Pattern Recognition and Minimal Words in Free Groups of Rank 2.

Robert M. Haralick, Alexei D. Miasnikov, Alexei G. Myasnikov

April 12, 2004

Abstract

We describe a linear time probabilistic algorithm to recognize Whitehead minimal elements (elements of minimal length in their automorphic orbits) in free groups of rank 2. Moreover, for a non-minimal element the algorithm gives an automorphism that is most likely to reduce the length of the element. This method is based on linear regression and pattern recognition techniques.

1 Introduction

The field of pattern recognition (PR) has been actively developing for several decades. It has been successfully applied in a large number of diverse fields, ranging from computer vision and speech recognition to geological analysis.

The present paper shows that PR techniques can be successfully used in group theory. There are several potential benefits of this approach. Firstly, it helps to produce fast stochastic algorithms to solve problems in groups; secondly, PR suggests heuristics which improve, on average, the performance of known group theoretic algorithms; and, the last but not least, one may use PR to reveal hidden algebraic structures and formulate rigorous mathematical hypotheses (see [5, 8] for more examples). Indeed, we believe that if a stochastic algorithm performs very well or some statistical observations persistently occur, then there must be a pure mathematical reason behind this phenomenon, which can be uncovered by a proper statistical analysis.

We introduce a PR system that recognizes *minimal* (sometimes also called *Whitehead minimal*) words, i.e., words of minimal length in their automorphic orbits, in free groups of rank 2. The corresponding probabilistic classification algorithm, a *classifier*, is very fast (linear time algorithm) and recognizes minimal words correctly with the accuracy rate of more then 98%. The recognition system is based on linear regression and does not use any particular results from group theory. To the contrary, some recovered patterns suggest a new notion of a weighted labeled directed graph $\Gamma(w)$ associated with a word w in a free group F. The graph $\Gamma(w)$ seems to be quite useful in recognizing minimal elements in F; indeed, our classifiers of minimal elements based on $\Gamma(w)$ outperform the classifiers based on the Whitehead graph of w (at least, in the case of a simple linear regression model). Moreover, we have found a very simple PR system which partitions all non-minimal elements in F_2 into two clusters M_1 and M_2. It also partitions the set of all elementary Whitehead automorphisms into two subsets T_1 and T_2 such that, with high probability, only automorphisms from T_i can reduce the length of elements in M_i, $i = 1, 2$. This allows one to reduce the search for a length reducing automorphism for a given $w \in F_2$ by half.

We used here a very simple linear regression model as a base for our classifiers. For free groups of higher ranks other models (quadratic regression and vector support machines) provide more accurate classification [5].

2 Whitehead's minimization algorithm

In this section we give a brief introduction to Whitehead's minimization problem.

Let $F = F(X) = F_2(X)$ be a free group of rank 2 with basis X. Put $X^{\pm 1} = \{x^{\pm 1} \mid x \in X\}$. A word $w = x_1 \cdots x_n$ in the alphabet $X^{\pm 1}$ is called *reduced* if $x_i \neq x_{i+1}^{-1}$, and it is *cyclically reduced* if $x_1 \neq x_n^{-1}$. We view elements in F as reduced words in $X^{\pm 1}$. Clearly, every element w in F can be presented in the form $w = u^{-1}\widetilde{w}u$ for some $u \in F(X)$ and a cyclically reduced element $\widetilde{w} \in F(X)$ such that $|w| = |\widetilde{w}| + 2|u|$. This $\widetilde{w}$ is unique and is called the *cyclically reduced form* of w.

Let $Aut(F)$ be the set of all automorphisms of the group F. The automorphic orbit $Orb(w)$ of a word $w \in F$ is the set of all automorphic images of w in F:

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

A word $w \in F$ is called *minimal* (or *automorphicaly minimal*) if $|w| \leq |\varphi(w)|$ for any $\varphi \in Aut(F)$. By $w_{\min}$ we denote a word of minimal length in $Orb(w)$. Notice that $w_{\min}$ is not unique.

A *classifier for minimal elements* in a free group F has to determine, for an arbitrary given element $w \in F$, whether w is minimal or not. Since every minimal word in F is cyclically reduced and since there exists a very fast algorithm for cyclic reduction, it suffices to construct a classifier for cyclically reduced words in F.

There is a famous deterministic Whitehead's algorithm which, for a given $w \in F$, finds some $w_{\min}$ in at most quadratic number of steps with respect to $|w|$ [11]. In fact, this algorithm works for arbitrary free groups, but in higher ranks it becomes extremely inefficient (it is still quadratic in the length of the input, but the constants grow exponentially with the rank). We refer to [8] for a detailed discussion of the complexity of Whitehead's algorithms. Here we want to mention only a few basic ideas related to Whitehead's description of minimal words. We denote by $\Omega(X)$ the following set of automorphisms $t \in Aut(F(X))$ (called *Whitehead's automorphisms*):

(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 $a^{-1}xa$.

