mathematics

MDPI

Review

Adaptive Clustering through Multi-Agent Technology: Development and Perspectives

Sergey Grachev [1], Petr Skobelev [1,*], Igor Mayorov [1] and Elena Simonova [2]

[1] Institute of Automation and Information Technologies, Samara State Technical University, 443100 Samara, Russia; sg@kg.ru (S.G.); imayorov@smartsolutions-123.ru (I.M.)
[2] Department of Information Systems and Technologies, Samara National Research University, 443086 Samara, Russia; simonova@smartsolutions-123.ru
* Correspondence: Petr.Skobelev@gmail.com; Tel.: +7-902-372-3202

Received: 13 September 2020; Accepted: 24 September 2020; Published: 27 September 2020

Abstract: The paper is devoted to an overview of multi-agent principles, methods, and technologies intended to adaptive real-time data clustering. The proposed methods provide new principles of self-organization of records and clusters, represented by software agents, making it possible to increase the adaptability of different clustering processes significantly. The paper also presents a comparative review of the methods and results recently developed in this area and their industrial applications. An ability of self-organization of items and clusters suggests a new perspective to form groups in a bottom-up online fashion together with continuous adaption previously obtained decisions. Multi-agent technology allows implementing this methodology in a parallel and asynchronous multi-thread manner, providing highly flexible, scalable, and reliable solutions. Industrial applications of the intended for solving too complex engineering problems are discussed together with several practical examples of data clustering in manufacturing applications, such as the pre-analysis of customer datasets in the sales process, pattern discovery, and ongoing forecasting and consolidation of orders and resources in logistics, clustering semantic networks in insurance document processing. Future research is outlined in the areas such as capturing the semantics of problem domains and guided self-organization on the virtual market.

Keywords: multi-agent technology; adaptive clustering; resource planning and scheduling; if-then rules; logistics; schedule generation; pattern extraction; order consolidation; real-time

1. Introduction

The known clustering task consists of categorizing a given matters collection according to its inner similarity, such that items belonging to the same group (cluster) are more alike to each other in comparison to ones located in additional sets. Such a problem is typically resolved with a predefined number of groups [1–8].

However, in many practical problems, data are received gradually, e.g., via real-time record-by-record fashion in small batches within unpredictable periods. The most representative example of such an application is customer classification in an internet portal, with a large number of visitors contributing a small but significant amount of data during each visit. User patterning aiming for coherent and up-to-date behavior suggests an adaptive and go-ahead clustering process, taking into account the dynamic nature of the data.

The main limitation of many current data mining methods and algorithms is the need to suggest a hypothesis about data configuration in advance that is frequently impossible in the online mode.

Despite considerable progress in data mining and machine learning technologies, the intended methods and algorithms cannot cope with the enormous amounts of data with critical uncertainty and

dynamics. Complexity becomes even more prominent because of the diversity of datasets and growing of application domains. This difficulty also appears since the individual business requirements and computational complexity of the clustering process requires additional resources for the practical applications. An example of such a complexity is given by the authors of [9], where the similarity of user preferences given as a cluster criterion was combined with many other metrics like topology, domain-specific semantics, etc.

The paper provides the first systematic overview and summarizes the results of previously published researches [10–19]. We review and generalize our previously developed principles, methods, and solutions for real-time adaptive clustering based on multi-agent technology, the main principles and approaches of which are set out, for example, by the authors of [20]. It is closely connected to multi-agent technology and solutions of adaptive resource management that proved their efficiency for industrial applications [21]. Following this perspective, some examples of industrial applications are considered, including the pre-analysis of customer datasets in the sales process, pattern discovery, ongoing forecasting, consolidation of orders, and resources in logistics and insurance document processing. The possible future expansions are also discussed in light of the provided research.

The primary purpose of this paper was to expand the generic methodology and tools for adaptive clustering methods to both real-time and batch procedures. Future clustering is comprehended as a collection of if–then rules, providing the most understandable manner of cluster visualization. This technique produces the possibility for smart solutions in business to learn the behavior of orders and resources "on the fly" and to improve the required business capacities in a real-time fashion.

The paper is organized as follows. Section 2 is devoted to a review of developed principles of self-organization of records and clusters and the basic adaptive clustering method. It also presents a practical approach to if-then rule generation. Section 3 is focused on multi-agent clustering applications in logistics. Section 4 expands the application to clustering semantic descriptors of documents for an insurance company. Section 5 examines the subsequent developments in the clustering of large and mixed numerical and non-numerical data. Section 5 provides a discussion and outlines of future works.

2. The Proposed Methods of Clustering and Rule Extraction Based on Multi-Agent Technology

In this section, we systemize and review our previously published articles [10–12].

2.1. The Problem of Adaptive Clustering

The clustering issue is well comprehended for known in advance data. A cluster is understood as a group of data elements with mutual features, for example, daily records of all customers purchasing bread and milk. The process of data clustering is considered as a process by which data elements are grouped into clusters according to a set of given criteria and decision-making rules. In our applications, it is a dynamic, adaptive process treating an unpredictably modified collection. The system can change the link between any cluster and any record, at any time, in response to new, entirely unpredictable events, or proactively, motivated by internal considerations and decision-making. The problem of adaptive clustering involves keeping the best possible allocation of records to clusters at any time.

2.2. Multi-Agent Technology as a Basis for Adaptive Clustering

Multi-agent technology is new information expertise suitable for solving complex problems in a distributed manner by "computations as interactions" [20].

Our original multi-agent technology for adaptive resource management was developed in 2000–2008 [21–23]. An agent is defined as an autonomous computer program capable of reacting to events, making decisions, and coordinating these decisions with other agents. Depending on design principles and application domains, agents can be represented as a simple state machine or more comprehensive program which can have objectives, generate plans and control their execution, consult from domain knowledge, make reasoning about tasks, compose meaningful messages, send them to other agents or humans, interpret received messages, etc. A Multi-Agent System (MAS) is a

system supporting the interaction of a large number of agents with the use of different protocols for agent negotiations, for example, widely used basic contract-net protocol (http://www.fipa.org/specs/fipa00029/SC00029H.html).

In developments of MAS for adaptive resource management, specific classes of agents and protocols are designed according to their interaction to allocate resources to demands, make planning and scheduling, monitor, and control their execution. For example, in demand-resource networks [22,23], an agent is assigned to each order (demand) and each resource (capacity), which form a virtual market with the ongoing matching of orders to resources. Within the virtual market, agents compete and cooperate to achieve the near to "optimal" plan or schedule by signing contacts and establishing links between demands and resources. However, as it has been learned from practice, "optimal" solutions are hardly achievable, because, in reality, many of the process participants often have competing interests.

Under such conditions, instead of finding one "global" optimum, the multi-agent solution takes into consideration the balance of interests (named "consensus") of all involved contributors. As it has been found, such "near to optimal," or at least satisfactory, allocation of resources to requirements is quite useful in practice in many resource management applications. Specifically, once several decision-makers are involved, a lot of their domain-specific conditions need to be taken into consideration.

