An Automated Predictor of Hydrogen Gas Production from a Sucrose Based Bioreactor System

Kenneth REVETT[1], Florin GORUNESCU[2], Marina GORUNESCU[3], Nikhil[4], and Bestamin OZKAYA[4]

[1]Harrow School of Computer Science
University of Westminster
London, England HA1 3TP, UK
[2]University of Medicine and Pharmacy of Craiova, Romania,
[3]University of Craiova, Romania
[4]Tampere University of Technology
Department of Chemistry and Bioengineering
Tampere, Finland 33720
Revettk@westminster.ac.uk
fgorun@rdslink.ro
mgorun@inf.ucv.ro
nikhil, ari.visa@tut.fi

Abstract. In this study, we investigate the use of rough sets and an artificial neural network as a classification mechanism for predicting the hydrogen gas production from a sucrose-based completely stirred tank reactor (CSTR). The data that was modeled consisted of a set of typical CSTR parameters and the hydrogen gas production from this system over a period of 12 hours. There were a total of 12 attributes relating to the typical operation of a bioreactor, and a single continuous output variable, corresponding to the concentration of hydrogen gas produced over time. In this preliminary study, the goal was to investigate how the features/parameters correlated with the hydrogen gas output of the bioreactor. Rough sets was employed to determine the correlation of the features with the output variable - and during this process the parameter space was reduced. The results indicate that 4 of the 12 attributes were critical with respect to predicting the decision output. We tested the efficacy of the reduced feature space with respect to classification using a multi-layer perceptron type neural network. The results using a multi-layer perceptron (MLP) were in excess of 90% in most cases (full and reduced feature space), indicating that this approximation approach provides a reasonable model.

Keywords: Hydrogen gas production, Multi-layer perceptrons, reducts, rough sets
Math. Subject Classification 2000: 68T05

1 Introduction

With the current energy crisis, alternative sources of energy are in high demand. The utlization of hydrogen (H2) gas produced from typical waste products may provide a source of energy that is both clean and is based solely on recycled materials [1]. Typically, most hydrogen production is based on dark fermentation bioreactors - implemented as completely stirred tank reactors (CSTRs). These systems provide the medium through which microbiological processes convert waste products into an energy source. In this study, a dataset from a CSTR was examined in order to determine whether the hydrogen gas production output from a typical CSTR can be predicted based on the values of key bioreactor parameters, depicted in table 1. In addition, since there are a fairly large number of features - it is useful to know the information content of each of the features with respect to hydrogen gas production. For this task, rough sets was employed to produce a reduced feature space. Rough sets is used to reduce the feature space by generating a set of reducts - collections of non-redundant attributes that map attributes to decision classes. In this particular experiment, the decision class was based on the hydrogen gas production. Since hydrogen gas producton is a continuous variable, it was discretised into 3 categories (low, medium, and high), based on a clustering approach. Then rough sets was applied to produce a collection of decision rules, which can be used as a classification tool. In order to corroborate the classification results generated by the application of rough sets, a MLP was employed as an independent classification approach. Firstly, the MLP was trained on the full dataset, and the classification accuracy was examined using a 75/25 training/testing protocol. The MLP was then re-trained using the reduced dataset, using the same 75/25 training/testing protocol. In order to compare the two results, some constraints such as the number of training epochs and a reduced architecture (number of input/hidden nodes) were employed in order to make the pre/post reduction MLP based classifiers somewhat more realistic.

In the next section, we briefly describe the application of rough sets and neural networks as employed in this study. This is followed by a description of the dataset, followed by a brief presentation of the major results, and lastly a discussion of this work is presented.

2 Rough sets

Rough set theory is a relatively new data-mining technique used in the discovery of patterns within data first formally introduced by Pawlak in 1982 [2,3]. Since its inception, the rough sets approach has been successfully applied to deal with vague or imprecise concepts, extract knowledge from data, and to reason about knowledge derived from the data. We demonstrate that rough sets has the capacity to evaluate the importance (information content) of attributes,

discovers patterns within data, eliminates redundant attributes, and yields the minimum subset of attributes for the purpose of knowledge extraction.

The first step in the process of mining any dataset using rough sets is to transform the data into a decision table. In a decision table (DT), each row consists of an observation (also called an object) and each column is an attribute, one of which is the decision attribute for the observation d. Formally, a DT is a pair $A = (U, A\ d)$ where d ω A is the *decision attribute*, U is a finite non-empty set of objects called the *universe* and A is a finite non-empty set of attributes such that $a : U -> V_a$ is called the value set of a. Once the DT has been produced, the next stage entails cleansing the data.

