A Resampling Technique for Learning the Fourier Spectrum of Skewed Data

Rajeev Ayyagari and Hillol Kargupta
University of Maryland, Baltimore County
{arajeev1, hillol}@cs.umbc.edu

Abstract

Function induction using the widely studied Walsh or Multidimensional Discrete Fourier Transform (MDFT) coefficient estimates has several benefits, including the fact that decision trees can be constructed efficiently from the spectrum. While these estimates are accurate for uniform data, highly skewed data is the norm. This paper gives a way of improving the accuracy of the MDFT coefficient estimates in the case of skewed data. An adaptive resampling algorithm for learning the MDFT coefficients is presented and verified experimentally. An equivalent estimator that can be learned using a single pass algorithm is defined. The effectiveness of the technique is demonstrated using controlled experiments.

1 Introduction

The Multidimensional Discreter Fourier Transform (MDFT) [1] is a popular way to study functions defined on discrete spaces. It has been studied in the field of genetic algorithms [8, 3, 9]. The MDFT can also be used to learn functions by estimating the coefficients. This has been studied in the case of the uniform distribution for AC^0 functions [11], and for more general functions [10, 14]. The issue of skewed distributions is addressed in [7]. However, the membership queries model used in these papers is often inappropriate in practical situations. In addition, the Fourier spectrum has several extremely practical applications which the alternative basis in [7] may not. Decision trees have an exponential decay property in their Fourier spectrum [13, 12, 11]. This has been exploited in learning decision trees efficiently using their Fourier spectrum [13, 12]. It is possible to convert a decision tree to its Fourier spectrum and vice versa efficiently [13]. It has been shown that the Fourier representation of a decision tree can be used to aggregate ensembles of decision trees to produce simpler decision trees or learn decision trees from time-changing data streams [13, 12]. The Fourier spectrum has the additional benefit over several data mining or machine learning algorithms that it is amenable to analysis.

One of the hurdles in the use of the Fourier spectrum as a learning tool is the requirement that the data be roughly uniformly distributed. If the features are Boolean, this means that the probability that a feature takes a value of one is equal to the probability that it equals zero. However, this is a scenario that is rarely encountered in the real world, where skewed data abounds. Common examples of highly skewed real-world data include market basket data and weblog data. It is safe to say that *any* process that can be modeled using a Poisson or exponential distribution, when converted to categorical attributes, gives rise to skewed data. Studying the effect of skewness on the usual Fourier coefficient estimates gives us a way of understanding how we could correct inaccuracies in the estimates. This paper is a preliminary attempt at learning the Fourier spectrum *accurately* from highly skewed data.

This paper is based on the observation that the bias introduced into the coefficient estimates by the non-uniformity of the data can be decomposed into two parts. As this paper shows, one of these parts can be dealt with using adaptive resampling techniques. Resampling techniques including the bootstrap [4, 5], boosting [6], bagging [1] and arcing [2] have been the subject of extensive study in combination with many learning algorithms. The scheme in this paper is less general in the sense that it is specialized towards the learning of Fourier coefficients only.

In order to explain the motivation for the specific resampling scheme used here, we first define the MDFT and examine the effect of skewed distributions on the bias of the estimates (Section 2). In Section 3 we study the source of this bias and identify a way of eliminating part of the bias. We then give a more efficient, equivalent way of computing the estimates. We then present experimental results to corroborate

[1] The MDFT is also referred to as the Walsh transform, especially in the Genetic Algorithms literature.

the theoretical claims made in Section 4. The next section, Section 5, points out areas for future research and concludes the paper.

2 The Effect of the Distribution on MDFT Estimates

Consider the discrete set $X = \{0,1\}^n$. We are interested in learning functions of the form $f : X \to \mathbb{R}$. As usual, we are given a sample from this function: a set of points $S = \{(\mathbf{x}_1, f(\mathbf{x}_1)), (\mathbf{x}_2, f(\mathbf{x}_2)), \ldots, (\mathbf{x}_N, f(\mathbf{x}_N))\}$, where $\mathbf{x}_i \in X \; \forall i = 1, \ldots, N$. The MDFT representation is our chosen representation for these functions for this section.

We describe a way to induce functions in this representation, and study how well the method performs under different distributions of data. To fix notation, we first give a description of the transform.

2.1 The Multidimensional Fourier Transform

The MDFT is recapitulated here. The reader is referred to [8, 3, 9] for further details.

