IMPROVING COTTON SPINNING QUALITY USING FUZZY SETS

Maria Elisabete Cabeço Silva
and Antonio Alberto Cabeço Silva
School of Engineering
University of Minho
Guimarães, Portugal

Abstract

This paper aims at describing the work conducted at Textile Engineering Department, University of Minho (Portugal) in the field of the applications of neural and fuzzy systems on Cotton Spinning.

Those days, the field has gained a more solid background by linking to the traditional systems sciences.

A textile engineer, who is faced with the characterization or the prediction of the plant behavior, has to model the considered process. The needs for process models arise from various requirements. In process design, one wants to understand the mechanical and physical phenomena in order to develop the process.

In control, the short-term behavior and dynamics of the process may need to be predicted. Anomalies in different parts of the process can be detected by comparing a model of known behavior with the measured behavior. The optimal operating strategies can be examined by simulating the process behavior under different conditions.

For linear processes, a multitude of efficient techniques exist already, as linear regression can be used in identification. For simple input-output relations, linear models are a relatively robust alternative. They are simple and efficient also when extending to the identification of adaptive dynamic models, and readily available control design methods can be found from the literature. With suitable preprocessing or reparametrisation, a seemingly non-linear problem may often be converted into a linear one.

However, cotton spinning processes are non-linear and poorly known. As the processes become more complex, a sufficiently correct non-linear input-output behavior is more difficult to obtain using linear methods. Whereas the linear black-box models have been extensively studied and can be handled fairly well, the non-linear case is more difficult. The literature is spread under various fields, such as neural networks and fuzzy models.

Our current emphasis is in neuro-fuzzy systems, where we expect to find the way to create models so transparent, that even a less experienced textile engineer faced with the need of characterization of a spinning plant can find them useful.

Neuro-fuzzy combination is considered for several years already. However, the term “neuro-fuzzy” still lacks of proper definition, and it has the flavor of a “buzz word”. In this paper we try to give it a meaning in the context of fuzzy classification systems. From our point of view “neuro-fuzzy” means the employment of heuristic learning strategies derived from the domain of neural network theory to support the development of a fuzzy system. We illustrate our ideas using our “TEXPERT NEUROFUZZY CLASSER” model, which is used to create a fuzzy classification system from data.

Reprinted from the *Proceedings of the Beltwide Cotton Conference*
Volume 2:1331-1334 (1999)
National Cotton Council, Memphis TN

Introduction

Ever since fuzzy systems were applied in industrial applications, developers know that the construction of a well performing fuzzy system is not always easy. The problem of finding appropriate membership functions and fuzzy rules is often a tiring process of trial and error. Therefore, the idea of applying learning algorithms to fuzzy systems was considered early. These kind of adaptive models usually use knowledge-based methods. However, neural networks give another possibility of learning parameters of fuzzy systems.

The learning capabilities of neural networks made them a prime target for a combination with fuzzy systems in order to automate or support the process of developing a fuzzy system for a given task. The first so-called neuro-fuzzy approaches were considered mainly in the domain of neuro-fuzzy control, but today the approach is more general. Neuro-fuzzy systems are applied in various domains, e.g. control, data analysis, decision support, etc.

Modern neuro-fuzzy-systems are usually represented as a multilayer feedforward neural network, but fuzzifications of other neural network architectures are also considered, for example self--organizing feature maps. In neuro-fuzzy models, connection weights and propagation and activation functions differ from common neural networks. Although there are a lot of different approaches, we want to restrict the term “neuro-fuzzy” to systems which display the following properties:

1. A neuro-fuzzy system is a fuzzy system that is trained by a learning algorithm (usually) derived from neural network theory. The (heuristical) learning procedure operates on local information, and causes only local modifications in the underlying fuzzy system. The learning process is not knowledge based, but data driven.
2. A neuro-fuzzy system can be viewed as a special 3-layer feedforward neural network. The units in this network use *t*-norms or *t*-conorms

instead of the activation functions common in neural networks. The first layer represents input variables, the middle (hidden) layer represents fuzzy rules and the third layer represents output variables. Fuzzy sets are encoded as (fuzzy) connection weights. This view of a fuzzy system illustrates the data flow within the system, and its parallel nature. However this neural network view is not a prerequisite for applying a learning procedure, it is merely a convenience.

