{in}}_v) + f(C^{\mathrm{out}}_v)$. Then

$$\begin{aligned}
\hat{f}(C) &= \min_{\Pi(C)\in\mathcal{P}_C} \sum\nolimits_{v\in\mathcal{V}} f(C^\Pi_v) && (6)\\
&= \min_{\Pi(C)\in\mathcal{P}_C} \sum\nolimits_{v\in\mathcal{V}} f(C^{\mathrm{in}}_v \cup C^{\mathrm{out}}_v) && (7)\\
&= \min_{\{(C^{\mathrm{in}}_v, C^{\mathrm{out}}_v)\}_v} \sum\nolimits_{v\in\mathcal{V}} (f(C^{\mathrm{in}}_v) + f(C^{\mathrm{out}}_v)) && (8)\\
&= \min_{\{(C^{\mathrm{in}}_v, C^{\mathrm{out}}_v)\}_v} \sum_{v\in\mathcal{V}} (\mathrm{cap}^{\mathrm{in}}_v(C^{\mathrm{in}}_v) + \mathrm{cap}^{\mathrm{out}}_v(C^{\mathrm{out}}_v)) &&\\
&= \min_{C^{\mathrm{in}}, C^{\mathrm{out}}} (\mathrm{cap}^{\mathrm{in}}(C^{\mathrm{in}}) + \mathrm{cap}^{\mathrm{out}}(C^{\mathrm{out}})) && (9)\\
&= \min_{C^{\mathrm{in}}\subseteq C} (\mathrm{cap}^{\mathrm{in}}(C^{\mathrm{in}}) + \mathrm{cap}^{\mathrm{out}}(C \setminus C^{\mathrm{in}})) && (10)\\
&= (\mathrm{cap}^{\mathrm{in}} * \mathrm{cap}^{\mathrm{out}})(C). && (11)
\end{aligned}$$

Eq. 6 is the definition of $\hat{f}$. The minimum in Eq. 8 is taken over all feasible partitions $\Pi(C)$ and their intersections with the $\delta^+ v, \delta^- v$. Then we use the notation $C^{\mathrm{in}} = \bigcup_{v\in\mathcal{V}} C^{\mathrm{in}}_v$ for all edges assigned to their head nodes, and $C^{\mathrm{out}} = \bigcup_{v\in\mathcal{V}} C^{\mathrm{out}}_v$. The minima in Eqs. 9 and 10 are again over all partitions in $\mathcal{P}_C$. The final equality follows from the above definition of a convolution of submodular functions. □

2.1.3. Approximation factor

Section 2.1.2 shows that Problem (5) can be solved exactly. With Lemma 1, we bound the approximation factor by a quantity that depends on the graph

[1] If a cut C is minimal, then no subset $A \subset C$ is a cut.

structure. Let C^* be the optimal cut for cost f. We define Δ_s to be the tail nodes of the edges in C^*: $\Delta_s = \{v \in \mathcal{V} \mid \exists (v,u) \in C^*\}$. These are still reachable from s. Similarly, Δ_t contains all nodes on the t side that are the head of an edge in C^*.

Theorem 3. *Let $\widehat{C}$ be the minimum cut for cost $\hat{f}$, and C^* the optimal cut for cost f. Then $f(\widehat{C}) \leq \min\{|\Delta_s|, |\Delta_t|\} f(C^*) \leq |\mathcal{V}| f(C^*)/2$.*

