# Learning Using Group Representations

(Extended Abstract)

Dan Boneh*
dabo@cs.princeton.edu
Department of Computer Science
Princeton University
Princeton, NJ 08544

## Abstract

We consider the problem of learning functions over a fixed distribution. An algorithm by Kushilevitz and Mansour [7] learns any boolean function over $\{0,1\}^n$ in time polynomial in the $L_1$-norm of the Fourier transform of the function.

We show that the KM-algorithm is a special case of a more general class of learning algorithms. This is achieved by extending their ideas using representations of finite groups. We introduce some new classes of functions which can be learned using this generalized KM algorithm.

---

*Supported by NSF grant CCR–8858788

# 1 Introduction

In recent years Fourier analysis has become a valuable tool in learning theory[8, 7, 9, 2, 1, 6]. This approach culminated in a recent result due to Jackson [6] showing that polynomial size DNF formula can be learned in polynomial time using membership queries. In an influential paper, Kushilevitz and Mansour [7] presented an algorithm for learning boolean decision trees over the uniform distribution. In fact, they prove a stronger result showing that any boolean function on $\{0,1\}^n$ can be learned in time polynomial in the $L_1$ norm of its Fourier transform. In this paper we generalize their result and show how the generalization can be used to learn new classes of functions. First we define the learning model.

**Definition 1.1** *Let $\mathcal{F}$ be a a family of boolean functions over some domain $G$. We say that $\mathcal{F}$ is* learnable *if there exists an algorithm $A$ which for every $f \in \mathcal{F}$ and $\epsilon, \delta > 0$ outputs an hypothesis $h = A(f, \epsilon, \delta)$ such that*

$$\Pr[\mathit{error}(f,h) \geq \epsilon] \leq \delta$$

*where $\mathit{error}(f,h) = \Pr[f(x) \neq h(x)]$ and $\Pr$ is the uniform distribution on $G$. The algorithm is polynomial if its running time is polynomial in $n, \frac{1}{\epsilon}, \log\frac{1}{\delta}$.*

Before we explain the results of this paper, we briefly describe the approach taken in [7]. The set of functions $f : \{0,1\}^n \rightarrow \mathbb{C}$ forms a vector space over $\mathbb{C}$. We denote this vector space by $\mathcal{W}_n$. Clearly the dimension of $\mathcal{W}_n$ is $2^n$. It has a natural inner product defined by:

$$(f,g) = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)\overline{g(x)}$$

where $\bar{z}$ denotes the complex conjugate of $z$. As usual, the norm of a function $f \in \mathcal{W}_n$ is $\| f \|^2 = (f,f)$.

The algorithm in [7] learns a function $f$ by learning its projections on an orthonormal basis of $\mathcal{W}_n$. The orthonormal basis used is made up of all parity functions on $n$ variables. Formally, for every $k = (k_1, \ldots, k_n) \in \{0,1\}^n$ define the function

$$\chi_k(x) = (-1)^{(k,x)} \quad \text{where} \quad (k,x) = \sum_{i=1}^{n} x_i k_i.$$

Let $P$ be the set of functions $P = \{\chi_k \mid k \in \{0,1\}^n\}$. The orthonormality of the functions in $P$ can easily be verified. Since there are $2^n$ such functions, the set $P$ forms an orthonormal basis of $\mathcal{W}_n$. Thus, any function $f \in \mathcal{W}_n$ can be uniquely written as $f(x) = \sum_k c_k \chi_k(x)$ for some constants $c_k \in \mathbb{C}$. Define the $L_1$ norm of a function with respect to $P$ as $L_P(f) = \sum_k |c_k|$. The main theorem in [7] is that a function $f \in \mathcal{W}_n$ can be learned in time polynomial in $L_P(f)$.

Note that $L_P(f)$ is dependent on the choice of basis $P$. For some functions $f \in \mathcal{W}_n$ the $L_1$ norm of $f$ with respect to $P$ is exponential in $n$. For such functions, the KM algorithm will fail to learn the function in polynomial time. However, there may exists a different orthonormal basis $B$ of $\mathcal{W}_n$ such that the $L_1$ norm of $f$ with respect to $B$ is 'small'. This raises the natural question of whether the KM algorithm can be generalized to learn functions using bases other than $P$.

In this paper we show that the KM algorithm can be generalized to work over orthonormal basis of $\mathcal{W}_n$ that arise from representations of finite groups. The practical implication is that in order to learn a family $\mathcal{F} \subseteq \mathcal{W}_n$ one has to find an orthonormal basis $B$ of the form discussed in this paper, so that $L_B(f)$ is polynomial for all $f \in \mathcal{F}$.

The underlying method is to impose some group structure $G$ on the set $\{0,1\}^n$. The irreducible linear representations of $G$ induce an orthonormal basis of $\mathcal{W}_n$. We show that for any group $G$, this orthonormal basis can be used to learn $\mathcal{F}$. Exact definitions are given in the next section. The same technique can be used to learn functions over any domain. That is, impose a group structure on the domain and use the orthonormal basis associated with the group to learn boolean functions over the domain.

## 2 Learning functions using an orthonormal basis

In this section we summarize some basic definitions taken from [7] and extend them to arbitrary orthonormal bases. Let $G$ be some finite domain of size $s$. Usually $G$ will be taken to be $\{0,1\}^n$. Let $\mathcal{W}$ be the vector space of all functions $f : G \to \mathbb{C}$. We have already seen that there is a natural inner product defined on $\mathcal{W}$, i.e. $(f,g) = \frac{1}{s}\sum_{x\in G} f(x)\overline{g(x)}$.

Let $B = \{b_1, \ldots, b_s\}$ be an orthonormal basis of $\mathcal{W}$. Any function $f \in \mathcal{W}$ can be uniquely written as

$$f = \sum_{k=1}^{s} c_k b_k \quad \text{where} \quad c_k = (f, b_k).$$

Parseval's identity states that $\| f \|^2 = \sum |c_k|^2$. Define the $L_1$-norm of $f$ with respect to $B$ as $L_B(f) = \sum_{k=1}^{s} |c_k|$.