An element $w \in F(X)$ is called *Whitehead minimal* if $|t(w)| \geq |w|$ for every $t \in \Omega(X)$. In 1936 Whitehead proved that $w \in F$ is minimal if and only if it is Whitehead minimal [11]. This allows one to construct a simple deterministic classifier for minimal words in $F(X)$, its complexity depends on cardinality of $\Omega(X)$.

In the free group of rank 2 with basis $X = \{a, b\}$ the set $\Omega(X)$ consists of some permutations of $X^{\pm 1}$, conjugations by letters from $X^{\pm 1}$, and the set T of eight Nielsen automorphisms:

$$T = \{x \to xy^{\pm 1}, x \to y^{\pm 1}x \mid x, y \in \{a, b\}, x \neq y\}.$$

Since we are working only with cyclically reduced elements we can ignore conjugations in the deterministic decision algorithm for the minimality problem, as well as permutations (they always preserve the length of the word).

Our goal here is to study minimal elements in $F(X)$ by pattern recognition methods and construct a probabilistic classifier which has linear time complexity and gives correct answers with a small classification error.

3 Recognition of minimal words in F_2

One of the main applications of pattern recognition (PR) techniques is classification of a variety of given objects into categories. Usually classification algorithms or *classifiers* use a set of measurements (properties, characteristics) of objects, called *features*, which gives a descriptive representation for the objects.

In this section we describe a pattern recognition system MIN_2 for recognizing minimal elements in free groups of rank 2. The corresponding classifier is a supervised learning classifier which means that the decision algorithm is "trained" on a prearranged *training* dataset, in which each pattern is labeled with its true class label. The algorithm is based on linear regression model with a decision rule of the Bayes type.

We refer to [3] for a detailed introduction to pattern recognition techniques.

3.1 Data generation: training datasets

A random element w of $F = F_2(X)$ can be produced as the result of a no-return simple random walk on the Cayley graph of F with respect to the set of generators X (see [1] for details). In practice this amounts to a pseudo-random choice of a number l (the length of w), and a pseudo-random sequence $y_1, \ldots, y_l$ of elements $y_i \in X^{\pm 1}$ such that $y_i \neq y_{i+1}^{-1}$, where y_1 is chosen randomly from $X^{\pm 1}$ with probability $1/4$, and all others are chosen randomly with probability $1/3$. Similarly, one can pseudo-randomly generate cyclically reduced words in

F, i.e., words $w = y_1 \cdots y_l$ where $y_1 \neq y_l^{-1}$. As we have mentioned in the introduction it suffices to construct a classifier for cyclically reduced words in $X^{\pm 1}$.

At first glance, the obvious choice for the training dataset would be the set of randomly generated cyclically reduced words from F. However, it has been shown in [6] that randomly taken cyclic words in F are already minimal with asymptotic probability 1. Therefore, a set of randomly generated words would be highly biased toward the class of minimal elements. To obtain fair numbers of representatives from both classes 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. This choice of parameters is purely practical: we want to have long words but be able to execute experiments in reasonable amount of time. Denote the resulting set by W. Then, using the deterministic Whitehead algorithm, one can effectively construct the corresponding set of minimal elements

$$W_{\min} = \{w_{\min} | w \in W\}.$$

With probability 0.5 we substitute each $v \in W_{\min}$ with the word $\widetilde{t(v)}$, where t is a randomly and uniformly chosen automorphism from $\Omega(X)$ such that $|\widetilde{t(v)}| > |v|$ (if $|\widetilde{t(v)}| = |v|$ 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 minimal and non-minimal elements in approximately equal proportions. However, it seems that the class of non-minimal elements is not quite representative, since every its element w has Whitehead complexity 1, i.e., there exists a single Whitehead automorphism which reduces w to $w_{\min}$ (see [8] for details on Whitehead complexity). We will see in Section 4 that the set described above is a sufficiently good training dataset which is much easier to generate than a set with uniformly distributed Whitehead complexity of elements. A possible mathematical explanation of this phenomenon is mentioned in [8].

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

$$D = \{< v, P(v) > | v \in L\},$$

where

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

3.2 Features

Let w be a reduced word in the alphabet $\in X^{\pm 1}$. In this section we describe the features of w which characterize the pattern of occurrences of specific words from $F(X)$ as subwords in w.

Let $K \in \mathbb{N}$ be a natural number, $v_1, \ldots, v_K \in F(X)$ be words from $F(X)$, and $U_1, \ldots, U_{K+1} \subseteq F(X)$ be subsets of $F(X)$. Denote by

$$C(w, U_1 v_1 U_2 v_2 \cdots U_K v_K U_{K+1})$$

the number of subwords of the type $u_1 v_1 u_2 \cdots v_K u_{K+1}$, where $u_j \in U_j$, which occur in w. For fixed $K, v_1, \ldots, v_K, U_1, \ldots, U_{K+1}$, this defines a *counting function*

$$w \in F \longrightarrow C(w, U_1 v_1 \cdots v_K U_{K+1}) \in \mathbb{N} \tag{1}$$

