Learning Submodular Functions*

Maria-Florina Balcan[1] and Nicholas J. A. Harvey[2]

[1] Georgia Institute of Technology, School of Computer Science.
[2] University of British Columbia

Abstract. Submodular functions are discrete functions that model laws of diminishing returns and enjoy numerous algorithmic applications that have been used in many areas, including combinatorial optimization, machine learning, and economics. In this work we use a learning theoretic angle for studying submodular functions. We provide algorithms for learning submodular functions, as well as lower bounds on their learnability. In doing so, we uncover several novel structural results revealing both extremal properties as well as regularities of submodular functions, of interest to many areas.

Submodular functions are a discrete analog of convex functions that enjoy numerous applications and have structural properties that can be exploited algorithmically. They arise naturally in the study of graphs, matroids, covering problems, facility location problems, etc., and they have been extensively studied in operations research and combinatorial optimization for many years [8]. More recently submodular functions have become key concepts both in the machine learning and algorithmic game theory communities. For example, submodular functions have been used to model bidders' valuation functions in combinatorial auctions [12, 6, 3, 14], and for solving feature selection problems in graphical models [11] or for solving various clustering problems [13]. In fact, submodularity has been the topic of several tutorials and workshops at recent major conferences in machine learning [1, 9, 10, 2].

Despite the increased interest on submodularity in machine learning, little is known about the topic from a learning theory perspective. In this work, we provide a statistical and computational theory of learning submodular functions in a distributional learning setting.

Our study has multiple motivations. From a foundational perspective, submodular functions are a powerful, broad class of important functions, so studying their learnability allows us to understand their structure in a new way. To draw a parallel to the Boolean-valued case, a class of comparable breadth would be the class of monotone Boolean functions; the learnability of such functions has been intensively studied [4, 5]. From an applications perspective, algorithms for learning submodular functions may be useful in some of the applications where

* This note summarizes several results in the paper "Learning Submodular Functions", by Maria Florina Balcan and Nicholas Harvey, which appeared The 43rd ACM Symposium on Theory of Computing (STOC) 2011.

these functions arise. For example, in the context of algorithmic game theory and economics, an auctioneer may use such an algorithm to sketch the players' valuation functions before designing the auction, companies might want to learn the valuation functions of their customers to in order to predict demand, etc. More broadly, the problem of learning submodular functions is natural for a wide range of settings where one would like to predict the value of some function over objects described by features, where the features have positive but decreasing marginal impact on the function's value. Examples include predicting the rate of growth of jobs in cities as a function of various amenities or enticements that the city offers, predicting the sales price of a house as a function of features (such as an updated kitchen, hardwood floors, extra bedrooms, etc.) that it might have, and predicting the demand for a new laptop as a function of various add-ons which might be included. In all of these settings (and many others) it is natural to assume diminishing returns, making them well-suited to a formulation as a problem of learning a submodular function.

To study the learnability of submodular functions, we introduce a learning model for approximate distributional learning, which can be described as follows. There is an underlying, fixed but unknown distribution over the subsets of the ground set and a fixed but unknown submodular target function f^*, and the goal is to algorithmically provide a good approximation of the target function with respect to the underlying distribution, in polynomial time, based on a polynomial number of samples from the underlying distribution. Formally, the goal is to output a hypothesis function f that, with probability $1 - \delta$ over the choice of examples, is a good approximation of the target f^* on most of the points coming from D. Here "most" means a $1 - \epsilon$ fraction and "good approximation" means that $f(S) \leq f^*(S) \leq \alpha \cdot f(S)$ for some approximation factor α. Our results on learning submodular functions are presented in this new model, which we call the *PMAC model*; this abbreviation stands for "Probably Mostly Approximately Correct". Note that this learning model differs from the usual PAC-learning model. In our model, one must *approximate* the value of a function on a set of large measure, with high confidence. In contrast, the traditional PAC-learning model usually studies learnability of much simpler classes of Boolean functions. There, one must compute the value *exactly* on a set of large measure, with high confidence.

We prove nearly matching $\alpha = O(n^{1/2})$ upper and $\alpha = \tilde{\Omega}(n^{1/3})$ lower bounds on the approximation factor achievable when the algorithm receives only $poly(n, 1/\epsilon, 1/\delta)$ examples from an arbitrary (fixed but unknown distribution). We additionally provide a better constant approximation factor learning algorithm for the case where the underlying distribution is a product distribution, which is based on a new result showing a strong concentration of submodular functions.

We start by showing that it is possible to PMAC-learn the general class of non-negative, monotone submodular functions with an approximation factor of $\sqrt{n+1}$. To prove this we use a structural result in [7] which shows that any monotone, non-negative submodular function can be approximated within a factor of $\sqrt{n+1}$ on every point by the square root of an additive function. Using this result, we show how to convert the problem of learning a submodular function in the PMAC model to the problem of learning a linear separator in R^{n+1} in the usual PAC model. We remark that an improved structural result for any subclass of submodular functions immediately implies an improved analysis of our algorithm for that subclass.

