HYDROLOGICAL PROCESSES
Hydrol. Process. **22**, 275–287 (2008)
Published online 24 July 2007 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/hyp.6592

Instance-based learning compared to other data-driven methods in hydrological forecasting

Dimitri P. Solomatine,[1]* Mahesh Maskey[2] and Durga Lal Shrestha[1]

[1] *UNESCO-IHE Institute for Water Education, P.O. Box 3015, 2601 DA Delft, The Netherlands*
[2] *Civil Engineer, NepalConsult (P.) Ltd. G.P.O. Box 492, Gushingal, Lalitpur, Kathmandu, Nepal*

Abstract:

Data-driven techniques based on machine learning algorithms are becoming popular in hydrological modelling, in particular for forecasting. Artificial neural networks (ANNs) are often the first choice. The so-called instance-based learning (IBL) has received relatively little attention, and the present paper explores the applicability of these methods in the field of hydrological forecasting. Their performance is compared with that of ANNs, M5 model trees and conceptual hydrological models. Four short-term flow forecasting problems were solved for two catchments. Results showed that the IBL methods often produce better results than ANNs and M5 model trees, especially if used with the Gaussian kernel function. The study showed that IBL is an effective data-driven method that can be successfully used in hydrological forecasting. Copyright © 2007 John Wiley & Sons, Ltd.

KEY WORDS hydrological modelling; floods; data-driven models; instance-based learning; artificial neural networks; locally weighted regression; k-nearest neighbour method

Received 11 February 2006; Accepted 25 September 2006

INTRODUCTION

In the context of flood management, the accurate forecasting of precipitation, runoff, water stages, etc. are of major focus in hydrological modelling. Modelling techniques can be classified into two large groups: (1) methods based on the detailed description of the physical processes, often referred to as process, physically-based, or simulation modelling (further divided into the more detailed physically-based, and simpler conceptual models) (Sugawara, 1995; Refsgaard, 1996), and statistical and data driven approaches, where a model is built on the basis of historical data (Becker and Kundzewicz, 1987; Solomatine, 2005). An approach where these two approaches are combined can be referred to as hybrid modelling (see, e.g. Solomatine and Price, 2004).

Both physically based and conceptual models need information about the parameters, some of which cannot be measured. Due to this constraint, the data-driven modelling is becoming more and more popular. The adequacy and the value of a data-driven model (DDM) depends on how well a modeller understands the essence of the physical processes being modelled. In the context of rainfall–runoff modelling, DDMs are typically based on the historical records about the relevant input (e.g. rainfall and temperature) and output (e.g. flow) variables, and they make a limited number of assumptions about the details of the processes transforming the rainfall into runoff.

Among the various types of DDMs, an artificial neural network (ANN) is the most popular one—see e.g. Hsu *et al.* (1995); Minns and Hall (1996); Dibike and Solomatine (1999, 2001); Abrahart and See (2000); Maier and Dandy (2000); Dawson and Wilby (2001); Govindaraju and Rao (2000); Cigizoglu (2003). Along with ANN, other numerical prediction (regression) methods are used as well: Solomatine and Dulal (2003) applied the so-called M5 model trees (MTs); Bray and Han (2004) used support vector machines; Solomatine and Xue (2004) used the modular models (committees) comprised of ANNs and M5 model trees.

ANNs or other numerical prediction models that reconstruct complex non-linear dependencies are typically more accurate than other empirical models (e.g. ARIMA or linear regression models) but suffer from a problem of being encapsulated in software codes and therefore not transparent enough, which is an issue during their acceptance by the end-users.

One of the techniques in machine learning that has a potential to resolve the issue of non-transparency is instance-based learning (IBL) when prediction is made on the basis of combining historical example (instances) that are in some way close to the new vector of inputs. In the hydrological literature one can find quite a limited number of references to this class of methods. Practically all of them refer to one method—the k-nearest neighbour (k-NN) method. Karlsson and Yakowitz (1987) were probably the first to use this method in hydrology, focussing, however only on (single-variate) time series forecasts. Given time series $\{x_t\}$, $t = 1, \ldots, n$, they generated d-dimensional vectors $\mathbf{x}^{\mathrm{d}}(t) = \{x_t, x_{t-1}, \ldots, x_{t-d+1}\}$ (for $t = d, \ldots, n-1$) and based the prediction of x_{T+1}

* Correspondence to: Dimitri P. Solomatine, UNESCO-IHE Institute for Water Education, P.O. Box 3015, 2601 DA Delft, The Netherlands. E-mail: d.solomatine@unesco-ihe.org

on averaging the values x_{t+1} that corresponded to the k vectors in this d-dimensional space that are close to $\mathbf{x}^{\mathrm{d}}(T)$ (Interestingly, their approach has an intuitive relation to the single-variate predictors based on non-linear dynamics and chaos theory, which however provides a much more solid foundation for this type of prediction). Galeati (1990) demonstrated the applicability of the k-NN method (with the vectors composed of the lagged rainfall and flow values) for daily discharge forecasting and it compared favourably to the ARX statistical model. Shamseldin and O'Connor (1996) used the k-NN method in adjusting the parameters of the linear perturbation model for river flow forecasting. Toth *et al.* (2000) compared the k-NN approach to other time series prediction methods in a problem of short-term rainfall forecasting.

In the present paper IBL is considered in a wider context of machine learning, several methods are explored, their applicability in short-term hydrological forecasting is tested and their performance is compared to other methods on the two case studies. Since M5 model trees are used in comparison and since they are not yet widely known in the hydrological community, this machine learning method will be introduced as well.

DATA-DRIVEN (MACHINE LEARNING) MODELS

In this paper using a DDM the following model will be understood:

$$y = f(X) \tag{1}$$

where $f =$ machine learning (e.g. ANN) or statistical (e.g. linear regression) model which internal parameters are found by calibration (i.e. training, or optimization); $y =$ scalar (typically, real-valued) output; $X \in R^n$ (n-dimensional real-valued input vector). Calibration (training) is done on a set T of instances (examples) for which both input and output values are known. When the model is trained, it can be used to predict the output value y (also called target value) for a new (unseen) input vector X_q. The model is tested (verified) by feeding the set V of the input vectors (verification data set) that do not belong to T and for which the output measured values are known as well. The predicted output values are compared to the measured ones, some error functions, e.g. root mean square error (RMSE) and/or volumetric fit, are calculated, and these values serve as the indicators of the model performance. Some considerations on how to build sets T and V are given in the section covering case studies.

For an example of a DDM that can be used for hydrological forecasting, it is possible to turn to the paper by Solomatine and Dulal (2003) where several machine learning models were built to predict the river flows Q_{t+H} several hours ahead (prediction horizon $H = 1$, 3 or 6):

$$Q_{t+H} = f(RE_{\mathrm{t}-\tau r}, Q_{t-\tau q}) \tag{2}$$

where $RE_{\mathrm{t}-\tau r} =$ previous values of rainfall, $Q_{t-\tau q} =$ previous (lagged) values of flow; $\tau r \in [0, 5]$ hours, $\tau q \in [0, 2]$ hours; (see also Equations (9) and (10)). The values of lags τr for rainfall and τq for flow are based on the hydrological analysis of the catchment, and on the analysis of correlation and average mutual information between inputs and outputs (for τr) or autocorrelation of flow (for τq). In the notation of the model in Equation (1) dimension n of the input vector X is equal to the total number of the lagged vector of rainfall and lagged vector of flow values used as inputs. For example, for the prediction horizon $H = 3$ (see Equation (10)) vector $X = \{RE_{\mathrm{t}}, RE_{\mathrm{t}-1}, RE_{\mathrm{t}-2}, RE_{\mathrm{t}-3}, Q_{t-1}, Q_t\}$, so $n = 6$. The training set T is composed of all the past records of the properly lagged values of rainfall and flow arranged as six-dimensional vectors, each accompanied by the value of the measured value of flow H time steps ahead.