Let $\mathcal{F}$ be the set of all real-valued functions on X. $\mathcal{F}$ forms a 2^n-dimensional vector space over $\mathbb{R}$. The MDFT of a function $f \in \mathcal{F}$ is a representation of the function as a linear combination of the basis set $\{\psi_{\mathbf{j}}(\mathbf{x}) = (-1)^{\mathbf{j}\cdot\mathbf{x}} : \mathbf{j} \in \{0,1\}^n\}$ Here $\mathbf{j} \cdot \mathbf{x}$ is defined as $\sum_{l=1}^{n} \mathbf{j}_l \mathbf{x}_l$, the subscript l denoting the l-th component. This basis is orthonormal with respect to the inner product defined on $\mathcal{F}$ by $\langle f, g \rangle = 1/2^n \sum_{\mathbf{x}\in\{0,1\}^n} f(\mathbf{x})g(\mathbf{x})$. Any $f \in \mathcal{F}$ can be represented *uniquely* as a linear combination of the basis elements:

$$f(\mathbf{x}) = \sum_{\mathbf{j}\in\{0,1\}^n} w_{\mathbf{j}} \psi_{\mathbf{j}}(\mathbf{x}) \tag{1}$$

The coefficients $\{w_{\mathbf{j}} : \mathbf{j} \in \{0,1\}^n\}$ are collectively called the MDFT of f. Given a function f, the MDFT can be computed using the equation:

$$w_{\mathbf{j}} = \langle \psi_{\mathbf{j}}, f \rangle = \frac{1}{2^n} \sum_{\mathbf{x}\in\{0,1\}^n} f(\mathbf{x})\psi_{\mathbf{j}}(\mathbf{x}) \tag{2}$$

Orthogonality of the Fourier basis can be characterized in probabilistic terms. We first note that for any $\mathbf{u}, \mathbf{v} \in X$, $\psi_{\mathbf{u}}(\mathbf{x})\psi_{\mathbf{v}}(\mathbf{x}) = \psi_{\mathbf{w}}(\mathbf{x})$ where $\mathbf{w}$ is the string defined by $\mathbf{w}_l = (\mathbf{v}_l - \mathbf{u}_l) \bmod 2$. We use the notation $\mathbf{w} = \mathbf{u} \oplus \mathbf{v}$. If points are drawn from the space X using a uniform distribution on the points of X. Let $\mathbf{X}$, in boldface, represent a random string drawn from X. Then the basis set $\{\psi_{\mathbf{j}}(\mathbf{x}) : \mathbf{j} \in X\}$ is orthogonal if and only if $E(\psi_{\mathbf{w}}(\mathbf{X})) = 0$ for every nonzero $\mathbf{w}$, where the expectation is taken over uniformly distributed $\mathbf{X}$. Note also that Equation 2 can be rewritten

$$w_{\mathbf{j}} = E\left(f(\mathbf{X})\psi_{\mathbf{j}}(\mathbf{X})\right) \tag{3}$$

2.2 Function Induction using the Fourier Transform

The function induction process in this representation consists of estimating the Fourier coefficients from the data. That is, we wish to find estimates $\{\hat{w}_{\mathbf{j}}\}_{\mathbf{j}\in X}$ for the Fourier coefficients $\{w_{\mathbf{j}}\}_{\mathbf{j}\in X}$ of the underlying function. Using Equation 2 above, $w_{\mathbf{j}}$ can be estimated as

$$\hat{w}_{\mathbf{j}} = \frac{1}{|S|} \sum_{\mathbf{x}_i \in S} f(\mathbf{x}_i)\psi_{\mathbf{j}}(\mathbf{x}_i)$$

if the sample S comes from the uniform distribution.

