A Hybrid Case Based Reasoning Approach for Wine Classification

Claudio A. Policastro, Andre C. P. L. F. Carvalho and Alexandre C. B. Delbem
Institute of Mathematical and Computer Sciences - University of Sao Paulo.
Sao Carlos, Sao Paulo, Brazil
{*capoli, andre, acbd*}*@icmc.usp.br*

L.H.C. Mattoso, E. Minatti, E.J. Ferreira, Carlos E. Borato
Embrapa-Brazilian Agricultural Research Corporation.
Sao Carlos, Sao Paulo, Brazil.
{*mattoso,minatti,ednaldo,borato*}*@cnpdia.embrapa.br*

M.C. Zanus
Embrapa-Brazilian Agricultural Research Corporation.
Bento Gonalves, Rio Grande do Sul, Brazil.
{*zanus*}*@cnpuv.embrapa.br*

Abstract

There is an increasing concern with the quality of beverage products, like water, milk, wine and coffee. For wine, in particular, the evaluation is usually performed by human tasters, which may require a long time for their training. Moreover, the use of human tasters usually presents high financial and health costs and has a strong subjective component.

The quality control of beverages may strongly benefit from the automatic monitoring of their properties by using taste sensors and intelligent systems. This work investigates how the application of a Artificial Intelligence, more specifically a Hybrid Case-Based System, can lead to an efficient tool for wine quality monitoring. For such, a set of measures extracted by a set of taste sensors is analysed by an Intelligent Hybrid system. Experimental results show the ability of the proposed system to evaluate wine quality.

1. Introduction

There is an increasing concern with the quality of beverage products like water, milk, wine and coffee. The beverage quality control may strongly benefit from the automatic monitoring of the beverage properties by using taste sensors and intelligent systems.

Currently, the evaluation of wine taste and quality is carried out by human tasters, which may take a long time to be trained. Human tasters may suffer from health risks, present high costs and provide analysis with a strong subjective component. The use of taste sensors together with intelligent systems may reduce the cost, risks, subjectivity and time associated with these tests, improving the quality control of wine production. This paper investigates how Machine Learning techniques can be employed to support quality control, by monitoring wine quality. For such, this work uses a data set composed by samples taken from a set of taste sensors associated with information regarding wine quality. The Artificial Intelligence based tool uses hybrid a Case Based Reasoning (CBR) system for monitoring product quality using values collected by taste sensors. The CBR system uses a committee of Machine Learning algorithms to perform automatic case adaptation. This paper is organized as follows: Section 2 explains the main steps associated with case adaptation and presents related works. Section 4 describes the hybrid system architecture used in this paper. Section 5 shows the main experimental results. Section 6 presents the final considerations.

2. Case Adaptation

When CBR systems are applied to real-world problems, the retrieved solutions for a new problem can rarely be directly used to solve it. Retrieved solutions

0-7695-2976-3/07 $25.00 © 2007 IEEE
DOI 10.1109/ISDA.2007.64

usually require adaptations in order to be applied to new contexts. The adaptation process may be either as simple as the substitution of a *component* (in this work, the case solution attributes are named components) from the retrieved solution or as complex as a complete modification of the solution structure. The adaptation can occur by inclusion, removal, substitution or transformation of the components of a previous solution.

Case adaptation is one of the major challenges for the development of CBR [4] [14] systems. Due to its complexity, several CBR systems avoid using adaptation at all. The most widely used form of adaptation employs hand coded adaptation rules, which demands a significant effort of knowledge acquisition for case adaptation, presenting several difficulties [4]. Usually, these hand coded adaptation rules are heuristics or knowledge packages acquired specifically for a particular application domain, like the set of adaptation rules proposed in [6]. Case adaptation knowledge is hard to acquire and demands a significant knowledge engineering effort.

An alternative used to overcome these difficulties is the adaptation knowledge acquisition by automatic learning, where case adaptation knowledge is extracted from previously obtained knowledge: the case base. Nevertheless, not many experiments using automatic learning adaptation knowledge are reported in the literature.