3. A neuro-fuzzy system can always (i.e. before, during and after learning) be interpreted as a system of fuzzy rules. It is both possible to create the system out of training data from scratch, and it is possible to initialize it by prior knowledge in form of fuzzy rules.
4. The learning procedure of a neuro-fuzzy system considers the semantical properties of the underlying fuzzy system. This results in constraints on the possible modifications applicable to the system parameters.
5. A neuro-fuzzy system approximates an *n*-dimensional (unknown) function that is partially given by the training data. The fuzzy rules encoded within the system represent vague samples, and can be viewed as vague prototypes of the training data. A neuro-fuzzy system should not be seen as a kind of (fuzzy) expert system, and it has nothing to do with fuzzy logic in the narrow sense.

In this paper "neuro-fuzzy" has to be understood in the way given by the five points above. Therefore, we consider "neuro-fuzzy" as a certain technique to derive a fuzzy system from data, or to enhance it by learning from examples. The exact implementation or "neuro-fuzzy model" does not matter. It is possible to use a neural network to learn certain parameters of a fuzzy system, like using a self-organizing feature map to find fuzzy rules (cooperative models), or to view a fuzzy system as a special neural network, and directly apply a learning algorithm (hybrid models).

Approaches where neural networks are used to provide inputs for a fuzzy system, or to change the output of a fuzzy system. We prefer to call "neural (network)/fuzzy (system) combinations" or "concurrent neural/fuzzy models" to stress the difference that in these approaches parameters of a fuzzy system are not changed by a learning process. If the creation of a neural network is the main target, it is possible to apply fuzzy techniques to speed up the learning process, or to fuzzify a neural network by the extension principle to be able to process fuzzy inputs. These approaches could be called "fuzzy neural networks" to stress that fuzzy techniques are used to create or enhance neural networks.

Neuro-Fuzzy Classification

Classification of data is an area of application where statistical methods, machine learning and neural networks are thoroughly examined and successfully used. It is also possible to use a fuzzy system for classification with rules like if x_1 is m_1 and x_2 is m_2 and … and x_n is m_n then pattern $(x_1, x_2, …, x_n)$ belongs to class i, where the m_1, ..., m_n are fuzzy sets.

What is the advantage of having another method for classification? A fuzzy classifier is not a replacement for the methods yielding better results, but a different way of achieving the same goal. If a decision is made for a fuzzy classifier usually, the following advantages are considered:

– vague knowledge can be used,
– the classifier is interpretable in form of linguistic rules,
– from an application view the classifier is easy to implement, to use and to understand.

The rule base of a fuzzy classifier that uses rules like mentioned above represents an approximation of an (unknown) function. Because of the inferences process, the rule base actually does not approximate Φ but the function $\Phi' : R^n \rightarrow [0; 1]^m$. We can obtain Φ (x) by $\Phi (x) = \psi (\Phi' (x))$, where ψ reflects the interpretation of the classification result obtained from the fuzzy classifier. Usually the class with the largest activation value is chosen.

Classifiers are usually derived from data and are not specified directly. In case of a fuzzy classifier, there are two common methods:

– fuzzy clustering, and
– neuro-fuzzy learning

In the case of fuzzy clustering, the input space is searched for clusters. The number of clusters is determined by an evaluation measure and the size and shape of the clusters is given by the cluster algorithm. The obtained clusters can function as a fuzzy classifier; in fact, there is no need to express the classifier by rules. However, the interpretability is lost, and therefore fuzzy rules are sometimes created by projection of clusters.

Fuzzy rules obtained by projection of cluster suffer from a loss of information. A fuzzy rule does not represent the cluster exactly, but only the smallest encompassing hyperbox. The performance of a fuzzy classifier obtained by clustering is therefore usually reduced, once it is expressed in form of fuzzy rules. In addition, the rules are often hard to interpret, because the resulting fuzzy sets can have almost any shape.