In 2010–2020, we introduced new classes of software agents of orders, technology processes and tasks, workers and equipment, products, etc. [24]. The designed agents have their own satisfaction and bonus–penalty functions for decision-making and use modifications of contract–net protocols to compensate for step-back changes in case of collective decisions. It makes the self-organization process more sophisticated but provides more opportunities to form multi-criteria schedules of orders and resources. The concept of the "adaptive schedule" as a "competitive equilibrium" capable of renegotiating agreements in case of unpredictable events provides high adaptability of the real-time solutions.

In 2009–2010, theoretical results were published [25,26], where the virtual market was recognized as the new class of distrusted algorithms based on contract–net protocols. In this definition of the virtual market, function Ci is provided for any set of tasks (T, C_i (T)). Each agent starts with some initial set of tasks, but in general, this assignment is not optimal in the sense of the total rate. The agents then enter into a negotiation process that improves the allocation of demands to resources and, hopefully, concludes in an optimal appointment with minimal total. The negotiation consists of agents repeatedly contracting out agreements among themselves, with each contract involving the exchange of demands and resources.

It is proven that for some responsibilities, the linear programming problem admits such an "auction-like" procedure with tight guarantees—for example, the case of weighted matching in a bipartite graph, known as the matching assignment problem. The task of scheduling is NP-complete, and it is not surprising that the auction-like procedure does not come with such guarantees yet.

Through the researches, many new interesting properties of methods and algorithms, based on principles of self-organization and the virtual market concept, have been recognized as intuitive, provably correct, naturally parallelizable, and appropriate for deployment in distributed systems. They tend to be adaptive and robust to perturbations of the problem specification.

Practical applications include multi-agent solutions for the adaptive scheduling of cargo flow for the International Space Station, assembly of Irkut MC-21 aircraft, high-speed and other trains for the Russian Railways, Gazpromneft supply chains for Yamal delivery, mobile teams of technicians for the 004 gas company, and others [21]. The result is up to 15–40% of the efficiency growth in resources because of the adaptive rescheduling of self-organized orders and resources in real time.

2.3. Multi-Agent Method of Clustering Based on Self-Organization of Records and Clusters

The main principle of the developed method of adaptive data clustering is to "allow" software agents to decide about jointing an item to groups. An agent is assigned to each data record element,

called a data agent, and to each cluster, called a cluster agent, forming a virtual market of a clustering system. Data agents and cluster agents negotiate according to the "optimal," or at least satisfactory, allocation of data elements consistent with the given criteria and the inbuilt decision-making rules.

The approach for adaptive clustering solves the task of allocating data elements arriving at the system individually in the "record-by-record" fashion in small batches.

The main system components are:

- Event queue (representing a sequence of data entering the system);
- Database storing the historical information;
- Multi-agent engine for adaptive clustering composed of two agent classes endowed by a communication auction-like protocols consisting of:
    - Data Agents;
    - Clusters Agents;
- User interface components.

Initially, the system is empty. At the arranging of a datum, the multi-agent engine creates a data agent intended to manage the datum behavior connected to its assignment to one of the clusters. This agent scans the current dataset and tries to turn out the groups to which the new element can be apportioned, aiming to improve the general system configuration. At the starting stage, an agent remains in standby mode until the arrival of the second data record. Thus:

1. At the starting moment, only a multi-agent engine exists in the empty system.
2. An agent is created by the multi-agent engine and assigned to a new data element appearing in the system.
3. This agent considers the available clusters enabled to include the new data element and sends a request for membership to the appropriate groups.
4. The attended clusters evaluate the applicant through their inner criteria, resting upon the candidate features aiming to increase their overall value.
5. The new datum agent accepts the most appropriate proposal and joins the datum to the corresponding cluster.
6. If no reply from a suitable cluster is received, then the datum agent attempts to form a new cluster with other data elements by sending cluster creation proposals to other agents, which accept the offer only if it increases the overall system value. In this case, the appropriate agents reorganize the whole system. So, the previously established relationships between the released data elements and their clusters are incrementally destroyed, and new connections between data elements are created targeting to increase the overall system value. ("self-organization" of records and clusters).
7. The process continues until all data elements belong to clusters, and no change in the cluster memberships can increase the system value, or until the number of iterations exceeds the given threshold.

The clusters and the datum agents work in parallel and asynchronously, continuously until none of them can further improve their target function. The datum agents interact mainly with their nearest neighbors. The cluster structure obtained during this self-organization process is presented as a provisional result and may be improved in the light of the new information. A cluster valuation criteria specifying the quality of the clusters may differ following the problem domain and specific tasks of the customer. As an example, we can recall cluster "density" as such a criterion. In the simplest case of one-parametrical data clustering (for example, the age interval of customers in the supermarket), the quality value could be, for example, the cluster density (fraction of the data lying in the space unit). Therefore, an agent of a cluster allows data records to join the group if only this operation increases the "density."

The proposed method can be considered as the basic outline and can be extended or modified to fit the expectations of a particular client and to cover a wide range of different types of tasks. The main conclusion is that the developed approach is comprehended to form a new methodology and framework of a wide range of methods and tools of adaptive clustering. The central idea here is not to search for a current clustering configuration within all possible combinations but to find dense record groups in a bottom-up way through self-organization of data and clusters. The resulting self-organized multi-level structure is also a dynamic equilibrium or a balance of interests of groups and records adaptively reconsidered according to the current online situation.

2.4. Examples of the Self-Organization Process

In the proposed method, "self-organization" means the capability of the MAS to modify autonomously existing and/or establish new relationships. In other words, the aim is to increase a given value or recover from a disturbance, such as the unexpected adding or omitting of an element. A change of a link between an item and a cluster is considered as a step in the self-organization process. Within the developed method, datum agents and clusters may use different criteria and different decision-making rules, even in the same clustering process. For example, for some managers, the "density" is more critical, representing new groups of purchasers with very similar performance. However, some others may look for the most prominent groups with the given properties, fast-growing groups, or groups with specific dynamics willing to transform their behavior.

Two examples of adaptive data clustering are exposed below, illustrating how different cluster valuation formulae lead to different cluster configurations [12]. Let us assume that in two-dimensional space (X and Y), four records come one by one (Figure 1). These records (R) have the following X and Y coordinates: R1 (2,4), R2 (3,3), R3 (6,3), and R4 (7,3). The cluster valuation formula is based on the "density" of clusters – the number of records associated with the square X-Y covered by points of a cluster. The negotiation rule is "first consider the nearest data element or cluster with the biggest increase of density."

