Michał Bernardelli

Collegium of Economic Analyses
Warsaw School of Economics

Parallel deterministic procedure based on hidden Markov models for the analysis of economic cycles in Poland

Summary

In the paper the deterministic version of the procedure based on hidden Markov models for the analysis of economic cycles is described. The quality of fitting hidden Markov models as well as the accuracy of the identification of turning points in the business cycle in Poland depends, among other things, on the number of states of the model and the size of panel data. Determinism however affects significantly on a time of computations. Speed up of computations could be achieved by adding the parallelism into the procedure. The usefulness of this approach is verified by the numerical experiments and comparative tests measuring a time of computations depending on the number of processor cores.

Keywords: hidden Markov model, parallel computing, computational complexity, Baum–Welch algorithm, business cycle turning points

1. Introduction

Various tools were developed for modeling the current and future economic situation. One of the primary sources of knowledge are business cycles. There are many computationally complex econometric models using both modern and established in theory and practice approaches from the border of mathematics and computer science. Due to the relatively big amount of data, potential presence of unspecified variables in created models or simply restrictive assumptions about the model and input data prognostic properties are very limited. Based on non-deterministic character and weak assumptions in many fields Markov models could be a better alternative.

The procedure based on hidden Markov models for the analysis of economic cycles was developed by Bernardelli and Dędys[1]. The quality of fitting hidden Markov models as well as the accuracy of the identification of turning points in the business cycle in Poland depends, among other things, on the number of states of the model and the size of panel data, see Bernardelli[2]. The proposed procedure uses Monte Carlo simulations and an iterative Expectation-Maximization method, therefore it has non-deterministic background. In order to achieve a reasonable approximation of the optimal solution a large number of simulations is required. Main purpose of the article is to present the deterministic version of that procedure. Determinism affects significantly on a time of computations. To solve this problem the parallel version of the procedure was implemented. Furthermore, the usefulness of this approach is verified by the numerical experiments and comparative tests measuring a time of computations depending on the number of processor cores.

The paper is composed of five sections. The short description of hidden Markov models with the particular emphasis on panel input data is in section 2. Section 3 presents the description of data and the steps of the deterministic procedure. In section 4 there are results from numerical experiments exploring parallelism of the procedure. The paper ends with the summary in section 5.

2. Hidden Markov models for panel data

Hidden Markov models (HMM) are widely used in analysis of processes and patterns in many fields, such as speech, handwriting or gesture recognition[3], bioinformatics[4], cryptanalysis or econometrics[5]. One of many definitions of hidden Markov chains uses the terminology from the field of finite-state probabilistic

---

[1] M. Bernardelli, M. Dędys, *Ukryte modele Markowa w analizie wyników testu koniunktury gospodarczej*, in: *Badanie koniunktury – zwierciadło gospodarki*, p. 1, "Prace i Materiały" IRG SGH, no. 90, Warszawa 2012, pp. 159–181.

[2] M. Bernardelli, *Non-classical Markov models in the analysis of business cycles in Poland*, „Roczniki" Kolegium Analiz Ekonomicznych SGH, z. 30, Oficyna Wydawnicza SGH, Warszawa 2013, pp. 59–74.

[3] F. Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, Cambridge 1997.

[4] R. Durbin, S. Eddy, A. Krogh, G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, Cambridge 1998.

[5] O. Cappé, E. Moulines, T. Rydén, *Inference in Hidden Markov Models*, Springer, New York 2005.

automaton[6]. To define a hidden Markov chain a set of the following parameters is necessary:

- a finite set of k states S_X with specified an initial state S_1,
- a transition matrix $P=[p_{i,j}]_{i,j=1}^{k}$, where $P_{i,j}$ is the probability of transition from the state i to the state j. It is assumed, that transition matrix is stochastic, that is for every i

$$\sum_{j=1}^{k} p_{i,j} = 1 .$$

- an alphabet Σ of symbols which are emitted in the specific state with the given probability distribution. We assume that in every state some symbol is emitted. For the finite alphabet the HMM in the state $i \in S_x$ the symbol $x \in \Sigma$ is emitted with the probability $e_i(x)$, and next it changes the state to j with the probability $p_{i,j}$. In the case of continuous probabilities by $e_i(x)$ a probability distribution is meant.

