Nykänen, O. (2006). Inducing Fuzzy Models for Student Classification. *Educational Technology & Society*, 9 (2), 223-234.

Inducing Fuzzy Models for Student Classification

Ossi Nykänen
Senior Researcher, Tampere University of Technology, Digital Media Institute
Hypermedia Laboratory, P.O.Box 553, FI-33101 Tampere, Finland
Tel: +358 3 3115 3544
ossi.nykanen@tut.fi

ABSTRACT

We report an approach for implementing predictive fuzzy systems that manage capturing both the imprecision of the empirically induced classifications and the imprecision of the intuitive linguistic expressions via the extensive use of fuzzy sets. From end-users' point of view, the approach enables encapsulating the technical details of the underlying information system in terms of an intuitive linguistic interface. We describe a novel technical syntax of fuzzy descriptions and expressions, and outline the related systems of fuzzy linguistic queries and rules. To illustrate the method, we describe it in terms of a concrete educational user modelling application. We report experiments with two data sets, describing the records of the students attending to a university mathematics course in 2003 and 2004. In brief, we aim identifying the failing students of the year 2004, and develop a procedure for empirically inducing and assigning each student a fuzzy property "poor", which helps capturing the students needing extra assistance. In the educational context, the approach enables the construction of applications exploiting simple and intuitive student models, that to certain extent are self-evident.

Keywords

Fuzzy systems, Student modelling, Linguistic interfaces, Fuzzy queries and rules

Introduction

Modern information management systems enable the recording and the management of data using sophisticated data models and a rich set of management tools. In the context of educational systems, the information typically includes details about learning material, the tasks and the objectives, the course information, the contact information, the teacher and the student profiles, and the information related to student assignments, the tests, the grades, and other records. When perceived in the technical terms of the underlying information system, it soon becomes very difficult to manage, integrate, and access different kinds of information. In this article, we seek means to model the imprecision of information and simplify the access to information systems, in terms of fuzzy modelling.

In brief, computer-based learning management systems seek to be equivalent, if not superior, to the traditional learning systems (Darbhamulla & Lawhead, 2004). However, an increased technological potential does not automatically mean better applications. For instance, while the standardisation of the application interfaces (see, e.g., (Dodds, 2001)) certainly improves the development of learning systems, and it is possible to semantically integrate different sources of information, say, by using standard general-purpose query languages (Boag et al. 2005; Prud'hommeaux & Seaborne, 2005), the engagement with the technical details may itself become an obstacle. A data format and a technical vocabulary that suits the needs of technical storage, does not necessarily suit the needs of the developers of the more abstract learning applications, not to mention the conceptualisations of the teachers and the students. In particularly, while it is relatively easy to store information about students' progress technically and to classify assignments as "easy", "intermediate", or "difficult" by hand, it is surprisingly difficult to automate the process of classifying students with respect to these semantic labels in terms of crisp computing. Assuming a good teacher and only a handful of students, this problem is largely irrelevant. However, when the number of the students increases or they are out of reach, seeking computer-supported means becomes interesting. In addition, the ability to capture the key domain objects using intuitive expressions not only supports the activities of the teacher, but also makes it easier to implement and integrate learning applications based on domain-specific design.

It is instructional to consider this problem from the perspective of domain, task, and user modelling (see, e.g., (Brusilovsky & Cooper, 2002)). In general, computer-aided teaching, learning, and collaboration systems (as information systems) enable the construction of various kinds of decision-support systems that help organising courses and adapting educational content to suit the needs of different kinds of students. At the heart of such systems lies a student (user) model, that records and explains the progress of the students, to be exploited by the learning environment. The basic requirement of student modelling is the ability to assign individuals and user groups meaningful labels related to their characteristics and activities, to be exploited on the level of the reacting application. Historically, the development of modelling has evolved from crisp to fuzzy models, eventually taking context explicitly into account (Jameson 1995; Jameson 2001). Nevertheless, a good user model is intuitive and simple, allows sharing domain knowledge, and matches the requirements of the modelling task.

The introduction of fuzziness typically aims managing imprecision (or vagueness) in applications. In short, the existing fuzzy systems fall into two categories. While fuzzy logic or *fuzzy expert systems* consider fuzziness in terms of fuzzy implication and a generalisation of crisp (two or finitely-valued) logic, *fuzzy control systems* aim reproducing the behaviour of intuitive control rule groups by exploiting fuzzy models, i.e., computing with fuzzy sets (Turunen, 1999; Jang & Sun, 1997; Hüllermeier, 2005; Cox, 2005). Further, the fact that fuzzy membership functions appear often overly precise in applications has motivated the development and application of, e.g., interval-valued and type-2 fuzzy sets (Vaucheret, 2002; Mendel, 2001). In the context of educational applications, the reported soft computing experiments include applications of Bayesian and Neuro-Fuzzy methods (Jameson, 1995; Xenos, 2004; Stathacopoulou et al., 2005). These methods have been widely applied, taking many aspects of education into account. The reported applications notably demonstrate analysing and evaluating both students based on individual assessments, students' views, education quality, grades of journals, and the performance of entire academic departments (Ma & Zhou, 2000; Liu & Maes, 2004; Suarez, 2003; Nokelainen et al., 2001; Arriaga et al., 2005; Ibrahim, 2001; Turban et al., 2000; Lopes & Lanzer, 2002). The claimed advantages of fuzzy modelling range from faithfully evaluating students' learning and cognitive abilities to moving towards personalised education (Stathacopoulou et al. 1999; Weon & Kim, 2001; Kavcic et al., 2003).

In this article, we report an approach of implementing a predictive fuzzy system that manages capturing both the imprecision of the empirically induced classifications and the imprecision of the intuitive linguistic expressions via the extensive use of fuzzy sets. We describe a novel technical syntax of fuzzy descriptions and expressions, and outline the related systems of fuzzy linguistic queries and rules. In the educational context, the approach provides a concrete way of inducing intuitive semantic labels from the existing data archives, suitable for capturing the domain objects (users, user groups, processes, tasks, and artefacts) using fuzzy linguistic expressions. The basic idea is to describe the domain objects and the associated linguistic expressions in terms of neutral fuzzy sets, subject of fuzzy queries and rules. For the benefit of a concrete discussion, we describe implementing a novel predictive fuzzy system that demonstrates fuzzy modelling in the context of a realistic application. We report experiments working with two data sets, describing the records of the students attending to a particular university course of mathematics in 2003 and 2004 at the Tampere University of Technology, Finland. In short, we aim predicting the poor students based on the archives, and semantically integrate this information with fuzzy linguistic rules and queries.

