Usage of a BART algorithm and cognitive services to research collaboration platforms

I A Voronkov¹ and S E Saradgishvili
Peter the Great St. Petersburg Polytechnic University, 29 Polytechnicheskaya St., St. Petersburg, 195251, Russia

E-mail: iliaftk@outlook.com

Abstract. This article examines the possibility of extending the collaborative filtering method in shared information systems. The need to build recommendations dictates the volume and the intensity of the emergence of more and more new information. In the scientific world, the possibilities of building recommendations are widely explored, and this work presents the authors' vision of a possible direction of development, which is based on a preliminary iteration with the highlighting of keywords.

1. Introduction
The research process is reduced to the systematization of data obtained using various research methods (observation, experiment, and others). In this work authors try to apply a similar algorithm to specific systems, which in the study are called collaboration platforms. Such systems may include information services, which basically contain the idea of a group of users working on one or several objects in this system. The simplest example of the implementation of such a system can be an ordinary dialogue, only it occurs by means of messaging on electronic devices, and it is possible to predict the behavior of the participants in the dialogue using an analysis of previously received messages. In this work, the authors sought to emulate the work of the enterprise (company, plant) using modern information resources:

1) Office 365 provides work with e-mail (as the most common way of correspondence in companies and provides the ability to use the chat system (Skype for business, Teams). The authors deliberately turned off the use of media files to simplify the work.
2) Emulation of the operation of an industrial device (conveyor) [1].
3) Document Management System based on the SharePoint online package [2].
4) SAP modules with Azure blockchain [3].

These components of the system emulated problems, worked with documents (real users), and processes for maintaining contracts. Based on the results obtained, time series analysis was applied using the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology [4]. The effectiveness of using this approach is found as a module of the difference in the value of the estimate at the forecasting stage with the value at each stage of the iteration.

1.1. Modeling and investigations approach in collaboration systems
Most often, researchers focus on highlighting interactions between users of the system to fully describe and understand the processes involved [5,6]. The topic of research on the impact of work in shared access systems is very popular in scientific papers. This emphasis can be explained by the study of the impact of collaboration between professional groups on issues [7–11]. The implementation of the technical component of the solution for the creation, support and application of work algorithms in systems is also widely used in articles [12][13]. The technology that is used in this work to form an enterprise model is also used to develop training systems, where the factor of perception of educational material in the system plays a key role [14]. Despite the presence of works on various topics, most of them focus on a specific issue based on the purpose of the study: the
technical structure or the impact on the work of the team. This work sets its task, using the experience of previous years, to look at the process of forming interactions from a new level: the system at the same time is limited in itself and at the same time can be affected from outside.

2. Modeling interactions in a collaboration environment

2.1. Usage of CRISP-DM

We use CRISP-DM to implement a model for collecting and analyzing information that is generated by users in collaboration platforms. Fig. 1 presents 6 steps that are in the cycle of this approach.

![Figure 1. Schematic implementation CRISP-DM](image)

This method represents a continuous cycle of work on data, during which N is performed several times, the error in the predicted data tends to zero based on the information obtained in the previous step. This method is one of the most common practices for modeling the data mining process.

2.2. Regression Tree

A regression tree is a class of regression models that allows you to divide the input space of factor variables into segments. Subsequently, the whole chain of the regression model can be supplemented and processed for each of the nodes representing the regression function in an intuitive visual form [15,16]. Let us introduce a definition of how the interaction model in a collaboration environment relates to a process expression in the form of a tree:

1) The internal node of the tree is a description of the rules for partitioning the space of explanatory variables. (Example: An element of the “message” system is an indivisible unit for selecting data for a finite segment of users. One author can have many messages, while each message has a single author in a simplified model).

2) Tree Leaves - Own Model of Local Regression.

3) Branch - transition conditions between nodes.