Observable are only symbols emitted by the model, but the current state of the hidden Markov chain remains unobservable. An alphabet Σ could be infinite and in this case the probability e_i is often given by the normal probability distribution defined by the expected value μ and the standard deviation σ. To generalize the normal distribution to more than one dimension is is enough to assume that e_i is given by a multivariate normal distribution. For independent normally distributed n random variables

$$e_i \sim N(\boldsymbol{\mu},\boldsymbol{\sigma}^2\mathbf{I}),$$

where $\boldsymbol{\mu} = [\mu_1,\mu_2,\ldots,\mu_n]^T$ is a vector of expected values and $\boldsymbol{\sigma} = [\sigma_1,\sigma_2,\ldots,\sigma_n]^T$ vector of standard deviations of each normally distributed random variables, whereas $\mathbf{I}$ is an identity matrix of degree n. Cases $n > 1$ could be treated as a panel input data of hidden Markov model. A schema of a three-state hidden Markov model with normal probability distributions of emitting symbols is presented in Figure 1.

[6] M.O. Rabin, *Probabilistic Automata*, “Information and Control” 1963, vol. 6(3), pp. 230–245.

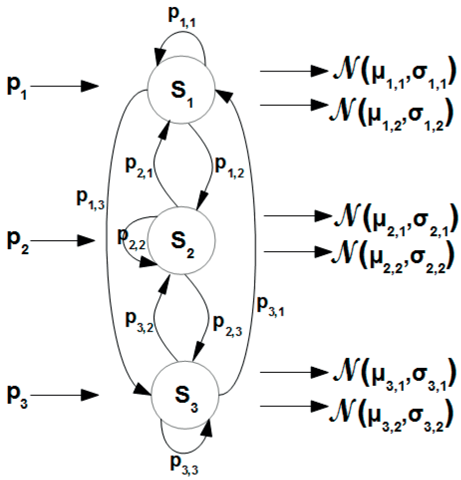

Figure 1. A schema of a three-state hidden Markov model with pair of normal probability distributions of emitting symbols

Source: own calculations.

3. Description of the data and the procedure

The source of the data is the business tendency survey conducted monthly by the Research Institute for Economic Development in Warsaw School of Economics. There are questions about current and future situation in Poland (due to a respondent's knowledge and prediction). The survey consists of eight questions:

- Question 1 – level of production,
- Question 2 – level of orders,
- Question 3 – level of export orders,
- Question 4 – stocks of finished goods,
- Question 5 – prices of goods produced,
- Question 6 – level of employment,
- Question 7 – financial standing,
- Question 8 – general economy situation.

For the calculations data from the March 1997 to February 2014 were taken. Input data were restricted to answers to the questions about current situation in the country, because models based on respondent's predictions turns out to be less accurate[7].

[7] M. Bernardelli, M. Dędys, op.cit.

The procedure takes on input the data from the survey and returns the path of states that has the highest probability in the whole considered period. Because of the numerical stability[8] and ease of interpretation computation were restricted to models with two, three and four states. In the case of a two states the interpretation is straightforward: the zero state is associated with periods determined by the respondents as worse in limiting to the considered question, while the state denote by one is related to the situation assessed as better. In three-state model there is an additional state ½ symbolizing the transient situation between states 0 and 1. It is a state designed for situations uncertain and difficult to unambiguous classification. Analogously the space of states of four-state hidden Markov chain has the form $\{0, \frac{1}{3}, \frac{2}{3}, 1\}$. State 0 indicates strong economic downturn, state 1 indisputable economic recovery, while the state ⅓ should be interpreted as indicating the uncertain status of a worse economic situation in the country and the state ⅔ suggests rather better economic conditions.