The main contribution of this article is threefold: First, we introduce a design of a realistic predictive fuzzy system that provides a basis for semantically integrating and encapsulating various kinds of information systems. Second, we develop the syntax of fuzzy expressions by introducing the notions of fuzzy linguistic constructors and pure linguistic concepts, and describe the related rule and query systems. Third, we provide a concrete bridge between the two complementary aspects of fuzziness in educational design, by syntactically integrating the imprecision due to lack of information (prediction accuracy) with the imprecision due to use of informal language (fuzzy linguistic expressions).

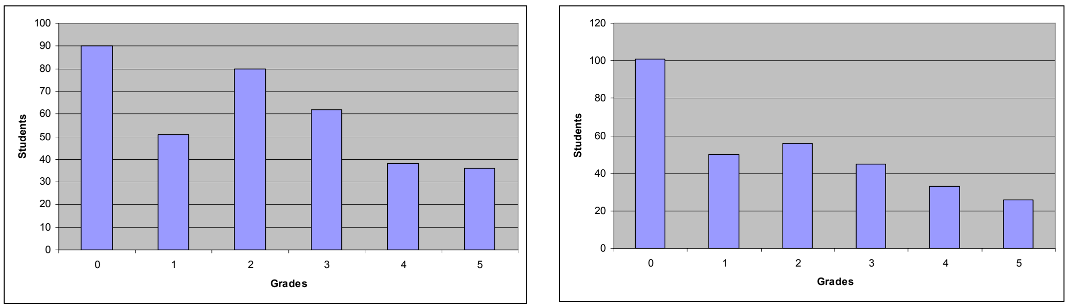
The rest of this article is organised as follows: Section 2 describes the empirical data sets and Section 3 demonstrates the potential of predicting the failing students. Section 4 presents the design of predictive fuzzy systems and Section 5 outlines the definitions related to fuzzy queries and rules. Section 6 briefly considers the applicability of model-free methods, and finally, Section 7 concludes the article with few notes.

Empirical Data

The success user modelling applications is determined by the quality of the user data. In the context of learning management applications, the backbone of modelling may be economically compiled from the formally recorded achievements. Our experiments were conducted using the accumulated data sets from the well-established basic course *Engineering Mathematics I*, during two successive years 2003 and 2004.

The data describes the performance of the individual students in terms of 22 attributes: points from 13 assignments (0-10, 'x' denotes a missing value), four mid-term exams (two attributes of 0-6 points each, 'x' denotes a missing value), and the final grade (0-5). The final grades were compiled using the attributes and scaled according to the final statistics. Compiled into chronological order, this approach provides a vector of 22 attributes for each student ([a_1, a_2, a_3, e_1, e_2, a_4, a_5, a_6, a_7, e_3, e_4, a_8, a_9, a_{10}, e_5, e_6, a_{11}, a_{12}, a_{13}, e_7, e_8, g]). Considering ongoing courses, the attribute vectors become available gradually.

The data includes 358 student instances considering the year 2003, and 311 instances considering the year 2004. This provides two data sets $D^{2003} = [\alpha_{ij}]$, i = 1, 2, ,..., 358, j = 1, 2, ..., 21, and $D^{2004} = [\beta_{ij}]$, i = 1, 2, ,..., 311, j = 1, 2, ..., 21, accompanied with the final grades [g_i], i = 1, 2, ,..., 358, and [h_i], i = 1, 2, ,..., 311, respectively. We can use this information to define a training data set for the concept *failing student*, [g'_i], where $g'_i = 1$ when $g_i = 0$ and $g'_i = 0$ otherwise (define [h'_i] accordingly). The training data sets [$\alpha_{ij}|g'_i$] and [$\beta_{ij}|h'_i$] may now be used for recognising the failing students. In particularly, the construction of the data sets enable using the first data set D^{2003} as training data for predicting the second data set D^{2004}.



(a) (b)

Figure 1. The grade histograms of 2003 and 2004

Figures 1a and 1b depict the grade frequencies of the years 2003 and 2004. Of course, the data from the year 2003 does not completely characterise the year 2004, even if we assume that the course format remained unchanged. In principle, *any* change in the course agenda makes it hard to use the data from the previous year for the benefit of the next year. Changes might notably include (and partly did include) change of topics, providing extra tutoring for the poorly performing students, extra tests and questionnaires, and simply showing statistics like Figure 2 to the students (please see below).

Using Training Data in Applications: Making Predictions

Considering applications that rely on user modelling in the educational domain, perhaps the most important use case is classifying the students based on their observed and expected performance. This yields categories of fuzzy user groups providing decision-support information for various applications. For instance, consider the expression "student is very poor" in the context of the following informal rule as a part of the adapting application:

If a <u>student is very poor</u> and lives far from the campus, 1) *then the need for extra attention is high.*

In order to establish rules, the related fuzzy terms must be defined. In this article, we assign students (as identifiable domain objects) fuzzy descriptions which are to be associated with fuzzy linguistic constructors (as a sort of hedges) in rules and queries. Let us first briefly consider the predictive component of the process of inducing fuzzy descriptions.

In our application, it is possible to establish a straightforward procedure for recognising the failing students using a simple domain-specific design, simply by assuming average performance: A student fails the course if she does not accomplish at least 40% of the assignments or gets less than 24 points from the mid-term exams (the point limit may decrease based on the final statistics). Additional requirements include attending to at least four computer exercises and completing a preliminary test. The additional requirements were not used in the predictions in this study. Accompanied by the fact that the grades were scaled based on the final statistics, this actually makes the prediction problem somewhat harder.

