# Improved Inapproximability for Submodular Maximization

Per Austrin*

Courant Institute of Mathematical Sciences
New York University
austrin@cims.nyu.edu

**Abstract.** We show that it is Unique Games-hard to approximate the maximum of a submodular function to within a factor 0.695, and that it is Unique Games-hard to approximate the maximum of a symmetric submodular function to within a factor 0.739. These results slightly improve previous results by Feige, Mirrokni and Vondrák (FOCS 2007) who showed that these problems are NP-hard to approximate to within $3/4 + \varepsilon \approx 0.750$ and $5/6 + \varepsilon \approx 0.833$, respectively.

## 1 Introduction

Given a ground set $U$, consider the problem of finding a set $S \subseteq U$ which maximizes some function $f : 2^U \to \mathbb{R}^+$ which is *submodular*, i.e., satisfies

$$f(S \cup T) + f(S \cap T) \leq f(S) + f(T).$$

for every $S, T \subseteq U$. The submodularity property is also known as the property of *diminishing returns*, since it is equivalent with requiring that, for every $S \subset T \subseteq U$ and $i \in U \setminus T$, it holds that

$$f(T \cup \{i\}) - f(T) \leq f(S \cup \{i\}) - f(S).$$

There has been a lot of attention on various submodular optimization problems throughout the years (e.g., [8,7,3], see also the first chapter of [10] for a more thorough introduction). Many natural problems can be cast in this general form – examples include natural graph problems such as maximum cut, and many types of combinatorial auctions and allocation problems.

A further restriction which is also very natural to study is *symmetric* submodular functions. These are functions which satisfy $f(S) = f(\overline{S})$ for every $S \subseteq U$, i.e., a set and its complement always have the same value. A well-studied example of a symmetric submodular maximization problem is the problem to find a maximum cut in a graph.

Since it includes familiar NP-hard problems such as maximum cut as a special case, submodular maximization is in general NP-hard, even in the symmetric

* Supported by NSF Expeditions grant CCF-0832795.

M. Serna et al. (Eds.): APPROX and RANDOM 2010, LNCS 6302, pp. 12–24, 2010.

case. As a side note, a fundamental and somewhat surprising result is that submodular *minimization* has a polynomial time algorithm [5].

To cope with this hardness, there has been much focus on efficiently finding good approximate solutions. We say that an algorithm is an $\alpha$-approximation algorithm if it is guaranteed to output a set $S$ for which $f(S) \geq \alpha \cdot f(S_{\mathrm{OPT}})$ where $S_{\mathrm{OPT}}$ is an optimal set. We also allow randomized algorithms in which case we only require that the expectation of $f(S)$ (over the random choices of the algorithm) is at least $\alpha \cdot f(S_{\mathrm{OPT}})$.

In many special cases such as the maximum cut problem, it is very easy to design a constant factor approximation (in the case of maximum cut it is easy to see that a random cut is a 1/2-approximation). For the general case of an arbitrary submodular functions, Feige et al. [3] gave a $(2/5-o(1))$-approximation algorithm based on local search, and proved that a uniformly random set is a 1/2-approximation for the symmetric case. The $(2/5-o(1))$-approximation has been slightly improved by Vondrák [9] who achieved a 0.41-approximation algorithm, which is currently the best algorithm we are aware of.

Furthermore, [3] proved that in the *(value) oracle model* (where the submodular function to be maximized is given as a black box), no algorithm can achieve a ratio better than $1/2 + \varepsilon$, even in the symmetric case. However, this result says nothing about the case when one is given an *explicit representation* of the submodular function – say, a graph in which one wants to find a maximum cut. Indeed, in the case of maximum cut there is in fact a 0.878-approximation algorithm, as given by a famous result of Goemans and Williamson [4]. In the explicit representation model, the best current hardness results, also given by [3], are that it is NP-hard to approximate the maximum of a submodular function to within $3/4 + \varepsilon$ in the general case and $5/6 + \varepsilon$ in the symmetric case.

### 1.1 Our Results

In this paper we slightly improve the inapproximability results of [3]. However, as opposed to [3] we do not obtain NP-hardness but only hardness assuming Khot's *Unique Games Conjecture* (UGC) [6]. The conjecture asserts that a problem known as Unique Games, or Unique Label Cover, is very hard to approximate. See e.g. [6] for more details. While the status of the UGC is quite open, our results still imply that obtaining efficient algorithms that beat our bounds would require a fundamental breakthrough.

For general submodular functions we prove the following theorem.

**Theorem 1.** *It is UG-hard to approximate the maximum of a submodular function to within a factor* 0.695*.*

In the case of symmetric functions we obtain the following bound.

**Theorem 2.** *For every $\varepsilon > 0$ it is UG-hard to approximate the maximum of a symmetric submodular function to within a factor $709/960 + \varepsilon < 0.739$.*

These improved inapproximability results still fall short of coming close to the 1/2-barrier in the oracle model. Unfortunately, while marginal improvments of our results may be possible, we do not believe that our approach can come close to a factor 1/2. It remains a challenging and interesting open question to determine the exact approximability of explicitly represented submodular functions.

