Bayesian analysis of time series using granular computing approach

Olgierd Hryniewicz, Katarzyna Kaczmarek

Systems Research Institute, Polish Academy of Sciences, Nowelska 6, 01-447 Warsaw, Poland

1. Introduction

The Bayesian methods for the time series analysis are proven successful in many practical applications, e.g., [1–3]. Nonetheless, the Bayesian time series analysis and probability theory are not meant to process directly the information described in natural language, and according to Zadeh [4], people granulate information and operate mainly on words and propositions easy to express in natural language. Hopefully, the soft computing methods, especially data mining, enable to retrieve and process the conceptually meaningful human-consistent information from large datasets. Such information entities have given rise to the general framework of Granular Computing [5,6].

Although, time series data mining with the use of the soft computing tools has gained a lot of attention in the literature, the interdisciplinary combination of soft computing and Bayesian approach does not seem to be extensively investigated. As presented in Table 1, according to the keyword and abstract search, there are over 2.7k articles on ‘Bayesian’ AND ‘time series’, and over 2.3k results about ‘granular computing’. However, only 1 result combing these keywords. Furthermore, the Scopus search engine returns only 6 positions for the keywords ‘soft computing’ AND ‘time series’ AND ‘Bayesian’.

The objective of this paper is to review the related work on the application of the soft computing methods for the time series analysis, and to propose the conceptual Bayesian Granular Computing (B-GC) framework for Time Series Forecasting. The goal is to incorporate the human-consistent soft computing methods, especially data mining and classification techniques, in the construction of the prior model probability distributions.

The proposed approach assumes employing techniques from the following research fields: the fuzzy sets theory, the knowledge discovery from sequential data, especially the time series abstraction and mining for linguistic summaries, Support Vector Machines for the classification problem and the Markov Chain Monte Carlo methods for the posterior simulation and the Bayesian inference. The approach is in line with the Generalized Theory of Uncertainty [7].

We provide an application example as a demonstration of concept and the numerical results of forecasting real-life data. It is observed that the granular computing approach may successfully support the probabilistic time series analysis and forecasting. In the opinion of the authors of this paper, the intelligent combination of the granular computing methods and the Bayesian modeling is a promising direction for the future research on the time series analysis and forecasting.

The structure of this paper is as follows. Next section summarizes state-of-the-art of the soft computing techniques supporting the time series analysis and forecasting. Section 3 recalls basic definitions related to the time series analysis and granular computing. Section 4 presents the proposed Bayesian Granular Computing...
Table 1

<table>
<thead>
<tr>
<th>Search field by article title, keywords, abstract</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>'granular computing' AND 'Bayesian' AND 'time series'</td>
<td>1</td>
</tr>
<tr>
<td>'soft computing' AND 'Bayesian' AND 'time series'</td>
<td>6</td>
</tr>
<tr>
<td>'granular computing' AND 'time series'</td>
<td>40</td>
</tr>
<tr>
<td>'soft computing' AND 'time series'</td>
<td>239</td>
</tr>
<tr>
<td>'Bayesian' AND 'time series'</td>
<td>2704</td>
</tr>
<tr>
<td>'granular computing'</td>
<td>2303</td>
</tr>
<tr>
<td>'soft computing'</td>
<td>9631</td>
</tr>
<tr>
<td>'time series'</td>
<td>127,933</td>
</tr>
</tbody>
</table>

(B-GC) method for Time Series Forecasting. The description of the experiments is gathered in Section 5. This paper concludes with discussion in Section 6.

2. State-of-the-art

Within this review, the focus lays on the approaches processing information that can be easily interpreted in natural language. First, we review the main developments and objectives of the time series data mining. Secondly, methods for the human-consistent time series forecasting are reviewed. Finally, we shortly discuss the Bayesian approach to the time series analysis.

2.1. Time series data mining

The Data Mining and Knowledge Discovery adapt the soft computing techniques to provide the human-consistent description of large datasets, that can be easily understood by users. Following Zadeh [8], the Soft Computing is based on the Computational Intelligence, and together with the Hard Computing (that is based on the Artificial Intelligence) form the Machine Intelligence. The field of Computational Intelligence was formally initiated in 1994 during the IEEE World Congress on Computational Intelligence in Orlando. Computational Intelligence (CI) is defined as a methodology involving computing that exhibits an ability to learn and/or deal with new situations such that the system is perceived to possess one or more attributes of reason, such as generalization, discovery, association, and abstraction cf. [9]. Data mining usually leads to some generic and conceptually meaningful information entities like information granules. The theory of information granulation [5] is inspired by the ways in which people granulate information and reason about it. The general framework of Granular Computing [6,10–12] studies the information granulation and the information granules. It endows the information processing with a facet of human-centricity.