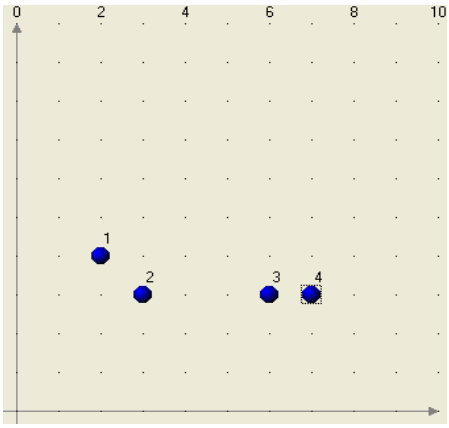
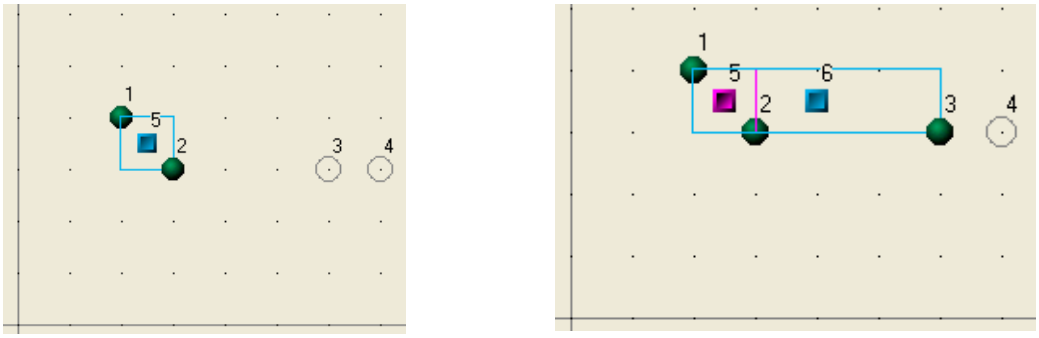


Figure 1. Dataset for clustering.

The adaptive clustering process is as follows:

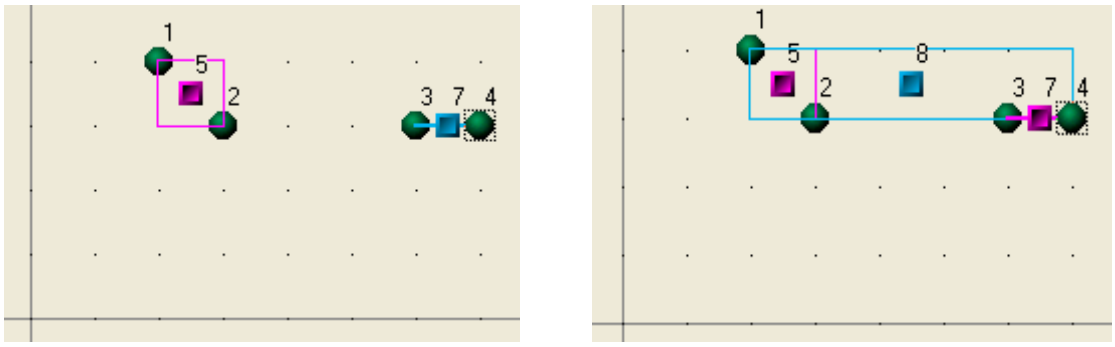
1. R1 arrives at the system.
2. R2 arrives and forms a new cluster with R1. The cluster is marked as 5 (Figure 2a).
3. R3 arrives and applies to Cluster 5 but it is rejected because this decision will reduce the cluster density. R3 then suggests to Cluster 5 to form a new cluster, which would include R3 and Cluster 5. As a result of the new agreement, Cluster 6 is formed (Figure 2b).
4. R4 arrives and suggests to R3 to leave Cluster 6 and join R4 in a new cluster. R3 agrees because it helps the record to improve its state, and the new cluster will have a bigger density than Cluster 6. Cluster 6 is destroyed, and a new Cluster 7 is formed from R3 and R4 (Figure 3a).

5. Cluster 7 then proposes to Cluster 5 to form a new Cluster 8 (Figure 3b).
6. Cluster 8 realizes that there are no more opportunities for clustering because all records and clusters have achieved their preferred memberships, and the clustering process terminates.



(**a**) (**b**)

Figure 2. (**a**) Formation of Cluster 5; (**b**) Formation of Cluster 6



(**a**) (b)

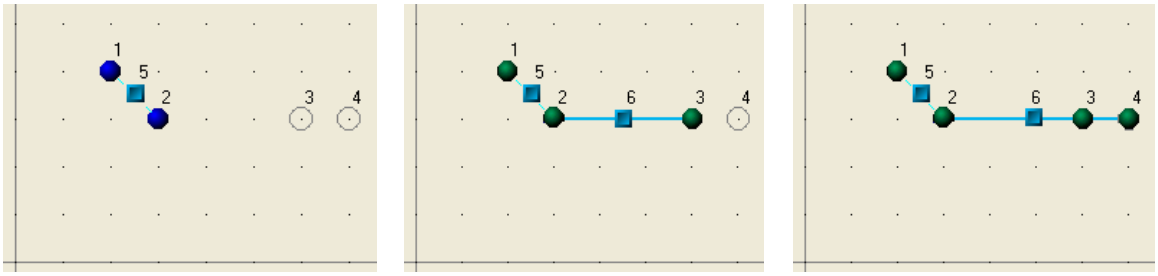
Figure 3. (**a**) Formation of Cluster 7; (**b**) Formation of Cluster 8

As a procedure outcome, a "competitive equilibrium" is attained once none of the agents can no longer improve their results. However, cluster and record agents continue working, and each of these agents wait for new changes. In such an event occurs through a new record arriving, then the process changing the previously completed decisions starts. It makes it possible to achieve a fast, flexible, and efficient response to the new data appearance without any additional computing.

Now, let us suppose that we have the same initial situation (Figure 1), but the decision-making process is different. Let us use the cluster estimation formula based on the shape of the cluster rather than its density—for example, it could be a horizontal line (several records with the same value of the attribute X). Then, if the cluster quality formula favors straight lines, the more records fall onto the same line, and the value of the cluster associated with this line is the greater one.

The steps of adaptive clustering will be as follows:

1. R1 arrives and waits for new opportunities.
2. R2 arrives. R1 and R2 form a new line cluster, Cluster 5 (Figure 4a),
3. R3 arrives. It suggests to R2 to form a new cluster since both records are on a straight line. R2 agrees to join R3 and form Cluster 6 but it also stays in Cluster 5 (Figure 4b).
4. R4 arrives and applies to join Cluster 6. It is accepted because membership of R4 increases the value of Cluster 6, raising the number of points on a straight line. Cluster 6 changes its boundaries and now incorporates records 2, 3, and 4 (Figure 4c).



(**a**) (**b**) (**c**)

Figure 4. (**a**) Formation of line Cluster 5; (**b**) Formation of Cluster 6; (**c**) New Cluster 6

As illustrated by these examples, even in simple clustering processes, different cluster validation formulae lead to drastically different results. The self-organization process of the adaptive clustering search always starts with the nearest points of neighbors and extends gradually. When record discoveries a proper cluster, it makes an offer and waits for a reply. The group reconsiders its locality, calculates its options, and either accepts or rejects the offer. Thus, instead of a centralized entirely "optimal" global decision, the proposed approach provides greedy solutions taken at the lowest level in a bottom-up way.

As a result, all decisions are made by clusters and records on their own but are based on some current local balance of interests of a particular record and cluster. If each of the sides agrees, the record enters the collection. If not, the record and cluster search for other variants. All this provides different variants of making collective and coordinated decisions while focusing on the dependency between the quality and computational cost/time effectiveness.

Examples demonstrate that the clustering process is a procedure of self-organization of clusters and records that creates various structures at the "micro" level. Still, the produced systems, in turn, also start to participate in the clustering process at the "macro" level. The route stops when the whole structure of clusters reaches competitive equilibrium—no record or group can improve its value. The outcome of the process generates multi-level structures of clusters, which are easily adjusted in real-time new records (events).

