Context-dependent Feedback Prioritisation in Exploratory Learning Revisited

Mihaela Cocea[1,2] and George D. Magoulas[2]

[1] School of Computing, University of Portsmouth,
Buckingham Building, Lion Terrace, Portsmouth, Hampshire, PO1 3HE, UK
mihaela.cocea@port.ac.uk
[2] London Knowledge Lab, Birkbeck College, University of London,
23-29 Emerald Street, London, WC1N 3QS, UK
gmagoulas@dcs.bbk.ac.uk

Abstract. The open nature of exploratory learning leads to situations when feedback is needed to address several conceptual difficulties. Not all, however, can be addressed at the same time, as this would lead to cognitive overload and confuse the learner rather than help him/her. To this end, we propose a personalised context-dependent feedback prioritisation mechanism based on Analytic Hierarchy Process (AHP) and Neural Networks (NN). AHP is used to define feedback prioritisation as a multi-criteria decision-making problem, while NN is used to model the relation between the criteria and the order in which the conceptual difficulties should be addressed. When used alone, AHP needs a large amount of data from experts to cover all possible combinations of the criteria, while the AHP-NN synergy leads to a general model that outputs results for any such combination. This work was developed and tested in an exploratory learning environment for mathematical generalisation called *eXpresser*.

Keywords: context-dependent personalised feedback, feedback prioritisation, exploratory learning, analytic hierarchy process, neural networks

1 Introduction

Exploratory learning is characterised by complex tasks such as constructing models and varying their parameters, that can be approached in different ways, leading to equally valid solutions. Although these solutions are varied, they are all characterised by some key points the learners need to address or be aware of. The actions of a learner when solving a task can indicate the points the learner may need help with, however, to be effective, the help that is given should take into consideration the personal characteristics of the learner. Moreover, relevant information could be extracted from the context which can lead to more appropriate feedback. There are many works in the literature that investigate the role of context in a diversity of fields such as recommender systems [2], artificial intelligence [1], educational psychology [40] and ubiquitous computing [27]. The definition of context is also diverse, varying from the wide social context to the specificity of network characteristics. In our approach context refers to the stages within a task, with each stage providing essential information about what is currently relevant for the learner.

Exploratory Learning Environments (ELEs) (e.g. SimQuest [22], Adaptive Coach for Exploration (ACE) [7], Vectors in Physics and Mathematics [18]) are characterised by freedom, allowing learners to explore the domain rather than guide their learning in a structured manner. On the other hand, complete lack of guidance in ELEs is not useful for learning [23]. Consequently, the challenge is to provide feedback in such a way that the learner does not feel restrained and at the same time perceives the feedback as relevant with respect to the current activity. This problem is not unique to exploratory learning environments, but also applies to educational simulated environments (e.g. [42]) and games (e.g. [38]) where the challenge is to provide feedback without breaking the flow [13].

This paper addresses the problem of *personalised feedback prioritisation* in ELEs which allow learners a high degree of freedom as opposed to the guided learning offered by more structured learning environments such as intelligent tutoring systems. The approach was developed using an ELE for mathematical generalisation and the prioritisations used to train the neural network are validated by experts in the field of mathematical education.

In previous work [11] [12], we have proposed an approach for feedback prioritisation based on the Analytic Hierarchy Process [35], a popular method in Multicriteria Decision-Making [43]. Due to the large amount of data needed from experts, the AHP approach was developed only for the most frequent combinations of criteria, where criteria refer to task difficulty, experience and arithmetic ability. This meant that when a combination of criteria was not available, the closest match to the available combinations of criteria was found and the prioritisation of the best match was used instead.

To address this issue, in this paper we present a context-dependent personalised feedback prioritisation mechanism using the Analytic Hierarchy Process and Neural Networks [3]. AHP is used to define feedback prioritisation as a multi-criteria decision-making problem, while NN is used to model the relation between the criteria and the order in which the conceptual difficulties should be addressed, i.e. the prioritisation. When used alone, AHP needs a large amount of data from experts to cover all possible combinations of the criteria, while the AHP-NN synergy leads to a general model that outputs results for any such combination. The experimental study aims to establish the feasibility of the AHP-NN approach for the personalised feedback prioritisation problem.