M5 MODEL TREES

Previous research (Solomatine and Dulal, 2003; Solomatine and Xue, 2004) found that the so-called model tree built by the M5 algorithm of Quinlan (1992) can be an effective tool for hydrological modelling. The essence of this method is in splitting the training data into subsets (and accordingly splitting the input space into regions), and building separate regression models for each of them. During the model operation, the input vector is attributed to one of the regions, the corresponding specialized model is run. The training set is split progressively, and the result is a hierarchy, a binary tree, with the splitting rules in non-terminal nodes and the models in leaves. Each such model is a linear regression model, however the overall modular model is piece-wise linear and hence non-linear. One can find a clear analogy between MTs and a combination of linear models used in dynamic hydrology already in the 1970s—a notable paper on multilinear models is by (Becker and Kundzewicz (1987). The M5 model tree approach, which is based on the principle of information theory, makes it possible, however, to partition the multi-dimensional parameter space and to generate the optimal models automatically according to an overall quality criterion; it also allows for varying the number of models. The optimised versions of the M5 algorithm were recently proposed by Solomatine and Siek (2006).

INSTANCE-BASED LEARNING (IBL)

Many machine learning methods following the so-called "eager learning" approach, construct a general explicit description of the target function when training examples are provided. In contrast, IBL referred to as "lazy learning" simply stores the presented training data and when a new input vector is presented, a set of similar related instances is retrieved from memory and their corresponding outputs are used to predict the output for the new query vector (instance). IBL algorithms are derived from the nearest neighbour pattern classifier (Cover and Hart, 1967; Aha *et al.*, 1991; Mitchell, 1997).

DOI: 10.1002/hyp

IBL methods, in fact, construct a local approximation to the modelled function as shown in Equation (1) that applies in the neighbourhood of the new query instance encountered, and never construct an approximation designed to perform well over the entire instance space. Thus it describes the very complex target function as a collection of less complex local approximations. It often gives competitive performance when compared with more modern methods such as decision trees and ANNs (Mitchell, 1997). Note that function f in Equation (1) is, in fact, never explicitly built.

IBL algorithms have several advantages: they are quite simple but robust learning algorithms, can tolerate noise and irrelevant attributes, and can represent both probabilistic and overlapping concepts and naturally exploit inter-attribute relationships (Aha *et al.*, 1991). However, for classification of each new instance when the training set is large, IBL can be quite time-consuming requiring order of $|T| \times n$ attribute examinations, where T is the training set and n is the number of attributes used to describe the instances (i.e. dimension of input in Equation (1)).

k-Nearest neighbours (k-NN)

The nearest neighbour classifier (Mitchell, 1997) is one of the simplest and oldest methods for classification. It classifies an unknown input vector X_q (denoted further also as q) by choosing the class of the nearest example X in the training set as measured by a distance metric, typically Euclidean.

Generalization of this method is the k-NN method. For a discrete valued target function, the estimate will just be the most common value among k training examples nearest to q. For real valued target functions, the estimate is the mean value of the k-nearest neighbouring examples. The k-NN algorithm can be improved by weighting each of the k-neighbours X_i according to their distance to the query point q so that the output value for q is calculated as follows:

$$f(q) = \sum_{i=1}^{k} w_i f(X_i) / \sum_{i=1}^{k} w_i \qquad (3)$$

where weight w_i is a function of the distance $d(X_q, X_i)$ between X_q and X_i. Typically, the following weight functions are used:

$$(a)\text{linear}: w_i = 1 - d(X_q, X_i) \qquad (4)$$
$$(b)\text{inverse}: w_i = (d(X_q, X_i))^{-1}$$
$$(c)\text{inverse square}: w_i = (d(X_q, X_i))^{-2}$$

In Weka software (Witten and Frank, 2000) used in this research the functions (a) and (b) are implemented.

Locally weighted regression (LWR)

Locally weighted regression (LWR) is inspired by the instance-based methods for classification (Atkeson *et al.*, 1996). In it, the regression model is built only when the output value for a new vector q should be predicted, so that all learning is performed at prediction time. It uses linear or non-linear regression to fit models locally to particular areas of instance space in a way quite different from M5 model trees. The training instances are assigned weights w_i according to their distance to the query instance q and regression equations are generated on the weighted data.

A number of distance-based weighting schemes can be used in LWR (Scott, 1992). A common choice is to compute the weight w_i of each instance X_i according to the inverse of their Euclidean distance $d(X_i, X_q)$ from the query instance q as given in Equation (4):

$$w_i = K(d(X_q, X_i)) = (d(X_q, X_i))^{-1} \qquad (5)$$

where $K(.)$ is typically referred to as the kernel function.

Atkeson *et al.* (1996) combined Euclidean distance with the Gaussian kernel function:

$$w_i = K(d(X_q, X_i)) = \exp(-d(X_q, X_i)^2 \qquad (6)$$

Alternatively, instead of weighting the data directly, the model errors for each instance used in the regression equation are weighted to form the total error criterion $C(q)$ to be minimized:

$$C(q) = \sum_{i=1}^{|T|} L(f(X_i, \beta), y_i) K(d(X_i, X_q)) \qquad (7)$$

where $f(X_i, \beta) =$ regression model giving an output value estimate y_i; $L(y_i{}^*, y_i) =$ error function (typically the sum of squared differences $(y_i{}^* - y_i)^2$ between the target $y_i{}^*$ and estimated y_i output values); β is a vector of parameters to be identified; $i = 1 \ldots |T|$; $T =$ training set.

Gasser and Muller (1979), Cleveland and Loader (1994) and Fedorov *et al.* (1993) addressed the issue of choosing weighting (kernel) functions: it should be maximum at zero distance, and the function should decay smoothly as the distance increases. Discontinuities in the weighting functions lead to discontinuities in the predictions, since training points cross the discontinuity as the query changes.

Yet another possibility to improve the accuracy of LWR is to use the so-called smoothing, or bandwidth parameter that scales the distance function by dividing it by this parameter (Scott, 1992; Cleveland and Loader, 1994). One way to choose it is to set it to the distance to the kth nearest training instance, so that its value becomes smaller as the volume of training data increases. Generally, an appropriate smoothing parameter is found using cross-validation. One can see certain analogy between LWR and the radial-basis function ANNs.

COMBINING MODELS: COMMITTEES AND COMPOSITE MODELS

Combination of classification or regression models often brings improvements in accuracy (e.g. Wolpert, 1992;

DOI: 10.1002/hyp

Weiss and Indurkhya, 1995; Kuncheva, 2004). Solomatine and Price (2004) and Solomatine and Siek (2006) distinguish between (1) modular models when separate models are trained on different subsets of input data, (2) committees (ensembles) of models when they are trained on the same data set and the results are combined by some "voting" scheme, and (3) complementary (composite) models when one is used to correct errors of another. Solomatine and Xue (2004) used the first approach (mixtures of models) in the flow predictions in the Huai river basin (China). In this study the third approach is used where a M5 model tree is complemented by an instance-based model.

One of the methods combining various models is that of Quinlan (1993)—it combines IBL with M5 model trees with and is further referred as "composite model". This method is implemented in the Cubist software (http://www.rulequest.com) and its essence is as follows. For an unseen example q, the target value is to be predicted. A subset of input vectors (prototypes) $\{X_1, X_2, \ldots, X_k\}$ would first be identified as nearest to q. In a standard IBL method the known values $\{f(X_1), f(X_2), \ldots, f(X_k)\}$ would be combined to give the predicted value of the unseen example q. In the composite model, however, these values are adjusted in the following way. Among such prototypes, one can be selected, say X_i. Now some model (Quinlan (1993) suggests M5 model tree) is used to predict target values so that its predictions for q and X_i are $f^*(q)$ and $f^*(X_i)$ respectively. Instead of $f(X_i)$, the adjusted value $f(X_i) - (f^*(X_i) - f^*(q))$ is used in IBL predictor. Such an approach is quite general and may involve any IBL method and any predictive model.

CASE STUDIES

In the present study two problems of hydrological forecasting for Bagmati and Sieve catchments were considered.