Proof. To use Lemma 1, we need to show that $f(C) \leq \hat{f}(C)$ for all $C \subseteq \mathcal{E}$, and find an α such that $\hat{f}(C^*) \leq \alpha f(C^*)$. We already argued for the first condition using subadditivity. It remains to bound α. We do so by referring to the flow analogy with capacities set to f:

$$\begin{aligned}
\hat{f}(C^*) &= (\mathrm{cap}^{\mathrm{in}} * \mathrm{cap}^{\mathrm{out}})(C^*) && (12)\\
&\leq \min\{\mathrm{cap}^{\mathrm{in}}(C^*),\ \mathrm{cap}^{\mathrm{out}}(C^*)\} && (13)\\
&\leq \min\Big\{ \textstyle\sum_{v\in\Delta_s} f(C^* \cap \delta^+ v),\ \ \sum_{v\in\Delta_t} f(C^* \cap \delta^- v)\Big\} &&\\
&\leq \min\Big\{ |\Delta_s| \max_{v\in\Delta_s} f(C^* \cap \delta^+ v),\ \ |\Delta_t| \max_{v\in\Delta_t} f(C^* \cap \delta^- v)\Big\} &&\\
&\leq \min\big\{\, |\Delta_s|,\ |\Delta_t| \,\big\}\ f(C^*). && (14)
\end{aligned}$$

Thus, Lemma 1 implies an approximation bound $\alpha \leq \min\big\{\, |\Delta_s|,\ |\Delta_t| \,\big\} \leq |\mathcal{V}|/2$. □

2.2. Alternative approximations

Minimizing $\hat{f}$ instead of f yields in general a good solution — on dense graphs, $n/2 = O(\sqrt{m})$. However, the approximation bound depends on the graph structure. Thus, we add three complementary algorithms.

2.2.1. Global approximation of f

Instead of $\hat{f}$ from Section 2.1, any other approximation of f can be used in Lemma 1, as long as it makes the minimum cut problem tractable. Goemans et al. (2009) approximate a submodular function by a square root $\hat{f}_{\mathrm{ell}}(C) = \sqrt{\sum_{e\in C} w_f(e)}$. This function stems from the submodular polyhedron. The *submodular polyhedron* is a subset of $\mathbb{R}^{\mathcal{E}}$ and defined as $P_f = \{x \in \mathbb{R}^{\mathcal{E}} \mid \sum_{e\in A} x(e) \leq f(A)\ \forall\ A \subseteq \mathcal{E}\}$. For the function f, it holds that

$$f(A) = \max\nolimits_{y\in P_f} y \cdot \chi_A. \tag{15}$$

Replacing P_f in (15) by a certain ellipsoid yields $\hat{f}_{\mathrm{ell}}$. Computing the ellipsoid, i.e., the weights w_f, is the bottleneck of this approximation, and takes $O(m^4 \log^2 m)$ time. For matroid rank functions, $\hat{f}_{\mathrm{ell}}$ guarantees an approximation factor $\alpha = \sqrt{m+1}$, and otherwise $\alpha = O(\sqrt{m}\log m)$. This leads to the following bound:

Lemma 4. *Let $\widehat{C} = \mathrm{argmin}_{C\in\mathcal{C}}\, \hat{f}_{ell}(C)$ be the minimum cut for cost $\hat{f}_{ell}$, and $C^* = \mathrm{argmin}_{C\in\mathcal{C}}\, f(C)$. Then $f(\widehat{C}) = O(\sqrt{m}\log m) f(C^*)$.*

Algorithm 1 Greedy randomized path cover

```
Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $s, t \in \mathcal{V}$, $f$
$C = \emptyset$, $x = 0$
while $\sum_{e\in P_{\min}} x(e) < 1$ for shortest path $P_{\min}$ do
  let $\beta \in (0, \min_{e\in P_{\min}} \rho_e(C)]$
  for $e$ in $P_{\min}$ do
    with probability $\beta/\rho_e(C)$, set $C = C \cup \{e\}$,
    $x(e) = 1$.
  end for
end while
prune $C$ to $C'$ and return $C'$
```

In comparison to $\hat{f}$, the approximation $\hat{f}_{\mathrm{ell}}$ is harder to compute, but the optimization is easier: minimizing $\hat{f}^2_{\mathrm{ell}}$ is equivalent to minimizing $\hat{f}_{\mathrm{ell}}$, and corresponds to a sum-of-weights minimum cut.