It is also assumed that in economy changes take place gradually. The reflection of this assumption is the structure of the transition matrix. Non-zero probabilities are permitted only between the adjacent states. For models with three and four states transitions matrices have forms respectively

$$P = \begin{pmatrix} * & * & 0 \\ * & * & * \\ 0 & * & * \end{pmatrix} \quad \text{and} \quad P = \begin{pmatrix} * & * & 0 & 0 \\ * & * & * & 0 \\ 0 & * & * & * \\ 0 & 0 & * & * \end{pmatrix}. \tag{1}$$

The procedure of business cycle turning points identification based on hidden Markov models can be described in the following steps:

1) decompose the input time series (using STL procedure[9]),
2) choose initial points, that are the starting values of the parameters of the hidden Markov chain,
3) for each point from the step 2 use the Baum–Welch algorithm[10] to estimate parameters of the hidden Markov model,

[8] M. Bernardelli, *Nieklasyczne modele Markowa – problemy numeryczne*, praca badawczo-rozwojowa, SGH, 2012.

[9] R.B. Cleveland, W.S. Cleveland, J.E. McRae, I. Terpenning, *STL: A Seasonal-Trend Decomposition Procedure Based on Loess*,„Journal of Official Statistics" 1990, vol. 6, pp. 3–73.

[10] L.E. Baum, T. Petrie, G. Soules, N. Weiss, *A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains*, „Ann. Math. Statist." 1970, vol. 41, no. 1, pp. 164–171.

4) assign to groups parameters of all calculated models on the basis of rounded to one decimal place expected values of conditional distributions; for each group define a representative model with parameters being averages of the respective parameters of models from this particular group,
5) for representative models from each group calculate the most probable path of a hidden Markov chain using the Viterbi algorithm[11],
6) based on selected optimization criteria or/and comparison with the reference time series choose the best HMM model.

A more detailed description of the procedure is presented in the article of Bernardelli[12]. An example of the output data from the procedure is shown in Figure 2.

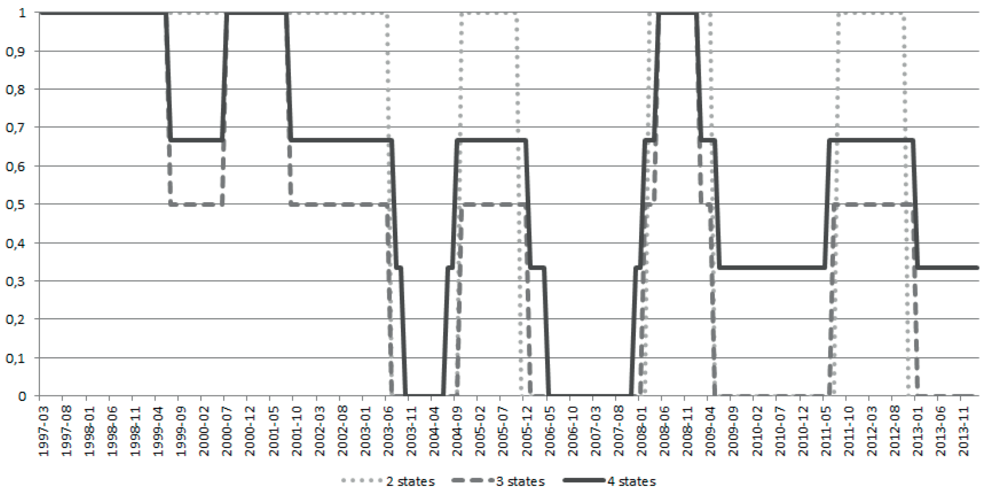

Figure 2. Viterbi paths of two-state, three-state and four-state hidden Markov model for the question about stocks of finished goods (from March 1997 to February 2014)

Source: own calculations.

