Best approximations of fitness functions of binary strings

Hang Zhang (h.zhang@cs.bham.ac.uk) and Jonathan E. Rowe (j.e.rowe@cs.bham.ac.uk)
School of Computer Science, University of Birmingham, Birmingham B15 2TT, Great Britain

Abstract. Fitness functions of binary strings (pseudo-boolean functions) can be represented as polynomials over a set of boolean variables. We show that any such function has a unique best approximation in the linear span of any subset of polynomials. For example, there is a unique best linear approximation and a unique best quadratic approximation. The error of an approximation here is root-mean-squared error. If all the details of the function to be approximated are known, then the approximation can be calculated directly. Of more practical importance, we give a method for using sampling to estimate the coefficients of the approximation, and describe its limitations.

Keywords: fitness function, pseudo-boolean function, approximation

1. Introduction

Evolutionary algorithms (EAs) have been found to be a good method for solving problems which are discontinuous and multi-modal. However, not all optimization problems can be solved by EAs efficiently, especially when the fitness functions are too noisy or the evaluation of the fitness is computationally very expensive. For example, this may be because the fitness function derives from a time-consuming simulation. One approach, in this case, is to estimate the fitness function by constructing an approximate model or surrogate which will be much quicker to optimize. (Jin, 2003) presents a comprehensive survey of the research on fitness approximation in evolutionary computation. However it only gives some general frameworks on how to build the approximate models. Of course, the approximations can come from many sources. For example, one may be able to make use of some underlying theory, or one might use function interpolation. In this paper we will concentrate on creating approximations to pseudo-boolean functions, which are a class of discrete functions often used as the fitness function of EAs. In addition to summarising general results related to this problem, we will also present new results concerning the possibilities and problems with finding approximations using sampling techniques.

In Section 2, we will give the definition of pseudo-boolean functions and map pseudo-boolean functions to a vector space. By using properties of vector spaces, we prove the existence of the unique best
approximation for any given pseudo-boolean function. The proof is presented in Section 3. Also in Section 3, we provide a method to obtain the formula of the best approximated functions. Several examples are given in Section 4 to illustrate the use of the method. A comparison of the results is made in this section as well. In Section 5, we point out a weakness of the method, which makes it impractical in real situations. Another method mentioned in (Hammer, 1992) is then briefly introduced. However, there are some restrictions to using this method. We then provide a general method using sampling techniques. Discussion on the results obtained by this method is in the last part of Section 5.

2. Pseudo-boolean functions and vector spaces

Let $B = \{0, 1\}$, then a pseudo-boolean function over $\ell$ variables is a function $f : B^\ell \rightarrow \mathbb{R}$ sometimes referred to as a fitness function. Any such function can be represented uniquely as a multilinear polynomial in $\ell$ boolean variables (Boros, 2001):

$$f(x_1, \ldots, x_\ell) = \sum_{k \in B^\ell} a_k \prod_{i : k_i = 1} x_i$$

Denote the set of all pseudo-boolean functions on $\ell$ variables by $PB(\ell)$. This set can be made into a vector space. If $p, q \in PB(\ell)$ and $\lambda, \mu$ are scalars, then define $\lambda p + \mu q \in PB(\ell)$ by

$$(\lambda p + \mu q)(x) = \lambda(p(x)) + \mu(q(x))$$

Let $n = 2^\ell$. Then given any function $p \in PB(\ell)$ we can write out its values as an $n$-dimensional vector which we will call the corresponding fitness vector

$$F(p) = (p(0,0,\ldots,0), \ldots, p(1,1,\ldots,1))$$

where we assume the standard binary ordering on $B^\ell$. In fact if we identify the integers $0, 1, \ldots, 2^\ell - 1$ with their binary representations as elements of $B^\ell$ then the fitness vector of $p \in PB(\ell)$ is given by $F(p)_i = p(i)$ for all $i = 0, 1, \ldots, 2^\ell - 1$. Clearly, any pseudo-boolean function has an associated fitness vector. Moreover, every possible fitness vector can be used to assign values to a pseudo-boolean function, which then corresponds to a unique multilinear polynomial. Thus $F : PB(\ell) \rightarrow \mathbb{R}^n$ is an invertible map between vector spaces. It is also a linear map, since

$$F(\lambda p + \mu q)_i = (\lambda p + \mu q)(i)$$
$$= \lambda p(i) + \mu q(i)$$
$$= \lambda F(p)_i + \mu F(q)_i$$
$$= (\lambda F(p) + \mu F(q))_i$$
3. Unique best approximations