Bagmati catchment

Bagmati catchment lies in the central part of Nepal. It is a medium sized foothill fed river basin (see Figure 1) with an area of about 3700 km^2 (catchment area in the hydrometric station at Pandheradobhan is about 2900 km^2). It originates from the southern slope of Shivapuri lake (Mahabharat within Kathmandu valley) and stretches to the plains of Terai (ending at Nepal–India border). The catchment covers eight districts of Nepal and is a perennial water body of Kathmandu. The problem was posed as a short-term flow forecasting at Pandheradobhan hydrometric station.

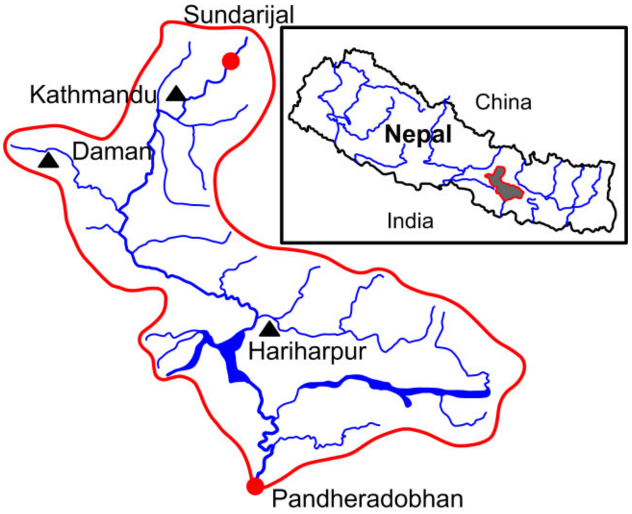

Figure 1. Bagmati catchment. Triangles denote the rainfall stations; the flow is measured at the Pandheradobhan gauge station

Time-series data of rainfall of three stations (Kathmandu, Hariharpurgadhi and Daman) within the basin with the daily resolution for 8 years (1988–1995) was collected. Daily flows were recorded only from one station so this precluded modelling the routing. In order to be able to use the results of lumped conceptual hydrological modelling performed for this catchment earlier (Shrestha, 2003) the mean rainfall was calculated using Thiessen polygons. (It is planned in the future to extend the modelling exercise and to include the available rainfall data from all stations.) On the basis of mean rainfall the daily evapotranspiration was computed using the modified Penman method recommended by FAO (Allen *et al.*, 1998).

Analysis of the relationships between the input and output variables was done by visual inspection and the correlation and average mutual information analysis. By visual inspection of several precipitation events the maximum value of peak-to-peak time lags of rainfall and runoff was found to be close to 1 day. The cross-correlation analysis of the rainfall and runoff gave the maximum correlation of 0·78 for 1 day lag, so this lag was accepted as the average lag time of rainfall (Figure 2a). This value of this lag is also consistent with the average mutual information analysis. Autocorrelation function of runoff drops rapidly within three time steps (days). Based on this analysis, the forecasting model had five input variables to predict the flow one time step ahead:

$$Q_{t+1} = f(RE_{t-2}, RE_{t-1}, RE_t, Q_{t-1}, Q_t) \qquad (8)$$

where Q represents discharge; RE effective rainfall (mean rainfall minus evapotranspiration).

An important problem is splitting of data into training and testing data sets. Ideally, these sets should include approximately equal number of precipitation events, have similar distribution of the low and high flows, or, in other words, the input and output variables should be statistically similar—have similar distributions, or at least mean, variance and range. This can be achieved, e.g. by randomization. This is a standard practice in machine learning to increase generalization ability by ensuring that the training and test sets are independent and identically distributed (i.i.d.).

DOI: 10.1002/hyp

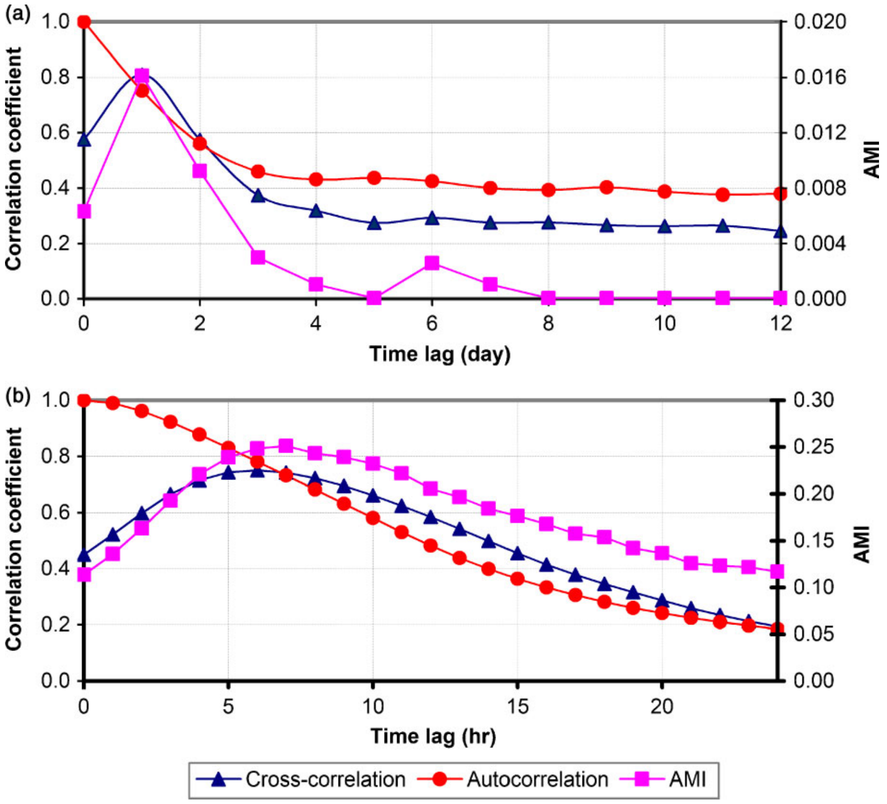

Figure 2. Cross correlation and average mutual information (AMI) of rainfall with discharge, Q_{t+1}, and autocorrelation of discharge, Q_{t+1}, for (a) Bagmati data and (b) Sieve data

However, if the results need to be compared with the physically based model or to produce a hydrograph, a constraint typical to hydrological studies is that the test data should be a set of points contiguous (adjacent and sequential) in time. This makes the generation of the training and test sets with the similar statistical properties not an easy task and leaves not too many choices. For the present study the 8 years of data (2919, five-dimensional vector instances) were split as follows: the first 919 vectors (1 January 1988 to 7 July 1990) were used as the testing data set while the remaining data (8 July 1990 to 30 December 1995) were used for training. Table I presents the statistical properties of the data set. Since this split was not ideal, yet another split into training and test sets was also made using a procedure ensuring close statistical proximity between these sets, but not allowing

Table I. Statistical properties of stream-flow of the data sets

Data set	Statistical properties	Data	Training	Test
Bagmati original				
	Period day/month/year (no. of data)—daily	03/01/1988– 30/12/1995 (2919)	08/07/1990– 30/12/1995 (2000)	03/01/1988– 07/07/1990 (919)
	Average	149·96	159·6	129·4
	Minimum	5·10	5·1	8·20
	Maximum	5030	5030	2110
	Standard deviation	271·12	288·2	228·9
Bagmati randomized				
	No. of data	2919	2000	919
	Average	149·96	150·9	148·2
	Minimum	5·10	5·1	5·1
	Maximum	5030	5030	2470
	Standard deviation	271·12	285·1	238·5
Sieve Q_{t+1}/*Sieve* Q_{t+3}				
	Period hour/day/month/year (no. of data)—hourly	07 : 00/01/12/1959– 00 : 00/28/02/1960 (2154)	09 : 00/03/12/1959– 00 : 00/28/02/1960 (1854)	07 : 00/01/12/1959– 18 : 00/13/12/1959 (300)
	Average	54·9	53·1	66·8/66·5
	Minimum	10·7	10·7	17·6/13·7
	Maximum	752·6	752·6	299·9
	Standard deviation	70·2	73·0/72·9	48·7/48·8

DOI: 10.1002/hyp

for contiguity in time. (Note that non-contiguity in time means that the rows in the data matrix are not contiguous, but, of course, the input variables like RE_{t-2}, RE_{t-1}, RE_t, etc. are always contiguous in time.)