The resulting clusters can be transformed into the form of If–Then decision-making rules, which are clearly understood by domain experts and managers [10].

3. Application of Adaptive Clustering for Transport Logistics

3.1. Cargo Transportation Logistics

The problem concerns developing a near-to-optimal schedule to allocate transportation orders to available resources. The objective of a smart solution for real-time scheduling is to analyze customer orders, assign resources, form the plan using the company's own and third-party fleets, optimize the schedule, and monitoring [13]. In a real-time transport logistics task, the problem of adaptive rescheduling of orders by resources is solved. Generally, the mission is close to the known traveling salesman's problem. Actually, it includes not only the path minimization, but also many other restrictions, such as the level of service for the customer, the desired windowing time for receiving goods, the order of loading goods, the need to return empty containers, etc.

It is often very difficult to understand which schedules are "good" and which are "bad," even with the involvement of domain experts complicating the process of forming requirements for intelligent resource management systems. It is possible to approach this task in a completely different way and look at the historical aspect of the records, past trips, and resources. So, a strategy can be evaluated based on this historical data to restore planning knowledge.

However, the problem is complicated for large fleets. It is not trouble-free to ensure that the resulting schedule is feasible, especially from the cost perspective. Other criteria, such as VIP clients, patterns of delivery, preferred carriers, trip shape, cost of plan, total mileage of all trucks, customer

satisfaction level for individual clients, satisfaction of drivers, etc., also have to be taken into account. There are even too many human-like heuristics, for example, considering future orders to optimize today's trips, such as the allocation of constrained orders/resources first, working from the most distant/close points, etc.

Future orders are altogether unknown in advance. Thus, scheduling is carried out in real time while focusing on many specific properties, such as source and destination points, time windows, weight, volume, type, clients' reputation, and other specific requirements for cargo delivery conditions. Information about orders and resources are usually accumulated in a table, where each row represents a unique entity, for example, a transportation order, including information on the source and destination locations, and other cargo-related information. Thus, a clustering problem deals with items in a heterogeneous multi-dimensional space composed of the records having different types of coordinates.

3.2. Adaptive Clustering for Discovering Rules and Validating Logic of Logistics Scheduling

The developed adaptive clustering solution for rules extracting in the logistics domain intends to monitor the flow of orders, scheduling results, constructing rules applied to improve the quality of the planning and the forecasting. The primary problem is to learn the rules of a suitable schedule for a customer being, for instance, an immense transportation company [13]. The customer is not able to establish the formal model components like metric criteria to estimate the proximity and similarity of schedules generated. In the considered example, the customer provides a dataset made of 920 orders planned manually by operators. An adaptive clustering procedure runs for extracting hidden rules from the dataset.

A table with information on orders was created such that each row represents an order:

- From—the source location of cargo transportation;
- To—the destination location of cargo transportation;
- Pallets—number of units to be transported;
- CarrierType—the name of the carrier company;
- Depot—location of the vehicle executing the transportation;
- VehicleType—the type of the vehicle;
- Orders—the total amount of orders transported in parallel on a vehicle.

As can be seen, the From, To, and Pallets axes belonged to the second category (we cannot control these fields), while the CarrierType, Depot, VehicleType, and Orders axes were situated in the first category. The developed solution of adaptive clustering found 218 rules. The spreading of their confidence level is presented in Figure 5.

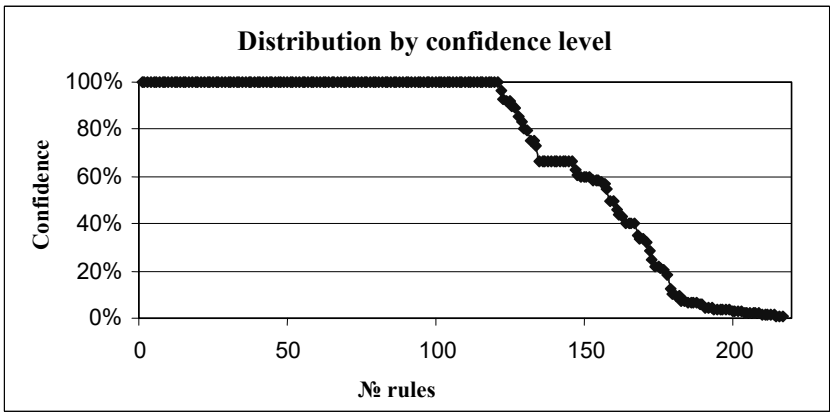


Figure 5. Association of system-detected rules with the confidence level.

Thus, more than 50% of rules had a confidence level of 100%, providing the value for decision making. Examples of system-found rules are shown in Figure 6, linking to the companies and geographical locations.

The company experts confirmed most of the rules and agreed that the revealed dependencies are persuasive for the problem domain. More interesting is the fact that the experts found that 8–12% of the rules had substantial business value. The discovered rules were incorporated into the knowledge base of the system for automatic scheduling, and innovative schedules were developed, such that the resulting schedules turned out very similar to ones created by experienced human dispatchers.

№	Pattern	%	cou
182	<To> = "Bryansk" -> <From> = "Moskow" ,<Carriertype> = "Bryansk TRANS" ,<Depot> = "Bryansk" ,<Pallets> = "26" ,<Vehicletype> = "ZIL"	100	37
110	<To> = "Samara" -> <Carriertype> = "Moskow" ,<Depot> = "Moskow" ,<Vehicletype> = "ZIL" ,<Orders> = "11"	100	33
52	<To> = "Archangelsk 1" -> <From> = "Volgograd" ,<Carriertype> = "TRANS GAZ" ,<Depot> = "Archangelsk" ,<Pallets> = "26" ,<Vehicletype> = "Gazel" ,<Orders> = "3"	100	30
111	<From> = "Archangelsk 3" ,<To> = "Kiev" -> <Carriertype> = "TRANS GAZ" ,<Depot> = "Kiev" ,<Pallets> = "26" ,<Vehicletype> = "Kamaz" ,<Orders> = "3"	100	30
8	<To> = "Bryansk" ,<Pallets> = "26" -> <From> = "Kiev" ,<Carriertype> = "TRANS GAZ" ,<Depot> = "Kiev"	100	28
101	<From> = "Samara" ,<To> = "Volgograd" -> <Carriertype> = "TRANS GAZ" ,<Depot> = "Volgograd" ,<Vehicletype> = "Gazel" ,<Orders> = "11"	100	28
132	<To> = "Bryansk" ,<Pallets> in [1.0 .. 3.0] -> <Carriertype> = "Bryansk TRANS" ,<Depot> = "Bryansk" ,<Vehicletype> = "Kamaz" ,<Orders> = "11"	100	26
143	<From> = "Krasnoyarsk" ,<To> = "Moskow" ,<Pallets> in [1.0 .. 3.0] -> <Carriertype> = "Bryansk TRANS" ,<Depot> = "Bryansk" ,<Vehicletype> = "ZIL" ,<Orders> = "11"	100	24

Figure 6. Examples of the discovered rules.

As a result of such a "learning from experience," the developed solution became much faster and provided several benefits for the customer:

- Manual rework needed is decreased by 32%;
- Trip quality is increased by 17%;
- Gaps presence in trips is decreased by 11%;
- Errors in product distribution plans decreased by 11%;
- Fleet mileage is decreased by 16%;
- Fleet usage is decreased by 8%.