In one of few works in this area, Hanney [4] proposed an algorithm that automatically acquires adaptation knowledge as a set of adaptation rules from a CB. When a new problem is presented to the CBR system, a case is retrieved from the CB and sent to the adaptation mechanism. This mechanism, in turn, extracts the differences between the retrieved case and the new problem description. Next, it searches in the adaptation rules set for proper rules to deal with the differences. Finally, the adaptation mechanism generalizes the selected rules and applies them to the retrieved solution, in order to obtain a new solution. However, this approach employs domain specific mechanisms and do not take advantage of existing Machine Learning algorithms.

In another work, Wiratunga et al. [14] investigated an inductive method for automatic acquisition of adaptation knowledge from a CB. The adaptation knowledge extracted from the CB is used to train a committee of Rise algorithms [1] by applying Boosting [3] to generate different classifiers. However, the knowledge generation process proposed is specific for design domains due the specific encoding employed for the adaptation training patterns and the extraction of differences between description attributes and between component solution attributes.

3. Problem Domain

The data set used comes from a study of wine quality where samples were taken from a set of 9 taste sensors and these data were assigned with information about wine quality. There is a total of 289 cases in this data set. The first 9 values of each case are the values taken from the chemical sensors. The last 3 values of each case are the PH, the absorption index and the taste score of the wine (see Table 1).

Table 1. Wine case base structure.

	Attribute	Value
Problem	Type of wine	white, red
	Sensor 1	Continue
	Sensor 2	Continue
	Sensor 3	Continue
	Sensor 4	Continue
	Sensor 5	Continue
	Sensor 6	Continue
	Sensor 7	Continue
	Sensor 8	Continue
	Sensor 9	Continue
Solution	PH	Continue
	Absorption Index	Continue
	Taste Score	61..88

4. Hybrid System Architecture

This work investigates the use of committees of Machine Learning (ML) algorithms to perform the adaptation of cases in a CBR system for wine classification. The committees investigated are composed of ML algorithms, here named estimators, based on different paradigms. One ML algorithm, here named combiner, combines the outputs of the individual estimators in order to produce the output of the committee. The estimators and the combiner are used to perform adaptations in the retrieved solution, which is used to predict the quality of a given wine.

4.1. Case Adaptation

The approach for case adaptation employs two modules. The first module (adaptation pattern generation) produces a dataset of adaptation patterns in the following way. Let x be a problem stored in the CB and y_i, $i = 1, \dots, n$, one of the cases retrieved by the CBR retrieval mechanism when x is presented. A pattern is obtained by uniting each component of the solution stored in x with the respective component of the solution stored in y_i. Next, the adaptation patterns are used in the training of the ML algorithm of the second module. After the training, the adaptation mechanism is used as a heuristic to adapt the component values of a retrieved solution.

The second module is composed by an automatic adaptation mechanism. This mechanism is composed by estimators generated by the training a committee of ML algorithms using the adaptation pattern data set. After the training, the adaptation mechanism is able to automatically perform case adaptation.

This approach assumes that a CB is representative [11], i.e. the CB is a good representative sample of the target problem space. Therefore, no re-training of the adaptation mechanism is required when the system creates new cases during the reasoning process.

The dataset generation module is capable of extracting implicit knowledge from a CB. This module employs an algorithm of knowledge extraction presented in [9].

Figure 1 illustrates the process of extraction of adaptation knowledge from the CB employed for the experiments performed in this paper (for the PH component). For each case in a CB, the pattern generation algorithm extracts a case from the CB (step 1 on Figure 1) and uses it as a new problem to be presented to the CBR system. The remaining cases compose a new CB without the proof case. Next, the algorithm extracts, from the proof case, the attributes of the problem and a component of the solution. Then, the algorithm returns the K most similar cases matching the proof case (step 2 on Figure 1). For each retrieved case, the attributes of the problem and a component of the corresponding solution are extracted. Next, the algorithm generates the adaptation patterns using (step 3 on Figure 1):

- Input attributes: the problem description stored in the test case; the problem description stored in the retrieved case; one component solution stored in the retrieved case.
- Output attribute: one solution component stored in the test case.

The generated adaptation pattern is then stored in a data set that will be used to train the adaptation mechanism of the system (step 4 on Figure 1).

After the generation of the adaptation pattern data set, it is employed to train the committee of ML algorithms. First, the MLP, the SVM and the M5 techniques are trained individually using the adaptation pattern data set generated. Next, the output of these three ML techniques are combined to produce a training data set for the ML technique that acts as the combiner for the committee.

