論文 / 著書情報
Article / Book Information

題目(和文)	予測型マルチエージェント学習分類子システムの設計と分析およびその推薦アルゴリズムへの適用
Title(English)	Design and Analysis of Predictive Multi-Agent Learning Classifier System and its Application to Recommender Algorithms
著者(和文)	ｲﾙﾊﾞﾝｴﾑﾊﾃﾞ
Author(English)	Mhd Irvan
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第9542号, 授与年月日:2014年3月26日, 学位の種別:課程博士, 審査員:寺野　隆雄,出口　弘,山本　学,高安　美佐子,小野　功
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第9542号, Conferred date:2014/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Category(English)	Doctoral Thesis
種別(和文)	要約
Type(English)	Outline

Design and Analysis of Predictive Multi-Agent Learning Classifier System and its Application to Recommender Algorithms

予測型マルチエージェント学習分類子システムの設計と分析
およびその推薦アルゴリズムへの適用

Doctoral Thesis by

Irvan Mhd

Supervised by
Professor Takao Terano
寺野隆雄　（教授）

Tokyo Institute of Technology

東京工業大学

Interdisciplinary Graduate School of Science and Engineering
Department of Computational Intelligence and Systems Science
総合理工学研究科・知能システム科学専攻

Tokyo, Japan
2014

Acknowledgement

First and foremost I would like to offer my deepest gratitude to my academic supervisor, Professor Takao Terano, who has been giving me his full support throughout my academic life in Tokyo Institute of Technology. His kind personality and valuable guidance have turned my research activities into very enjoyable moments. I highly appreciate his continuous dedication and this thesis would not have been finished without his support.

I also would like to express my sincere thanks to the review committee members of this thesis, Professor Hiroshi Deguchi, Professor Isao Ono, Professor Gaku Yamamoto, and Professor Misako Takayasu for their insightful comments and reviews throughout the making process of this thesis. They helped me understand aspects from different point of views that I might have missed.

I thank the Ministry of Education, Culture, Sports, Science and Technology of Japan for awarding me the Monbukagakusho Scholarship to pursue my Master and Doctoral degree in Japan. The scholarship helped me so much in covering my living expense and research needs.

In addition, I thank all the members and alumni of Professor Terano's lab who taught me the joy of research. They have been such tremendous help both inside and outside the laboratory, supporting me to get used the Japanese academic culture.

I would also like to take this opportunity to express my special thanks and my utmost appreciation to my family who have been giving me their encouragement and mental support. Having to live in a separate country far away from them, there were difficult times, but their warm encouragement kept me maintain my calm and always think positively.

Last but not least, I thank all of my friends who, directly and indirectly, have lent their helping hand in my venture. I would not have managed to come this far without their support.

Outline

1. Introduction

Machine Learning (ML) is an attractive field pursuing the idea of computers learning by themselves to solve problems. There are limits to the complexity of problems humans can address. We can evade these problems by building machines that generate desired behavior on their own (learning machines). Of course, designing learning machines which can solve complex problems is itself a very difficult task. Nevertheless, it has been found that ML promises to repay the enormous effort invested in them and to allow them to handle things which humans cannot. ML does not only address engineering problems. It lies on top of the foundation of many other fields, such as understanding intelligence, cognitive science, psychology, and others.

This thesis concerns about Multi-Agent Systems (MAS) and Learning Classifier Systems (LCS), new machine learning paradigms that are currently still under active research. MAS typically deals with distributing learning procedures across multiple intelligent agents. This allows agents to handle a very complicated problems by dividing it into sub-problems, and then each agent is responsible to handle the sub-problems assigned to them. Meanwhile LCS typically involves using Evolutionary Computation and Reinforcement Learning techniques to complex problems. It relies on an agent to evolve its knowledge by interacting with the environment which in turn, give rewards based on the impact made by the agent.