As defined by Zadeh [5], the information granule is a clump of points (objects) grouped together by indistinguishability, similarity, proximity or functionality. The information granules are formally described with the use of fuzzy sets, intervals, rough sets, shadowed sets, probabilistic sets, etc. Pedrycz and contributors [13,14] study the optimal granular representation of time series. In [15], the authors review the granular computing developments in general. In [16], the fast interval predictors for large-scale, nonlinear time series with noisy data using fuzzy granular support vector machines are presented. The knowledge discovery process from sequential data may be divided into segmentation, clustering, classification of identified meaningful intervals or patterns, detecting anomalies, frequent patterns and discovery of association rules. For the review of the time series data mining methods, see e.g., Fu [17]. The comparative analysis of the time series representations and the similarity measures is provided in [18]. Batyrshin et al. [19] introduce an interesting perception-based approach to time series data mining.

Linguistic summaries in the sense of Yager [20] are an example of information granules. Linguistic summaries describe general facts about evolution of time series with (quasi) natural language e.g., Among all increasing segments, majority are long, and are intuitive and easily interpretable for people. Kacprzyk et al. [21–23] employ the classic calculus of linguistically quantified propositions for the development of the linguistic summaries.

Another example of the information granules are linguistic descriptions based on the Computational Theory of Perceptions [4] and the Systemic Functional Linguistics by Halliday. The linguistic descriptions [24–28] are characterized by more complex semantic and lexico-grammar structures than linguistic summaries, e.g., Before the knee lesion, the gait quality is high because the gait symmetry is medium and the gait homogeneity is high cf. [26]. In [29], the authors further develop the concept of the Granular Linguistic Model of a Phenomenon and provide basic architecture of a computational system to generate the linguistic descriptions.

Apart from the linguistic summaries and descriptions, another example of imprecise information entities mined from the time series datasets are temporal patterns. Höppner et al. [30] propose the approach for the identification of frequent patterns based on the Allen’s temporal logic [31]. In [32], the authors focus on enhancing patterns with a context information and operating on block constraints instead of Allen’s relations e.g., If A happens before B and in the meantime we do not observe C, then we have a failure of class X. Schockaert et al. [34] fuzzyfy Allen’s temporal interval relations and propose a complete framework to represent, calculate and reason about the temporal relationships for fuzzy intervals.

Temporal patterns are used to build the association rules, that may be exemplified by If exchange rate for currency X decreases slowly, the number of international travels will increase rapidly. In [33], Agrawal et al. propose an interesting algorithm for discovery of rules in sets of items. Methods for the frequent patterns and association rules recognition have been successfully applied to describe huge datasets in different contexts, e.g., [35–39].

The primer goal of the data mining in most of the cited papers is the interpretation of huge datasets by generation of meaningful information granules. The retrieval of such information helps to increase the understanding of the dataset and may reduce the storage space. In the opinion of the authors of this paper, the generic methodology on how to benefit in statistical forecasting from the results of the time series data mining shall be further investigated in the literature.

2.2. Time series forecasting

Traditional autoregressive and moving average (ARMA) processes [40] are the most popular simple probabilistic models for forecasting, and though, very successful in applications. In the recent application [41], D’Urso et al. apply the autoregressive estimates of models for the time series classification and clustering. One of the attractive features of the Box–Jenkins approach to forecasting is that the class of its processes is rich, and it is usually possible to find a process or a combination of processes, which provide an adequate description to the dataset. Nonetheless, the identification of the mathematical model and the specification of its required parameters may be time consuming and difficult to interpret for practitioners (experts of the field) involved in the forecasting process.

