Regression analysis and automorphic orbits in free groups of Rank 2.

Alexei D. Miasnikov, Robert M. Haralick
Department of Computer Science
The Graduate Center of The City University of New York
365 Fifth Avenue
New York, New York 10016
{amiasnikov,haralick}@gc.cuny.edu

Abstract

The main goal of this paper is to show that pattern recognition techniques can be successfully used in abstract algebra. We introduce a pattern recognition system to recognize words of minimal length in their automorphic orbits in free groups of rank 2. This system is based on linear regression and does not use any particular results from group theory. The corresponding classifier is very fast and surprisingly accurate.

1. Introduction

The field of pattern recognition has been actively developing for several decades. It has been successfully applied in a large number of diverse fields.

In this paper we show that pattern recognition techniques can be successfully used in abstract algebra and the theory of infinite groups in particular. The statistical approach gives one an exploratory methods which could be helpful in revealing hidden mathematical structures and formulating rigorous mathematical hypotheses. Our philosophy here that if irregular or non-random behavior has been observed during an experiment then there must be a pure mathematical reason behind this phenomenon, which can be uncovered by a proper statistical analysis. The discovered knowledge can be of great interest to mathematicians. In addition, one can try to develop new (perhaps probabilistic) methods to solve hard combinatorial problems in algebra.

Note that this is a very novel application area of pattern recognition. Some of the previous work can be found in [2, 4].

We start by giving a brief introduction to the Whitehead minimization problem. Let $X$ be a finite alphabet, $X^{-1} = \{ x^{-1} \mid x \in X \}$ be the set of formal inverses of letters from $X$, and $X^{\pm 1} = X \cup X^{-1}$. For a word $w$ in the alphabet $X^{\pm 1}$ by $|w|$ we denote the length of $w$. A word $w$ is called reduced if it does not contain subwords of the type $xx^{-1}$ or $x^{-1}x$ for $x \in X$. Applying reduction rules $xx^{-1} \to \varepsilon, x^{-1}x \to \varepsilon$ (where $\varepsilon$ is the empty word) one can reduce each word $w$ in the alphabet $X^{\pm 1}$ to a reduced word $\overline{w}$. The word $\overline{w}$ is uniquely defined and does not depend on the order in a particular sequence of reductions. The set $F = F(X)$ of all reduced words over $X^{\pm 1}$ forms a group with respect to multiplication defined by $u \cdot v = \overline{w}$ (i.e., to compute the product of words $u, v \in F$ one has to concatenate them and then reduce). The group $F$ with the multiplication defined as above is called a free group with basis $X$. The cardinality $|X|$ is called the rank of $F(X)$. Free groups play a central role in modern algebra and topology.

A bijection $\phi : F \to F$ is called an automorphism of $F$ if $\phi(uv) = \phi(u)\phi(v)$ for every $u, v \in F$. The set $\text{Aut}(F)$ of all automorphisms of $F$ forms a group with respect to composition of automorphisms. Every automorphism $\phi \in \text{Aut}(F)$ is completely determined by its images on elements from the basis $X$ since $\phi(x_1 \cdots x_n) = \phi(x_1) \cdots \phi(x_n)$ and $\phi(x^{-1}) = \phi(x)^{-1}$ for any letters $x_i, x_i \in X^{\pm 1}$. An automorphism $t \in \text{Aut}(F(X))$ is called a Whitehead’s automorphism if $t$ satisfies one of the two conditions below:

1) $t$ permutes elements in $X^{\pm 1}$,

2) $t$ fixes a given element $a \in X^{\pm 1}$ and maps each element $x \in X^{\pm 1}, x \neq a^{\pm 1}$ to one of the elements $x, xa, a^{-1}x, or a^{-1}xa$.

By $\Omega(X)$ we denote the set of all Whitehead’s automorphisms of $F(X)$. It is known [5] that every automorphism from $\text{Aut}(F)$ is a product of finitely many Whitehead’s automorphisms.

The automorphic orbit $\text{Orb}(w)$ of a word $w \in F$ is the set of all automorphic images of $w$ in $F$:

$$\text{Orb}(w) = \{v \in F \mid \exists \varphi \in \text{Aut}(F) \text{ such that } \varphi(w) = v\}.$$  