The normalized value

$$\frac{1}{|w|} C(w, U_1 v_1 \cdots v_K U_{K+1})$$

is called a *feature* of w and the function

$$w \in F \longrightarrow \frac{1}{|w|} C(w, U_1 v_1 \cdots v_K U_{K+1}) \in \mathbb{R}$$

is called a *feature function* on F. Usually we omit U_i in our notations if $U_i = \emptyset$. If $\bar{C} = (C_1(w), \ldots, C_N(w))$ is a sequence of counting functions like (1) one can associate with w a vector of real numbers:

$$f_{\bar{C}}(w) = \frac{1}{|w|} < C_1(w), \ldots, C_N(w) > \in \mathbb{R}^N$$

which is called a feature vector. Every choice of the sequence $\bar{C}$ gives a vector $f_{\bar{C}}(w)$ which reflects the structure of w.

For example, if $a \in X^{\pm 1}$ then $C(w, a)$ counts the number of occurrences of the letter a in w. The feature vector (where for simplicity we assume that the components are written in some order which we do not specify)

$$f_0(w) = \frac{1}{|w|} < C(w, a) \mid a \in X^{\pm 1} >$$

shows the frequencies of occurrences of letters from $X^{\pm 1}$ in w. The feature vector

$$f_1(w) = \frac{1}{|w|} < C(w, v) \mid |v| = 2 >$$

shows the numbers of occurrences of words of length two in w relative to the length of w. If $x_1, x_2 \in X^{\pm 1}$ then the counting function $C(w, x_1 U x_2)$, where $U = X^{\pm 1}$ gives the number of occurrences of x_1 and x_2 in w one letter apart.

To visualize some structures described by the counting functions above we associate with a given word $w \in F(X)$ a weighted labeled directed graph $\Gamma(w)$. Put $V(\Gamma(w)) = X^{\pm 1}$. For given $x, y \in X^{\pm 1}$ and $v \in F(X)$ we connect the vertex x to the vertex y by an edge with a label v and weight $C(w, xvy)$. Now, with every edge from x to y with label xvy one can associate a counting function $C(w, xvy)$, and vice versa. It follows that every subgraph Γ of $\Gamma(w)$ gives rise to a particular set of counting functions $\bar{C}_\Gamma$ of the type $C(w, xvy)$, and conversely, every set $\bar{C}$ of counting functions of the type $C(w, xvy)$ determines a subgraph $\Gamma_{\bar{C}}$ of $\Gamma(w)$.

For instance, the feature mapping f_1 corresponds to the subgraph $\Gamma_1(w)$ of $\Gamma(w)$ which is in a sense a directed version of the so-called *Whitehead graph* of w [12].

Here is a reference to Whitehead graph

3.3 Model

The classification algorithm has to predict the value $P(w)$ of the predicate P for a given word w. One of the approaches is to explore the relationship between $P(w)$ and the corresponding feature vector $v = f(w)$ of the word w. We can try to approximate the value of $P(w)$ via a linear function on $f(w)$:

$$P(w) \approx \beta^T f(w),$$

where β is an unknown column vector of coefficients. Inferring β from the training set is the task of the classical linear regression analysis [2, 10]. Given a dataset

$$D = \{(w_i, P(w_i) \mid i = 1, \ldots, n\}$$

one can compute the feature vectors $f(w_i)$, $i = 1, \ldots, n$ and form the standard regression model as:

$$\mathcal{P} = V\beta + \epsilon,$$

where $\mathcal{P} =< P(w_1), \ldots, P(w_n) >$ is a (column) vector of the known values of P, V is the matrix

$$V = \begin{bmatrix} f(w_1) \\ \vdots \\ f(w_n) \end{bmatrix}$$

with the feature vectors as rows, β is a vector of unknown regression coefficients and ϵ represents the approximation error. Using the least squares method we find β such that the mean square error

$$\|\mathcal{P} - V\beta\|^2 = \|\epsilon\|^2$$

is as small as possible.

Now, for a given word w and the computed vector β one can obtain the value $\hat{P}(w)$ predicted by the regression model as

$$\hat{P}(w) = \beta^T f(w).$$

Packages for computing linear regression models are now standard and they are available in many software distributions [4, 7]. We used routines from SYLModel Library [9].

One of the possible classifiers based on linear regression model is as follows. Given a word $w \in F(X)$ it returns the answer *decide*(w) according to the following formula:

$$decide(w) = \begin{cases} 1, & \text{if } \hat{P}(w) > \Theta; \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

where Θ is a given threshold. However, there is an ambiguity in selection of the parameter Θ in the decision rule (2). Therefore we elected to use the following Bayesian 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

$$Pr(P(w) = 1 \mid \hat{P}(w) = \alpha) \quad \text{and} \quad Pr(P(w) = 0 \mid \hat{P}(w) = \alpha)$$

so, theoretically, the corresponding decision rule is:

$$decide(w) = \begin{cases} 1, & \text{if } Pr(P(w) = 1 \mid \hat{P}(w) = \alpha) > Pr(P(w) = 0 \mid \hat{P}(w) = \alpha); \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Since we cannot compute the conditional probabilities above precisely, we estimate them as follows. We partition the set $\mathbb{R}$ into intervals Δ of equal length and estimate the conditional probabilities:

$$Pr(P(w) = 1 \mid \hat{P}(w) \in \Delta) \quad \text{and} \quad Pr(P(w) = 0 \mid \hat{P}(w) \in \Delta)$$