2. Background

Most LCSs were designed with a single agent in mind. As such, building a successful LCS to solve a very complex problem requires tremendous efforts. We will show in this thesis that, for complex problems which can be separated into sub-problems, it is possible to design a simple MAS, where each agent maintain its own LCS to solve those sub-problems. We believe, if the goals of Multi-Agent LCS were met, many of the goals from the area of machine learning would be accomplished.

3. Research Questions

This thesis arises in response to these questions:

- Can LCS works in a multi-agent environment?
- How do LCS agents work in organization to transfer knowledge?
- Can Multi-Agent LCS makes good prediction?
- Is LCS a feasible approach for Recommender System?

There are still very few works on multi-agent system using LCS and most of them concentrate on a high-risk-high-return approach. They are willing to take risk in making decision hoping for big reward, despite the questionable probability. In situations where accuracy errors can not be tolerated (such as recommender systems), this can be a big issue. However, calculating predictions for all possible solutions in a complex problem require a massive amount of computing time and resource.

4. Goals

One particular goal that we aim to address in machine learning is prediction. Humans always wonder how things could have gone differently had they predicted the current situation.

- If an insurance company could have better anticipated the risk of accepting a sport-loving person as a customer, it could have priced the premium accordingly.
- A financially-troubled company could've stayed in good shape should it able to foresee which marketing activities would pay off.
- An accurate spam filter could heavily increase work efficiency.
- Streaming-media companies put faith in recommendation systems to correctly predict their customer preference.

To put it simply, predictions can make or break a business. Prediction is power. However, prediction in machine learning is a very complex issue. This is especially true when the number of available data is massive. That is why we are proposing a multi-agent approach to distribute the computing process across many agents to balance the workload that the machines have to compute and learn.

5. Proposed Method

This thesis is about achieving precise and valuable computerized prediction. Simple goal, but challenging. The solution we propose is a distributed rule-based machine learning. By combining the power of MAS and LCS, we introduce a new learning paradigm to simplify the process to solve complex problems.

We propose a few new parameters for LCS to make it works in a multi-agent environment. Those parameters also indicate mechanism for the agents to communicate and transfer knowledge from individual level to organizational level.

We aim for a highly predictive machine learning technique. We will show in this thesis that we can achieve it by designing an accuracy-oriented Learning Classifier System. The Genetic Algorithm and Reinforcement Learning are used to evolve accurate rules and punish inaccurate ones.

To demonstrate the capabilities f our proposed method, we will show its application in recommender system RSs problems. RS is chosen because it is one of the most ubiquitous system on the various web services nowadays. It requires highly predictive capabilities so learn users' preference and it is very complex due to the large set of data. This makes it a very ideal test-bed for our proposed method.

6. XOLCS, a Predictive Multi-Agent Learning Classifier System

XOLCS is the name of the model proposed in this paper. It is a new type of multi-agent learning classifier system. It introduces the concept of organizational learning to XCS classifier system. The concept deals with the management of knowledge (the collection of classifiers) known by agents. By implementing organizational learning methods, the agents can save time (for example, the number of learning cycles) needed to achieve the ideal collection of classifiers that produce good results when applied to various situations in the environment. It allows the agents to distinguish classifiers that are better tested earlier or later in the process. It also provides a way for agents to share their high accuracy classifiers by allowing other agents to access and to learn those classifiers.

XOLCS is a learning classifier system that takes inspiration from XCS and organizational learning concepts specifically designed for multi-agent systems. XOLCS deals with the management of the knowledge inside agents’ classifier system, allowing new knowledge to be created and good

knowledge to be retained and transferred between agents. It is designed for modeling the behavior for cooperative agents, in contrast to competitive agents.

In XOLCS, each agent maintains its own classifier system. Each agent has a framework consisting: (1) knowledge, (2) organizational learning mechanisms, and (3) decision-making algorithm. At first, the agent perceives the environment. Based on what it sees, it checks inside its knowledge population, select an action, and receive feedback from the environment. Organizational learning may be applied to modify the knowledge population before choosing an action. After an action has been executed, genetic algorithm operations are applied with a defined probability to evolve the rules that propose the same action.

