Minimizing a Submodular Function from Samples

Authored by:

Yaron Singer
Eric Balkanski

Abstract

In this paper we consider the problem of minimizing a submodular function from training data. Submodular functions can be efficiently minimized and are conse- quently heavily applied in machine learning. There are many cases, however, in which we do not know the function we aim to optimize, but rather have access to training data that is used to learn the function. In this paper we consider the question of whether submodular functions can be minimized in such cases. We show that even learnable submodular functions cannot be minimized within any non-trivial approximation when given access to polynomially-many samples. Specifically, we show that there is a class of submodular functions with range in [0, 1] such that, despite being PAC-learnable and minimizable in polynomial-time, no algorithm can obtain an approximation strictly better than 1/2 ? o(1) using polynomially-many samples drawn from any distribution. Furthermore, we show that this bound is tight using a trivial algorithm that obtains an approximation of 1/2.

1 Paper Body

For well over a decade now, submodular minimization has been heavily studied in machine learning (e.g. [SK10, JB11, JLB11, NB12, EN15, DTK16]). This focus can be largely attributed to the fact that if a set function f : $2N$! R is submodular, meaning it has the following property of diminishing returns: f (S [{a}) f (S) f (T [{a}) f (T) for all S ? T ? N and a 62 T , then it can be optimized efficiently: its minimizer can be found in time that is polynomial in the size of the ground set N [GLS81, IFF01]. In many cases, however, we do not know the submodular function, and instead learn it from data (e.g. [BH11, IJB13, FKV13, FK14, Bal15, BVW16]). The question we address in this paper is whether submodular functions can be (approximately) minimized when the function is not known but can be learned from training data. An intuitive approach for optimization from training data is to learn a surrogate function from training data that predicts the behavior of the submodular function well, and then find the minimizer of the surrogate learned and use that as a proxy for the true minimizer we seek. The problem however, is that this approach does

not generally guarantee that the resulting solution is close to the true minimum of the function. One pitfall is that the surrogate may be non-submodular, and despite approximating the true submodular function arbitrarily well, the surrogate can be intractable to minimize. Alternatively, it may be that the surrogate is submodular, but its minimum is arbitrarily far from the minimum of the true function we aim to optimize (see examples in Appendix A). Since optimizing a surrogate function learned from data may generally result in poor approximations, one may seek learning algorithms that are guaranteed to produce surrogates whose optima wellapproximate the true optima and are tractable to compute. More generally, however, it is possible that there is some other approach for optimizing the function from the training samples, without learning a model. Therefore, at a high level, the question is whether a reasonable number of training samples suffices to minimize a submodular function. We can formalize this as optimization from samples. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Optimization from samples. We will say that a class of functions F = {f : 2N ! [0, 1]} is ?-optimizable from samples over distribution D if for every f 2 F and 2 (0, 1), when given poly(—N —) i.i.d. samples {(Si , f (Si))}m over the i=1 where Si ? D, with probability at least 1 samples one can construct an algorithm that returns a solution S ? N s.t, f (S)

min f (T) ? ?.
T ?N

This framework was recently introduced in [BRS17] for the problem of submodular maximization where the standard notion of approximation is multiplicative. For submodular minimization, since the optimum may have zero value, the suitable measure is that of additive approximations for [0, 1]bounded functions, and the goal is to obtain a solution which is a o(1) additive approximation to the minimum (see e.g. [CLSW16, EN15, SK10]). The question is then: Can submodular functions be minimized from samples? Since submodular functions can be minimized in polynomial-time, it is tempting to conjecture that when the function is learnable it also has desirable approximation guarantees from samples, especially in light of positive results in related settings of submodular maximization: ? Constrained maximization. For functions that can be maximized in polynomial time under a cardinality constraint, like modular and unit-demand functions, there are polynomial time algorithms that obtain an arbitrarily good approximation using polynomially-many samples [BRS16, BRS17]. For general monotone submodular functions which are NPhard to maximize under cardinality constraints, there is no algorithm that can obtain a reasonable approximation from polynomially-many samples [BRS17]. For the problem of unconstrained minimization, submodular functions can be optimized in polynomial time; ? Unconstrained maximization. For unconstrained maximization of general submodular functions, the problem is NP-hard to maximize (e.g. MAX-CUT) and one seeks constant factor approximations. For this problem, there is an extremely simple algorithm that uses no queries and obtains a good approximation: choose elements uniformly at random with probability 1/2 each. This algorithm achieves a constant factor approximation of 1/4 for general