### 1.2 Our Approach

As in [3], the starting point of our approach is hardness of approximation for constraint satisfaction problems (CSPs), an area which, due to much progress during the last 15 years, is today quite well understood. Here it is useful to take a slightly different viewpoint. Instead of thinking of the family of subsets $2^U$ of $U$, we consider the set of binary strings $\{0,1\}^n$ of length $n = |U|$, indentified with $2^U$ in the obvious way. These views are of course equivalent and throughout the paper we shift between them depending on which view is the most convenient.

For a string $x \in \{0,1\}^n$ and a $k$-tuple $C \in [n]^k$ of indices, let $x_C \in \{0,1\}^k$ denote the string of length $k$ which, in position $j \in [k]$ has the bit $x_{C_j}$. Now, given a function $f : \{0,1\}^k \to \mathbb{R}^+$, we define the problem Max CSP$^+(f)$ as follows. An instance of Max CSP$^+(f)$ consists of a list of $k$-tuples of variables $C_1, \ldots, C_m \in [n]^k$. These specify a function $F : \{0,1\}^n \to \mathbb{R}^+$ by

$$F(x) = \frac{1}{m} \sum_{i=1}^{m} f(x_{C_i})$$

and the problem is to find an $x \in \{0,1\}^n$ to maximize $x$.

Note that if $f$ is submodular then every instance $F$ of Max CSP$^+(f)$ is submodular and Max CSP$^+(f)$ is a special case of the submodular maximization problem.

Next, we use a variation of a result by the author and Mossel [2]. The result of [2] is for CSPs where one allows *negated literals*[1], which can not be allowed in the context of submodular maximization. However, in Theorem 3 we give a simple analogue of the result of [2] for the Max CSP$^+(f)$ setting.

Roughly speaking the hardness result says the following. Suppose that there is a *pairwise independent* distribution $\mu$ such that the expectation of $f$ under $\mu$ is at least $c$, but that the expectation of $f$ under the *uniform* distribution is at most $s$. Then Max CSP$^+(f)$ is UG-hard to approximate to within a factor of $s/c$.

The hardness result suggests the following natural approach: take a pairwise independent distribution $\mu$ with small support, and let $\mathbf{1}_\mu : \{0,1\}^k \to \{0,1\}$ be the indicator function of the support of $\mu$. Then take $f$ to be a "minimum submodular upper bound" to $\mathbf{1}_\mu$, by which we mean a submodular function satisfying $f(x) \geq \mathbf{1}_\mu(x)$ for every $x$ while having small expectation under the uniform distribution.

[1] Where each "constraint" $f(x_{C_i})$ of $F$ is of the more general form $f(x_{C_i} + l_i)$ for some $l_i \in \{0,1\}^k$, where $+$ is interpreted as addition over $GF(2)^k$.

To make this plan work, there are a few small technical complications (hidden in the "roughly speaking" part of the description of the hardness result above) that we need to overcome, making the final construction slightly more complicated. Unfortunately, understanding the "minimum submodular upper bound" of the families of indicator functions that we use appears difficult, and to obtain our results, we resort to explicitly computing the resulting submodular functions for small $k$.

Let us compare our approach with that of [3]. As mentioned above, their starting point is also hardness of approximation for constraint satisfaction. However, here their approach diverges from ours: they construct a gadget reduction from the $k$-Lin problem (linear equations mod 2 where each equation involves only $k$ variables). This gadget introduces two variables $x_i^0$ and $x_i^1$ for every variable $x_i$ in the $k$-Lin instance, where $x_i^j$ is intended to be an indicator of the event that $x_i = j$. Each equation $x_{i_1} \oplus \ldots \oplus x_{i_k} = b$ is replaced by some submodular function $f$ on the $2k$ new variables corresponding to the $x_{i_j}$'s. The analysis then has to make sure that there is always an optimal assignment where for each $i$ exactly one of $x_i^0$ and $x_i^1$ equals 1, which for the inapproximability of 3/4 becomes quite delicate. In our approach, which we feel is more natural and direct, we don't run into any such issues.

### 1.3 Organization

In Section 2 we set up some more notation that we use throughout the paper and give some additional background. In Section 3 we describe the hardness result that is our starting point. In Section 4 we describe in more detail the construction outlined above, and finally, in Section 5, we describe how to obtain the concrete bounds given in Theorems 1 and 2.

## 2 Notation and Background

Throughout the paper, we identify binary strings in $\{0,1\}^n$ and subsets of $[n]$ in the obvious way. Analogously to the notation $|S|$ and $\overline{S}$ for the cardinality and complement of a subset $S \subseteq [n]$ we use $|x|$ and $\overline{x}$ for the Hamming weight and coordinatewise complement of a string $x \in \{0,1\}^n$.

### 2.1 Submodularity

Apart from the two definitions in the introduction, a third characterization of submodularity is that a function $f : 2^X \to \mathbb{R}^+$ is submodular if and only if

$$f(S) - f(S \cup \{i\}) - f(S \cup \{j\}) + f(S \cup \{i\} \cup \{j\}) \leq 0 \qquad (1)$$

for every $S \subseteq X$, and $i, j \in X \setminus S$, $i \neq j$. It is straightforward to check that this condition is equivalent to the diminishing returns property mentioned in the introduction.

## 2.2 Probability

For $p \in [0,1]$, we use $\{0,1\}^k_{(p)}$ to denote the $k$-dimensional boolean hypercube with the $p$-biased product distribution, i.e., if $x$ is a sample from $\{0,1\}^k_{(p)}$ then the probability that the $i$'th coordinate $x_i = 1$ is $p$, independently for each $i \in [k]$.