To alleviate this problem and increase the overall understanding of the probabilistic time series analysis, there have been proposed some hybrid systems supporting the model selection like e.g., [42,43]. In [44], the autoregressive and moving average models are combined with the fuzzy rule based approach using Mamdani inference mixed with some techniques of counting in fuzzy sets. Chen et al. [45] propose an approach for the automatic generation of ARMA models.
of the fuzzy rule base. In [46], the imprecise linguistic information is incorporated into the forecasting process by means of the linear regression. In [47], the authors propose a hybridization of intelligent techniques such as ANNs, fuzzy systems and evolutionary algorithms and the use of fuzzy rules to identify the ARMA model for prediction.

Alternatively, to reflect the imprecision about the dataset and the model itself, there have been studied fuzzy forecasting models. Tanaka [48] suggests the fuzzy regression. Tseng et al. [49,50] propose to use the fuzzy triangular numbers instead of crisp parameters in the ARIMA models. Song and Chissom introduce fuzzy time series [52], that are based on the concepts of the linguistic variables and fuzzy relations. The approach assumes the split of the universe into intervals, defining the linguistic variables and relations, fuzzification of the observations, forecasting and then, defuzzification of the outputs. Further modifications to the Song and Chissom’s approach are proposed in e.g., [53–56]. Chen et al. [57] introduce the fuzzified variation and the fuzzy logical relationship groups. The fuzzy time series are linked to the linguistic concepts, and therefore, are usually easy for interpretation. However, fuzzy time series scheme of representation of the fuzzy logical relationships may not be relevant for the complex forms of the information granules like the linguistic summaries or linguistic descriptions.

Hopefully, the intelligent combination of the soft computing methods with the statistical time series analysis is gaining attention in the literature. Other examples of the applications of the soft computing methods for the time series forecasting include e.g., the Fuzzy Transform byPerfilieva [58,59], the neural networks approaches [60–62], fuzzy cognitive maps [63], rough sets [64], genetic algorithms and support vector regression [65,66] or fuzzy clustering and fuzzy rule interpolation techniques [67].

Within this paper, we introduce an innovative approach to identify the a priori probability distributions for the Bayesian forecasting with the use of the information granules mined from the time series datasets.

2.3. Bayesian approach to time series analysis

The practitioners are very often posed to the dilemma of choice between the wealth of mathematic models for forecasting for the imprecise real-world data. Clemen [68] review the aggregation methods used for the predictive purposes concluding that the Bayesian approach in the risk analysis is a very powerful tool. One of the main advantages of the Bayesian approach is the possibility to describe the data imprecision, also about the model selection, in terms of the a priori probability distributions. However, the proper selection of a priori distributions is essential for the overall performance.

Geweke [1] describes the Bayesian inference for the time series and provides a detailed review of the Bayesian forecasting elements with numerous references. Prior distributions are usually assumed subjectively. Doan, Litterman and Sims [69] introduce a method to calculate the priors based on the simple statistics. The Bayesian approach for inconsistent information is discussed also by Beer and contributors in [70,71].

3. Basic definitions

In this section we recall basic definitions related to both, the Bayesian time series analysis and the granular computing.

3.1. Bayesian time series analysis

The purpose of the time series analysis is to describe the model (stochastic process) that produced the time series.

Discrete time series [40]

Let \( Q = \{ q_1, q_2, \ldots, q_q \} \) denote a finite set of objects in a considered domain. The properties of objects are measured by observables \( P = \{ p_1, p_2, \ldots, p_p \} \). Discrete time series \( \mathbf{y} = \{ y_t \}_{t=1}^{T} \in \mathcal{V}_y \) is a sequence of observations of given object’s property \((a, p)\) such that \( a = 0 \) and \( p = P \) measured at successive \( t = 1, 2, \ldots, n \) moments and at uniform time intervals. For each \( t \in T \) the observation \( y_t \) is a realization of the random variable \( Y_t \) defined on the probability space \((\Omega, A, P)\). A sequence of \( Y_t \) formulates a stochastic process.

The main principles for the Bayesian analysis are to express all assumptions using probability statements, and then, to design the distributions for the future events conditional on the observed values and a certain loss function. The reflect the uncertainty about identification of the model (process) that generated the time series, we average multiple models.

