

# The Power of Optimization from Samples

**Authored by:**

Yaron Singer  
Eric Balkanski  
Aviad Rubinstein

## **Abstract**

We consider the problem of optimization from samples of monotone submodular functions with bounded curvature. In numerous applications, the function optimized is not known a priori, but instead learned from data. What are the guarantees we have when optimizing functions from sampled data? In this paper we show that for any monotone submodular function with curvature  $c$  there is a  $(1 - c)/(1 + c - c^2)$  approximation algorithm for maximization under cardinality constraints when polynomially-many samples are drawn from the uniform distribution over feasible sets. Moreover, we show that this algorithm is optimal. That is, for any  $c \leq 1$ , there exists a submodular function with curvature  $c$  for which no algorithm can achieve a better approximation. The curvature assumption is crucial as for general monotone submodular functions no algorithm can obtain a constant-factor approximation for maximization under a cardinality constraint when observing polynomially-many samples drawn from any distribution over feasible sets, even when the function is statistically learnable.

## **1 Paper Body**

Traditionally, machine learning is concerned with predictions: assuming data is generated from some model, the goal is to predict the behavior of the model on data similar to that observed. In many cases however, we harness machine learning to make decisions: given observations from a model the goal is to find its optimum, rather than predict its behavior. Some examples include: ? Ranking in information retrieval: In ranking the goal is to select  $k \leq N$  documents that are most relevant for a given query. The underlying model is a function which maps a set of documents and a given query to its relevance score. Typically we do not have access to the scoring function, and thus learn it from data. In the learning to rank framework, for example, the input consists of observations of document-query pairs and their relevance score. The goal is to construct a scoring function of query-document pairs so that given a query we can decide on the  $k$  most relevant documents. ? Optimal tagging:

The problem of optimal tagging consists of picking  $k$  tags for some new content to maximize incoming traffic. The model is a function which captures the way in which users navigate through content given their tags. Since the algorithm designer cannot know the behavior of every online user, the model is learned from observations on user navigation in order to make a decision on which  $k$  tags maximize incoming traffic.  $\bullet$  Influence in networks: In influence maximization the goal is to identify a subset of individuals who can spread information in a manner that generates a large cascade. The underlying 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

assumption is that there is a model of influence that governs the way in which individuals forward information from one to another. Since the model of influence is not known, it is learned from data. The observed data is pairs of a subset of nodes who initiated a cascade and the total number of individuals influenced. The decision is the optimal set of influencers. In the interest of maintaining theoretical guarantees on the decisions, we often assume that the generative model has some structure which is amenable to optimization. When the decision variables are discrete quantities a natural structure for the model is submodularity. A function  $f : 2N \rightarrow \mathbb{R}$  defined over a ground set  $N = \{e_1, \dots, e_n\}$  of elements is submodular if it exhibits a diminishing marginal returns property, i.e.,  $f(S \cup \{e\}) - f(S) \geq f(T \cup \{e\}) - f(T)$  for all sets  $S \subseteq T \subseteq N$  and element  $e \notin T$  where  $f(S \cup \{e\}) - f(S)$  is the marginal contribution of element  $e$  to set  $S \subseteq N$ . This diminishing returns property encapsulates numerous applications in machine learning and data mining and is particularly appealing due to its theoretical guarantees on optimization (see related work below). The guarantees on optimization of submodular functions apply to the case in which the algorithm designer has access to some succinct description of the function, or alternatively some idealized value oracle which allows querying for function values of any given set. In numerous settings such as in the above examples, we do not have access to the function or its value oracle, but rather learn the function from observed data. If the function learned from data is submodular we can optimize it and obtain a solution with provable guarantees on the learned model. But how do the guarantees of this solution on the learned model relate to its guarantees on the generative model? If we obtain an approximate optimum on the learned model which turns out to be far from the optimum of the submodular function we aim to optimize, the provable guarantees at hand do not apply. Optimization from samples. For concreteness, suppose that the generative model is a monotone submodular function  $f : 2N \rightarrow \mathbb{R}$  and we wish to find a solution to  $\max_{S \subseteq N} |S| = k f(S)$ . To formalize the concept of observations in standard learning-theoretic terms, we can assume that we observe samples of sets drawn from some distribution  $D$  and their function values, i.e.  $\{(S_i, f(S_i))\}_{i=1}^m$ . In terms of learnability, under some assumptions about the distribution and the function, submodular functions are statistically learnable (see discussion about PMAC learnability). In terms of approximation guarantees for optimization, a simple greedy algorithm obtains a  $(1 - 1/e)$ -approximation. Recent work shows that optimization from samples is generally impossible [4], even for models that are learnable and optimizable. In particular, even for maximizing