A word $w \in F$ is called minimal (or automorphically minimal) if $|w| \leq |\varphi(w)|$ for any $\varphi \in \text{Aut}(F)$. By $w_{\text{min}}$ we denote a word of minimal length in $\text{Orb}(w)$. Notice that $w_{\text{min}}$ is not unique. By $\text{WC}(w)$ (the Whitehead’s complex-
ity of \( w \) we denote a minimal number of automorphisms 
\( t_1, \ldots, t_m \in \Omega(X) \) such that \( t_m \ldots t_1(w) = w_{\text{min}} \). The 
algorithmic problem which requires finding \( w_{\text{min}} \) for a given 
\( w \in F \) is called the Minimization Problem for \( F \), it is one of the 
principal problems in combinatorial group theory and 
topology. There is a famous Whitehead’s decision algorithm 
for the Minimization Problem, it is based on the following 
result due to Whitehead ([7]): if a word \( w \in F(X) \) is not 
minimal then there exists an automorphism \( t \in \Omega(X) \) such 
that \( |t(w)| < |w| \). Unfortunately, its complexity depends 
on cardinality of \( \Omega(X) \) which is exponential in the rank of 
\( F(X) \). We refer to [4] for a detailed discussion on complex-
ity of Whitehead’s algorithms.

In this paper we focus on the Recognition Problem 
for minimal elements in \( F \). It follows immediately from the 
Whitehead’s result that \( w \in F \) is minimal if and only if 
\( |t(w)| \geq |w| \) for every \( t \in \Omega(X) \) (such elements sometimes 
are called Whitehead’s minimal). This gives one a simple 
deterministic decision algorithm for the Recognition Prob-
lem, which is of exponential time complexity in the rank of 
\( F \). Below we construct a probabilistic classifier which 
is based on linear regression, it has real time complexity and 
gives correct answers with a sufficiently high probability.

In fact, it is convenient to consider the Minimization 
Problem only for cyclically reduced words in \( F \). A word 
\( w = x_1 \ldots x_n \in F(X) \) \( (x_i \in X^{\pm 1}) \) is cyclically reduced 
if \( x_1 \neq x_n^{-1} \). Clearly, every \( w \in F \) can be presented in the 
form \( w = u^{-1} \tilde{w} u \) for some \( u \in F(X) \) and a cyclically 
reduced element \( \tilde{w} \in F(X) \) such that \( |w| = |\tilde{w}| + 2|u| \). This 
\( \tilde{w} \) is unique and it is called a cyclically reduced form of \( w \). 
Every minimal word in \( F \) is cyclically reduced, therefore, it 
suffices to construct a classifier only for cyclically reduced 
words in \( F \).

2. Recognition of minimal words in \( F_2 \)

In this section we describe a particular pattern recogni-
tion system for recognizing minimal elements in free groups 
of rank 2. The corresponding classifier is a supervised-
learning classifier based on linear regression model with a 
decision rule of the Bayes’ type.

2.1. Data generation: training datasets

A pseudo-random element \( w \) of \( F = F_2(X) \) can be 
generated as a pseudo-random sequence \( y_1, \ldots, y_l \) of elements 
\( y_i \in X^{\pm 1} \) such that \( y_i \neq y_{i+1} \), where the length \( l \) is also 
chosen pseudo-randomly. However, it has been shown in [4] 
and in [3] that randomly taken cyclic reduced words in \( F \) 
are already minimal with asymptotic probability 1. There-
fore, a set of randomly generated cyclically words in \( F \) 
would be highly biased toward the class of minimal ele-
ments. To obtain fair training datasets we use the following 
procedure.

For each positive integer \( l = 1, \ldots, 1000 \) we generate 
pseudo-randomly and uniformly 10 cyclically reduced 
words from \( F(X) \) of length \( l \). Denote the resulting set by 
\( W \). Then using the deterministic Whitehead algorithm we 
construct the corresponding set of minimal elements

\[
W_{\text{min}} = \{ w_{\text{min}} \mid w \in W \}.
\]

With probability 0.5 we substitute each \( v \in W_{\text{min}} \) with 
the word \( t(v) \), where \( t \) is a randomly and uniformly chosen 
automorphism from \( \Omega(X) \) such that \( |t(v)| > |v| \) (if 
\( |t(v)| = |v| \) we chose another \( t \in \Omega(X) \), and so on). Now, 
the resulting set \( L \) is a set of pseudo-randomly generated 
cyclically reduced words representing the classes of mini-
mal and non-minimal elements in approximately equal 
portions. It follows from the construction that our choice of 
non-minimal elements \( w \) is not quite representative, since 
all these elements have Whitehead’s complexity one (which 
is not the case in general). One may try to replace the au-
tomorphism \( t \) above by a random finite sequence of auto-
morphisms from \( \Omega \) to get a more representative training set. 
However, we will see in Section 3 that the training dataset 
\( L \) is sufficiently good already, so we elected to keep it as it is.

From the construction we know for each element \( v \in L \) 
whether it is minimal or not. Finally, we create a training set

\[
D = \{ v, P(v) \mid v \in L \},
\]

where

\[
P(v) = \begin{cases} 
1, & v \text{ is minimal;} \\
0, & \text{otherwise.}
\end{cases}
\]

2.2. Features

Let \( w \) be a reduced word in the alphabet \( X^{\pm 1} \). In this section 
we describe the features of \( w \) which characterize a cer-
tain placement of specific words from \( F(X) \) in \( w \).

