Vol.2010-ICS-158 No.8
2010/3/1

Problem Digestion by Assembly of Interest-Driven Agents

Hamed Khandan[*1] and Takao Terano[*1]

It is well understood that an ensemble of problem solvers works better than any single of them, as a fact that is studied with Bagging, Boosting and Hierarchical Mixture of Experts. We bring this power of ensembles to a whole new front by employing autonomous problem solver agents who engage in problem solving organizational assemblies, and propose it as a superior framework for these sorts of problems. The organization has higher level nodes, namely, agencies, who keep track of other agents, but do not manage them. Experts have absolute freedom to choose their parts in the problem at hand, giving the assembly certain advantages. Implementation of this idea demonstrates considerable improvement both in terms of accuracy and speed.

1. Introduction

Several decades ago, Marvin Minsky introduced the concept of agents; and the way these simple agents can work together to carry out considerably complicated intelligent tasks. Today software agents are widely used for simulation purposes and networking (as mobile agents). In terms of AI, also, making so-called intelligent agents are common. But despite the ever increasing popularity of agent-based systems, the direction of original Minsky's vision is the least regarded. Yet, nothing is more idealistic than the ability of solving arbitrary problems by a computational system; and no other theory can make such a strong claim, but Minsky's "Society of Mind[8)]".

With its aim in that direction, this report is a part of our research called "Trade-Oriented Artificial Social Intelligence" or TASI for short. TASI is an effort to establish a "society of mind" with a socio-mimicry discipline, assuming cooperation of people and organizations via trade, as the model. Currently we restrict this research to regression and classification problems. Specifically, regression solutions are vital for a wide range of intelligent machine control applications; and we hope to mark an advancement in machine control by focusing on that topic.

*1 Presently with Tokyo Institute of Technology

Our discussion here is dedicated to the interest-driven behavior of problem-solver agents. We testify this management philosophy that freedom helps productivity and leads to better results. Therefore, we eliminate the control of higher level authorities over their employees to establish them the right of *freedom*, and define a measure of *joy* for them, which is related to their payoff. We observe then, with less number of agents the problem can be effictively solved — a good lesson to learn not only by AI community, but also by corporate managers (More discussion in Section 3 and 3.1).

There exist a number of classic solutions, collectively known as *ensemble methods*. Although these are not formulated from an agent-based perspective, they can be merely seen as if they were. These methods including, Bagging[4)], Boosting[6)], Classification and Regression Trees (CART)[3)], and Hierarchical Mixture of Experts (HME)[7)1)], put together the outputs of an array of problem-solvers to carry out higher quality results. In ensemble literature, these problem solvers are called *experts*, but for certain reasons we rather the term *specialist* here. There is a lesson to learn from the simplest one, namely, Bagging, which says the result of joining several problem-solvers together is just better than any single of them.[5)] Boosting, yet works better than Bagging. HME is generalized decision tree with soft-switching functionality that identifies the best solvers for each input. In order to achieve our objective in this paper, we came up with a *meta* framework that not only covers all of these ensemble methods, but helps to create newer ones easily. However, in contrast with these methods, our main concern will be the specialists rather than the fusion algorithm.

In the next section of this paper (section 2) we talk about the framework just mentioned above. In section 3, functional behavior of agents is explained. Section 4 describes the platform we created to implement our solution as well as the practical results. Section 5 concludes this paper with a few discussions.