**Lemma 2.1** *Let $f : G \to \{-1,1\}$, and $t > 0$. Define*

$$g_t(x) = \sum_{|c_i|>t} c_i b_i(x)$$

*and let $h_t(x) = \text{sign}(\text{real}(g_t(x)))$. Then $\Pr[f(x) \neq h_t(x)] \leq tL_B(f)$*

**Proof** Observe that when $f(x) \neq h_t(x)$ we have that $|f(x) - g_t(x)| \geq |f(x) - \text{real}(g_t(x))| \geq 1$. Hence,

$$\Pr[f(x) \neq h_t(x)] \leq \| f - g_t \|^2 = \sum_{|c_k|\leq t} |c_k|^2 \leq t\sum_{k=1}^{s} |c_k| = tL_B(f)$$

□

Setting $t = \epsilon/L_B(f)$ implies that $\Pr[f(x) \neq h_t(x)] \leq \epsilon$. Hence, $h_t(x)$ is a valid output of a learning algorithm applied to $f$. By Parseval's identity, $1 = \| f \|^2 = \sum |c_k|^2$. Therefore, the number of coefficients larger than $\epsilon/L_B(f)$ is at most $(L_B(f)/\epsilon)^2$. When $L_B(f)$ is polynomial in $n = \log s$, the hypothesis description length will be polynomial in $n$ and $\frac{1}{\epsilon}$ and may be generated by a polynomial time algorithm.

The above discussion shows that in order to learn a set functions $\mathcal{F}$ one has to find an orthonormal basis $B$ of $\mathcal{W}$ that satisfies the following two properties:

1. Every $f \in \mathcal{F}$ has a polynomial $L_B(f)$ norm.

2. There exists a polynomial time algorithm that finds the large projections of $f$ on the basis $B$.

A large portion of the work in learning theory related to Fourier analysis concentrates on finding $\mathcal{F}$ that satisfy property 1 for the basis made up of parity functions $P$. In this paper we concentrate on finding bases that satisfy property 2. This will enable us to use these methods for learning functions which are not learnable using the basis $P$.

## 3 Group Representations

In this section we recall some basic definitions from the theory of linear representations of finite groups. For an introduction to this subject see [11]. For a vector space $V$ over $\mathbb{C}$ we let $GL(V)$ denote the group of invertible linear transformations of $V$ under composition. For the remainder of the paper $G$ and $H$ will denote finite groups.

**Definition 3.1** *A* representation *$\rho$ of $G$ is a homomorphism $\rho : G \to GL(V)$ where $V$ is a finite dimensional vector space over $\mathbb{C}$. The dimension of $V$ is called the degree of the representation and is denoted by $d_\rho$.*

**Definition 3.2** *A representation $\rho : G \to GL(V)$ is said to be* irreducible *if there is no non trivial subspace of $V$ which is invariant under $\rho(g)$ for all $g \in G$.*

**Definition 3.3** *Two representations $\rho : G \to GL(V)$ and $\sigma : G \to GL(W)$ are equivalent if there is an isomorphism $f : V \to W$ such that all $g \in G$ satisfy $\rho(g) = f^{-1}\sigma(g)f$.*

Equivalent representations have the same degree and their images are conjugate subgroups of $GL(V)$. Naturally, we will be interested in representations up to equivalence. A finite group admits only a finite number of nonequivalent irreducible representations.

**Theorem 3.1** *Let $\rho_1, \ldots, \rho_c$ be all the irreducible representations (up to equivalence) of a finite group $G$ and let $d_1, \ldots, d_c$ denote their degrees. Then $\sum_{k=1}^{c} d_k^2 = |G|$.*

We note that a group $G$ is Abelian if and only if all its irreducible representations have degree 1. We now give and example of the representations of a non Abelian group.

**Example** The quaternion group $Q$ consists of 8 elements $\{\pm 1, \pm i, \pm j, \pm k\}$ and the group operation satisfies the relations: $-1 = i^2 = j^2 = k^2 = ijk$. This group is not Abelian since $ij = -ji = k$. The irreducible representations of $G$ are:

$$\rho_1 \equiv 1$$
$$\rho_2(\pm 1) = \rho_2(\pm i) = 1 \quad ; \quad \rho_2(\pm j) = \rho_2(\pm k) = -1$$
$$\rho_3(\pm 1) = \rho_3(\pm j) = 1 \quad ; \quad \rho_3(\pm i) = \rho_3(\pm k) = -1$$
$$\rho_4(\pm 1) = \rho_4(\pm k) = 1 \quad ; \quad \rho_4(\pm i) = \rho_4(\pm j) = -1$$
$$\rho_5(\pm 1) = \begin{pmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{pmatrix} \quad ; \quad \rho_5(i) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$
$$\rho_5(j) = \begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix} \quad ; \quad \rho_5(k) = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}$$

The square of the degrees sum to 8. Hence, by theorem 3.1, these are all the irreducible representation of $Q$.

**Theorem 3.2** *Let $\rho : G \rightarrow GL(V)$ be any representation of $G$. There exists a basis of $V$ for which the matrix representing $\rho(g)$ is unitary*[1] *for all $g \in G$. Further more, when $G$ is a p-group*[2] *there exists a basis of $V$ for which the matrix representing $\rho(g)$ is monomial*[3] *for all $g \in G$.*

See [5, p. 77] for a proof of the second part of the theorem. The matrices representing $\rho_5$ above are monomial. From now on, whenever we speak of the matrix of a representation we will assume that some basis satisfying the assertion of Theorem 3.2 has been fixed. Given a representation $\rho$ of $G$, we denote by $\rho_{k,l}(g)$ the element in the $k$'th row and $l$'th column of the unitary matrix representing $\rho(g)$. The next theorem describes the orthonormal basis of $\mathcal{W}$ that will be used by the learning algorithm.

