International Environmental Modelling and Software Society (iEMSs)
2012 International Congress on Environmental Modelling and Software
Managing Resources of a Limited Planet, Sixth Biennial Meeting, Leipzig, Germany
R. Seppelt, A.A. Voinov, S. Lange, D. Bankamp (Eds.)
http://www.iemss.org/society/index.php/iemss-2012-proceedings

Fuzzy classification trees as environmental indicators

S. Marsili-Libelli*, E. El Basri, C. Plotegher, E. Giusti

Department of Systems and Computers, University of Florence, Italy
* Corresponding author, email: marsili@dsi.unifi.it

Abstract: Classification trees are valuable data mining tools, but sometimes they lack generality and flexibility. The idea described in this paper is to increase the generality of classification trees by deriving a fuzzy inferential engine from its hierarchical structure. The paper describes the steps through which the tree structure is decomposed and translated into fuzzy rules. To obtain a complete inferential system fuzzy memberships are then added and optimized. The combined algorithm is demonstrated with classification examples and the improvements with respect to the original tree are discussed.

***Keywords**:* Data mining; Fuzzy inferential systems; Environmental indicators; Knowledge representation.

1. INTRODUCTION

Classification decision trees (CDT) [Breiman et al., 1984], are nonparametric modelling tools that can be profitably employed when the structural dependence between inputs and outputs is unknown. Their nonparametric nature makes CDTs valuable data mining tools, as described by Witten et al. [2011], and represents a viable alternative to parametric models requiring assumptions and simplifications in addition to parameter calibration. Their hierarchical nature is important to rank the decision variables but often conceals the logical rules behind the decision and the resulting decision sequence often lacks generality and flexibility.

The idea described in this paper is to make a CDT more flexible by using its hierarchical structure to build a fuzzy inference engine (FIE). On a similar basis a combination of CDTs and artificial neural networks has been presented in Tsai et al. [2012], whereas the construction of more general semantic networks for knowledge discovery is described in Chen et al. [2007].

The knowledge building process is organized along the following steps, illustrated in Figure 1

- A CDT is generated based on a set of training instances;
- A parsing algorithm scans the tree and extracts decision rules based on its branching. These rules are then put in a semantic form compatible with the FIE structure, which is then complemented with the appropriate connectives and membership functions;
- These memberships are initially specified by the tree branching and then optimized by minimizing an error criterion based on the training instances using a Genetic Algorithm (GA).

The whole algorithm was developed in the Matlab platform, using the Fuzzy and Statistics toolboxes.

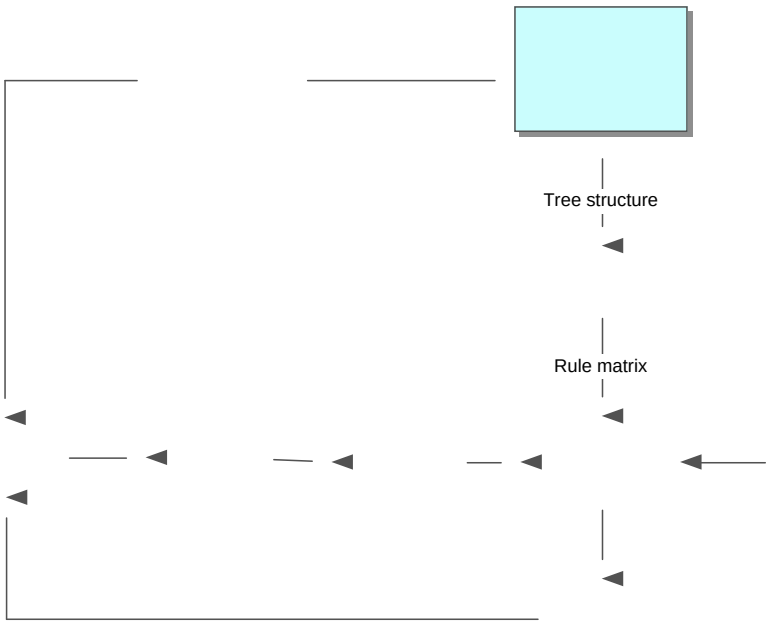
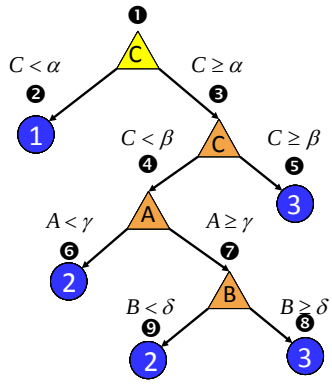