Using Bayes' formula, one can rewrite these probabilities as:

$$Pr(P(w) = i \mid \hat{P}(w) \in \Delta) = \frac{Pr(\hat{P}(w) \in \Delta \mid P(w) = i) \cdot Pr(P(w) = i)}{Pr(\hat{P}(w) \in \Delta)}$$

(Here $i = 0, 1$.) Therefore

$$Pr(P(w) = 1 \mid \hat{P}(w) \in \Delta) \;>\; Pr(P(w) = 0 \mid \hat{P}(w) \in \Delta)$$

if and only if

$$Pr(\hat{P}(w) \in \Delta \mid P(w) = 1)P_1 \;>\; Pr(\hat{P}(w) \in \Delta \mid P(w) = 0)P_0, \tag{4}$$

where the probabilities $P_1 = Pr(P(w) = 1)$ and $P_0 = Pr(P(w) = 0)$ are prior probabilities corresponding to the distribution of minimal and non-minimal elements among the inputs given to the classifier. We have already mentioned that in general situation a randomly chosen element of a free group is Whitehead-minimal with probability 1. This makes the classification task simple if we assume that inputs will be chosen randomly. However, the rate of false positive error (the error of classifying non-minimal element as minimal, see Section 3.4.2) in this case will be very high. The class of non-minimal elements is of the same interest as the class of minimal elements. To avoid bias toward minimal elements, we choose a more conservative approach by choosing equal prior probabilities for both classes.

Thus the inequality (4) takes the form

$$Pr(\hat{P}(w) \in \Delta \mid P(w) = 1) > Pr(\hat{P}(w) \in \Delta \mid P(w) = 0)$$

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

$$d_i(\Delta) = |\{w \mid \hat{P}(w) \in \Delta, < w, i > \in D\}| \;/\; |D|.$$

Then

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

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

$$decide(w) = \begin{cases} 1, & \text{if } \hat{P}(w) \in \Delta \text{ and } d_1(\Delta) > d_0(\Delta) \text{ for some interval } \Delta; \\ 0, & \text{if } \hat{P}(w) \in \Delta \; and \; d_0(\Delta) > d_1(\Delta) \text{ for some interval } \Delta. \end{cases} \tag{5}$$

3.4 Evaluation

3.4.1 Test datasets

To test and evaluate our pattern recognition system MIN_2 we generate several test datasets of different type:

- 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 elements of $F(X)$. We used the random walk described in the beginning of Section 3.1 to generate S_R.
- A test set S_P of (pseudo-) randomly generated *primitive* elements in $F(X)$. Recall that $w \in F(X)$ is primitive if and only if there exists a sequence of Whitehead automorphisms $t_1 \ldots t_l \in \Omega(X)$ such that $xt_1 \ldots t_l = w$ for some $x \in X^{\pm 1}$ (here $wt = t(w)$ for $t \in \Omega(X)$). Elements in S_P are generated by the procedure described in [8], which, roughly speaking, amounts to a random choice of $x \in X^{\pm 1}$ and a random choice of a sequence of automorphisms $t_1 \ldots t_l \in \Omega(X)$.
- 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.

Dataset	size	% min	% non-min	(min,avg,max) word lengths
D	10000	51.9	48.1	(1,541,1202)
S_e	5000	49.5	50.5	(1,542,1200)
S_{10}	5000	48.6	51.4	(1,691,10629)
S_R	5000	98.8	1.2	(1,499,998)
S_P	6000	0	100	(2,30,3443)

Table 1: Description of the datasets.

3.4.2 Accuracy measure

Let D_{eval} be a test data set. To evaluate the performance of the given **PR** system we use a simple accuracy measure:

$$A = |\{w \mid decide(w) = P(w),\ w \in D_{\text{eval}}\}| \ / \ |D_{\text{eval}}|,$$

which gives the fraction of the correctly classified elements from the test set D_{eval}.

Notice, that the numbers of correctly classified elements follow the Binomial distribution and A is approximately normally distributed with mean μ and estimated variance $A(1-A)/|D_{\text{eval}}|$. One can compute a particular confidence interval for A. The estimated length of such an interval gives another measure of accuracy of the classifier.