We abuse notation somewhat by making no distinction between probability distribution functions $\mu : \{0,1\}^k \to [0,1]$ and the probability space $(\{0,1\}^k, \mu)$ for such $\mu$. Hence we write, e.g., $\mu(x)$ for the probability of $x \in \{0,1\}^k$ under $\mu$ and $\mathbb{E}_{x\sim\mu}[f(x)]$ for the expectation of a function $f : \{0,1\}^k \to \mathbb{R}$ under $\mu$.

A distribution $\mu$ over $\{0,1\}^k$ is *balanced pairwise independent* if every two-dimensional marginal distribution of $\mu$ is the uniform distribution, or formally, if for every $1 \le i < j \le n$ and $b_1, b_2 \in \{0,1\}$, it holds that

$$\Pr_{x\sim\mu}[x_i = b_1 \wedge x_j = b_2] = 1/4.$$

Recall that the support $\mathrm{Supp}(\mu)$ of a distribution $\mu$ over $\{0,1\}^k$ is the set of strings with non-zero probability under $\mu$, i.e., $\mathrm{Supp}(\mu) = \{\, x \in \{0,1\}^k \,:\, \mu(x) > 0 \,\}$.

We conclude this section with a lemma that will be useful to us.

**Lemma 1.** *Let $f : \{0,1\}^k \to \mathbb{R}^+$ be a symmetric set function. For $t \in \{0,\ldots,k\}$ let $a(t)$ denote the average of $f$ on strings of weight $t$, $a(t) = \frac{1}{\binom{k}{t}} \sum_{|x|=t} f(x)$. If $a$ is monotonely nondecreasing in $\{0,\ldots,k/2\}$, then the maximum average of $f$ under any $p$-biased distribution is achieved by the uniform distribution. I.e.,*

$$\max_{p\in[0,1]} \mathop{\mathbb{E}}_{x\sim\{0,1\}^k_{(p)}}[f(x)] = 2^{-x} \sum_{x\in\{0,1\}} f(x)$$

This intuitively obvious lemma is probably well known but as we do not know a reference we give a proof in the full version of this paper [1].

# 3 Hardness from Pairwise Independence

In this section we state formally the variation of the hardness result of [2] that we use. We first define the parameters which control the inapproximability ratio that we obtain.

**Definition 1.** *Let $f : \{0,1\}^k \to \mathbb{R}^+$ be a submodular function.*

*We define the* completeness $c_\mu(f)$ of $f$ with respect to a distribution $\mu$ *over $\{0,1\}^k$ by the expected value of $f$ under $\mu$, i.e.,*

$$c_\mu(f) := \mathop{\mathbb{E}}_{x\sim\mu}[f(x)]$$

*We define the* soundness $s_p(f)$ of $f$ with respect to bias $p$ *by the expected value of $f$ under the $p$-biased distribution, i.e.,*

$$s_p(f) := \mathop{\mathbb{E}}_{x\sim\{0,1\}^k_{(p)}}[f(x)].$$

*Finally, we define the* soundness $s(f)$ of $f$ *by its maximum soundness with respect to any bias, i.e.,*

$$s(f) := \max_{p \in [0,1]} s_p(f)$$

We can now state the hardness result.

**Theorem 3.** *Let $\mu$ be a balanced pairwise independent distribution over $\{0,1\}^k$. Then for every function $f : \{0,1\}^k \to \mathbb{R}^+$ and $\varepsilon > 0$, given a* Max CSP$^+(f)$ *instance $F : \{0,1\}^n \to \mathbb{R}^+$ it is UG-hard to distinguish between the cases:*

**Yes:** *There is an $S \subseteq X$ such that $F(S) \geq c_\mu(f) - \varepsilon$.*
**No:** *For every $S \subseteq X$ it holds that $F(S) \leq s(f) + \varepsilon$.*

The proof of Theorem 3 follows the proof of [2] almost exactly. A proof can be found in the full version of this paper [1].

Consequently, for any submodular function $f$ and pairwise independent distribution $\mu$ with all marginals equal, it is UG-hard to approximate Max CSP$^+(f)$ to within a factor $s(f)/c_\mu(f)+\varepsilon$ for every $\varepsilon > 0$. Note also that the **No** case is the best possible: there is a trivial algorithm which finds a set such that $F(S) \geq s(f)$ for every $F$, by simply letting each input be 1 with probability $p$ for the $p$ that maximizes $s_p(f)$.

As a somewhat technical remark, we mention that Theorem 3 still holds if $\mu$ is not required to be balanced – it suffices that all the one-dimensional marginal probabilities $\Pr_{x \sim \mu}[x_i = 1]$ are identical, not necessarily equal to $1/2$ as in the balanced case. We state the somewhat simpler form since that is sufficient to obtain our results for submodular functions and since that makes it more similar to the result of [2], which requires the distribution $\mu$ to be balanced.