The next section briefly introduces adaptive feedback, mathematical generalisation and the system employed. Section 3 presents the AHP-NN approach, while Section 4 presents the experimental results obtained using the proposed approach. Section 5 discusses the results and concludes the paper.

2 Adaptive Feedback in Our Exploratory Learning Environment

Feedback is usually a response to the actions of a learner aiming to correct future iterations of the actions [30]. It includes information about what happened or did not happen as a consequence of the user's actions in relations to the goal [41]; this information is given to the users to compare their performance with the expected one [21] and to make use of it in the following attempt [41].

In exploratory learning, the freedom given to learners leads to situations when feedback is required on several aspects. This is also the case of *eXpresser* [3] [31] [33], which is an ELE for mathematical generalisation that aims to link the visual with the algebraic-like representation of rules. It enables constructions of patterns, creating dependences between them, naming properties of patterns and creating algebraic-like rules with either names or numbers. Some screenshots are displayed in Figure 1, illustrating the system, two constructions, the *properties list* of a pattern that is dependent on another one, the *properties list* of an independent pattern and two examples of *rules*.
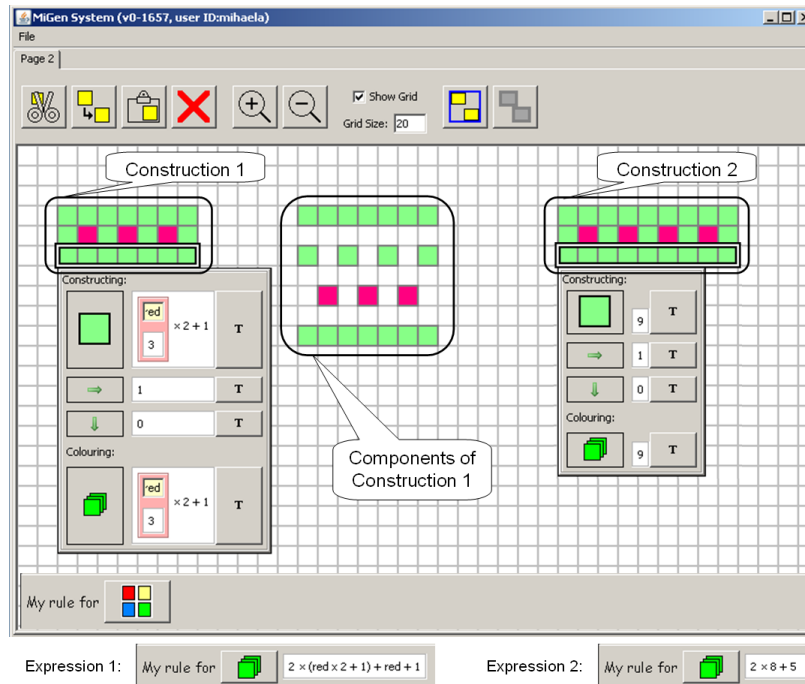

Fig. 1. eXpresser screenshots. The main screenshot includes a toolbar, an area for pattern construction and an area for defining rules/algebraic-like expressions; the toolbar (at the top) allows the following actions: cut, copy, paste, delete, zoom in, zoom out, show grid, grid size (changeable from here or using the zoom tools), group and ungroup; the main area has two constructions for the "footpath" task and two *property lists*; the components of *Construction 1* are also presented separately. The two screenshots at the bottom illustrate the rules defined by the learners who built the two constructions.