submodular functions. For symmetric submodular functions (i.e. f (S) = f (N
S)), this algorithm is a 1/2-approximation which is optimal, since no algorithm
can obtain an approximation ratio strictly better than 1/2 using polynomially-
many value queries, even for symmetric submodular functions [FMV11]. For
unconstrained symmetric submodular minimization, there is an appealing ana-
logue: the empty set and the ground set N are guaranteed to be minimizers of
the function (see Section 2). This algorithm, of course, uses no queries either.
The parallel between these two problems seems quite intuitive, and it is tempt-
ing to conjecture that like for unconstrained submodular maximization, there
are optimization from samples algorithms for general unconstrained submodu-
lar minimization with good approximation guarantees. Main result. Somewhat
counter-intuitively, we show that despite being computationally tractable to
optimize, submodular functions cannot be minimized from samples to within
a desirable guarantee, even when these functions are learnable. In particu-
lar, we show that there is no algorithm for minimizing a submodular function
from polynomially-many samples drawn from any distribution that obtains an
additive approximation of 1/2 o(1), even when the function is PAC-learnable.
Furthermore, we show that this bound is tight: the algorithm which returns
the empty set or ground set each with probability 1/2 achieves at least a 1/2
approximation. Notice that this also implies that in general, there is no learning
algorithm that can produce a surrogate whose minima is close to the minima of
the function we aim to optimize, as otherwise this would contradict our main
result. Technical overview. At a high level, hardness results in optimization
from samples are shown by constructing a family of functions, where the values
of functions in the family are likely to be indistinguishable for the samples, while
having very different optimizers. The main technical difficulty is to construct
a family of functions that concurrently satisfy these two properties (indistin-
guishability and different optimizers), and that are also PAC-learnable. En
route to our main construction, we first construct a family of functions that
are completely indistinguishable given samples drawn from the uniform distri-
bution, in which case we obtain a 1/2 o(1) impossibility result (Section 2). The
2

general result that holds for any distribution requires heavier machinery to
argue about more general families of functions where some subset of functions
can be distinguished from others given samples. Instead of satisfying the two
desired properties for all functions in a fixed family, we show that these prop-
erties hold for all functions in a randomized subfamily (Section 3.2). We then
develop an efficient learning algorithm for the family of functions constructed
for the main hardness result (Section 3.3). This algorithm builds multiple lin-
ear regression predictors and a classifier to direct a fresh set to the appropriate
linear predictor. The learning of the classifier and the linear predictors relies on
multiple observations about the specific structure of this class of functions. 1.1
Related work

The problem of optimization from samples was introduced in the context of
constrained submodular maximization [BRS17, BRS16]. In general, for maxi-
mizing a submodular function under a cardinality constraint, no algorithm can

obtain a constant factor approximation guarantee from any samples. As discussed above, for special classes of submodular functions that can be optimized in polynomial time under a cardinality constraint, and for unconstrained maximization, there are desirable optimization from samples guarantees. It is thus somewhat surprising that submodular minimization, which is an unconstrained optimization problem that is optimizable in polynomial time in the value query model, is hard to optimize from samples. From a technical perspective the constructions are quite different. In maximization, the functions constructed in [BRS17, BRS16] are monotone so the ground set would be an optimal solution if the problem was unconstrained. Instead, we need to construct novel non-monotone functions. In convex optimization, recent work shows a tight 1/2-inapproximability for convex minimization from samples [BS17]. Although there is a conceptual connection between that paper and this one, from a technical perspective these papers are orthogonal. The discrete analogue of the family of convex functions constructed in that paper is not (even approximately) a family of submodular functions, and the constructions are significantly different.