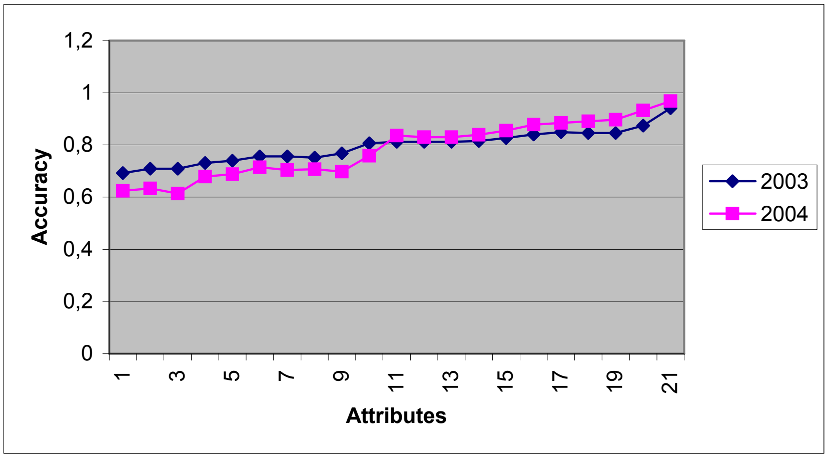


Figure 2. Prediction accuracy of the years 2003 and 2004

Figure 2 depicts a demonstration of using the series of the first 1, 2, ..., 21 attributes of D^{03} and D^{04} for predicting the concept failing student by simply assuming average performance. In the experiment, the data was divided into k sets, according to the number of the attributes used in the prediction. The experiment shows that a naïve classifier, relying upon domain-specific information and simple design insight, manages to predict, or recognise, the failing students with a relatively high accuracy.

Informally speaking, Figure 2 provides a clear message for the students struggling to pass courses. When grades are considered, failing performance can be reasonably well recognised after mid-course. Of course, this data alone can not tell what to do in order to improve the situation, expect for the "work harder" signal. In addition, when separated from the more decisive contextual characteristics, the recorded points and the mid-term exams do not necessarily faithfully reflect the students' skills nor understanding. Considering fuzzy systems, however, the experiments seem to support the rationale of implementing predictive fuzzy models, since the trend of the prediction accuracy is clearly positive.

Implementing a Predictive Fuzzy System

Assume we wish to assert rules and perform queries on top of a database, using fuzzy linguistic expressions. For instance, a teacher might want to establish heuristic rules like (1) and search for the students needing extra attention. Further, assume all the information is not readily available and we have to rely on generalisations learned from the historical archives. For instance, the points from assignments become available only during courses. In practice, we face this task during the internal design of an adapting educational system.

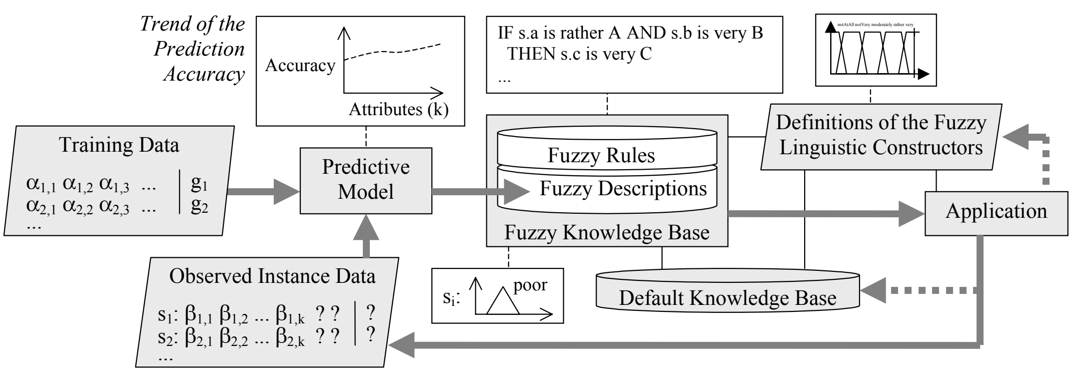


Figure 3. The basic components of a predictive fuzzy system

Figure 3 depicts the basic components of a predictive fuzzy system. At the heart of the systems lies the predictive model which assigns domain objects (e.g. students s_i) fuzzy descriptions, based on the available training data and the observed instance data from the application. The fuzzy descriptions add to the default knowledge base (e.g. a default thesaurus defining the basic concept taxonomies and the terminological synonyms), providing a basis for interpreting the fuzzy rules and queries. The linguistic component is defined by identifying the linguistic concepts to which the descriptions apply (e.g. *poor*), and establishing the fuzzy linguistic constructors (e.g. *rather*), to be used in the rules and queries. In brief, applications use the predictive fuzzy system for performing fuzzy inference and making fuzzy linguistic queries. This might also include, e.g., interpreting fuzzy SQL queries (see, e.g., (Cox, 2005)). More complex applications might in addition explicitly control the default knowledge base and the definitions of the fuzzy linguistic constructors, in terms of *bootstrapping* the predictive fuzzy system with application-specific defaults (e.g., modifying the fuzzy linguistic constructors upon context).

Let us next consider the task of establishing the *fuzzy linguistic expressions* for fuzzy linguistic rules and queries. For technical purposes, let us equate, in the context of students attending a course, the concepts *failing* and *very poor*. Effectively, this provides a fuzzified notion that enables capturing the students that are "close to" failing. Turned around, a student that passes courses with (nearly) maximal grades is *not at all poor*.

Technically speaking, we would thus like to assert, e.g., the following kinds of expressions concerning students: T = { *not at all poor, not very poor, moderately poor, rather poor, very poor* }. In a typical fuzzy application, these labels would be modelled in terms of a *linguistic variable*, using the elements of T as the *term set* (see, e.g., (Jang & Sun, 1997; Mendel, 2001; Cox, 2005)). Alternatively, however, the expressions include internal structure: a *constructor* part, and a *concept* part. The constructor resembles the notion of a hedge or a modifier while the concept part identifies the (pure) linguistic concept that defines the *type* of the particular expression. For instance, instead of asserting

s is RatherPoor, (2)

we may assert

s isRather Poor. (3)

The subtle syntactical difference becomes noteworthy when considering the constructors in the context of crisp classes. Two practical consequences are particularly significant: First, when compared to (2), separating the constructor and the concept part in (3) reduces the number of the semantic labels required in applications since the constructors may be associated with several concepts. Second, since the shape of the associated fuzzy set does not have to follow the limitations of simple functional hedges (e.g., concentration for "very"), the constructors may be defined empirically. In particularly, this includes the case of taking carefully into account the actual use of particular semantic labels in applications (about survey-based terms, see Mendel (2001)).