Figure 1. Algorithmic flowchart of the fuzzy decision tree: the knowledge extracted from the tree is converted into a fuzzy inferential engine, whose membership functions are obtained by training and optimization.

2. RULES EXTRACTION FROM A TREE

A decision tree (BDT) represents knowledge as a sequence of binary branches in which each variable is sequentially compared to a series of decision thresholds according to a "divide-and-conquer" algorithm [Witten et al., 2011]. Starting with the most significant test (root) two decisions (branches) are generated. The branching sequence is normally guided by an information entropy reduction and the tests are iterated with the aim of progressively reducing the decision uncertainty until no further split is possible (terminal leaf). In the sequel the analysis will be limited to binary decision trees (BDT). An example of a BDT representation and the ensuing set of decision rules is shown in Figure 2.



Variables : $A - B - C$

Decision thresholds : $\alpha - \beta - \gamma - \delta$

Terminal leaves : 1 - 2 - 3

Rules :

output	*condition*
1	$C < \alpha$
3	$C \geq \alpha \wedge C \geq \beta$
2	$C \geq \alpha \wedge C < \beta \wedge A < \gamma$
2	$C \geq \alpha \wedge C < \beta \wedge A \geq \gamma \wedge B < \delta$
3	$C \geq \alpha \wedge C < \beta \wedge A \geq \gamma \wedge B \geq \delta$

Figure 2. Knowledge organization in a BDT and the corresponding set of rules leading to the terminal leaves.

The first step in constructing fuzzy binary decision trees (FBDT) is the automatic rule extraction from a computer generated BDT.

2.1. Tree representation in the Matlab context

Given a regressor set `X` and the corresponding output set `y` the command `t=classregtree(X,y)` creates a binary decision tree `t` for predicting the output

y as a function of the regressors X. whose threshold values determine the two branches originating from each node. Given a set of training instances, the generated tree is represented as a Matlab structure, which is an array where data of different kinds can be stored in "fields", that can be any kind of data, such as arrays, strings, or real numbers. The output structure `t` is composed of the following fields (among others):

- `t.node`: is a vector consecutively numbering the tree nodes.
- `t.parent`: is a vector containing the number of the parent node. The root node has number zero.
- `t.children`: is a 2-column matrix with the indexes of the two nodes generated from the present node according to the numbering in the `t.node` field The first column contains the node number corresponding to the < decision with respect to the `t.cut` threshold, whereas the second column indicates the node number for the ≥ case. For terminal leaves both values are zero.
- `t.var`: for each node indicates the corresponding decision variable. If its value is zero the node is a terminal leaf.
- `t.class`: indicates the classification for the terminal leaves.
- `t.cut`: contains the threshold values for each node. A zero value indicates a terminal leaf.

The rule extraction process is performed by scanning the fields of the tree structure shown in Figure 3.