2.2.2. Cuts via greedy covers

Our final strategy relates cuts to covers. An (s,t)-cut is also a hitting set: a cut "hits" (intersects) or "covers" each (s,t)-path. Therefore, we write Problem (3) as

$$\begin{aligned}
\min\ \ & f(x) \\
\text{s.t.}\ \ & \textstyle\sum_{e\in P} x(e) \geq 1 \quad \forall (s,t)\text{-paths } P \subseteq \mathcal{E} \\
& x \in \{0,1\}^E.
\end{aligned} \tag{16}$$

Here, with a little abuse of notation, we write f as a function on binary indicator vectors, $f(\chi_A) = f(A)$. The constraints imply that Problem (16) is a minimum-cost cover problem. There can be exponentially many constraints, one for each path. Luckily, it is not hard to find a violated constraint. We merely compute the shortest path $P_{\min}$, using x as the edge lengths. If $P_{\min}$ is longer than one, then x is feasible, otherwise not.

Owing to the form of the constraints, we can adapt a randomized greedy cover-algorithm (Koufogiannakis & Young, 2009) to Problem (16) and obtain Algorithm 1. In each step, we compute the shortest path with weights x to find a possibly uncovered path. Ties are resolved arbitrarily. To cover the path, we randomly pick edges from $P_{\min}$, with probabilities inversely proportional to the marginal cost $\rho_e(C) \triangleq f(C \cup \{e\}) - f(C)$. We must also specify an appropriate β. With the maximum possible β we select the cheapest edge deterministically, and others randomly. To pick exactly one edge in each iteration, we set $\beta = (\sum_{e\in P_{\min}} \rho_e(C)^{-1})^{-1}$, and then sample one edge from $P_{\min}$, with probabilities $p(e) = \beta/\rho_e(C)$. Since C grows by at least one edge in each iteration, the algorithm terminates after at most m iterations.

Finally, the algorithm may return a set C that is feasible but not a minimal cut. Then we prune C to a minimal cut $C' \subseteq C$. Since f is nondecreasing, $f(C') \leq f(C)$. We assign infinite weight to all edges in $\mathcal{E} \setminus C$, and weight $f(e)$ to each edge $e \in C$ (or contract nodes accordingly). The standard minimum (s,t)-cut in the resulting graph is the desired C'.

The last important question is the approximation bound. Lemma 5 implies that Algorithm 1 returns at least an $O(n)$-approximation, because the longest path spans at most $|\mathcal{V}| - 1$ edges.

Lemma 5. *$\mathbb{E}[f(\widehat{C'})] \leq |P_{\max}| f(C^*)$, where $P_{\max}$ is the longest simple path in $\mathcal{G}$.*

Proof. We already argued that the pruned C' can only be better than C. By Theorem 7 in (Koufogiannakis & Young, 2009), a greedy randomized procedure like Algorithm 1 gives a Δ-approximation for a cover, where Δ is the maximum number of variables in any constraint. In (16), Δ is the maximum number of edges in any simple path, i.e., the length of the longest path. This implies that $f(C') \leq f(C) \leq |P_{\max}| f(C^*)$. □

2.2.3. Relaxation

An alternative to the greedy randomized algorithm is to solve a relaxation of Problem (16). For the relaxation, we need to extend f from a set function to a function on a continuous domain. We view f as a function on binary indicator vectors, $\{0,1\}^{\mathcal{E}}$, and extend it to $[0,1]^{\mathcal{E}}$ via its *Lovász extension* $\tilde{f} : [0,1]^{\mathcal{E}} \to \mathbb{R}_+$,

$$\tilde{f}(x) = \max\nolimits_{y \in P_f} y \cdot x.$$

The maximization over the submodular polyhedron P_f takes $O(m \log m)$ time (Edmonds, 1970). Furthermore, a submodular function satisfies $f(\chi_A) = \max_{y \in P_f} y \cdot \chi_A = \tilde{f}(\chi_A)$. The Lovász extension is convex and piecewise linear. We substitute $\tilde{f}$ for f in Program (16), and allow $x \in [0,1]^{\mathcal{E}}$. The result is a non-smooth convex program with exponentially many constraints. The constraints can be summarized by the $m+1$ constraints of a standard linear program for minimum cut (Papadimitriou & Steiglitz, 1998):

$$\begin{aligned} \min \quad & \tilde{f}(x) \\ \text{s.t.} \quad & x(e) \geq \pi(v) - \pi(u) \;\; \forall (u,v) \in \mathcal{E} \\ & \pi(t) - \pi(s) \geq 1 \\ & \pi \in [0,1]^{\mathcal{V}}, \; x \in [0,1]^{\mathcal{E}} \end{aligned} \tag{17}$$