coverage functions, which are a special case of submodular functions and widely used in practice, no algorithm can obtain a constant factor approximation using fewer than exponentially many samples of feasible solutions drawn from any distribution. In practice however, the functions we aim to optimize may be better behaved. An important property of submodular functions that has been heavily explored recently is that of curvature. Informally, the curvature  $\rho$  is a measure of how far the function is to being modular. A function  $f$  is modular if  $f(S) = \sum_{e \in S} f(e)$ , and has curvature  $\rho \in [0, 1]$  if  $f(S) \geq (1 - \rho)f(e) + \rho f(S \cup e)$  for any  $S \subseteq N$ . Curvature plays an important role since the hard instances of submodular optimization often occur only when the curvature is unbounded, i.e.,  $\rho$  close to 1. The hardness results for optimization from samples are no different, and apply when the curvature is unbounded. What are the guarantees for optimization from samples of submodular functions with bounded curvature? In this paper we study the power of optimization from samples when the curvature is bounded. Our main result shows that for any monotone submodular function with curvature  $\rho$  there is an algorithm which observes polynomially-many samples from the uniform distribution over feasible sets and obtains an approximation ratio of  $(1 - \rho)/(1 + \rho/2) + o(1)$ . Furthermore, we show that this bound is tight. For any  $\rho \leq 1$ , there exist monotone submodular functions with curvature  $\rho$  for which no algorithm can obtain an approximation better than  $(1 - \rho)/(1 + \rho/2) + o(1)$  given polynomially many samples. We also perform experiments on synthetic hard instances of monotone submodular functions that convey some interpretation of our results. For the case of modular functions a  $1 - o(1)$  algorithm can be obtained and as a consequence leads to a  $(1 - \rho)/2$  algorithm for submodular functions with bounded curvature [4]. The goal of this work is to exploit the curvature property to obtain the optimal algorithm for optimization from samples. 2

A high-level overview of the techniques. The algorithm estimates the expected marginal contribution of each element to a random set. It then returns the (approximately) best set between the set of elements with the highest estimates and a random set. The curvature property is used to bound the differences between the marginal contribution of each element to: (1) a random set, (2) the set of elements with highest (estimated) marginal contributions to a random set, and (3) the optimal set. A key observation in the analysis is that if the difference between (1) and (3) is large, then a random set has large value (in expectation). To obtain our matching inapproximability result, we construct an instance where, after viewing polynomially many samples, the elements of the optimal set cannot be distinguished from a much larger set of elements that have high marginal contribution to a random set, but low marginal contribution when combined with each other. The main challenge is constructing the optimal elements such that they have lower marginal contribution to a random set than to the other optimal elements. This requires carefully defining the way different types of elements interact with each other, while maintaining the global properties of monotonicity, submodularity, and bounded curvature. 1.1