Let \( U_2 \) be the set of all words in \( F_2 \) that are length 2. De-
note by \( C(w, u) \) the number of subwords \( u \in U_2 \) occurring 
in \( w \). The normalized value

\[
C(w, u) / |w|
\]

is a feature of \( w \) and feature vector

\[
f(w) = \frac{1}{|w|} < C(w, u) \mid \forall u \in U_2 >
\]
gives the numbers of occurrences of words of length two in 
\( w \) relative to the length of \( w \).

This is the basic feature vector in all our considerations, 
it corresponds to the so-called Whitehead graph of \( w ([5]) \).
2.3. Decision Rule

The classification algorithm has to predict the value \( P(w) \) of the predicate \( P \) for a given word \( w \). We use the regression classifier as the basis of the decision rule, (see [1], [6]). For any word \( w \) having feature vector \( f(w) \) we compute

\[
\hat{P}(w) = \beta' f(w),
\]

where \( \hat{P}(w) \) is the value of \( P(w) \) predicted by the regression model and \( \beta' \) is the vector of regression coefficients.

Unfortunately, we cannot guarantee that \( P(w) \) is, indeed, a linear function of \( f(w) \). We explore non-linear dependencies by using a general quadratic mapping. Let \( \varphi(f(w)) \) be a vector consisting of components of \( f(w) \) and all their pair-wise products written in some order. The corresponding prediction value

\[
\hat{P}(w) = \beta' \varphi(f(w)).
\]

The decision rule, \( \mathcal{R}(w) \), of minimal or not is made according to the following formula:

\[
\mathcal{R}(w) = \begin{cases} 
1, & \text{if } \hat{P}(w) > \Theta; \\
0, & \text{otherwise.}
\end{cases}
\]

(1)

where \( \Theta \) is a given threshold. However, there is an ambiguity in selection of the parameter \( \Theta \) in the decision rule (1).

Here we elected to use the following Bayesian type of the decision rule. Suppose an event \( \hat{P}(w) = \alpha \), where \( \alpha \in \mathbb{R} \), is observed. We are going to make a prediction on whether \( P(w) = 1 \) or \( P(w) = 0 \) based on estimations of conditional probabilities

\[
\Pr(P(w) = 1|\hat{P}(w) = \alpha) \text{ and } \Pr(P(w) = 0|\hat{P}(w) = \alpha).
\]

Let \( P_1(w) \) and \( P_0(w) \) denote the events \( P(w) = 1 \) and \( P(w) = 0 \) respectively. Similarly, by \( \hat{P}_\alpha(w) \) we denote event \( \hat{P}(w) = \alpha \). Theoretically, the decision rule is:

\[
\mathcal{R}(w) = \begin{cases} 
1, & \text{if } \Pr(P_1(w)|\hat{P}_\alpha(w)) > \Pr(P_0(w)|\hat{P}_\alpha(w)); \\
0, & \text{otherwise.}
\end{cases}
\]

(2)

Since we cannot compute the conditional probabilities above precisely, we estimate them as follows. We partition the set \( \mathbb{R} \) into intervals \( \Delta \) of equal length. Now, let \( \hat{P}_\Delta(w) \) denote event \( \hat{P}(w) \in \Delta \). We estimate the conditional probabilities:

\[
\Pr(P_1(w)|\hat{P}_\Delta(w)) \text{ and } \Pr(P_0(w)|\hat{P}_\Delta(w))
\]

Using Bayes’ formula one can rewrite the probabilities above \( i = 0, 1 \):

\[
\Pr(P_i(w)|\hat{P}_\Delta(w)) = \frac{\Pr(\hat{P}_\Delta(w)|P_i(w))\Pr(P_i(w))}{\Pr(\hat{P}_\Delta(w))}
\]

Therefore

\[
\Pr(P_1(w)|\hat{P}_\Delta(w)) > \Pr(P_0(w)|\hat{P}_\Delta(w))
\]

if and only if

\[
\Pr(\hat{P}_\Delta(w)|P_1(w))\Pr_1 > \Pr(\hat{P}_\Delta(w)|P_0(w))\Pr_0
\]

The probabilities \( \Pr_1 = \Pr(P_1(w)) \) and \( \Pr_0 = \Pr(P_0(w)) \) are prior probabilities corresponding to the distribution of minimal and non-minimal elements among the inputs given to the classifier. It is safe to assume that the prior probabilities are equal. Thus the inequality above takes the form

\[
\Pr(\hat{P}_\Delta(w)|P_1(w)) > \Pr(\hat{P}_\Delta(w)|P_0(w))
\]

