Tournament Searching Method for Optimization of the Forecasting Model Based on the Nadaraya-Watson Estimator

Grzegorz Dudek

Department of Electrical Engineering, Czestochowa University of Technology,
Al. Armii Krajowej 17, 42-200 Czestochowa, Poland
dudek@el.pcz.czest.pl

Abstract. In the article the tournament searching method is used for optimization of the forecasting model based on the Nadaraya-Watson estimator. This is a nonparametric regression model useful for forecasting the nonstationary in mean and variance time series with multiple seasonal cycles and trend. The tournament searching is a stochastic global optimization algorithm which is easy to use and competitive to other stochastic methods such as evolutionary algorithms. Three types of tournament searching algorithms are proposed: for estimation of the forecasting model parameters (continuous optimization), for the predictor selection (binary optimization) and for both predictor selection and parameter estimation (mixed binary-continuous optimization). The effectiveness of the proposed approach is illustrated through applications to electrical load forecasting and compared with other optimization methods: grid search method, genetic and evolutionary algorithms, and sequential methods of feature selection. Application examples confirm good properties of tournament searching.

Keywords: Tournament searching, binary and continuous optimization, Nadaraya-Watson estimator, multiple seasonal time series forecasting, short-term load forecasting.

1 Introduction

Time series forecasting plays a significant role in economy, industry, seismology, meteorology, geophysics etc. The purpose of forecasting is to support decision-making processes, to stimulate for action favoring or opposing the realization of the forecast or to provide information about the changes of some phenomenon in the future. In general, time series consists of four types of components: trend, seasonality, cycling and irregularity. They combine in an additive or multiplicative fashion. Sometimes there are multiple seasonal variations. This complicates the construction of the forecasting model. A typical procedure in such a case is to simplify the problem by deseasonality or decomposition of the time series. After decomposition the components showing less complexity than the original time series can be forecasted using simpler models.

The most commonly used conventional approaches to the modeling of time series with seasonality are the autoregressive moving average models (ARMA, ARIMA, SARMA etc.) and the Holt-Winters exponential smoothing models. The rapid

L. Rutkowski et al. (Eds.): ICAISC 2014, Part II, LNAI 8468, pp. 351–360, 2014.
© Springer International Publishing Switzerland 2014

development of computational intelligence and machine learning in recent years has brought many new methods of forecasting such as: artificial neural networks, fuzzy inference systems, regression trees and support vector machines. The conventional approaches as well as computational intelligence ones usually require many parameters (tens, hundreds or even thousands) for modeling nonstationary time series with trend and multiple seasonal cycles. The searching of the model space to find the optimal solution in this case is a very complex optimization task. It is due to different types of parameters (qualitative, continuous, discrete, logical) and multimodality of the error function. The choice of the appropriate methods of learning or optimization and values of their parameters is often a separate optimization problem. In the case of unstable models (e.g. neural networks), where we observe different learning results for the same training data, the optimization process is much more difficult.

In the article we describe a simple deterministic forecasting model based on Nadaraya–Watson estimator. This is a similarity–based model working on patterns of the seasonal cycles of time series [1]. Using patterns we filter the time series removing the trend and seasonal variations of periods longer than the basic period and we get stationary time series. We propose the tournament searching method for optimization of the model. This is a stochastic, global optimization method with only one or two parameters controlling the local/global optimization property. The tournament searching effectively optimizes the Nadaraya–Watson estimator in the continuous and binary spaces.

2 Forecasting Model Based on the Nadaraya-Watson Estimator

The Nadaraya-Watson estimator (N-WE) as a forecasting tool was derived from the conditional density estimator using kernel functions in [2]. Nonparametric conditional density estimation is a way to model the relationship between past and future realisation of the random variable. The regression function in this case is defined as:

$$m(x)=\frac{\sum_{j=1}^{N}K\left(\frac{x-x_j}{h}\right)y_j}{\sum_{j=1}^{N}K\left(\frac{x-x_j}{h}\right)}, \tag{1}$$

where: N is a number of elements in a random sample, x is a predictor, y is a response variable, $K(.)$ is a kernel function and h is its bandwidth.