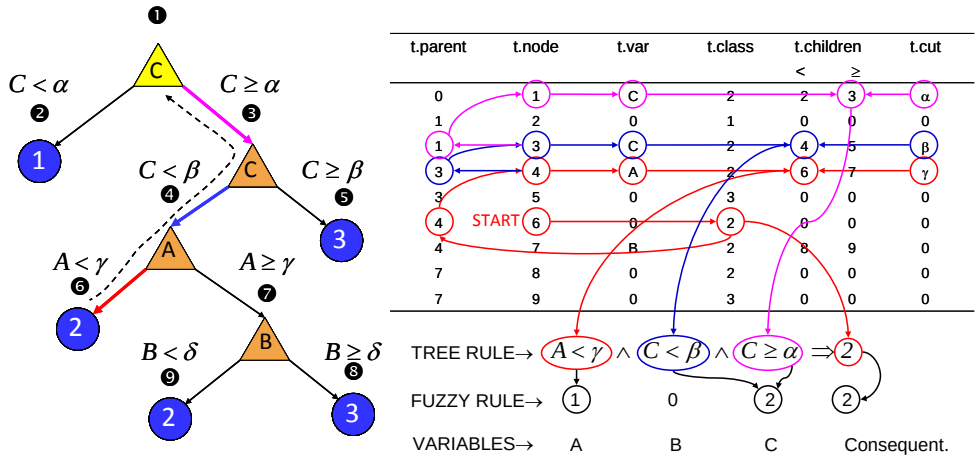


Figure 3. Automatic rule extraction process showing an example of the third rule of Figure 2. The decision path is indicated by the dashed line. The numbers in the black circles identify each node.

For example, consider the formation of the 6th rule, indicated in Figure 3. Starting with the pertinent terminal leaf (6 in this case) the tree table is inspected in reverse. The corresponding value in `t.class` (2) represents the tree output, i.e. the rule consequent. The corresponding element in `t.parent` shows the parent node (4) which implies that the involved variable is A and the threshold value is γ. Since there are no further occurrences of the variable A in the tree, the condition $A < \gamma$ is reconstructed. A progressive number identifies the intervals in which the threshold values divide the variable range and, since $A < \gamma$ is the condition we want to represent, 1 is entered in the first column of the fuzzy rule matrix. Continuing the backwards exploration the other two conditions are inferred. Special attention is needed when the same variable is involved in more than one condition. The algorithm checks each node where the variable appears and uses the information in the `t.children` matrix to infer the correct interval defining the rule. Therefore the conditions $\alpha \leq C < \beta$, produces a 2 entry in the third column of the fuzzy rule matrix. In general the numbers in the matrix are not just the counts of how many

occurrences we have for each variable, but they depend on the intervals generated by the cuts and on the branching taken.

After all the tree branches have been explored the following matrix is obtained, in a form compatible with the format of the file containing all the information of a fuzzy inferential system with extension (`.fis`). This ASCII file, produced by the Fuzzy editor of the Matlab Fuzzy Toolbox, contains all the details of the fuzzy inference system and is normally produced by the Fuzzy interface. While all the other attributes are user-defined, the matrix rule is imported from the `tree2fuzz.m`.