Let us then briefly discuss the difference between Theorem 3 and the main result of [2]. First, the result of [2] only applies in the more general setting when one allows negated literals, which is why it can not be used to obtain inapproximability for submodular functions. On the other hand, this more general setting allows for a stronger conclusion: in the **No** case, [2] achieves a soundness of $s_{1/2}(f) + \varepsilon$ which in general can be much smaller than $s(f)$. As an example, consider the case when $f : \{0,1\}^3 \to \{0,1\}$ is the logical OR function on 3 bits. In this case the Max CSP$^+(f)$ problem is of course trivial – the all-ones assignment satisfies all constraints – and $s(f) = 1$, whereas $s_{1/2}(f) = 7/8$. Letting $\mu$ be the uniform distribution on strings of odd parity (it is readily verified that this is a balanced pairwise independent distribution) one gets $c_\mu(f) = 1$, showing that the Max 3-Sat problem is hard to approximate to within $7/8 + \varepsilon$.

## 4 The Construction

In this section we make formal the construction outlined in Section 1.2.

Theorem 3 suggests the following natural approach: pick a pairwise independent distribution $\mu$ over $\{0,1\}^k$ and let $\mathbf{1}_\mu : \{0,1\}^k \to \{0,1\}$ be the indicator function of the support of $\mu$. Then take $f$ to be a "minimum submodular upper bound" to $\mathbf{1}_\mu$, by which we mean a submodular function satisfying $f(x) \geq \mathbf{1}_\mu(x)$ for every $x$ while having $s(f)$ as small as possible (whereas $c_\mu(f)$ is clearly at least 1). Note that the smaller the support of $\mu$, the less constrained $f$ is, meaning that there should be more room to make $s(f)$ small.

To this end, let us make the following definition.

**Definition 2.** *For a subset $\mathcal{C} \subseteq \{0,1\}^k$, we denote by $\mathsf{SM}(\mathcal{C})$ the optimum function $f : \{0,1\}^k \to \mathbb{R}^+$ of the following program*[2]*:*

$$\begin{aligned} &\textit{Minimize } s(f) \\ &\textit{Subject to } f(x) \geq 1 \textit{ for every } x \in \mathcal{C} \\ &\qquad\qquad f \textit{ is submodular} \end{aligned}$$

*In addition, we write $\mathsf{SM}_p(\mathcal{C})$ for the optimal $f$ when the objective to be minimized is changed to $s_p(f)$ instead of $s(f)$. Analogously, we define $\mathsf{SM}^{\mathsf{sym}}(\mathcal{C})$ and $\mathsf{SM}_p^{\mathsf{sym}}(\mathcal{C})$ as the optimal $f$ with the additional restriction that $f$ is symmetric.*

While the objective function $s(f)$ is not linear (or even convex), it turns out that for the $\mathcal{C}$'s that we are interested in, $\mathsf{SM}(\mathcal{C})$ is actually quite well approximated by $\mathsf{SM}_{1/2}(\mathcal{C})$, i.e., we simply minimize $\sum_x f(x)$ (in fact, we even believe that for our $\mathcal{C}$'s $\mathsf{SM}_{1/2}(\mathcal{C})$ gives the exact optimum for $\mathsf{SM}(\mathcal{C})$, though we have not attempted to prove it). The advantage of considering $\mathsf{SM}_{1/2}(\mathcal{C})$ is of course that it is given by a linear program, which gives us a reasonably efficient way of finding it. Armed with this definition, let us now describe the constructions we use.

### 4.1 The Asymmetric Case

The family of pairwise independent distributions $\mu$ that we consider is a standard construction based on the Hadamard code. Fix a parameter $l > 0$ and let $k = 2^l - 1$. We identify the set of coordinates $[k]$ with the set of non-empty subsets of $[l]$, in some arbitrary way. A string $x$ from the distribution $\mu$ is sampled as follows: pick a uniformly random string $y \in \{0,1\}^l$ and defining, for each $\emptyset \neq T \subseteq [l]$, the coordinate $x_T = \bigoplus_{i \in T} y_i$.

This construction already has an issue: since the all-zeros string $\mathbf{0}$ is in the support of the distribution, any submodular upper bound to $\mathbf{1}_\mu$ must have $f(\mathbf{0}) \geq 1$, implying that $s_0(f) = 1$. To fix this, we simply ignore $\mathbf{0}$ when constructing $f$. Formally, let $\mathcal{C}_l = \mathrm{Supp}(\mu) \setminus \{\mathbf{0}\} \subseteq \{0,1\}^k$ be the $2^l - 1$ strings in the support of $\mu$ except $\mathbf{0}$. Now we would like to take our submodular function $f$ to be $\mathsf{SM}(\mathcal{C})$, but we instead take it to be $\mathsf{SM}_{1/2}(\mathcal{C})$, as this function is much more easily computed.

**Definition 3.** *For a parameter $l > 0$, let $k = 2^l - 1$ and take $\mathcal{C}_l \subseteq \{0,1\}^k$ as above. We define $f_l = \mathsf{SM}_{1/2}(\mathcal{C}_l)$.*

[2] In the case when the optimum is not unique, we choose an arbitrary optimal $f$ as $\mathsf{SM}(\mathcal{C})$.

Note that using only $\mathcal{C}_l$ instead of the entire support costs us a little in that the completeness is now reduced from 1 to $c_\mu(f_l) \geq 1 - 2^{-l}$, but one can hope (and it indeed turns out that this is the case) that this loss is compensated by a greater improvement in soundness.

Also, we stress that $s(f_l)$ is typically *not* given by the average $s_{1/2}(f_l)$ (which is the quantity actually minimized by $f_l$). Indeed, the points in $\mathcal{C}_l$ all have Hamming weight $(k+1)/2$ and this is also where $f_l$ is typically the largest. This causes $s(f)$ to be achieved by the $p$-biased distribution for some $p$ slightly larger than 1/2.