The main area of the screen in Figure 1 displays two constructions. These are solutions of two learners working independently on a task called "footpath", which is typical in the UK curriculum. The task requires to find out the number of green tiles needed to surround *any* pattern of red tiles (representing the footpath). The components of *Construction 1* are displayed separately for ease

of understanding; this construction has four patterns: (a) two compact rows of green (lighter colour) tiles and (b) two rows with gaps in between tiles: one green and one red (darker colour). The first two mentioned are the same, and consequently, have the same properties displayed in the *property list* of the highlighted row in *Construction 1*. The first property, i.e. number of iterations, shows that the pattern depends on the red one because the number of iterations of the green tiles is set to 'the number of red tiles multiplied by 2 plus 1'; the *T box* with the name *red* and the corresponding value of 3 is called an *icon variable* and is used to make a pattern dependent on another; the use of *icon variables* leads to general constructions, i.e. they work for any number of red tiles. The second property, *moving left*, is set to 1 and the third property, *moving down*, is set to 0, which makes the pattern a row; for the red pattern *moving left* is set to 2 and *moving down* is set to 0, which makes a row with gaps between the tiles. The last property establishes the number needed to colour all the tiles in the pattern; in the current case it is the same as the number of iterations in the pattern. However, if a pattern is a group of several tiles, this would not be the case anymore; for example, if a pattern is a group of three tiles and is iterated five times, the number required to colour it would be three times five.

Construction 2 is built in a similar fashion, but the compact rows of green tiles do not depend on the red pattern: the first property (number of iterations) from the *property list* is set to 9. At the bottom of Figure 1, two expressions corresponding to the two constructions are displayed. *Expression 1* uses the name *red* for the number of red tiles, while *Expression 2* is numeric.

In the constructions of Figure 1, both learners follow the same strategy in surrounding the footpath: two rows of tiles at top and bottom, and one row of tiles in the gaps of the red pattern; also, for both constructions, the row of green tiles with gaps in between (the middle one) does not depend on the red pattern and the expressions do not correspond to their corresponding constructions. However, there are a few differences: (a) they work with a different number of red tiles, i.e. 3 and 4, respectively; (b) the first learner is very close to a general solution, while the second is still working with the particular case of 4 red tiles; (c) the expression of the first learner (*Expression 1* in Figure 1) is already general, while the expression of the second learner (*Expression 2* in Figure 1) is numeric.

Construction 2 could be used at this point to illustrate how the need for feedback prioritisation emerges during exploration. In this instance, from pedagogical point of view, several issues need to be addressed: (a) the construction is correct only when the red pattern consists of four tiles, i.e. it is specific, whilst the aim of the activity is to create a general construction that would work for any number of tiles; (b) the learner may need to be reminded how to make a pattern dependent on another (i.e. the use of icon variables); (c) the expression does not correspond to the construction and contains a mistake; (d) the expression is specific. To this end, different types of feedback are needed depending on learner's characteristics and contextual information. In the next section, we describe an approach that leads to prioritising feedback on these issues based on a multi-criteria decision making method called the *Analytic Hierarchy Process*.

3   The AHP&NN Feedback Prioritisation Approach

Multicriteria Decision Making (MDM) defines a class of problems where a decision from a predefined set of alternatives needs to be reached by taking into account two or more criteria. Each alternative is evaluated on the set of criteria; the outcomes provide a means of comparison between the alternatives that will facilitate a selection of one or more alternatives, or a ranking between them. Other purposes are classification of alternatives into groups (clustering) and group ranking [43]. Among the possible approaches of decision problems that correspond to this description are: statistical techniques, multi-attribute utility analysis, analytic hierarchy process, knowledge bases, mathematical models, etc.

MDM has many applications in fields where decisions need to be taken. The Analytic Hierarchy Process (AHP) is one of the most popular methods in MDM and is widely applied in a diversity of areas like logistics, military, manufacturing and health-care [20]. Frequently AHP is used in combination with other methods - a recent literature review [20] reports five main categories of tools integrated with AHP: (a) mathematical programming, (b) quality function development, (c) meta-heuristics, (d) SWOT analysis, and (e) data envelopment analysis. Four works related to higher education are reported in areas of IT-based project selection [26], teaching method selection [28], education requirement selection [24] and faculty course assignment [32].

In the area of learner/user modelling, AHP has been used in combination with fuzzy logic [17] for student diagnosis in an adaptive hypermedia educational system and in combination with Multi-Attribute Utility Theory (MAUT), another method from MDM, in recommender systems [37], where the evaluation function from MAUT is used to rate how well each alternative fulfills the decision criteria.