To demonstrate what goes wrong when the distribution is not uniform, we assume that the bits are independent with probability of success being p for each bit. Let $\mathbf{w} \in X$ be such that $|\mathbf{w}|$, the number of nonzero bits in $\mathbf{w}$, is k. Then it can be shown that $E(\psi_{\mathbf{w}}(\mathbf{X})) = (1-2p)^k$. This implies that $E(\hat{w}_{\mathbf{j}}) = \sum_{\mathbf{i}\in X} w_{\mathbf{i}}(1-2p)^{|\mathbf{i}\oplus\mathbf{j}|}$. (If $|\mathbf{i} \oplus \mathbf{j}| = 0$, $(1-2p)^{|\mathbf{i}\oplus\mathbf{j}|}$ is interpreted as 1.) If p equals 0.5, we are in the uniform case and we get $E[\hat{w}_{\mathbf{j}}] = w_{\mathbf{j}}$. In this case, $\hat{w}_{\mathbf{j}}$ is an unbiased estimate for $w_{\mathbf{j}}$. If $p \neq 0.5$, $\hat{w}_{\mathbf{j}}$ is no longer an unbiased estimate of $w_{\mathbf{j}}$; nor is there a simple correction such as an additive or multiplicative factor that will rectify the bias. (Note that the bias is dependent on the coefficients themselves, which we do not know.) Figure 1 shows this effect on a simulated dataset with 10 features and a second order function with two nonzero coefficients. Of course, we are able to derive an expression for the bias in this case because of the distribution assumption. In the most general situation, where the distribution of the data is not a product distribution, the analysis is more complicated.

3 Learning Functions with Uniformized Data

The analysis in Section 2.2 shows that a uniform distribution is a sufficient condition for the MDFT estimate $\hat{\mathbf{w}}_{\mathbf{j}}$ to be an unbiased estimator of $\mathbf{w}_{\mathbf{j}}$. Since $E(\hat{w}_{\mathbf{j}}) = E\left[f(\mathbf{X})\psi_{\mathbf{j}}(\mathbf{X})\right]$, we restrict our attention to $E[f(\mathbf{X})\psi_{\mathbf{j}}(\mathbf{X})]$. Define a new operator $\otimes$ as follows: $\otimes : X \times X \to \{0,1\}^*$ extracts the features of $\mathbf{x}$ corresponding to 1's in the partition $\mathbf{j}$. (The notation $\{0,1\}^*$ means $\bigcup_{n\in\mathbb{N}}\{0,1\}^n$.) For example, if $X = \{0,1\}^5$, $\mathbf{x} = 01010$, $\mathbf{j} = 11010$, then $\mathbf{x} \otimes \mathbf{j} = 011$.

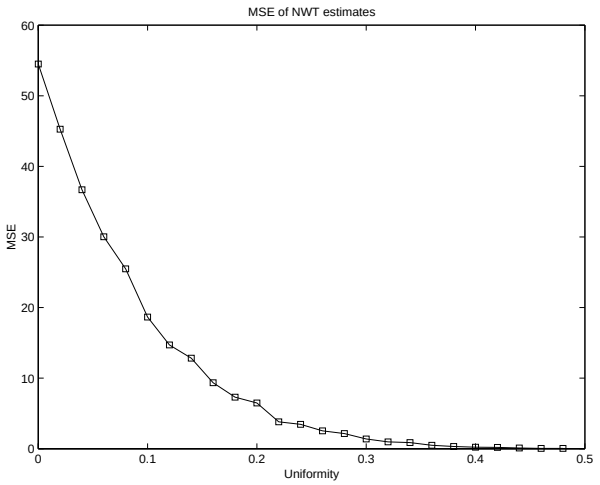

Figure 1: MSE (over the entire space) of the function induced using MDFT estimates. The x-axis represents the probability of success in the data generation.

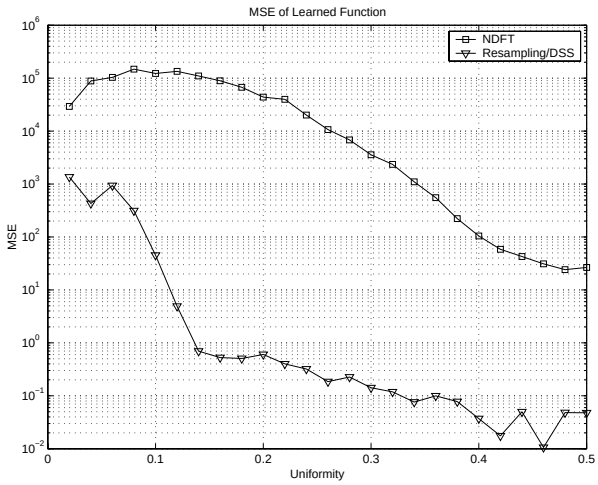

Figure 2: Comparison of the MSE for two learning algorithms (MDFT, Resampling/DSS) for a second-order function with two nonzero coefficients. The x-axis is the probability of success. The farther from .5, the more skewed the data. The y-axis is log-scaled.