Bayesian averaging [1]

The posterior density of the vector of interest \( \omega \) conditional on multiple competitive probabilistic models for forecasting \( M = \{ M_1, M_2, \ldots, M_J \} \) is defined as follows

\[
p(\omega | y, M) = \sum_{j=1}^{J} p(M_j | y, M_j)p(\omega | y, M_j) \tag{1}\]

This posterior density \( p(\omega | y, M) \) is a weighted average of the posterior densities of models \( \{ M_1, M_2, \ldots, M_J \} \) which are defined as follows:

\[
p(M_j | y, M) = \frac{p(M_j | y)P(y | M_j)}{\sum_{j=1}^{J} p(M_j | y)P(y | M_j)} \tag{2}\]

We propose the approach to construct the prior model distributions \( p(M_i) \), \( i \in \{ 1, \ldots, J \} \) with the support of the intelligent techniques.

3.2. Granular computing

We now recall definitions related to the selected granular computing aspects.

Information granule [5]

Information granule is a clump of points (objects) grouped together by indistinguishability, similarity, proximity or functionality.

A simple example of an information granule is a labeled interval. Labeled intervals are used to build e.g., linguistic summaries.

Linguistic summary [20]

Let \( A = \{ a_1, a_2, \ldots, a_a \} \) denote a finite set of attributes (eg. dynamics, duration, variability), \( S = \{ s_1, s_2, \ldots, s_s \} \) denotes a finite set of imprecise labels for attributes (eg. low, increasing, short). The protoform-based linguistic summary

\[
Q \ R \ y's \ are \ P \tag{3}\]

consists of quantifier \( Q \) (e.g., most, among all), summarizer \( P \) (attribute together with an imprecise label), qualifier \( R \).

For further details see also [21,22].

Another constructs created from the labeled intervals are temporal patterns. The relations between labeled intervals in temporal patterns may be expressed according to Allen’s interval temporal logic [31]. For any 2 intervals, at least one of the available 13 relationships, e.g., before, meets, during, overlaps is defined. The set of all possible relationships is denoted as I.
Temporal pattern [30]

A pair \( P = (s, R) \) such that \( s: \{1, \ldots, n\} \in S \) and \( R \in P^{n+} \) is called a

The starts

specification

\{

exemplary

\}

\( \{A_1, s_1\}, (A_2, s_2), \ldots, (A_n, s_n) \) such that for every \( i \in \{1, \ldots, n\} \) \( s(i) = s_i \) and \( R[i, j] = \text{ir}(A_i, A_j) \)

Frequent temporal patterns are building elements for the association rules.

Association rule [33]

An association rule is an implication of the form \( P \Rightarrow C \) such that \( P, C \) are sets of items representing premise and conclusion, respectively.

There have been proposed in the literature different measures to evaluate the quality of the information granules depending on their type and the considered evaluation purposes. Table 2 presents exemplary definitions of quality measures.

4. Bayesian approach meets granular computing

The information granules like linguistic summaries, frequent patterns or association rules are usually in line with the visual pattern recognition and perception capabilities of humans, and therefore, seem appealing to support not only the interpretation of time series, but also the time series forecasting. From the range of the statistical methods, we adapt the Bayesian approach, which has been proven successful in many real-life applications. Multiple models are included in the forecasting process through the Bayesian averaging and the focus of the approach is to establish the prior probabilities for the competitive models.

The input data for the proposed Bayesian Granular Computing (B-GC) method for forecasting are multivariate time series \( Y^k = \{Y^k_{t=1}^{n}, \ldots, Y^k_{t=n+h}\} \) being a k-vector of discrete time series. We assume that the preprocessing, that is, the normalization of the time series and handling of the missing data, is completed beforehand.

The output data are both, forecasts \( Y^k_h = \{Y^k_{t=1}^{n+h}, \ldots, Y^k_{t=n+h}\} \) and the descriptions of the imprecise information used in the forecast generation procedure.

The overview of the proposed Bayesian Granular Computing (B-GC) method for forecasting is presented in Fig. 1. The procedure starts with the definition of the linguistic concepts, then mining for the information granules, identification of the probabilistic model, and finally, the Bayesian posterior simulation with the prediction generation procedure.

We show that the imprecise information processed with the selected soft computing methods is supportive for the construction of a priori model probability distributions required for the Bayesian forecasting.

4.1. Defining of concepts

We start from defining the linguistic variables related to the imprecise information. Then, we define the selected template probabilistic predictive models used to build the training database.