#### Related work

Submodular maximization. In the traditional value oracle model, an algorithm may adaptively query polynomially many sets  $S_i$  and obtain via a black-

box their values  $f(S_i)$ . It is well known that in this model, the greedy algorithm obtains a  $1 + 1/e$  approximation for a wide range of constraints including cardinality constraints [23], and that no algorithm can do better [6]. Submodular optimization is an essential tool for problems in machine learning and data mining such as sensor placement [20, 12], information retrieval [28, 14], optimal tagging [24], influence maximization [19, 13], information summarization [21, 22], and vision [17, 18]. Learning. A recent line of work focuses on learning submodular functions from samples [3, 8, 2, 10, 11, 1, 9]. The standard model to learn submodular functions is  $\epsilon$ -PMAC learnability introduced by Balcan and Harvey [3] which generalizes the well known PAC learnability framework from Valiant [26]. Informally, a function is PAC or PMAC learnable if given polynomially samples, it is possible to construct a function that is likely to mimic the function for which the samples are coming from. Monotone submodular functions are  $\epsilon$ -PMAC learnable from samples coming from a product distribution for some constant  $\epsilon$  and under some assumptions [3]. Curvature. In the value oracle model, the greedy algorithm is a  $(1 + e^{-c})/c$  approximation algorithm for cardinality constraints [5]. Recently, Sviridenko et al. [25] improved this approximation to  $1 + c/e$  with variants of the continuous greedy and local search algorithms. Submodular optimization and curvature have also been studied for more general constraints [27, 15] and submodular minimization [16]. The curvature assumption has applications in problems such as maximum entropy sampling [25], column-subset selection [25], and submodular welfare [27].

2

#### Optimization from samples

We precisely define the framework of optimization from samples. A sample  $(S, f(S))$  of function  $f$  is a set and its value. As with the PMAC-learning framework, the samples  $(S_i, f(S_i))$  are such that the sets  $S_i$  are drawn i.i.d. from a distribution  $D$ . As with the standard optimization framework, the goal is to return a set  $S$  satisfying some constraint  $M \subseteq 2N$  such that  $f(S)$  is an  $\epsilon$ -approximation to the optimal solution  $f(S^*)$  with  $|S| \leq M$ .

A class of functions  $F$  is  $\epsilon$ -optimizable from samples under constraint  $M$  and over distribution  $D$  if for all functions  $f \in F$  there exists an algorithm which, given polynomially many samples  $(S_i, f(S_i))$ , returns with high probability over the samples a set  $S \subseteq M$  such that  $f(S)$

$$\geq (1 - \epsilon) \max_{T \subseteq M} f(T).$$

In the unconstrained case, a random set achieves a  $1/4$ -approximation for general (not necessarily monotone) submodular functions [7]. We focus on the constrained case and consider a simple cardinality constraint  $M$ , i.e.,  $M = \{S : |S| \leq k\}$ . To avoid trivialities in the framework, it is important to fix a distribution  $D$ . We consider the distribution  $D$  to be the uniform distribution over all feasible sets, i.e., all sets of size at most  $k$ .

We are interested in functions that are both learnable and optimizable. It is already known that there exists classes of functions, such as coverage and submodular, that are both learnable and optimizable but not optimizable from samples for  $M$  and  $D$  defined above. This paper studies optimization from samples under some additional assumption: curvature. We assume that the

curvature  $c$  of the function is known to the algorithm designer. In the appendix, we show an impossibility result for learning the curvature of a function from samples.