An obvious question to ask is whether using $\mathsf{SM}(\mathcal{C}_l)$ would give a better result than using $\mathsf{SM}_{1/2}(\mathcal{C}_l)$. For the values of $l$ that we have been able to handle, it appears that the answer to this question is negative: computing $\mathsf{SM}_p(\mathcal{C}_l)$ for a $p$ that approximately maximizes $s_p(f_l)$ gives $f_l$, indicating that we in fact have $f_l = \mathsf{SM}(\mathcal{C}_l)$.

### 4.2 Symmetric Functions

One way of constructing symmetric functions would be to use the exact same construction as above but taking $\mathsf{SM}^{\mathsf{sym}}(\mathcal{C}_l)$ rather than $\mathsf{SM}(\mathcal{C}_l)$. However, that is somewhat wasteful, and we achieve better results by also taking symmetry into account when constructing the family of strings $\mathcal{C}$.

Thus, we alter the above construction as follows: rather than identifying the coordinates with all non-empty subsets of $[l]$, we identify them with all subsets of $[l]$ of odd cardinality. In other words, we take $k = 2^{l-1}$ and associate $[k]$ with all $T \subseteq [l]$ such that $|T|$ is odd. The resulting distribution $\mu$ is symmetric in the sense that if $x$ is in the support then so is $\overline{x}$.

In this case, both the all-zeros string $\mathbf{0}$ and the all-ones string $\mathbf{1}$ are in the support which is not acceptable for the same reason as above. Hence, we construct a submodular function by taking $\mathcal{C}_l^{\mathsf{sym}} = \mathrm{Supp}(\mu) \setminus \{\mathbf{0}, \mathbf{1}\}$ (note that $|\mathcal{C}_l^{\mathsf{sym}}| = 2^l - 2$).

**Definition 4.** *For a parameter $l > 0$, let $k = 2^{l-1}$ and take $\mathcal{C}_l^{\mathsf{sym}} \subseteq \{0,1\}^k$ as above. We define $f_l^{\mathsf{sym}} = \mathsf{SM}_{1/2}^{\mathsf{sym}}(\mathcal{C}_l^{\mathsf{sym}})$.*

In this case, since we removed 2 out of the $2^l$ points of the support of $\mu$ to construct $\mathcal{C}_l^{\mathsf{sym}}$, we have that $c_\mu(f_l^{\mathsf{sym}}) \geq 1 - 2^{1-l}$.

An salient feature of $f_l^{\mathsf{sym}}$ is that all strings of $\mathcal{C}_l^{\mathsf{sym}}$ have Hamming weight exactly $k/2$. By Lemma 1, this causes $s_p(f_l^{\mathsf{sym}})$ to be maximized by $p = 1/2$ (the monotonicity of the function $a$ in Lemma 1 is not immediately clear). This means that in the symmetric case, using $\mathsf{SM}_{1/2}^{\mathsf{sym}}(\mathcal{C}_l^{\mathsf{sym}})$ rather than $\mathsf{SM}^{\mathsf{sym}}(\mathcal{C}_l^{\mathsf{sym}})$ is provably without loss of generality.

## 5 Concrete Bounds

Unfortunately, understanding the behaviour of the two families of functions $f_l$ and $f_l^{\mathsf{sym}}$ (or even just their soundnesses) for large $l$ appears difficult. There seems to be two conflicting forces at work: on the one hand, $\mathcal{C}_l$ only has $2^l - 1 = k$ points so even though $f_l$ is forced to be large on these there may still be plenty of room to make it small elsewhere. But on the other hand, since $\mathcal{C}_l$ is a good code the elements of $\mathcal{C}_l$ are very pread out (their pairwise Hamming distances are roughly $k/2$), which together with the submodularity condition appears to force $f_l$ to be large.

In this section we study $f_l$ for small $l$, obtaining our hardness results. As discussed towards the end of the section, there are indications that the inapproximability given by $f_l$ actually becomes worse for large $l$ and that our results are the best possible for this family of functions, but we do not yet know whether these indications are correct.

### 5.1 Symmetric Functions

We start with the symmetric functions, as these are somewhat nicer than the asymmetric ones in that their symmetry turn out to cause $s(f_l^{\mathsf{sym}})$ to be achieved by $p = 1/2$, i.e., $s(f_l^{\mathsf{sym}})$ simply equals the average of $f_l^{\mathsf{sym}}$. Table 1 gives a summary of the completeness, soundness, and inapproximability obtained by $f_l^{\mathsf{sym}}$ for $l \in \{3, 4, 5\}$. We now describe these functions in a more detail.