The AHP uses a hierarchy to represent a decision problem and to establish priorities between alternatives depending on a set of criteria involved in the decision process. It includes three main steps: (a) construction of the hierarchy; (b) analysis of priorities and (c) verification of consistency.

The *hierarchy* has the general structure presented in Figure 2. The highest level represents the *goal*, which, in our context, is personalised feedback. The second level includes the *criteria* based on which the decision should be taken; in our case, the criteria refer to the stage in the exploratory task. The third level includes the *alternatives* to be prioritised with respect to the criteria; the alternatives correspond to pedagogical aspects of mathematical generalisation. The first step includes a decomposition of the decision problem into parts defined by all relevant attributes; these attributes are arranged into hierarchical levels so as to reach the hierarchical structure presented in Figure 2.

To obtain a prioritisation of the alternatives, pairwise comparisons are needed between each pair of criteria and between each pair of alternatives. The later requires comparisons with respect to each criteria, i.e. if there are n criteria and m alternatives, $_nC^2$ pairwise comparisons are needed for the criteria and $n *_m C^2$ pairwise comparisons are needed for the alternatives ($_mC^2$ comparisons per criterion). In previous work [12] we have used AHP alone to produce the prioritisation feedback. Due to the amount of pairwise comparisons, however, we

have considered as criteria only the stage within the task and produce a pairwise comparisons for different combinations of learner characteristics. This approach was taken because if the learner characteristics would have been introduced in the AHP hierarchy as criteria, it would have led to a vast amount of pairwise comparisons.
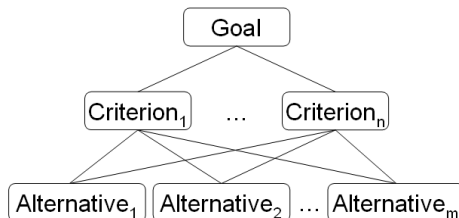

Fig. 2. Hierarchy in the Analytic Hierarchy Process

The next step in AHP is the verification of consistency, calculated by a set formula based on the pairwise comparisons. If there is a lack of consistency, the pairwise comparisons need to be reviewed and the consistency is checked again. This process is repeated until the consistency criteria is satisfied. Consequently, there could be even more effort needed from the experts at this stage.

To address this limitation, we propose to use the AHP hierarchy with the context, task difficulty and learner characteristics as criteria, and use neural networks to model the relation between the criteria and the prioritisation of alternatives. More specifically, we use a back-propagation network which is a multi-layered feed-forward neural network which is fully connected [15]. Each layer can have several units, with each unit connected to all the units in the next layer. By training the network, the optimal weights between units are learned and therefore, one could say that the knowledge about the aspect being modelled is encoded in these weights. For our purpose, by training a network on instances representing combinations of criteria with their corresponding order of alternatives, the network will learn the association between the two and will be able to output prioritisations for any combination of criteria.

Neural networks have the ability to derive meaning from complex or imprecise data and are used to extract patterns and to detect trends that cannot be noticed by humans due to their complexity [3]. Neural networks have been used in a variety of fields such as medicine [5], biology and chemistry [9], engineering [36], finance [29] and management science [25]. In the area of elearning, neural networks were used for personalised recommendations of learning objects [4].

For our purpose, the neural network will have as input the criteria and as output the prioritisation of alternatives. The criteria are the stages within a task: (1) specific construction; (2) variation of parameters; (3) general construction and (4) expression. To identify the stage a learner is at, a set of rules are used. Basically, for each stage, the presence or absence of certain actions or properties of the construction/expression are used. A learner is considered to be in the *specific construction* stage if he has not used T-boxes yet and the construction is not complete (i.e. it does not fit the mask of the task construction). The *variation of parameters* is indicated by the change in the values of the properties

of patterns. The *general construction* stage is identified by the presence of T-boxes, while the *expression* stage is identified by modifications made to the expression. The other criteria are task difficulty and learners' characteristics, i.e. their experience with tasks of various degrees of difficulty and their arithmetic ability.