Sieve catchment

The second case study addressed the Sieve catchment. Sieve is the tributary of the Arno River (Figure 3) and is located in the Central Italian Apennines, Italy. The basin covers mostly hills, forests and mountainous areas except in the valley with an average elevation of 470 m above sea level. The cathcment has an area of 822 km^2. A problem of short-term flow forecasting (1 and 3 h ahead) was posed.

For Sieve catchment, 3 months of hourly runoff discharge, precipitation and potential evapotranspiration data were available (December 1959 to February 1960), which represent various types of hydrological conditions. The discharge data were available at the closure section of Fornacina. The spatial average of hourly rainfall from 11 raingauge stations was calculated by Thiessen polygon method and hourly evapotranspiration data were calculated using radiation method (Solomatine and Dulal, 2003). The Arno basin that includes the Sieve catchment has been extensively studied in various modelling exercises (Todini, 1996; Marsigli *et al.*, 2002). In a recent study Solomatine and Dulal (2003) compared the performance of a number of data-driven rainfall–runoff models using ANNs and M5 model trees.

Visual inspection of a number of rainfall events makes it possible to approximately identify the time lags between several peak rainfall and runoff, which is between 5 and 7 h. Additional analysis of lags was performed using the average mutual information and cross-correlation analysis of rainfall and runoff (Solomatine and Dulal, 2003). The cross-correlation between the rainfall and runoff is increasing with the lag, reaches maximum (0·75) when the lag is 6 h and then starts decreasing (Figure 2b). Such analysis helps in choosing the lags for effective rainfall so that the corresponding time range would permit to take into account rainfalls taking place far from the point where runoff is measured. After some experiments it was decided to use six lagged values of RE_t.

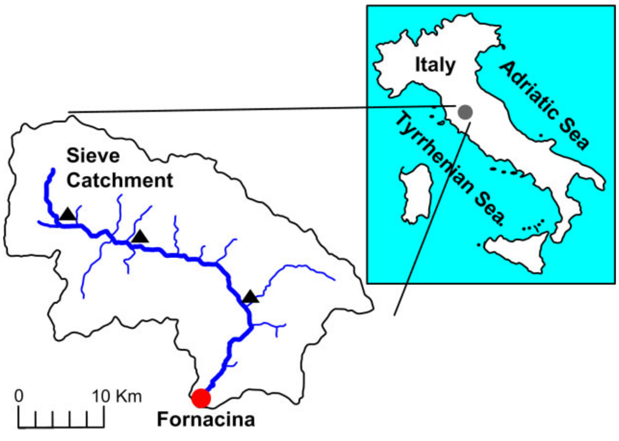

Figure 3. Sieve catchment. Triangles show the rainfall stations; the flow is measured at the Fornacina gauge station

Choosing the number of the lagged discharges was based on analysing its autocorrelation which is 0·989 for 1-h lag and drops as the lag increases. After several experiments two lagged values were left for the final models.

Two models were built: to predict flow at one time step ahead Q_{t+1} (with eight inputs), and at three-time steps ahead Q_{t+3} (with six inputs):

$$Q_{t+1} = f(RE_t, RE_{t-1}, RE_{t-2}, RE_{t-3}, RE_{t-4}, RE_{t-5}, Q_{t-1}, Q_t) \tag{9}$$

$$Q_{t+3} = f(RE_t, RE_{t-1}, RE_{t-2}, RE_{t-3}, Q_{t-1}, Q_t) \tag{10}$$

(These sets of inputs were selected after a number of experiments with various combinations of variables, including moving averages of effective rainfall.)

In making the decision on how to perform the split into the training and test set, the same considerations as given for the Bagmati case study were applied. As it has been mentioned earlier, the split that is conditioned by the necessity of having contiguous data in the test set and it is quite traditional for many hydrological studies. In spite of the fact that the data series covers only 3 months, several rainfall events could be identified, as well as the corresponding low- and high-flow conditions. The split was made in such a way that both low-flow and high-flow conditions were included into the both data sets. Records between 13 December 1959, 7 : 00 p.m. and 28 December 1959, 12 : 00 a.m. were used to form the training data set, and the first 300 records (1 December 1959, 07 : 00 a.m. to 13 December 1959, 6 : 00 p.m.) comprised the test data (for both forecasting horizons).

MODELS SET UP

Lazy learning or IBL

For IBL algorithms one of the main parameters is the number of nearest neighbours. In the present study, experiments were carried out with 1, 3, 5 and 9 numbers of the nearest neighbours (the latter number is the upper limit in Cubist software). For distance-weighted k-NN implemented in Weka software (further abbreviated as IBK), the first two weight functions shown in Equation (4) were used. For locally weighted regression, apart from these two functions, the Gaussian kernel function was used as well.

Eager learning

For comparison purpose different types of eager learning methods such as model trees of Weka and Cubist software and ANNs were used as well. Cubist implements a proprietary software realization of the M5 algorithm of Quinlan (1992): it builds rules based on M5 model trees and these model will be referred to as MT(C). Note that

DOI: 10.1002/hyp

MT and MT(C) present basically the same M5 model tree algorithm, but they are implemented slightly differently in the software packages. In model tree default pruning factor (it is the only one parameter that controls the complexity of the model and hence the performance) of two was considered.

In ANNs, multi-layered perceptron (MLP) network trained by the Levenberg–Marquardt (Haykin, 1999) optimization method algorithm was used for Bagmati data because of its fast convergence. The hyperbolic tangent function was used for the hidden layer with linear transfer function at the output layer. The number of epochs was set to 500. One hidden layer with four hidden nodes was used. The values for learning and momentum rates were set to 0·1 and 0·7, respectively. For Sieve data, the results obtained by Solomatine and Dulal (2003) were presented. They used the backpropagation algorithm with momentum rule for 5000 epochs and training was stopped when the mean squared error (MSE) reached threshold of 0·0001. Their ANN was composed of one hidden layer with six hidden nodes for 1 h ahead prediction, and five hidden nodes for 3 h head prediction.

Composite model

A composite model (Quinlan, 1993) combining rules from an eager learning method (model tree) and lazy learning method (k-NN) was used as well. The number of neighbours was set the same as that of IBL methods (Table II).

EXPERIMENTAL RESULTS AND DISCUSSION

Time needed for model development and execution

Model development time for the DDMs was approximately the same and was only several hours for each of the models considered. However, the results of the data analysis and data preparation performed during the previous studies were used, which for one catchment may take several days. We used a PC with the Pentium III processor running at 600 MHz. Training of ANN takes typically from 5 to 30 min, of MT—only 4 s. Execution time on verification data set is negligible (less than 0·5 s for both models). IBL algorithms in execution needed from several seconds to several minutes, depending on the size of the data set. The development and execution time of DDMs is approximately similar to that for the conceptual lumped hydrological model, and it is much smaller than the development time of the distributed hydrological models needing a lot of effort in data collection and setting it up.

Table II. Models set up

Methods	Parameters of methods
IBK	$k = 1, 3, 5, 9$; linear and inverse weight functions
LWR	$k = 1, 3, 5, 9$; linear, inverse and Gaussian kernel weight functions
ANN	FFBP, learning rate = 0·1, momentum = 0·7
MT MT(C)	Pruning factor = 2
Composite	$k = 1, 3, 5, 9$

Results

Bagmati catchment. For this catchment the results reported by Shrestha (2003) on hydrological conceptual modelling with Sugawara's Tank model (Sugawara, 1995) and ADM model (Franchini, 1996) both run in simulation mode without updating were also included. The conceptual models were calibrated using GLOBE software for global optimization (http://www.datamachine.com). Two data sets were used to predict flow at 1 day ahead (Q_{t+1}) with the same sets of input attributes. The first experiment involved the original data set. Further, all these models were built for the randomized training set to ensure the statistical proximity of the training and test sets, as adopted in most machine learning studies.