The node variables π essentially indicate membership of a node in the s side ($\pi(v) = 0$) or t side ($\pi(v) = 1$) 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. At closer inspection, the label $\pi(v)$ indicates the length of the shortest path from s to v, measured by additive distances x. Program (17) can be solved using any solver for non-smooth convex problems, or by adapting the approach in (Chudak & Nagano, 2007).

The nonlinear Program (17) usually does not have an integral solution, and thus we must round appropriately. The rounding procedure, shown in Algorithm 2, will determine the approximation guarantee. Let x^* be the optimal solution of Program (17). We test the values of $x^*(e)$ as thresholds θ_i in decreasing order (or by binary search). If the set C_i of edges e with $x^*(e) \geq \theta_i$ contains a cut, we stop and prune C_i to a minimal cut.

Algorithm 2 Rounding procedure given x^*

```
order $\mathcal{E}$ such that $x^*(e_1) \geq x^*(e_2) \geq \ldots \geq x^*(e_m)$
for $i = 1, \ldots, m$ do
  let $C_i = \{e_j \mid x^*(e_j) \geq x^*(e_i)\}$
  if $C_i$ is a cut then
    prune $C_i$ to $\widehat{C}$ and return $\widehat{C}$
  end if
end for
```

A faster, cruder rounding uses a threshold that is at most as large as the inverse of the length of the longest path in the graph (threshold $(n-1)^{-1}$ always works). The reason for this quantity becomes clear in the proof of the following lemma, the approximation bound.

Lemma 6. *Let $\widehat{C}$ be the rounded solution returned by Algorithm 2, 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. Program (16) is a submodular covering program. Thus, thresholded rounding is possible, similar to the case of cover problems (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

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

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, $x^*(e) \geq \theta \chi_{C_i}(e)$ for all $e \in E$, and hence $\chi_{C_i}(e) \leq \theta^{-1} x^*(e)$. 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 $\theta^{-1} \leq |P_{\max}|$. □

2.3. Discussion

We presented four methods to solve Problem (3). Beyond inference, a special case of Problem (3) arises is the analysis of attack graphs in computer security. Zhang et al. (2011) propose an algorithm for this special case, but their method does not apply to general nondecreasing submodular functions. Other applications are based on mean-risk minimization in discrete stochastic optimization (Atamtürk & Narayanan, 2008).

Which of our algorithms performs best depends on the problem at hand. At first sight, all guarantees might appear as $O(\sqrt{m})$ or $O(n)$ for $n = |\mathcal{V}|$, and almost equivalent. Still, the exact structural terms can make a difference. For sparse graphs with $m = O(n)$, the approximation $\hat{f}_{ell}$ is theoretically the best. On dense graphs, the flow-based approximation dominates theoretically. As an illustrative example, consider a chain of $\sqrt{n}$ cliques between s and t, each clique consisting of roughly $\sqrt{n}$ nodes. Two adjacent cliques intersect at one node. Then the longest path has length $n - 1$, whereas $|\Delta_s| \leq \sqrt{n}$ and $\sqrt{m} \approx n^{3/4}$. In any case, it is important to note that the theoretical factors are *worst-case* approximation bounds – on many examples, the algorithms perform much better, as we demonstrate in the next section.

From an implementation viewpoint, the greedy cover is the simplest, and often fast. Since it is randomized, its solution quality in single runs can vary. Often, a heuristic to include the edge with the lowest marginal cost works well, too.