3

### An optimal algorithm

We design a  $(1 - c)/(1 + c - c^2) - o(1)$ -optimization from samples algorithm for monotone submodular functions with curvature  $c$ . In the next section, we show that this approximation ratio is tight. The main contribution is improving over the  $(1 - c)^2 - o(1)$  approximation algorithm from [4] to obtain a tight bound on the approximation. The algorithm. Algorithm 1 first estimates the expected marginal contribution of each element  $e_i$  to a uniformly random set of size  $k - 1$ , which we denote by  $R$  for the remaining of this section. These expected marginal contributions  $ER[fR(e_i)]$  are  $P$  estimated with  $v_i$ . The estimates  $v_i$  are the differences between the average value  $\text{avg}(S_{k,i}) := (T - 2S_{k,i} f(T))/k$  of the collection  $S_{k,i}$  of samples of size  $k$  containing  $e_i$  and the average value of the collection  $S_{k-1,i}$  of samples of size  $k - 1$  not containing  $e_i$ . We then wish to return the best set between the random set  $R$  and the set  $S$  consisting of the  $k$  elements with the largest estimates  $v_i$ . Since we do not know the value of  $S$ , we lower bound it with  $v_S$  using the curvature property. We estimate the expected value  $ER[f(R)]$  of  $R$  with  $v_R$ , which is the average value of the collection  $S_{k-1}$  of all samples of size  $k - 1$ . Finally, we compare the values of  $S$  and  $R$  using  $v_S$  and  $v_R$  to return the best of these two sets. Algorithm 1 A tight  $(1 - c)/(1 + c - c^2) - o(1)$ -optimization from samples algorithm for monotone submodular functions with curvature  $c$  Input:  $S = \{S_i : (S_i, f(S_i)) \text{ is a sample}\}$  1:  $v_i \leftarrow \text{avg}(S_{k,i}) - \text{avg}(S_{k-1,i})$  2:  $S \leftarrow \arg\max_{S \subseteq [n], |S|=k} \sum_{e_i \in S} v_i$  3:  $v_S \leftarrow (1 - c) \sum_{e_i \in S} v_i$  4:  $v_R \leftarrow \text{avg}(S_{k-1})$  5: if  $v_S \geq v_R$  then 6: return  $S$  7: else 8: return  $R$  9: end if The analysis. Without loss of generality, let  $S = \{e_1, \dots, e_k\}$  be the set defined in Line 2 of the algorithm and define  $S_i$  to be the first  $i$  elements in  $S$ , i.e.,  $S_i := \{e_1, \dots, e_i\}$ . Similarly, for the optimal solution  $S^*$ , we have  $S^* = \{e_{i_1}, \dots, e_{i_k}\}$  and  $S_i^* := \{e_{i_1}, \dots, e_{i_i}\}$ . We abuse notation and denote by  $f(R)$  and  $fR(e)$  the expected values  $ER[f(R)]$  and  $ER[fR(e)]$  where the randomization is over the random set  $R$  of size  $k - 1$ .  $P$  At a high level, the curvature property is used to bound the loss from  $f(S)$  to  $\sum_{e_i \in S} fR(e_i)$  and  $P$  from  $\sum_{e_i \in S} fR(e_i)$  to  $f(S^*)$ . By the algorithm,  $\sum_{e_i \in S} fR(e_i)$  is greater than  $\sum_{e_i \in S} fR(e_{i_i})$ . When  $P$  bounding the loss from  $\sum_{e_i \in S} fR(e_{i_i})$  to  $f(S^*)$ , a key observation is that if this loss is large, then it must be the case that  $R$  has a high expected value. This observation is formalized in our analysis by bounding this loss in terms of  $f(R)$  and motivates Algorithm 1 returning the best of  $R$  and  $S$ . Lemma 1 is the main part of the analysis and gives an approximation for  $S$ . The approximation guarantee for Algorithm 1 (formalized as Theorem 1) follows by finding the worst-case ratios of  $f(R)$  and  $f(S)$ . Lemma 1. Let  $S$  be the set defined in Algorithm 1 and  $f(\cdot)$  be a monotone submodular function with curvature  $c$ , then  $f(R) - f(S) \geq (1 - o(1)) \cdot v_S - (1 - c) \cdot v_S - o(1) \cdot f(S^*)$ .  $f(S^*) \geq 4$

Proof. First, observe that  $f(S) =$

$X$   
 $f(S_{i-1}(e_i))$   
 $(1 - c)$   
 $i^*k$   
 $X_{i^*k}$   
 $f(e_i)$   
 $(1 - c)$   
 $X$   
 $f_R(e_i)$   
 $i^*k$