Overall, the system brought approximately 20% of the increase in schedule quality. The time required to learn and customize domain-specific logic of operational scheduling in a new custom domain decreased from 1–2 months to 10–15 days.

3.3. Adaptive Clustering for Consolidation of Orders in Transport Logistics

The second problem is to find possible options for order consolidation aiming to improve efficiency [14]. The clustering method was applied here to find groups of orders similar in the geographical location and time windows. It can be supplied simultaneously by one truck with a specific capacity. The consolidation of orders requires the use of journey time metrics (JTM) and analysis of nearness of source locations by geography and by distance/time (JTM), as well as destination locations in combination with the overlap of time intervals if one truck takes all consolidated orders.

Such orders can be delivered in different ways: All orders in the cluster could be shipped by one truck, or by several trucks similar or not to each other. Particular heuristics were developed to expose the options for decision-makers. The resulting clusters discovered some interesting hidden rules:

- 90% of orders found consolidations with at least one order;
- 423 consolidations (21–27 pallets), improving the efficiency of transportation, were found.

The obtained results are given in Figure 7.

	Orders			
Pallets	Natural Consolidation	Inbound	Outbound	Pattern Seeker
26	715	736	1202	1250
25	61	52	96	79
24	35	63	71	77
23	83	77	48	44
22	66	66	71	67
21	74	57	60	50
20	60	51	24	30
Total >= 20	1094	1102	1572	1597

Figure 7. Results of cargo consolidations.

In general, the solution is 1.5-times more consolidations, improving the schedule quality by 15%.

4. Combining Adaptive Clustering with Text Understanding in Insurance Contracts Analysis

4.1. The Problem of Insurance Contracts Analysis

One of the most complex and exciting applications of adaptive clustering by the self-organization of clusters and records has been developed for the insurance business [15].

The problem is to cluster a vast number of documents, including various modifications of basic car insurance and some other contracts with many individual adaptations, for clients. However, the problem is even more complicated, because it is also necessary to consider associations with contract emails, business letters, licenses, manuals, financial, technical reports, etc. The objective of document clustering is to systematize, generalize, and optimize the contracts, and manage these documents in a standard and controllable manner using typical templates and the automation of business processes.

It has become evident that the customer needs to process too many documents manually, but all existing tools are not applicable for more in-depth analysis of texts, a semantic search of records with the view on context (not keywords), comparison of documents, a grouping of documents with similar meanings and, finally, automatic document generation with the use of templates.

One of the biggest insurance companies in the UK struggled with the problem of car insurance premium contracts. Such contracts take into consideration the client's gender and age, education level, yearly income, class of car, driving history, etc. Lawyers of the company produced more than 25,000 documents over the last 20 years. The given task was to analyze the dataset of these documents, classify them semantically, and create a contract template for each document's clusters. Contract templates were expected to include the most frequent clauses from various available contracts and to be used as the primary contracts for all new clients. Another part of the task was to process and classify the contracts from competitors which were recognized as the most popular or as the best practice for the insurance industry.

Roughly speaking, there are around 100 main groups of contracts that could take about 16 person-years of very intensive manual work of top-level experts in the insurance business. There are some known clustering systems, methods, and algorithms for clustering documents, such as LSA [27], Scatter/Gather [28], STC [29], etc.

The key constraint of these methods is the use presumptive information, which requires some primary document in a cluster, or the pre-analysis of documents by experts, providing batch clustering, limiting the number of clusters, or putting some other restrictions to reduce decision-making space. Such deterministic top-down conditions and limitations produce computational complexity, inadequate results, big noise, and irrelevant results. However, one of the most problematic discovered issues was that existing methods and tools use keywords instead of semantic networks of concepts and relations and semantic search in documents.

The task was formulated to develop a solution capable of combining the adaptive text understanding, adaptive clustering, and adaptive template creation to provide an opportunity to process new documents one by one in the real-time mode.

4.2. The Problem Solution

The proposed method, technology, and solution were developed with the use of ontologies and multi-agent technology. First, our previously developed ontology and multi-agent method for text understanding was used to represent the document meaning. Then, our multi-agent clustering method was applied to classify documents. Finally, a heuristic method was used to create the templates based on the groups of semantically similar documents.

4.2.1. Designing the Ontology of Problem Domain

As the first step, we created an ontology of the problem domain of the car insurance formalized as a semantic network of classes of concepts and relations with an innovative constructor [16]. The developed ontology included more than 400 types of objects ("contract," "document," "email," "agreement," "client," etc.), and 37 classes of relations ("requires," "whole-part," "have," "between," "part of contract," "belongs to," "guarantee," etc.). In average, each object has about six attributes ("amount of payment," "class of car," "car parameters," etc.).

Domain ontology makes it possible to create a semantic descriptor for each document, which specifies the content as a set of facts represented by instances of objects and relations. So, it becomes possible to evolve a clustering method in a knowledge-based, rather than data-driven, manner, and grant self-organization of the sematic document descriptors.

4.2.2. Semantic Analysis of Insurance Contracts

To solve the mentioned above problem, we applied a multi-agent text considerate solution, which was also developed based on the ontologies and multi-agent technology [17]. Such a solution was employed for the automatic creation of a semantic descriptor to each document (Figure 8).

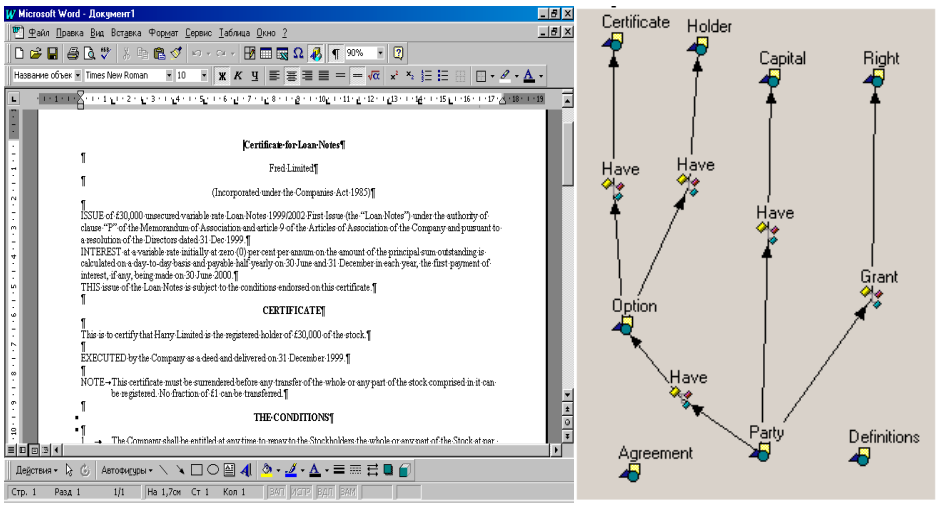


Figure 8. An example of a semantic descriptor for an insurance contract.

In some sense, ontology plays the role of a dictionary in the problem domain and multi-agent text. Subsequently, the understanding solution is the document's annotator in the dictionary terms. Words missing in the ontology dictionary are merely ignored. Still, the experiments show that recognition of about 20% of terms can help capture and identify about 80% of the meaning, which is associated with the most relevant and essential information contained in the documents.