2. Modular Model

2.1 Social Roles

In factories or corporations, employees are usually assigned to a set of tasks such

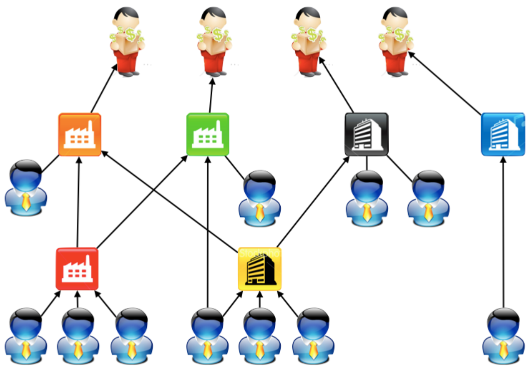

Fig. 1 Product chain model

as cleaning the floor or research for a cure for cancer. We say, these employees are specialists in those tasks and call him an *specialist agents*. The job of specialists is always production, whether they produce goods or services. For example, product of an specialist can be the service of clearing the floor. Factories and corporations and other forms of institutions are not just a gathering of agents but they provide facilities for them to put their products together and produce more sophisticated products. We apply the term *agency* to these institutions. Agencies are, in a sense, complex specialist; since they are productive in specific fields. So, in the terms of object-oriented programming, it is fair to call `Agency` as a sub-classes of `Specialist` which have references to a collection of other specialists. Specialists (and agencies) bound themselves via *contracts* with their respective institution(s), which, among other things, indicates how they are going to be rewarded for their productivity. The network of interconnected agents that leads to production of an specific product is called a *supply chain*. At the top of the supply chain stand the *customers* who are the actual payers for the whole chain. The operation of agents in a supply chain is driven by end customers (see Figure 1).

Hence, we defined three essential types of agents:

- Specialist — those who implement a specific solution and focus on a limited portion of the pattern-space.
- Agency — A specialist institution emerged by the gathering of some specialists. The function of agency is (1) to maintain a record of its members and their performance, (2) provide members with information about the overall performance of their group, (3) identify and advertise opportunities for higher payoff to its members.
- Customer — A human or machine agent who asks the system to solve a problem.

2.2 Ensemble Framework

Here, three types of agents introduced above are used to establish a framework for ensemble methods. These framework is essential for future topics.

A regression or classification problem is about the mapping between a set of inputs or patterns to a set of output or targets. We call the set of inputs, the pattern-space, and the set of outputs, the target-space. In classification problems, the target-space is discreet and countable. If we imagine a classifier or a regressor as an specialist, formulating an ensemble method is the matter of designing the agency which is going to combine the products of those specialists working on the same patterns-space to produce outputs with higher accuracy.

The function of agency is illustrated with M specialists in Figure 2. First, it feeds training data to the corresponding specialists in a proper manner. Then, customer comes and makes a query asking a pattern to be solved. The agency forwards the query to appropriate specialists and each individual of them sends back a value in the target space. The agency uses a function per expert, called the gating function (π_m) which indicates the trustability of each specialist's results. π_m is sometimes a number which is derived from overall error of each expert, and sometimes is a function of pattern x, denoted as $\pi_m(x)$, which determines the trustability of each expert for each point in the patterns space. Having the value of π_m, the agency performs a weighted average on the outputs of experts and provides the customer with the final result in the target-space.

Table 1 summarizes how we can fit all types of ensemble methods inside this framework. It can be seen how it was easy for us to propose a better ensemble method. And it shall be easy for others as well.

Now, although this framework is very useful, our real intention was not to formulate the agency, but to put them aside with peace of mind, and open up a

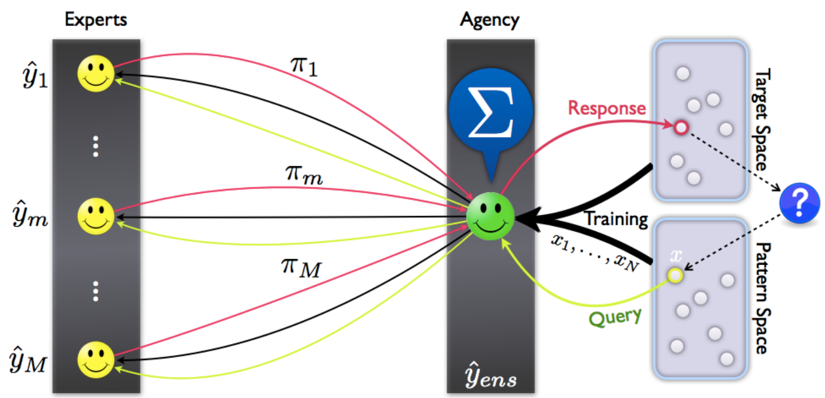

Fig. 2 Framework for ensemble methods

space for the specialists to show up. This will be the topic of the next section.