2

Warm up: the Uniform Distribution

As a warm up to our main impossibility result, we sketch a tight lower bound for the special case in which the samples are drawn from the uniform distribution. At a high level, the idea is to construct a function which considers some special subset of ?good? elements that make its value drops when a set contains all such ?good? elements. When samples are drawn from the uniform distribution and ?good? elements are sufficiently rare, there is a relatively simple construction that obfuscates which elements the function considers ?good?, which then leads to the inapproximability. 2.1

Hardness for uniform distribution

p We construct a family of functions F where fi 2 F is defined in terms of a set G_i ? N of size n. For each such function we call G_i the set of good elements, and B_i = N G_i its bad elements. We denote the number of good and bad elements in a set S by gS and bS , dropping the subscripts (S and i) when clear from context, so g = —G_i S— and b = —B_i S—. The function fi is defined as follows: 8 p 1 1 ¿ ¡ + ? (g + b) if g ¡ n fi (S) := 2 2n p ¿ : 1 ?b if g = n 2n It is easy to verify that these functions are submodular with range in [0, 1] (see illustration in Figure 1a). Given samples drawn uniformly at random (u.a.r.), pit is impossible to distinguish good and bad elements since with high probability (w.h.p.) g ¡ n for all samples. Informally, this implies that a good learner for F over the uniform distribution D is f 0 (S) = 1/2 + —S—/(2n). Intuitively, F is not 1/2 o(1) optimizable from samples because if an algorithm cannot learn the set of good elements G_i , then it cannot find S such fi (S) ¡ 1/2 o(1) whereas the optimal solution Si? = G_i has value fi (G_i) = 0. Theorem 1. Submodular functions f : 2N ! [0, 1] are not 1/2 o(1) optimizable from samples drawn from the uniform distribution for the problem of submodular minimization. 3

Proof. The details for the derivation of concentration bounds are in Appendix B. Consider fk drawn u.a.r. from F and let f ? = fk and G? = Gk . Since the samples are all drawn from the uniform distribution, by standard ap-

plication of the Chernoff bound we have that every set Si in the sample respects —Si — ? 3n/4, w.p. 1 e ?(n) . For sets S1 , ... , Sm , all of size at most 3n/4, when fj is p 1/2 drawn u.a.r. from F we get that —Si Gj — ¡ n, w.p. 1 e ?(n) for all i 2 [m], again by 1/2 Chernoff, and since m = poly(n). Notice that this implies that w.p. 1 e ?(n) for all i 2 [m]: 1 —Si — fj (Si) = + 2 2n Now, let F 0 be the collection of all functions fj for which fj (Si) = 1/2 + —Si —/(2n) on all sets 1/2 0 {Si }m e ?(n))—F—. Thus, since f ? is drawn u.a.r. i=1 . The argument above implies that —F — = (1 1/2 from F we have that f ? 2 F 0 w.p. 1 e ?(n) , and we condition on this event. Let S be the (possibly randomized) solution returned by the algorithm. Observe that S is independent of f ? 2 F 0 . In other words, the algorithm cannot learn any information about which function in F 0 1/6 generates the samples. By Chernoff, if we fix S and choose f u.a.r. from F, then, w.p. 1 e ?(n) : 1 f (S) o(1). 2 0

1/2

1/6

— Since —F e ?(n) , it is also the case that f ? (S) 1/2 o(1) w.p. 1 e ?(n) over the —F — = 1 choice of f ? 2 F 0 . By the probabilistic method and since all the events we conditioned on occur with exponentially high probability, there exists f ? 2 F s.t. the value of the set S returned by the algorithm is 1/2 o(1) whereas the optimal solution is f ? (G?) = 0.