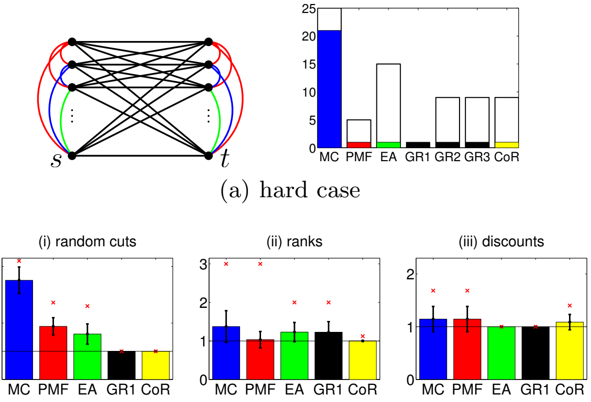
2.4. Experiments

As a proof of concept, Figure 4(a) shows an example graph with coupled edges. It is a complete graph, and the minimum cut contains the maximum possible number of edges. We compare the described algorithms to a minimum cut with $\hat{f}_{\text{add}}$. The cost function is

$$f(A) = \mathbf{1}[|A \cap E_k| \geq 1] + \sum_{i=1}^{n/2-1} \frac{n}{2} \cdot \mathbf{1}[|A \cap E_i| \geq 1],$$

where E_k is the set of black edges, and the E_i are the other sets of edges with identical color.

The proposed algorithms all find the optimal solution. A standard minimum cut with $\hat{f}_{\text{add}}$ yields a solution with an approximation factor of $\Omega(n^2/4)$ – its worst case. The cost of its solution is larger than permissible with the approximation factors of the other algorithms. Thus, the example illustrates that approximation bounds do indeed matter.



(a) hard case

(b) average cases

Figure 4. Empirical approximation factors for (a) the shown structure graph ($n = 10$); (b) more common cases. MC: mincut with $\hat{f}_{\text{add}}$, PMF: $\hat{f}$ via polymatroidal flows, EA: approximation with $\hat{f}_{\text{ell}}$, GR: greedy cover (1) picking an edge with minimum marginal cost, (2) sampling one edge, (3) maximum β, CoR: convex relaxation. White bars in (a) indicate theoretical bounds where applicable, red crosses in (b) worst empirical results. (Figure best viewed in color.)

Most inputs, however, are more benign. Therefore, we show empirical approximation bounds on three other classes of cost functions on clustered graphs ($n = 30$, $m = 90$): (i) functions similar to the worst case, where the optimal cut was picked randomly and the function designed to make it optimal; (ii) matroid rank functions and sums thereof; (iii) concave functions (log and square root) of a sum of weights. Figure 4(b) shows averages over 45, 100 and 50 instances for computing the minimum cut by a sequence of (s, t)-cuts. The approximation factors are in general between 1 and 2, and much better than the theoretical bounds. For more detailed experiments, see (Jegelka & Bilmes, 2010).

3. Hardness

The approximation factors that we derived in the previous section are put into context by the following lower bound. It assumes oracle access to f.

Theorem 7. *No polynomial-time algorithm can solve Problem (3) to an approximation factor of* $o(\sqrt{m/\log m})$.

Theorem 7 implies that the best possible approximation factor in the general case is on the order of $\sqrt{|\mathcal{E}|}$. The proof is information-theoretic.

Proof. The key idea is to construct two submodular cost functions f, h with different minima that are almost indistinguishable. With high probability they cannot be discriminated within 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. Hence, a polynomial-time algorithm with an approximation factor α would lead to a contradiction. The proof technique is similar to (Goemans et al., 2009; Svitkina & Fleischer, 2008).