Another method to obtain a fuzzy classifier from data is to use a neuro-fuzzy approach. This means the classifier is created from data by a heuristic learning procedure. If the neuro-fuzzy approach meets the five points listed above, then the interpretability of the resulting classifier, and a acceptable performance might be obtained. A neuro-fuzzy approach is also often computationally less expensive than a clustering approach, because of its simplicity.

A neuro-fuzzy classifier is nothing more than a fuzzy classifier obtained by a learning procedure. There are several known approaches to find a fuzzy classifier. We developed a “TEXPERT NEUROFUZZY CLASSER”. After creation of the classifier there is usually no hint on how it was derived. Therefore, the term “neuro-fuzzy” strictly only applies to the creation or training phase. Afterwards, once the classifier is applied and not changed further, there remains a simple fuzzy system for classification. However the term “neuro-fuzzy classifier” is usually kept, to stress the mode of obtaining the classifier. The interpretation of the classifier is also often illustrated by representing it in a neural network structure like e.g. “TEXPERT NEUROFUZZY CLASSER” in Fig. 1

Neuro-Fuzzy Learning

The rule learning algorithm needs an initial fuzzy partitioning for each variable. This is given by a fixed number of equally distributed triangular membership functions (Fig. 2). The combination of the fuzzy set forms a “grid” in the data space i.e. equally distributed overlapping rectangular clusters. Then the training data is processed, and those clusters that cover areas where data is located are added as rules into the rule base of the classifier. In a next step, this rule base is reduced by just keeping the best performing rules. The result after this stage of training can e.g. look like the situation in Fig. 2.

After the rule base has been created, the membership functions are tuned by a simple heuristic. For each rule a classification error is determined, and used to modify that membership function that is responsible for the rule activation (i.e. delivers the minimal membership degree of all fuzzy sets in the rules antecedent). The modification results in shifting the fuzzy set and enlarging or reducing its support, such that a larger or smaller membership degree is obtained depending on the current error.

The learning result might look like the situation in Fig. 3.

To obtain an interpretable classifier some restrictions can be specified by the user. The “TEXPERT NEUROFUZZY CLASSER” software allows impose the following restrictions on the learning algorithm:

- a membership function must not pass on of its neighbors,
- a membership function may be asymmetrical,
- membership functions must intersect at 0.5.

To further enhance the learning capabilities, and to obtain classifiers that can be interpreted more easily, we introduce some new concepts for the “TEXPERT NEUROFUZZY CLASSER” learning algorithm:

- Re-learning of the rule base: If the error cannot be further reduced by modifying the membership functions, the rule base can be learned anew. The “grid” given by the fuzzy sets is “distorted” during learning. It can be possible that restarting the creation and evaluation of the rule base leads to new and better rules, not considered before. The current rules are used as prior knowledge and are compared to newly created rules (if any).
- Rule pruning: To reduce the rule base, rule with poor performance, and rules that cover data that is also covered by other rules, can be deleted.
- Variable pruning: For each rule it is checked whether there are variables which never (or rarely) supply the minimal membership degree of all variables of the rule's antecedent. These variables can be deleted from the antecedent of the considered rule.

In the following section, we present an example that shows the capabilities of our neuro-fuzzy learning strategy, in Cotton Spinning Technology.

Neuro Fuzzy Classification in Cotton Spinning - An Application

As an example for the learning capabilities of “TEXPERT NEUROFUZZY CLASSER”, we use a “Carded-Combed-Open End Yarns” data set.

The data set contains 524 cases distributed into three classes (carded yarns, combed yarns and open-end yarns). We used only 504 cases (254 for training, 250 for testing), because 20 cases have missing values.

Each pattern has nine features (the Count Variation – CV Ne, the Breaking Tenacity - TENACITY, the Breaking Elongation - ELONGATION, the Work of Rupture – WORK TO BREAK, the Coefficient of Variation of Yarn Mass - CV_USTER, the Thin Places per 1000 m - THIN, the Thick Places per 1000 m - THICKS, the Neps per 1000 m - NEPS, the Hairiness - HAIRINESS).

The input features may be real or integer values. The classes must be coded as binary vectors.