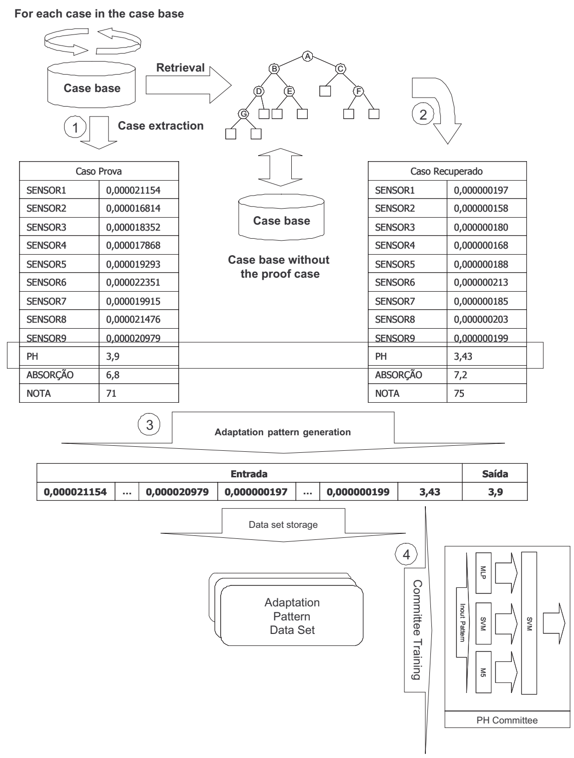

Figure 1. Process of extraction of adaptation knowledge.

For the application of this adaptation approach, the method employs a strategy where one independent adaptation pattern data set and an independent adaptation mechanism must be used for each different component of the case solution structure. This strategy preserves the independence of the approach from the structure of the cases.

The committees investigated are composed of ML algorithms, here named estimators, based on different paradigms. One ML algorithm, here named combiner, combines the outputs of the individual estimators to produce the output of the committee. The estimators and the combiner are used to perform adaptations in the recovered solution to predict wine quality. The following ML algorithms compose the committee:

- Estimators – a *Multi Layer Perceptron* (MLP) neural network [5]; a symbolic learning algorithm M5 [13]; a *Support Vector Machine* (SVM) technique [12], based on the statistical learning theory.
- Combiner: in this work were investigated three ML algorithms as the combiner of the committee: a MLP neural network, the M5 learning algorithm

and the SVM technique. The combiner receives the outputs from the other three algorithms as input, combines the results, and produces the output of the committee.

MLP networks are the most commonly used Artificial Neural Network model for pattern recognition. A MLP network usually presents one or more hidden layers with nonlinear activation functions (generally sigmoidal) that carry out successive nonlinear transformations on the input patterns. In this way, the intermediate layers can transform nonlinearly separable problems into linearly separable ones [5].

M5 is a learning algorithm that generates models on the form of regression trees combined with regression equations (Model Tree) [13]. This model works similarly to a classification tree. However, the leaves contain linear expressions instead of predicted values. The Model Tree is constructed by a divide-and-conquer approach that recursively creates new nodes. This approach applies a standard deviation test to divide the remaining data into subsets and associates the test results to each new node. This process is carried out for all data subsets, creating an initial model. Afterward, a linear model is calculated for each inner node of the tree using a standard regression process. Next, the tree is pruned by evaluating the linear model of each node and its subtrees [10].

SVM is a family of learning algorithms based on statistical learning theory [12]. It combines generalization control with a technique that deals with the dimensionality problem [12]. This technique basically uses hyperplanes as decision surface and maximizes the separation borders between positive and negative classes. In order to achieve these large margins, SVM follows a statistical principle named *structural risk minimization* [12]. Another central idea of SVM algorithms is the use of kernels to build support vectors from the training data set.

4.2. Case Adaptation Mechanism

The proposed case adaptation mechanism allows the learning of the necessary modifications in the components values of the retrieved solutions in order to achieve an adequate solution for a new problem. The most important characteristic of this mechanism is the employment of implicit knowledge obtained from the CB with a minimum effort for the knowledge acquisition.