For example, suppose we choose to compute the length of the 95 percent confidence interval for the mean of A. It is known that for the standard normal variable z

$$Pr\{|z| < 1.96\} \approx 0.95.$$

Therefore

$$Pr\left\{\left|(A-\mu) \ / \ \sqrt{A(1-A)/|D_{\text{eval}}|}\right| < 1.96\right\} \approx 0.95$$

$$Pr\left\{A - 1.96\sqrt{A(1-A)/|D_{\text{eval}}|} < \mu < A + 1.96\sqrt{A(1-A)/|D_{\text{eval}}|}\right\} \approx 0.95$$

The formula above gives an interval $I(A)$ where the expected value for accuracy A lies with nearly 95 percent confidence. Obviously, the smaller is the interval the better is our approximation.

Note, that there are two types of error, called false positive and false negative, that can occur during the classification of minimal elements. A false positive is an error of classifying a non-minimal element as minimal. A false negative error is when a minimal element is classified as a non-minimal element.

In some applications it could be important to maintain one of the errors at the predefined level even if it will cause the second error to increase. In our case we do not give any preference to either of the classes and will expect the rates of the two errors be relatively equal.

3.5 Feature selection algorithm

Let $\mathcal{S}$ be a **PR** system and **P** be the corresponding classifier. The performance of the classifier **P** often directly depends on the set of features built into $\mathcal{S}$. Sometimes it is possible to reduce the number of features in $\mathcal{S}$ maintaining the same level of classification accuracy of **P**, and even find more efficient combinations of the given features. The corresponding procedure is called *feature selection*. We give a description of one of possible procedures below.

Let $\mathcal{C}$ be a finite collection of counting functions (see Section 3.2). Every sequence $\bar{C} =< C_1, \ldots, C_l >$ of functions from $\mathcal{C}$ gives rise to the corresponding feature mapping $f_{\bar{C}}$. Denote by $\mathcal{S}_{\bar{C}}$ the **PR** system obtained from $\mathcal{S}$ by replacing the feature set in $\mathcal{S}$ by $\bar{C}$. Let $\mathbf{P}_{\bar{C}}$ be the classifier that corresponds to the system $\mathcal{S}_{\bar{C}}$. Every system $\mathcal{S}_{\bar{C}}$ has one and the same test data set D_{eval} and the same accuracy measure A. Denote by $A(f_{\bar{C}}) = A(\bar{C})$ the accuracy of the classifier $\mathbf{P}_{\bar{C}}$ evaluated on the set D_{eval}.

We implement the feature selection as an iterative greedy procedure. At each iteration i, we select a new feature mapping f_i with the current best evaluation value $A(f_i)$ and add it to the set $\mathcal{F}$ of feature mappings constructed before. The procedure stops in at most $|\mathcal{C}|$ iterations. The best overall feature mapping $f^* \in \mathcal{F}$ of minimal length is returned as the output of the procedure. More precisely, the algorithm proceeds as follows:

Iteration 1:

Choose $C_1 \in \mathcal{C}$ such that $A(C_1) = \max\{A(C) \mid C \in \mathcal{C}\}$;

Set $f_1 = f_{C_1}$ and $\mathcal{F} = \{f_1\}$

Iteration N**:**

Suppose feature mappings $f_1, \ldots, f_{N-1}$ are constructed and

$$f_{N-1} = f_{<C_1,\ldots C_{N-1}>}$$

for some $C_1, \ldots, C_{N-1} \in \mathcal{C}$. Choose $C_N \in \mathcal{C} \smallsetminus \{C_1, \ldots, C_{N-1}\}$ such that the sequence $\bar{C}_N =< C_1, \ldots, C_N >$ satisfies the following condition:

$$A(\bar{C}_N) = \max\left\{A(\bar{C}) \mid \bar{C} =< C_1, \ldots, C_{N-1}, C >,\ C \in \mathcal{C}\right\}.$$

Put $f_N = f_{\bar{C}_N}$ and $\mathcal{F} = \mathcal{F} \cup \{f_N\}$.

If $N = |\mathcal{C}|$ then STOP.

Output:

Put $A_{\max} = \max\{A(f) \mid f \in \mathcal{F}.\}$

Select the mapping $f^* \in \mathcal{F}$ such that $A(f^*) \in I(A_{\max})$, where $I(A_{\max})$ is the 95% confidence interval described in Section 3.4.2, and f^* has the smallest possible length among all such feature mappings.

Observe that this feature selection procedure does not check all possible feature mappings that can be built from the counting functions from $\mathcal{C}$. There would be too many of them even for reasonably small sets $\mathcal{C}$. Instead, it makes at most $|\mathcal{C}|$ iterations, though each iteration could be time consuming since it requires evaluation of the current classifier $\mathbf{P}_{\bar{C}}$.

4 Experiments

4.1 Evaluating classifiers

In this section we present results of evaluation of classifiers $\mathbf{P}_f$ on the test dataset S_e when f runs over a particular set of feature mappings. By $A(f)$ we denote the accuracy of the classifier $\mathbf{P}_f$.