Table III compares the results of IBL methods with other data driven methods and conceptual models. LWR and IBK gave the best performance as compared to the other DDMs. Their performance is comparable for the original data set, but LWR is superior when training set was randomized. If compared to the conceptual models the performance of the DDMs is not satisfactory; note, however, that the conceptual models are run in simulation and not in predictive mode. The results were also compared with naïve (no-change) prediction which represents a good bottom line benchmark against which one step ahead prediction can be measured. It is seen that RMSE of naïve model is higher than those of IBL. Comparing the performance of IBL methods to different weight functions it was observed that LWR with Gaussian kernel function and IBK with inverse distance gave best performance. The performance of the IBL methods is also compared with different numbers of neighbours and found that nine numbers of neighbours gave the best performance. The results shown in Table III and Figure 4 are using nine numbers of neighbours for IBL methods (LWR with Gaussian kernel weight function and IBK with inverse distance) and five numbers of neighbours for composite model.

Figure 4 shows the comparison of various instance-based learners. The accuracy of the IBK, LWR and composite models are comparable, except for some points. In a number of occasions all models are late in reacting to the sudden increase of rainfall, but note that the highest peak is predicted very well. Figure 5 shows the scatter plot of the observed and predicted discharges by the three models for randomized data. The overall accuracy of all models, given the fact that the areal average (not distributed) rainfall was used as input, is quite satisfactory.

DOI: 10.1002/hyp

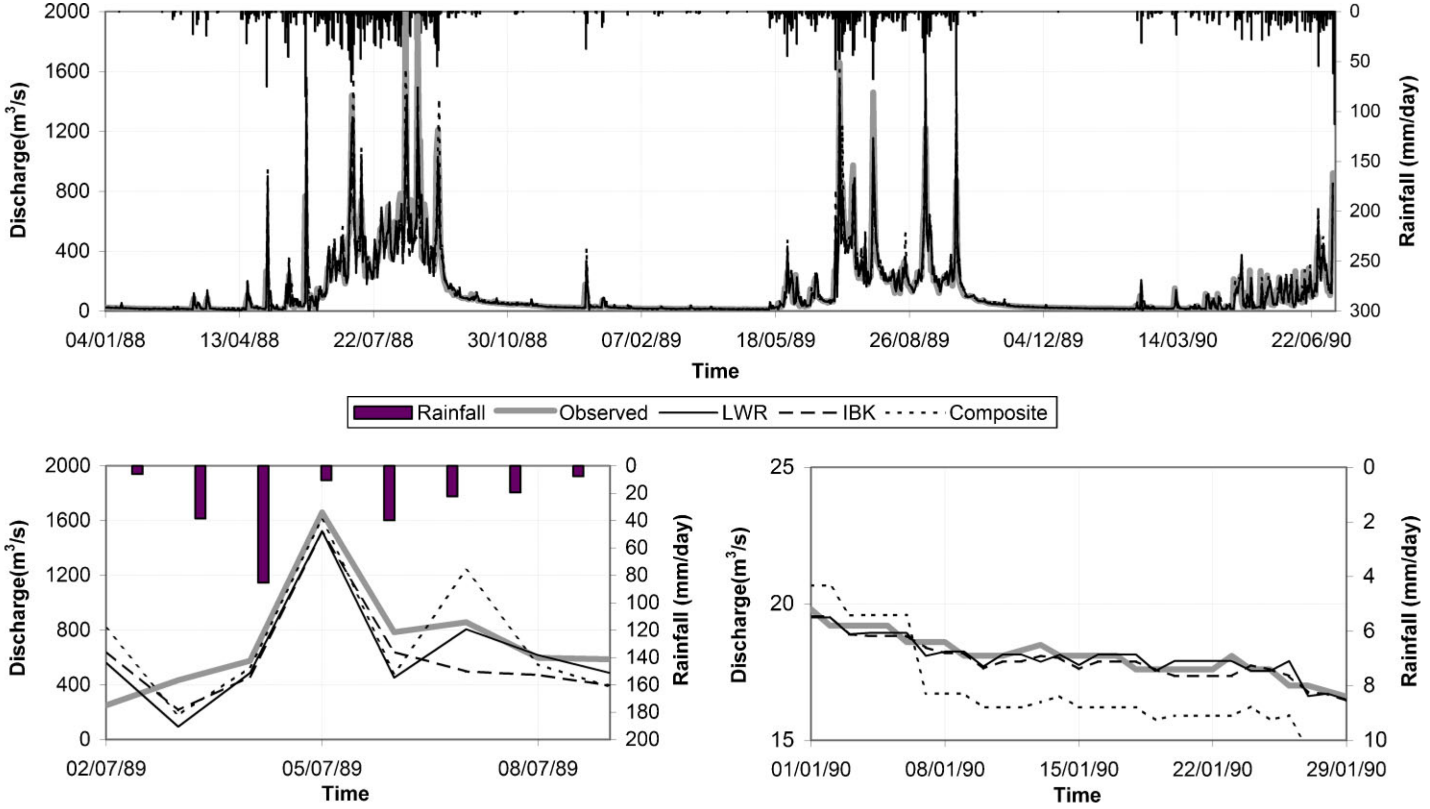

Figure 4. Performance of instance-based learner on test set of Bagmati original data. The lower figures represent parts of the test data zoomed at the peak discharge and the base-flow, respectively

Table III. Comparison of model performance in terms of RMSE for Bagmati data sets

Methods	Original		Randomized	
	Training	Testing	Training	Testing
Tank model	149·2	105·9	—	—
ADM model	101·7	93·0	—	—
ANN	93·4	111·4	101·9	110·6
MT	109·3	112·6	106·5	113·8
LWR	87·4	108·3	87·7	*102·3*
IBK	*36·9*	*107·1*	*37·3*	121·9
MT(C)	104·5	117·9	102·7	111·3
Composite	89·6	110·8	86·5	117·2
naïve	209·94	141·4		

Note: Italic type indicates the minimum value of RMSE for each data set.

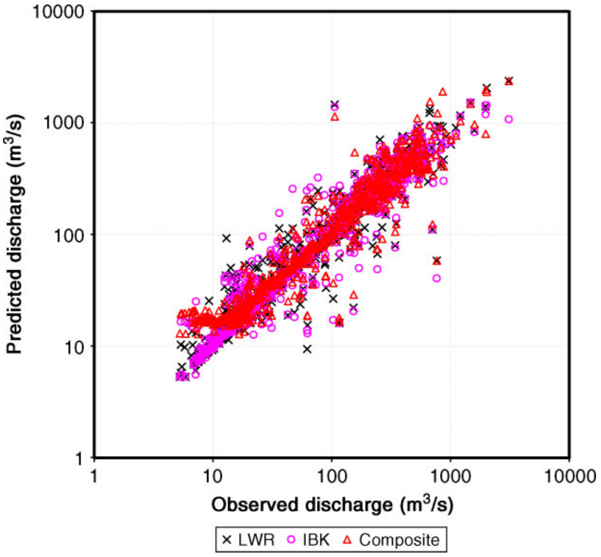

Figure 5. Scatter plot for performance of instance-based learner on test set of Bagmati randomized data

In order to give an idea of the structure of rules generated by MT(C) model (Cubist software) on the basis of the built M5 model tree, the three rules (out of nine) are presented later. The algorithm splits the data set into subsets (the number of examples is given in parentheses) and builds linear regression models for each of them. In fact, the splitting into such subsets often has a reasonable hydrological interpretation of representing various types of hydrological conditions—low flows, high past precipitation and low current flow, high flows, etc.