Random or deterministic character of the procedure depends on the selection method of the initial points to the Baum–Welch algorithm (second step).

[11] A. Viterbi, *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*, „IEEE Transactions on Information Theory" 1967, vol. 13, issue 2.

[12] M. Bernardelli, *Kryteria optymalizacyjne w procedurze wykorzystującej ukryte modele Markowa do analiz danych ekonomicznych*, in: *Rola informatyki w naukach ekonomicznych i społecznych. Innowacje i implikacje interdyscyplinarne*, ed. Z. Zieliński, vol. 2, Wydawnictwo Wyższej Szkoły Handlowej, Kielce 2013, pp. 43–53.

The Baum–Welch algorithm is an iterative method that maximizes the expected value with respect to the likelihood function. Due to the way of finding the maximum, obtained solutions may be far from optimal. There is no guarantee that found by the algorithm local maximum is actually a global maximum. Therefore choosing the right initial parameters is crucial. To increase the chance of finding better approximation of the real solution, the algorithm is used repeatedly for the same input data, but different initial parameters. If the initial points are generated in the random (on the computer rather pseudo-random) way, then each use of the Baum–Welch algorithm is in fact the Monte Carlo simulation. Such version of a procedure was used and described by Bernardelli, Dędys[13] and Bernardelli[14]. Main purpose of the article is to present the deterministic version of the procedure.

Let describe in more detail the initial parameters for the Baum–Welch algorithm. We denote by k number of state and by n number of independent normally distributed input time series that is the size of panel data.

- Initial probabilities for each of k states. Each of them is a value from the interval [0,1], with the restriction, that sum of them must be equal to 1. By default in the procedure all probabilities are equal to $1/k$.
- The transition matrix P, see (1) with zero probabilities of transition between non-adjacent states. Only $4 + 3(k-2)$ non-zero elements of the matrix need to be specify. By default in the procedure probabilities are set as follows

$$p_{i,j} = \begin{cases} \frac{1}{2} & i=1,\ j=1,2;\ \ i=k,\ j=k-1,k \\ \frac{1}{3} & i=2,3,\ldots,k-1,\ j=i-1,i,i+1 \end{cases}. \qquad (2)$$

- Parameters ($\boldsymbol{\mu}$, $\boldsymbol{\sigma}$) of a normal distribution defining probability of emission of symbol in each state. There are $2kn$ parameters that determine final values of model parameters. In the procedure these initial values are chosen from the following intervals: an expected value $\mu_{i,j} \in \left[\overline{\mu_{l,j}} - 3\overline{\sigma_{l,j}}, \overline{\mu_{l,j}} + 3\overline{\sigma_{l,j}}\right]$ and a standard deviation $\sigma_{i,j} \in \left[0.5\overline{\sigma_{l,j}}, 3\overline{\sigma_{l,j}}\right]$, where $\overline{\mu_{l,j}}$ and $\overline{\sigma_{l,j}}$ are empirical parameters calculated for every state ($i=1,2,\ldots,k$) and each of input time series ($j=1,2,\ldots,n$). Of course intervals could be wider, but according to three sigma rule for the normal distribution in high probability they cover the vast majority of possible values.

[13] M. Bernardelli, M. Dędys, op.cit.
[14] M. Bernardelli, *Non-classical Markov*..., op.cit.