3. Functional Model

We believe providing employees with enough information and letting them to be free to choose what they want to do, will enhance their performance. We put this idea into test here. We want specialists to be free to choose their part in the work at hand, rather than to be chosen by agencies. In fact, the specialists (1) shall create their higher-level agencies, and (2) shall decide which part of the problem they want to solve. The following sub-sections describe the functionality of agencies, and the personal and social aspects of the functionality of specialists.

3.1 Functionality of Agencies

In fact, the important capability of higher level agents is not their potential to give orders to the others, but their broader knowledge of the problem space and meta parameters. That means, it doesn't heart to eliminate management privilege from agencies to set specialists free. Instead, an agency can provide its knowledge to it member agents a service. Hence, we define agency as an agent providing the following services:

- Provision of training data for the area of its coverage.
- Storing and providing meta-information of its member agents.
- Calculation and publication of the ensemble error profile of its under-cover agents.
- Identification of potential places with higher payoff.

	Gating	Training	Query
Bagging	$\pi_m = \frac{1}{M}$	Bootstrap	Send to all
Boosting	$\pi_m = g(1 - \epsilon_m)$	$w_{m,n} = f(\hat{y}_{m-1}(x_n))$	Send to all
Decision Tree	$\pi_m \in \{0, 1\}$	Send to all	To the best
HME	$\pi_m(x) = \frac{e^{\xi_m(x)}}{\sum_{k=1}^{M} e^{\xi_k(x)}}$	Send to all	Send to all
Ours	Mixture of Gaussians	Interest agents	Eligible agents
Your's	...	...	...

Table 1 Summary of ensemble methods

3.2 Personal Functionality of Specialists

A specialist should be able to decide which part of the pattern-space it wants to focus on, and they make that decision based on their sense of "joy". We call this, *the pattern-space allocation problems*. Basically, joy is a measure of satisfaction. One can imagine many interpretations for joy as well as many approaches to pattern-space allocation. We explain only a simple one of them here. In this work, the satisfaction is determined with respect to the amount of contribution of an individual in the reduction of ensemble error.

The pattern-space allocation problem can be discussed with two parameter: the center of focus, and the area of focus. As of the center, we implement a sort of steepest ascend, that can also be compared with sugar-scape problem. An agent, first asks its agency for information about ensemble error profile. Then, with the help of agency, it identifies the nearest and steepest way up and moves in that direction. The movement paste is proportional to the attractiveness of the target point. For the focus range, the expert can employ recursive least squares to determine whether to shrink or expand its coverage, and to converge to a appropriate amount of coverage.

3.3 Social Functionality of Specialists

When a specialist finds a problem hard to solve, it advertises that opportunity for other specialists. Perhaps, one may be interested to make profit out of that problem and replies to the advertisement of that agent. Two agents, then, establish an agency together in order to produce ensemble results and farther continue their collaboration. If the problem is hard enough, or wide enough, that agency may employ more specialists later.

Please notice this interesting phenomena of bottom-up assembly of the organization, which is very rare in computational intelligence, but very common in the real business world.

The bottom-up approach in formation of the social assembly comes up with an interesting property of self-healing, from within its very nature. If an agency fails for any reason, replacement will be generated by specialists. And if specialists fail, replacements will be also generated by agencies. If it was top-down approach, higher level agencies which are more vital, could not be replaced.