7. Organizational Learning

Our implementation of organizational learning algorithms introduces new parameters for classifier systems. Classifiers within the population maintain a boolean flag termed immunity (imm) that determines whether a classifier is immune against genetic algorithm operation. It also maintains extra parameter experience (exp) denoting how often the classifier was tested and selected into the action set by the XCS. Whenever XCS selects a classifier, the system increases the exp parameter of that particular classifier. The organizational learning algorithm uses the parameter as a reference for its functions to look for classifiers with certain level of experience (such as whether they are above or below the threshold level of "good" classifiers).

These new parameters introduced in our proposed method contribute in helping learning classifier systems to recognize which classifiers are more relevant than the others. It also act as indicators for LCS agents to wisely manage and distribute their classifiers. This ensures that only high quality classifiers are being shared to other agents.

At first, every new classifier is treated as individual knowledge. Over time, those classifiers will earn experience. Classifier with exp above the threshold of λ is considered experienced enough, and the agent is confident to share the classifier with other agents, thus turning the individual knowledge into a collective knowledge.

During learning mode, several organizational learning-inspired mechanisms are activated. Organization learning (OL) is triggered when the last occurrence of OL is greater than an OL's frequency parameter ϕ. A timestamp parameter h is associated to each classifier as the timestamp of

the last OL occurrence.

These mechanisms are used to modify the classifier population before they are fed into the decision-making algorithm described previously. The job of organizational learning algorithms is to mark the importance level of every classifier so that the learning algorithm can tell, for example, which classifier has higher priority to be evolved by genetic algorithms, as well as which classifier needs to be exchange during the current situation inside the environment.

By splitting the knowledge into experienced and non-experienced classifiers, the algorithm is better guided in differentiating good classifiers from bad classifiers. The probability of good classifiers getting better fitness is increased.

Also, by exchanging experienced knowledge between agents, it allows an agent not to learn from the beginning again when it encounters new situation that has been experienced by other agents.

Together with genetic algorithms, organizational learning algorithms help the system to learn online. The online learning process is composed into a sequence of trials. First the system senses the current situation, then it predict how the situation will change during the next trial. After the system makes a prediction, if GA or OL algorithms is triggered, their operations will be applied toward the classifiers that made the prediction before the system makes another prediction. The new classifiers' values produced by these algorithms are used as basis for subsequent decision-making process.

8. Recommender System with XOLCS

The recent popularity of smart devices provides recommender systems with more types of inputs than ever. Those inputs come from various sensors and applications of the devices, such as location sensor, device information, social networks connection, etc. More inputs mean more factors to consider for calculating recommendation. Furthermore, the stream of inputs coming constantly from those devices means that the data grows bigger very fast.

XOLCS can be used to fill the role of addressing these latest issues. We will show how Classifier System-based recommendation can tackle problems plaguing other recommendation methods.

To the best of our knowledge, there is no other work exist in applying LCS or XCS to recommendation problems. We hope our work can contribute to advancing both LCS and RS

research community. In our example, we implement our proposed model for movie recommender problem.

Smart Devices, such as SmartTVs and Smartphones, provide the integration of various web services into televisions and mobile phones. One of such services is on-demand streaming media. On-demand streaming allows users to choose shows or movies they would like to watch. However, there are countless choices available from the streaming providers. This may lead users to confusion, as they might not know what would be interesting to watch.

Recommender systems address this problem. Rather than waiting for users to choose a movie, the system recommends movies that it thinks they will like. The recommendations are usually generated through learning from users' past behavior, or from other similar users' interest.

Many current recommender algorithms rely on users' feedback, such as rating, or profile. For example, when a user liked a movie, the system will search for other users who liked the same movie, and then recommend other movies that those users also liked. While this might work very well for shopping websites, it might not be suitable for media streaming.