Figure 2 illustrates the process of case adaptation and problem solving (for the PH component). When a new problem is presented to the system (step 1 on Figure 2), it description is extracted (step 2 on Figure 2) and used for retrieve a similar case stored in the CB by a retrieval mechanism [2] (step 3 on Figure 2). Then, the retrieved case is sent to the adaptation algorithm (module 2 of the adaptation approach) together with the problem description. Next, the adaptation algorithm extracts the problem description and the solution stored in the retrieved case. Then, the algorithm generates an appropriated input pattern for the committee of ML algorithms developed for this component (step 4 on Figure 2). Then, the committee indicates the solution for the new problem (step 5 on Figure 2).

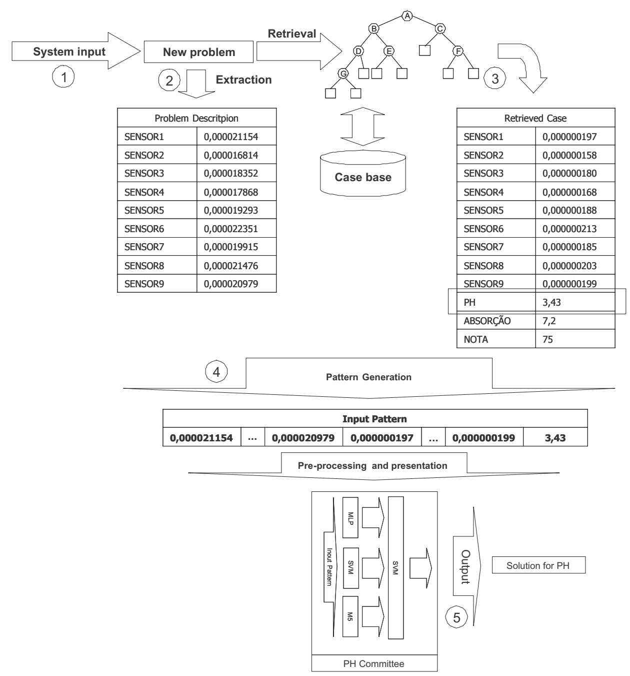

Figure 2. Example of the automatic case adaptation process.

5. Empirical Evaluation

This Section presents a set of experiments carried out to evaluate the performance of the hybrid strategy for automatic case adaptation. For such, the performances obtained with the use of committees of ML algorithms are compared to those obtained by using individual ML algorithms for case adaptation: a MLP network, a M5 algorithm and a SVM technique. In order to show that the automatic case adaptation may result in considerable gain in the prediction of the desired values for the solution attribute, both case adaptation approaches, using committees of ML algorithms and individual ML algorithms, have their performance compared with the performances obtained by the individual ML algorithms and individual committees for the prediction of the solution attribute values.

As previously mentioned, estimators generated by three ML techniques were employed in the experiments. The techniques are: MLP, M5 algorithm and SVM. The estimators generated by the MLP network have 37 input units, a hidden layer with 20 neurons and 1 output neuron.

The MLP networks were trained using the back propagation with momentum algorithm [5], with moment term equal to 0.2 and learning rate equal to 0.3. Different topologies were tried. The M5 algorithm was trained using default parameters. The SVM was trained using the Radial Basis Function kernel and default parameters. The MLP network and the M5 algorithm were simulated using the WEKA library, version 3.2 - which includes a set of Machine Learning algorithms. The SVM was simulated using the LIBSVM tool.

Three different adaptation pattern data sets were created by generating adaptation patterns using the 1, 3, and 5 most similar cases. The data preprocessing was only performed if the retrieval and adaptation mechanisms required it. The cases were stored in the CB in their original format. The numerical values were normalized for the interval $[0.0 \ldots 1.0]$. For the MLP, SVM and M5 techniques, the input attributes with symbolic values were transformed into orthogonal vectors of binary values.

The tests followed the 10-fold-cross-validation methodology. The same set of folds used to train the ML algorithms were employed to generate the adaptation patterns for the training of the case adaptation mechanism. The results presented are the average and standard deviation of the absolute error for the test folds.

The following notation is used to illustrate the results produced by the CBR models: CBR(AA - k). In this notation, AA indicates the approach used for case adaptation (MLP, $M5$, SVM, $CMLP$, $CM5$ or $CSVM$) and k indicates the number of similar cases to the proof case considered during the adaptation patterns generation. For example, $CBR(CSVM - 3)$, means a CBR system employing a committee using the SVM technique as a combiner for case adaptation, trained with adaptation patterns generated using the 3 most similar case to the proof case.