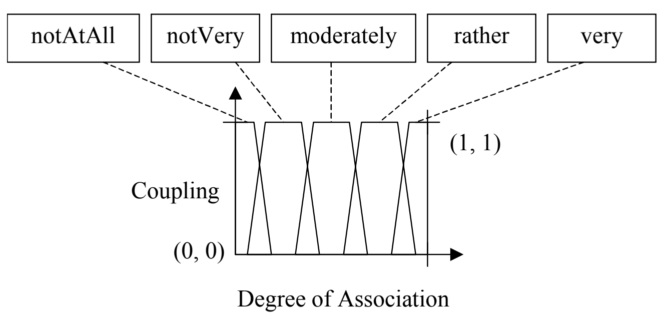


Figure 4. A fuzzy linguistic constructor sequence

Figure 4 depicts a *fuzzy linguistic constructor* sequence that models the constructor terms. In short, a constructor may be perceived as a predicate that assigns a relationship between an subject and an object (see (3)), in terms of a (type-2) fuzzy set. This provides a proper generalisation of the *is-a* relationship or a crisp membership function and is suitable for describing the relationship between *any* domain objects and the selected *linguistic concepts* (e.g., students and the concept *poor*). Using fuzzy sets in modelling thus allows capturing imprecise descriptions which establishes an extension to description logic (see (Baader et al., 2002)). A singleton fuzzy set coincides with the plain (scalar) degree of fuzzy membership while fuzzy sets of other shapes enable modelling, e.g., the imprecision of the fuzzy definition itself. It is worth emphasising that while in this article we use fuzzy linguistic expressions in the syntax of the rules and the queries, they also enable recording imprecise assertions as well-established statements. However, the considerations related to the *assertion context* are not trivial and are not discussed here.

Let us then address the question of adequately modelling the semantic descriptions of the domain objects in the context of the course application. In brief, we wish to assign students a fuzzy property *poor*, using a predictive model. We do this by assigning each student identifier a fuzzy description that associates the identifier with the linguistic concept *poor* in terms of a type-1 fuzzy set (see Figure 5 below). In queries and rules, we may then compute with these properties using the related term *poor* (establishing the type of the expression) and a linguistic constructor, e.g., "moderately" in expressions (establishing the meaning of the constructor term as a fuzzy set).

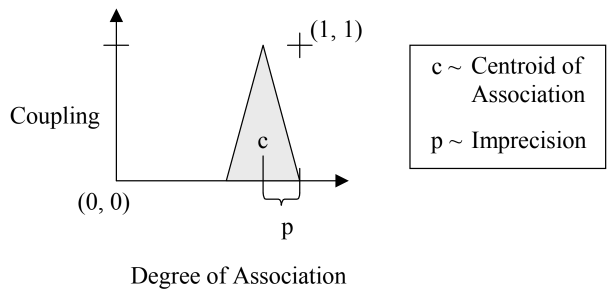


Figure 5. A fuzzy model for a student being poor

We say that the fuzzy set depicted in Figure 5 is *neutral*. This is due to the fact that the domain of the membership function is normalised to [0, 1] and is not assigned any particular datatype information. Datatype information, however, might be required for establishing the definition via *bindings* to explicit quantitative data (e.g., predicted points of a course, distance in meters, etc.). Neutral fuzzy sets provide an intuitive explanation of the concepts *imprecision* (or expectancy, modelled by p), *association* (centroid of which is denoted by c), *coupling* (the max height of the membership function), *precise*, and *crisp*. In particularly, a type-1 fuzzy set is called *precise* if it is defined by a singleton membership function (p=0). Precise fuzzy sets for which c = 0 or c = 1 are called *crisp*. Note that the definitions of association and coupling are not traditionally introduced in terms of type-1 fuzzy sets; they become meaningful only when interpreted in the context of the following rule and query semantics (or in the context of type-2 fuzzy sets).

The experiment yielding Figure 2 provides a basis for modelling students being poor in terms of two observations. First, the predicted or the projected performance enables the definition of an appropriate semantic label in terms of a recognition task. Second, the precision of the assigned semantic labels is not a constant: it changes according to the available information or according to the estimated accuracy of the predictions. As a consequence, we may capture students throughout courses with a fixed set of semantic labels that get more precise as more information becomes available. In other words, the fuzziness of the application to certain extent decreases in time.

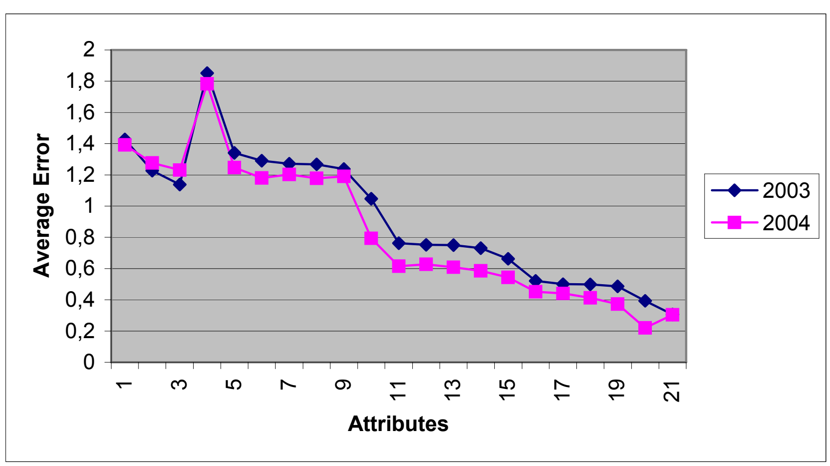


Figure 6. The average error of predicting the exact grades

It seems obvious that we may try to categorise students directly by predicting the grades. As suspected, assuming average performance provides a concrete method for doing this. Figure 6 demonstrates an example of trying to predict the exact grades. The average errors from using the data of 2003 and 2004 follow a similar pattern. It seems that the difference can be explained by the statistical corrections of the grades. After the second mid-term exam, the grades can be predicted with the average error of 0.8. Indeed, the experiments (see Figures 2 and 6) seem to demonstrate that the naïve prediction algorithm provides systematic results and that we may use the archives for estimating the prediction errors reliably.