There are several issues involved in small datasets - such as missing values, various types of data (categorical, nominal and interval) and multiple decision classes. Each of these potential problems must be addressed in order to maximise the information gain from a DT. Missing values is very often a problem in biomedical datasets and can arise in two different ways. It may be that an omission of a value for one or more subject was intentional - there was no reason to collect that measurement for this particular subject (i.e. 'not applicable' as opposed to 'not recorded'). In the second case, data was not available for a particular subject and therefore was omitted from the table. We have 2 options available to us: remove the incomplete records from the DT or try to estimate what the missing value(s) should be. The first method is obviously the simplest, but we may not be able to afford removing records if the DT is small to begin with. So we must derive some method for filling in missing data without biasing the DT. In many cases, an expert with the appropriate domain knowledge may provide assistance in determining what the missing value should be - or else is able to provide feedback on the estimation generated by the data collector. In this study, we employ a conditioned mean/mode fill method for data imputation. In each case, the mean or mode is used (in the event of a tie in the mode version, a random selection is used) to fill in the missing values, based on the particular attribute in question, conditioned on the particular decision class the attribute belongs to. There are many variations on this theme, and the interested reader is directed to [3] for an extended discussion on this critical issue. Once missing values are handled, the next step is to discretise the dataset. Rarely is the data contained within a DT all of ordinal type - they generally are composed of a mixture of ordinal and interval data. Discretisation refers to partitioning attributes into intervals - tantamount to searching for "cuts" in a decision tree. All values that lie within a given range are mapped onto the same value, transforming interval into categorical data. As an example of a discretisation technique, one can apply equal frequency binning, where a number of bins n is selected and after examining the histogram of each attribute, n-1 cuts are generated so that there is approximately the same number of items in each bin. See the discussion in [4,9] for details on this and other methods of discretisation that have been successfully applied in rough sets. Now that the DT has been pre-processed, the rough sets algorithm can be applied to the DT for the purposes of supervised classification.

The basic philosophy of rough sets is to reduce the elements (attributes) in a DT based on the information content of each attribute or collection of attributes (objects) such that the there is a mapping between similar objects and a corresponding decision class. In general, not all of the information contained in a DT is required: many of the attributes may be redundant in the sense that they do not directly influence which decision class a particular object belongs to. One of the primary goals of rough sets is to eliminate attributes that are redundant. Rough sets use the notion of the lower and upper approximation of sets in order to generate decision boundaries that are employed to classify objects. Consider a decision table A = (U, A d) and let $B \subseteq A$ and $X \subseteq U$. What we wish to do is to approximate X by the information contained in B by constructing the B-lower (B_L) and B-upper (B^U) approximation of X. The objects in B_L ($B_L X$) can be classified with certainty as members of X, while objects in B^U are not guaranteed to be members of X. The difference between the 2 approximations: B^U - B_L, determines whether the set is rough or not: if it is empty, the set is crisp otherwise it is a *rough set*. What we wish to do then is to partition the objects in the DT such that objects that are similar to one another (by virtue of their attribute values) are treated as a single entity. One potential difficulty arises in this regard is if the DT contains inconsistent data. In this case, antecedents with the same values map to different decision outcomes (or the same decision class maps to two or more sets of antecedents). This is unfortunately the norm in the case of small biomedical datasets, such as the one used in this study. There are means of handling this and the interested reader should consult [4] for a detailed discussion of this interesting topic. The next step is to reduce the DT to a collection of attributes/values that maximises the information content of the decision table. This step is accomplished through the use of the indiscernibility relation *IND(B)* and is defined for any subset $B \subseteq A$ ($B \subseteq A \cup \{d\}$).

The elements of *IND(B)* correspond to the notion of an equivalence class. The advantage of this process is that any member of the equivalence class can be used to represent the entire class - thereby reducing the dimensionality of the objects in the DT. This leads directly into the concept of a *reduct*, which is the minimal set of attributes from a DT that preserves the equivalence relation between conditioned attributes and decision values. It is the minimal amount of information required to distinguish objects with in U. The collection of all reducts that together provide classification of all objects in the DT is called the *CORE*(A). The CORE specifies the minimal set of elements/values in the DT which are required to correctly classify objects in the DT. Removing any element from this set reduces the classification accuracy. It should be noted that searching for minimal reducts is an NP-hard problem, but fortunately there are good heuristics that can compute a sufficient amount of reducts in reasonable time to be usable. In the software system that we employ an order based genetic algorithm (o-GA) which is used to search through the decision table for approximate reducts [5]. The reducts are approximate because we do not perform an exhaustive search via the o-GA which may miss one or

more attributes that should be included as a reduct. Once we have our set of reducts, we are ready to produce a set of rules that will form the basis for object classification.