1. If $Q_t \le 222$ then

$$Q_{t+1} = 1{\cdot}317 + 0{\cdot}96Q_t + 3{\cdot}1\ RE_t(1512 \text{ examples})$$

2. if $RE_t \le 15{\cdot}56$ and $222 \le Q_t > 40{\cdot}1$ then

$$Q_{t+1} = 3{\cdot}55 + Q_t + 2{\cdot}9\ RE_t(553 \text{ examples})$$

3. if $RE_t \le 34{\cdot}19$ and $297 \le Q_t > 222$ then

$$Q_{t+1} = 71{\cdot}687 + 0{\cdot}68Q_t + 1{\cdot}3\ RE_t + 0{\cdot}03Q_{t-1}(137 \text{ examples})$$

Sieve catchment. For Sieve catchment, two models were built: for 1 and 3 h ahead predictions of flow (respectively Q_{t+1} and Q_{t+3}). Table IV compares the performance of various models using nine nearest neighbours with Gaussian kernel weight function for LWR, linear and inverse distance function for IBK. For Q_{t+1}, MT(C) shows better performance than the other methods. For Q_{t+3}, ANN is the best (it is 9·53% more accurate than MT and 11·02% more accurate than LWR). Figure 6 compares the three IBL methods. In predicting Q_{t+1} all of them are reasonably accurate, the best method being the composite model with nine neighbours. The other two methods (LWR and IBK) have higher errors on the peaks. In predicting Q_{t+3} the results are, understandably, less accurate (Figure 7).

Uncertainty of forecasts

No demonstrable systematic bias was associated with any of the presented methods. Statistical analysis of forecasts' uncertainty for the reported case studies was addressed in a separate study and is presented by Shrestha and Solomatine (2006).

Discussion

The performed experiments showed that the results of IBL methods are comparable with those of other data driven methods. Concerning the particulars of the model structure, in LWR the Gaussian kernel function, and in IBK (*k*-NN method) the inverse weighted distance were the best choices.

Is it possible to make a universal judgement about the appropriateness of this or that IBL method for hydrological forecasting? The authors do not think so. Any machine learning (data-driven) method can excel on one data set and show a meagre performance on another, and there are no universally applicable rules for selection of the method that would be best in all cases. For the presented experiments it can be said that the IBL

Table IV. Comparison of model performance in terms of RMSE for Sieve data sets

Methods	Q_{t+1}		Q_{t+3}	
	Training	Testing	Training	Testing
ANN	5·826	5·175	*13·55*	*11·3*
MT	*4·550*	*3·612*	14·38	12·5
RT	10·29	10·574	12·94	16·5
LWR	9·097	10·67	12·56	12·7
IBK	12·56	12·70	17·57	13·7
MT(C)	4·525	3·215	13·42	13·9
Composite	4·674	3·350	13·30	14·6
naïve	10·545	6·661		

Note: Italic type indicates the minimum value of RMSE for each data set.

DOI: 10.1002/hyp

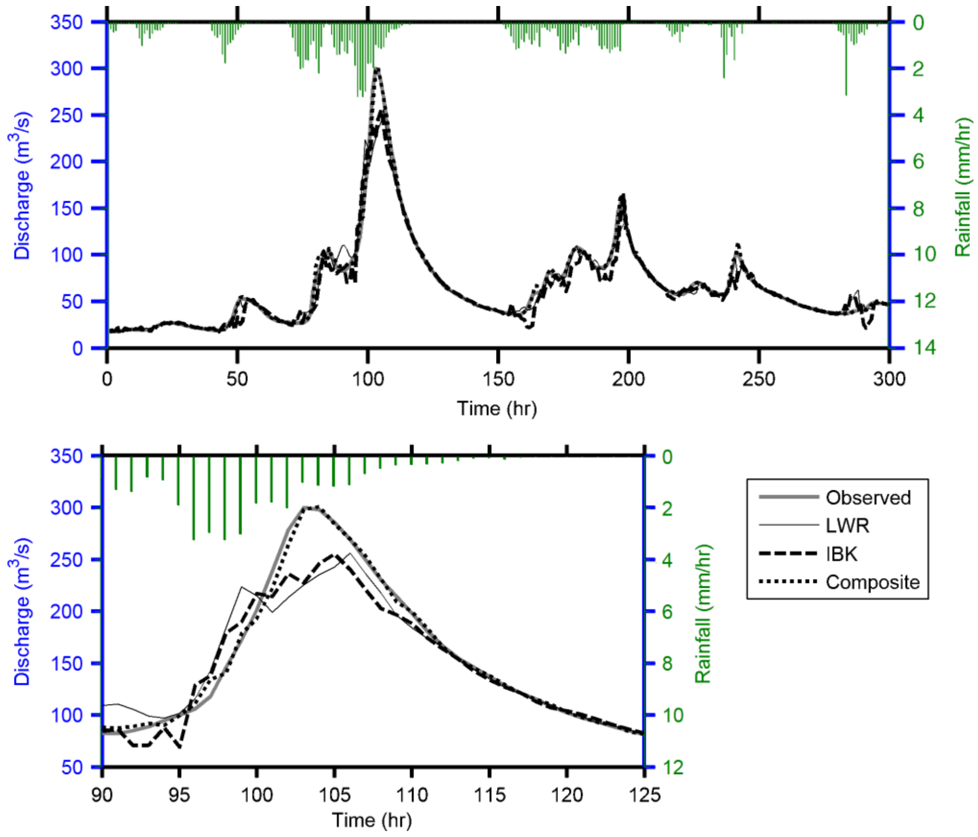

Figure 6. Performance of instance-based learners on test set of Sieve data for 1 h ahead prediction (Q_{t+1}). The lower figures represent parts of the test data zoomed at the peak discharge. The hourly data corresponds to the period from 07 : 00 1 December 1959 to 18 : 00 13 December 1959

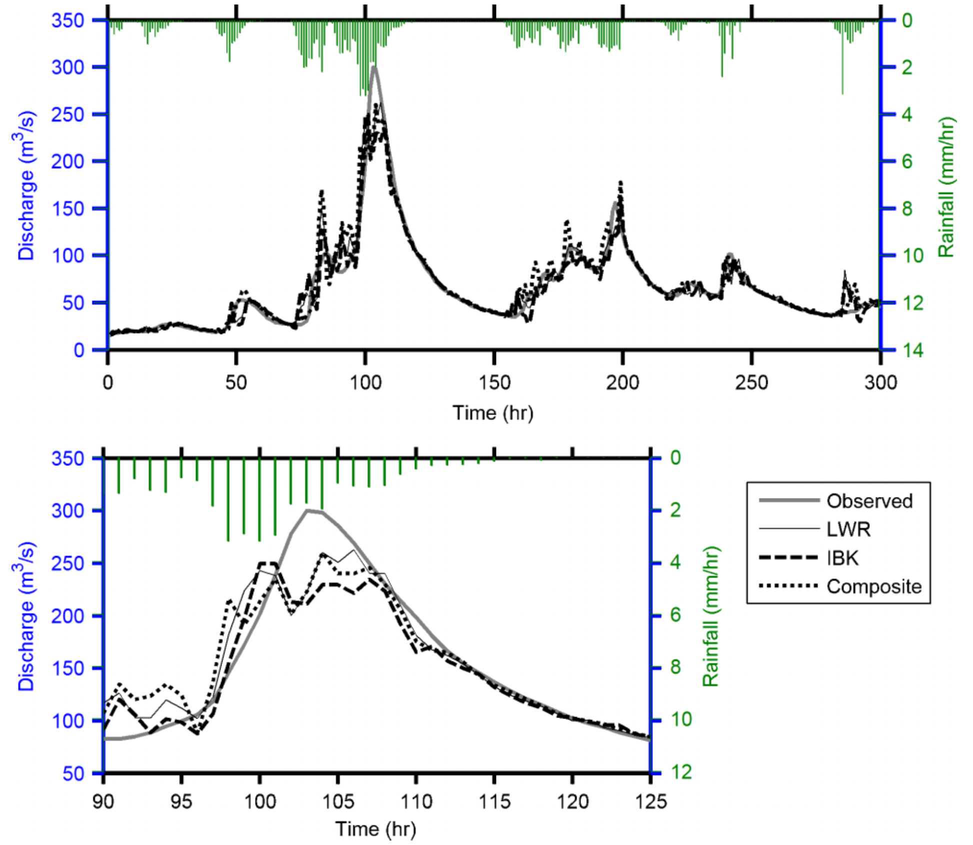

Figure 7. Performance of instance-based learners on test set of Sieve data for 3 h ahead prediction (Q_{t+3}). The lower figures represent parts of the test data zoomed at the peak discharge. The hourly data corresponds to the period from 07 : 00 1 December 1959 to 18 : 00 13 December 1959

methods and the M5 model trees perform the "local" modelling, i.e. they use models based on subsets of the whole data set. In the presented hydrological problems these methods have superior performance if compared with the "global" methods where models are trained on the whole data set (ANNs). In hydrological context this may mean that the modelled processes consist, in fact, of a number of different processes (e.g. resulting in low, medium and high flows) each of which should, in principle, be modelled separately.