Algorithm: Resampling

- For each $\mathbf{j} \in J$ do
  1. $D_{\mathbf{j}}(i) = 1/\text{freq}(\mathbf{x}_i \otimes \mathbf{j})$, $i = 1, \ldots, N$.
  2. Normalize $D_{\mathbf{j}}$ so that it is a probability mass function.
  3. Sample from S according to $D_{\mathbf{j}}$ to get a new sample S' of size N.
  4. Estimate the coefficient $w_{\mathbf{j}}$ using the usual estimates on the new sample S'.

We also use $\neg\mathbf{j}$ to denote the bitwise NOT of $\mathbf{j}$. We now have

$$E(\hat{w}_{\mathbf{j}}) = E\left[f(\mathbf{X})\psi_{\mathbf{j}}(\mathbf{X})\right] = \quad (4)$$
$$E_{\mathbf{U}}\left[E\left(f(\mathbf{X})\psi_{\mathbf{j}}(\mathbf{X})|\mathbf{X}\otimes\mathbf{j} = \mathbf{U}\right)\right] =$$
$$\sum_{\mathbf{u}\in\{0,1\}^{|\mathbf{j}|}} E\left(f(\mathbf{X})\psi_{\mathbf{j}}(\mathbf{X})|\mathbf{X}\otimes\mathbf{j} = \mathbf{u}\right) P(\mathbf{U} = \mathbf{u})$$

Our approach to reducing the bias is to "correct" the non-uniformity in $\mathbf{X}\otimes\mathbf{j}$. The idea is to resample the data according to a distribution that causes $\mathbf{X}\otimes\mathbf{j}$ to be uniformly distributed. The algorithm is described in Figure 3.

We note that while this gives significantly better results in terms of accuracy in many cases (as the experiments show), it is inefficient because we go through a separate resampling procedure for each coefficient we estimate. To overcome this problem, we define a new estimator. If $\mathbf{X}\otimes\mathbf{j}$ were uniformly distributed, Equation 5 would reduce to

$$w^*_{\mathbf{j}} = \frac{1}{2^{|\mathbf{j}|}} \sum_{\mathbf{u}\in\{0,1\}^{|\mathbf{j}|}} E\left(f(\mathbf{X})\psi_{\mathbf{j}}(\mathbf{X})|\mathbf{X}\otimes\mathbf{j} = \mathbf{u}\right) \quad (5)$$
$$= \frac{1}{2^{|\mathbf{j}|}} \sum_{\mathbf{u}\in\{0,1\}^{|\mathbf{j}|}} \psi_{\mathbf{j}\otimes\mathbf{j}}(\mathbf{u}) E\left(f(\mathbf{X})|\mathbf{X}\otimes\mathbf{j} = \mathbf{u}\right) \quad (6)$$

Our new estimator $\hat{w}^*_{\mathbf{j}}$ is based on equation 6. The idea now is to estimate $E\left(f(\mathbf{X})|\mathbf{X}\otimes\mathbf{j} = \mathbf{u}\right)$ for each value of $\mathbf{u}$ and use Equation 6. Let $S_{\mathbf{u},\mathbf{j}} = \{\mathbf{x}\in S|\mathbf{x}\otimes\mathbf{j} = \mathbf{u}\}$. The estimator $\hat{w}^*_{\mathbf{j}}$ is now defined naturally as

$$\hat{w}^*_{\mathbf{j}} = \frac{1}{2^{|\mathbf{j}|}} \sum_{\mathbf{u}\in\{0,1\}^{|\mathbf{j}|}} \psi_{\mathbf{j}\otimes\mathbf{j}}(\mathbf{u}) \hat{f}_{\mathbf{u},\mathbf{j}} \quad \text{where}$$
$$\hat{f}_{\mathbf{u},\mathbf{j}} = \frac{1}{|S_{\mathbf{u},\mathbf{j}}|} \sum_{\mathbf{x}\in S_{\mathbf{u},\mathbf{j}}} f(\mathbf{x})$$

The algorithm maintains a histogram of counts corresponding to $(\mathbf{j},\mathbf{u})$ pairs it encounters in the data. It is evident that the histograms can be created in a single scan.

The estimates $\hat{f}_{\mathbf{u},\mathbf{j}}$ and thus $\hat{w}^*_{\mathbf{j}}$ can be computed directly from the histograms. It can be shown that under the assumption that $\mathbf{X}\otimes\neg\mathbf{j}$ is independent of $\mathbf{X}\otimes\mathbf{j}$, bounds on the bias of $\hat{w}^*_{\mathbf{j}}$ are smaller than those on $\hat{w}_{\mathbf{j}}$. Thus, we can give better guarantees about the performance of the new estimator.