2.2

A tight upper bound

We now show that the result above is tight. In particular, by randomizing between the empty set and the ground set we get a solution whose value is at most 1/2. In the case of symmetric functions, i.e. f (S) = f (N S) for all S ? N , ; and N are minima since f (N) + f (;) ? f (S) + f (N S) for all S ? N as shown below.1 Notice, that this does not require any samples. Proposition 2. The algorithm which returns the empty set ; or the ground N with probability 1/2 each is a 1/2 additive approximation for the problem of unconstrained submodular minimization. Proof. Let S ? N , observe that f (N S) f (;) f ((N S) [S) f (S) = f (N) where the inequality is by submodularity. Thus, we obtain 1 1 1 (f (N) + f (;)) ? f (S) + f (N S) ? f (S) + 2 2 2 In particular, this holds for S 2 argminT ?N f (T).

3

f (S) 1 . 2

General Distribution

In this section, we show our main result, namely that there exists a family of submodular functions such that, despite being PAC-learnable for all distributions, no algorithm can obtain an approximation better than 1/2 o(1) for the problem of unconstrained minimization. The functions in this section build upon the previous construction, though are inevitably more involved in order to achieve learnability and inapproximability on any distribution. The functions constructed for the uniform distribution do not yield inapproximability for general distributions due to the fact that the indistinguishability between two functions no longer holds when sets S of large size are sampled with non-negligible probability. Intuitively, in the previous construction, once a set is

sufficiently large the good elements of the function can be distinguished from the bad ones. The main idea to get around this issue is to introduce masking elements M . We construct functions such that, for sets S of large size, good and bad elements are indistinguishable if S contains at least one masking element. [1]

Although ; and N are trivial minima if f is symmetric, the problem of minimizing a symmetric submodular function over proper nonempty subsets is nontrivial (see [Que98]).

4
1
f(S)
f(S)
1
0.5
0.5 m = —S— b = —S— g = —S— b = —S—-1, m=1 g = —S—-1, m=1
b = —S— g = —S—
0
0 ?n - 1 ?n
0
n ? ?n —S—
(a) Uniform distribution
0
1 n1/4
?n —S—
n/2
(b) General distribution
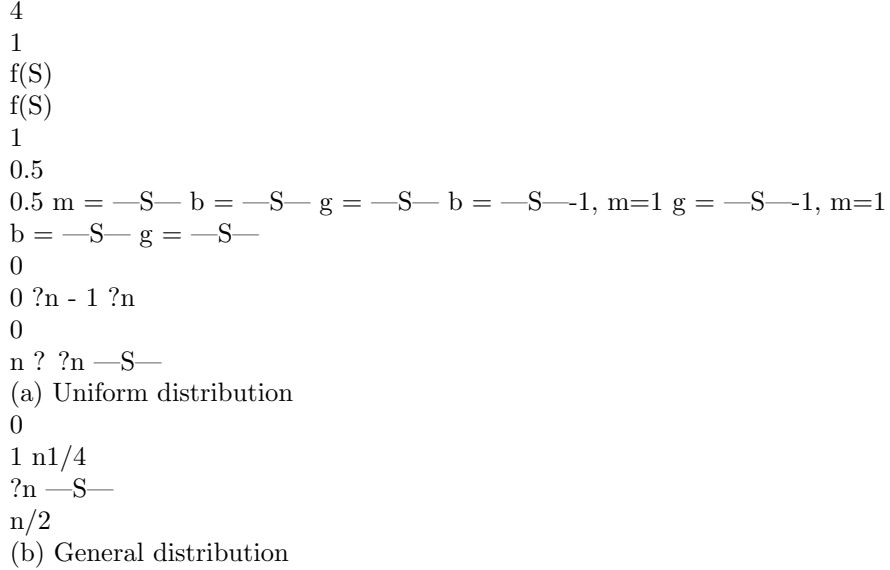
Figure 1: An illustration of the value of a set S of good (blue), bad (red), and masking (green) elements as a function of —S— for the functions constructed. For the general distribution case, we also illustrate the value of a set S of good (dark blue) and bad (dark red) elements when S also contains at least one masking element.