To apply the regression tree, we will build connections in the collaboration system on the principle of dividing and isolating a set of information that does not intersect with other classes of the system. We will carry out the division sequentially until there remains the possibility of isolating a new class from the essence of the entity. Each model can be defined and limited by itself. Types of decision trees for collaboration are presented in Table 1.
Table 1. Types of decision trees

| Tree name         | Description                                                                 | Example in systems |
|-------------------|-----------------------------------------------------------------------------|--------------------|
| for classification| the result of the prediction is the data ownership class                    | Office 365 message |
| for regression    | the result in this case is the predicted value of the target function       | Media File in SharePoint |

2.3. BART algorithm

Consider a Bayesian approach to evaluating nonparametric functions using regression trees. This algorithm allows us to generalize the regression tree that was allocated above and time series for iterations using the CRISP-DM method. BART is a combination (C&RT) [17] of the algorithm and the standard autoregressive integrated moving average (ARIMA) models and their components (AR, MA). The SETAR and ASTAR models are linear models of homogeneous models (Inst message, office 365 Exchange file) that build several adaptive regression splines (MARS) based on time series in a single iteration of the processing complex. BART has two main differences from the SETAR and ASTAR models:

1) Error estimates for BART models can differ from each other both for each node and for each iteration of the cycle.
2) BART is characterized by a gap between the models of autoregression.

To convert the model according to the time series, we will use the conversion method, where the resulting variable $CD_t$ corresponds to the sum of the previous value $CD_{t-1}$ and the delayed value $CD_{t-p}$ adjusted for the Sentiment coefficient $\beta$, which in turn, it is a cumulative estimate for each node of the system over time $t$ obtained using Azure Data Text functions. Release the rules for dividing input data into segments (Fig. 2).

Figure 2. Autoregressive tree building diagram
Most algorithms use recursive separation of the data on which training takes place. BART uses an iterative construction method. Also, in our example, we add the Sentiment $\beta$ coefficient at each step of the calculation, which is in the range of values $[0,1]$ and is cumulative. This coefficient is calculated for each individual and influences the result in each node. 0 is the correspondence of an absolute negative comment in the system (message in correspondence, with pronounced indignation or discontent). 1 is a positive value. During the simulation, no 0 to 1 were met.

Consider a data processing algorithm:

Step 1: build a regression tree for the root value (the entire Collaboration system considered). The construction of a regression tree begins from a single value (root node), which is defined as the Median (ME, second quartile $Q_{50\%}$) of the entire time series $CD_t^i$ and is calculated the equation (1).

$$ME = Q_{50\%} = \frac{1}{2}(CD_{i_{\min}} + CD_{i_{\max}}) + \beta \quad (1)$$

In this expression, the median is the sum of a real number with a probability of exceeding an arbitrary size equal to 0.5 and a non-negative value $\beta$, which tends to 1 with an ideal system of interaction in the system.

Step 2: for each unprocessed node we find the best partition. The partition itself will be selected based on a predefined rule.

The split rule selection mechanism is like the C&RT algorithm. The difference lies in the rule for selecting the criterion for evaluating the termination of cleavage. During testing, we used the information criterion for a better separation based on the entropy indicator, because it prefers options with less complexity of the tree. This algorithm will determine an entropy information gain. When using BART, any vertex in the tree (except the root) has two children. The final chain is built from tree nodes from top to bottom and an informational assessment of the node predictor occurs, dividing the time series into a subset of experiments.

Entropy criterion. The value for the sample $CD_t^i$ is calculated by the formula (2):

$$H^*(P,N) = H(P/(P + N)) + N/(P + N) \quad (2)$$

After receiving information about the current node and its predecessors, the entropy value is calculated by the following relation (3):

$$H_{\varphi} (P, N, p, n) = (p + n/P + N)H^*(p, n) + (P + N - (p + n)/P + N)H^*(P - p, N - n) \quad (3)$$

$P$ is the number of objects in the subset $C$, $p$ is the number of objects, all objects that correspond to $p \in P \in C, n \in N$ are the total number of nodes in the system at each iteration step $n \in N$.