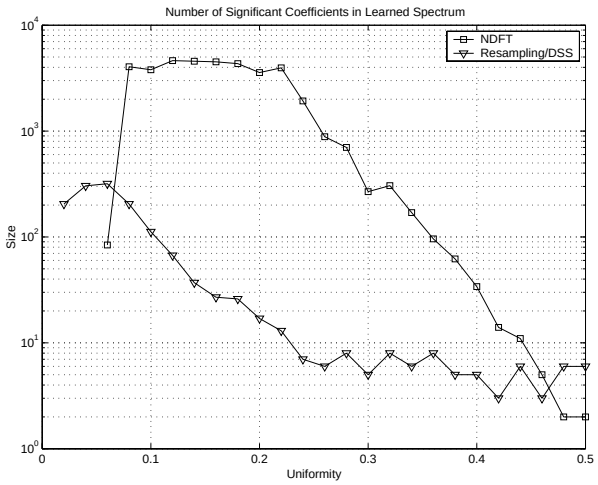

Figure 3: A comparison of the spectrum sizes corresponding to Figure 2.

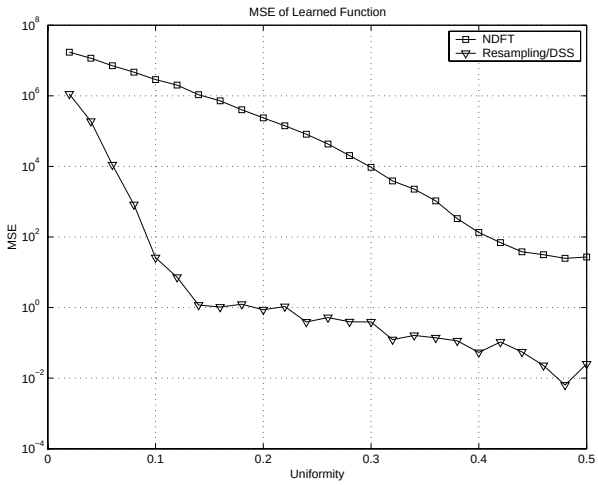

Figure 4: Comparison of the MSE for another second-order function with two nonzero coefficients. The y-axis is log-scaled.

Algorithm: Deterministic Single Scan (DSS)

- For each $\mathbf{x}_i \in S$ do
  - For each $\mathbf{j} \in J$ do
    - $\text{count}[\mathbf{j}, \mathbf{x}_i \otimes \mathbf{j}] \leftarrow \text{count}[\mathbf{j}, \mathbf{x}_i \otimes \mathbf{j}] + 1$

4 Controlled Experiments

This section provides experimental verification of the performance of the proposed algorithm on simulated data. Four datasets were generated using a product distribution with the probability of success of all bits being set at a given value. The instances so generated were labeled using four different functions. The number of features in each of the cases was 100, a fairly large number. Training data size was set to 750 instances. Note that this is extremely sparse in addition to being skewed — the whole space consists of 2^{100} instances. Since the functions have in truth only a small number of nonzero Fourier coefficients, such small amounts of data should be sufficient for estimation. The challenge for the learning algorithm is to ignore the abundant spurious information. In all our test cases, the Resampling/DSS algorithm-based estimates performed significantly better than the MDFT estimates. The performance was measured both in terms of the Mean Squared Error (MSE) over a testing data set and the size of the learned spectrum. The testing data set was generated independently of the training data set according to the same distribution, and had 5500 instances classified using the same function.

Figures 2, 4, 6 and 8 show the results of the experiments. The y-axis in these figures is logarithmic scale. The x-axis represents the probability that a bit in a string is a 1. Thus, a very low x value corresponds to data in which most of the features take the value of 0. The smaller the x value, the more skewed the data. As can be seen, the NWT estimates perform very poorly in the case of highly skewed data. In some of the cases (Figure 2 and Figure 8), the NWT estimates appear to get worse and then improve as the data becomes more uniform. This is an artefact of the MDFT learning algorithm: no learning takes place when the data is very skewed.

The figures show that not only does the Resampling/DSS algorithm perform better than the MDFT in terms of the MSE, but the spectrum learned is also *much* (orders of magnitude) smaller for skewed data. This is important for algorithms that utilize the Fourier spectrum for tasks such as building decision trees and aggregation of an ensemble model [12].