**Theorem 3.3** *Let $\rho, \psi$ be two irreducible representations of $G$ and let $d$ be the degree of $\rho$. Then*

$$(\rho_{k,l}, \psi_{r,s}) \overset{def}{=} \frac{1}{|G|} \sum_{g \in G} \rho_{k,l}(g)\overline{\psi_{r,s}(g)} = \begin{cases} \frac{1}{d} & \textit{if } \rho \equiv \psi, k = r, l = s, \\ 0 & \textit{otherwise.} \end{cases}$$

For an irreducible representations $\rho$ of $G$ and $k, l$ between 1 and the degree of $\rho$ define the function

$$b_{\rho,k,l}(x) \overset{def}{=} \sqrt{d_\rho}\; \rho_{k,l}(x)$$

Let $\mathcal{B}_G \subset \mathcal{W}$ be the set of functions $\{b_{\rho,k,l}(x)\}$ where $\rho$ ranges over all the irreducible representations of $G$ and $k, l$ range between 1 and the degree of $\rho$. Theorem 3.3 tells us that $\mathcal{B}_G$ is an orthonormal set in $\mathcal{W}$. By theorem 3.1, the number of functions in $\mathcal{B}_G$ is $|G|$. Hence, $\mathcal{B}_G$ is an orthonormal basis of $\mathcal{W}$.

**Example** The orthonormal basis generated by the representation of $Q$ is:

$$\begin{array}{cc}
[1,1,1,1,1,1,1,1] & \sqrt{2}\,[1,-1,0,0,0,0,i,-i] \\
[1,1,1,1,-1,-1,-1,-1] & \sqrt{2}\,[0,0,-1,1,-i,i,0,0] \\
[1,1,-1,-1,1,1,-1,-1] & \sqrt{2}\,[0,0,1,-1,-i,i,0,0] \\
[1,1,-1,-1,-1,-1,1,1] & \sqrt{2}\,[1,-1,0,0,0,0,-i,i]
\end{array}$$

where each vector is a function $b(x)$ whose values are $[b(1), b(-1), b(i), b(-i), b(j), b(-j), b(k), b(-k)]$.

An irreducible representation $\rho$ of $G$ of degree $d_\rho$ gives rise to $d_\rho^2$ orthonormal functions. Given a function $f : G \rightarrow \mathbb{C}$ it is convenient to write the projection of $f$ on the orthonormal functions generated by $\rho$ in a $d_\rho \times d_\rho$ matrix denoted by $\hat{f}_\rho$. Formally, the *coefficient matrix* is defined by:

$$\hat{f}_\rho = \frac{\sqrt{d_\rho}}{|G|} \sum_{x \in G} f(x)(\rho(x))^*$$

where $(\rho(x))$ is the unitary matrix representing $\rho(x)$. Using this notation, the following theorem is a simple consequence of Theorem 3.3.

---

[1] A matrix $P$ is unitary if $PP^* = I$ where $P^* = \overline{P}^t$.

[2] A $p$-group is a group whose order is a power of $p$.

[3] A monomial matrix is a unitary matrix in which all the elements are either 0 or a root of unity. Groups that can be represented using monomial matrices are called $\mathcal{M}$ groups. It can be shown that any nilpotent group is an $\mathcal{M}$ group (a group is called nilpotent if it is a direct product of $p$-groups). Specifically, any group of size $2^n$ is an $\mathcal{M}$ group.

**Theorem 3.4 (Fourier Inversion)** *Any function $f : G \to \mathbb{C}$ satisfies*

$$f(x) = \sum_{\rho} \sqrt{d_\rho} \quad \mathrm{Tr}(\hat{f}_\rho \cdot (\rho(x)))$$

*where the sum is taken over all the irreducible representations of $G$.*

For a function $f : G \to \mathbb{C}$ we denote by $L_G(f)$ the $L_1$-norm of $f$ with respect to $\mathcal{B}_G$, i.e. $L_G(f) = \sum_\rho \sum_{k,l} |(\hat{f}_\rho)_{k,l}|$. We now turn our attention to determining the irreducible representations of a direct product of two groups.

**Theorem 3.5** *Let $T = G \times H$. Any element $t \in T$ can be uniquely written as $g_t h_t$ where $g_t \in G$ and $H_t \in H$. Let $\rho_1, \ldots, \rho_p$ be the irreducible representations of $G$ and $\sigma_1, \ldots, \sigma_q$ be the irreducible representations of $H$. Then $T$ has exactly $pq$ irreducible representations given by*[4]

$$\tau_{k,l}(t) = \rho_k(g_t) \otimes \sigma_l(h_t) \quad \textit{for} \quad 1 \leq k \leq p\ ,\ 1 \leq l \leq q$$

The above theorem enables us to easily determine the orthonormal basis associated with groups which are a direct product of small groups. For instance, an embedding of the group $Q^n$($n$ copies of the quaternion group) in the boolean cube $\{0,1\}^{3n}$ gives rise to an orthonormal basis of the space of functions $\{0,1\}^{3n} \to \mathbb{C}$. Furthermore, the functions which make up the basis are easy to evaluate.

Let $\rho$ be an irreducible representation of a group $G$. By Theorem 3.3 the function $\rho_{k,l}(x)$ has norm $1/d_\rho$. To normalize it, the function has to be multiplied by $\sqrt{d_\rho}$. When the dimension is large this normalization could make the function attain large values. The following simple lemma shows that when learning a boolean function, the basis functions associated with representations of large degrees can be ignored.