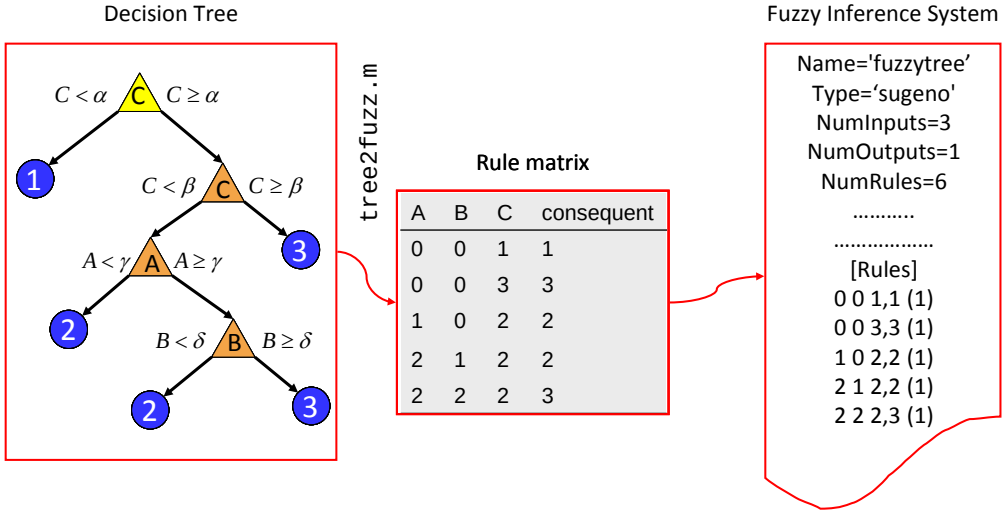


Figure 4. Starting with the BDT (left) the rule matrix (centre) is generated by `tree2fuzz.m` and inserted into the Fuzzy Toolbox inference system (`.fis`) file (right).

2.2. Membership generation and optimization

In the FIE definition shown in Figure 4 the shape and position of the membership functions (mf) should now be defined. This is achieved with reference to the example of Figure 4, together with the selection of connectives required to implement a set of N inferential rules with the semantics defined by Sugeno [Takagi and Sugeno, 1985]

$$R_i : if\ \left(A\ is\ A_i\right)\ and\ \left(B\ is\ B_i\right)\ and\ \left(C\ is\ C_i\right)\ then\ Out = Y_i \quad i = 1,...,N$$

$$Y = \frac{\sum_{i=1}^{N} \mu_i Y_i}{\sum_{i=1}^{N} \mu_i} \quad . \qquad (1)$$

The initial mfs are defined with the following rationale:

- The tree branch points represent the support values with maximum uncertainty, thus the adjacent mf's should cross in their neighbourhood;
- Each intermediate mf should reach its maximum value ($\mu = 1$) in the middle point between two adjacent breakpoints
- The shape of each mf is selected in order to minimize the number of parameters for its characterization. The terminal mfs are represented by Z-shaped spline-based functions (`zmf`), whereas the intermediate mfs are represented by π-shaped functions (`pimf`).

For the subsequent optimization, the parameters of the mfs are grouped into a single vector structured as follows

$$\boldsymbol{P} = \left[\overbrace{\underbrace{p_{11}}_{A_1} \; \underbrace{p_{12}}} \; \overbrace{\underbrace{p_{13}}_{A_3} \; \underbrace{p_{14}}} \; \overbrace{\underbrace{p_{21}}_{B_1} \; \underbrace{p_{22}}} \; \overbrace{\underbrace{p_{23}}_{B_3} \; \underbrace{p_{24}}} \; \overbrace{\underbrace{p_{31}}_{C_1} \; \underbrace{p_{32}}} \; \overbrace{\underbrace{p_{33}}_{C_2} \; \underbrace{p_{34}} \; \underbrace{p_{35}}} \; \overbrace{\underbrace{p_{36}}_{C_3} \; \underbrace{p_{37}}} \right] \tag{2}$$

Using the components of $\boldsymbol{P}$ as the decision variables the shape of the mfs is then optimized in order to minimize the error functional

$$\arg\min_{\boldsymbol{P}} \; E(\boldsymbol{P}) = \sum_{i \in N_{instances}} \left| Class_i^{exp} - Class_i^{FDT}(\boldsymbol{P}) \right| \tag{3}$$

A genetic algorithm (GA) is used for the optimization because of its global convergence properties. Figure 5 shows the shape of the initial and optimized membership functions defining the decision variable C in the previous example. They have two breakpoints (α and β) as shown in Figure 2.