5 Conclusions and Future Work

This paper presented an algorithm that improves upon the accuracy of the standard MDFT estimates in the case of skewed data. Experiments show that the results can have errors that are orders of magnitude smaller. In addition, the size of the spectrum learned falls much more rapidly for the proposed algorithm than for the standard MDFT estimates. This is important for many algorithms that use the MDFT to learn other models such as decision trees. The paper pointed out that better bounds can be given for the estimates computed by the new algorithm than for the

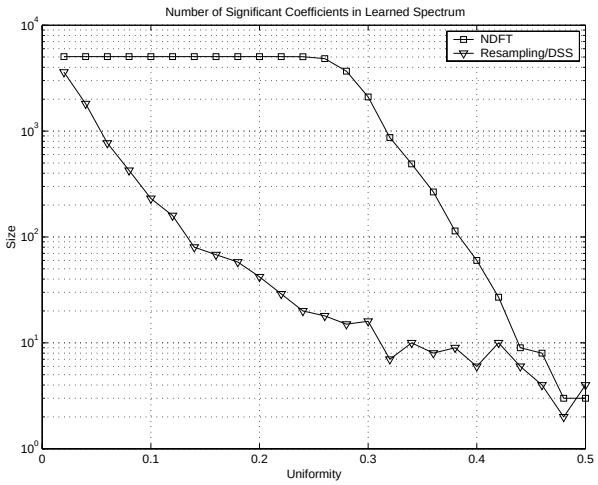

Figure 5: A comparison of the spectrum sizes corresponding to Figure 4.

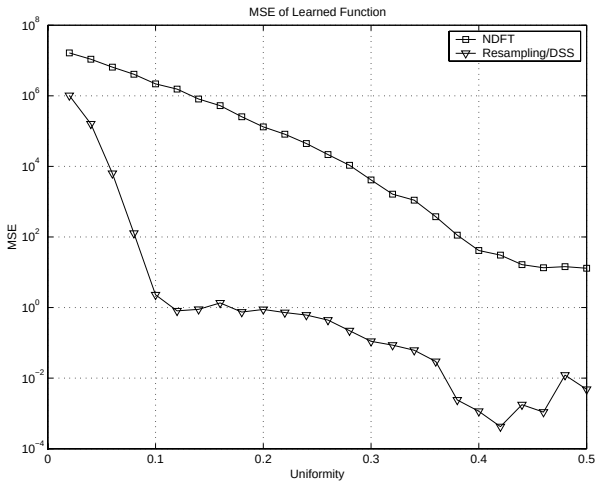

Figure 6: Comparison of the MSE of two learning algorithms for a second-order function with four nonzero coefficients. The y-axis is log-scaled.

MDFT estimates.

The work in this paper is a preliminary effort. The algorithm needs to be studied further under the following headings:

- While the algorithm handles hundreds of attributes and is linear in the sample size, it does not scale to thousands of attributes. The scalability of the algorithm needs to be addressed by the use of an appropriate (heuristic or otherwise) pruning technique.

- Bounds on the bias of the new estimator can be proven to be better than those on MDFT estimates. However, the variance of the estimator is yet to be studied in order to determine the reliability of the new estimates (i.e. have we reduced bias at the expense of variance).

- The ability to learn coefficients accurately from skewed data also enables us to learn effectively in the case of vertically partitioned distributed data. The application of this algorithm to distributed data needs to be investigated.

In summary, this paper presented an interesting, promising new algorithm for mining highly skewed data sets.

6 Acknowledgements

The research in this paper was partially supported by NSF Award IIS-0093353 and NSF Grant IIS-0083946 and from IIS-0196401 (earlier IIS-9803360).

References

[1] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[2] L. Breiman. Arcing classifiers. *Annals of Statistics*, 26:801–824, 1998.

[3] C. Bridges and D. E. Goldberg. The nonuniform walsh-schema transform. In G. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 13–22. Morgan Kaufmann, San Mateo, CA, 1991.

[4] B. Efron. Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 7:1–26, 1979.

[5] B. Efron. The jackknife, the bootstrap and other resampling plans, 1982. Philadelphia: SIAM.

[6] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156, 1996.

[7] M. L. Furst, J. C. Jackson, and S. W. Smith. Improved learning of AC^0 functions. In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, pages 317–325, 1991.