**Lemma 3.1** *Let $G$ be a $p$-group and $f$ a boolean function $f : G \to \{1, -1\}$. Let $\rho$ be a representation of $G$. The norm of all entries of $\hat{f}_\rho$ is less than $\frac{1}{\sqrt{d_\rho}}$.*

**Proof** Using the fact that the matrices representing $\rho$ are monomial and Theorem 3.3 we get:

$$\left|(\hat{f}_\rho)_{l,k}\right| = \frac{\sqrt{d_\rho}}{|G|} \left| \sum_{x \in G} f(x)\overline{\rho_{k,l}(x)} \right| \leq \frac{\sqrt{d_\rho}}{|G|} \sum_{x \in G} |\rho_{k,l}(x)| = \frac{\sqrt{d_\rho}}{|G|} \sum_{x \in G} |\rho_{k,l}(x)|^2 = \frac{\sqrt{d_\rho}}{d_\rho} = \frac{1}{\sqrt{d_\rho}}$$

□

## 4 Learning Using Group Representations

Let $G$ be a group on $\{0,1\}^n$ and let $\mathcal{B}_G$ be the orthonormal basis of $\mathcal{W}$ induced by $G$. In section 2 we saw that to learn a function $f : \{0,1\}^n \to \{0,1\}$ it suffices to find the coefficients of $f$ on $\mathcal{B}_G$ of norm bigger than some given $\beta > 0$. In this section we describe an algorithm for finding all such coefficients of $f$. The algorithm is randomized and runs in time polynomial in $n$, $\frac{1}{\beta}$, $\log\frac{1}{\delta}$ where $\delta$ is the probability of failure.

---

[4] For $A : V \to V$ and $B : W \to W$ the tensor product $A \otimes B : V \otimes W \to V \otimes W$ is the operator sending a basis element $(v \otimes w)$ to $(Av \otimes Bw)$. Technically, if $A$ is an $n \times n$ matrix and $B = (b_{i,j})$ is an $m \times m$ matrix, then $A \otimes B$ is an $nm \times nm$ matrix made up of $m \times m$ blocks where the $(i,j)$'th block is $b_{i,j}A$.

For the rest of the paper, whenever we speak of a representation it is assumed to be irreducible. To learn a function $f : \{0,1\}^n \to \{0,1\}$ we perform the following steps:

1. Embed a group $G$ of size $2^n$ in $\{0,1\}^n$. This is simply to say that we define a group operation on the set $\{0,1\}^n$.
2. Find all representations $\rho$ of $G$ such that

$$\| \hat{f}_\rho \|^2 = \operatorname{Tr}(\hat{f}_\rho \hat{f}_\rho^*) = \sum_{k,l} |(\hat{f}_\rho)_{k,l}|^2 > \beta^2$$

3. For each representation $\rho$ generated in step 2, find all the elements of the matrix $\hat{f}_\rho$ whose norm is bigger than $\beta$.

The algorithm works for any group $G$ of size $2^n$. However, the mathematical theory required to prove its correctness is quite involved and will be presented in the full version of the paper. In this abstract, we restrict our attention to the simpler case where $G$ is a direct product of "small" groups. Formally, we will use groups that satisfy the following definition.

**Definition 4.1** *A family of groups $\{G_k\}_{k\geq 1}$ is said to be a* product *group family if $G_k = H^k$ for all $k$ and some fixed p-group $H$. A member of a product group family will be called a product group.*

Thus, a produce group is a direct product of a number of copies of some group $H$. A product group $G$ admits a normal series of subgroups

$$\{1\} = N_0 \lhd N_1 \lhd \ldots \lhd N_t = G$$

where $N_j = H^j$. Observe that for each $0 < j < t$ we have $G = G/N_j \times N_j$. We call this normal series a *product tower* of $G$. Note that $|N_{j+1}/N_j| = |H| = O(1)$.

The product tower of $G$ plays an important role in the algorithm. It is worth noting that the algorithm works for any $p$-group $G$ by using a principle composition series[5] of the group instead of a product tower. This requires an application of Clifford theory [3] and will not be presented in this abstract.

First we describe how to implement step 3 of the algorithm. Given a representation $\rho$ we wish to find all the entries $(k,l)$ of the matrix $\hat{f}_\rho$ such that $|(\hat{f}_\rho)_{k,l}| > \beta$. By Lemma 3.1 the representations of $G$ of degree greater than $(1/\beta)^2$ need not be considered. Hence, step 3 of the algorithm may be carried out by testing the norm of $(\hat{f}_\rho)_{k,l}$ for all $1 \leq k,l \leq d_\rho$. To approximate the value of

$$(\hat{f}_\rho)_{k,l} = \mathrm{E}[\sqrt{d_\rho}\ f(x)\ \overline{\rho_{l,k}(x)}]$$

for a given $k,l$ we use random sampling. This is explained in more detail in section 4.1 .

When the group $G$ is Abelian all the irreducible representations have degree 1. In fact, this is a characterization of Abelian groups. It follows that when $G$ is Abelian, step 3 of the algorithm becomes trivial.

---

[5] A principle composition series is a series of subgroups $N_0 < N_1 < \ldots < N_k = G$ where each $N_i$ is normal in $G$ and $N_{i+1}/N_i$ is simple. Every $p$-group, i.e. group whose order is a power of $p$, has a principle composition series.

### 4.1 Finding Representations $\rho$ with $\| \hat{f}_\rho \| > \beta$

We now describe how to implement step 2 of the algorithm. Here is quick a overview: we perform a kind of binary search on the representations of $G$. We partition the representations into a small number of classes. We then determine which classes contain large coefficients, and repeat the search in those classes. The process terminates when all classes contain only one representation.

We use the product tower of $G$ to partition the representations of $G$ into classes. By Theorem 3.5 we know that all representations of $G$ are of the form $\sigma \otimes \tau$ where $\sigma, \tau$ are representations of $N_j$ and $G/N_j$ respectively. For a representation $\sigma$ of $N_j$, define the following function on $G$

$$f_\sigma^{(j)}(x) = \sum_{\rho=\sigma\otimes\tau} \sqrt{d_\rho}\ \mathrm{Tr}(\hat{f}_\rho \rho(x))$$

where the sum is over all representations $\tau$ of $G/N_j$. By Theorem 3.4 it should be clear that $f \equiv \sum_\sigma f_\sigma^{(j)}$ since every representation of $G$ is counted exactly once in the sum. By Parseval's identity,

$$\| f_\sigma^{(j)} \|^2 = \sum_{\rho=\sigma\otimes\tau} \| \hat{f}_\rho \|^2$$

Thus, if some representation $\rho = \sigma \otimes \tau$ of $G$ satisfies $\| \hat{f}_\rho \| > \beta$ then certainly $\| f_\sigma^{(j)} \| > \beta$. This leads to the following algorithm which is similar to the KM-algorithm.