In the non-deterministic version of the procedure all values of $\mu_{i,j}$ and $\sigma_{i,j}$ are randomized with a uniform distribution from the specified intervals. To change the procedure into deterministic one those intervals have to be discretized. Let $m_{i,j}^{\mu}$ and $m_{i,j}^{\sigma}$ be numbers of nodes in the interval for respectively an expected value and a standard deviation of the j-th input time series of the i-th state. Number of all nodes M in the discretization grid is defined by the formula

$$M = \prod_{i=1}^{k}\prod_{j=1}^{n} m_{i,j}^{\mu} m_{i,j}^{\sigma}. \tag{3}$$

Mesh nodes may be distributed uniformly, but it is not always the best possible choice. Certainly, the greater value M, the finer grid and the better chance of finding a closer approximation of the exact solution. Assuming that for all $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, n$ values $m = m_{i,j}^{\mu} = m_{i,j}^{\sigma}$ number of nodes

$$M = m^{2kn}. \tag{4}$$

For different numbers of states k and values of m, numbers of nodes M for single input time series are given in Table 1, and for input data consists of a pair of time series in Table 2.

Table 1. Number of nodes for different sizes of discretization grid and different numbers of states in hidden Markov chain for a single question

M	k = 2	k = 3	k = 4
2	16	64	256
5	625	15 625	390 625
10	10 000	1 000 000	100 000 000
20	160 000	64 000 000	25 600 000 000
50	6 250 000	15 625 000 000	39 062 500 000 000
100	100 000 000	1 000 000 000 000	10 000 000 000 000 000

Source: own calculations.

Table 2. Number of nodes for different sizes of discretization grid and different numbers of states in hidden Markov chain for a pair of questions

m	k = 2	k = 3	k = 4
2	256	4 096	65 536
5	390 625	244 140 625	152 587 890 625
10	100 000 000	1 000 000 000	10 000 000 000 000 000
20	25 600 000 000	4 096 000 000 000 000	655 360 000 000 000 000 000
50	3,91E+13	2,44E+20	1,53E+27
100	1,00E+16	1,00E+24	1,00E+32

Source: own calculations.

Due to the high dimension of the grid, number of nodes is increasing exponentially with increasing number of states as well as with increasing size of panel data. The computation time is proportional to the number of nodes. Assuming that we have access to a supercomputer that can calculate one million of models in a second, a computation time for $k = 4$, $n = 2$ and $m = 100$ would take over 3×10^{18} years, while the age of the universe is estimated for not more than 14×10^{9} years. Therefore to get the result in a reasonable time the grid used in the procedure must by rather thick. Even then a time of calculations is essential. It should be emphasized that many methods are extremely susceptible to parallelization including those used in economy and finance like Monte Carlo simulations[15], Markov models[16], Bayesian averaging[17], option pricing[18] and others[19].

In modern computers parallel computing[20] is available in the form of hardware (multiple cores or/and processors) and software (threads, processes, automatic or semi-automatic parallelization). Nowadays those possibilities are not reserved for supercomputers or computer clusters, but accessible in every

[15] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Springer-Verlag, New York 2003.

[16] *Hidden Markov Models in Finance*, eds R.S. Mamon, R.J. Elliott, Springer International Series in Operations Research & Management Science 2007, vol. 104.

[17] M. Próchniak, B. Witkowski, *Time Stability of the Beta Convergence among EU Countries: Bayesian Model Averaging Perspective*, "Economic Modeling" 2013, vol. 30, pp. 322–333.

[18] Y. Peng, B. Gong, H. Liu, Y. Zhang, *Parallel Computing for Option Pricing Based on the Backward Stochastic Differential Equation*, High Performance Computing and Applications, Lecture Notes in Computer Science, Springer, Berlin–Heidelberg 2010, vol. 5938, pp. 325–330.

[19] Ł. Goczek, *Przegląd i ocena ekonometrycznych metod używanych w modelach empirycznych wzrostu gospodarczego*, "Gospodarka Narodowa" 2012, t. 10, pp. 49–73.

[20] A. Grama, G. Karypis, V. Kumar, A. Gupta, *Introduction to Parallel Computing (2nd Edition)*, Addison-Wesley, Reading, MA 2003.

personal computer. The procedure was implemented in R using the library *parallel*. To check susceptibility of the procedure for the parallelization numerical experiments were performed. The specification of the experiments and their results are described in the next section.