Table 2 has the results of the evaluations carried out with the hybrid CBR systems, with individual classifiers and with committees, using the three settings for the parameter the controls the number of retrieved cases, indicated by the column named K. The results obtained by the individual techniques employed alone (MLP, M5, and SVM) are also shown.

Table 2. Average error results for the proposed approach.

Model	Average Absolute Error		
	Absorption Index	Taste Score	PH
CBR (M5-1)	$11,93 \pm 0,66$	$3,16 \pm 0,21$	$0,18 \pm 0,01$
CBR (M5-2)	$12,20 \pm 0,67$	$3,59 \pm 0,57$	$0,19 \pm 0,02$
CBR (M5-3)	$12,87 \pm 1,03$	$4,07 \pm 0,86$	$0,20 \pm 0,02$
CBR (SVM-1)	$7,90 \pm 0,32$	$2,91 \pm 0,16$	$0,12 \pm 0,01$
CBR (SVM-2)	$7,32 \pm 0,29$	$2,91 \pm 0,16$	$0,11 \pm 0,01$
CBR (SVM-3)	$7,36 \pm 0,30$	$2,91 \pm 0,16$	$0,11 \pm 0,01$
CBR (MLP-1)	$7,55 \pm 1,21$	$4,25 \pm 0,95$	$0,19 \pm 0,02$
CBR (MLP-2)	$13,17 \pm 1,09$	$4,28 \pm 0,76$	$0,20 \pm 0,01$
CBR (MLP-3)	$13,05 \pm 1,92$	$4,63 \pm 0,70$	$0,20 \pm 0,02$
CBR (CM5-1)	$11,53 \pm 0,85$	$4,08 \pm 0,82$	$0,18 \pm 0,01$
CBR (CM5-2)	$12,09 \pm 0,66$	$4,10 \pm 0,72$	$0,19 \pm 0,01$
CBR (CM5-3)	$12,72 \pm 0,68$	$4,84 \pm 0,41$	$0,24 \pm 0,07$
CBR (CMLP-1)	$12,31 \pm 0,63$	$4,04 \pm 0,58$	$0,19 \pm 0,01$
CBR (CMLP-2)	$12,10 \pm 0,86$	$4,13 \pm 0,65$	$0,19 \pm 0,01$
CBR (CMLP-3)	$12,13 \pm 0,69$	$4,43 \pm 0,66$	$0,20 \pm 0,01$
CBR (CSVM-1)	$5,57 \pm 0,31$	$2,31 \pm 0,08$	$0,08 \pm 0,00$
CBR (CSVM-2)	$3,86 \pm 0,33$	$1,58 \pm 0,07$	$0,06 \pm 0,00$
CBR (CSVM-3)	$3,57 \pm 0,31$	$1,29 \pm 0,07$	$0,06 \pm 0,00$
M5	$12,00 \pm 0,71$	$3,17 \pm 0,21$	$0,19 \pm 0,01$
SVM	$8,05 \pm 0,53$	$2,88 \pm 0,15$	$0,13 \pm 0,02$
MLP	$8,71 \pm 1,27$	$4,41 \pm 0,63$	$0,24 \pm 0,07$
CM5	$10,30 \pm 0,72$	$3,06 \pm 0,34$	$0,17 \pm 0,03$
CSVM	$5,73 \pm 0,11$	$2,78 \pm 0,22$	$0,12 \pm 0,01$
CMLP	$7,61 \pm 1,35$	$4,33 \pm 0,63$	$0,23 \pm 0,07$

According to the results for the solution component, the $CBR(CSVM - 5)$ model had better accuracy than individual algorithms and committees without CBR. If the comparison is made among the CBR systems employing individual ML algorithms for case adaptation and the CBR systems employing Committees with the same ML algorithm as a combiner – for instance, $CBR(SVM - 1)$ and $CBR(CSVM - 1)$ – , the committees improved the accuracy of most of the hybrid CBR systems.

The *paired t test* [7] [8] shows that the increase of the number of similar cases considered during the generation of the adaptation patterns (column k) did not, in general, cause significance changes in the accuracy values. The exceptions are the $CBR(SVM - *)$ models, where the increase of k improved the accuracy.