The fragment of the semantic descriptor (Figure 6) outlines, for example, a certificate for options, belonging to a party of an insurance contract, and options fitting to the registered holder. The specification of multi-agent text understanding method and solution has been described by the authors of [18].

4.2.3. Adaptive Clustering of Documents

The developed multi-agent solution is able to quickly build the semantic descriptor of a contract. It is also possible to adjust the semantic descriptor manually if required. The resulting semantic descriptors receive the input data intended for an adaptive clustering advanced for managing not only records, but also semantic descriptors. The similarity between sematic descriptors is analyzed by a special heuristic method, which sequentially recognizes the most frequently used concepts and relations. The structure of clusters is created where each contact belongs to a cluster formed by a group of similar semantic descriptors.

An example of the clustering structure of insurance documents is shown in Figure 9.

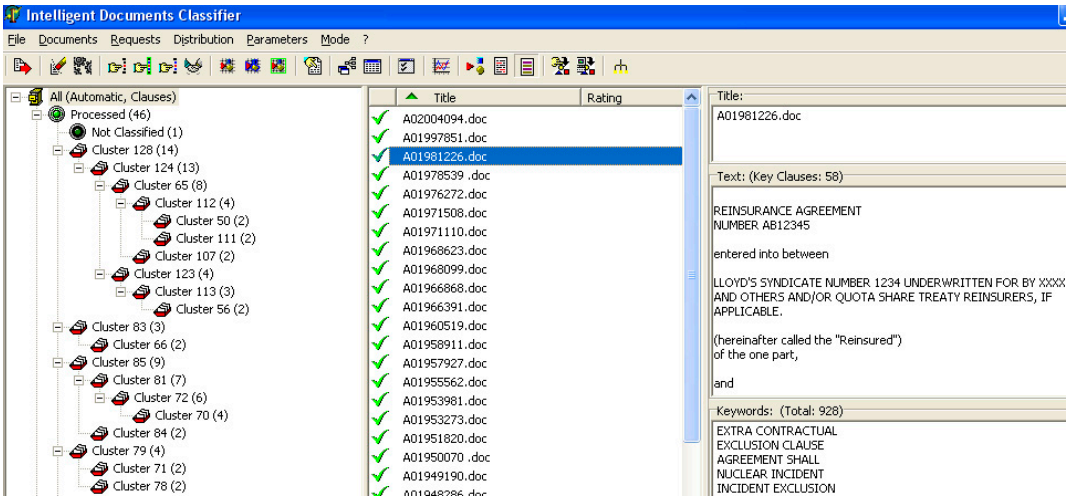


Figure 9. Clusters of documents for car insurance.

Modification of the multi-agent clustering engine is not very complex and time-consuming because the matching rules for semantic descriptors are mostly redesigned. From the standpoint of the domain experts, each cluster represents a record with several key clauses from relevant documents, which describe the similarity of these documents. The procedure of comparing sections takes into account the number of similar words and their relations. Clauses with a high degree of similarity are considered to be identical. The most frequent clauses among documents in a set are labeled as the key clauses.

4.2.4. Creating Templates of New Generic Contracts

Most popular clusters with similar clauses recognized by the adaptive clustering solution could be considered as the preferred candidates to templates of the most generic contracts. Additionally, one of the most useful features of the developed method is to find not only similar clauses but also anomalies in each group of clusters. All the key clauses, which are popular and unique/abnormal, are joined together to form the final templates of contracts. To determine the order in which they should appear in the resulting document, a heuristic algorithm was developed (Figure 10).

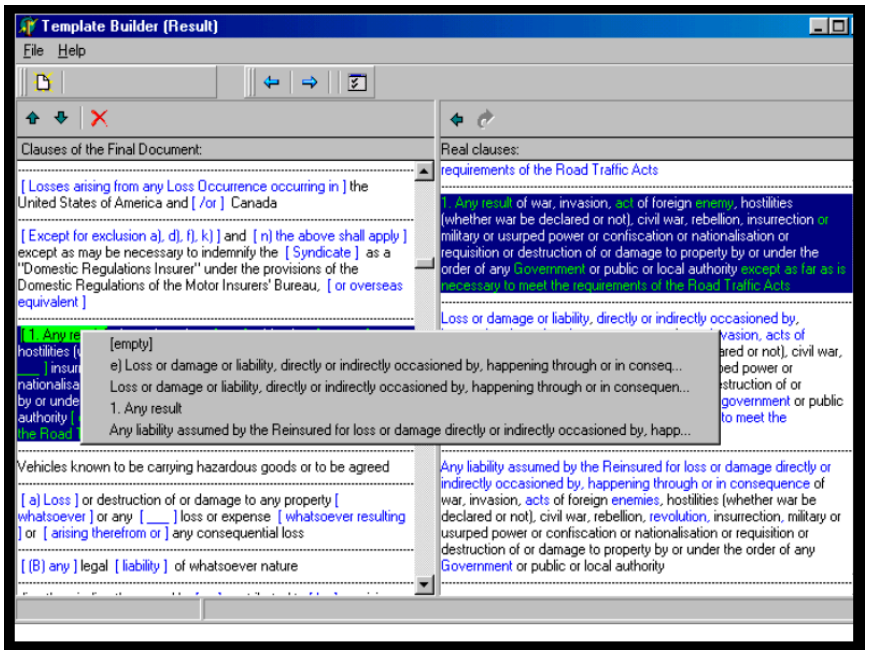


Figure 10. Formation of a template based on the key clauses in a cluster of contracts.

At the final stage, the selected clauses were manually analyzed by the domain experts able to readjust the order of clauses, select options, or edit the clauses or include additional specific provisions.

4.3. The Results

The process of validation of the adaptive clustering solution was organized with the use of the manually selected documents, categorized in several groups formed by the semantic similarity. Domain experts evaluated the detected clusters as "very similar," quite similar, more or less similar," "not similar." The results given in Table 1 show the overall number of groups rather than only high-level groups, as well as also the high validity level of automatically generated clusters. Let us state several interesting conclusions:

1. Experts generally preferred to simplify the document structure.
2. The number of hierarchical levels and number of groups manually produced was considerably lower.
3. Usually, all clusters specified by experts on top levels of hierarchy were actually the same as ones produced automatically.
4. Domain experts preferred to include a document into only one cluster similar to the document, while the system preferred to assign such documents to all similar groups.
5. Humans had difficulties with accurately identifying and remembering similarities.
6. Several related documents were missed by experts (approximately 35%), and some documents were wrongly assigned (11%).
7. The number of virtual clusters missed by the developed solution but recognized by domain experts was relatively low (about 7%).

To summarize the results, we can note that the developed system, working in an automatic mode, provides about 90% reasonable clusters with the percentage of accepted templates produced by the system was about 80%.

At the next stage, this multi-agent clustering solution was deployed for industrial applications. The processing of approximately 25,000 car insurance contracts (each consisting of 30 pages) can take about 16 person-years. It was done by the system in about 4 person-months, saving time and money for the client.

Table 1. Validation of clustering results: Experts vs. system results.