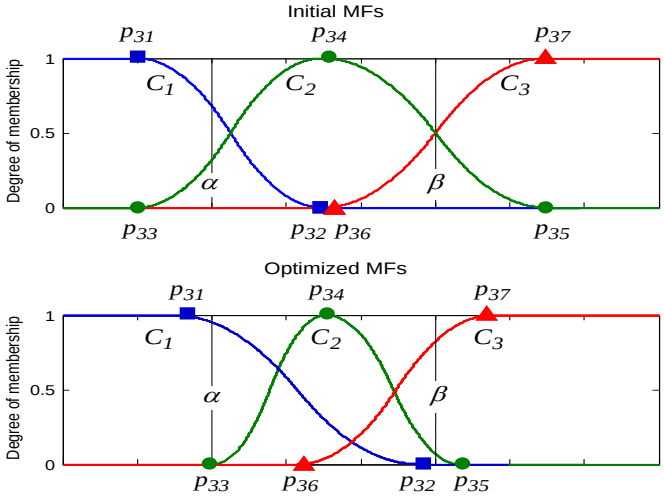


Figure 5. Membership function optimization according to (3): starting from the initial shape (above), the optimal shape is obtained (below). The solid shapes indicate the breakpoints of each mf.

3. APPLICATIONS

Two applications of the fuzzy binary decision tree (FBDT) are now described. In both cases the classification produced by the algorithm is assessed by comparison with other independent classifications and with experimental observations.

3.1. Assessing the ecological quality of Swiss lowland ponds

Recently a multimetric ecological index has been proposed to assess the quality of a number of Swiss ponds [Menetreya et al., 2011]. Seven variables were considered to define the degradation of the ecological status of the ponds: the diversity of macrophytes, the trophic state expressed as total phosphorus and total nitrogen, and five anthropogenic stressors related to land use. Using expert opinion as a yardstick the proposed metrics CIEPT *(Coleoptera genera richness (C) +macroinvertebrate family richness (I) + EPT family richness (EPT)*) were found to correctly represent the environmental quality of the ponds in 88% of cases,. That study was replicated here using the above algorithm. The pond quality was re-classified based on the seven variables according to five integer scores (5-Optimal, 4-Good, 3-Moderate, 2-Poor, 1-Hugely degraded). The mean score for each classification variable was then subdivided in three classes (1 - Reference, 2 - Intermediate, 3 - Degraded) to represent the ecological condition.

Using the data available in Menetreya et al. [2011] the classification tree of Figure 6 (left) was derived. Notice that there are six terminal leaves, so the FIE will be composed of six rules and that only four of the initial seven variables are involved. Hence a further complexity reduction of the classifier is possible. By processing the

above tree though the *tree2fuzz* function the following rules are obtained, shown in Figure 6 (right) as produced by the rule editor of the Matlab Fuzzy Toolbox.

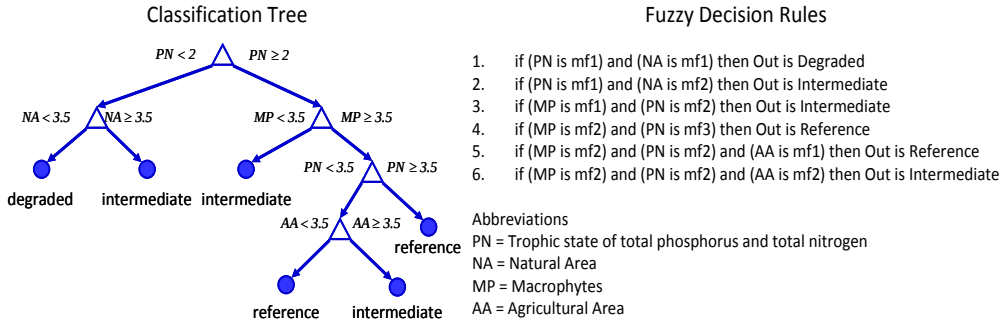


Figure 6. Classification tree for the ecological quality of Swiss ponds [Menetreya et al., 2011].

To obtain the full inferential system, the rules are then supplemented with mfs, as described in sect. 2.2. For each classification variables two or three mfs were used, according to the number of breakpoints. The final results are summarized in Table 1, showing that this new classifier dramatically improves the classification accuracy. The slight degradation of the initial fuzzy tree (before optimization) can be explained with the uncertainty introduced by the fuzzification of the breakpoints, later to be absorbed through their optimization.