4. Numerical experiments

In the numerical experiments all questions as well as all combinations of pairs of questions from the survey were examine. There were considered hidden Markov chains with two, three and four states and different numbers of nodes of the grid. All of those computations were repeated on the same five-core computer with the use of cores from one to five. The total time of computations was measured to calculate speedup, that is

$$S(c) = \frac{t_1}{t_c}, \tag{5}$$

where c is the number of processors or cores in the processor and t_c is the execution time of the procedure with c processors (or cores of processor).

Results of computations on one core versus five cores are presented in Table 3. Speedups depending on numbers of cores used in the computations are depicted in Figure 3.

Table 3. Comparison of speedup on five cores for different sizes of discretization grid *M*, numbers of states *k* in hidden Markov chain and size of input panel data *n*

M	k = 2 n = 1	k = 3 n = 1	k = 4 n = 1	k = 2 n = 2	k = 3 n = 2	k = 4 n = 2
10	1.618	1.840	2.061	1.951	2.052	2.173
20	1.782	1.923	2.123	2.012	2.380	2.263
50	2.033	2.231	2.403	2.487	2.491	2.320
100	2.864	2.652	2.417	2.751	2.688	2.625
500	3.194	2.797	2.476	2.963	2.853	2.834
1000	3.243	2.849	2.492	3.428	3.264	3.153

Source: own calculations.

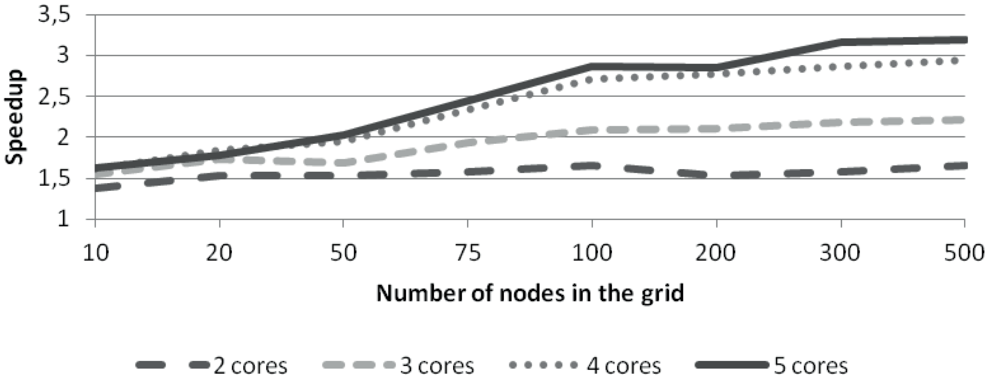

Figure 3. Speedups for two-state model and single time series input data

Source: own calculations.

Speedup with increasing number of cores is significant, but not satisfactory especially for single time series input and small number of states. For smaller grids computations on multiple cores could be even slower then those perform on a one core. The cause lies in the need for handling the parallelization. What is more, multiple cores used on one machine, still have shared memory. With many short tasks advantage of multicore computational power is outweighed by limited access to the memory. That is why the speedup is greater for HMM with greater number of states and size of input panel data. It is worth to emphasize that for computers with distributed architecture, speedup is expected to be ideal, that is equal c for c processors.

5. Conclusions

The implementation of the deterministic version of the procedure based on hidden Markov models may be used for the analysis of economic cycles. This implementation uses parallel capabilities of modern computers. In numerical experiments speedup related with the number of cores used in the computations was measured. It is justified to draw the following conclusions:

1. Both versions of the procedure: deterministic and non-deterministic, have an excellent susceptibility for the parallelization.
2. For small number of nodes of the grid and simple models (small number of states, small size of input data) computations with smaller number of cores are faster, due to the time needed for handling the parallelization of the

computations and shared memory. For distributed computers and larger number of nodes speedup is almost proportional to the number of processors.
3. Panel data on input of the procedure could improve the fit of the model but amount of time necessary for the computations increases exponentially with the number of states and the size of the panel data. Therefore full computation of the deterministic version of the procedure is impossible even with the use of parallelization capabilities.
4. The deterministic version of the procedure is better for smaller grids. The non-deterministic version is a better solution for the larger mesh size.