When a user enjoys a movie, there are many factors that affect her enjoyment at that moment. For example, a user who usually enjoys action movies on weekend might prefer watching drama movies on other days at night to relax after getting tired from work. A user might really like science fictions, but he only enjoys watching them from a large screen TV at home, and never on smartphones due to the small screen.

In other words, even if a user liked a movie, she might not be going to enjoy the same movie, had she watched it under different circumstances. Locations, devices, days, times, and other factors contribute to whether she will enjoy a movie or not. SmartTVs and smartphones can provide all these details, and it is only natural to use the information as basis for recommender systems. However, designing a recommender system to consider all these factors using typical recommender algorithms will end with a very complicated decision-making process. This thesis shows how XOLCS Classifier System (LCS), implementing genetic algorithm (GA) and reinforcement learning (RL) can be used to approach this problem.

9. Implementation

XOLCS maintains a population of classifiers that predicts the best action given its input. The input we use is information that is available from smart devices, such as sensory data, geographical and device information, as well as date and time. GA is used to search the possible solution space to figure out which part of the inputs, or what kind of input combination affects the viewing experience. Solutions proposed by the GA are evaluated by RL, giving feedback whether they are accurate or not. During training, XOLCS repeat this process over and over again until it has a good population set with high average accuracy.

XOLCS keeps a collection of classifiers, referred as population set. Each classifier is essentially a rule of condition-action set. The classifiers have a parameter that predicts the reward that the agent will receive, should it choose the action proposed by the relevant classifiers. XOLCS agent learns to perform the best action based on the condition. Whenever the agent performs an action, it receives feedback from the environment to inform the quality of the action.

The condition part of the classifiers is a string of input reflecting the situation that the agent encounters. In our recommender system, the input string consists of information that can be provided by smart devices: Day, time, user's age, gender, type of device, location, movie's release date, movie genres, movie stars, movie ID. From the training set, when a user "Like"d a movie, the system generates several classifiers that represent the situation. It takes into consideration the information mentioned before.

Suppose a user likes Titanic (a fact). The system considers the situation when it happened. On what day? What time? What kind of user? What kind of movie? Who are the actors? An example of a classifier could be read: "A female teenagers who live in Tokyo who likes 1990s movies AND likes drama and romantic movies AND likes movies starring Leonardo DiCaprio, Kate Winslet, and Billy Zane AND likes movies directed by James Cameron AND likes English movie WILL enjoy watching Titanic ON a tablet AT night ON Sunday".

Since movies are usually related to multiple genres and have many actors/actresses involved, for each fact the system generates multiple classifiers picking (in our example) random genres and random actors/actresses related to the movie. This means that during learning the system will look for which combination of genres and actors/actresses are relevant to the user's profile.

Additionally, for each fact the system also generates classifiers that have “Wildcard” symbols along the input string. This means that during learning the system will look for which parts of the input that are not relevant (the noise) to the user’s profile.

A classifier may think that the “Day”, “Time”, “Device”, “Tag”, “Actresses”, and “Director” parts are not relevant. In this case, the previously mentioned classifier could be generalized into: “Regardless of the Day, Time, and Device, a female teenager who lives in Tokyo who likes 1990s romantic movies AND likes movies starring Leonardo DiCaprio, WILL enjoy watching Titanic”.

The more wildcard symbols a classifier has, the more generalized it is. After generating all the classifiers from the initial set of facts, the system will test those classifiers against users in the testing set. During each learning loop, the system will evolve classifiers that make accurate prediction and delete inaccurate classifiers through evolutionary process. After the learning process is finished, the population set should consist the generalized classifiers (but not too generalized) with high accuracy.

The input length for XOLCS is flexible. If, for example, more profile data are available from the smart devices, it is possible to add those details into the input. If necessary, more metadata about the movies (such as more genres, studio name, music composer, or other metadata) can be added too. The longer the classifier, the more details are taken into consideration.

