

Structure and Learning of Valuation Functions

VITALY FELDMAN

and

JAN VONDRÁK

IBM Almaden Research Center

We discuss structural results and learning algorithms for submodular and fractionally subadditive valuation functions. While learning these valuation functions over general distributions turns out to be hard, we present compact approximate representations and efficient learning algorithms for such functions over the uniform distribution.

Categories and Subject Descriptors: I.2.6 [Artificial Intelligence]: Learning

General Terms: Economics, Theory

Additional Key Words and Phrases: Valuation Functions, Submodular Functions, PAC Learning

1. INTRODUCTION

In this letter, we discuss the structure and learnability of several classes of real-valued functions that are used to model *valuation functions*. A *valuation function* $f : \{0, 1\}^n \rightarrow \mathbb{R}^+$ is a function that assigns a value $f(x)$ to a bundle of goods represented by x , or a product with attributes represented by x (we identify functions on $\{0, 1\}^n$ with set functions on $[n] = \{1, 2, \dots, n\}$ in a natural way). Valuation functions have been a fundamental tool to express the preferences of agents in economic settings, such as combinatorial auctions [Lehmann et al. 2006].

The primary class of functions that we consider here is the class of submodular functions. Recent interest in submodular functions has been fueled by their role in algorithmic game theory, as valuations with the property of *diminishing returns* [Lehmann et al. 2006]. Along with submodular functions, other related classes have been studied in the context of algorithmic game theory context: budget-additive, coverage functions, gross substitutes, fractionally subadditive functions, etc. It turns out that these classes are contained in a broader class, that of *self-bounding functions*, introduced in the context of concentration of measure inequalities [Boucheron et al. 2000]. We briefly summarize the definitions of the classes relevant to our discussion (in the set function notation).

Definition 1.1. A set function $f : 2^{[n]} \rightarrow \mathbb{R}$ is
submodular, if $f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$ for all $A, B \subseteq N$.
fractionally subadditive, if $f(A) \leq \sum \beta_i f(B_i)$ whenever $\beta_i \geq 0$ and $\sum_{i:a \in B_i} \beta_i \geq 1 \forall a \in A$.
subadditive, if $f(A \cup B) \leq f(A) + f(B)$ for all $A \subseteq B \subseteq N$.
 a -self-bounding, if $\sum_{i \in N} (f(S) - f(S \Delta \{i\}))_+ \leq a \cdot f(S)$ for all $S \subseteq N$.

Fractionally-subadditive functions can be equivalently defined as “XOS functions”,

Authors’ addresses: vitaly@post.harvard.edu, jvondrak@us.ibm.com

which are functions in the following form: $f(A) = \max_{1 \leq i \leq t} \sum_{j \in A} a_{i,j}$ for some non-negative constants $a_{i,j}$. This class includes all (nonnegative) monotone submodular functions (but does not contain non-monotone functions). Further, all submodular and XOS functions are also subadditive. XOS functions are 1-self-bounding and submodular functions are 2-self-bounding. However, subadditive and a -self-bounding functions are incomparable.

A natural question, first posed by Balcan and Harvey [2011], is whether valuation functions can be *learned* from random examples. More precisely, given a collection of pairs $(x, f(x))$ for points $x \in \{0, 1\}^n$ sampled randomly and independently from some distribution \mathcal{D} , can we find a function $h : \{0, 1\}^n \rightarrow \mathbb{R}$ which is close to f (under some metric of interest)? For example, companies might want to learn valuations from past data to predict future demand, or to learn the preferences that customers have for different combinations of product features to guide future development.

Balcan and Harvey [2011] introduced the PMAC¹ learning model where the goal is to find a hypothesis that approximates the unknown valuation function within an α multiplicative factor on all but a δ -fraction of points under \mathcal{D} . They gave a distribution-independent $O(\sqrt{n})$ -factor PMAC learning algorithm for submodular functions and showed an information-theoretic factor- $\sqrt[3]{n}$ inapproximability for submodular functions. Subsequently, Balcan et al. [2012] gave a PMAC learning algorithm for XOS functions that achieves an $\tilde{O}(\sqrt{n})$ -factor approximation and showed that this is essentially optimal. These results imply hardness of learning of submodular and XOS functions even with the more relaxed ℓ_2 -error (that is, finding h such that $\|f - h\|_2 = \sqrt{\mathbf{E}_{x \sim \mathcal{D}}[(f(x) - h(x))^2]} \leq \epsilon$). These strong lower bounds rely on a specific distribution supported on a sparse set of points and motivate the question of whether these classes of functions can be learned efficiently over “simpler” distributions such the uniform or product distributions. This setting is the focus of our work and therefore in the following discussion we restrict our attention to learning over the uniform distribution U .