The construction. Each function fi 2 F is defined in terms of a partition Pi of the ground set into good, bad, andpmasking elements. The p partitions we consider are Pi = (Gi , Bi , Mi) with —Gi — = n/2, —Bi — = n, and —Mi — = n/2 n. Again, when clear from context, we drop indices of i and S and the number of good, bad, and masking elements in a set S are denoted by g, b, and m. For such a given partition Pi , the function fi is defined as follows (see illustration in Figure 1b): 8 1 ¿ ¿ p ? (b + g) Region X : if m = 0 and g ¡ n1/4 ¿ ¿2 n ¿ ¿ ? ? 1 ? ? 1 ¡ 1 fi (S) = + p ? b + n1/4 ? g n1/4 Region Y : if m = 0 and g n1/4 2 ¿ n 2 n ¿ ¿ ¿ ¿ 1 ¿ :1 ? (b + g) Region Z : otherwise 2 n 3.1

Submodularity

In the appendix, we prove that the functions fi constructed pas above are indeed submodular (Lemma 10). By rescaling fi with an additive term of n1/4 /(2 n) = 1/(2n1/4), it can be easily verified that its range is in [0, 1]. We use the non-normalized definition as above for ease of notation. 3.2

Inapproximability

We now show that F cannot be minimized within a 1/2 o(1) approximation given samples from any distribution. We first define F M , which is a randomized subfamily of F. We then give a general lemma that shows that if two conditions of indistinguishability and gap are satisfied then we obtain inapproximability. We then show that these two conditions are satisfied for the subfamily F M . A randomization over masking elements. Instead of considering a function f drawn u.a.r. from F as in the uniform case, we consider functions f in a randomized subfamily of functions F M ? F to obtain the indistinguishability and gap the family of functions F, let M be a p conditions. Given uniformly random subset of size n/2 n and define F M ? F: F M := {fi 2 F : (Gi , Bi , M)}.

Since masking elements are distinguishable from good and bad elements, they need to be the same set of elements for each function in family F M to obtain indistinguishability of functions in F M . The inapproximability lemma. In addition to this randomized subfamily of functions, another main conceptual departure of the following inapproximability lemma from the uniform case is that no assumption can be made about the samples, such as their size, since the distribution is arbitrary. We denote by U (A) the uniform distribution over the set A. 5

Lemma 3. Let F be a family of functions and F 0 = {f1 , . . . , fp } ? F be a subfamily of functions drawn from some distribution. Assume the following two conditions hold: 1/4

1. Indistinguishability. For all S ? N , w.p. 1 e ?(n fi (S) = fj (S);
)
over F 0 : for every fi , fj 2 F 0 ,

2. ?-gap. Let Si? be a minimizer of fi , then w.p. 1 over F 0 : for all S ? N , E 0 [fi (S) fi (Si?)] ?; fi ?U (F)

Then, F is not ?-minimizable from strictly less than e?(n
1/4
)
samples over any distribution D.

Note that the ordering of the quantifiers is crucial. The proof is deferred to the appendix, but the main ideas are summarized as follows. We use a probabilistic argument to switch from the randomization over F 0 to the randomization over S ? D and show that there exists a deterministic F ? F such that fi (S) = fj (S) for all fi , fj 2 F w.h.p. over S ? D. By a union bound this holds for all samples S. Thus, for such a family of functions F = {f1 , . . . , fp }, the choices of an algorithm that is given samples from fi for i 2 [p] are independent of i. By the ?-gap condition, this implies that there exists fi 2 F for which a solution S returned by the algorithm is at least ? away from fi (Si?). Indistinguishability and gap of F. We now show the indistinguishability and gap conditions, with ? = 1/2 o(1), which immediately imply a 1/2 o(1) inapproximability by Lemma 3. For the indistinguishability, it suffices to show that good and bad elements are indistinguishable since the masking elements are identical for all functions in F M . Good and bad elements are indistinguishable since, w.h.p., a set S is not in region Y, which is the only region distinguishing good and bad elements. Lemma 4. For all S ? N s.t. —S— ¡ n1/4 : For all fi 2 F M