Elements of the algorithm for the definition phase are as follows:

1. Define linguistic variables for imprecise labels
2. Define probabilistic models and build training dataset

4.1.1. Define linguistic variables for imprecise labels

The first step of the definition phase requires providing the specification of trend attributes \( A \) and the fuzzy sets for the interpretation of imprecise labels \( S \). Membership functions for all the linguistic variables are established. We assume that the interpretation of linguistic variable for a given property \( p \in P \) of an object \( o \in O \) in a given domain is known and does not change in time.

4.1.2. Define probabilistic models and build training dataset

The second step requires defining the template probabilistic predictive models \( M = \{M_1, M_2, \ldots, M_J\} \). Then, the training time series dataset is built based on models from \( M \). The definitions of the a priori distributions for all parameters related to the models are saved as \( M^0 = \{M^0_1, M^0_2, \ldots, M^0_J\} \). The \( M^0_1 \) is a resulting s-vector of the sample time series generated from the template models \( M \), and \( C^0_i \) is the respective s-vector denoting the template models that produced the time series from \( T^0_n \).

4.2. Time series data mining

The data mining phase aims at processing of the crisp time series datasets into meaningful information objects. Elements of the algorithm are as follows:

1. Mine for the information granules
2. Discover linguistic information
3. Calculate the validity matrix

4.2.1. Mine for the linguistic information granules

Data mining is performed successively on the training dataset \( T^0_n \) and then, for the time series to be predicted \( V^k_h \), to find meaningful information granules. As a result, e.g., labeled intervals are mined from the time series datasets.

4.2.2. Discover linguistic information

Secondly, human-consistent and more complex information granules are constructed like linguistic summaries, descriptions, frequent patterns or association rules. The selection of the considered types of information granules depends on the application context. As a result, the \( k \times 1 \) matrix \( L^k \) of the information entities discovered for \( V^k_h \) time series is created and the \( s \times 1 \) matrix \( L^s \) respectively. The linguistic information may be also formulated with the support of experts of the field thanks to the human-computer interaction.

4.2.3. Calculate validity matrix

Finally, the set of the considered quality measures is defined \( Q^M = \{q_1, \ldots, q_w\} \) and for each instance of the information granule in \( L^k \) and \( L^s \), their quality is measured. The applied quality measures shall be chosen adequately depending on the considered types of linguistic information. The results form the validity matrices \( V^k \) and \( V^s \) of size \( k \times z \) and \( s \times z \), respectively, where \( 1 \leq z \leq \text{card}(Q^M) \).

4.3. Probabilistic model identification with soft computing

The prior model distributions \( p(M_i), i \in \{1, \ldots, J\} \) are established with the use of the classification methods. The classification problem is defined and solved for the vectors of the information granules.

The algorithm for the probabilistic model identification phase consists of the following elements:

1. Classify the probabilistic models
2. Identify prior distribution of the unobservables

4.3.1. Classify the probabilistic models

The rows of the validity matrices \( V^k \) and \( V^s \) are feature vectors in the classification tasks. The classifier is trained on \( V^k \), and then applied for \( V^k \) to identify the probabilistic models.

First, we assign categories (classes) related to \( T^m_n \) that correspond to the template probabilistic models \( M \) indicated by \( C^0_i \). Support Vector Machines (SVM) [72] are adapted to analyze the training
Table 2: Exemplary quality measures for the entities of information granules.

<table>
<thead>
<tr>
<th>Linguistic information</th>
<th>Quality measure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Linguistic summary</strong></td>
<td>- the degree of truth [20]: ( T = \mu_Q \left( \sum_{i=1}^{n} \left( \mu_x(y_i) \right) \right) ) (4) where ( \mu_x, \mu_y ) are the membership functions ( \mu_x, \mu_y : \Omega \rightarrow [0,1] ) determining the degree to which ( R, P ), respectively, are satisfied for the time series ( y ) at the given moment ( n ).</td>
</tr>
<tr>
<td></td>
<td>- the degree of imprecision (fuzziness) [21]: ( T_2 = 1 - \frac{\sum_{j=1}^{m} \frac{c_j}{c_{\max}}}{\sum_{j=1}^{m} \frac{c_j}{c_{\max}}} ) (5) where ( \mu_x ) is given as a family of fuzzy sets ( P = { s_1, s_2, \ldots, s_m } ) and ( \text{card} ) denotes the cardinality of the corresponding (non-fuzzy) set.</td>
</tr>
<tr>
<td><strong>Temporal pattern</strong></td>
<td>- the support of temporal pattern [30]: ( \text{supp} ) - the total time in which the instances of ( P ) can be observed in the sliding window.</td>
</tr>
<tr>
<td><strong>Association rule</strong></td>
<td>- confidence ( \text{conf} = \frac{\text{supp}_p}{\text{supp}_a} ) (6) where ( \text{supp}_p ) represents the support of the temporal pattern.</td>
</tr>
<tr>
<td></td>
<td>- j-measure compares a prior distribution of ( X ) with a posterior distribution of ( X ) given ( Y=y ), and is defined by the following sum: ( \sum_{z \in {x, \bar{x}}} \frac{p(X=x\mid Y=y) \log \frac{p(X=x\mid Y=y)}{p(X=\bar{x})}}{p(X=\bar{x})} ) (7) where ( \text{IF } Y=y \text{ THEN } X=x ) \text{is the rule, } X, Y \text{ denote random variables, } X \text{ takes values } {x, \bar{x}} \text{ representing two events: the rule is true or false.}</td>
</tr>
</tbody>
</table>