The change in entropy is calculated by the formula (4) which shows the amount of information corresponding to class $C$ and not corresponding to this separation:

$$IGain_{C}(\varphi, CD_t^{C}) = H^*(P, N) - H^*(P, N, p, n) + \beta \quad (4)$$

For the early Q information criterion, we use the extended Bayesian information criterion [18] (5):

$$EBIC = J \ast (\ln n + 2 \ln p) + n \ln n (SSE/n) \quad (5)$$

SSE – is the sum of squares of the residuals of the model; $J$ is the number of model parameters; $n$ is the training number of examples; $p$ is the mathematical ratio of the number of vertices in the tree and the unifying criteria. In a first approximation, we will use multiplication. We will calculate the EBIC value until the next value is less than the previous one.

Step 3: continue dividing the model and consider the change in entropy until the EBIC value at step $n$ is greater than at step $n-1$ and is a real number. As soon as these conditions are no longer fulfilled, the algorithm completes its work. The tree is considered formed.

We will study the effectiveness of the BART$+\beta$ approach with traditional ARIMA (Autoregressive integrated moving average) algorithms. We will build a series of experiments on the leaves of a tree according to the rule (6):
\[ F(L)(1^d + d(n - 1)L^{d-1})X_t = \mu + \theta(L)\varepsilon_t, \varepsilon_t \sim N(0, \sigma^2) \quad (6) \]

\( X_t \) – time series, \( s, L \) – is the lag operator, \( F(L) \) – is the polynomial degree \( p \) from \( L \), \( \mu \) – is the average process value, \( d \) – is the order of process integration \( X_t \). If \( d=0 \), then process \( X_t \) can be described by ARIMA \((p, q)\) or ARIMA \((p, 0, q)\). During the simulation, we also conducted studies using the classical configuration parameters for the algorithms: ARIMA \((1, 0, 1)\) and ARFIMA \((1, d, 1)\).

Step 4: we will describe methods of receiving \( \beta \). In this study, we used the existing Azure Cognitive services functionality to calculate the correction at each new iteration, considering the results of previous values for interactions in the system. This approach is since the calculation of such a coefficient is a separate area and there are several approaches to solving this problem. In our experiment, we use this service according to the black box principle: we do not know the internal structure of the service, but we introduce a correction value in the equation. Options for using the service are presented in Fig. 3 and Fig. 4 [19].

![Figure 3](image3.png)

**Figure 3.** Azure Cognitive Services as service.

![Figure 4](image4.png)

**Figure 4.** Azure Cognitive Services as service for escalation.

### 3. Empirical results

For our model, we carried out we simulated work for 30 days, with the participation of real users (who talked in the mail, worked with internal messengers, used the workflow system on SharePoint, reacted to systems simulating industrial capacities), and taking into account randomly generated events (breakdowns, failures, external contact by mail and more). According to the idea of the experiment, for each interaction node, the target variable will be the log-return value for the time period after the incident. The results of the experiment are displayed in Table 2.

**Table 2.** Simulation results for each individual node.

| System node | Description of Interactions | Interactions with other system nodes | The presence of interaction from an external source | The difference between the predicted value and the real |
|-------------|-----------------------------|-------------------------------------|---------------------------------------------------|-----------------------------------------------------|
| Exchange    | Email Correspondence, Chats, | True                                |                                                   |                                                     |
|                  |                 |                |        |
|------------------|-----------------|----------------|--------|
| emails           | SharePoint      |                | 0.245  |
| Chats            | DMS             | Emails         | 0.113  |
| SharePoint       |                 | Exchange       | 0.117  |
| DMS              |                 | SAP            | 0.430  |
| Imitation        |                 |                |        |
| conveyor belt    |                 |                |        |
| on Raspberries   |                 |                |        |
| SAP modules      |                 |                |        |
| Customer workflow|                 | Conveyors      | 0.175  |

The closer the value in the last column to zero, the more accurate the forecast gives the system about the results of future interaction. Note some interesting features in the experimental results:

1. The best value for individual nodes showed a study of the chat system. This node is not critical for maintaining the efficiency of the enterprise model, as Email can be duplicated. This fact and the fact that in the calculation we used a cognitive analysis of chat text messages that can be deeply analysed by the system at each iteration can explain the leadership in using this approach for text messages.