The conditional probabilities above can be estimated from the given training dataset \( D \). For \( i = 0, 1 \) put

\[
d_i(\Delta) = \frac{|\{w \mid \hat{P}(w) \in \Delta, < w, i \in \xi \}|}{|D|}
\]

Then

\[
\Pr(\hat{P}(w) \in \Delta|P(w) = i) \approx d_i(\Delta), \quad i = 0, 1.
\]

Finally we can define the following decision rule, which is a variation of the Bayes’ decision rule above:

\[
\mathcal{R}(w) = \begin{cases} 
1, & \hat{P}(w) \in \Delta \text{ and } d_1(\Delta) > d_0(\Delta); \\
0, & \hat{P}(w) \in \Delta \text{ and } d_0(\Delta) > d_1(\Delta).
\end{cases}
\]

(3)

2.4. Test datasets

To test and evaluate our pattern recognition system we generate several test datasets of different types:

- A test set \( S_e \) which is generated by the same procedure as for the training set \( D \), but independently of \( D \).
- A test set \( S_R \) of pseudo-randomly generated cyclically reduced elements of \( F(X) \), as described in Section 2.1.
- A test set \( S_P \) of pseudo-randomly generated cyclically reduced primitive elements in \( F(X) \). Recall that \( w \in F(X) \) is primitive if and only if there exists a sequence of Whitehead automorphisms \( t_1 \ldots t_m \in \Omega(X) \) such that \( t_m \ldots t_1(x) = x \) for some \( x \in X^{\pm 1} \). Elements in \( S_P \) are generated by the procedure described in [4], which, roughly speaking, amounts to a random choice of \( x \in X^{\pm 1} \) and a random choice of a sequence of automorphisms \( t_1 \ldots t_m \in \Omega(X) \).
- A test set \( S_{10} \) which is generated in a way similar to the procedure used to generate the training set \( D \). The only difference is that the non-minimal elements are obtained by applying not one, but several randomly chosen automorphisms from \( \Omega(X) \). The number of such automorphisms is chosen uniformly randomly from the set \( \{1, \ldots, 10\} \), hence the name.

Some comparative characteristics of the generated datasets are given in Table 1.
3. Results of experiments

Let \( f(w) \) be the feature mapping discussed in Section 2.2, and \( f_\varphi(w) \) be the image of \( f(w) \) under the quadratic mapping \( \varphi \) as was discussed in Section 2.3.

We run experiments with two different classifiers \( P \) and \( P_\varphi \) which are based on the linear regression model applied to vectors \( f(w) \) and \( f_\varphi(w) \).

The results of evaluation of the accuracy of the classifier \( P \) are given in Table 2a. This data shows that the accuracy of \( P \) decreases when the Whitehead’s complexity of inputs grows.

However, the classifier \( P_\varphi \) achieves almost perfect classification accuracy (see Table 2b).

<table>
<thead>
<tr>
<th></th>
<th>( S_R )</th>
<th>( S_e )</th>
<th>( S_{10} )</th>
<th>( S_P )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>w</td>
<td>&gt; 0 )</td>
<td>0.960</td>
<td>0.954</td>
</tr>
<tr>
<td>(</td>
<td>w</td>
<td>&gt; 4 )</td>
<td>0.962</td>
<td>0.957</td>
</tr>
<tr>
<td>(</td>
<td>w</td>
<td>&gt; 100 )</td>
<td>0.984</td>
<td>0.975</td>
</tr>
</tbody>
</table>

\[ (a) \]

<table>
<thead>
<tr>
<th></th>
<th>( S_R )</th>
<th>( S_e )</th>
<th>( S_{10} )</th>
<th>( S_P )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>w</td>
<td>&gt; 0 )</td>
<td>0.991</td>
<td>0.995</td>
</tr>
<tr>
<td>(</td>
<td>w</td>
<td>&gt; 4 )</td>
<td>0.993</td>
<td>0.996</td>
</tr>
<tr>
<td>(</td>
<td>w</td>
<td>&gt; 100 )</td>
<td>1.000</td>
<td>1.000</td>
</tr>
</tbody>
</table>

\[ (b) \]

Table 2. Classification accuracy of the classifiers a) \( P \); b) \( P_\varphi \).

Conclusions:

- The classifier \( P_\varphi \) is remarkably reliable;
- Very short words are more difficult to classify (perhaps, because they do not provide sufficient information for the classifiers);
- The estimated conditional probabilities for \( P_\varphi \) (which come from the Bayes’ decision rule, see Section 2.3) are presented in Figure 1b. Clearly, the classes of minimal and non-minimal elements are separated around 0.5 with a small overlap. So the regression works perfectly with the threshold \( \Theta \approx 0.6 \). From the figure we can see that the probability of misclassification of classifier \( P \) is much higher then the one for \( P_\varphi \).

References