4.3.2. Identify prior distribution of the unobservables

Having trained the SVM classifier, we apply it to classify the sets of the information granules describing the time series \( Y_\Omega \). The rows of the validity matrix \( V_\Omega \) are the feature vectors. The classification assumes that a score is assigned to each of \( J \) possible categories (models).

\[ \text{Score}(V_\Omega(i),J) = B_i V_\Omega(i) \] (8)

where \( V_\Omega(i) \) is the feature vector being \( i \)-th row of \( V_\Omega \), \( B_i \) is the vector of weights corresponding to category \( j \), and \( \text{Score}(V_\Omega(i),J) \) is the score associated with assigning instance \( i \) to the category (model) \( j \). The resulting scores for all time series to be predicted form the \( k \)-vector \( C_\Omega = [\text{Score}(V_\Omega(1),J), \ldots, \text{Score}(V_\Omega(k),J)] \). The prior model distributions \( p(M_j) \) are estimated with the scores from \( C_\Omega \). Finally, the prior distributions for the parameters \( p(\theta_{M_j} | M_j) \) for \( M_j \in M \) are adapted from \( M^0 \).

4.4. Bayesian prediction

The last phase of the proposed approach assumes prediction basing on the identified prior distributions for the probabilistic models with the Bayesian inference. The elements of the algorithm for the prediction phase are as follows:

1. Simulate posterior distributions
2. Generate forecast and describe its components

4.4.1. Simulate posterior distribution of the unobservables and the vector of interest

In line with [1], the posterior density of the vector of interest

\[ p(\omega | y^o, M) = \sum_{j=1}^{J} p(M_j | y, M) p(\omega | y^o, M_j) \] (9)

\[ p(\omega | y^o, M_j) = \int_{E_{M_j}} p(\omega | y^o, \theta_{M_j}, M_j) p(\theta_{M_j} | y^o, M_j) d\theta_{M_j} \] (10)

and the posterior density of the unobservables \( \theta_{M_j} \)

\[ p(\theta_{M_j} | y^o, M_j) = \frac{p(\theta_{M_j} | M_j)p(y^o | \theta_{M_j}, M_j)}{p(y^o | M_j)} \] (11)

in which \( \theta_{M_j} \in E_{M_j} \) is a \( k_{M_j} \times 1 \) vector of unobservables. We adapt the Markov Chain Monte Carlo (MCMC) posterior simulator that yields pseudo-random sequence of the vector of interest to estimate the posterior moments.

4.4.2. Generate forecast and describe its components

The estimates from the formula (9) are used to generate point forecasts and their errors according to the model definitions \( M \). Once the prediction is calculated, additional information that supported the forecasting process is provided. This information consists of the objects with the information granules and the quality measures, and therefore, shall be easily understood and helpful for the experts of the field responsible for decision making based on the forecasting results.