Obviously, when more details are necessary to be considered, using general decision process, such as decision tree, the decision-making process will end up with a very complicated procedure. XOLCS simplifies this process by representing the details as strings of possible solution and let the evolutionary process search for the good ones.

10. Learning Phase

Genetic Algorithm (GA) [1] is used to evolve the rules in [A]. GA is triggered when the average time period for classifiers within [A] since the last occurrence of GA is greater than GA’s frequency parameter. The GA starts by selecting two “parent” classifiers from [A] with roulette wheel selection.

Once two parents have been selected, new offspring classifiers are generated by crossover and mutation. Crossover operation selects a point on the parent classifier strings. All digits beyond that

point of one parent classifier are swapped with the digits of another classifier. Finally, the digits of the resulting strings from the crossover are mutated into different acceptable values. This means that two new offspring classifiers are generated maintaining some traits from their parents. The offspring classifiers are inserted into the population set, replacing classifiers with low fitness value. Then the learning process is repeated again until the population set evolves into a collection of classifiers with relatively high accuracy values.

Meanwhile, Organizational Learning algorithm divides the rules into two levels: individual level and organizational level. The individual level rules are accessible only for the relevant agents, while the organizational level rules are available for all agents.

Rules that are proven to be good at individual level are gathered as candidates for promotion to the organizational level. Those “good” rules are evaluated by Roulette Wheel algorithm based on their fitness value. The bigger the value, the higher the probability a rule gets selected.

When the population of the organizational level of knowledge is full, rules that perform poorly at organizational level are put into candidates for demotion back into individual level. Similar to the promotion system, those “poor” rules are evaluated by Roulette Wheel algorithm based on their fitness value. The lower the value, the bigger the probability of a rule gets demoted back to the individual level.

11. Simulation

Since there is no public data is available that best represent our model, we generates artificial big data to test our method and compare its performance against other recommendation methods, such as collaborative filtering, content-based filtering, and naive bayes classifier. One public data that slightly closely resembles our model (after some modification) is MovieLens dataset. However, less detailed information are provided in the data (such as device information, etc). We also test our method with this data and compare the performance with other recommendation methods. The training and testing are done with 5-fold cross validation method. At each fold, the system is trained using 80% of the dataset, and tested against the remaining 20% datasets.

MovieLens datasets have many limitations. It has very few metadata and it doesn't provide details of under what circumstances users liked a movie. In movie recommender case, a user does not simply like a movie, there are many factors involved in it. For example, a female teenager might

often watched cartoons on her tablet, but prefer watching science fictions on a TV (for better screen resolution). Another example, a married man might often watched comedy on weekdays, but prefer watching family-oriented movies on weekend (with his family). Since there is no public data available to simulate this situation, we generated artificial big data sets consisting of one million agents, one million movies, and 100 million “LIKE”s / ratings.

The system using this dataset is evaluated according to the rules of precision & recall. Precision for a class is the number of true positives (i.e. the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class. While Recall in this context is defined as the number of true positives divided by the total number of elements that actually belong to the positive class.

12. Result

Both standard LCS and XOLCS took significantly longer time to finish the computation. While Standard LCS does not seem to provide accurate prediction, XOLCS managed to reach a high precision value. When we don't mind sacrificing computing in chase of high accuracy, XOLCS is the best choice compared to other methods.

From our experiments, the frequency of triggering the Organizational learning need not to be high. This is because the rules at organizational level are already proven to be good enough at individual level that chances are they would perform similarly well at organizational level. A relative low value is generally good to achieve a high accuracy.

As we already know, collaborative filtering method is not capable to recommend new items due to no rating data about them. Meanwhile content-based filtering and naive bayes classifier method managed to provide recommendation to more than half of the items inside the new item set. Standard LCS and XOLCS are the winners in coverage category because they have rules available to recommend 98% of the items.

XOLCS took much longer time to learn, but in the end it provided the most accurate recommendation for the dataset. Naive bayes classifier is very costly, especially considering their accuracy is not much different than collaborative filtering and content-based filtering. Finally collaborative filtering and content-based filtering are reasonably good choices if the system need to make quick recommendation with acceptable accuracy.