For improving the accuracy of the HMM fit, larger number of nodes is needed. The nodes could be and for large size of the grid have to be chosen in the random way. Especially in this case parallelization is very helpful. Numerical experiments provide evidence for the usefulness of parallelization and should be consider as an incentive to use in other methods in economy and finance.

References

1. Baum L.E., Petrie T., Soules G., Weiss N., *A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains*, „Ann. Math. Statist." 1970, vol. 41, no. 1, pp. 164–171.
2. Bernardelli M., *Kryteria optymalizacyjne w procedurze wykorzystującej ukryte modele Markowa do analiz danych ekonomicznych*, in: *Rola informatyki w naukach ekonomicznych i społecznych. Innowacje i implikacje interdyscyplinarne*, ed. Z. Zieliński, vol. 2, Wydawnictwo Wyższej Szkoły Handlowej, Kielce 2013, pp. 43–53.
3. Bernardelli M., *Nieklasyczne modele Markowa – problemy numeryczne*, praca badawczo-rozwojowa, SGH, 2012.
4. Bernardelli M., *Non-classical Markov models in the analysis of business cycles in Poland*, "Roczniki" Kolegium Analiz Ekonomicznych, z. 30, Oficyna Wydawnicza SGH, Warszawa 2013, pp. 59–74.
5. Bernardelli M., Dędys M., *Ukryte modele Markowa w analizie wyników testu koniunktury gospodarczej*, in: *Badanie koniunktury – zwierciadło gospodarki*, p. 1, "Prace i Materiały" IRG SGH, no. 90, Warszawa 2012, pp. 159–181.
6. Cappé O., Moulines E., Rydén T., *Inference in Hidden Markov Models*, Springer, New York 2005.

7. Cleveland R.B., Cleveland W.S., McRae J.E., Terpenning I., *STL: A Seasonal--Trend Decomposition Procedure Based on Loess*, "Journal of Official Statistics" 1990, vol. 6, pp. 3–73.
8. Durbin R., Eddy S., Krogh A., Mitchison G., *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, Cambridge 1998.
9. Glasserman P., *Monte Carlo Methods in Financial Engineering*, Springer-Verlag, New York 2003.
10. Goczek Ł., *Przegląd i ocena ekonometrycznych metod używanych w modelach empirycznych wzrostu gospodarczego*, "Gospodarka Narodowa" 2012, t. 10, pp. 49–73.
11. Grama A., Karypis G., Kumar V., Gupta A., *Introduction to Parallel Computing*, Addison-Wesley, Reading, MA 2003.
12. *Hidden Markov Models in Finance*, eds R.S. Mamon, R.J. Elliott, Springer International Series in Operations Research & Management Science, vol. 104, 2007.
13. Jelinek F., *Statistical Methods for Speech Recognition*, MIT Press, Cambridge 1997.
14. Peng Y., Gong B., Liu H., Zhang Y., *Parallel Computing for Option Pricing Based on the Backward Stochastic Differential Equation, High Performance Computing and Applications*, Lecture Notes in Computer Science, vol. 5938, Springer, Berlin–Heidelberg 2010, pp. 325–330.
15. Próchniak M., Witkowski B., *Time Stability of the Beta Convergence among EU Countries: Bayesian Model Averaging Perspective*, "Economic Modeling" 2013, vol. 30, pp. 322–333.
16. Rabin M.O., *Probabilistic Automata*, "Information and Control" 1963, vol. 6(3), pp. 230–245.
17. Viterbi A., *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*, "IEEE Transactions on Information Theory" 1967, vol. 13, issue 2.