3.1. Existence of the Unique Best Approximations

Given a function \( p \in PB(\ell) \) we may try to approximate it by some other function \( q \) from a restricted subset of \( PB(\ell) \). We will measure the error of this approximation by using the root-mean-squared error of the fitness values. That is, the error is the Euclidean distance between the corresponding fitness vectors:

\[
err(p, q) = \|F(p) - F(q)\| = \left( \sum_i (p(i) - q(i))^2 \right)^{1/2}
\]

Given a subset of functions \( A \subseteq PB(\ell) \), we seek a function \( q \in A \) such that \( err(p, q) \) is a minimum over all functions in \( A \). Such a function, if it exists, is called a best approximation to \( p \).

**Theorem 1.** Given a subset \( A \subseteq PB(\ell) \), let \( \mathcal{L}(A) \) be the linear span of \( A \) (that is, the set of all linear combinations of functions in \( A \)). Then given any pseudo-boolean function \( p \in PB(\ell) \), there exists a unique best approximation \( q \in \mathcal{L}(A) \).

**Proof** Since \( F \) is a linear map, then the image \( F(\mathcal{L}(A)) \) is a subspace of \( \mathbb{R}^n \). Consider \( x, y \in F(\mathcal{L}(A)) \). Then there exist \( a, b \in \mathcal{L}(A) \) such that \( x = F(a) \) and \( y = F(b) \). So given scalars \( \lambda, \mu \) we have

\[
\lambda x + \mu y = \lambda F(a) + \mu F(b) = F(\lambda a + \mu b) \in F(\mathcal{L}(A))
\]

Now \( \mathbb{R}^n \) with the standard inner product is a Hilbert space, and \( F(\mathcal{L}(A)) \) is a finite dimensional subspace. Therefore, given any \( u \in \mathbb{R}^n \), there is a unique best approximation \( v \in F(\mathcal{L}(A)) \). (This is a standard result from approximation theory — see for example (Kreyszig, 1989)). So given any function \( p \in PB(\ell) \), let \( u = F(p) \). Since \( F \) is an invertible map, there exists a function \( q \in \mathcal{L}(A) \) such that \( v = F(q) \) is the unique best approximation to \( p \) under the Euclidean norm.\(^1\)

\(^1\) An alternative proof of this result may be found in (Boros, 2001).
3.2. Calculate the best approximations

The following discussion is standard approximation theory, and follows (Kreyszig, 1989; Powell, 1981).

Let \( b^{(1)}, \ldots, b^{(m)} \) be a basis for \( \mathcal{L}(A) \). Then, since \( F \) is a invertible, \( F(b^{(1)}), \ldots, F(b^{(m)}) \) forms a basis for the image \( F(\mathcal{L}(A)) \). Given any function \( p \in PB(\ell) \), let \( q \) be the best approximation in \( \mathcal{L}(A) \). Then \( F(q) \) can be written as

\[
F(q) = \alpha_1 F(b^{(1)}) + \cdots + \alpha_m F(b^{(m)})
\]

which is the fitness vector corresponding to the function \( q \in \mathcal{L}(A) \). The error vector \( z = F(p) - F(q) \) is normal to the subspace. That is, for all vectors \( w \in F(\mathcal{L}(A)) \), we have \( w \cdot z = 0 \). Thus we can write down a system of \( m \) equations in the \( m \) unknowns \( \alpha_1, \ldots, \alpha_m \) given by

\[
F(p) \cdot F(b^{(k)}) = F(q) \cdot F(b^{(k)})
\]

for each \( k = 1, \ldots, m \).

Let \( M \) be the matrix \( M_{i,j} = F(b^{(i)}) \cdot F(b^{(j)}) \) and let \( D = \det M \). Then we can solve these equations directly by Cramer's rule:

\[
\alpha_k = \frac{D_k}{D}
\]

where \( D_k \) is the determinant obtained by replacing the \( k \)th column of the determinant \( D \) by the values \( F(p) \cdot F(b^{(1)}), \ldots, F(p) \cdot F(b^{(m)}) \).