13. Discussion

The main advantage of LCS-based RS is its flexibility:

- The length of classifiers can be adjusted depending on the number of inputs is available. When more inputs are available (e.g. future device may add new sensory information), RS designers simply need to lengthen the classifier's chromosome. The learning procedures stay the same.

- The number of classifiers can be increased or decreased accordingly to meet the available computing resource. When more clusters are available, the RS designer only need to change the parameter the defines that maximum number of classifiers.

- Other methods need more data to refine their accuracy, but LCS-based RS may only need more learning time to refine the accuracy. In the long run, LCS-based RSs reach more accurate predictions than other recommendation methods. In the era where even a 0.1 increase of RMSE may translate into millions of revenue, the additional time necessary to learn the model could be worth the waiting.

14. Limitation of XOLCS

At its current state, our XOLCS model is designed for single-step environment in mind. In a multi-step environment, another type of reinforcement learning with discount factor parameter is necessary. Covering the multi-step environment is beyond the scope of this thesis but the author is committed to update the model upon completion of this thesis.

XOLCS generally takes more computing resources to preprocess large data compared to other methods. In a system that is not capable to handle distributed multi-agent programming, XOLCS recommender may not be a good choice.

15. Concluding Remarks

This thesis shows possibilities of applying Learning Classifier Systems (LCS) technique in a multi-agent environment. To make it work, we implemented methods inspired by organizational learning

concept. Our architecture provides individual level knowledge and organizational level knowledge and mechanisms to transfer the knowledge between the two levels. By applying the organizational learning concept, the agents are able to make use knowledge the is acquired by other agents and preventing them learning new situation from the beginning. Furthermore, by extending the classifier to include accuracy parameters that keep track the prediction error, the LCS will evolve the population set to include only high quality rules, in terms of prediction accuracy.

We also show how to approach recommender systems (an example of predictive analytics problem) with LCS. By replacing ratings with classifier rules as the transaction connection between items and users, the system can learn not only the users' preference, but also able to deduce factors that make them prefer certain items under certain conditions. We demonstrated that LCS is a feasible approach for recommender systems.

Most recommender algorithms were designed with rating system in mind. This means that all previous ratings will affect future predictions regardless of user's change of taste. Furthermore, in this age of big data, ratings are not the only relevant inputs that can be gathered from users. More types of inputs are helpful to create user model that can predict user's behavior and preferences.

With XOLCS, the system does not only predict whether a user like a movie, but also guess why he likes it. For example, a married businessman might prefer watching romance movies at home on TV, but prefer watching comedy shows on a smartphone while commuting on a train. Another example, at home a teenager might often watch cartoons on his tablet, but often watch science fiction movies on a High Definition TV. Overall, the results showed that Multi-Agent LCS is able to generate good predictions.

XOLCS is more flexible, in a sense that, it accepts any inputs, such as profile data, time, location, etc. It can by anything as long as it can be represented as real valued number. Since the population is dynamic, old rules that are irrelevant now (regardless of how good they were before) are either deleted or evolved by genetic algorithms. This means the system is able to recognize if there are changes to the users' preferences.

16. Future Potentials

There are many potentials for future extensions of our model. A possible extension would be a method for classifier traversal. The idea is to allow user to customize their recommendation by

themselves. This way, the system will only need to compute those classifiers that are relevant to user's questions, such as:

- “What movies were my friends watching?”
  (limit only recommendation related to their friends on social network)
- “What movies my friends 'like'd that stars Tom Cruise?”
  (limit recommendation related to their friends, starring Tom Cruise as the lead actor)
- “What action movies most people watched last week?”
  (limit the recommendation to action movies that were popular the previous week)

This could introduce a new concept of recommendation-on-demand and might as well become a necessity in future web service. We hope that our thesis can be a good reference for those future models.