Cheraghchi et al. [2012] showed that submodular functions can be approximated within ℓ_2 -error of ϵ by polynomials of degree $O(1/\epsilon^2)$. This leads to an $n^{O(1/\epsilon^2)}$ -time learning algorithm. Their approximation relies on Fourier analysis on the discrete cube, in particular the analysis of noise stability of submodular functions.

More recently, Feldman et al. [2013] building upon the work of Gupta et al. [2011], showed that submodular functions can be ϵ -approximated in ℓ_2 by a decision tree of depth $O(1/\epsilon^2)$. They used this structural result to give a PAC learning algorithm running in time $2^{O(1/\epsilon^4)} \cdot \tilde{O}(n^2)$. Feldman et al. [2013] also showed that $2^{\Omega(1/\epsilon^{2/3})}$ random examples (or even value queries) are necessary to PAC-learn monotone submodular functions within an ℓ_2 -error of ϵ .

2. JUNTA APPROXIMATIONS AND LEARNING

As the reader may notice, the learning results above rely on various (approximate) compact representations of valuation functions. This is typically the case in machine

¹The PMAC (“Probably *Mostly* Approximately Correct”) learning model is based on the PAC learning model from [Valiant 1984].

learning — if we want to learn an unknown function efficiently, we need to work with a representation of the function that is not too complicated. Perhaps the simplest form of such a representation is a function that depends only on a small subset of its coordinates. Such functions are referred to as *juntas*. Approximation by a junta is fundamental object of study in Boolean function analysis as well as a useful tool in several areas of theoretical computer science (see e.g. [Friedgut 1998; Bourgain 2002; Dinur and Safra 2005; O’Donnell and Servedio 2007]). A classical result in this area is Friedgut’s theorem [Friedgut 1998] which states that every Boolean function f is ϵ -close to a function of $2^{O(\text{Infl}(f)/\epsilon^2)}$ variables, where $\text{Infl}(f) = \sum_{i=1}^n \Pr_{x \sim U}[f(x) \neq f(x \oplus e_i)]$ is the *total influence* of f (also referred to as *average sensitivity*).

A natural question to ask is whether similar results can be proved for real-valued functions and applied to the valuation functions that we study. However, prior to this work [Feldman and Vondrák 2013], an analogue of Friedgut’s theorem for general real-valued functions was not known. In fact, one natural generalization of Friedgut’s theorem using ℓ_2 total influence, $\text{Infl}^2(f) = \sum_{i=1}^n \mathbf{E}_{x \sim U}[(f(x) - f(x \oplus e_i))^2]$, was known not to hold [O’Donnell and Servedio 2007].

In this work [Feldman and Vondrák 2013], we show that Friedgut’s theorem does hold for real-valued functions if we include a (polynomial) dependence on the ℓ_1 total influence, defined as $\text{Infl}^1(f) = \sum_{i=1}^n \mathbf{E}_{x \sim U}[|f(x) - f(x \oplus e_i)|]$, in addition to an exponential dependence on $\text{Infl}^2(f)$. More precisely, we prove that any function $f : \{0, 1\}^n \rightarrow \mathbb{R}$ is ϵ -approximated in ℓ_2 by a function that only depends on $2^{O(\text{Infl}^2(f)/\epsilon^2)} \cdot \text{poly}(\text{Infl}^1(f))$ variables.

We then show that submodular, XOS and indeed all $O(1)$ -self-bounding functions have low total influence in both ℓ_1 and ℓ_2 norms. Specifically for a -self-bounding functions (which includes XOS functions with $a = 1$) with a normalized range, $f : \{0, 1\}^n \rightarrow [0, 1]$, we get $\text{Infl}^2(f) \leq \text{Infl}^1(f) \leq a$. Combined with the real-valued analogue of Friedgut’s theorem this yields the following result.

THEOREM 2.1. *For any $\epsilon \in (0, \frac{1}{2})$ and any $O(1)$ -self-bounding function $f : \{0, 1\}^n \rightarrow [0, 1]$ (which includes XOS functions), there exists $g : \{0, 1\}^n \rightarrow [0, 1]$ depending only on a subset of variables $J \subseteq [n]$, $|J| = 2^{O(1/\epsilon^2)}$ such that $\|f - g\|_2 \leq \epsilon$.*

We show that this result is close to optimal. Namely, there exists an XOS function that requires $2^{\Omega(1/\epsilon)}$ variables to approximate within an ℓ_2 -error of ϵ .