# of Docs	# of Groups(Auto/Manual)	Max Hierarchy Level (Auto/Manual)	Average Number of Docs in a Group (Auto/Manual)	Same and Similar Groups (%)	Template Validity Level (%)
11	7/9	4/3	2/2	94%	96%
43	23/14	5/2	4/3	91%	90%
125	54/28	9/3	6/4	87%	81%
864	279/43	13/3	5/14	88%	79%

The developed methods could be combined with new tools of social network clustering [30–32]. The new idea consists of considering similarity relationships with semantic and topological ones. It aims to cope with user messages content, pointing to the real-time discovery of communities with specific interest groups and individual users.

5. Adaptive Clustering for Telecom Companies

The telecom companies handle millions of records with hundreds of attributes of their users and targeting to extract knowledge about their clients' behavior. Aiming to solve such a problem using a clustering method based on our multi-agent technology, we meet the problem of processing a mix of numeric and text data. Many approaches are available for numeric data (ROCK, DBSCAN, BIRTH, CP, CURE, etc.), but all of them are applicable for symbolic text data and their combination.

The developed version of multi-agent clustering for solving the problem has been described by the authors of [19] as such. The developed multi-agent engine process the datasets with symbolic and numerical attributes. The developed method is capable of processing records having mixed characteristics and also of generating rule If-Then-Else, pointed to interpret the clustering results.

In this application, the primary method was adjusted to form clusters characterized by their centroids. Thus, in the case of numeric data, it links to centers of gravity, but for symbolic data, it considers the most representative attributes. In general, the similarity of records is defined here as such [33]:

$$Sim(C,\ t) = \sum_{i=1}^{m} \frac{Sup(a_i)}{\sum_j Sup\left(a_j\right)} \tag{1}$$

where *C*—a cluster (given by a centroid), *t*—a record, *a*—value of an attribute, *m*—dimension of the record, and the sum is related to all the records of the given cluster for each attribute. The Support level (Sup) of each attribute equals to the number of records in the group. Features with immense Support mean a bigger probability of getting these values. A distance in the decision-making space connected to this similarity is specified as:

$$\tilde{d}(C,t) = 1 - SimN(C,t), \tag{2}$$

where *SimN*(*C*,*t*)—normalized relevance. It was proven by the authors of [34] that objective function in the space of mixed attributes allows separating variables. Using normalized values for both types of attributes, it is possible to process data in the same way.

The experimental testing of results is provided with the use of benchmarks presented in UCI Repository of Machine Learning Databases http://www.ics.uci.edu/~mlearn/MLRepository.html. The results are quite similar, but the developed methods give the possibility to process data in real-time in an adaptive style.

6. Discussion and Future Steps

The main paper contribution is a comparative review and generalization of the previously developed approaches of adaptive-based multi-agent technology clustering. This novel methodology

is not yet systemized and has not been sufficiently discussed in the literature. However, it can be integrated with existing approaches aiming to provide significant value under uncertain conditions.

The developed methods are implemented and adapted for distinguishing clustering tasks. Even though the presented techniques can be considered as the initial ones, the practical results encourage these methods for the following reasons:

- Complex and various clustering problems can be treated from a general unified standpoint;
- The process can be comprehended in an adaptive manner when changes affect just the necessary delicate objects, making it possible to work in a real-time fashion;
- There is a fast, flexible, and efficient possibility to respond to unexpected events;
- There is a possibility, using ontologies, to take into account the individual characteristics of the subject area and business specifics of each specific organization;
- The methods reduce the programming complexity and timing together with the total cost.

Multi-agent technology provides various recently developed methods for adaptive clustering by data their self-organization to expand the current technologies, which can be considered as a new generic methodology. The stated results designate the following research directions for future research in the efficiency of adaptive clustering using multi-agent technology:

1. Application of ontologies for capturing the semantics of the problem domain

The clustering process significantly depends on domain-specific knowledge and decision-making rules. One of the very promising new approaches for capturing domain semantics is an ontology in the form of semantic networks of concepts and relations, which can help improve clustering with the use of specific domain knowledge, e.g., in manufacturing, transport, agriculture, etc. In this case, the generic multi-agent framework of clustering could be customized for particular customers by domain-specific ontologies.

2. Developing a virtual market

At the moment, agents use real money in decision-making. However, virtual money can be introduced as a way to regulate the abilities of clusters and records to make decisions and solve conflicts. For example, in transportation logistics, the sum of money available for a record can be set as a price for the order, which the customer is due to pay. Then, the history of VIP order is "richer" than a record of a single order from a regular customer. It makes it possible to form clusters with distant records, taking into account a growing context, and generating more groups as a result.

In the case of an advanced model of virtual markets, clusters and records pay virtual taxes to act in the system and stay in groups. This affects their financial resources, making some clusters and records evolutionally disappear from the system, thus decreasing the load on the network, while some of them grow and become more assertive.

A different model of microeconomics can also be applied. For example, clusters can take charges from records for the right to enter the following "club system" model (the amount of payment does not depend on the situation, the productivity of the records, or the number of club members) or according to the "shareholder" model (the amount depends on the case). Thus, agents of clusters and records are supposed to think not only about "money," but about their "lifespan," balancing between the criteria. The foremost could stay the same. Many advanced economic models for better managing agent behavior could be applied in the framework of the same multi-agent system.

3. Guided self-organization on the virtual market of clusters and records

The self-organization process can be stopped by local attractors, which is hard for agents to avoid. One of the solutions consists of the appointment of an agent of the "whole" system. Its mission is to monitor the situation in clustering, recover the potential local optimums, and to cascade changes during different types of interventions, for example, investing virtual money into some promising clusters or records.

4. Self-learning

Agents of records and clusters may use different self-learning tools, including neuro-networks or deep learning ones, to alter their decision-making rules and analyze the results.

5. Multi-object optimization for the best possible use of virtual money

As the next step in method development, we consider a situation when one record (cluster) can locate in a few groups simultaneously, paying for a membership or some virtual taxes, etc. In this case, the collected virtual money of each agent needs to be assigned by other agents of clusters and records in the best possible way on the virtual market, aiming to decide which candidates to choose.

The results of the research could be useful for designing a new generation of smart software products based on multi-agent technology that is able to adapt, learn, and evolve over their life cycle.