4. Examples

4.1. Example: Linear approximation

Let us consider the problem of finding the unique best linear approximation to a non-linear fitness function. We will take \( \ell = 3 \) and consider the following pseudo-boolean function

\[
p(x_1, x_2, x_3) = 5x_1 + 13x_3 + 9x_1x_2 - 4x_1x_3 - 4x_2x_3 + 4x_1x_2x_3
\]

which has fitness vector \( F(p) = (0, 13, 0, 9, 5, 14, 14, 23) \). We take as the basis of our approximating subspace, the linear monomials \( x_1, x_2, x_3 \) and the constant function 1. Thus we seek the best approximation to \( p \) of the form

\[
q(x_1, x_2, x_3) = q_0 + q_1 x_1 + q_2 x_2 + q_3 x_3
\]
The fitness vectors corresponding to these functions form a basis of the subspace of \( \mathbb{R}^8 \)

\[
F(1) = (1, 1, 1, 1, 1, 1, 1, 1) \\
F(x_1) = (0, 0, 0, 1, 1, 1, 1, 1) \\
F(x_2) = (0, 0, 1, 1, 0, 0, 1, 1) \\
F(x_3) = (0, 1, 0, 1, 0, 1, 0, 1)
\]

The determinant \( D \) is therefore

\[
D = \begin{vmatrix} 8 & 4 & 4 & 4 \\ 4 & 4 & 2 & 2 \\ 4 & 2 & 4 & 2 \\ 4 & 2 & 2 & 4 \end{vmatrix} = 64
\]

We also require: \( F(p) \cdot F(1) = 78, F(p) \cdot F(x_1) = 56, F(p) \cdot F(x_2) = 46, F(p) \cdot F(x_3) = 59 \). Applying Cramer’s rule gives us

\[
q(x_1, x_2, x_3) = -1.25 + 8.5x_1 + 3.5x_2 + 10x_3
\]

which has corresponding fitness vector \( F(q) = (-1.25, 8.75, 2.25, 12.25, 7.25, 17.25, 10.75, 20.75) \) and error 8.155. Interestingly, from an optimization point of view, this approximation does have the same global optimum as the original function.

4.2. Example: Quadratic Approximation

We now look at quadratic approximation, taking as the basis for our approximating subspace the monomials \( 1, x_2x_3, x_1x_3, x_1x_2 \). These give a basis for the subspace of \( \mathbb{R}^8 \)

\[
F(1) = (1, 1, 1, 1, 1, 1, 1, 1) \\
F(x_1x_2) = (0, 0, 0, 0, 0, 0, 1, 1) \\
F(x_1x_3) = (0, 0, 0, 0, 0, 1, 0, 1) \\
F(x_2x_3) = (0, 0, 0, 1, 0, 0, 0, 1)
\]

Again following Cramer’s rule we get the unique best approximation \( 5 + 8x_1x_2 + 8x_1x_3 + 3x_2x_3 \). This has an error of 10.863 which is slightly worse than the linear approximation.

If we take all the linear, quadratic monomials and constant function 1 as the basis for the approximating subspace, then the approximation will be \( 0.5 + 4x_1 - x_2 + 12x_3 + 11x_1x_2 - 2x_1x_3 - 2x_2x_3 \). And the error will be only 1.414.
4.3. Example: Function with 10 Variables

Now, we consider a pseudo-boolean function with more variables. The following function has 10 variables.

\[
p(x_1, x_2, \ldots, x_{10}) = 5x_1 + 13x_3 + 7x_5 - 9x_6 + 9x_{10} - 4x_1x_3
- 20x_1x_9 - 4x_2x_3 + x_2x_5 - 9x_3x_4 + 7x_4x_8
+ 4x_1x_2x_3 + 7x_1x_2x_4 - 3x_3x_5x_6 - 12x_6x_8x_{10}
+ 20x_1x_5x_7x_9 - 8x_4x_7x_8x_9x_{10}
\]

The result of finding various approximations is shown in Table I. Here, we use two additional measures of the error of the approximation. One is the max error, which is defined as :

\[
max Err(p, q) = \| F(p) - F(q) \|_\infty = \max_i |p(i) - q(i)|
\]

(2.1)

The other is the average error. It is defined as:

\[
average Err(p, q) = \frac{1}{2^1} \| F(p) - F(q) \|_1 = \frac{1}{2^1} \sum_i |p(i) - q(i)|
\]