For multidimensional predictors the kernels are expressed using a multidimensional product kernel function. The selection of the kernel function form is not as important as the selection of their bandwidths. When we use normal kernels the N-WE for multidimensional predictors is of the form:

$$m(\mathbf{x})=\frac{\sum_{j=1}^{N}\exp\left(-\sum_{t=1}^{n}\frac{(x_t-x_{j,t})^2}{2h_t^2}\right)y_j}{\sum_{j=1}^{N}\exp\left(-\sum_{t=1}^{n}\frac{(x_t-x_{j,t})^2}{2h_t^2}\right)}, \tag{2}$$

where $\mathbf{x}\in\mathbb{R}^n$.

The response variable can be a vector as well.

Estimator (2) is a linear combination of response variables y_j weighted by the normalized kernel functions. Kernels map nonlinearly the distance between points $\mathbf{x}$ and $\mathbf{x}_j$. The bandwidth h_t decides about the share of the t-th component of $\mathbf{x}$ in the distance (greater value of h implies larger share). The bias-variance tradeoff of the regression model (2) is controlled by bandwidth values. Too small values of h result in undersmoothing, whereas too large values result in oversmoothing. Proper selection of the h values is therefore a key issue. For the normal product density estimators Scott proposed a rule [3]:

$$h_t^S = \hat{\sigma}_t N^{-\frac{1}{n+4}}, \tag{3}$$

where $\hat{\sigma}_t$ is the estimated standard deviation of the t-th component of $\mathbf{x}$.

The N-WE is a flexible forecasting method due to the local nature of fitting of the simple regression models. In the next section we describe how the N-WE is optimized using tournament searching.

One more issue should be clarified. How are predictors and response variable defined? For the time series considered in Section 4 and presented in Fig. 2 we define patterns of the daily cycles:

$$x_{i,t} = \frac{z_{i,t} - \bar{z}_i}{\sqrt{\sum_{l=1}^{n} (z_{i,l} - \bar{z}_i)^2}}, \tag{4}$$

where: $z_{i,t}$ is a component of the vector $\mathbf{z}_i = [z_{i,1}\ z_{i,2}\ \ldots\ z_{i,n}]$ including the elements of time series from the i-th daily cycle (electrical loads at successive hours of the day i in our example), $\bar{z}_i$ is a mean value of elements in cycle i, $x_{i,t}$ is the component of the pattern vector $\mathbf{x}_i = [x_{i,1}\ x_{i,2}\ \ldots\ x_{i,n}]$ representing the daily cycle $\mathbf{z}_i$.

Patterns $\mathbf{x}$ defined using (4) are normalized versions of vectors $\mathbf{z}$. Thus they have the unity length, zero mean and the same variance. It is worth nothing that after normalization the nonstationary in mean and variance time series $\{z_k\}$ is represented by patterns having the same mean and variance. The trend and additional seasonal cycles longer that the daily one are filtered. This simplification of the time series facilitates the construction of the forecasting model.

In the similar way to the predictors the response variables are defined:

$$y_i = \frac{z_{i+\tau,t} - \bar{z}_i}{\sqrt{\sum_{l=1}^{n} (z_{i,l} - \bar{z}_i)^2}}, \tag{5}$$

where: $z_{i+\tau,t}$ is the t-th element in the $(i+\tau)$-th daily cycle, τ is a forecast horizon (in daily cycles).

The y_i value encodes the actual time series element $z_{i+\tau,t}$ from the forecast period $i+\tau$ using current time series parameters ($\bar{z}_i$ and dispersion of a daily cycle in the denominator of (5)) from the nearest past, which allows to take into consideration current variability of the process and ensures possibility of decoding: when we get the forecast of y_i we can determine the forecast of $z_{i+\tau,t}$ using (5).

3 Tournament Searching for N-WE Optimization

The N-WE is optimized using the tournament searching method (TS). Three types of optimization procedures are performed. The first type concerns estimation of the bandwidth values. This is a continuous optimization problem, where we are searching for the vector $\mathbf{h} = [h_1, h_2, ..., h_n]$. The second type is the feature selection. This is a combinatorial optimization problem, where we are searching for the set of predictors. And the third type is the combined optimization, where we are searching for the bandwidths and the set of predictors in the same time. This is the mixed binary-continuous problem. The optimization criterion in these procedures is the forecast error (MAPE).