The essence of using IBL methods for hydrological forecasting is in following a simple idea: use the flow value (or a function of several values) that resulted from the similar hydrological situations in the past. In this respect it is of utmost importance to choose the relevant input variables which are properly lagged—this is where the knowledge of hydrology of the catchment comes into play and directly used in a DDM. Ideally, a field investigation is needed to determine all hydrologic properties of the catchment and use them in setting up the model; however in this study this was not fully possible. Work had to be carried out with the data available, and this was only a brief physical description of the area and the rainfall and runoff data. Based on the soil type data, land use and topography of the catchment; it is possible to make a judgement about the lag times, but such data was not available. That is why the statistical analysis of the interdependencies between rainfall and runoff was used to determine the lags.

It is also important to stress that one of the features of IBL is that it is possible for model users and decision-makers to judge why and how a certain prediction is made. The reason is that it is easy to find the nearest neighbours of the new data vector. To illustrate this, an example from the Bagmati data set is selected together with the nearest neighbours and a prediction is made by LWR and IBK (Table V)). It can be easily seen that the prediction for day 4 July 1989 in high flow season is based on the hydrological situation on days 14 September 1994, 14 July 1991 and 14 August 1993 when the precipitation and flow records were similar (e.g. high rainfall and high flows) to those observed on 4 July 1989 (see Figure 8). Similarly for the medium flow on 15 July 1989, prediction is based on the records on 6 August 1990, 23 June 1992 and 28 July 1990 and so on for low flow on 16 August 1989. IBL approach makes the prediction transparent and explainable.

CONCLUSIONS

The performed experiments with the selected case studies showed that the IBL methods are comparable with the other data driven methods: In the Bagmati case study the accuracy of IBL methods are higher than that of the other methods. In the Sieve case study M5 model trees (piece-wise linear models) were better in predicting Q_{t+1}. The performances of the DDMs has same order of magnitude as the performance of lumped conceptual hydrological model (tested only for Bagmati catchment).

Overall, the IBL methods appear to be accurate numerical predictors and can be successfully used in forecasting. Among the IBL methods, LWR is the most accurate one, but the k-NN method deserves a credit as well. In many cases it was found to be the second best, and only marginally worse in accuracy than LWR. Its attractive feature is that it permits to identify the instances (hydrological events) in the past on the basis of which a very simple (averaging) predictive model is built.

Dependency of the model accuracy on the choice of various kernels for LWR and distance functions was investigated. It was also noted that the use of the inverse weighted kernel functions lead to the higher computational time than in the case of using linear and Gaussian kernels (linear weighting kernel function is the fastest but inaccurate).

In the context of hydrological modelling, IBL and modular models like M5 model trees can be seen as a combination of "local" models, each responsible for forecasting in a particular region of the input space—corresponding to a particular hydrological condition. IBL, using the discharges resulting from the similar past hydrological situations to compute the forecast,

Table V. Example of nearest neighbours of the three query points for Bagmati data

Point	Date	Attributes						Prediction	
		RE_t	RE_{t-1}	RE_{t-2}	Q_t	Q_{t-1}	Q_{t+1}	LWR	IBK
Query point 1	04/07/1989	80·5	33·6	1·2	576	434	1660	1496·6	1655·7
	14/09/1994	88·1	30·0	7·1	525	390	1471		
Neighbours	14/07/1991	86·3	19·1	0	583	339	1370		
	14/08/1993	93·2	21·9	1·3	673	439	2260		
Query point 2	15/07/1989	33·5	39·1	0	974	832	548	848·3	566
	06/08/1990	24·0	31·4	14·1	835	736	501		
Neighbours	23/06/1992	29·2	36·6	5·3	717	586	705		
	28/07/1990	28·5	35·0	1·7	752	536	502		
Query point 3	16/05/1989	0	1·6	2·7	17·7	19·9	16·4	17·73	13·4
	07/02/1995	0	0·7	2·9	12·7	12·8	12·2		
Neighbours	18/05/1992	0	0·8	2·8	20·7	30	14·3		
	29/03/1994	0	0	2·5	14·2	21·7	13·8		

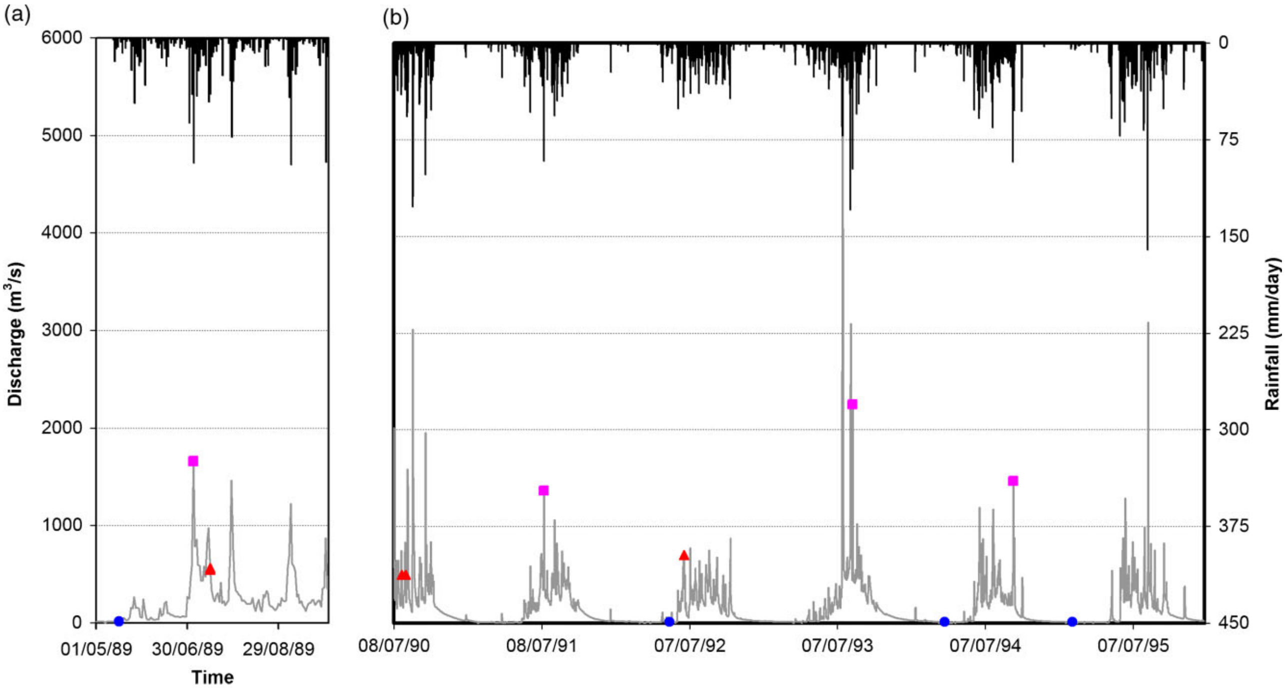

Figure 8. Example of three nearest neighbours in the Bagmati data. (a) Three query points in the test data; (b) three nearest neighbours of the query points in the training data. Square blocks in (b) are the three nearest neighbours of the query point represented by the square block in (a). Similarly triangles are the nearest neighbours of the query point shown by the triangle shape in (a) and so on. The thick grey line represents the observed discharge and the black line on the right and on the inverted axis show the rainfall

is dependent on the appropriate choice of the lagged hydrological variables characterizing such conditions. The hydrological characteristics of the catchment are embodied in the set of input variables and in this sense the DDMs cannot be considered "black-box" models.