, (1 p ? (b + g) if m = 0 (Region X) 1 fi (S) = + 21 n 1 2 otherwise (Region Z) 2 n ? (b + g) 1/4

and for all S ? N such that —S—

n1/4 , with probability 1 e ?(n) over F M : For all fi 2 F M , 1 fi (S) = 1 ? (b + g) (Region Z). n

Proof. Let S ? N . If —S— ¡ n1/4 , then the proof follows immediately from the definition of fi . If —S— n1/4 , then, the number of masking elements m in S is m = —M S— for all fi 2 F M . We 1/4 then get m 1, for all fi 2 F M , with probability 1 e ?(n) over F M by Chernoff bound. The proof then follows again immediately from the definition of fi . Next, we show the gap. The gap is since the good elements can be any subset of N M . Lemma 5. Let Si? be a minimizer of fi . With probability 1 over F M , for all S ? N , 1 E [fi (S)] o(1). M 2 fi ?U (F) Proof. Let S ? N and fi ? U (F M). Note that the order of the quantifiers in the statement of the lemma implies that S can be dependent on M , but that it is independent of i. There are three cases. If m 1, then S is in region Z and fi (S) 1/2. If m = 0 and —S— ? n7/8 , then S is in region X or Y and fi (S) 1/2 n7/8 /n = 12 o(1). Otherwise, m = 0 and —S— n7/8 . Since S is independent of i, by Chernoff bound, we get p p n/2 + n n/2 + n p (1 o(1)) ? —S— ? ? b and ? g ? (1 + o(1)) ? —S— n/2 n with probability 1 1 fi (S) + (1 2

?(n1/4)

. Thus S is in region Y and p 1 n 1 n/2 p ? —S— (1 + o(1)) ? p ? —S— o(1)) p ? n n/2 + n 2 n n/2 + n

e

Thus, we obtain Efi ?U (F M) [fi (S)]

1 2

o(1). 6

1 2

o(1).

Combining the above three lemmas, we obtain the inapproximability result. Lemma 6. The problem of submodular minimization cannot be approximated with a 1/2 additive approximation given poly(n) samples from any distribution D.

o(1)

Proof. For any set S ? N , observe that the number g + b of elements in S that are either good or bad is the same for any two functions fi , fj 2 F M and for any F M . Thus, by Lemma 4, we obtain the indistinguishability condition. Next, the optimal solution Si? = Gi of fi has value fi (Gi) = o(1), so by Lemma 5, we obtain the ?-gap condition with ? = 1/2 o(1). Thus F is not 1/2 o(1) minimizable from samples from any distribution D by Lemma 3. The class of functions F is a class of submodular functions by Lemma 10 (in Appendix C).

3.3

Learnability of F

We now show that every function in F is efficiently learnable from samples drawn from any distribution D. Specifically, we show that for any ?, 2 (0, 1) the functions are (?,) PAC learnable with the absolute loss function (or any

Lipschitz loss function) using poly(1/?, 1/ , n) samples and running time. At a high level, since each function fi is piecewise-linear over three different regions Xi , Yi , and Zi , the main idea is to exploit this structure by first training a classifier to distinguish between regions and then apply linear regression in different regions. The learning algorithm. Since every function f 2 F is piecewise linear over three different regions, there are three different linear functions fX , fY , fZ s.t. for every S ? N its value f (S) can be expressed as fR (S) for some region R 2 {X , Y, Z}. The learning algorithm produces a predictor f? by using a multi-label classifier and a set of linear predictors {fX? , fY? } [{[i2M? fZ?i }. ? [{[? Z?i }, The multi-label classifier creates a mapping from sets to regions, g : 2N ! {X? , Y} i2M ? ? ? s.t. X , Y, Z are approximated by X , Y, [i2M? Zi . Given a sample S ? D, using the algorithm then retuns f?(S) = fg(S) (S). We give a formal description below (detailed description is in Appendix D). Algorithm 1 A learning algorithm for f 2 F which combines classification and linear regression.