where the first inequality is by curvature and the second is by monotonicity. We now claim that w.h.p. and with a sufficiently large polynomial number of samples the estimates of the marginal contribution of an element are precise,  $f(S_{i-1}(e_i)) \approx f_R(e_i)$  and  $f(S_{i-1}(e_i)) \approx f_R(e_i)$ . Thus  $f(S_{i-1}(e_i)) \approx f_R(e_i)$  and  $f(S_{i-1}(e_i)) \approx f_R(e_i)$ . Next, by the definition of  $S$  in the algorithm, we get  $X_{i^*k} \approx f(S_{i-1}(e_i)) \approx f_R(e_i)$ . It is possible to obtain a  $(1 - c)$  loss between  $f(S_{i-1}(e_i))$  and  $f(S_{i-1}(e_i))$  with a similar argument as in the first part. The key idea to improve this loss is to use the curvature property on the elements in  $R$  instead of on the elements in  $S$ . By curvature, we have that  $f(S_{i-1}(e_i)) \approx f_R(e_i)$ . We now wish to relate  $f(S_{i-1}(e_i))$  and  $f_R(e_i)$ . Note that  $f(S_{i-1}(e_i)) + f_R(e_i) = f(R \cup \{e_i\}) = f(R) + f_R(e_i)$  by submodularity. We get  $f(S_{i-1}(e_i)) \approx f_R(e_i)$  by combining the previous equation and inequality. By  $f_R(e_i)$  the previous curvature observation, we conclude that  $f(S_{i-1}(e_i)) \approx f_R(e_i)$  and  $f(S_{i-1}(e_i)) \approx f_R(e_i)$ .

Combining Lemma 1 and the fact that we obtain value at least  $\max\{f(R), (1 - c)f(S_{i-1}(e_i))\}$ , we obtain the main result of this section. Theorem 1. Let  $f$  be a monotone submodular function with curvature  $c$ . Then Algorithm 1 is a  $(1 - c)/(1 + c^2) \approx 0.1$  optimization from samples algorithm. Proof. In the appendix, we show that the estimate  $\hat{f}(R)$  of  $f(R)$  is precise, the estimate is such that  $f(R) \approx \hat{f}(R)$  and  $f(S_{i-1}(e_i)) \approx \hat{f}(S_{i-1}(e_i))$ . In addition, by the first inequality in Lemma 1,  $f(S_{i-1}(e_i)) \approx \hat{f}(S_{i-1}(e_i))$ . So by the algorithm and the second inequality in Lemma 1, the approximation obtained by the set returned is at least  $\max\{f(R), (1 - c)f(S_{i-1}(e_i))\} \approx \max\{f(R), (1 - c)f(S_{i-1}(e_i))\}$ .

Let  $x := f(R)/f(S_{i-1}(e_i))$ , the best of  $f(R)/f(S_{i-1}(e_i))$  and  $(1 - c)f(R)/f(S_{i-1}(e_i))$  is minimized when  $x = (1 - c)/(1 + c^2)$ , or when  $x = (1 - c)/(1 + c^2)$ . Thus, the approximation obtained is at least  $(1 - c)/(1 + c^2) \approx 0.1$ .

#### Hardness

We show that the approximation obtained by Algorithm 1 is tight. For every  $c \in [1, \infty)$ , there exists monotone submodular functions that cannot be  $(1 - c)/(1 + c^2)$







bad elements. There is also a  $1/(1 + c c_2)$  loss between the contribution 1 of a good element to a set with no poor elements and its contribution  $1/(1 + c c_2)$  to a random set. We add a function  $p(sP)$  to  $f_G(?)$  so that it is monotone increasing when adding poor elements.  $(1 - c k sP) \geq 1 + c$  if  $sP \geq k/2 \log n c_2$   $k/2 \log n$   $p(sP) = (1 - c)2^{1 - c k (sP / (k/2 \log n))} 1 + c c_2 + (k/2 \log n) 1 + c c_2 k/2 \log n$  otherwise. The next two lemmas show that these functions satisfy Lemma 3 and that  $f_G(?)$  is monotone submodular with curvature  $c$ , which concludes the proof of Theorem 2. Lemma 4. The functions  $g(?)$ ,  $b(?)$ , and  $p(?)$  defined above satisfy the conditions of Lemma 3 with  $\epsilon = (1 - c)/(1 + c c_2) + o(1)$ . Proof. We start with the first condition. Assume  $sG + sB = s_0G + s_0B \geq \log n$  and  $sP, s_0P \leq k/2 \log n$ . Then,  $1 - g(sG, sP) + b(sB) = (sG + sB) \geq (s_0G + s_0B) \geq g(s_0G, s_0P) + b(s_0B)$ .  $1 + c c_2 \geq 1 + c c_2$  For the second condition, assume  $sG \geq n$  and  $sG + sB + sP \geq k$ . It is without loss to assume that  $sB + sP \leq k n$ , then  $\epsilon \geq 1 - c \geq G f(S) \geq (1 + o(1)) \geq (sB + sP) \geq k + o(1) \geq 1 + c c_2 \geq 1 + c c_2$

We conclude by noting that  $g(k, 0) = k$ .