The alternatives are feedback on the following aspects: (a) correctness of construction (CC); (b) correctness of expression (CE); (c) construction-expression correspondence (C-E); (d) symmetry of construction (Sym); (e) generality of construction (CGen); (f) generality of expression (EGen); (g)linking patterns (LP).

4   Experimental Results

The network has four input nodes and seven output nodes. The inputs are the AHP criteria and the outputs are the AHP alternatives that were introduced in the previous section. The data used for our experiments consists of 108 instances of criteria combinations and their corresponding prioritisations of the alternatives. The criteria and their coding is presented in Table 1. The 108 instances were obtained by combining the number of values for all the criteria: $4 \times 3 \times 3 \times 3$.

Table 1. Criteria and their coding.

Criteria	Possible values	Coding
Context, i.e. stage in the task	specific construction	1
	variation of parameters	2
	general construction	3
	expression	4
Task difficulty	low	1
	medium	2
	high	3
Experience	low	1
	medium	2
	high	3
Arithmetic ability	low	1
	medium	2
	high	3

The alternatives are the ones mentioned in the previous section and they are coded as 1 to 7 in the order they were introduces, i.e. CC is coded as 1 and LP is coded as 7. Both inputs and outputs are normalised by mapping minimum and maximum values to [-1 1].

One expert produced the prioritisations (i.e. the ranking of the alternatives from 1 to 7) for all instances and two other experts were asked to validate these prioritisations for 48 instances (approximately 45% of all instances). One expert agreed with 97% of the prioritisations, while the other agreed only with 91%. On the intsances on which there was disagreement, the prioritisations were modified to reflect the agregated opinion of all experts. One could argue that the two experts that validated the prioritisations could have been asked to validate separate sets of prioritisations, thus covering 90% of all instances. This approach,

however, could have led to inconsistencies between the prioritisations of the two experts which could affect the performance of the neural network. Therefore, we gave the same set of instances to the two experts to avoid this problem.

The data was randomly partitioned into three sets used for training (55%, i.e. 60 instances), validation (15%, i.e. 16 instances) and testing (32%, i.e. 30 instances). We have tested several networks architectures and found that the best performance is obtained when the number of hidden nodes is 10. The sigmoid function was used as activation function in hidden layer and the linear activation function was used in the output layer - this can be seen in Figure 3. The Levenberg-Marquart [19] algorithm was used for training. Early stopping technique was applied to check the validation error rate periodically during training [34]. The stopping conditions were the gradient magnitude (with a value of 1e-5) and the number of validation error checks (set to 6).
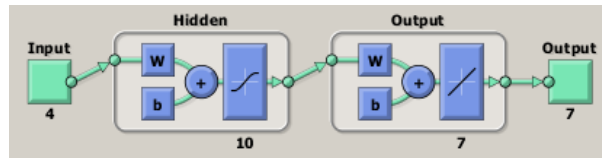

Fig. 3. Network architecture.

Using the set-up presented above, the network was used in 500 independent trials and the results are given in Table 2. The average number of iterations was 15.81 with a standard deviation of 1.96; the maximum and minimum number of iterations were, respectively, 53 and 5.

Table 2. The results of the 500 independent runs.

	Training	Validation	Testing
Average performance	0.899	0.902	0.906
Standard deviation	0.101	0.168	0.129
Average error(mse)	0.002	0.002	0.002
Standard deviation	0.033	0.055	0.042
Best performance	0.936	0.982	0.964
Worst performance	0.864	0.795	0.871

Performance refers to correct ranking of the alternatives, while error refers to the means squared error (mse), i.e. the average squared error between the network outputs and the target outputs. For our particular purpose, it is important that the neural network returns a prioritisation according to the context and the learning characteristics. Consequently, to judge the performance of the network, we need to measure if the network outputs match the target outputs not in terms of the values returned, but of the order between those values. Therefore, to calculate the network performance, we compared the ranking produced at output with the target ranking.