Considering the application, we may describe the students of the year 2004 using the fuzzy property *poor*, based on the above definitions. Thus, as the course proceeds, we receive more information about the students (in terms of attributes) and may classify the students more accurately. In brief, the application involves two kinds of imprecision. The first flavour of imprecision is due *lack of information*. The second is due *choice*. Roughly speaking, a student may change her projected trajectory by studying harder (or less harder). However, when considering getting a grade via the mid-term exams, the window of influence obviously gets narrower as the course proceeds.

We will next describe a simple algorithm that can be used to associate each student a fuzzy description. It might be helpful to imagine that we assign each student s a triangular neutral fuzzy set $\tilde{A}_s$ of the type *poor*. The shape of the related membership function μ_s can be described in terms of two parameters, centroid of the degree of the association, s.c, and imprecision, s.p: $\mu_s(x) = \mu_s(x; s.c, s.p)$. By letting $u = s.c - s.p - \varepsilon$, $v = s.c$, and $z = s.c + s.p + \varepsilon$, $0 < \varepsilon << 1$, we get the conventional definition (Jang & Sun, 1997):

$$\mu_s(x) = \text{triangle}(x; u, v, z) = \max(\min((x-u)(v-u)^{-1}, (z-x)(z-v)^{-1}), 0), x \in [0, 1]. \quad (4)$$

Let $s.x \in X = \{0, 1, ..., 54\}$ denote the number of points student s will earn from the mid-term exams and the exercises. By doing sufficiently exercises, students may earn max 54 - 48 = 6 bonus points to be added to the results of the mid-term exams. In practice, s.x = s.x(k) where k is the number of the attributes available during the course. Thus, s.x is a projected value, based on observing the performance of the past and assuming an average performance in the future. Define a function $s.c: X \rightarrow [0, 1]$, for modelling the (centroid of the) *degree of association*, associating the student s with the concept *poor*:

$$s.c(s.x) = \max(0, \min(1, (24 - (s.x-24)) / 24)). \quad (5)$$

Let Error(k) denote the error of the prediction of the grades during the previous year(s), using k attributes (see Figure 6). Define a function $s.p: K \rightarrow E \subset \Re$, where $k \in K = \{1, 2, ..., 21\}$ is the number of the attributes available, and E denotes the set of errors, for modelling the imprecision of the association:

$$s.p(k) = \text{Error}(k) / 5. \quad (6)$$

We can now assign each student s a pair $(c, p)_s$ which models her being poor. Figure 6 depicts the development of the imprecision as more information becomes available. Intuitively, the pair $(c, p)_s$ may be interpreted as a neutral type-1 fuzzy set of certain shape (for triangular functions, consider (4) and Figure 5). The shape, however, must be decided by the application designer.

The definitions (5) and (6) establish the induced fuzzy properties. For instance, the fuzzy set depicted in Figure 5 suffices modelling a student s when s.c = 0.8 and s.p = 0.2. Considering the binding, this happens when s.x = 28.8 and Error(k) = 1.0.

Fuzzy descriptions provide intuitive labels of data in terms of the fuzzy property *poor*. As an example, Figure 7 depicts *classification trajectories* describing six particular students that took the year 2004 course with the grades 0, 1, 2, 3, 4, and 5. In brief, the trajectories denote the association of the students being poor, suitable for fuzzy modelling. The imprecision associated with the fuzzy models may be read from the Figure 6, using the errors of the year 2003. In other words, the classifications get rather precise around mid-course. Intuitively, the analysis reveals that the student s3 seems to improve her performance during the course while the student s1 is doing worse and worse as the course proceeds. Of course, more information is needed for explaining *why* the behaviour occurs.

Other fuzzy properties may be modelled and studied in a similar fashion. For instance, a fuzzy property *passive* can be induced from the attendance statistics. Examples of completely crisp properties include the number of credit units, sex, age, etc. Finally, students might also be explicitly asked further attributes for various properties that can help categorising them. For instance, learning style, interest towards the subject, estimated effort, time to travel to the campus, studying at home, and being active or passive in the course. As the user models become more refined, it is possible to model and interpret students' behaviour more faithfully. Note however, that these extra sources of information inevitably introduce new sources of fuzziness as well, in particularly the imprecision related to communication. Nevertheless, using neutral fuzzy sets for modelling these properties provides a well-established reference system for various kinds of applications. The chief benefit is the ability to model a wide range of fuzzy properties in terms of uniform descriptions. From the perspective of fuzzy systems and applications, this helps integrating information from various sources. Associated with the idea of linguistic constructors, the approach provides a concrete way to describe domain objects using a relatively small set of intuitive linguistic concepts and expressions.

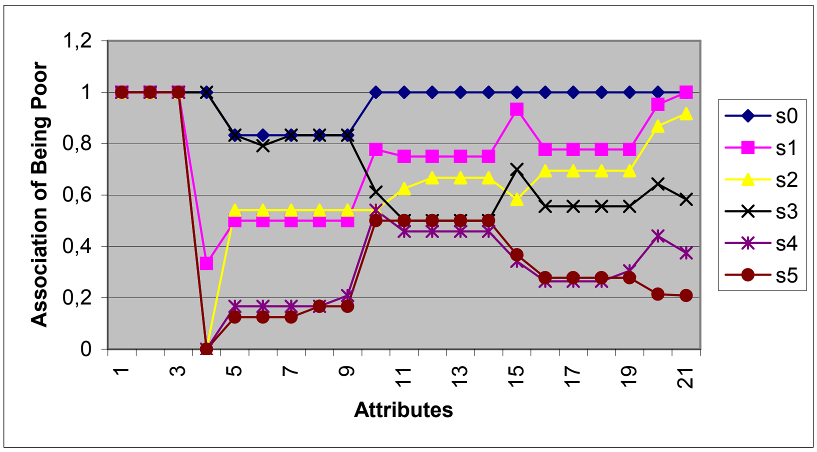


Figure 7. The classification trajectories of five particular students

Notes About Implementing Fuzzy Linguistic Queries and Rules

In the previous section, we demonstrated the construction of a predictive fuzzy model yielding a simple knowledge base of fuzzy descriptions. However, the model and the rationale behind the associated technical definitions can not be appreciated unless the related query and the rule applications are outlined as well. In order to illustrate the technical use cases, let us next very briefly reconsider fuzzy properties in the context of linguistic expressions, namely fuzzy linguistic queries and rules (see (1)).