**Algorithm** COEF( $j, \sigma$ )
Input: A representation $\sigma$ of $N_j$.
Output: All representations $\rho = \sigma \otimes \tau$ such that $\| \hat{f}_\rho \| > \beta$.

```
1. If $j = t$, where $t$ is the length of the product tower of $G$, then $\sigma$ is a representation of $N_t = G$.
   Output $\sigma$ and stop.
2. For each representation $\nu$ of $H$ do:
   a. Define $\sigma' = \sigma \otimes \nu$ to be a representation of $N_{j+1}$.
   b. If $\| f_{\sigma'}^{(j+1)} \| > \beta$, then call COEF( $j+1, \sigma'$ ).
```

Calling COEF(0,$T$) where $T$ is the trivial representation of $N_0 = \{1\}$ will clearly only produce representations $\rho$ of $G$ for which $\| \hat{f}_\rho \| > \beta$. By the discussion above, all such representations will be generated.

By Parseval's identity, for any $N_j$ we have $\sum_\sigma \| f_\sigma^{(j)} \|^2 = \| f \|^2 = 1$. Hence, for any $N_j$ at most $1/\beta^2$ of the $f_\sigma^{(j)}$ can satisfy $\| f_\sigma^{(j)} \| > \beta$. Thus, at any level of the recursion at most $1/\beta^2$ functions need be checked.

Thus the algorithm will run in polynomial time in $1/\beta$ and will output the correct values. The problem that remains is performing the test in step 2b. Given $j$ and $\sigma$ the following theorem enables us to quickly approximate $\| f_\sigma^{(j)} \|^2$.

**Theorem 4.1**

$$\| f_\sigma^{(j)} \|^2 = \mathop{\mathrm{E}}_{x\in G, y\in N_j, z\in N_j} \left[ d_\sigma^2 f(xy) f(xz)\, \overline{\mathrm{Tr}(\sigma(y))}\ \mathrm{Tr}(\sigma(z)) \right]$$

Before proving the theorem we first discuss how it can be used to evaluate $\| f_\sigma^{(j)} \|^2$. For $(x,y,z) \in G \times N_j \times N_j$ define

$$\alpha(x,y,z) = d_\sigma^2 f(xy)\overline{f(xz)\ \mathrm{Tr}(\sigma(y))}\ \mathrm{Tr}(\sigma(z))$$

By Lemma 3.1 the largest degree that has to be considered is $(1/\beta)^2$. Hence, the norm of the values attained by $\alpha(x,y,z)$ are bounded by $(1/\beta)^4$. To approximate $\| f_\sigma^{(i)} \|^2$ we sample $\alpha(x,y,z)$ at many uniformly and independently chosen points $(x_1,y_1,z_1),\dots,(x_m,y_m,z_m) \in G \times N_j \times N_j$ and then approximate $\| f_\sigma^{(j)} \|^2$ by $\frac{1}{m}\sum_k \alpha(x_k,y_k,z_k)$. To bound the error we use the standard Chernoff bounds(see [10, p. 240]). The number of sample points, $m$, required to make the error less than $\beta/2$ with probability $\delta$ comes out to be $O({(1/\beta)}^{10}\log\frac{1}{\delta})$. A similar method can be used to approximate the value of the entries in $\hat{f}_\rho$ for a representation $\rho$ of $G$.

Theorem 4.1 is crucial for the algorithm to work. The fact that such a theorem exists when the basis is associated with a group is the reason why the group structure is essential. No such theorem holds in general for arbitrary orthonormal bases. The proof of Theorem 4.1 requires the following two lemmas.

**Lemma 4.1** *Let $G = N \times (G/N)$ where $N \lhd G$. Consider a representation $\rho = \sigma' \otimes \tau'$ of $G$ and a representation $\sigma$ of $N$, where $\sigma'$ is a representation of $N$ and $\tau'$ representation of $G/N$. Then for any $d_\rho \times d_\rho$ matrix $A$:*

$$B \overset{def}{=} \frac{1}{|N|}\sum_{x\in N} \mathrm{Tr}(A\rho(x))\ \mathrm{Tr}(\sigma(x)^*) = \begin{cases} 0 & \textit{if } \sigma' \neq \sigma \\ \frac{1}{d_\sigma}\ \mathrm{Tr}(A) & \textit{if } \sigma' = \sigma \end{cases}$$

**Proof** Let $E_{k,l}$ be the $d_\rho \times d_\rho$ matrix whose $(k,l)$'th element is 1 and the rest are 0. We first prove the lemma when $A = E_{k,l}$ for some $(k,l)$. Observe that $\mathrm{Tr}(E_{k,l}\rho(x)) = \rho_{l,k}(x)$. For all $x \in N$ we have $\rho(x) = \sigma' \otimes I_{d_{\tau'}}$. Hence, the function $(\rho_{l,k}(x))_{x\in N}$ is either identically 0 or a function $\sigma'_{u,v}(x)$ for some $u, v$.

Assume $\sigma \neq \sigma'$. Clearly, if $\rho_{l,k}(x)$ is identically 0 then $B = 0$. Otherwise, by the orthogonality relations for the representations of the group $N$ we get

$$B = \Big(\ \mathrm{Tr}(E_{k,l}\rho(x)),\ \mathrm{Tr}(\sigma(x)^*)\Big) = \Big(\rho_{l,k}(x),\ \mathrm{Tr}(\sigma(x)^*)\Big) = \Big(\sigma'_{u,v}(x),\ \mathrm{Tr}(\sigma(x)^*)\Big) = 0$$

Similarly, when $\sigma = \sigma'$ and $k \neq l$ we find that $B = 0$. Since $\mathrm{Tr}(E_{k,l}) = 0$ when $k \neq l$, the lemma is correct in this case as well.

When $\sigma = \sigma'$ and $k = l$ we need to show that $B = \frac{1}{d_\sigma}$. This follows from the orthogonality relations (Theorem 3.3) since for $x \in N$, $\rho_{k,k}(x) = \sigma_{u,u}(x)$ and $(\sigma_{u,u}(x),\ \mathrm{Tr}(\sigma(x)^*)) = \frac{1}{d_\sigma}$.