On the other hand, we are interested in learning the mapping between the inputs and the outputs regardless of the exact ranking because the neural network will be used to rank alternatives on the basis of imprecise information about the criteria, i.e. real values not just integers could be used for task difficulty,

experience and arithmetic ability. For example, a learner's experience with tasks of medium difficulty, does not suddenly go from low (coded as 1) to medium (coded as 2), but could have intermediate values such as 1.2 or 1.8. The network should be able to return prioritisations when these intermediate values are used, and therefore, we are interested in the network's performance in terms of the mean squared error.

Looking at the results for the average error and its standard deviation, we can see that the network fits the data well. In terms of performance from ranking point of view, the results show that the network's output is the same as the targets in 91% of the cases.

5   Discussion and Conclusions

In this paper we proposed an AHP-NN approach to address the problem of feedback prioritisation in an exploratory learning environment for mathematical generalisation. In our previous work we used a sole AHP approach and run into difficulties due to the number of pairwise comparisons needed and the amount of time experts would need to spend on providing the pairwise comparisons and validating the outputs of the AHP mechanism.

In this paper we addressed this issue by proposing the use of neural networks that are capable of generalising complex relationships by mapping one data space to another one using a number of examples. Moreover, the AHP-NN approach offers the advantage of returning prioritisations when real values rather than integers are used for criteria. In AHP, the knowledge of the mapping is in the pairwise comparisons and it is only possible to take into account the exact value of a criterion, e.g. a value of 1.2 and a value of 1.8 may be considered the same. In other works, in AHP an integer actually covers a range of values; for example 2 could be used for any value between 1.51 and 2.49, or between 2.00 and 2.99 (different experts have different views on this). The AHP-NN approach has the advantage of producing more refined prioritisations, i.e. a value of 1.2 for one criterion may lead to a different prioritisation compared with a value of 1.8 of the same criterion.

Due to the nature of our problem, the neural network needed to be tested in terms of how well it generalises, i.e. how well the networks performs with unknown data at the input, with respect to the following two aspects: the mean squared error and the correspondence between the ranking produced at output and the target ranking. The first aspect is the typical way to judge a neural network and the results show that, from this point of view, the performance is very good. The other aspect, however, is equally important and the findings are promising with an overall success of 91%.

The second aspects on which we tested out network is in fact a label ranking problem, i.e. a complex prediction task with the goal to map instances to a total order over a finite set of predefined labels [39]. Several approaches have been used to address this problem such as kernel methods [16], instance-based learning [10], case-based reasoning [6] and log-linear models [14]. We found only one previous

work on label ranking using neural networks [8] developed specifically for ranking results returned by internet search engines.

Looking at the worst performing cases for the training, validation and testing sets, we have identified 10 instances that led to high errors. When looking at these instances we found they reflect infeasible combinations of criteria that fall in three categories: combining low experience and low arithmetic ability with the last two stages of the tasks; combining low experience and low arithmetic ability with high task difficulty, and combining low arithmetic ability with the last stage of the task (which requires the development of an arithmetic expression). In reality, in a classroom situation teachers would not expect learners with low arithmetic ability (with or without experience) to solve difficult tasks or finalise the last two stages of the tasks. The last two stages require the use of arithmetic expressions; in the penultimate phase, expressions need to be defined in order to make the construction general, while the last stage requires the learner to define an algebraic expression that corresponds the the construction that s/he built. Therefore, these situations could be compared to learners attempting to solve problems for which they do not have the necessary prerequisites.

To address the issue outlined above, in future work we will work on two possible solutions. One is to filter the input data to detect infeasible situations that could then be dealt with appropriate feedback or by informing the teacher. For example, if low experience and low arithmetic ability is detected the student could be given feedback to redo some stages of the task before going into the the highly difficult part. The other solution that we will investigate is to add an extra output for the neural network that could help the network learn infeasible situations and provide special type of feedback with the highest priority.

In conclusion, the research presented in this paper proposed an AHP-NN approach to address the problem of feedback prioritisation in ELEs. Although tailored for *eXpresser* the approach could be used in other ELEs provided that information is available about the aspects to give feedback on, and experts can provide some representative cases to train the neural network.