---

Please cite this article in press as: O. Hryniewicz, K. Kaczmarek, Bayesian analysis of time series using granular computing approach, Appl. Soft Comput. J. (2014), http://dx.doi.org/10.1016/j.asoc.2014.11.024
5. Experimental results

The performance of the proposed Bayesian Granular Computing (B-GC) method for Time Series Forecasting is illustrated with the experiments on the artificial and real-life time series datasets. The following forecast accuracy measures are adapted to evaluate the performance:

- Mean Absolute Percentage Error (MAPE)

\[
MAPE = \frac{1}{h} \sum_{i=1}^{h} \frac{100e_{t,i}}{X_{t,i}}
\]

- Symmetric Mean Absolute Percentage Error (sMAPE)

\[
sMAPE = \frac{1}{h} \sum_{i=1}^{h} \frac{200|e_{t,i}|}{(X_{t,i} + F_{t,i})}
\]

The goal of the experiments is to provide an illustrative demonstration of the proposed approach and to evaluate its forecasting accuracy.

The program has been implemented in Python with the support of NumPy, SciPy and the PyMc extension modules. Linguistic summaries are generated with the Trend Analysis System [22].

5.1. Artificial datasets

An illustrative demonstration of the proposed approach is provided for the autoregressive datasets. This simulation study enables to investigate in detail the theoretical aspects, like importance of the prior knowledge, of the proposed approach.

Time series datasets are created from the autoregressive processes \( M = \{M_1, M_2, \ldots, M_{10}\} \) according to the following formula:

\[
y_{t} = \sum_{i=0}^{r} \phi_i y_{t-i} + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma^2), \quad y_0, \ldots, y_{n-R}.
\]

The autoregressive coefficient \( \phi_i \in \{0.1, \ldots, 0.9\} \) for \( M_1 \), \( \phi_i \in \{0.1, \ldots, 0.9\} \) for \( M_2 \), \ldots, \( \phi_i \in \{0.1, \ldots, 0.9\} \) for \( M_{10} \). As a result, 80 time series are created with 100 observation in each as training dataset \( T_{\frac{n}{80}} \) and another 20 time series are generated randomly as dataset for prediction (testing dataset) \( Y_{\frac{n}{80}} \).

Fig. 2 shows the exemplary time series generated from the 2 competitive probabilistic models from the testing dataset \( Y_{\frac{n}{80}} \). As observed by visual inspection, the characteristics of trends presented in (a) and (b) are different.

The set of attributes \( A \) consists of length, dynamics of change, variability, and the imprecise labels for the attributes \( S \) as follows: short, medium, long, increasing, constant, decreasing, low, moderate, high. The information granules considered in this experiment are the extended linguistic summaries, that may be exemplified by Most increasing trends are short. We apply the quantifier \( Q = \langle \text{most} \rangle \). The resulting sets of linguistic summaries \( L_{\hat{U}} \) and \( U_{\hat{L}} \) consist of linguistic summaries for each time series. The degree of true \( QM = \langle T \rangle \) is applied to evaluate the quality of the considered information granules and to calculate the validity matrices \( V_{\hat{U}}, V_{\hat{L}} \).

The crisp time series datasets are segmented, then summarization is performed. We group the sets of the information granules by the template models that produced the time series. It is observed that some types of linguistic summaries may be good discriminant for the model classification analysis. Table 3 presents average degree of truth \( T \) within these groups. For example, the degree of truth for the summary Among all trends, most are short amounts to 1 for time series generated from \( M_1 \) and to 0.49 for time series generated from \( M_{10} \). At the same time, there exist linguistic summaries, e.g., Among all long trends, most are constant that do not bring value to the classification in this context because the degree of truth is 1 for any model.

The next step of the proposed approach is the classification of the time series by probabilistic models based on the sets of the linguistic granules. For the clarity reasons, we group the template probabilistic models into 3 classes as follows: \( AR \) with weak positive autocorrelation \( C_1 = \{M_1, M_2, M_3\} \), \( AR \) with medium positive autocorrelation \( C_2 = \{M_4, M_5, M_6, M_7\} \) and \( AR \) with strong positive autocorrelation \( C_3 = \{M_8, M_9, M_{10}\} \), and these classes are processed in the remaining of this experiment.

The classification scores based on the linguistic summaries for time series \( Y_{\frac{n}{80}} \) are presented in Table 4. For example, for TS1, which

![Fig. 2. Sample time series generated from the autoregressive process (a) M1 (b) M10.](image-url)
Table 4
The SVM classification scores based on the sets of linguistic summaries for the time series Y_{10}^{20} (dataset for prediction) in the 3-class problem.