<table>
<thead>
<tr>
<th>Basis</th>
<th>Constant + Linear</th>
<th>Constant + Quadratic</th>
<th>Constant+Linear + Quadratic</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td>233.512</td>
<td>157.049</td>
<td>111.857</td>
</tr>
<tr>
<td>max error</td>
<td>23.5</td>
<td>22.045</td>
<td>11</td>
</tr>
<tr>
<td>average error</td>
<td>5.966</td>
<td>3.86</td>
<td>2.791</td>
</tr>
</tbody>
</table>

5. Estimating the coefficients by sampling

In order to calculate the coefficients of the best approximation, we need to know \( F(p) \), that is, the entire set of fitness values of \( p \). Obviously this is impractical in real situations.

In (Hammer, 1992), a characterization of the best \( k \)-th approximation of a pseudo-boolean function is obtained in terms of the average value of the function and its \( m \)-th order derivatives, where \( m = 0, 1, \ldots, k \).
THEOREM 2. Given a pseudo-boolean function \( f : B^n \to \mathbb{R} \) and an integer \( k, 0 \leq k \leq n \), the best \( k \)-th approximation of \( f \) is characterized as the unique function \( g : B^n \to \mathbb{R} \) of degree \( \leq k \) that agrees with \( f \) in all average \( m \)-th order derivatives for \( m = 0, 1, \ldots, k \).

Based on this theorem, explicit formulae for best \( k \)-th approximations can be derived. The details of the analysis can be found in (Hammer, 1992).

However, there are some restrictions on using this method. Firstly, to calculate the approximations, all the coefficients in the original pseudo-boolean functions have to be used, which means the explicit formulae of the original functions must be known. This may be the case, for example, with MAX-SAT problems. However, sometimes they are only a black box function, where we only know the input and output values. For those functions we cannot use this method to get their best approximations. Secondly, the best \( k \)-th approximations have to contain the subfunctions of all degrees less or equal to \( k \). For example, the best 2nd approximations must have constant, linear and quadratic subfunctions. An approximation that only contains constant and quadratic terms cannot be found by this method.

To make the process of finding the best approximations more general and practical, we will use the following method to estimate the approximations.

The idea is based on the method described in Section 2. Here, for each basis function \( b \) we will estimate the value of \( F(p) \cdot F(b) \), by samples. Assume \( b \) is a monomial function. That is, it is the product of boolean variables. Then

\[
F(p) \cdot F(b) = \sum_{x \in B^\ell} p(x) b(x) = \sum_{x : b(x) = 1} p(x)
\]

That is, the sum is only over those cases in which all the variables in \( b \) take on the value 1. Let \( w(b) \subseteq \{1, 2, \ldots, \ell\} \) be defined to be the set of indices corresponding to terms in the monomial \( b \). That is

\[
b(x) = \prod_{i \in w(b)} x_i
\]

Then, corresponding to each monomial \( b \) we have a set \( \phi(b) \subseteq B^\ell \) defined by

\[
\phi(b) = \{ x \in B^\ell : x_i = 1, \text{ for all } i \in w(b) \} \]
That is, $\phi(b)$ is the set of all binary strings that have a value 1 wherever the monomial $b$ has a corresponding term in its product. We can therefore write

$$F(p) \cdot F(b) = \sum_{x \in \phi(b)} p(x)$$

If the monomial $b$ contains $k$ terms (so $|w(b)| = k$), then there are $2^{\ell-k}$ elements in $\phi(b)$, and therefore the same number of terms in the sum. So we could estimate $F(p) \cdot F(b)$ by generating $N$ random binary strings from $\phi(b)$, taking the average fitness value, and then multiplying this by $2^{\ell-k}$.

For example, suppose we have the monomial $b(x) = x_2x_4$ and suppose $\ell = 8$. Then we generate $N$ random strings that match the pattern * 1 * 1 * * * * (where * indicates “don’t care”). We take the average of the fitness values of these strings, and multiply by $2^{8-2} = 64$ to get an estimate of $F(p) \cdot F(x_2x_4)$. For example, if we took $N = 4$ samples we might find:

$$p(01011100) = 4$$
$$p(11010111) = 6$$
$$p(11110010) = 5$$
$$p(01011000) = 3$$

These have an average value of $(4 + 6 + 5 + 3)/4 = 4.5$. Our estimate for $F(p) \cdot F(x_2x_4)$ is therefore $4.5 \times 64 = 288$.