References

1. Akman, V., Bouquet, P., Thomason, R., Young, R. (eds.): Modeling and Using Context: Third International and Interdisciplinary Conference, CONTEXT 2001, LNAI, vol. 2116. Springer (2001)
2. Anand, S., Mobasher, B.: Contextual recommendation. In: Proceedings of Web-Mine 2006, pp. 142–160 (2007)
3. Anderson, J.A.: An Introduction to Neural Networks. MIT Press (1995)
4. Baylari, A., Montazer, G.: Design a personalized e-learning system based on item response theory and artificial neural network approach. Expert Systems with Applications 36(4), 8013 – 8021 (2009)
5. Behrman, M., Linder, R., Assadi, A.H., Stacey, B.R., Backonja, M.M.: Classification of patients with pain based on neuropathic pain symptoms: Comparison of an artificial neural network against an established scoring system. European Journal of Pain 11(4), 370 – 376 (2007)
6. Brinker, K., Hullermeier, E.: Label ranking in case-based reasoning. In: Weber, R., Richter, M. (eds.) Case-Based Reasoning Research and Development, LNCS, vol. 4626, pp. 77–91. Springer (2007)

7. Bunt, A., Conati, C.: Probabilistic student modelling to improve exploratory behaviour. User Modelling and User-Adaptive Interaction 13(3), 269–309 (2003)
8. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: Proceedings of the 22nd international conference on Machine learning. pp. 89–96. ICML '05, ACM (2005)
9. Cartwright, H.M.: Artificial neural networks in biology and chemistrythe evolution of a new analytical tool. In: Walker, J.M., Lvingstone, D.J. (eds.) Artificial Neural Networks, Methods in Molecular Biology, vol. 458, pp. 1–13. Humana Press (2009)
10. Cheng, W., Hühn, J., Hüllermeier, E.: Decision tree and instance-based learning for label ranking. In: Proceedings of the 26th International Conference on Machine Learning. pp. 161–168. ICML '09, ACM (2009)
11. Cocea, M., Magoulas, G.: Context-dependent personalised feedback prioritisation in exploratory learning for mathematical generalisation. In: Proceedings of UMAP2009, LNCS, vol. 5535, pp. 271–282. Springer (2009)
12. Cocea, M., Magoulas, G.D.: Hybrid model for learner modelling and feedback prioritisation in exploratory learning. International Journal of Hybrid Intelligent Systems 6(4), 211–230 (2009)
13. Csikszentmihalyi, M.: Finding Flow: The Psychology of Engagement With Everyday Life. BasicBooks, New York (1997)
14. Dekel, O., Manning, C.D., Singer, Y.: Log-linear models for label ranking. In: Advances in Neural Information Processing Systems. vol. 16, pp. 497–504 (2003)
15. Dreyfus, G.: Neural networks: methodology and applications. Springer (2005)
16. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Advances in Neural Information Processing Systems. vol. 14, pp. 681–687 (2001)
17. Grigoriadou, M., Kornilakis, H., Papanikolaou, K.A., Magoulas, G.D.: Fuzzy inference for student diagnosis in adaptive educational systems. In: Proceedings of SETN2002. LNAI, vol. 2308, pp. 191–202. Springer (2002)
18. Grigoriadou, M., Samarakou, M., Mitropoulos, D., Rigoutsos, A., Stavridou, E., Solomonidou, C.: Vectors in physics and mathematics. In: Proceedings of the International Conference on Technology and Education. pp. 71–73 (1999)
19. Hagan, M., Menhaj, M.: Training feedforward networks with the marquardt algorithm. IEEE Transactions on Neural Networks 5, 989–993. (1994)
20. Ho, W.: Integrated analytic hierarchy process and its applications – A literature review. European Journal of Operational Research 186(1), 211–228 (2008)
21. Johnson, D., Johnson, R.: Cooperative learning and feedback in technology-based instruction. In: J.V., D., G.C., S. (eds.) Interactive Instruction and Feedback. pp. 133–157. Englewood Cliffs, NJ (1993)
22. van Joolingen, W.R., King, S., de Jong, T.: The simquest authoring system for simulation-based discovery environments. In: de Boulay, B., Mizoguchi, R. (eds.) Knowledge and media in learning systems. pp. 79–87. Amsterdam: IOS (1997)
23. Kirschner, P., Sweller, J., Clark, R.E.: Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential and inquiry-based teaching. Educational Psychologist 41(2), 75–86 (2006)
24. Koksal, G., Egitman, A.: Planning and design of industrial engineering education quality. Computers & Industrial Engineering 35(3-4), 639–642 (1998)
25. Krycha, K.A., Wagner, U.: Applications of artificial neural networks in management science: a survey. Journal of Retailing and Consumer Services 6(4), 185 – 203 (1999)