Let

$$f_1(w) = \frac{1}{|w|} < C(w,v) \mid |v| = 2 >$$

be the feature mapping discussed in Section 3.2. Recall that in view of the characterization of feature mappings as corresponding to the subgraphs of the graph $\Gamma(w)$ (see the end of Section 3.2) the mapping f_1 corresponds to the subgraph $\Gamma_1(w)$ which is a directed analog of the Whitehead graph of w. The accuracy of the classifier $\mathbf{P}_1 = \mathbf{P}_{f_1}$ is over 95%, which is quite good, but leaves some room for improvements. Consider the following feature mappings which correspond to various subgraphs of the graph $\Gamma(w)$:

$$\begin{aligned}
f_2(w) &= \frac{1}{|w|} < C(w, x_1vx_2) \mid x_1, x_2 \in X^{\pm 1}, |v| = 1 >;\\
f_3(w) &= \frac{1}{|w|} < C(w, x_1vx_2) \mid x_1, x_2 \in X^{\pm 1}, |v| = 2 >;\\
f_4(w) &= \frac{1}{|w|} < C(w, x_1vx_2) \mid x_1, x_2 \in X^{\pm 1}, |v| = 3 >;\\
f_5(w) &= \frac{1}{|w|} < C(w, x_1vx_2) \mid x_1, x_2 \in X^{\pm 1}, 0 \le |v| \le 1 >;\\
f_6(w) &= \frac{1}{|w|} < C(w, x_1vx_2) \mid x_1, x_2 \in X^{\pm 1}, 0 \le |v| \le 3 > .
\end{aligned}$$

The results of evaluation of the classifiers $\mathbf{P}_i = \mathbf{P}_{f_i}$, $i = 1, \ldots, 6$ on S_e are given in Table 2. In all cases rates of false positive and false negative errors were very close to each other.

	$A(f_1)$	$A(f_2)$	$A(f_3)$	$A(f_4)$	$A(f_5)$	$A(f_6)$
$\|w\| > 0$	0.954	0.968	0.926	0.869	0.977	0.980
$\|w\| > 4$	0.957	0.969	0.927	0.870	0.977	0.981
$\|w\| > 100$	0.975	0.984	0.947	0.893	0.992	0.994

Table 2: Performance of the classifiers $\mathbf{P}_1, \ldots, \mathbf{P}_6$ on the set S_e.

Conclusions:

- The accuracy of the classifiers increases when one adds new edges to the graphs related to the feature mappings (though it is not clear what is the optimum set of features);
- The classifier $\mathbf{P}_6$ is the best so far, it is remarkably reliable;

- Very short words are difficult to classify (possibly because they do not provide sufficient information for the classifiers);
- The estimated conditional probabilities for $\mathbf{P}_6$ (which come from the Bayes' decision rule, see Section 3.3) are presented in Figure 1. 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 $\Theta = 0.5$.

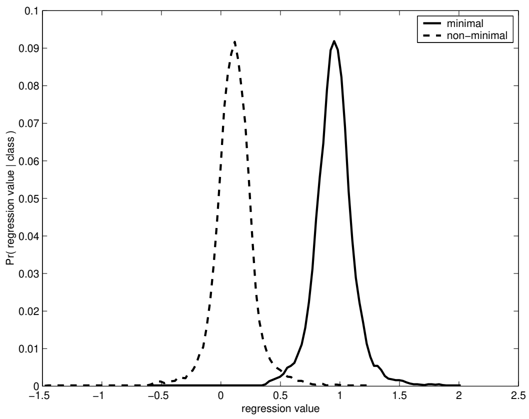


Figure 1: Conditional probabilities for $\mathbf{P}_6$.

4.2 Feature selection and analysis of the pattern recognition systems

In this section we are looking for a feature mapping which is at least as effective as f_6, but contains considerably less features. Observe, that f_6, as a vector, consists of 60 components (features). In search for the most effective feature mapping we apply the Feature Selection Algorithm from Section 3.5 to the set of all counting functions involved in f_6. Put

$$\mathcal{C} = \{C(w, xvy) \mid x, y \in X^{\pm 1},\, v \in F(X),\, 1 \leq |v| \leq 3\},$$

so counting functions from $\mathcal{C}$ correspond to edges of the graph $\Gamma_6(w)$.

We were very surprised when the Feature Selection Algorithm, when applied to the set $\mathcal{C}$, found a feature mapping based on only two counting functions:

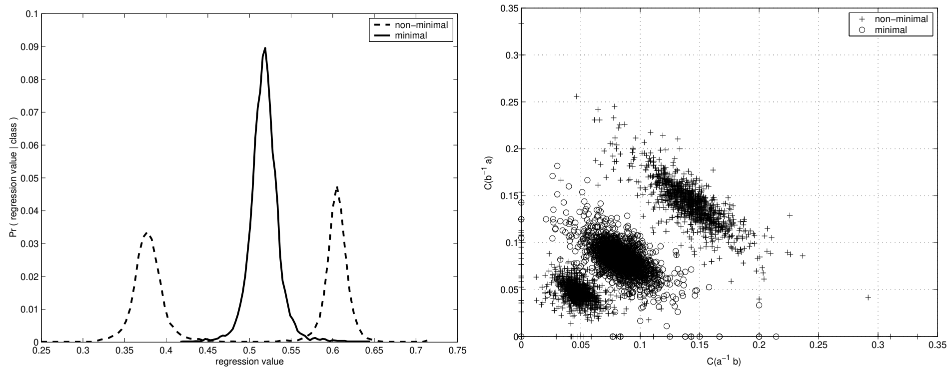
$$f^*(w) = \frac{1}{|w|} < C(w, a^{-1}b),\, C(w, b^{-1}a) >$$

where $X = \{a, b\}$.