A typical use case of educational decision-support systems might be retrieving a list of students that are likely to fail the course, i.e. a list of poor students as a fuzzy user group. Consider the following informal *fuzzy linguistic query*:

Find students that are at least rather poor and very passive. (7)

The expression (7) can be formalised in terms of conjuncted *match patterns* where match patterns may include an associated *modifier* part (*at least*, *around*, *at most*).

Formally speaking, the expression (7) establishes a sequence of two match patterns $<A_j>$ = <*atLeast(rather)*, *very*> (where, in general, j = 1, 2, .., m) now associated with the fuzzy properties $<p_j>$ = <*poor*, *passive*>. Assuming each student s = 1, 2, ..., n is appropriately associated with the related fuzzy properties $<v_{sj}>$, j = 1, 2, ..., m, we may compute the match for each student and rank the students according to the matches.

Perhaps the most straightforward method for computing fuzzy matches is due to aggregating the firing strengths computed from the matching sequences of the fuzzy sets. Let A_j and v_j denote two neutral fuzzy sets, associated with the fuzzy membership functions $\mu_{Aj}: [0, 1] \rightarrow [0, 1]$ and $\mu_{vj}: [0, 1] \rightarrow [0, 1]$. Define

$$w_j = \max_x \mu_{Aj}(x) \wedge \mu_{vj}(x). \quad (8)$$

This definition demonstrates linearity which is useful in implementations. We say that w_j denotes the *firing strength* of $<A_j, v_j>$ and define the *fuzzy match* of $<A_j, v_j>$ as:

$fMatch(<A_j, v_j>) = \min_j <w_j>$. (9)

In other words, the fuzzy match is a value between 0 and 1, suitable to be used to order the students based on the query. There exists other strategies for computing fuzzy matches as well, e.g., aiming to preserve the aspect of imprecision of the matches. For brevity, we ignore the other definitions.

Fuzzy linguistic queries are useful but do not provide good syntactic means for using the information further. Sometimes it is useful to assign students new fuzzy properties based on the existing ones. For instance, one might wish to assert the following *simple fuzzy linguistic rule*:

IF a student is at least rather poor AND very passive, (10)
THEN she is problematic.

It is easy to see that fuzzy queries establish a special case of (simple linguistic) fuzzy rules. For instance, query (7) might be equivalently modelled in terms of assigning each student a new fuzzy property, *problematic*. The key difference, however, is that the new property *problematic* can be utilised in further queries and rules, to be aggregated with other properties of the same type. With certain restrictions, simple fuzzy linguistic queries provide the semantics of simple fuzzy linguistic rules. This is due to the fact that it is possible to interpret the fuzzy match as a fuzzy set; in our simple example, a fuzzy singleton would do.

In general, more complex fuzzy rules may be implemented in terms of *(control) rule groups*. With certain assumptions, rule groups may be modelled using (slightly modified) Mamdani fuzzy systems (see (Jang & Sun, 1997; Cox, 2005)). The key differences between simple fuzzy linguistic rules and rule groups lie in descriptional complexity and coupling. Rule groups enable the modelling of more complex relationships than simple linguistic rules, but by default, the coupling (max height) of the implied fuzzy properties might decrease during the inference (outputting non-normal fuzzy sets). This causes certain difficulties related to interpretation in applications. For brevity, we will not consider rule groups further in this article.

From Domain-Specific Design to Model-Free Methods

There are several ways to implement predictive fuzzy models. In particularly, even if applying domain-specific design (e.g., assuming average performance) may provide the best results, it is not always feasible nor necessary. In principle, there exists a wide range of machine learning, data mining, and exploration methods suitable for the task (Russel & Norvig, 1995; Mitchell, 1997; Cox, 2005).

We describe elsewhere a study and a method for inducing membership functions of fuzzy sets empirically upon a training set of positive and negative examples (Nykänen, 2004). In short, the study provides a method that generalises concept learning to include learning fuzzy concepts. The basic idea is to train a redundant Decision Tree (DT) to classify data according to a goal attribute that denotes the related crisp concept, eventually representing the positive end of the bipolar fuzzy concept. The membership grades of the instances are then computed from the topology of the tree, based on the depth of the tree and the number of attribute tests required for recognising a negative instance. Assuming the crisp concept can be learned well, the method allows assigning objects fuzzy membership degrees with an information-theoretic explanation.

Let us next briefly describe an approach for implementing a model-free fuzzy system. A decision tree was trained for each j = 1, 2, ..., k data sets, using $[\alpha_{ij} \mid g'_i]$, i = 1, 2, ..., 180 as the training data for predicting $[h'_i]$, i = 1, 2, ,..., 311, based on the instance vector $[\beta_{ij}]$, i = 1, 2, ,..., 311. The available training data was simply split into half, in order to minimise the negative effect of overfitting. Figure 8 depicts an example of learning the concept *failing student* for the year 2004 with a rudimentary ID3 decision tree learning algorithm (for details, please see (Mitchell, 1997)). The experiment shows that even if blind concept learning is indeed possible (using a transparent DT learner), the naïve classification method clearly outperforms it.