26. Kwak, N.K., Lee, C.W.: A multicriteria decision-making approach to university resource allocation and information infrastructure planning. European Journal of Operational Research 110(2), 234–242 (1998)
27. Kwon, O.: The potential roles of context-aware computing technology in optimization-based intelligent decision-making. Expert Systems with Applications 31(3), 629–642 (2006)
28. Lam, K., Zhao, X.: An application of quality function deployment to improve the quality of teaching. International Journal of Quality and Reliability Management 15(4), 389–413 (1998)
29. Landajo, M., de Andrs, J., Lorca, P.: Robust neural modeling for the cross-sectional analysis of accounting information. European Journal of Operational Research 177(2), 1232 – 1252 (2007)
30. Mason, B., Bruning, R.: Providing feedback in computer-based instruction: What the research tells us. http://dwb.unl.edu/Edit/MB/MasonBruning.html (2001)
31. Noss, R., Hoyles, C., Mavrikis, M., Geraniou, E., Gutierrez-Santos, S., Pearce, D.: Broadening the sense of 'dynamic': a microworld to support students mathematical generalisation. The International Journal on Mathematics Education 41(4), 493–503 (2009)
32. Ozdemir, M.S., Gasimov, R.N.: The analytic hierarchy process and multiobjective 0–1 faculty course assignment. European Journal of Operational Research 157(2), 398-408 (2004)
33. Pearce, D., Mavrikis, M., Geraniou, E., Gutiérrez, S.: Issues in the design of an environment to support the learning of mathematical generalisation. In: Proceedings of EC-TEL 2008. pp. 326–337 (2008)
34. Prechelt, L.: Automatic early stopping using cross validation: quantifying the criteria. Neural Networks 11, 761–767 (1998)
35. Saaty, T.: The Analytic Hierarchy Process. New York: McGraw-Hill (1980)
36. Schlechtingen, M., Santos, I.F.: Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection. Mechanical Systems and Signal Processing In Press (2010)
37. Schmitt, C., Dengler, D., Bauer, M.: Multivariate preference models and decision making with the MAUT machine. In: Proceedings of UM2003. LNAI, vol. 2702, pp. 297–302. Springer (2003)
38. Vasilyeva, E.: Towards personalized feedback in educational computer games for children. In: Proceedings of the sixth conference on IASTED International Conference Web-Based Education - Volume 2. pp. 597–602. ACTA Press (2007)
39. Vembu, S., Gärtner, T.: Label ranking algorithms: A survey. In: Johannes Fürnkranz, E.H. (ed.) Preference Learning. Springer–Verlag (2010)
40. Wang, S., Treat, T., Brownell, K.: Cognitive processing about classroom-relevant contexts: Teachers' attention to and utilization of girls' body size, ethnicity, attractiveness, and facial affect. Journal of Educational Psychology 100(2), 473–489 (2008)
41. Wiggins, G.: Feedback: how learning occurs. http://www.authenticeducation.org/bigideas/article.lasso?artId=61 (2008)
42. Zigmont, J.J., Kappus, L.J., Sudikoff, S.N.: Theoretical foundations of learning through simulation. Seminars in Perinatology 35(2), 47 – 51 (2011)
43. Zopounidis, C., Doumpos, M.: Multicriteria classification and sorting methods: A literature review. European Journal of Operational Research 138(2), 229–246 (2002)