Rough sets generates a collection of 'if..then..' decision rules that are used to classify the objects in the DT. These rules are generated from the application of reducts to the decision table, looking for instances where the conditionals match those contained in the set of reducts and reading off the values from the DT. If the data is consistent, then all objects with the same conditional values as those found in a particular reduct will always map to the same decision value. In many cases though, the DT is not consistent, and instead we must contend with some amount of indeterminism. In this case, a decision has to be made regarding which decision class should be used when there are more than 1 matching conditioned attribute values. Simple voting may work in many cases, where votes are cast in proportion to the support of the particular class of objects. In addition to inconsistencies within the data, the primary challenge in inducing rules from decision tables is in the determination of which attributes should be included in the conditional part of the rule. If the rules are too detailed (i.e. they incorporate reducts that are maximal in length), they will tend to overfit the training set and classify weakly on test cases. What are generally sought in this regard are rules that possess low cardinality, as this makes the rules more generally applicable. This idea is analogous to the building block hypothesis used in genetics algorithms, where we wish to select for highly accurate and low defining length gene segments. There are many variations on rule generation, which are implemented through the formation of alternative types of reducts such as *dynamic* and *approximate* reducts. Discussion of these ideas is beyond the scope of this paper and the interested reader is directed towards [4] for a detailed discussion of these alternatives. In the next section, we describe the application of a multi-layer perceptron based neural network, trained with the back-propagation algorithm. This is followed by the results of this work, and lastly a conclusion follows.

3 Neural networks

Machine learning (ML), one of the broad subfield of the Artificial Intelligence, is concerned with the development of algorithms and techniques that allow computers to "learn". One of the main topics covered by ML is represented by the artificial neural networks or, simply, neural networks (NN's), also known as neural computing, attempting to imitate the way a human brain works, by creating connections between processing elements, the computer equivalent of neurons. NN is an information processing paradigm that is inspired by the way the brain processes information. The key of this paradigm is the novel architecture of the information processing system, consisting of a large number of highly interconnected processing elements (neurons) working together to solve

specific problems. NN's represent a Computer Science discipline concerned with nonprogrammed adaptive information processing systems that develop associations between objects and response to their environment. The basic unit of any NN is represented by the artificial neuron, which captures the essence of the biological neural model. Basically, the neuron receives a certain number of inputs x_i and sums them to produce an output. Usually the sums of each node are weighted (the weight parameters w_i), and the sum is passed through the activation function, to produce the output of the neuron.

There are three phases in neural information processing: the training phase, the testing phase and the using phase. In the training phase, a training dataset is used to determine the weight parameters w_i that define the neural model. This trained neural model will be then tested on a testing dataset, different from the training dataset, in order to check up the model performance on a new dataset. Finally, the network will be used later in the using phase to process real, unknown, patterns, yielding classification results. One of the most used NN's is the multi-layer perceptron (MLP). A MLP has three distinctive characteristics, making it capable, at least theoretically, to represent a wide range of computable functions:
- The model of each neuron in the network usually includes a smooth (i.e. differentiable everywhere) nonlinear activation function, as opposed to the Rosenblatt's perceptron, generalizing the input-output relation of the network;
- The network contains one or more layers of hidden neurons that are not part of the input or output of the network, enabling the network to learn complex tasks by extracting progressively more meaningful features from the input data;
- The network exhibits a high degree of connectivity between neurons.

Remark. Note that networks with just two layers are capable of approximating any continuous function. The architecture of NN (number of neurons and topology of connections) can have significant impact on its performance in any particular application. Various techniques have been developed for optimizing the architecture, in some cases as part of the network training process itself. Techniques such as an exhaustive search through a restricted class of network architecture, pruning algorithms or network committee and mixture of experts are commonly adopted in practice. To conclude, in the training mode, NN is trained to associate outputs with input patterns. When the NN is used, it identifies the input pattern and tries to output the associated output pattern. If a pattern that has no output associated with it is given as an input, NN gives the output that corresponds to a taught input pattern that is least different from the given pattern. For more details concerning NN's, see [6], [7].