Author Contributions: Conceptualization: S.G.; methodology: P.S.; Software development: I.M.; Validation: E.S.; Supervision: P.S. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by the Ministry of Education and Science of the Russian Federation within contract agreement No. 075-15-2019-1691—unique ID number RFMEFI60419 × 0224.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. *Pattern Recognition*, 3rd ed.; Theodoridis, S.; Koutroumbas, K., Eds.; Academic Press: Cambridge, MA, USA, 2010; p. 984.
2. Everit, B. *Cluster Analysis*, 5th ed.; Wiley: Chichester, UK, 2011; p. 346.
3. Larose, D.T.; Larose, C.D. *Discovering Knowledge in Data*, 2nd ed.; Wiley: Hoboken, NJ, USA, 2014; p. 336.
4. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; The MIT Press: London, UK, 2018; p. 525.
5. *Contemporary Experimental Design, Multivariate Analysis and Data Mining*, 1st ed.; Fan, J., Pan, J., Eds.; Springer: Basel, Switzerland, 2020; p. 386. [CrossRef]
6. Shirkhorshidi, A.S.; Aghabozorgi, S.; Ying, T.; Herawan, W. Big Data Clustering: A Review. In *ICCSA 2014: Computational Science and Its Applications—ICCSA 2014*; Springer: Basel, Switzerland, 2014; pp. 707–720.
7. Mirkin, B. *Clustering for Data Mining a Data Recovery Approach*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2016; p. 350.
8. Volkovich, Z.; Toledano-Kitai, D.; Weber, G.-W. Self-Learning k-eans Clustering: A Global Optimization Approach. *J. Glob. Optim.* **2013**, *56*, 219–232. [CrossRef]
9. Taha, K. Static and Dynamic Community Detection Methods That Optimize a Specific Objective Function: A Survey and Experimental Evaluation. *IEEE Access* **2020**, *8*, 98330–98358. [CrossRef]
10. Minakov, I.; Rzevski, G.; Skobelev, P. Data Mining. Patent Reference No. GB 2 411 015 A, 17 August 2005.
11. Andreev, V.; Volhontsev, D.; Iwkushkin, K.; Karyagin, D.; Minakov, I.; Rzevski, G.; Skobelev, P. Multi-agent system of knowledge extraction. In Proceedings of the 3rd International Conference Complex Systems: Control and Modelling Problems, Samara, Russia, 4–9 September 2001; pp. 201–212. (In Russian)
12. Rzevski, G.; Skobelev, P.; Minakov, I.; Volman, S. Dynamic Pattern Discovery Using Multi-Agent Technology. In Proceedings of the 6th WSEAS International Conference on Telecommunications and Informatics (TELE_INFO '07), Dallas, TX, USA, 22–24 March 2007; pp. 75–81.
13. Himoff, J.; Rzevski, G.; Skobelev, P. Multi-Agent Logistics i-Scheduler for Road Transportation. In Proceedings of the AAMAS'06, Hakodate, Hokkaido, Japan, 8–12 May 2006; pp. 1514–1521.
14. Minakov, I.; Rzevski, G.; Skobelev, P.; Volman, S. Automatic extraction of business rules to improve quality in planning and consolidation in transport logistics based on multi-agent clustering. In Proceedings of the Autonomous Intelligent Systems: Multi-Agents and Data Mining 2007, LNAI 4476, St. Petersburg, Russia, 3–5 June 2007; pp. 124–137.
15. Minakov, I.; Rzevski, G.; Skobelev, P.; Volman, S. Creating Contract Templates for Car Insurance Using Multi-Agent Based Text Understanding and Clustering. In *International Conference on Industrial Applications of Holonic and Multi-Agent Systems*; LNAI 4569; Springer: Berlin/Heidelberg, Germany, 2007; pp. 361–370.

16. Andreev, V.; Iwkushkin, K.; Minakov, I.; Rzevski, G.; Skobelev, P. The Constructor of Ontologies for Multi-Agent Systems. In Proceedings of the 3rd International Conference Complex Systems: Control and Modelling Problems, Samara, Russia, 4–9 September 2001; pp. 480–488. (In Russian)
17. Andreev, V.; Iwkushkin, K.; Karyagin, D.; Minakov, I.; Rzevski, G.; Skobelev, P.; Tomin, M. Development of the Multi-Agent System for Text Understanding. In Proceedings of the 3rd International Conference Complex Systems: Control and Modelling Problems, Samara, Russia, 4–9 September 2001; pp. 489–495. (In Russian)
18. Minakov, I.; Rzevski, G.; Skobelev, P. Automated Text. Analysis. Patent Application No. 305,634, 20 May 2004.
19. Vittikh, V.A.; Mayorov, I.V.; Skobelev, P.O.; Surnin, O.L. Data Mining with Clustering. In Proceedings of the 8nd International Conference Complex Systems: Control and Modelling Problems, Samara, Russia, 24–28 June 2006; pp. 460–466. (In Russian)
20. Jennings, N.R.; Sycara, K.; Woolridge, M. A roadmap of agent research and development. *Auton. Agents Multi Agent Syst. J.* **1998**, *1*, 7–38. [CrossRef]
21. Rzevski, G.; Skobelev, P. *Managing Complexity*, 1st ed.; WIT Press: London, UK; Boston, MA, USA, 2014; p. 216.
22. Skobelev, P. Open multi-agent systems for decision making support. *Avtometria* **2002**, *36*, 45–61.
23. Skobelev, P.; Vittikh, V. Models of agents interaction in demand-resource networks. *Autom. Remote Control* **2003**, *64*, 162–169.
24. Skobelev, P. Multi-Agent Systems for Real Time Adaptive Resource Management. In *Industrial Agents: Emerging Applications of Software Agents in Industry*; Leitão, P., Karnouskos, S., Eds.; Elsevier: Amsterdam, The Netherlands, 2015; pp. 207–230.
25. Shoham, Y.; Leyton-Brown, K. *Multi-Agent Systems: Alghoritmic, Game Theoretic and Logical Foundations*; Cambridge University Press: Cambridge, UK, 2009. Available online: http://www.masfoundations.org (accessed on 20 August 2020).
26. Easley, D.; Kleinberg, J. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*; Cambridge University Press: Cambridge, UK, 2010. Available online: http://www.cs.cornell.edu/home/kleinber/networks-book/ (accessed on 15 August 2020).
27. Dumains, S.T.; Furnas, G.W.; Landauer, T.K. *Indexing by Latent Semantic Analysis*; Bell Communications Research 435 South St.: Morristown, NJ, USA; University of Western Ontario: London, UK, 1990.
28. Cutting, D.R.; Karger, D.R.; Pedersen, O.J.; Tukey, J.W. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In Proceedings of the 15th Ann Int'1 SIGIR '92, Copenhagen, Denmark, 21–24 June 1992; ACM: New York, NY, USA, 1992; pp. 318–329.
29. Zamir, O.E. A Phrase-Based Method for Grouping Search Engine Results. Ph.D. Thesis, Department of Science & Engineering, University of Washington, Seattle, WA, USA, 1999.
30. Agreste, S.; De Meo, P.; Fiumara, G.; Piccione, G.; Piccolo, S.; Rosaci, D.; Sarné, G.M.L.; Vasilakos, A.V. An Empirical Comparison of Algorithms to Find Communities in Directed Graphs and Their Application in Web Data Analytics. *IEEE Trans. Big Data* **2017**, *3*, 289–306. [CrossRef]
31. Jiang, H.; Sun, L.; Ran, J.; Yang, X. Community Detection Based on Individual Topics and Network Topology in Social Networks. *IEEE Access* **2020**, *8*, 124414–124423. [CrossRef]
32. Liu, Z.; Barahona, M. Graph-based data clustering via multiscale community detection. *Appl. Netw. Sci.* **2020**, *5*, 3. [CrossRef]
33. He, Z.; Xu, X.; Deng, S. Squeezer an Efficient Algorithm for Clustering Categorical Data. *J. Comput. Sci. Technol.* **2002**, *17*, 611–624. [CrossRef]
34. Huang, Z. Clustering large data sets with mixed numeric and categorical values. In Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining, Singapore, 23–24 February 1997; p. 14.