This proves the lemma for the elementary matrices $E_{k,l}$. Observe that if the theorem is true for matrices $X, Y$ then it is true for any linear combination of $X$ and $Y$. Since any $d_\rho \times d_\rho$ matrix is a linear combination of the elementary matrices, the lemma is true for all matrices. □

Using this lemma we can now show how to evaluate $f_\sigma^{(j)}(x)$ for a given $x \in G$.

**Lemma 4.2**

$$\frac{1}{d_\sigma} f_\sigma^{(j)}(x) = \frac{1}{|N_j|}\sum_{y\in N_j} f(xy)\ \mathrm{Tr}(\sigma(y)^*) = \mathop{\mathrm{E}}_{y\in N_j}[f(xy)\ \mathrm{Tr}(\sigma(y)^*)]$$

**Proof** Expanding the right hand side by using the inversion formula (Theorem 3.4) gives

$$\frac{1}{|N_j|}\sum_{y\in N_j} f(xy)\ \mathrm{Tr}(\sigma(y)^*) = \frac{1}{|N_j|}\sum_{y\in N_j}\sum_{\rho}\sqrt{d_\rho}\ \mathrm{Tr}(\hat{f}_\rho\rho(xy))\ \mathrm{Tr}(\sigma(y)^*) =$$

$$\sum_{\rho}\sqrt{d_\rho}\frac{1}{|N_j|}\sum_{y\in N_j}\mathrm{Tr}(\hat{f}_\rho\rho(x)\rho(y))\ \mathrm{Tr}(\sigma(y)^*) = \sum_{\rho=\sigma\otimes\tau}\frac{\sqrt{d_\rho}}{d_\sigma}\ \mathrm{Tr}(\hat{f}_\rho\rho(x)) = \frac{1}{d_\sigma}f_\sigma^{(j)}(x)$$

The third equality follows from Lemma 4.1 by setting $A = \hat{f}_\rho\rho(x)$. □

**Proof of Theorem 4.1:**

$$\begin{aligned}\|\ f_\sigma^{(j)}\ \|^2 &= \mathop{\mathrm{E}}_{x\in G}\left[|f_\sigma^{(j)}(x)|^2\right] = \mathop{\mathrm{E}}_{x\in G}\left[|d_\sigma \mathop{\mathrm{E}}_{y\in N_j}[f(xy)\ \mathrm{Tr}(\sigma^*(y))]|^2\right] = \\ &= \mathop{\mathrm{E}}_{x\in G}\left[d_\sigma^2 \mathop{\mathrm{E}}_{y\in N_j}[f(xy)\overline{\mathrm{Tr}(\sigma(y))}]\mathop{\mathrm{E}}_{z\in N_j}[f(xz)\ \mathrm{Tr}(\sigma(z))]\right] = \\ &= \mathop{\mathrm{E}}_{x\in G, y\in N_j, z\in N_j}\left[d_\sigma^2 f(xy)f(xz)\overline{\mathrm{Tr}(\sigma(y))}\ \mathrm{Tr}(\sigma(z))\right]\end{aligned}$$

□

The algorithm requires the ability to randomly sample elements of $N_j$. Therefore, the embedding of $G$ in the boolean cube must be a "natural" one in the sense that this sampling be feasible. When $G = H^n$ for $H$ some fixed group of size $2^k$ one can assign distinct sets of $k$ variables to encode the copies of $H$.

## 5 Applications

In this section we discuss some applications for the learning algorithm. We begin by generalizing the results in [7] for learning decision trees, to decision trees with arbitrary group operations.

**Definition 5.1** *Let $G$ be some group. A G-Decision Tree, denoted $G$–$DT$, is a tree where the leaves are labeled $\{1,-1\}$ and the internal nodes are labeled with normal subgroups of $G$. A node $v$ labeled with a subgroup $N_v \lhd G$ has $|G/N_v|$ outgoing edges, each labeled with a different coset of $N_v$. A $G$–$DT$ defines a function $f : G \to \{1,-1\}$ by following a path from the root down to the leaves according to which coset the input is in.*

To illustrate this we show that every standard decision tree may be viewed as a $\mathbb{Z}_2^n$–DT. Let $T$ be a decision tree, i.e. each internal node $v$ of $T$ is labeled with some variable $x_v$. An input $x \in \{0,1\}^n$ will branch left at node $v$ if $x_v = 0$ and right otherwise. To view $T$ as a $\mathbb{Z}_2^n$–DT label each internal node $v$ with the following subgroup $N_v$:

$$N_v = \{x \in \mathbb{Z}_2^n \ \mid\ x_v = 0\}$$

where $x_v$ is the label of node $v$ in the tree $T$. The group $N_v$ is a normal subgroup of $\mathbb{Z}_2^n$ of index 2. The coset of $N_v$ is the set of elements in $\mathbb{Z}_2^n$ having a 1 at coordinate $x_v$. Thus, in this resulting $\mathbb{Z}_2^n$–DT, input $x$ will branch left if $x$ is in $N_v$ and right if $x$ is in the coset of $N_v$. This is clearly equivalent to the original tree $T$.

The advantage of group decision trees over standard ones is that they can be defined over any domain. Furthermore, a function might only have very large $\mathbb{Z}_2^n$–DTs, but may have a small $G$–DT for a different 2-group $G$ embedded in the boolean cube. The following theorem shows that we can learn any $G$–DT in time proportional to its size.

**Theorem 5.1** *A function $f : G \to \{1, -1\}$ computed by a $G-DT$ with $m$ leaves satisfies $L_G(f) \leq m$.*

The proof of this theorem is given in the appendix. This result generalizes the result in [7] to trees with arbitrary group operations. It would be interesting to determine whether the results in [2] regarding decision trees generalize to these settings.