3.1 Estimation of the Bandwidth Values

The TS method has been proposed in [4] for combinatorial optimization (feature selection) as an alternative to the more complex stochastic global optimization methods such as genetic algorithm and simulated annealing. Application of TS to estimation of bandwidths requires redefinition of the algorithm (see Fig. 1).

Starting from the solution created according to the Scott's rule TS explores the solution space generating new solutions by perturbing the parent solution. The set of l candidate solutions $\{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_l\}$ is created from the parent solution in each iteration using the move (or mutation) operator defined as:

$$h_{i,t} = h_t^* + \xi_{i,t}, \quad i = 1, 2, ..., l, \quad t = 1, 2, ..., n, \tag{6}$$

where $h_{i,t}$ is the mutated value of the parent solution component h_t^*, $\xi_{i,t} \sim N(0, \sigma_t)$.

The standard deviation of the normal distribution determines the range of moving. It is assumed that $\sigma_t = w\, h_t^S$, where $w = \text{const} \in \mathbb{R}^+$. Thus the moving range in the t-th direction is dependent on the initial value of h_t, i.e. on the variance of the t-th component of $\mathbf{x}$.

1. Initialization of the parent solution using the Scott's rule (3).
2. Generation of the set of *l* candidate solutions from the parent solution using the move operator.
3. Evaluation of the candidate solutions using N-WE.
4. Tournament – selection of the best solution among the candidate solutions.
5. Replacement of the parent solution by the tournament winner.
6. Repeat steps 2-6 until the stop criterion is reached.

Fig. 1. Algorithm of tournament searching for estimation of bandwidth values

After moving the candidate solutions are evaluated and the best one is selected. It replaces the parent solution and other set of candidate solution is generated from it in the next iteration. The l parameter (called the tournament size) and the standard deviation of mutation σ controls exploration/exploitation properties of the algorithm. When l is large the local minima attracts the searching process more intensively. The probability of escaping from the basin of attraction of the local minimum increases when l decreases. But in this case the searching process is more random. The standard deviation σ determines the length of jumps, i.e. the distance between the parent solution and the candidate solutions generated from it.

3.2 Selection of the Predictors

The components of pattern **x** (predictors) are strongly correlated. So elimination some of them should simplify the forecasting model without deteriorating its quality. The solution (selected predictors) is represented by a binary vector $\mathbf{b} = [b_1, b_2, \ldots, b_n]$. Ones in **b** indicates the selected components. TS algorithm for the problem of binary vector searching was proposed in [4]. The searching scheme is similar to that presented in Fig. 1, except the first step. Now the parent solution is initialized by random. The move operator generates $l \in \{1, 2, \ldots, n\}$ candidate solutions by switching the value of one randomly chosen bit (different for each candidate solution) of the parent solution. The tournament size l controls the exploration/exploitation properties, as in the case of TS for continuous optimization described in Section 3.1. When $l = 1$ the solution space is searched with a random walk algorithm, resistant to local minima. When $l = n$ we get a hill climbing procedure, which gets stuck in local minima. We recommend $l = \text{round}(n/3)$. In this case the algorithm quite intensively searches the neighborhoods of local minima but is able to leave their basins of attraction.

3.3 Mixed optimization: Selection of Predictors and Estimation of Bandwidth Values

Results of both the bandwidth estimation and selection of predictors are obviously interdependent. We propose TS method for simultaneous searching of two spaces: binary space of selected predictors and continuous space of bandwidths. The algorithm processes two paired vectors: **b** encoding selected predictors and **h** encoding bandwidths. The algorithm scheme is presented in Fig. 1, except the first step. The ways of initialization of both vectors **h** and **b** are the same as described in Sections 3.1 and 3.2, respectively. The move operator for vector **b** is the same as described in Section 3.2. The move operator for vector **h** has form (6), wherein only these components of **h** are modified which correspond to ones in the paired **b** vector. The paired vectors (**b**, **h**) after moving are evaluated together using N-WE. The tournament size $l \in \{1, 2, \ldots, n\}$ plays the same role as in the TS algorithms described above.