The function f depends on a hidden random set $R \subseteq \mathcal{E}$ that will be its optimal cut. Construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with ℓ parallel disjoint paths from s to t; each path has k edges. Let the random set $R \subset \mathcal{E}$ 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 $C \subseteq \mathcal{E}$,

$$h(C) = \min\{|C|, \ell\} \tag{18}$$

$$f(C) = \min\{|C \setminus R| + \min\{|C \cap R|,\ \beta\},\ \ell\}. \tag{19}$$

The functions differ only for the relatively few sets C with $|C \cap R| > \beta$ and $|C \setminus R| < \ell - \beta$. Define ε such that $\varepsilon^2 = \omega(\log m)$, and set $k = 8\sqrt{m}/\varepsilon$ and $\ell = \varepsilon\sqrt{m}$. By a Chernoff bound, one can show that the probability (over all choices of R) that f and h differ for a given C is very small:

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

By a union bound, the probability of distinguishing f and h with a polynomial number of queries C still vanishes as m grows.

As argued above, the bound will be the ratio of optima of h and f. The minimum cooperative-cost cut for f is R with $f(R) = \beta$, and h has uniform cost $h(C) = \ell$ for all minimal cuts C. Hence, the ratio is $h(R)/f(R) = \ell/\beta = \sqrt{m}/\varepsilon = o(\sqrt{m/\log m})$. □

Acknowledgments. We thank Jens Vygen for the example of a very hard subadditive function.

References

Abdelbar, A.M. and Hedetniemi, S.M. Approximating MAPs on belief networks is NP-hard and other theorems. *Artificial Intelligence*, 102, 1998.

Atamtürk, A. and Narayanan, V. Polymatroids and mean-risk minimization in discrete optimization. *Operations Research Letters*, 36(5):618–622, 2008.

Boykov, Y. and Jolly, M.-P. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *Proc. of the Int. Conf. on Computer Vision (ICCV)*, 2001.

Chandrasekaran, V., Srebro, N., and Harsha, P. Complexity of inference in graphical models. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2008.

Chudak, F. A. and Nagano, K. Efficient solutions to relaxations of combinatorial problems with submodular penalties via the Lovász extension and non-smooth convex optimization. In *Proc. of the ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2007.

Edmonds, J. *Combinatorial Structures and their Applications*, chapter Submodular functions, matroids and certain polyhedra, pp. 69–87. Gordon and Breach, 1970.

Fujishige, S. *Submodular Functions and Optimization.* Number 58 in Annals of Discrete Mathematics. Elsevier Science, 2nd edition, 2005.

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.

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

Jegelka, S. and Bilmes, J. Cooperative cuts: graph cuts with submodular edge weights. Technical Report TR-189, Max Planck Institute for Biological Cybernetics, 2010.

Jegelka, S. and Bilmes, J. Submodularity beyond submodular energies: coupling edges in graph cuts. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.

Kolmogorov, V. and Zabih, R. What energy functions can be minimized via graph cuts? *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

Koufogiannakis, C. and Young, N. E. Greedy Δ-approximation algorithm for covering with arbitrary constraints and submodular costs. In *Proc. of the Int. Colloquium on Automata, Languages and Programming (ICALP)*, 2009.

Lawler, E. L. and Martel, C. U. Computing maximal "Polymatroidal" network flows. *Mathematics of Operations Research*, 7(3):334–347, 1982.

Lovász, L. *Mathematical programming – The State of the Art*, chapter Submodular Functions and Convexity, pp. 235–257. Springer, 1983.

Papadimitriou, C. and Steiglitz, K. *Combinatorial Optimization.* Dover Publications, 1998.

Sheldon, D. et. al. Maximizing the Spread of Cascades Using Network Design. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2010.

Svitkina, Z. and Fleischer, L. Submodular approximation: Sampling-based algorithms and lower bounds. In *Proc. of the Ann. Symp. on Foundations of Computer Science (FOCS)*, 2008.

Tardos, E., Tovey, C. A., and Trick, M. A. Layered augmenting path algorithms. *Mathematics of Operations Research*, 11(2), 1986.

Zhang, P., J.-Y, Cai, Tang, L.-Q., and Zhao, W.-B. Approximation and hardness results for label cut and related problems. *Journal of Comb. Optim.*, 21(2):192–208, 2011.