We mention a few more applications of these methods. The results for weak learning of polynomial size DNF reported in [1] may be extended to arbitrary groups of the form $G = H^n$ for some fixed group $H$. The implication is that it is possible to generate many different weak approximations for DNF formulae by varying the group $H$. Of course, all these different approximations are correlated.

Another application comes up when implementing the KM-algorithm. To learn a given function $f$, one can run the algorithm using various groups and then pick the output that best approximates $f$. Clearly the most useful group $G$ will be the one which has the lowest $L_G(f)$. Optionally, instead of picking the best approximation, one can use boosting methods similar to the one used in [6].

## 6 Conclusions

The main result of this paper is to show that the KM algorithm is not restricted to using the orthonormal basis of parity functions. We showed that it can use many different orthonormal bases. This is done by varying the underlying group structure acting on the boolean cube. To illustrate this extra degree of freedom we note that there are 2,328 groups of order 128. Any one of these groups can be used to define an orthonormal basis of the space of function on $\{0,1\}^{7n}$. The learning algorithm can use any one of these bases.

Thus, our results show that for some problems the learning problem is reduced to finding a group for which the target concept class has low norm. Furthermore, the generalized KM-algorithm presented in this paper can be made to work over any finite domain by defining a group structure on the domain.

It is worth noting that the generation of arbitrary 2-groups of a given size is quite easy. It can be carried out by picking an arbitrary presentation of a 2-groups. Once the group is determined it is possible to compute its representations using an algorithm due to Dixon [4].

## Acknowledgments

I wish to thank Richard Lipton for helpful discussions on this subject. I am also grateful to Herve Brönnimann for his help in reviewing the manuscript.

## References

[1] Blum A., Furst M., Jackson J., Kearns M., Mansour Y., and Rudich S. Weakly learning DNF

and characterizing statistical query learning using fourier analysis. In *Proc. 26th Annu. ACM Sympos. Theory Comput.* ACM Press, New York, NY, 1994.

[2] M. Bellare. A technique for upper bounding the spectral norm with applications to learning. In *Proc. 5th Annu. Workshop on Comput. Learning Theory*, pages 62–70. ACM Press, New York, NY, 1992.

[3] A.H. Clifford. Representations induced in an invariant subgroup. *Ann. of Math.*, 38(2):533–550, 1937.

[4] J. Dixon. Computing irreducible representations of groups. *Math. Comp.*, 24:707–712, 1970.

[5] L. Dornhoff. *Group Representation Theory, Part A.* Mercel Dekker, New York, 1971.

[6] J. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. In *Proc. of the 35th Symposium on the Foundations of Comp. Sci.*, pages 42–53. IEEE Computer Society Press, 1994.

[7] E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. In *Proc. of the 23rd Symposium on Theory of Computing*, pages 455–464. ACM Press, New York, NY, 1991.

[8] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform, and learnability. In *Proc. of the 31st Symposium on the Foundations of Comp. Sci.*, pages 574–579. IEEE Computer Society Press, 1989.

[9] Y. Mansour. An $o(n^{\log\log n})$ learning algorithm for DNF under the uniform distribution. In *Proc. 5th Annu. Workshop on Comput. Learning Theory*, pages 53–59. ACM Press, New York, NY, 1992.

[10] J. Spencer N. Alon. *The probabilistic method.* Wiley-Interscience, New York, 1992.

[11] J.P. Serre. *Linear representations of finite groups.* Springer-Verlag, Berlin, Heidelberg, New York, 1977.

# A Proof of Theorem 5.1

Let $f : G \to \{1, -1\}$ be computed by a $G$-decision tree $T$. Let leaf$(T)$ be the set of leaves of $T$.

For a matrix $A$ let $|A| = \sum_{i,j} |(A)_{i,j}|$. For a leaf $v$ define inp$(v)$ to be the set of $x \in G$ that reach leaf $v$.

$$L_G(f) = \sum_\rho |\hat{f}_\rho| = \sum_\rho \left| \frac{\sqrt{d_\rho}}{|G|} \sum_{x \in G} f(x)(\rho(x))^* \right| \leq \frac{1}{|G|} \sum_{v \in \text{leaf}(T)} \sum_\rho \sqrt{d_\rho} \left| \sum_{x \in \text{inp}(v)} \rho(x) \right|$$

Let $v_1, \ldots, v_d$ be the nodes on the path to the leaf $v$. Consider the group $N = N_{v_1} \cap \ldots \cap N_{v_d} \lhd G$. It is not difficult to see that inp$(v)$ must be a coset of N. A simple application of Schur's lemma shows that since $N$ is normal in $G$

$$\sum_{x \in \text{inp}(v)} \rho(x) \neq 0 \iff \rho|_N \equiv I$$

Since the matrices representing $\rho(x)$ are monomial, in the case where $\rho|_N \equiv I$ we have for any $a \in \text{inp}(v)$:

$$\left| \sum_{x \in \text{inp}(v)} \rho(x) \right| = |N||\rho(a)| = |N| d_\rho$$

When $\rho|_N \not\equiv I$ the above sum is 0. The representations of $G$ satisfying $\rho|_N \equiv I$ can in fact be regarded as representations of $G/N$, since they are constant on cosets of $N$. Therefore,

$$\sum_\rho \sqrt{d_\rho} \left| \sum_{x \in \text{inp}(v)} \rho(x) \right| = \sum_{\rho|_N \equiv I} \sqrt{d_\rho}\, |N| d_\rho \leq |N| \sum_{\rho|_N \equiv I} d_\rho^2 = |N||G/N| = |G|$$

Hence,

$$L_G(f) \leq \frac{1}{|G|} \sum_{v \in \text{leaf}(T)} |G| = |\text{leaf}(T)|$$

□

# B Learning over Arbitrary Groups

The results in this paper were restricted to learning over product groups. We briefly sketch how the methods can be extended to any $p$-group, and in fact to any finite nilpotent group.

Recall that in the case of a product group, the product tower was used to partition the representations into equivalence classes. In the case of $p$-groups we use a principal composition series of the group.