4 Application Example

We illustrate the optimization of N-WE using tournament searching on example of forecasting time series with multiple seasonal cycles. That is a short-term electrical load forecasting problem. We use the time series of the hourly electrical load of the Polish power system from the period 2002–2004, which is shown in Fig. 2. This time series is nonstationary and exhibits trend and tree seasonal variations: annual, weekly and daily.

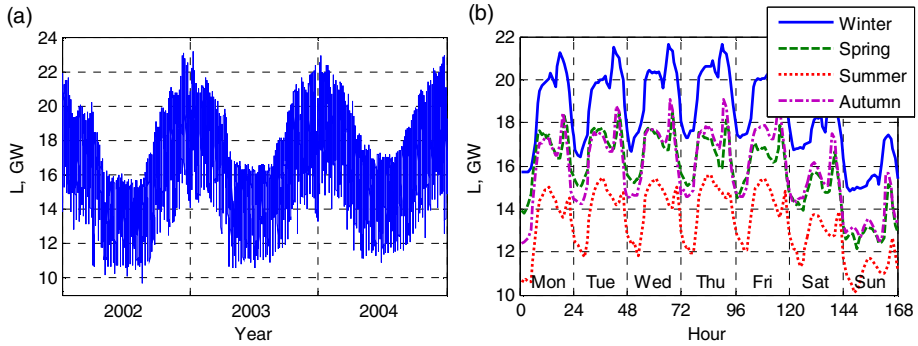

Fig. 2. The time series of electrical load of the Polish power system in three-year (a) and one-week (b) intervals

Our goal is to forecast the power system load for the next day ($\tau = 1$) at hours $t = 1, 6, 12, 18$ and 24. The test set includes 30 days from January 2004 (without untypical 1 January) and 31 days from July 2004. The training set for each forecasting task (load forecasting at hour t of the day j) is prepared individually. It contains patterns $\mathbf{x}$ representing the same days of the week (Monday, ..., Sunday) as the query pattern and paired with them y-values representing load at hour t for the next day. The training set is determined from the historical data. For each of the 305 forecasting tasks the separate N-WE model is created and optimized.

The proposed TS were compared with other optimization methods. For estimation of bandwidth values the grid search method (GS) and evolutionary algorithm (EA) are applied.

The GS searches the neighborhood of the point $\mathbf{h}^S = [\, h_1^S\,,\ h_2^S\,,\ \ldots,\ h_n^S\,]$ determined using Scott's rule (3). In the k-th iteration of the GS algorithm $\mathbf{h}_k$ point is generated as follows:

$$\mathbf{h}_k = a_k \mathbf{h}^S, \quad k = 1, 2, \ldots, \tag{7}$$

where $a_k = a_0 + \Delta(k-1)$, $a_0 \in \mathbb{R}^+ \le 1$ and Δ is the step defining the grid density.

The stop criterion (N iterations without improvement in results) determines the final value of k. GS is sub-optimal and searches the sets of discretized values of $\mathbf{h}$ components. The multidimensional optimization problem: estimation of $h_1, h_2, \ldots, h_n$ is replaced here with one-dimensional problem: estimation of a.

The EA searches n-dimensional space to estimate vector **h**. The vectors **h**, which are individuals in EA, are initialized by the Scott's rule. The operators used in evolutionary process are: mutation, recombination and selection. The mutation is the same as in TS for continuous optimization (6). The recombination creates two new individuals by linear combinations of two parent individuals selected by random (so-called arithmetic or intermediate recombination) [5]:

$$h'_{a,t} = h_{a,t} + c(h_{b,t} - h_{a,t}),\ h'_{b,t} = h_{b,t} + c(h_{a,t} - h_{b,t}), \tag{8}$$

where $c \sim U(0,1)$, $t = 1, 2, \ldots, n$.

The selection operator was the tournament selection [5]. The tournament size T controls the selection pressure. The elitist strategy was also applied: the best individual was copied form population i to i+1.