Lemma 5. The function  $f_G(?)$  is a monotone submodular function with curvature  $c$ . Proof. We show that the marginal contributions are positive (monotonicity), decreasing (submodularity), but not by more than a  $1 - c$  factor (curvature), i.e., that  $f_S(e) - f_T(e) \leq (1 - c)f_S(e)$  for all  $S \subseteq T$  and  $e \notin T$ . Let  $e$  be a good element, then  $\epsilon \geq 1 - c \geq 1 + c c_2 \geq sP \geq k/2 \log n$  if  $sP \geq k/2 \log n$   $G f_S(e) = 1$  otherwise.  $1 + c c_2 \geq 7$

Since  $sP \geq tP$  for  $S \subseteq T$ , we obtain  $f_S(e) - f_T(e) \geq 0$ . It is also easy to see that we get  $f_T(e) \geq 1 + c c_2 (1 - c) (1 - c)f_S(e)$ . For bad elements,  $(1 - c) \geq sB \geq \log n$   $G f_S(e) = 1 + c c_2$  otherwise.  $2 \geq 1 + c c_2$  Thus,  $f_S(e) - f_T(e) =$  Since  $sG \geq k$ ,

$c)f_S(e) \geq 0$  for all  $S \subseteq T$  and  $e \notin T$ . Finally, for poor elements,  $\epsilon \leq 1 - c k \geq sG \geq k/2 \log n + 1 + c$  if  $sP \geq k/2 \log n$   $2 \geq 1 + c c_2 \geq k/2 \log n$

$f_T(e) \leq (1 - c) \geq 1$

$c)2^{k/2 \log n} 1 + c c_2 k/2 \log n$

$\geq (1$

$1 - c k \geq 2 \geq 1 + c c_2 k/2 \log n$  Consider  $S \subseteq T$ , then  $sG \geq tG$ , and  $f_S(e)$

$\geq 5$

otherwise.

$f_S(e) - f_T(e)$

$(1 - c)2^{k/2 \log n} \geq 1 + c c_2 k/2 \log n (1 - c)f_S(e) \geq 0$ .

Experiments

We perform simulations on simple synthetic functions. These experiments are meant to complement the theoretical analysis by conveying some interpretations of the bounds obtained. The synthetic functions are a simplification of the construction for the impossibility result. The motivation for these functions is to obtain hard instances that are challenging for the algorithm. More precisely, the function considered is  $f(S) = |S \cap G| - |S \cap B|$  if  $|S| \leq 10$   $f(S) = |S \cap G| + |S \cap B| - (1 - c)10c$  otherwise, where  $G$  and  $B$  are fixed sets of size 102 and 103 respectively. The ground set  $N$  Figure 2: The objective  $f(?)$  as a function of the contains 105 elements. It is easy to verify that cardinality constraint  $k$ .  $f(?)$  has curvature  $c$ . This function is hard to optimize since the elements in  $G$

and  $B$  cannot be distinguished from samples. We consider several benchmarks. The first is the value obtained by the learn then optimize approach where we first learn the function and then optimize the learned function. Equivalently, this is a random set of size  $k$ , since the learned function is a constant with the algorithm from [3]. We also compare our algorithm to the value of the best sample observed. The solution returned by the greedy algorithm is an upper bound and is a solution obtainable only in the full information setting. The results are summarized in Figure 2 and 3. In Figure 2, the value of greedy, best sample, and random set do not change for different curvatures  $c$  since w.h.p. they pick at most 10 elements from Figure 3: The approximation as a function of the  $B$ . For curvature  $c = 0$ , when the function curvature  $1/c$  when  $k = 100$ . is modular, our algorithm performs as well as the greedy algorithm, which is optimal. As the curvature increases, the solution obtained by our algorithm worsens, but still significantly outperforms the best sample and a random set. The power of our algorithm is that it is capable to distinguish elements in  $G \setminus B$  from the other elements. 8