Thus, while the performance of recognising the failing students is better than chance, the prediction accuracy does not seem sufficient for inducing good descriptions. The situation might be improved by encoding the data differently, strengthening the algorithm, and introducing appropriate domain heuristics (Murthy, 1997). Of course, the other more powerful concept learning algorithms, such as neural networks or other equally expressive statistical methods, might perform better. It is worth noticing, however, that by definition, no statistical learning algorithm is able to demonstrate positive performance in every domain (Schaffer, 1994).

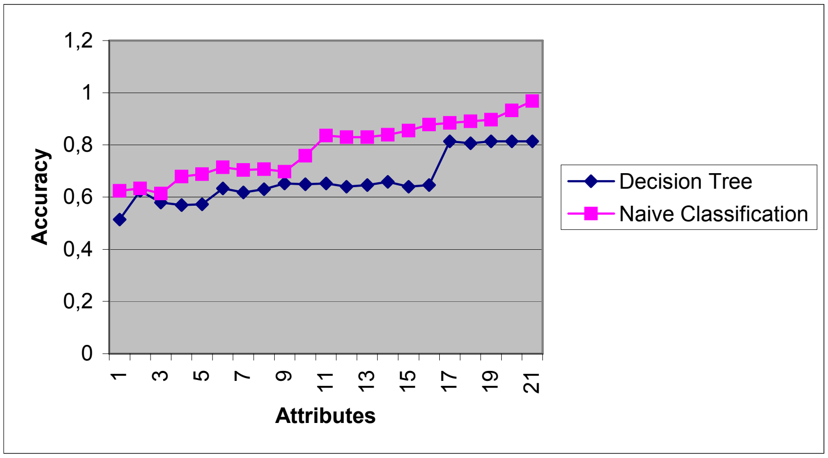


Figure 8. A comparison of the prediction accuracy

Conclusions

We have reported a design and a demonstration of a predictive fuzzy system that enables the description of the domain objects with fuzzy descriptions, suitable for fuzzy linguistic rules and queries. We have discussed the method in the context of an educational application, providing a way to capture students with fuzzy descriptions based on accumulative attribute sequences. We have developed the notion of neutral fuzzy sets, and introduced a system of fuzzy linguistic constructors, to model the structure of fuzzy linguistic expressions in rule and query expressions and logical statements. In brief, this construction relies upon the notion of pure linguistic concepts, providing a fuzzy extension to the relationship.

The design of our system demonstrates two pivotal aspects of fuzziness in applications: While we welcome fuzziness in linguistic expressions because it enables, e.g., writing useful queries using intuitive expressions, we would nevertheless appreciate our knowledge base to be completely precise. For instance, we would appreciate an exact, a priori characterisation of the poor students – which is of course impossible because such data is simply not available.

Motivation for our work is due rather practical considerations. As more information becomes available in applications, it gets difficult to set up user models and formulate expressions solely using the crisp design properties and the technical terms. Developing means to capture domain objects with fuzzy descriptions provides a concrete alternative that helps managing the built-in complexity in applications. Using intuitive semantic labels and expressions in modelling should also help making more transparent systems, thus supporting the activities of the end-users more openly. This process may be perceived as an activity of constructing (fuzzy) *linguistic (end-user) interfaces*. From the end-users' point of view, the main contribution thus lies in the potential of using intuitive expressions, without the need of adopting expert user interfaces, or assuming a strict terminology and in-depth understanding about the encapsulated data model. Note that in order to simplify the discussion we have ignored, e.g., the issues related to contextual interpretation (e.g., bootstrapping of the constructors) and heuristic modelling (selection of the labels) in this treatment.

The quality of our fuzzy system is determined by the (induced) fuzzy model and the rule heuristics. For instance, the early classifications are rather imprecise. However, since this imprecision is transparent, the approach manages to escape several problems related to crisp modelling. In particularly, the system does not force crisp nor overly precise fuzzy classifications, and the "minor" errors in the classifications are not severe, due to fuzzy rules and queries. The approach is quite general and suits the needs of various application domains that require inducing fuzzy classifications and decorating archives with intuitive labels. In fact, we have developed the notion of our predictive fuzzy systems in the context of general-purpose Semantic Web technologies (Nykänen, 2005).

Finally, we wish to emphasise that it is not clear whether the introduced semantic categories should be made publicly available, even if it might support, say, the creation of certain kinds of student groups. Indeed, being "accurate" or not, people might find certain semantic labels uncomfortable or even stigmatising. In turn, this highlights the subtle issues related to user modelling in the fundamentally social context of humans.

Acknowledgements

The author wishes to thank the teachers of the course *Engineering Mathematics I*, in particular Dr. Lasse Vehmanen, for the opportunity to use the anonymous records of the years 2003 and 2004 in this study.

References

Arriaga, F. de, Alami, M. El., & Arriaga, A. (2005). Evaluation of Fuzzy Intelligent Learning Systems. In Méndez-Vilas, A. & González, J. A. M. (Eds.), *Proceedings of m-ICTE2005*, June 7-10, Spain, retrieved November 24, 2005, from http://www.formatex.org/micte2005/200.pdf.

Baader, F., Calvanese, D., McGuiness, D. L., Nardi, D., & Patel-Schneider, P. F. (2002). *Description Logic Handbook*, Cambridge University Press.

Boag, S., Chamberlin, D., Fernández, M. F., Florescu, D., Robie, J., & Siméon, J. (2005) (Eds.) *XQuery 1.0: An XML Query Language*, W3C Working Draft 04 April 2005, retrieved November 24, 2005 from http://www.w3.org/TR/2005/WD-xquery-20050404/.

Brusilovsky, P., & Cooper, D. W. (2002). Domain, task, and user models for an adaptive hypermedia performance support system. In *Proceedings of the 7th International Conference on Intelligent user interfaces*, San Francisco, California, USA: ACM Press, 23-30.

Cox, E. (2005). *Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration*, USA: Elsevier.

Darbhamulla, R., & Lawhead, P. (2004). Paving the way towards an efficient Learning Management System. In Seong-Moo Yoo, S-M. & Etzkorn, L. H. (Eds.), *Proceedings of the 42nd annual Southeast regional conference*, Huntsville, Alabama, USA: ACM Press, 428-433.