**Theorem B.1** *Let $G$ be a $p$-group. $G$ has a principle composition series*

$$\{1\} = N_0 < N_1 < \ldots < N_t = G$$

*where for each $0 < j \leq t$ we have $N_j \lhd G$ and $N_j / N_{j-1} \cong \mathbb{Z}_p$.*

**Proof** Any $p$-group is nilpotent and hence has a central series. The central series can be refined to satisfy the conditions of the theorem. □

We use the principle composition series of $G$ to define functions $f_\sigma^{(j)}$ as we did in Section 4.1. We will then show how to generalize Theorem 4.1 which will enable us to determine the norm of $f_\sigma^{(j)}$. First we present some elementary results from Clifford theory. See [3, 5] for a more thorough presentation.

Let $G$ be a finite group and $N$ a normal subgroup of $G$. Two irreducible representations $\tau, \sigma$ of $N$ are called *$G$-conjugate* if there exists $g \in G$ such that for all $x \in N$, $\sigma(x) = \tau(g^{-1}xg)$. Note that all $G$-conjugates have the same degree.

Let $\rho$ be an irreducible representation of $G$. Let $\rho'$ be the restriction of $\rho$ to $N$. The representation $\rho'$ of $N$ will decompose into irreducible representations of $N$.

**Theorem B.2** *Let $\sigma$ be an irreducible representation of $N$ which is a constituent of $\rho'$. Let $m$ be the multiplicity of $\sigma$ in $\rho'$, i.e. the number of irreducible constituents of $\rho'$ equivalent to $\sigma$. Then all the irreducible constituents of $\rho'$ are $G$-conjugate to $\sigma$ and they all have the same multiplicity $m$.*

The value of $m$ is called the *$N$-multiplicity* of $\rho$ and is denoted by $m_\rho$. An irreducible representation $\rho$ of $G$ is said to *contain* an irreducible representation $\sigma$ of $N$ if $\sigma$ is an irreducible constituent of $\rho$ restricted to $N$. Two irreducible representations of $G$ are called *$N$-associates* if when restricted to $N$ they have an irreducible constituent in common. Note that associates can only differ in the multiplicity $m$.

Let $N_j$ be a group from the principal composition series of $G$. For an irreducible representation $\sigma$ of $N_j$ define the function:

$$f_\sigma^{(j)} = \sum_\rho \sqrt{d_\rho}\ \operatorname{Tr}(\hat{f}_\rho \rho(x))$$

where the sum is taken over all $\rho$ that contain $\sigma$. Observe that all such $\rho$ are associates of one another.

**Theorem B.3** *Let $\rho$ be an irreducible representation of $G$ containing $\sigma$ with $N_j$-multiplicity $m$. Then*

$$\| f_\sigma^{(j)} \|^2 = \mathop{\mathrm{E}}_{x\in G, y\in N_j, z\in N_j} \left[ \left(\frac{d_\sigma}{m}\right)^2 f(xy)\overline{f(xz)}\ \overline{\operatorname{Tr}(\rho(y))}\ \operatorname{Tr}(\rho(z)) \right]$$

The proof is similar to the proof of Theorem 4.1. We only need to generalize Lemma 4.1 in the following way:

**Lemma B.1** *Let $G$ be a finite group and $N$ a normal subgroup of $G$. Let $\rho, \rho'$ be irreducible representations of $G$. Let $\sigma$ be an irreducible representation of $N$ contained in $\rho'$, and $m_{\rho'}$ be the $N$-multiplicity of $\rho'$. Then for any $d_\rho \times d_\rho$ matrix $A$:*

$$B \overset{def}{=} \frac{1}{|N|} \sum_{x \in N} \operatorname{Tr}(A\rho(x))\ \operatorname{Tr}(\rho'(x)^*) = \begin{cases} \frac{m_{\rho'}}{d_\sigma}\ \operatorname{Tr}(A) & \textit{if } \rho, \rho' \textit{ are } N\textit{-associates,} \\ 0 & \textit{otherwise.} \end{cases}$$

**Proof Sketch** As in the proof of Lemma 4.1 it is suffices to prove the lemma when $A$ is an elementary matrix $E_{k,l}$. By Theorem B.2 we know that $\rho$ decomposes over $N$ into conjugate representations

$\sigma^1, \ldots, \sigma^r$ of $N$. Similarly, $\rho'$ decomposes into $\tau^1, \ldots, \tau^{r'}$. Observe that $\mathrm{Tr}(E_{k,l}\ \rho(x)) = \rho_{l,k}(x)$. Hence, $\mathrm{Tr}(E_{k,l}\ \rho(x))_{x \in N}$ is either identically 0 or is $\sigma^w_{u,v}(x)$ for some $u, v, w$.

If $\rho, \rho'$ are not $N$-associates then $\sigma^1, \ldots, \sigma^r$ and $\tau^1, \ldots, \tau^{r'}$ are distinct sets of representations of $N$. Hence, $\sigma^w_{u,v}(x)$ is orthogonal to $\mathrm{Tr}(\tau^s(x)^*)$ implying that $B = 0$.

Assume $\rho, \rho'$ are $N$-associates then $\rho, \rho'$ decompose over $N$ into the same set of representations $\sigma^1, \ldots, \sigma^r$ of $N$. Then when $k = l$

$$B = (\rho_{k,k}(x),\ \mathrm{Tr}(\rho'(x))) = m_{\rho'} \sum_{j=1}^{r} (\sigma^w_{u,u}(x),\ \mathrm{Tr}(\sigma^j(x))) = \frac{m_{\rho'}}{d_{\sigma^w}} = \frac{m_{\rho'}}{d_\sigma}$$

□

In order to use Theorem B.3 we must know for a given representation $\rho$ of $G$ the values of $m_\rho$ and $d_\sigma$. They can be computed without actually decomposing $\rho$ over $N_j$.

Now that we have defined the functions $f^{(j)}_\sigma$ and showed how to approximate their norm, the rest of the algorithm remains unchanged.

The big advantage of using product groups is that the representations are easy to evaluate. When using arbitrary groups, one must have enough information about the group to be able to evaluate the representations. For instance, this can be achieved when the group $G$ is a semidirect product of small groups.