For selection of predictors two deterministic sub-optimal methods were applied [6]: sequential forward selection (SFS) and sequential backward selection (SBS), as well as genetic algorithm (GA). In GA solutions are represented by binary vectors **b**. As in EA three operators are used: mutation, recombination and selection. Mutation switches bits selected from the population of individuals by random with probability of p_m. One-point crossover is used as the recombination operator [5]. Tournament selection is used to select individuals to the next generation.

The parameters of the studied optimization algorithms were as follows:

- TS for estimation of bandwidth values (TSh): $l = 30$, $w = 0.1$, number of iterations $M = 100$,
- TS for selection of predictors (TSb): $l = 8$, number of iterations $M = 100$,
- TS for selection of predictors and estimation of bandwidth values (TSbh): $l = 8$, $w = 0.1$, number of iterations $M = 500$,
- GS: $a_0 = 0.1$, $\Delta = 0.05$, $N = 20$,
- EA: population size – 30, number of iterations $M = 100$, $T = 2$, probability of crossover – 0.9, probability of individual mutation – 1, $w = 0.1$,
- GA: population size – 8, number of iterations $M = 100$, $T = 2$, probability of crossover – 0.9, probability of mutation – 0.05.

These parameters were adjusted in the preliminary tests. The stop criterion in EA and TSh was: there is no improvement in results in $0.25M$ successive iterations.

The N-WE was optimized in leave-one-out procedure. The forecast errors (mean absolute percentage error MAPE) on validation and test samples for different methods of parameter estimation in Table 1 are shown. From this table it can be seen that the validation error was reduced when using GS, EA and TSh but the test error was not reduced. This could be due to insufficient information about the target function contained in the learning points which are sparse distributed in the n-dimensional space.

The bandwidth values estimated using GS were in 91% of cases higher than the values determined according to the Scott's rule. This percentage for EA and TSh was 65 and 67%, respectively. The optimal bandwidths for one of the forecasting task in Fig. 3 are shown.

Table 1. The forecast errors for different methods of estimation of the bandwidth values

Method	January		July		Mean	
	$MAPE_{val}$	$MAPE_{tst}$	$MAPE_{val}$	$MAPE_{tst}$	$MAPE_{val}$	$MAPE_{tst}$
Scott's rule	1.62	1.20	1.54	0.92	1.58	1.05
GS	1.58	1.21	1.51	0.96	1.55	1.09
EA	1.32	1.36	1.28	0.90	1.30	1.13
TSh	1.30	1.23	1.25	0.93	1.28	1.08

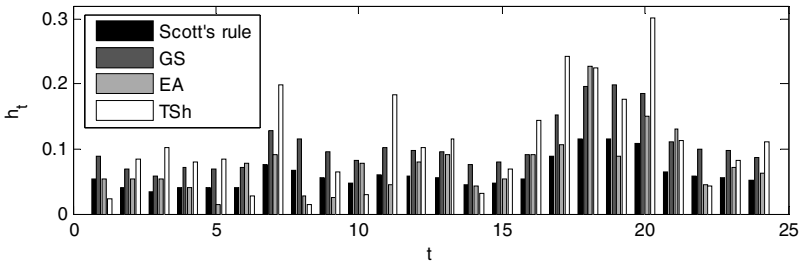

Fig. 3. The bandwidth values for the forecasting task of July 1, 2004, hour 12

The forecast errors for different methods of the predictor selection in Table 2 are shown. The bandwidths were determined using Scott's rule in this study except TSbh method which searches both set of predictors and bandwidth values at the same time. It can be seen from this table that both the validation and test errors were reduced when using stochastic optimization methods comparing to the case without selection but the test errors are statistically indistinguishable (Wilcoxon signed-rank test was used).

The selected predictors and bandwidth values for one of the forecasting task in Table 3 are shown. The frequencies of the predictor selection in Fig. 4 are shown.