4. Implementation and Results

We created a class library called *TASI* in Java in order to make a development platform for interest-driven trade-oriented agents. Because existing agent-development platforms were not sufficient for our purposes, we also developed a general-purposed agent-development with the desired properties and called it *Sarv*. Figure 3 shows the Sarv/TASI platform bundle at one glance. We also made a basic GUI to observe the performance of TASI-based agents.

In TASI platform, `TASIAgent` implements "ContractNet" protocol[9] for interaction between agents. `RegressionAgent` implements the interest-driven behavior explained above, and `RegressionAgency` is a sub-class of `RegressionAgent`, as expected.

The figure 4 shows the test we conducted with this system. The target function is a period of sine function, and specialists are single Gaussian bell-shaped function who fit themselves to the portion of the problem under their coverage using simple IR-LS algorithm[2] ⋆1. It can be seen that in the second step, the experts move towards two high "hills" marked in the picture. As of the third step, the estimation already makes sense, and it takes about 10 steps for the system to converge to an acceptable estimation.

While usually a linear regression solution needs about 25 instances of basis function[2] ⋆2, we could achieve with only a few, by implementing our solution.

⋆1 Section 4.3.3
⋆2 See figure 3.5

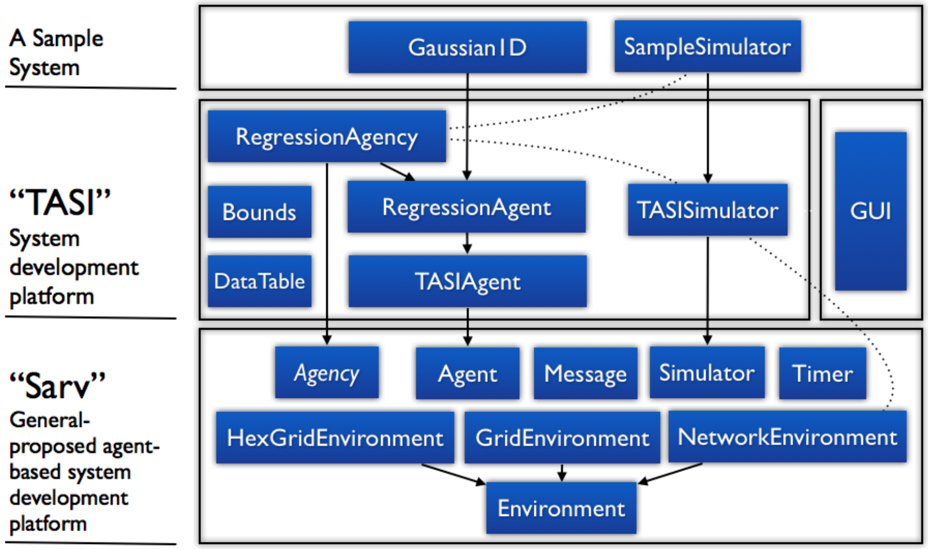

Fig. 3 Inheritance graph for Sarv/TASI class library bundle

Although in training phase, this system has high message cost and may wast more time for interaction between agents, it is obviously faster and more accurate in evaluation phase and is expected to demonstrate an enormous degree of adaptability.

5. Conclusion and Future Works

The innovations and experiences reported in this paper was made to revive the theory of "Society of Mind" with a new discipline of socio-mimicry. Our work in that respect, is to use business as a model and design three basic types of software agents, namely, specialists, agencies, and customers, with certain properties and behaviors to establish an institutionalized society accordingly. We are also constantly working to develop a software platform to make it easy to create these agent.