**Table 1.** Behaviour of $f_l^{\mathsf{sym}}$ for small $l$

| $l$ | $c$ | $s(f_l^{\mathsf{sym}})$ | Inapproximability $s/c$ |
|---|---|---|---|
| 3 | 3/4 | 5/8 | $5/6 < 0.8334$ |
| 4 | 7/8 | 43/64 | $43/56 < 0.7679$ |
| 5 | 15/16 | 709/1024 | $709/960 < 0.7386$ |

As a warmup, let us first describe the quite simple function $f_4^{\mathsf{sym}} : 2^{[8]} \to [0, 1]$ (we leave the even easier function $f_3^{\mathsf{sym}}$ to the interested reader). Its definition is as follows:

$$f_4^{\mathsf{sym}}(S) = \begin{cases} f(\overline{S}) & \text{if } |S| > 4 \\ |S|/4 & \text{if } |S| < 4 \\ 1 & \text{if } |S| = 4 \text{ and } S \text{ is in } \mathcal{C}_4^{\mathsf{sym}} \\ 3/4 & \text{otherwise} \end{cases} .$$

That $f_4^{\mathsf{sym}}(S)$ is submodular is easily verified. It is also easy to check that Lemma 1 applies and therefore we have that $s(f_4^{\mathsf{sym}}) = s_{1/2}(f_4^{\mathsf{sym}})$, which is straightforward to compute (note that $|\mathcal{C}_4^{\mathsf{sym}}| = 14$):

$$s_{1/2}(f_4^{\mathsf{sym}}) =$$
$$2^{-8}\left(2\binom{8}{1}\cdot\frac{1}{4} + 2\binom{8}{2}\cdot\frac{2}{4} + 2\binom{8}{3}\cdot\frac{3}{4} + 14\cdot 1 + \left(\binom{8}{4} - 14\right)\cdot\frac{3}{4}\right) = \frac{43}{64}$$

Let us then move on to the next function $f_5^{\mathsf{sym}} : 2^{[16]} \to [0,1]$, giving an inapproximability of 0.7386. It turns out that one can take $f_5^{\mathsf{sym}}(S)$ to be a function of two simple properties of $S$, namely its cardinality $|S|$, and the distance from $S$ to $\mathcal{C}_5^{\mathsf{sym}}$. Specifically, for $|S| \leq 8$ let us define the *number of errors* $e(S)$ as the minimum number of elements that must be removed from $S$ to get a subset of some set in $\mathcal{C}_5^{\mathsf{sym}}$. Formally

$$e(S) = \min_{C \in \mathcal{C}_5^{\mathsf{sym}}} |S \setminus C|,$$

or equivalently, $d(S, \mathcal{C}_5^{\mathsf{sym}}) = 8 - |S| + 2e(S)$, where $d(S, \mathcal{C}_5^{\mathsf{sym}})$ is the Hamming distance from the binary string corresponding to $S$ to the nearest element in $\mathcal{C}_5^{\mathsf{sym}}$. Table 2 gives the values of $f_5^{\mathsf{sym}}$ for all $|S| \leq 8$, and for $|S| > 8$ the value of $f_5^{\mathsf{sym}}(S)$ is given by $f_5^{\mathsf{sym}}(\overline{S})$. Note that, for sets with $e(S) = 0$, i.e., no errors, $f_5^{\mathsf{sym}}(S)$ is simply $|S|/8$, which is what one would expect. However, for sets with errors, $f_5^{\mathsf{sym}}(S)$ has a more complicated behaviour and it is far from clear how this generalizes to larger $l$.

**Table 2.** Description of $f_5^{\mathsf{sym}}(S)$ as a function of $|S|$ and $e(S)$ for $|S| \leq 8$

| | $\lvert S\rvert$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $e(S)$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | 0 | 1/8 | 2/8 | 3/8 | 4/8 | 5/8 | 6/8 | 7/8 | 1 |
| 1 | – | – | – | – | – | 19/32 | 22/32 | 24/32 | 26/32 |
| 2 | – | – | – | – | – | – | 20/32 | 23/32 | 24/32 |

Veryfing that $f_5^{\mathsf{sym}}$ is indeed submodular is not as straightforward as with $f_4^{\mathsf{sym}}$. We have not attempted to construct a shorter proof of this than simply checking condition (1) for every $S$, $i$ and $j$, a task which is of course best suited for a computer program (which is straightforward to write and runs in a few seconds).

A computer program is also the best way to compute the soundness $s(f_5^{\mathsf{sym}})$. It is almost obvious from inspection of Table 2 that $f_5^{\mathsf{sym}}$ satisfies the monotonicity condition of Lemma 1 (the only possible source of failure is that the table only implies that the average of $f_5^{\mathsf{sym}}$ on sets of size 6 is between 20/32 and 24/32, and that the average on sets of size 7 is between 23/32 and 28/32). It turns out that the conditions of Lemma 1 are indeed satisfied and that the average of $f_5^{\mathsf{sym}}$ is $s_{1/2}(f_5^{\mathsf{sym}}) = 709/1024$.

Concluding this discussion on $f_l^{\mathsf{sym}}$, it is tempting to speculate on its behaviour for larger $l$. We have made a computation of $f_6^{\mathsf{sym}} : 2^{[32]} \to [0,1]$, under the assumption that $f_6^{\mathsf{sym}}(S)$ only depends on $|S|$ and the multiset of distances to every point of the support of $\mathcal{C}_6^{\mathsf{sym}}$. Under this assumption, our computations indicate that $s(f_6^{\mathsf{sym}}) \approx 0.7031$ giving an inapproximatibility of $s(f_6^{\mathsf{sym}})/(31/32) \approx 0.7258$, improving upon $f_5^{\mathsf{sym}}$. However, as these computations took a few days they are quite cumbersome to verify (and we have not even made a careful verification of them ourselves) and therefore we do not claim this stronger hardness as a theorem.