[8] D. E. Goldberg. Genetic algorithms and walsh functions: Part I, a gentle introduction. *Complex Systems*, 3(2):129–152, 1989.

[9] J. Jackson. *The Harmonic Sieve: A Novel Application of Fourier Analysis to Machine Learning Theory and Practice*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1995.

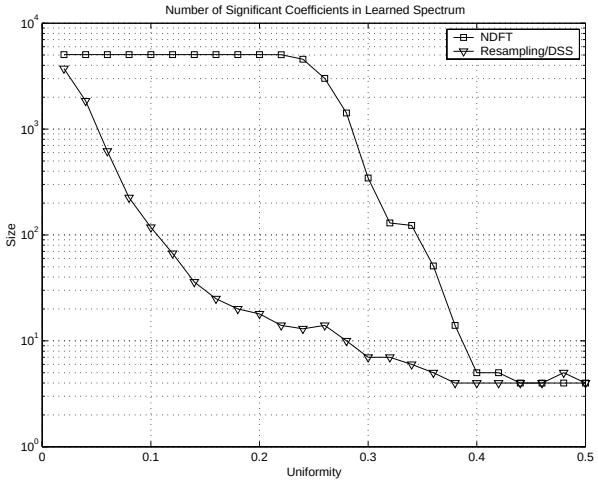

Figure 7: A comparison of the spectrum sizes corresponding to Figure 6.

[10] E. Kushilevitz and Y. Mansour. Learning decision trees using the fourier spectrum. *SIAM Journal of Computing*, 22(6):1331–1348, 1993.

[11] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, fourier transform, and learnability. In *IEEE Symposium on Foundations of Computer Science*, pages 574–579, 1989.

[12] B. Park. *Knowledge Discovery From Heterogeneous Data Streams Using Fourier Spectrum of Decision Trees.* PhD thesis, Washington State University, Pullman, WA, 2001.

[13] B. Park, R. Ayyagari, and H. Kargupta. A fourier analysis based approach to learning decision trees in a distributed environment. In *Procceedings of the first SIAM international conference on data mining*, 2001.

[14] S. Sahar. Learning under the fourier transform algorithm. Master's thesis, Tel Aviv University, 1996.

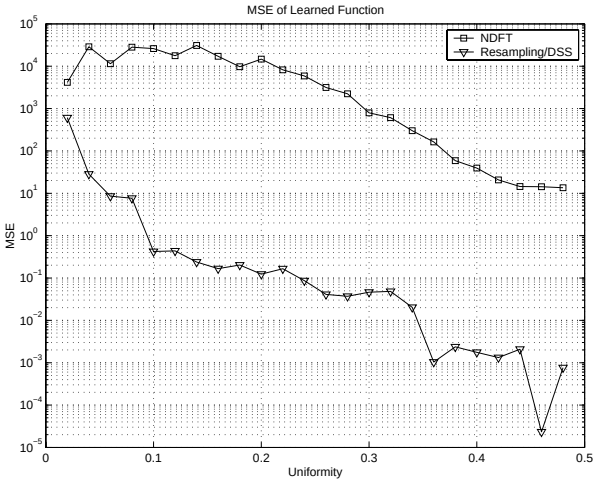

Figure 8: Comparison of the MSE of two learning algorithms for a second-order function with four nonzero coefficients. The y-axis is log-scaled.

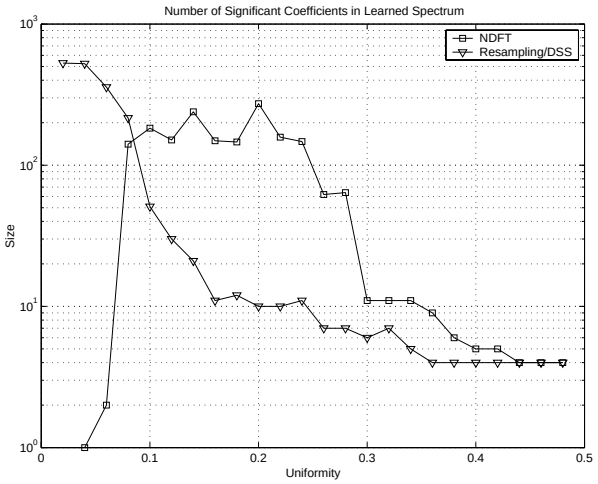

Figure 9: A comparison of the spectrum sizes corresponding to Figure 8.