We introduce a new family of matroids to show a comparable lower bound: any algorithm that uses a polynomial number of examples cannot PMAC-learn the class of submodular functions with an approximation factor $o(n^{1/3}/\log n)$. In fact, we show that even *weak* PMAC-learning is not possible — any algorithm can do only negligibly better than random guessing for this class of functions. Moreover, this lower bound holds even if the algorithm is told the underlying distribution and it is given the ability to query the function on inputs of its choice and even if the queries are adaptive. In other words this lower bound holds even in the PMAC model augmented with value queries.

This lower bound holds even for matroid rank functions, but it uses a distribution on inputs which is a non-product distribution. It turns out that the use of such a distribution is necessary: using Talagrand's inequality, we prove that a constant approximation factor can be achieved for matroid rank functions under product distributions.

To prove the lower bound, we consider the following technical problem. We would like find an injective map $\rho : \{0,1\}^d \to \{0,1\}^n$ and real numbers $\alpha \ll \beta$ such that every Boolean function f on $\{0,1\}^d$ can be mapped to a non-negative, monotone, submodular function $\tilde{f}$ on $\{0,1\}^n$ satisfying $f(x) = 0 \Rightarrow \tilde{f}(\rho(x)) \leq \alpha$ and $f(x) = 1 \Rightarrow \tilde{f}(\rho(x)) \geq \beta$. This implies a lower bound on learning submodular functions with approximation factor $\frac{\beta}{\alpha}$ when $d = \omega(\log n)$. A trivial construction is obtained using partition matroids, with $\alpha = 0$, $\beta \leq \frac{n}{2}$ and $d \leq \log(\lfloor n/\beta \rfloor)$; here d is too small to be of interest. Another easy construction is obtained using paving matroids, with $\alpha = \frac{n}{2}$, $\beta = \frac{n}{2} + 1$, and any $d = n - \Omega(\log^4(n))$; here d is large, but there is only a small additive gap between α and β. Our new family of matroids is a common generalization of partition and paving matroids. We use them to obtain a construction with $\alpha = 16d$, $\beta = n^{1/3}$ and any $d = o(n^{1/3})$; this gives a large multiplicative gap between α and β.

Our work has several interesting by-products. One is the PMAC-learning model, which studies both the probability mass of points on which the hypothesis does well and the multiplicative approximation achieved on those points. Another by-product of our work is our new family of matroids which reveals interesting extremal properties of submodular functions. Roughly speaking, we show that a small Boolean cube can be embedded into a large Boolean cube so that any $\{0,1\}$-valued function on the small cube maps to a function that is submodular on the large cube but is now $\{\alpha, \beta\}$-valued with $\alpha \ll \beta$ (on the points to which the small cube was embedded). This lower bound

References

1. NIPS workshop on discrete optimization in machine learning: Submodularity, sparsity & polyhedra (DISCML), 2009. `http://www.discml.cc/`.
2. NIPS workshop on discrete optimization in machine learning (discml): Uncertainty, generalization and feedback, 2011. `http://las.ethz.ch/discml/`.
3. M.F. Balcan, A. Blum, and Y. Mansour. Item pricing for revenue maxmimization. In *ACM Conference on Electronic Commerce*, 2009.
4. A. Blum, C. Burch, and J. Langford. On learning monotone boolean functions. In *FOCS*, 1998.
5. D. Dachman-Soled, T. Malkin K. Lee, R. Servedio, A. Wan, and H. Wee. Optimal cryptographic hardness of learning monotone functions. In *ICALP*, 2008.
6. S. Dobzinski, N. Nisan, and M. Schapira. Truthful Randomized Mechanisms for Combinatorial Auctions. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 644–652, 2006.
7. M. Goemans, N. Harvey, S. Iwata, and V. Mirrokni. Approximating submodular functions everywhere. In *ACM-SIAM Symposium on Discrete Algorithms*, 2009.
8. M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer Verlag, 1993.
9. A. Krause and C. Guestrin. Beyond convexity: Submodularity in machine learning, 2008. http://www.select.cs.cmu.edu/tutorials/icml08submodularity.html.
10. A. Krause and C. Guestrin. Intelligent information gathering and submodular function optimization, 2009. http://submodularity.org/ijcai09/index.html.
11. Andreas Krause and Carlos Guestrin. Near-optimal nonmyopic value of information in graphical models. In *UAI*, 2005.
12. B. Lehmann, D. J. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55:270–296, 2006.
13. M. Narasimhan and J. Bilmes. Local search for balanced submodular clusterings. In *Twentieth International Joint Conference on Artificial Intelligence*, 2007.
14. J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*, 2008.