4 Bioreactor attributes

A typical bioreactor for producing hydrogen gas (biohydrogen) typically deploys the use of microorganisms for fermenting waste products, yielding carbon dioxide (CO_2) and hydrogen gas (H_2), the later of which can be used in

many industrial applications, in addition to providing a usable source of energy [8], [9]. A typical bioengineering approach to this fermentation process deploys a completely stirred reactor tank (CSRT). The sledge is added, along with sucrose, as a direct food source for microorganisms, which then proceed to produce the relevant gases. In this experiment, the bioreactor was operated in a continuous feeding mode and the hydraulic reaction time (HRT) was 12 hours. During the HRT, the system was stabilized by monitoring and correcting for pH (set at 6.7), and maintained until a steady-state was reached, and 6-10 samples of the parameters were recorded for model development. Table 1 below provides the parameters that were monitored in this study, and were used as inputs to both MLP neural network and rough sets.

Table 1. Set of attributes and decision classes used in this study.

Parameter Name	**Data type / Correlation**
Recycle ratio	Categorical
Sucrose concentration	Categorical
Substrate degradation %	Continuous/(0.18)
Biomass	Continuous /(0.27)
pH	Continuous/ (0.31)
Alkalinity	Continuous / (0.15)
ORP (oxidation-reduction potential)	Continuous / (-0.12)
Ethanol concentration	Continuous / (0.29)
Acetate concentration	Continuous / (0.21)
Butyrate concentration	Continuous / (0.16)
HRT	Categorical
CO_2 concentration	Discretised - Decision class
H_2 concentration	Discretized - Decision class

Note that since this study was investigating hydrogen gas production only, the CO_2 decision class was not used in this study.

5 Results

The principal purpose of this study was to investigate whether there was a differential information content of the various bioreactor attributes. To this end, the rough sets approach to data mining was applied in order to quantify the relative importance of the features against their decision class in the context of classification accuracy. In order to apply rough sets, the data must be discretised in order to reduce the number of rules to a reasonable value. Since most of the features were continuous, including the decision attribute, the features were discretized using an equal frequency binning approach (see [4] for other examples of this approach). The decision attribute was binarised based on the mean value: those values less then or equal to the mean were set to '0' and those above the mean were set to '1'. Then the decision table was split into

a 75/25 (209/69) training/testing paradigm, and this process was repeated 50 times, with replacement, and the results reported here are the averages of those 50 trials. Reducts were generated using an exhaustive approach (available from Rosetta v 1.4.1 - the rough sets software employed in this study - see [10]), and the decision rules were generated. The decision rules were then used to classify the testing set, from which the results can be summarised within Rosetta via a confusion matrix. Table 2 presents a confusion matrix, which summarizes in tabular form the type I & II errors, the positive and negative predictive values (PPV & NPV respectively), and the overall accuracy of the classification.

Table 2. Sample confusion matrix using the full dataset, based on a 72/25 training/testing split of the data.

Decision class	0	1	
0	39	0	1
1	1	29	0.967
	0.975	1.00	**0.985**

Note that bold value in the lower right corner is the overall classification accuracy.

The application of rough sets to the full dataset yielded a significant number of rules (4,480) without any rule filtering. By filtering on the left-hand support (the number of times a particular feature appears as an antecedent within the rule set), the number of rules was reduced without significantly reducing the classification accuracy (average classification accuracy of 93.7% from 1,088 rules). One of the principle results obtainable from the rule base is statistics on the particular attributes that were found within the rule set. In this study, the rule set contained 7 of the total number of features (11 in total), a significant reduction in the feature space. Note that the time and CO_2 features were not included in any of the experiments reported in this paper. Table 3 presents several rules that support each decision class (low or high hydrogen gas production):

Table 3. Sample of rules that were generated using the complete set of attributes.

substrate degradation (%)([98.57470, 98.58360)) AND Biomass (g VSS/l)([*, 3.50)) AND ORP (mV)([*, -431)) =>	=> High hydrogen gas production
substrate degradation (%)([99.35820, 99.45350)) AND Biomass (g VSS/l)([3.81, 3.87)) AND ORP (mV)([*, -431)) =>	=> Low hydrogen gas production
substrate degradation (%)([99.13690, *)) AND Biomass (g VSS/l)([*, 3.39)) AND pH([*, 6.59)) AND ALK (mg/L as CaCO3)([*, 4575)) AND EtOH (mg COD/L)([3140.68994, *)) AND HAc (mg COD/L)([2932.27002, *)) AND HPr (mg COD/L)([870.62305, *))=>	=> Low hydrogen gas production

The left-hand column depicts the attributes and their discritized values that yield the consequents depicted in the right-hand column.