Table 2. The forecast errors for different methods of selection of predictors

Method	January		July		Mean	
	$MAPE_{val}$	$MAPE_{tst}$	$MAPE_{val}$	$MAPE_{tst}$	$MAPE_{val}$	$MAPE_{tst}$
SFS	1.37	1.25	1.32	0.90	1.34	1.07
SBS	1.37	1.20	1.35	0.90	1.36	1.05
GA	1.38	1.17	1.34	0.90	1.36	1.03
TSb	1.34	1.17	1.30	0.90	1.32	1.03
TSbh	1.25	1.20	1.21	0.86	1.23	1.03

The average reduction in the number of predictors was as follows: for SFS – 76%, for SBS – 52%, for GA – 60%, for TSb – 67% and for TSbh – 57%. Thus the rejection of more than half of the predictors should not negatively affect the accuracy of the forecasting model. The most often selected predictors were x_{23} and x_{24}.

Table 3. The bandwidth values ($\cdot 10^{-3}$) for the forecasting task of July 1, 2004, hour 12

t	1	2	3	4	5	6	7	8	9	10	11	12
SFS	37	–	–	–	–	–	–	–	–	–	41	40
SBS	45	35	–	–	–	36	–	58	48	41	51	–
GA	44	34	–	–	–	35	64	57	47	40	–	–
TSb	44	34	–	–	–	35	–	57	47	40	–	–
TSbh	22	60	90	–	–	31	194	14	–	28	89	–
t	13	14	15	16	17	18	19	2–	21	22	23	24
SFS	–	–	–	37	–	–	–	–	–	–	39	–
SBS	–	–	41	46	–	–	–	–	–	50	49	–
GA	–	–	–	–	76	–	–	92	–	–	48	–
TSb	–	37	–	45	–	–	–	–	–	49	48	–
TSbh	–	54	59	40	–	–	165	151	–	42	–	–

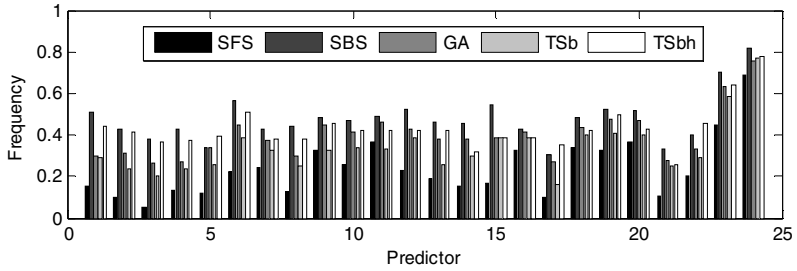

Fig. 4. The frequencies of predictor selection

5 Conclusions

The tournament searching method is a simple generic stochastic search method. It has an efficient, flexible algorithm which can be adapted to binary, continuous and mixed binary–continuous optimization. TS in a binary version has only one parameter controlling the global/local searching properties of the algorithm – the tournament size. In the continuous version TS has the second parameter – the standard deviation used for generating new solutions. This parameter determines the length of jumping from the parent solution to the candidate one, i.e. the size of the locally explored region.

Application of TS to optimization of the forecasting model based on Nadaraya-Watson estimator gave good results comparing to other optimization methods such as grid search, genetic and evolutionary algorithms, and sequential methods of feature selection. In a result we get simple, easy to use and accurate model for forecasting "hard" nonstationary time series with trend, multiple seasonal cycles and random noise.

Acknowledgment. The study was supported by the Research Project N N516 415338 financed by the Polish Ministry of Science and Higher Education.

References

1. Dudek, G.: Pattern-Similarity Machine Learning Models for Short-Term Load Forecasting. Academic Publishing House Exit, Warsaw (2012) (in Polish)
2. Dudek, G.: Short-Term Load Forecasting Based on Kernel Conditional Density Estimation. Przegląd Elektrotechniczny (Electrical Review) 86(8), 164–167 (2010)
3. Scott, D.W.: Multivariate Density Estimation: Theory, Practice, and Visualization. Wiley (1992)
4. Dudek, G.: Tournament Searching Method to Feature Selection Problem. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part II. LNCS, vol. 6114, pp. 437–444. Springer, Heidelberg (2010)
5. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer (1996)
6. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Elsevier (2009)