We focus on regression problem, designing an *adaptable* ensemble meta architecture, and programming a number of simple agents for that purpose. Giving

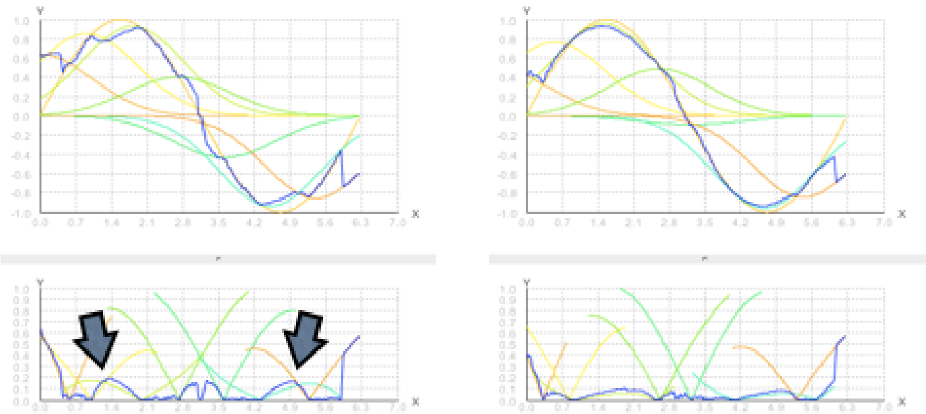

Fig. 4 Evaluation result — Upper parts show the output of indivisuals and the ensemble. The lower parts whos the error. The left set of images show the system after one step, and the right set of images shows it after 10 steps.

the agents the right of "freedom" and feeling of "joy" we observe that the ensemble becomes far more productive than known ensemble method, producing better results by consuming less resources. It can also be argued that a social assembly generated by these interest-driven agents demonstrates the capability of self-healing.

As of the future, we will continue this trend. We plan to implement heterogeneous specialists and let this system solve more sophisticated problems. Also an economical system is missing in our design which will be designed and implemented.

References

1) Bishop, C.M. and Svensn, M.: Bayesian hierarchical mixtures of experts, *Proceedings Nineteenth Conference on Uncertainty in Artificial Intelligence* (Kjaerulff, U. and Meek, C., eds.), Morgan Kaufmann, p.5764 (2003).
2) Bishop, C.M.: *Pattern Recognition and Machine Learning*, Springer Science (2006).
3) Bramer, M.: *Principles of Data Mining*, Springer London (2007).
4) Breiman, L.: Bagging predictors, *Machine Learning*, Vol. 24, No. 2, pp. 123–140 (1996).
5) Bhlmann, P. and Yu, B.: Analyzing Bagging, *Annals of Statistics*, Vol.30, No.4, pp.927–961 (2002).
6) Freund, Y.: Boosting a Weak Learning Algorithm by Majority, *Information and Computation*, Vol.121, No.2, pp.256 – 285 (1995).
7) Jordan, M.I. and Jacobs, R.A.: Hierarchical Mixtures of Experts and the EM Algorithm, *Neural Computation*, Vol.6, No.2, pp.181–214 (1994).
8) Minsky, M.: *The Society of Mind*, Simon & Schuster (1988).
9) Smith, R.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, *Computers, IEEE Transactions on*, Vol.C-29, No.12, pp.1104–1113 (1980).

(Received February 2, 2010)
(Accepted February 2, 2010)

Hamed Khandan is a second-year Ph.D. student at Tokyo Institute of Technology, Department of Computational Intelligence and System Sciences. He received his bachelor degree in computer software engineering in 2004, and holds a M.S. in mechatronics from the Science and Research branch of I. Azad University (2008). Research interests include: philosophy of science, emotional computing, artificial social intelligence, and distributed machine control.

Takao Terano is a professor at Department of Computational Intelligence and System Science, Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology. He received BA degree in Mathematical Engineering in 1976 from University of Tokyo, M. A. degree in Information Engineering in 1978 from University of Tokyo, and Doctor of Engineering Degree in 1991 from Tokyo Institute of Technology. His research interests include Agent-based Modeling, Knowledge Systems, Evolutionary Computation, and Service Science. He is a member of the editorial board of major Artificial Intelligence- and System science- related academic societies in Japan and a member of IEEE, AAAI, and ACM. He is also the president of PAAA.