The data in table 3 (the 3^rd rule) contains all of the attributes that were informative in the decision mapping process - the rest of the attributes were not included in the rule set (at least with any significant support). To corroborate the classification efficacy of this particular decision table, a multi-layer Perceptron based neural network (MLP-NN), trained with a standard vanilla back-propagation algorithm (via Matlab) was used to produce a classifier, based on the full feature set and the reduced set obtained from the rough sets analysis. Table 4 presents a summary of the application of the MLP-NN to the full dataset. Note that a k-means algorithm was used to generate two decision classes prior to the application of the MLP-NN.

Table 3. The classification error associated with running the full dataset through the MLP-NN, as a function of the number of hidden nodes.

	No. hidden neurons	**Training performance**	**Testing performance**
1	1	0.8333333	0.8804348
2	1	0.8333333	0.8804348
3	2	0.8494624	0.8804348
4	2	0.8602151	0.9021739
5	8	0.8924731	0.8695652
6	6	0.9247312	0.8695652
7	6	0.8978495	0.8586957
8	8	0.9569892	0.8369565
9	8	0.9408602	0.8369565
10	14	0.9516129	0.8478261
11	8	0.9516129	0.8478261
12	14	0.9731183	0.8804348
13	10	0.9731183	0.9021739
14	10	0.9139785	0.8478261
15	16	0.9784946	0.8913043
16	16	0.983871	0.8043478
Average/SD performance	**5-Aug**	**0.92 / 0.05**	**0.86 / 0.03**

The data was trained and tested on a 75/25 split of the data.

The MLP-NN was also run using the reduced dataset (containing 7 of the attributes), under the same conditions as was used to generate the data in table 4. The classification accuracy decreased somewhat, to an average of 90.5% (0.09), not statistically different from the results employing the complete dataset. These results confirm the classification accuracy was only marginally diminished when using the reduced dataset.

6 Conclusions

The results from this preliminary study indicate that rough sets is a useful technique for analyse data consisting of complex sets of features which may contain many levels of non-linearity. The application of rough sets not only reduced the input feature space from 11 to 7 attributes, but also provided a classification accuracy that surpassed a standard MLP-NN trained with vanilla back-propagation. The discretization which is required for the application of rough sets is a critical feature in this processing pipeline, and clustering or some statistical measure such as the mean can be sued to automated this process. In this work, the mean was used to partition the decision attribute into 2 classes - though equal frequency binning was used to discretise the other features. This is a usual practise when applying rough sets to a dataset, as the discretization is conditioned on the decision class, and hence can not be applied to discretize the decision class attribute. Exploring the discretization of the decision class is a timely process - and requires further exploration, a matter for future work with this dataset.

References

1. **Das, D. & Veziroglu**, Hydrogen production by biological processes: a survey of literature, Int J. Hydrogen Energy, 26, pp. 13-28, (2001).
2. **Pawlak, Z., Rough Sets**, International Journal of Computer and Information Sciences, 11, pp. 341-356, (1982).
3. **Pawlak, Z.**, Rough sets - Theoretical aspects of reasoning about data. Kluwer, (1991).
4. **Slezak.D.**, Approximate Entropy Reducts. Fundamenta Informaticae, (2002).
5. **Wroblewski, J.**, Theoretical Foundations of Order-Based Genetic Algorithms. Fundamenta Informaticae 28(3-4) pp. 423-430, (1996).
6. **Gorunescu, F, Gorunescu, M, El-Darzi, E, Gorunescu, S., & Revett K.**, A Cancer Diagnosis System Based on Rough Sets and Probabilistic Neural Networks, First European Conference on Health care Modelling and Computation, University of medicine and Pharmacy of Craiova, pp 149-159.
7. **Gorunescu, M., & Revett, K.**, A novel noninvasive cancer diagnosis using neural networks, The 7th International Conference on ARTIFICIAL INTELLIGENCE and DIGITAL COMMUNICATIONS, AIDC 2007, September 15-16, Craiova, ROMANIA, (2007).
8. **Nikhil, Ozkaya, B., Visa.A. , Chiu-Yue, L., Puhakka, J.A., & Yli-Harja, O.**, An rtificial neural network based model for predicting Hydrogen production in a sucrose-based bioreactor system, Proceedings of World Academy of Science, engineering,a nd Technology, vol 27, Cairo, Egypt, 6-8 February, 2008, pp. 20-25, (2008).
9. **Kapdam, I.K. & Kargi, F.**, Bio-hydrogen production from waste materials, Enzyme Microb Tech, 38, pp. 1-39, (2007).
10. ***, Rosetta: http://www.idi.ntnu.no/∼aleks/rosetta