In order to have a statistical estimation of the performance obtained by the hybrid approach, the paired t test for bilateral procedures with 99% of certainty [7] [8] was applied to the results. The relevant conclusions are shown in Table 3.

Table 3. T test conclusions.

Models	Conclusion
CBR(CSVM - 5) and CBR(CSVM - 3)	Similar performance
CBR(CSVM - 5) and CSVM	CBR(CSVM - 5) is better
CBR(CSVM - 5) and CMLP	CBR(CSVM - 5) is better

6. Conclusions

This work investigated the use of a hybrid Case Based Reasoning System for monitoring wine quality. According to the obtained results, the use of a hybrid committee improved the prediction ability and the $CBR(CSVM - *)$ models produced the best accuracies. The later is probably due to the properties of the SVM algorithm. SVMs use a generalization control (inductive bias) and kernel functions that allow the construction of hyperplanes able to create more efficient decision areas. The results also indicate the potential of combining Instance Based Learning with Inductive Learning, suggesting that the adaptation patterns dataset extracted may contain a good representation of the necessary adaptations for the solution components.

Besides, it can be seen in the results that, in general, the committees of ML algorithms improved both accuracy and precision. The average error rates and the standard deviations were usually smaller than those obtained by the hybrid CBR systems where the case adaptation is carried out by individual ML algorithms. Finally, the hybrid CBR proposed employs a process of adaptation pattern generation that can reduce the effort necessary for knowledge acquisition. This hybrid approach is not computationally expensive, since the generation of the adaptation patterns demands no comparisons between solution components. Moreover, the process to obtain an adaptation pattern data set is fully integrated with the case retrieval mechanism and can employ standard retrieval techniques [2]. Therefore, the set of adaptation rules extracted from the case base can provide a useful tool for case adaptation in real-world problems.

7. Acknowledgment

The authors would like to thank CNPq and FAPESP, for their support.

8. References

[1] P. Domingos. ”Unifying instance-based and rule-based induction”. *Machine Learning*, 24:141–168, 1996.

[2] R. Duda, P. hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, 2001.

[3] Y. Freund and R. Schapire. ”Experiments with a new boosting algorithm”. In L. Saitta, editor, *13th. International Conference on Machine Learning, Bari, Italy, July 1996*, pages 148–156. Morgan Kaufmann, 1996.

[4] K. Hanney. ”Learning adaptation rules from cases”. Master’s thesis, University College Dublin, 1996.

[5] S. Haykin. ”Support Vector Machines”. In *Neural Networks: A Comprehensive Foundation. Captulo 6*. Prentice Hall, 1999.

[6] D. Leake. ”CBR in context: The present and future”. In *Case-Based Reasoning: Experiences, Lessons and Future Directions. Captulo 1*. AAAI Press/MIT Press, 1996.

[7] R. Mason, R. Gunst, and J. Hess. *Statistical Design and Analysis of Experiments*. John Wiley & Sons, 1989.

[8] T. Mitchell. ”Evaluating Hypotheses”. In *Machine Learning. Chapter 5*, 1997.

[9] C. Policastro, A. Carvalho, and A. Delbem. ”Hybrid Approaches for Case Retrieval and Adaptation”. In *26th German Conference on Artificial Inteligence. Hamburg, German. Gnter,A. et al (eds.)*. Lecture Notes in Artificial Intelligence. Springer Verlag, 2003.

[10] J. Quinlan. ”Learning with Continuous Classes”. In *5th. Australian Joint Conference on Artificial Inteligence, Hobart, Tasmania, World Scientific, Singapore*, pages 343–348, 1992.

[11] B. Smyth. ”Case Base Maintenance”. In J. Mira and A. Pobil, editors, *12th. International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Cairo, Egypt, May 1999*, pages 507–516. Springer Verlag, 1998.

[12] V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

[13] Y. Wang and I. Witten. ”Induction of model trees for predicting continuous classes”. In M. Someren and G. Widmer, editors, *9th. European Conference on Machine Learning, Prague, Czech Republic, April 1997*, pages 128–137. Springer Verlag, 1997.

[14] N. Wiratunga, S. Craw, and R. Rowe. ”Learning to Adapt for Case-Based Design”. In S. Craw and A. Preece, editors, *6th. European Conference on Case-Based Reasoning, Aberdeen, Scotland, Uk, September 2002*, pages 421–435. Springer Verlag, 2002.