Table 1. Comparison of misclassified instances for the Swiss ponds [Menetreya et al., 2011] (FBDT = Fuzzy Binary Decision Tree).

Number of sites	CIEPT index	Unpruned tree	Initial FBDT	Optimal FBDT
46	22 (47.8%)	8 (17.4%)	9 (19.6%)	7 (15.2%)

3.2. Classification of lake stratifications.

Lake thermal stratification is the separation of its waters into an upper layer of relatively warm water (epilimnion) and a lower layer with cooler water (hypolimnion), separated by a comparatively thinner layer (thermocline) characterized by a sharp temperature variation [Wetzel, 2001]. There exist many complex mathematical models to describe the seasonal stratification and mixing of lakes, see e.g. Bell et al., [2006], but here a data-mining approach is considered rather than a deterministic model. Stratification in the Bilancino reservoir, used as a drinking water reservoir for the city of Florence (Italy), has been recorded for several years. As an example, the temperature variations for a typical year (2003) are plotted in Figure 7.

An in depth analysis of the stratification data [Balducci, 2009] showed that the onset of stratification was highly correlated with the magnitude and depth of the second derivative of the temperature profile; the latter were assumed as decision variables. Three conditions were also defined: mixed lake (no stratification), transition (a transitory situation before stratification is established), and stratified.

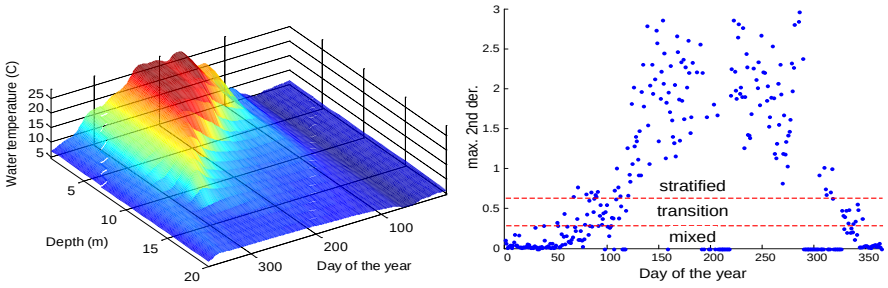


Figure 7. Water temperature variations and related changes in the shape of the thermocline in the Bilancino reservoir during 2003.

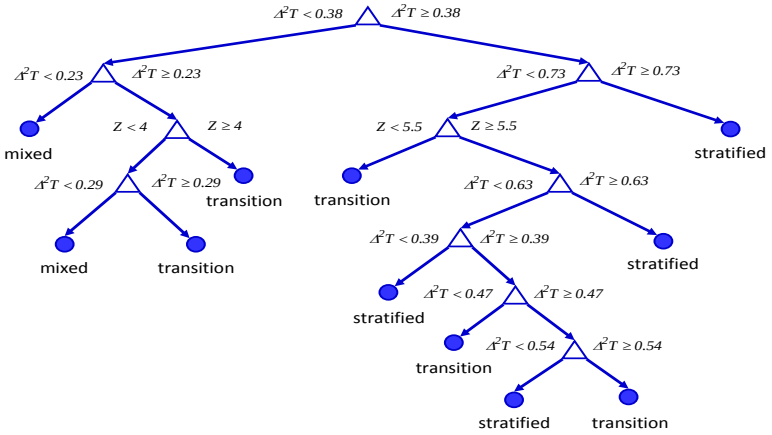


Figure 8. Classification tree for the Bilancino reservoir stratification.

The mfs were then derived and optimized as described before. The data collected during 2003 and 2005 were used to train the algorithm, whereas data from 2008-2009 were used for validation. In all cases direct observations were used to check the accuracy of the FBDT.