The form vector represents specific yarn characteristics linked to the yarn type. In this way the form vector - V_F - is represented by:

V_F = {CV Ne, TENACITY, ELONGATION, WORK TO BREAK, CV_USTER, THIN, THICKS, NEPS, HAIRINESS}

The operation of fuzzyfication consists on the numeric/symbolic conversion of the different components conditioning the yarn quality indicators. Each symbol is characterizes by a linguistic term defined by a membership function m(x) of the form vector to a given class. A human expert according to statistical data defines the parameters characterizing the form vector components.

The establishment of the rules is tributary of the human expertise in the field. It is however evident that the high number of output fuzzy classes enables the establishment of an optimal spreading of the value.

To show how “TEXPERT NEUROFUZZY CLASSER” performs when prior knowledge is supplied, we used a fuzzy clustering method to obtain fuzzy rules. Fuzzy clustering discovered three clusters that were interpreted as fuzzy rules by projecting the clusters to each dimension and finding trapezoidal membership functions that closely matched the projections. The membership functions were interpreted by small, medium and large resulting in three rules that caused 94 classification errors overall data set:

R_1 : if (m,l,l,l,s,s,s,s,m) then combed yarn,
R_2 : if (l,m,s,m,l,m,l,l,l) then carded yarn,
R_3 : if (s,s,l,s,m,l,m,m,s) then open-end yarn.

When we initialize the “NEUROFUZZY CLASSER” system with these three rules, we at first get 240 classification errors. However, after 70 epochs of training, using the constraint, that fuzzy sets must not pass each other, we obtained a result of only 40 errors altogether (92.1% correct).

A neuro-fuzzy learning strategy is a tool to support the creation of a fuzzy classifier, but not to completely automate it. This means the user should supervise the learning process, and interpret the result.

By analyzing the fuzzy sets obtained by training the three rules used as prior knowledge, we found that the fuzzy set medium substantially overlapped with the fuzzy set large for some variables.

Therefore we again trained the “TEXPERT NEUROFUZZY CLASSER” system with “best per class” rule learning, allowing it to create new rules This time we only used two fuzzy sets to partition the domains of each variable. After 100 epochs of training “TEXPERT NEUROFUZZY CLASSER” made only 22 errors on the complete set (95.6% correct).

It has found the rules

R_1 : if (m,l,m,l,s,s,s,s,m) then combed yarn,
R_2 : if (l,m,s,m,l,m,l,l,l) then carded yarn,
R_3 : if (s,s,l,s,m,l,m,m,s) then open-end yarn.

Conclusions

We have discussed our notion of a neuro-fuzzy system in the context of fuzzy classification. We consider a neuro-fuzzy method to be a tool for creating fuzzy systems from data. The learning algorithm should consider the semantics of the desired fuzzy system, and adhere to certain constraints.

The learning result should also be interpreted, and the insights gained by this should be used to restart the learning procedure to obtain better results if necessary.

A fuzzy classifier, especially a neuro-fuzzy classifier, is only used, when interpretation and the employment of (vague) prior knowledge are required.

Fuzzy classification is not a replacement, but an addition to other methods like statistics or neural networks. The price for the interpretation of the classifier in form of simple fuzzy rules, and for the simple and computationally efficient learning algorithm might be paid by a classification result that is not as good as it could be if other methods are used.

The example with the “Carded-Combed-Open End Yarns” data shows that “TEXPERT NEUROFUZZY CLASSER” can be used as an interactive data analysis method. It is useful to provide prior knowledge when it is possible. A combination with fuzzy clustering can help here.

The learning result of “NEUROFUZZY CLASSER” should be analyzed, and the obtained information can be used for another run that yields an even better result.

References

1. Detlef Nauck and Rudolf Kruse, “What are Neuro-Fuzzy Classifiers?”, Proc. Seventh International Fuzzy Systems, Prague, 1996.
2. P. M. Frank, “Application of Fuzzy Logic to Process and Fault Diagnosis”, IFAC Symposium, Finland, 1994.
3. B. Hansruedi, “Réglage par la Logique Floue”, Presses Polytecniques, Paris, 1996.
4. M. E. Cabeco-Silva, A. A. Cabeco-Silva, N. B. Nasrallah and J. L. Samarao, "Quality Engineering in Combed Cotton Yarns Processing Using Neural Networks", 1997 Beltwide Cotton Conferences, Cotton Textile Processing Research Conference, New Orleans, Louisiana, USA, January, 1997.