2. The relatively high (second) place of SharePoint DMS is due to the formalization of processes, where in the workflow for processing documents it is not possible to introduce serious disturbances (comments). At the same time, the usefulness of using cognitive services in such formalized systems can be questioned.

3. The relative success of modeling in SAP modules is also explained by the high formalization and standardization of processes. The result obtained (the value is worse than that of the previous paragraphs) may indicate the imperfection of the method when working with external disturbances and the difficulty of predicting emergency situations for the conveyor.

4. The simulation results for the conveyor may indicate the imperfection of the applied approach for a particular node.

### 4. Conclusion

During the research, an enterprise model was developed that uses various collaboration nodes to organize work. To predict the results of individual components based on the previous values, an extended implementation of the BART algorithm was used, which showed effectiveness in studying the results of individual nodes. In the future, this approach can be laid in the idea of applying emergency forecasting, possible safety problems [20] and other critical components in the enterprise. In future studies, the authors will strive to complicate and approximate the enterprise model to the existing ones; at the same time, it is planned to use various methods for working with a large amount of data on time series.

### References

[1] Jo B W and Khan R M A 2018 An internet of things system for underground mine air quality pollutant prediction based on azure machine learning Sensors (Switzerland) 18

[2] Buchal R and Songsore E 2018 Collaborative Knowledge Building using Microsoft SharePoint Proc. Can. Eng. Educ. Assoc.

[3] Wang Y, Lahiri S K, Chen S, Pan R, Dillig I, Born C and Naseer I 2018 Formal Specification and Verification of Smart Contracts for Azure Blockchain
[4] Kurgan L A and Musilek P 2006 A survey of Knowledge Discovery and Data Mining process models *Knowl. Eng. Rev.* 21 1–24

[5] Briggs R, Kolfschoten G, Vreede G and Dean D 2006 Defining Key Concepts for Collaboration Engineering undefined

[6] Frost A 2006 Meetings Around the World : The Impact of Collaboration on Business Performance

[7] Dennis A R, Nunamaker J F and Vogel D R 1990 A comparison of laboratory and field research in the study of electronic meeting systems *J. Manag. Inf. Syst.* 6 107–35

[8] Soller A, Martinez A, Jermann P and Muehlenbrock M 2005 *From Mirroring to Guiding: A Review of State of the Art Technology for Supporting Collaborative Learning* vol 15

[9] Webster J and Staples D S 2006 Comparing Virtual Teams to Traditional Teams: An Identification of New Research Opportunities *Res. Pers. Hum. Resour. Manag.* 25 181–215

[10] Fjermestad J and Hiltz S R 1998 An assessment of group support systems experimental research: Methodology and results *J. Manag. Inf. Syst.* 15 7–149

[11] Fjermestad J and Hiltz S R 2000 Group support systems: A descriptive evaluation of case and field studies *J. Manag. Inf. Syst.* 17 115–59

[12] Beranek P M, Beise C M and Niederman F 1993 Facilitation and group support systems *Proceedings of the Annual Hawaii International Conference on System Sciences* vol 4 (IEEE Computer Society) pp 199–207

[13] Witte E H 2007 Toward a Group Facilitation Technique for Project Teams *Gr. Process. Intergr. Relations* 10 299–309

[14] Joo J, Selingo J and Alamuddin R 2019 *Unlocking the Power of Collaboration: How to Develop a Successful Collaborative Network in and around Higher Education*

[15] Moore D H 1987 Classification and regression trees, by Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. Brooks/Cole Publishing, Monterey, 1984,358 pages

[16] Breiman L 2001 Random forests *Mach. Learn.* 45 5–32

[17] Tong H 1983 Threshold Models pp 59–121

[18] Chen J and Chen Z 2008 Extended Bayesian information criteria for model selection with large model spaces *Biometrika* 95 759–71

[19] Anon Text Analytics | Microsoft Azure

[20] Kim Y Bin, Kim J G, Kim W, Im J H, Kim T H, Kang S J and Kim C H 2016 Predicting Fluctuations in Cryptocurrency Transactions Based on User Comments and Replies ed W-X Zhou *PLoS One* 11 e0161197