IBL methods, together with the M5 model trees (Solomatine and Dulal, 2003; Solomatine and Siek, 2006) can be seen as important alternatives to statistical models, non-linear methods like ANNs, and may play important role in hydrological forecasting, complementing thus the physically-based distributed models. They also have an advantage of being more transparent than ANNs and hence may be easier accepted by decision-makers.

The further directions of research are seen in (1) extending the models types that are combined into a modular model, and (2) considering the interpretable hydrological events together with the simulation model runs as inputs to IBL.

REFERENCES

Abrahart RJ, See L. 2000. Comparing neural network and autoregressive moving average techniques for the provision of continous river flow forecasts in two contrasting catchments. *Hydrological Processes* **14**: 2157–2172.

Aha D, Kibler D, Albert M. 1991. Instance-based learning algorithms. *Machine Learning* **6**: 37–66.

Allen RG, Pereira SL, Raes D, Smith M. 1998. *Crop Evapotranspiration: Guidelines for Computing Crop Water Requirements*. FAO: Rome.

Atkeson CG, Moorey AW, Schaal S. 1996. Locally weighted learning. *Artificial Intelligence Review* **11**: 11–73.

Becker A, Kundzewicz ZW. 1987. Nonlinear flood routing with multilinear models. *Water Resources Research* **23**: 1043–1048.

Bray M, Han D. 2004. Identification of support vector machines for runoff modelling. *Journal of Hydroinformatics* **6**: 265–280.

Cigizoglu HK. 2003. Estimation, forecasting and extrapolation of river flows by artificial neural networks. *Hydrological Sciences Journal* **48**(3): 349–361.

Cleveland WS, Loader C. 1994. *Smoothing by Local Regression: Principles and Methods*, Technical Report 95·3. AT and T Bell Laboratories, Statistics Department: Murray Hill, NJ.

Cover TM, Hart PE. 1967. Nearest neighbour pattern classification. *IEEE Transactions on Information Theory* **13**: 21–27.

Dawson CW, Wilby RL. 2001. Hydrological modeling using artificial neural networks. *Progress in Physical Geography* **25**(1): 80–108.

Dibike Y, Solomatine DP. 1999. *River Flow Forecasting using Artificial Neural Networks*. Geophysical Research Abstracts, EGS XXIV General Assembly: The Hague.

Dibike Y, Solomatine DP. 2001. River flow forecasting using artificial neural networks. *Journal of Physics and Chemistry of the Earth, Part B: Hydrology, Oceans and Atmosphere* **26**(1): 1–8.

Fedorov VV, Hackl P, Muller WG. 1993. Moving local regression: the weight function. *Nonparametric Statistics* **2**(4): 355–368.

Franchini M. 1996. Use of genetic algorithm combined with a local search method for the automatic calibration of conceptual rainfall–runoff models. *Hydrological Science Journal* **41**(1): 21–39.

Galeati G. 1990. A comparison of parametric and non-parametric methods for runoff forecasting. *Hydrological Sciences Journal* **35**(1): 79–94.

Gasser T, Muller HG. 1979. Kernel estimation of regression functions. In Smoothing Techniques for Curve Estimation, *Lecture Notes in Mathematics*, Gasser T, Rosenblatt M, (eds) Springer-Verlag: Heidelberg; 23–67.

Govindaraju RS, Rao AR. 2000. *Artificial Neural Networks in Hydrology.* Kluwer Academic Publishers: Norwell, MA; 329.

Haykin S. 1999. *Neural Networks: A Comprehensive Foundation.* Prentice Hall: Englewood Cliffs, NJ.

Hsu K, Gupta HV, Sorooshian S. 1995. Artificial neural network modelling of the rainfall–runoff process. *Water Resources Research* **31**(10): 2517–2530.

Karlsson M, Yakowitz S. 1987. Nearest neighbour methods for non-parametric rainfall–runoff forecasting. *Water Resources Research* **23**(7): 1300–1308.

Kuncheva LI. 2004. *Combining Pattern Classifiers: Methods and Algorithms.* John Wiley and Sons: Chichester.

Maier HR, Dandy GC. 2000. Neural network for the prediction and forecasting of water resources variables: a review of modeling issues and application. *Environmental Modeling & Software* **15**: 101–124.

Marsigli M, Todini F, Diomede T, Liu Z, Vignoli R. 2002. Calibration of rainfall–runoff models. *MUSIC—Multiple-Sensor Precipitation Measurements, Integration, Calibration and Flood Forecasting*. Brussels: Commission European Communities.

Minns AW, Hall MJ. 1996. Artificial neural networks as rainfall–runoff models. *Hydrolgical Science Journal* **41**: 399–417.

Mitchell TM. 1997. *Machine Learning.* McGraw-Hill: Singapore; 414.

Quinlan JR. 1992. Learning with continuous classes. In *The 5th Australian Joint Conference on AI*, World Scientific: London; 343–348.

Quinlan JR. 1993. Combining instance-based and model-based learning. In *Machine Learning: Proceedings of the Tenth International Conference*. Morgan Kaufmann: Amherst, MA; 236–243.

Refsgaard JC. 1996. Terminology, modelling protocol and classification of hydrological model codes. In *Distributed Hydrological Modelling*, Abbott MB, Refsgaard JC (eds). Kluwer Academic Publishers: Dordrecht; 321.

Scott DW. 1992. *Multivariate Density Estimation.* Wiley: New York.

Shamseldin AY, O'Connor KM. 1996. A nearest neighbour linear perturbation model for river flow forecasting. *Journal of Hydrology* **179**: 353–375.

Shrestha I. 2003. *Conceptual and data-driven hydrological modelling of bagmati river basin, Nepal*. MSc thesis HH451, UNESCO-IHE Institute for Water Education, Delft.

Shrestha DL, Solomatine DP. 2006. Machine learning approaches for estimation of prediction interval for the model output. *Neural Networks* **19**(2): 225–235.

Solomatine DP. 2005. Data-driven modeling and computational intelligence methods in hydrology. In *Encyclopedia of Hydrological Sciences*, Anderson M (ed.). Wiley: New York.

Solomatine DP, Dulal KN. 2003. Model trees as an alternative to neural networks in rainfall–runoff modelling. *Hydrological Sciences Journal* **48**(3): 399–411.

Solomatine DP, Price RK. 2004. Innovative approaches to flood forecasting using data driven and hybrid modelling. In *6th International Conference on Hydroinformatics*. World Scientific Publishing Company: Singapore.

Solomatine DP, Siek MB. 2006. Modular learning models in forecasting natural phenomena. *Neural Networks* **19**(2): 215–224.

Solomatine DP, Xue Y. 2004. M5 model trees and neural networks: application flood forecasting in the upper reach the Huai River in China. *Journal of Hydrologic Engineering* **9**(6).

Sugawara M. 1995. Tank model. In *Computer Models of Watershed Hydrology*, Singh VP (ed.). Water Resources Publication: Highlands Ranch, CO; 165–214.

Todini E. 1996. The arno rainfall–runoff model. *Journal of Hydrology*, **175**: 339–382.

Toth E, Brath A, Montanari A. 2000. Comparison of short-term rainfall prediction models for real-time flood forecasting. *Journal of Hydrology* **239**: 132–147.

Weiss SM, Indurkhya N. 1995. Rule-based machine learning methods for functional prediction. *Journal of Artificial Intelligence Research* **3**: 383–403.

Witten IH, Frank E. 2000. *Data Mining: Practical Machine Learning Tools with Java Implementations.* Morgan Kaufmaan: San Francisco, CA; 132–147.

Wolpert D. 1992. Stacked generalisation. *Neural Networks* **5**: 241–259.

DOI: 10.1002/hyp