Input: samples S = {(Sj , f (Sj))}j2[m] ? M ?) (Z, (;, ;) for i = 1 to n do ? Z?i {S : ai 2 S, S 62 Z} reg fZ?i ERM ({(Sj , f (Sj)) : Sj 2 Z?i }) linear regression P if (Sj ,f (Sj)):Sj 2Z?i —fZ?i (Sj) f (Sj)— = 0 then ? ? [{ai } Z? Z? [Z?i , M M cla ? j ? m/2) C ERM ({(Sj , f (Sj)) : Sj 62 Z, train a classifier for regions X , Y ? ? C(S) = 1}, {S : S 62 Z, ? C(S) = 1}) (X? , Y) ({S : S 62 Z, 8 p if S2X? ¡—S—/(2 n) reg ? ? return f S 7! fY? (S) = ERM ({(Sj , f (Sj)) : Sj 2 Y, j ¿ m/2}) if S2Y? : ? }) ? fZ?i (S) : i = min({i0 : ai0 2 S M if S2Z

Overview of analysis of the learning algorithm. There are two main challenges in training the algorithm. The first is that the region X , Y, or Z that a sample (Sj , f (Sj)) belongs to is not known. ? Z? using the samples, Thus, even before being able to train a classifier which learns the regions X? , Y, we need to learn the region a sample Sj belongs to using f (Sj). The second is that the samples SR used for training a linear regression predictor fR over region R need to be carefully selected so that SR is a collection of i.i.d. samples from the distribution S ? D conditioned on S 2 R (Lemma 20).

We first discuss the challenge of labeling samples with the region they belong to. Observe that for a fixed masking element ai 2 M , f 2 F is linear over all sets S containing ai since these sets are all in region Z. Thus, there must exist a linear regression predictor fZ?i = ERMreg (?) with zero empirical loss over all samples Sj containing ai if ai 2 M (and thus Sj 2 Z). ERMreg (?) minimizes the empirical loss on the input samples over the class of linear regression predictors with bounded norm 7

Figure 2: An illustration of the regions. The dots represent the samples, the corresponding full circles represent ? Z? learned by the the regions X (red), Y (blue), and Z (green). The ellipsoids represent the regions X? , Y, ? classifier. Notice that Z has no false negatives.

(Lemma 19). If fZ?i has zero empirical loss, we directly classify any set S containing ai as being in ? Next, for a sample (Sj , f (Sj)) not in Z, ? we can label these samples since Sj 2 X if and only if Z. p f (Sj) = —Sj —/(2 n). With these labeled samples S 0 , we train a binary classifier C = ERMcla (S 0) ? ERMcla (S 0) minimizes the empirical loss on that indicates if S s.t. S 62 Z?

9

is in region X? or Y. 0 labeled samples S over the class of halfspaces w 2 Rn (Lemma 23).

Regarding the second challenge, we cannot use all samples Sj s.t. Sj 2 Y? to train a linear predictor ? so they are not a collection of i.i.d. fY? for region Y? since these same samples were used to define Y, ? samples from the distribution S ? D conditioned on S 2 Y. To get around this issue, we partition the samples into two distinct collections, one to train the classifier C and one to train fY? (Lemma 24). ? we predict f ? (T) where i is s.t. ai 2 T M ? (breaking ties lexicographically) Next, given T 2 Z, Zi ? ? (Lemma 22). Since we break which performs well since fZ?i has zero empirical error for ai 2 M ties lexicographically, f?Z?i must be trained over samples Sj such that ai 2 Sj and ai0 62 Sj for i0 s.t. ? to obtain i.i.d. samples from the same distribution as T ? D conditioned on T i0 ¡ i and ai0 2 M being directed to f?Z?i (Lemma 21).

The analysis of the learning algorithm leads to the following main learning result. Lemma 7. Let f? be the predictor returned by Algorithm 1, then w.p. 1 over m 2 O(n3 + 2 2 n (log(2n/))/?) samples S drawn i.i.d. from any distribution D, ES?D [—f?(S) f (S)—] ? ?. 3.4

Main Result

We conclude this section with our main result which combines Lemmas 6 and 7. Theorem 8. There exists a family of [0, 1]-bounded submodular functions F that is efficiently PAClearnable and that cannot be optimized from polynomially many samples drawn from any distribution D within a 1/2 o(1) additive approximation for unconstrained submodular minimization.