5. A. A. Cabeco-Silva, M. E. Cabeco-Silva, N. B. Nasrallah, J. L. Samarao, "Using Fuzzy Logic in Cotton Processing", 1997 Beltwide Cotton Conferences, Cotton Textile Processing Research Conference, New Orleans, Louisiana, USA, January, 1997.

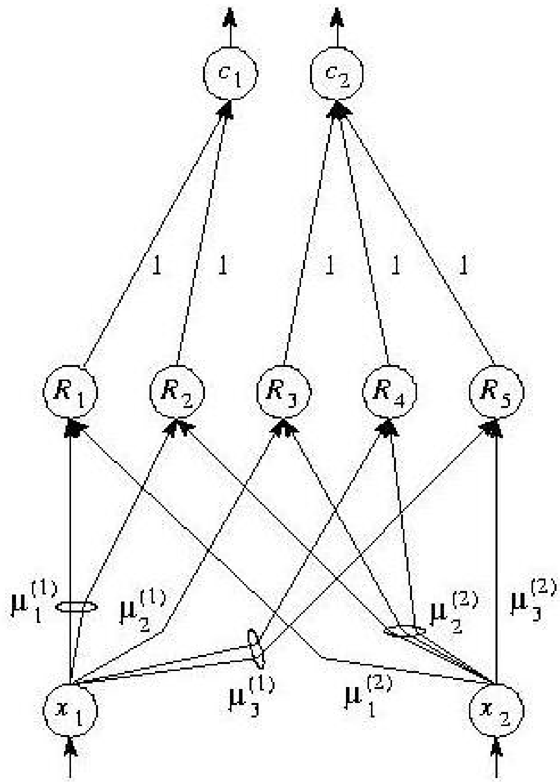

Figure 1: A TEXPERT NEURO-FUZZY CLASSER system represented as a 3 layers feedforward neural network

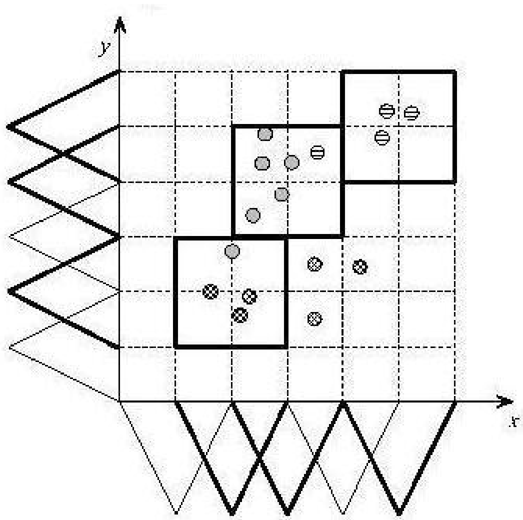

Figure 2: Situation after 3 fuzzy classification rules has been created using initial membership functions

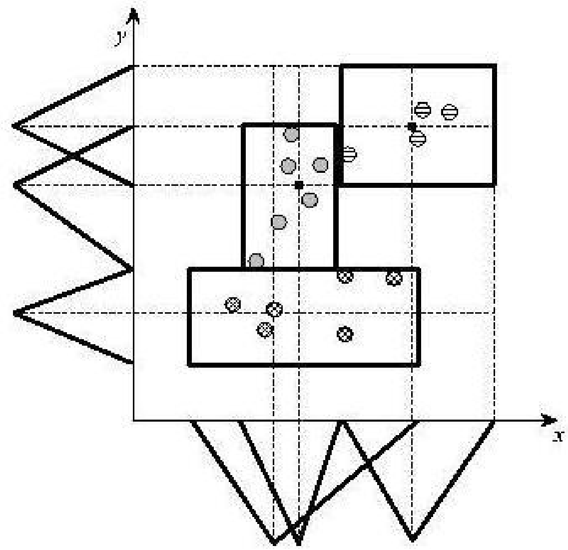

Figure 3: Situation after training the classifier, i.e. modifying the membership functions