Table 2 reports the misclassified instances both in the calibration and in the validation phase. In the latter, there are no initial mfs because the previously optimized FBDT was used. Though the fuzziness introduced in the BDT does not seem to improve its accuracy, it definitely enhances its capability for generalization. It should also be noticed that there was a considerable time lapse between the calibration and validation data and within that period some modification in the stratification mechanisms may have occurred.

Table 2. Classification results for the Bilancino reservoir stratification.

	Days	Unpruned tree	Initial FBDT	Optimal FBDT
Calibration (2003 – 2005)	329	21 (6.4%)	64 (19.5%)	23 (7.0%)
Validation (2008 – 2009)	29	5 (17.2%)	-	5 (17.2%)

The apparent lack of improvement due to fuzzification can be explained with the classification nature of the tree, implying the approximation to the nearest integer of the defuzzified result from (1). Truncation may thus reintroduce some uncertainty in the results, resulting in an apparent lack of accuracy.

4. CONCLUSION

The main idea pursued in this research is to enhance the generalizing properties of a decision tree by adding fuzziness to the decision branching, hence the relation of the membership function definition to the breakpoint locations.

First a parsing algorithm translates the tree branching sequence into a matrix compatible with the syntax of the fuzzy inference system. Membership functions are initially constructed based on the branching thresholds of the tree, and then optimized with a genetic algorithm in order to maximize the success of classification.

The method was applied to the classification of the environmental condition of Swiss ponds and to the stratification of a reservoir. In the first case a significant improvement in the classification accuracy was obtained because of the fine mapping of the FBDT output, which partially offsets the truncation effect of the classification. In the Bilancino case, the continuous nature of the decision variables and the uncertainty with which they were measured made the three-state classification perhaps too coarse to accommodate the fine differences in stratification conditions, hence the negligible fuzzy improvement.

The results presented show that the effectiveness of fuzzification is circumstantial and that the starting classification should be considered. If the classification tree is already highly discriminatory, little improvement could be expected. Further, the aim of the algorithm is, in addition to improving the tree discriminatory capacity, the *generalization* of the inferred semantics and in this regard we do believe that the added flexibility of fuzzification can significantly contribute.

Future developments of this work will be aimed at testing a wider paradigm of cases and at extending the algorithm to regression trees, where the approximation to the nearest integer for classification will not be required (see the Bilancino case 3.2).

REFERENCES

Balducci, *A., Decision tree analysis of the lake stratification in the Bilancino reservoir.* BA Thesis in Environmental Engineering, University of Florence, 2009.

Bell, V.A., D.G. George, R.J. Moore and J. Parker, Using a 1-D mixing model to simulate the vertical flux of heat and oxygen in a lake subject to episodic mixing, *Ecological Modelling* **190**, 41–54, 2006.

Breiman, L., J. Friedman, R. Olshen and C. Stone, *Classification and Regression Trees*. Boca Raton, FL. CRC Press, 1984.

Chen, Z., A. Gangopadhyay, G. Karabatis and M. McGuire, Semantic integration and knowledge discovery for environmental research, *J. of Database Management*, **18**(1), 43-67, 2007.

Menetreya, N., B. Oertli and J.B. Lachavannea, The CIEPT: A macroinvertebrate-based multimetric index for assessing the ecological quality of Swiss lowland ponds. *Ecological Indicators*, **11**, 590–600, 2011.

Tsai, C.C., M.C. Lu and C.C Wei, Decision Tree–Based Classifier Combined with Neural-Based Predictor for Water-Stage Forecasts in a River Basin During Typhoons: A Case Study in Taiwan. *Environ. Eng. Sci.*, **29** (2), 108-116, 2012.

Takagi, T. and M. Sugeno, Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. on Systems, Man, and Cybernetics* **15**, 116 – 132, 1985.

Wetzel, R.G., *Limnology: Lake and River Ecosystems* (Third Edition). Academic Press, 1006 pp., San Diego CA, 2001.

Witten, I.H., E. Frank and M.A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques* (Third Edition). Morgan Kaufmann, 629 pp., Burlington MA, 2011.