The corresponding classifier $\mathbf{P}_* = \mathbf{P}_{f^*}$ showed the best overall performance when tested on the dataset S_e, the results of comparison of $\mathbf{P}_*$ with $\mathbf{P}_1$ and

	$A(f_1)$	$A(f_6)$	$A(f^*)$
$\|w\| > 0$	0.954	0.980	0.987
$\|w\| > 4$	0.957	0.981	0.989
$\|w\| > 100$	0.975	0.994	0.993

Table 3: Comparative results for $\mathbf{P}_*$.



(a) (b)

Figure 2: Results of experiments with $\mathbf{P}_*$: (a) Conditional probabilities for $\mathbf{P}_*$; (b) Scatter plot of points $f^*(w)$, $w \in S_e$.

$\mathbf{P}_6$ are presented in Table 3. The estimated conditional probabilities for $\mathbf{P}_*$ are given on Figure 2a.

One can see that non-minimal elements in S_e are divided into two clusters M_1 (left) and M_2 (right) such that the regression values for the class of the minimal elements lay in between the regression values for elements in M_1 and M_2. This shows that the linear regression cannot predict correctly values $P(w)$ of the predicate P for $w \in S_e$. Indeed, the standard threshold-based decision rule (2) will always give an error, at least, in 25% of trials, no matter what threshold value is chosen.

However, there is an obvious separation between minimal and non-minimal elements on Figure 2a and the Bayesian decision rule (5) was able to catch it. Since $f^*(w)$ is a two-dimensional vector, one can plot points $f^*(w), w \in S_e$, on a Euclidean plane. On Figure 2b we show scatter plot for f^*. Again, one can clearly see three groups of points. The one in the middle corresponds to the class of minimal elements, two others formed by non-minimal elements.

Now we test classifiers $\mathbf{P}_1, \mathbf{P}_6$, and $\mathbf{P}_*$ on the datasets S_R, S_{10}, and S_P. The results of these tests are given in Table 4.

One can see that the classifiers $\mathbf{P}_6, \mathbf{P}_*$ are robust and perform well even on datasets which are essentially different from the training dataset D. The classifier $\mathbf{P}_1$ has some difficulties with primitive elements.

What is left with no explanation so far is the unexpected partition of the

	S_{10}			S_R			S_P		
	$A(f_1)$	$A(f_6)$	$A(f^*)$	$A(f_1)$	$A(f_6)$	$A(f^*)$	$A(f_1)$	$A(f_6)$	$A(f^*)$
$\|w\| > 0$	0.828	0.981	0.981	0.960	0.978	0.967	0.567	0.879	0.945
$\|w\| > 4$	0.828	0.982	0.983	0.962	0.979	0.975	0.532	0.922	0.922
$\|w\| > 100$	0.842	0.994	0.993	0.984	0.993	0.992	0.494	1.000	0.979

Table 4: Performance of the classifiers $\mathbf{P}_1, \mathbf{P}_6, \mathbf{P}_*$ on the test datasets S_R, S_{10}, S_P.

class $NM(S_e)$ of non-minimal elements from S_e into the clusters M_1 and M_2 reflected on Figure 2a for the conditional probabilities for f^*.

A direct inspection of clusters M_1 and M_2 shows that the clustering was based on the type of elementary Whitehead automorphisms that reduce the length of a given element from $NM(S_e)$. More precisely, Table 5 shows that the set of elementary Nielsen automorphisms T can be partitioned into two subsets $T = T_1 \cup T_2$, where

$$\begin{aligned} T_1 &= \left\{ \begin{pmatrix} a \to ba \\ b \to b \end{pmatrix}, \begin{pmatrix} a \to ab \\ b \to b \end{pmatrix}, \begin{pmatrix} a \to a \\ b \to ab \end{pmatrix}, \begin{pmatrix} a \to a \\ b \to ba \end{pmatrix} \right\} \\ T_2 &= \left\{ \begin{pmatrix} a \to b^{-1}a \\ b \to b \end{pmatrix}, \begin{pmatrix} a \to ab^{-1} \\ b \to b \end{pmatrix}, \begin{pmatrix} a \to a \\ b \to a^{-1}b \end{pmatrix}, \begin{pmatrix} a \to a \\ b \to ba^{-1} \end{pmatrix} \right\} \end{aligned}$$

such that automorphisms from T_i are most likely to reduce the length of elements from the cluster M_i, and very rarely reduce the length of elements from the other cluster. Therefore, the classifier $\mathbf{P}_*$ not only solves the minimality classification problem, but it also appears to predict length reducing automorphisms for a given $w \in F(X)$.