Dodds, P. (2001). *Sharable Content Object Reference Model (SCORM^{TM}) Version 1.2: The SCORM Overview*, Advanced Distributed Learning, retrieved November 24, 2005 from http://www.adlnet.org.

Hüllermeier, E. (2005). Fuzzy methods in machine learning and data mining: Status and prospects. *Fuzzy Sets and Systems, 156* (3), 387-406.

Ibrahim, A. M. (2001). Assessment of Distance Education Quality Using Fuzzy Sets Model. In *Proceedings of the SEFI Annual Conference*, Copenhagen, 12-14 September, 2001 (in CD).

Jameson, A. (1995). Numerical Uncertainty Management in User and Student Modeling: An Overview of Systems and Issues, *User Modeling and User-Adapted Interaction, 5* (3-4), 193-251.

Jameson, A. (2001). Modelling both the Context and the User, *Personal and Ubiquitous Computing, 5* (1), 29-33.

Jang, J-S. R., & Sun, E. M. (1997). *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, USA: Prentice Hall.

Kavcic, A., Pedraza-Jimenez, R., Molina-Bulla, H., Valverde-Albacete, F. J., Cid-Sueiro, J., & Navia-Vazquez, A. (2003). Student Modelling Based on Fuzzy Inference Mechanisms. In Zajc, B. & Tkalcic, M. (Eds.), *Proceedings of the EUROCON 2003*, Ljubljana, Slovenia, 379-383.

Liu, H., & Maes, P. (2004). What would they think? a computational model of attitudes. In Vanderdonckt, J., Nunes, N. J. & Rich, C. (Eds.), *Proceedings of the 9th International Conference on Intelligent User Interfaces*, Funchal, Madeira, Portugal, USA: ACM Press, 38-45.

Lopes, A. L. M., & Lanzer, E. A. (2002). Data envelopment analysis - DEA and fuzzy sets to assess the performance of academic departments: a case study at Federal University of Santa Catarina - UFSC, *Pesquisa Operacional, 22* (2), 217-230.

Ma. J., & Zhou, D. (2000). Fuzzy Set Approach to the Assessment of Student-Centered Learning, *IEEE Transactions on Education, 43* (2), May 2000, 237-241.

Mendel, J. M. (2001). *Uncertain Rule-Based Logic Systems: Introduction and New Directions*, USA: Prentice Hall.

Mitchell, T. (1997). *Machine Learning*, USA: McGraw-Hill.

Murthy, S. K. (1997). On Growing Better Decision Trees from Data. *Doctoral dissertation*, University of Maryland, retrieved June 21, 2005 from http://www.cbcb.umd.edu/~salzberg/docs/murthy_thesis/thesis.ps.gz.

Nokelainen, P., Silander, T., Tirri, H., Nevgi, A., & Tirri, K. (2001). Modeling Students' Views on the Advantages of Web-Based Learning with Bayesian Networks. In Ruokamo, H., Nykänen, O., Pohjolainen, S. & Hietala, P. (Eds.), *Proceedings of the 10th International PEG Conference*, June 23-26, 2001, Tampere, Finland, 101-108.

Nykänen, O. (2004). Fuzziness as a Recognition Problem: Using Decision Tree Learning Algorithms for Inducing Fuzzy Membership Functions. In Zanasi, A., Ebecken, N. F. F. & Brebbia, C. A. (Eds.), *Data Mining V*, UK: WIT Press, 143-153.

Nykänen, O. (2005). Implementing Semantic Web Agents that Learn Upon Experience: A Machine Learning Module for the CWM Rule System. In Guimaraes, N. & Isaias, P. (Eds.), *Proceedings of the IADIS International Conference on Applied Computing*, 22-25 February, 2005 Algarve, Portugal, 409-416.

Prud'hommeaux, E., & Seaborne, A. (2005). *SPARQL Query Language for RDF*, W3C Working Draft 19 April 2005, retrieved November 24, 2005 from http://www.w3.org/TR/2005/WD-rdf-sparql-query-20050419/.

Russel, S., & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*, USA: Prentice-Hall.

Schaffer, C. A. (1994). Conservation Law for Generalization Performance. In *Proceedings of the International Conference on Machine Learning*, San Mateo, Ca., USA: Morgan Kaufmann, 259-265.

Stathacopoulou R., Magoulas G. D., & Grigoriadou, M. (1999). Neural network-based fuzzy modeling of the student in intelligent tutoring systems. In *Proceedings of the INNS-IEEE International Joint Conference on Neural Networks*, Washington, USA, July 1999, retrieved December 5, 2005 from http://www.dcs.bbk.ac.uk/~gmagoulas/JCNN0679.PDF.

Stathacopoulou, R., Magoulas, G. D., Grigoriadou, M., & Samarakou, M. (2005). Neuro-fuzzy knowledge processing in intelligent learning environments for improved student diagnosis. *Information Sciences, 170* (2-4), 273-307.

Suarez, J. (2003). Student evaluation trough membership functions in CAT systems. *Revista Mexicana De Fisica, 49* (4), 371-378.

Turban, E., Zhou, D., & Ma. J. (2000). A Methodology for Evaluating Grades of Journals: A Fuzzy Set-based Group Decision Support System. *Paper presented at the 33rd Hawaii International Conference on System Sciences*, January 4-7, 2000, Maui, Hawaii.

Turunen, E. (1999). *Mathematics Behind Fuzzy Logic*, Heidelberg, Germany: Physica-Verlag.

Vaucheret, C., Guadarrama, S., & Muñoz. S. (2002). Fuzzy Prolog: A Simple General Implementation using CLP(R). *Paper presented at the 9th International Conference on Logic for Programming Artificial Intelligence and Reasoning*, October 14-18, 2002, Tbilisi, Georgia.

Weon, S., & Kim. J. (2001). Learning Achievement Evaluation Strategy Using Fuzzy Membership Function. *Paper presented at the 31st ASEE/IEEE Frontiers in Education Conference*, October 10-13, 2002, Reno, NV, USA.

Xenos, M. (2004). Prediction and Assessment of Student Behaviour in Open and Distance education in Computers Using Bayesian Networks. *Computers & Education, 43* (4), 345-359.