<table>
<thead>
<tr>
<th>TS</th>
<th>True class</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
<th>C9</th>
<th>C10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score(C1)</td>
<td>0.68</td>
<td>0.64</td>
<td>0.55</td>
<td>0.88</td>
<td>0.79</td>
<td>0.27</td>
<td>0.43</td>
<td>0.34</td>
<td>0.66</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>Score(C2)</td>
<td>0.20</td>
<td>0.24</td>
<td>0.35</td>
<td>0.10</td>
<td>0.14</td>
<td>0.54</td>
<td>0.44</td>
<td>0.41</td>
<td>0.27</td>
<td>0.54</td>
<td></td>
</tr>
<tr>
<td>Score(C3)</td>
<td>0.12</td>
<td>0.12</td>
<td>0.10</td>
<td>0.02</td>
<td>0.06</td>
<td>0.19</td>
<td>0.12</td>
<td>0.25</td>
<td>0.08</td>
<td>0.21</td>
<td></td>
</tr>
</tbody>
</table>

was generated from C1. The classification scores assigned to models C1, C2 and C3 are 0.68, 0.20 and 0.12, respectively. It is observed that the classifier yields very good results for the identification of C1, in 5/6 cases the highest score indicates the true model. The overall SVM classification accuracy (and its error) measured by 5-fold-cross validation for the training dataset Y_{10}^{50} amounts to 0.55 (0.09).

The final step of the approach is the posterior simulation and the verification of forecasting accuracy. The summary of the forecasting accuracy is demonstrated in Table 5. As demonstrated in Table 5, the forecasting error for the Bayesian Granular Computing (B-GC) method is 7 times smaller than for the naive method. More importantly, for the considered artificial dataset, the approach also outperforms the traditional Bayesian averaging with uniform prior distributions. The mean absolute percentage error for the proposed approach and the approach with uniform distributions are 0.0528 and 0.0563, respectively.

5.2. Real-life datasets

The comparative forecasting analysis is performed for the real-life time series from the M3-Competition Datasets Repository by Makridakis and Hibon [73]. We use the subset of the 5 first yearly time series that have length 47. The performance of the proposed approach is compared to the selected benchmark methods that have been claimed as one of the best according to the Makridakis’ study. These methods are briefly presented in Table 6. 

The results presenting the forecasting accuracy of the proposed method and the competitive ones are gathered in Table 7. As demonstrated in Table 7, none of the benchmark methods outperform the proposed B-GC method for all time series. On average, the proposed method scored a very good sMAPE result of 3.81, which is the best among the competitive methods. At the same time, it is observed that for some time series there exist methods that deliver more accurate forecast, so there is still potential for improvement. For example, when considering the time series N156, the sMAPE amounts to 2.33 and 1.9 for the proposed B-GC and the Theta method, respectively.

We conclude that the proposed approach delivers very competitive results in terms of the forecasting accuracy. However, it should be noted, that the obtained results are limited to the autoregressive models, and therefore are of preliminary character. In the future, they should be confirmed also on an extensive set of benchmark datasets to yield a complete empirical validation of the general approach.

6. Conclusion

In this paper we have reviewed the main challenges and the recent developments of incorporating the soft computing techniques into the statistical time series analysis and forecasting. We have introduced the Bayesian Granular Computing (B-GC) method for Time Series Forecasting, that aims at the identification of the a priori model distributions with the use of the information granules mined from the time series datasets with soft computing techniques.

One of the main advantages of the proposed solution is its human-centricity, being a result of including the imprecise information granules. This is especially important in practice for experts involved in the forecasting process for large datasets. The data mining and classification methods employed in the work provide useful information granules. It is shown that the considered linguistic summaries constitute intuitive and very informative knowledge.
Future research assumes the incorporation of other types of information granules and probabilistic models into the forecasting framework. We conclude that the intelligent combination of the granular computing achievements and the Bayesian modeling is a promising direction for the future research on the time series analysis and forecasting.

Acknowledgements

The authors thank Janusz Kacprzyk and Anna Wilbik for their advice and support concerning the Trend Analysis. Katarzyna Kaczmarek is supported by the Foundation for Polish Science under International PhD Projects in Intelligent Computing financed from the European Union within the Innovative Economy Operational Programme 2007–2013 and European Regional Development Fund.

References

[9] R. Eberhart, P. Simpson, R. Dobbins, Computational Intelligence, PC Tools Aca-