### 5.2 Asymmetric Functions

We now return our focus to the asymmetric case. Table 3 describes the hardness ratios obtained from $f_l$ for the cases $l = 3$ and $l = 4$.

**Table 3.** Behaviour of $f_l$ for small $l$

| $l$ | $c$ | $s(f_l)$ | Inapproximability $s/c$ |
|---|---|---|---|
| 3 | 7/8 | $< 0.6275$ | $< 0.7172$ |
| 4 | 15/16 | $< 0.6508$ | $< 0.6942$ |

We begin with the description of the function $f_3 : 2^{[7]} \longrightarrow [0, 1]$. Similarly to the definition $e(S)$ used in the description of $f_5^{\mathsf{sym}}$, let us say that $S \subseteq [7]$ *has no errors* if it is a subset or a superset of some $C \in \mathcal{C}_3$. In other words, if $|S| < 4$ it has no errors if it can be transformed to a set in $\mathcal{C}_3$ by adding some elements, and if $|S| > 4$ it is has no errors if it can be transformed to a codeword by removing some elements. The function $f_3$ is as follows:

$$f_3(S) = \begin{cases} |S|/4 & \text{if } |S| \le 4 \text{ and has no errors} \\ (7 - |S|)/3 & \text{if } |S| > 4 \text{ and has no errors} \\ 11/24 & \text{if } |S| = 3 \text{ and has errors} \\ 17/24 & \text{if } |S| = 4 \text{ and has errors} \end{cases}$$

As with $f_5^{\mathsf{sym}}$, it is not completely obvious that $f_3$ satisfies the submodularity condition and there are a few cases to verify, best left to a computer program.

The average of $f_3$ is $637/1024 \approx 0.622$. However, since $f_3$ takes on its largest values at sets of size $(k + 1)/2 = 4$, the $p$-biased average is larger than this for some $p > 1/2$. It turns out that $s(f_4)$ is obtained by the $p$-biased distribution for $p \approx 0.542404$, giving $s(f_4) \approx 0.627434 < 0.6275$.

We are left with the description of $f_4 : 2^{[15]} \longrightarrow [0, 1]$, which is also the most complicated function yet. One might hope that $f_4$ shares the simple structure of the previous functions – that it depends only on $|S|$ and the distance of $S$ to the nearest $C \in \mathcal{C}_4$. However, the best function under this assumption turns out to give a worse result than $f_3$. Instead, $f_4$ depends on $|S|$ and the multiset of distances to all elements of $\mathcal{C}_4$.

To describe $f_4$, define for $S \subseteq [15]$ the multiset $\mathcal{D}(S)$ as the multiset of distances to all the 15 strings in $\mathcal{C}_4$. For instance, for $S = \emptyset$, $\mathcal{D}(S)$ consists of the number 8 repeated 15 times, reflecting the fact that all strings of $\mathcal{C}_4$ have weight 8, and for $S \in \mathcal{C}_4$ we have that $\mathcal{D}(S)$ consists of the number 8 repeated 14 times,

**Table 4.** Description of $f_4$

| $\lvert S\rvert$ | $\mathcal{D}(S)$ | $\#S$ | $448 \cdot f_4(S)$ |
|---|---|---|---|
| 0 | $8^{15}$ | 1 | 0 |
| 1 | $7^{8}9^{7}$ | 15 | 56 |
| 2 | $6^{4}8^{8}10^{3}$ | 105 | 112 |
| 3 | $5^{2}7^{6}9^{6}11^{1}$ | 420 | 168 |
| 3 | $7^{12}11^{3}$ | 35 | 138 |
| 4 | $4^{2}8^{12}12^{1}$ | 105 | 224 |
| 4 | $4^{1}6^{4}8^{6}10^{4}$ | 840 | 224 |
| 4 | $6^{6}8^{6}10^{2}12^{1}$ | 420 | 194 |
| 5 | $3^{1}5^{1}7^{6}9^{6}11^{1}$ | 840 | 280 |
| 5 | $5^{5}9^{10}$ | 168 | 280 |
| 5 | $5^{3}7^{6}9^{4}11^{2}$ | 1680 | 250 |
| 5 | $5^{2}7^{8}9^{4}13^{1}$ | 315 | 220 |
| 6 | $2^{1}6^{3}8^{8}10^{3}$ | 420 | 336 |
| 6 | $4^{2}6^{3}8^{6}10^{4}$ | 1680 | 306 |
| 6 | $4^{1}6^{5}8^{6}10^{2}12^{1}$ | 2520 | 276 |
| 6 | $6^{9}10^{6}$ | 280 | 276 |
| 6 | $6^{6}8^{8}14^{1}$ | 105 | 216 |
| 7 | $1^{1}7^{7}9^{7}$ | 120 | 392 |
| 7 | $3^{1}5^{2}7^{5}9^{6}11^{1}$ | 2520 | 332 |
| 7 | $3^{1}7^{11}11^{3}$ | 420 | 302 |
| 7 | $5^{4}7^{5}9^{4}11^{2}$ | 2520 | 302 |
| 7 | $5^{3}7^{7}9^{4}13^{1}$ | 840 | 272 |
| 7 | $7^{14}15^{1}$ | 15 | 197 |
| 8 | $0^{1}8^{14}$ | 15 | 448 |
| 8 | $2^{1}6^{4}8^{7}10^{3}$ | 840 | 358 |
| 8 | $4^{3}8^{11}12^{1}$ | 420 | 328 |
| 8 | $4^{2}6^{4}8^{5}10^{4}$ | 2520 | 328 |
| 8 | $4^{1}6^{6}8^{5}10^{2}12^{1}$ | 2520 | 298 |
| 8 | $6^{7}8^{7}14^{1}$ | 120 | 253 |
| 9 | $1^{1}7^{8}9^{6}$ | 105 | 384 |
| 9 | $3^{1}5^{2}7^{6}9^{5}11^{1}$ | 2520 | 324 |
| 9 | $5^{6}9^{9}$ | 280 | 324 |
| 9 | $5^{4}7^{6}9^{3}11^{2}$ | 1680 | 294 |
| 9 | $5^{3}7^{8}9^{3}13^{1}$ | 420 | 279 |
| 10 | $2^{1}6^{4}8^{8}10^{2}$ | 315 | 320 |
| 10 | $4^{2}6^{4}8^{6}10^{3}$ | 1680 | 290 |
| 10 | $4^{1}6^{6}8^{6}10^{1}12^{1}$ | 840 | 275 |
| 10 | $6^{10}10^{5}$ | 168 | 260 |
| 11 | $3^{1}5^{2}7^{6}9^{6}$ | 420 | 256 |
| 11 | $3^{1}7^{12}11^{2}$ | 105 | 256 |
| 11 | $5^{4}7^{6}9^{4}11^{1}$ | 840 | 241 |
| 12 | $4^{3}8^{12}$ | 35 | 192 |
| 12 | $4^{1}6^{6}8^{6}10^{2}$ | 420 | 192 |
| 13 | $5^{3}7^{8}9^{4}$ | 105 | 128 |
| 14 | $6^{7}8^{8}$ | 15 | 64 |
| 15 | $7^{15}$ | 1 | 0 |