For the constant function 1 we simply take an average over completely random strings and multiply by $2^\ell$.

For the 10-variable pseudo-boolean function used above, we used the sampling method to look for the following three kind of approximated functions: function with constant and linear terms; function with constant and quadratic terms and function with constant, linear and quadratic terms. Since for different basis function $b$, the size of the corresponding set $\phi(b)$ may also be different, we choose sampling binary strings from set $\phi(b)$ with a specified ratio. In our test the samples ratio is from 10% to 90%. the results is as follows:

The approximated function with constant, linear and quadratic terms should have a smaller error than the other two types of approximations. However, this is not always the case. From the results shown above, it can be seen that, if there are enough number of samples, the constant+linear+quadratic approximation can get good results. But if there is not a sufficient number of samples, the error can become very large.
Table II. Error of Approximations for 10-variable Pseudo-Boolean Function using Samples

<table>
<thead>
<tr>
<th>Samples ratio (%)</th>
<th>Error</th>
<th>Constant + Linear</th>
<th>Constant + Quadratic</th>
<th>Constant + Linear + Quadratic</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>295.39</td>
<td>466.64</td>
<td>800.30</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>269.89</td>
<td>334.71</td>
<td>515.43</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>257.68</td>
<td>280.97</td>
<td>395.26</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>246.83</td>
<td>243.48</td>
<td>295.04</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>243.26</td>
<td>218.22</td>
<td>253.49</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>240.58</td>
<td>199.05</td>
<td>246.22</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>236.67</td>
<td>183.64</td>
<td>198.31</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>235.64</td>
<td>174.65</td>
<td>169.99</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>234.66</td>
<td>164.88</td>
<td>138.55</td>
<td></td>
</tr>
</tbody>
</table>

The reason is the matrix $M$, where $M_{i,j} = F(h^{(i)}) \cdot F(h^{(j)})$. This matrix is an ill-conditioned matrix, which means a small change in the samples will affect the coefficients in the approximated function substantially. We use the condition number to measure the ill-conditioning of a matrix. The condition number is defined as the product of the norm of $M$ and the norm of $M^{-1}$. An ill-conditioned matrix will have large condition number. From figure 2, we can see the more variables the original function has, the larger the condition number is, especially when the approximation contains all the constant, linear and quadratic terms. This would be a very bad approximation if there are not too many samples.

Another issue that should be mentioned is the time consumed for getting the approximations (see table 6 and figure 6) which is significantly greater when linear terms are included. In practice, therefore, we can often achieve a better approximation, and obtain it more efficiently, using just constant and quadratic terms than if we additionally use linear terms.

6. Conclusions

In this study, we proved the existence of unique best approximation for any given pseudo-boolean function. The method for getting the approximation was evaluated through the examination of several test examples. The data obtained in this study showed that the results of
Figure 1. Error of Approximations

the approximation was acceptable, although it was not always so good when there was more variables.

Table III. Time Used(s)

<table>
<thead>
<tr>
<th>Samples ratio (%)</th>
<th>Time(s)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Constant + Linear</td>
<td>Constant + Quadratic</td>
<td>Constant + Linear + Quadratic</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1.37</td>
<td>2.68</td>
<td>4.43</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>3.43</td>
<td>7.03</td>
<td>8.85</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>6.57</td>
<td>11.35</td>
<td>19.50</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>10.98</td>
<td>17.76</td>
<td>28.48</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>17.32</td>
<td>30.38</td>
<td>40.98</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>26.13</td>
<td>37.94</td>
<td>61.24</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>39.40</td>
<td>53.29</td>
<td>92.11</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>61.46</td>
<td>78.74</td>
<td>134.08</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>105.30</td>
<td>130.47</td>
<td>205.98</td>
<td></td>
</tr>
</tbody>
</table>

Clearly, the results of the approximation by sampling are not very desirable. The errors may be very large. We analyzed the results and find the reason is the ill-conditioned matrix which we used to calculate
the coefficients of the approximated function. However, if we choose too many samples, it is impractical. So more sampling techniques need to be studied and applied to finding the best approximations of pseudo-boolean functions in future research.

References


Figure 3. Time Used for Finding the Approximation