## 2 References

- [1] Balcan, M. (2015). Learning submodular functions with applications to multi-agent systems. In AAMAS.
- [2] Balcan, M., Constantin, F., Iwata, S., and Wang, L. (2012). Learning valuation functions. In COLT.
- [3] Balcan, M. and Harvey, N. J. A. (2011). Learning submodular functions. In STOC.
- [4] Balkanski, E., Rubinstein, A., and Singer, Y. (2015). The limitations of optimization from samples. arXiv preprint arXiv:1512.06238.
- [5] Conforti, M. and Cornuéjols, G. (1984). Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the rado-edmonds theorem. Discrete applied mathematics.
- [6] Feige, U. (1998). A threshold of  $\ln n$  for approximating set cover. JACM.
- [7] Feige, U., Mirrokni, V. S., and Vondrak, J. (2011). Maximizing non-monotone submodular functions. SIAM Journal on Computing.
- [8] Feldman, V. and Kothari, P. (2014). Learning coverage functions and private release of marginals. In COLT.
- [9] Feldman, V., Kothari, P., and Vondrak, J. (2013). Representation, approximation and learning of submodular functions using low-rank decision trees. In COLT.
- [10] Feldman, V. and Vondrak, J. (2013). Optimal bounds on approximation of submodular and XOS functions by juntas. In FOCS.
- [11] Feldman, V. and Vondrak, J. (2015). Tight bounds on low-degree spectral concentration of submodular and XOS functions. CoRR.
- [12] Golovin, D., Faulkner, M., and Krause, A. (2010). Online distributed sensor selection. In IPSN.
- [13] Gomez Rodriguez, M., Leskovec, J., and Krause, A. (2010). Inferring networks of diffusion and influence. In SIGKDD.
- [14] Hang, L. (2011). A short introduction to learning to rank. IE-ICE.
- [15] Iyer, R. K. and Bilmes, J. A. (2013). Submodular optimization with submodular cover and submodular knapsack constraints. In NIPS.
- [16] Iyer, R. K., Jegelka, S., and Bilmes, J. A. (2013). Curvature and optimal algorithms for learning and minimizing submodular functions. In NIPS.
- [17] Jegelka, S.

and Bilmes, J. (2011a). Submodularity beyond submodular energies: coupling edges in graph cuts. In CVPR. [18] Jegelka, S. and Bilmes, J. A. (2011b). Approximation bounds for inference using cooperative cuts. In ICML. [19] Kempe, D., Kleinberg, J., and Tardos, ?. (2003). Maximizing the spread of influence through a social network. In SIGKDD. [20] Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., and Glance, N. (2007). Cost-effective outbreak detection in networks. In SIGKDD. [21] Lin, H. and Bilmes, J. (2011a). A class of submodular functions for document summarization. In NAACL HLT. [22] Lin, H. and Bilmes, J. A. (2011b). Optimal selection of limited vocabulary speech corpora. In INTERSPEECH. [23] Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions ii. Math. Programming Study 8. [24] Rosenfeld, N. and Globerson, A. (2016). Optimal Tagging with Markov Chain Optimization. arXiv preprint arXiv:1605.04719. [25] Sviridenko, M., Vondr?k, J., and Ward, J. (2015). Optimal approximation for submodular and supermodular optimization with bounded curvature. In SODA. [26] Valiant, L. G. (1984). A Theory of the Learnable. Commun. ACM. [27] Vondr?k, J. (2010). Submodularity and curvature: the optimal algorithm. RIMS. [28] Yue, Y. and Joachims, T. (2008). Predicting diverse subsets using structural svms. In ICML.