The statement of this theorem for submodular functions follows from the decision tree approximation by Feldman et al. [2013] which, interestingly, is based on completely unrelated techniques. However, it turns out that for submodular functions this bound is not tight and a much stronger approximation result holds: $\tilde{O}(1/\epsilon^2)$ variables are sufficient for an ϵ -approximation.

THEOREM 2.2. *For any $\epsilon \in (0, \frac{1}{2})$ and any submodular function $f : \{0, 1\}^n \rightarrow [0, 1]$, there exists a submodular function $g : \{0, 1\}^n \rightarrow [0, 1]$ depending only on a subset of variables $J \subseteq [n]$, $|J| = O(\frac{1}{\epsilon^2} \log \frac{1}{\epsilon})$, such that $\|f - g\|_2 \leq \epsilon$.*

This result is nearly tight since even linear functions require $\Omega(1/\epsilon^2)$ variables for an ϵ -approximation.

We prove this result using a technique unrelated to the Fourier-analytic proof of Friedgut’s theorem. The proof relies on a new greedy procedure to select the signifi-

cant variables, a *boosting lemma* by Goemans and Vondrák [2006] and concentration properties of Lipschitz submodular functions.

These structural results lead to several new and improved learning algorithms. First, we obtain that results previously known only for submodular functions extend to XOS functions and monotone a -self-bounding functions. (An additional issue in learning is how to identify the important variables but it is easy for monotone functions). Namely, XOS functions can be also learned in the PAC model within ℓ_2 -error ϵ in time $2^{O(1/\epsilon^4)} \cdot \tilde{O}(n)$.

The smaller junta approximation for submodular functions does not improve the PAC learning result of Feldman et al. [2013] dramatically. (As we mentioned, the exponential dependence on ϵ is necessary for PAC learning of submodular functions.) However, what we gain from this junta approximation is (somewhat unexpectedly) a learning result in the PMAC model which was not known before: submodular functions can be learned within a $(1 + \epsilon)$ -multiplicative error, on all but the δ -fraction of $\{0, 1\}^n$, in time $2^{O(1/(\epsilon\delta)^2)} \cdot \tilde{O}(n^2)$. This result requires a more involved recursive procedure to identify the significant variables — we refer the reader to [Feldman and Vondrák 2013] for more details.

Finally, we remark that our results cannot be extended to subadditive functions: they do not admit any junta approximation independent of n and are not amenable to efficient learning within constant ℓ_2 -error.

REFERENCES

- BALCAN, M., CONSTANTIN, F., IWATA, S., AND WANG, L. 2012. Learning valuation functions. In *COLT, JMLR 23*. 4.1–4.24.
- BALCAN, M. AND HARVEY, N. 2011. Learning submodular functions. In *ACM STOC*. 793–802.
- BOUCHERON, S., LUGOSI, G., AND MASSART, P. 2000. A sharp concentration inequality with applications. *Random Struct. Algorithms* 16, 3, 277–292.
- BOURGAIN, J. 2002. On the distribution of the fourier spectrum of boolean functions. *Israel Journal of Mathematics* 131(1), 269–276.
- CHERAGHCHI, M., KLIVANS, A., KOTHARI, P., AND LEE, H. 2012. Submodular functions are noise stable. In *ACM-SIAM SODA*. 1586–1592.
- DINUR, I. AND SAFRA, S. 2005. On the hardness of approximating minimum vertex cover. *Annals of Mathematics* 162, 439–485.
- FELDMAN, V., KOTHARI, P., AND VONDRÁK, J. 2013. Representation, approximation and learning of submodular functions using low-rank decision trees. In *COLT, JMLR 30*. 711–740.
- FELDMAN, V. AND VONDRÁK, J. 2013. Optimal bounds on approximation of submodular and XOS functions by juntas. In *IEEE FOCS*. 227–236.
- FRIEDGUT, E. 1998. Boolean functions with low average sensitivity depend on few coordinates. *Combinatorica* 18, 1, 27–35.
- GOEMANS, M. AND VONDRÁK, J. 2006. Covering minimum spanning trees of random subgraphs. *Random Struct. Algorithms* 29:3, 257–276.
- GUPTA, A., HARDT, M., ROTH, A., AND ULLMAN, J. 2011. Privately releasing conjunctions and the statistical query barrier. In *ACM STOC*. 803–812.
- LEHMANN, B., LEHMANN, D. J., AND NISAN, N. 2006. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior* 55, 1884–1899.
- O’DONNELL, R. AND SERVEDIO, R. 2007. Learning monotone decision trees in polynomial time. *SIAM J. Comput.* 37, 3, 827–844.
- VALIANT, L. G. 1984. A theory of the learnable. *Communications of the ACM* 27, 11, 1134–1142.