The clusters M_1 and M_2 are defined on page 14

To find further evidence in support of this observation, we looked at the distributions of the conditional probabilities for f^* on the test datasets S_{10} and S_P. Even though the clustering structure of these datasets was more complicated, we were able to see a similar decomposition of the sets $NM(S_{10})$, $NM(S_P)$ of non-minimal elements in S_{10} and S_P into two clusters M_1 and M_2.

Table 5 shows that the sets T_1 and T_2 of Nielsen automorphisms play a similar role in clustering of $NM(S_{10})$ and $NM(S_P)$ as in $NM(S_e)$, thus expanding the scope of the observation made for $NM(S_e)$. It is significant that, for elements of length more than 100, the two clusters become mutually exclusive, i.e. none of the automorphisms from T_1 reduces elements in M_1 and vice versa. It shows again that "long" words are easier to classify.

One of the reasons why automorphisms from T_1 reduce length of elements from M_1 is that about 75 percent of elements in M_1 have positive exponent sum for one letter and negative exponent sum for another letter. Similarly, in about 75 percent of elements in M_2 the exponents sums are positive for both letters, so automorphisms from T_2 have a better chance to reduce the length of such elements. However, the accuracy of the recognizer is much higher then that, so there must be some other governing rule for such clustering. We are going to address this problem in the future, here we state the following conjecture.

Conjecture 1 *The set of feature vectors of non-minimal elements in a free group of rank 2 can be partition into finitely many bounded disjoint clusters in such a way that the length of elements in a cluster can be reduced by Nielsen automorphisms of a very particular type that correspond to this cluster. Moreover, these clusters can be separated from each other by hyperplanes.*

	$NM(S_e)$		$NM(S_{10})$		$NM(S_P)$	
	M_1	M_2	M_1	M_2	M_1	M_2
$a \to ba, b \to b$	0.7152	0.0008	0.7480	0.0008	0.76714	0.04057
$a \to ab, b \to b$	0.7136	0.0023	0.7488	0.0023	0.76714	0.04057
$a \to a, b \to ab$	0.7522	0.0000	0.7457	0.0023	0.76633	0.05428
$a \to a, b \to ba$	0.7458	0.0038	0.7417	0.0031	0.76633	0.05428
$a \to b^{-1}a, b \to b$	0.0016	0.7320	0.0000	0.7567	0.00000	0.69956
$a \to ab^{-1}, b \to b$	0.0016	0.7328	0.0008	0.7559	0.00000	0.69956
$a \to a, b \to a^{-1}b$	0.0000	0.7199	0.0008	0.7291	0.00000	0.69243
$a \to a, b \to ba^{-1}$	0.0008	0.7184	0.0000	0.7322	0.00000	0.69243

Table 5: Fraction of elements in $NM(S_e)$, $NM(S_{10})$ and $NM(S_P)$ reduced by automorphisms from T_1 and T_2.

Conclusions:

1. Feature Selection Algorithm is useful, it found by far the most economic and effective feature mapping f^*;

2. The classifier $\mathbf{P}_*$ not only solves the minimality classification problem, as a bonus it also predicts what are the most likely automorphisms which reduce the length of a given non-minimal element $w \in F(X)$.

References

[1] A. V. Borovik, A. G. Myasnikov and V. Remeslennikov. Multiplicative measures on groups. *Internat. J. Algebra Computat.* **13** (2003), 705–732.

[2] N. Draper and H. Smith. *Applied regression analysis.* (Wiley, 1998).

[3] R. O. Duda, P. E. Hart and D. G. Stork. *Pattern classification.* (Wiley-Interscience, 2000).

[4] B. E. Gough. *Gnu Scientific Library Reference Manual.* (Network Theory Ltd., 2003).

[5] R. M. Haralick, A. D. Miasnikov and A. G. Myasnikov. Heuristics for the Whitehead minimization problem. *Preprint* (2004).

[6] I. Kapovich, P. Schupp and V. Shpilrain. Generic properties of Whitehead's algorithm, stabilizers in $Aut(f_k)$ and one-relator groups. *Preprint* (2003).

[7] W. L. Martinez and A. R. Martinez. *Computational Statistics Handbook with MATLAB*. (CRC Press, 2001).

[8] A. Miasnikov and A. Myasnikov. Whitehead method and genetic algorithms. *Preprint* (2003).

[9] J. Rome, A. Miasnikov and R. Haralick. Regression models, generalized linear models and graphical models: A guide to using the SYLModel library. Technical report TR-2003011. (Department of Computer Science, Graduate Center of CUNY, 2003).

[10] T. Ryan. *Modern regression methods*. (John Wiley and Sons Inc., 1968).

[11] J. H. C. Whitehead. On equivalent sets of elements in a free group. *Annals Math.* **37** (1936), 782–800.

[12] J. H. C. Whitehead. On certain sets of elements in a free group. *Proc. London Math. Soc.* **41** (1936), 48–56.