together with a single 0, because the distance between any pair of strings in $\mathcal{C}_4$ is 8.

Table 4 describes the behaviour of $f_4(S)$ as a function of $|S|$ and $\mathcal{D}(S)$.[3] In the table $\mathcal{D}(S)$ is described by a string of the form $d_1^{m_1} d_2^{m_2} \ldots$, with $d_1 < d_2 < \ldots$ and $\sum m_i = 15$, indicating that $m_1$ strings of $\mathcal{C}_4$ are at distance $d_1$ from $S$, that $m_2$ strings are at distance $d_2$, and so on. Thus, for $S = \emptyset$ the description of $\mathcal{D}(S)$ is "$8^{15}$", and for $S \in \mathcal{C}_4$ the description of $\mathcal{D}(S)$ is "$0^1 8^{14}$".

The $\#S$ column of Table 4 gives the total number of $S \subseteq [15]$ having this particular value of $(|S|, \mathcal{D}(S))$, and the last column gives the actual value of $f_4$, multiplied by 448 to make all values integers.

[3] It is not necessary to include $|S|$ as it is uniquely determined by $\mathcal{D}(S)$, but we find that explicitly including $|S|$ makes the table somewhat less obscure.

Again, checking that $f_4$ is submodular is a tedious task best suited for a computer. The average of $f_4$ is $9519345/(448 \cdot 2^{15}) \approx 0.6485$, but, as with $f_3$, $s(f_4)$ is somewhat larger than this. It turns out that the $p$ maximizing $s_p(f_4)$ is roughly $p \approx 0.526613$, and that $s(f_4) \approx 0.650754 < 0.6508$.

Finally, we mention that as in the symmetric case, we have made a computation of the next function, $f_5$, again under the assumption that it depends only on the multiset of distances to the codewords. Under this assumption it turns out that $s_{1/2}(f_5) \approx 0.6743$, meaning that the inapproximability obtained can not be better than $s_{1/2}(f_5)/(31/32) \approx 0.6961$ which is worse than the inapproximability obtained from $f_4$.

## Acknowledgments

We are grateful to Jan Vondrák for stimulating discussions.

## References

1. Austrin, P.: Improved Inapproximability For Submodular Maximization. arXiv:1004.3777v1 [cs.CC] (2010)
2. Austrin, P., Mossel, E.: Approximation Resistant Predicates from Pairwise Independence. Computational Complexity 18(2), 249–271 (2009)
3. Feige, U., Mirrokni, V.S., Vondrák, J.: Maximizing non-monotone submodular functions. In: IEEE Symposium on Foundations of Computer Science (FOCS), pp. 461–471 (2007)
4. Goemans, M.X., Williamson, D.P.: Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. Journal of the ACM 42, 1115–1145 (1995)
5. Grötschel, M., Lovász, L., Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization. Combinatorica 1(2), 169–197 (1981)
6. Khot, S.: On the Power of Unique 2-prover 1-round Games. In: ACM Symposium on Theory of Computing (STOC), pp. 767–775 (2002)
7. Lovász, L.: Submodular functions and convexity. In: Grötschel, M., Bachem, A., Korte, B. (eds.) Mathematical Programming: The State of the Art - Bonn 1982, pp. 235–257. Springer, Heidelberg (1983)
8. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions–I. Mathematical Programming 14, 265–294 (1978)
9. Vondrák, J.: Submodular maximization by simulated annealing. Unpublished manuscript
10. Vondrák, J.: Submodularity in Combinatorial Optimization. PhD thesis, Charles University (2007)