4

Discussion

In this paper, we studied the problem of submodular minimization from samples. Our main result is an impossibility, showing that even for learnable submodular functions it is impossible to find a non-trivial approximation to the minimizer with polynomially-many samples, drawn from any distribution. In particular, this implies that minimizing a general submodular function learned from data cannot yield desirable guarantees. In general, it seems that the intersection between learning and optimization is elusive, and a great deal still remains to be explored.

8

2 References

[Bal15] Maria-Florina Balcan. Learning submodular functions with applications to multi-agent systems. In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015, page 3, 2015. [BH11] Maria-Florina Balcan and Nicholas JA Harvey. Learning submodular functions. In Proceedings of the forty-third annual ACM symposium on Theory of computing, pages 793?802. ACM, 2011. [BRS16] Eric Balkanski, Aviad Rubinstein, and Yaron Singer. The power of optimization from samples. In Advances in Neural Information Processing Sys-

tems, pages 4017?4025, 2016. [BRS17] Eric Balkanski, Aviad Rubinstein, and Yaron Singer. The limitations of optimization from samples. Proceedings of the Forty-Ninth Annual ACM on Symposium on Theory of Computing, 2017. [BS17] Eric Balkanski and Yaron Singer. The sample complexity of optimizing a convex function. In COLT, 2017. [BVW16] Maria-Florina Balcan, Ellen Vitercik, and Colin White. Learning combinatorial functions from pairwise comparisons. In Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016, pages 310?335, 2016. [CLSW16] Deeparnab Chakrabarty, Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. Subquadratic submodular function minimization. arXiv preprint arXiv:1610.09800, 2016. [DTK16] Josip Djolonga, Sebastian Tschiatschek, and Andreas Krause. Variational inference in mixed probabilistic submodular models. In Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, pages 1759?1767, 2016. [EN15] Alina Ene and Huy L. Nguyen. Random coordinate descent methods for minimizing decomposable submodular functions. In Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, pages 787?795, 2015. [FK14] Vitaly Feldman and Pravesh Kothari. Learning coverage functions and private release of marginals. In COLT, pages 679?702, 2014. [FKV13] Vitaly Feldman, Pravesh Kothari, and Jan Vondr?k. Representation, approximation and learning of submodular functions using low-rank decision trees. In COLT, pages 711?740, 2013. [FMV11] Uriel Feige, Vahab S. Mirrokni, and Jan Vondr?k. Maximizing non-monotone submodular functions. SIAM J. Comput., 40(4):1133?1153, 2011. [GLS81] Martin Grotschel, Laszlo Lovasz, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. Combinatorica, 1(2):169?197, 1981. [IFF01] Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. J. ACM, 48(4):761?777, 2001. [IJB13] Rishabh K. Iyer, Stefanie Jegelka, and Jeff A. Bilmes. Curvature and optimal algorithms for learning and minimizing submodular functions. In Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States., pages 2742?2750, 2013. [JB11] Stefanie Jegelka and Jeff Bilmes. Submodularity beyond submodular energies: coupling edges in graph cuts. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 1897?1904. IEEE, 2011. [JLB11] Stefanie Jegelka, Hui Lin, and Jeff A Bilmes. On fast approximate submodular minimization. In Advances in Neural Information Processing Systems, pages 460?468, 2011. [NB12] Mukund Narasimhan and Jeff A Bilmes. A submodular-supermodular procedure with applications to discriminative structure learning. arXiv preprint arXiv:1207.1404, 2012. [Que98] Maurice Queyranne. Minimizing symmetric submodular functions. Mathematical Programming, 82(1-2):3?12, 1998. [SK10] Peter Stobbe and Andreas Krause. Efficient minimization of decomposable submodular functions. In Advances in Neural Information Processing Systems, pages 2208?2216, 2010. [SSBD14] Shai Shalev-Shwartz and Shai Ben-David. Understanding machine learning: From

theory to algorithms. Cambridge university press, 2014.
  9