

A Novel Insight into Learning Theory: The Gap between Teaching and Learning

Jian Yu, and Pengwei Hao

Abstract¹

In our education system, teacher hopes his students to learn something specific from his demonstrations and textbook, his students try to understand his teacher's demonstration and textbook by their own learning methods. Obviously, the aim of the teacher may not be achievable for all students' learning methods. Therefore, students' final learning results are different from teacher's expectation in general, which is called the gap between teaching and learning (in short, GTL) in this paper. As the goal of machine learning is to design a computer program with learning ability, it is naturally questioned if GTL occurs in the machine learning fields. In this paper, we prove that there exists GTL in machine learning. As a common assumption in the current learning theory is that a learning algorithm usually realizes the original expectation, GTL provides a new insight into learning theory. According to the GTL Theory, the learning algorithms can be classified into four types, Type I through Type-IV. Comparison with human learning, the GTL Theory substantiates an intuitive observation: artificial intelligence can never surpass human intelligence from the learning point of view.

Keywords: data, model, learning algorithm

1. Introduction

This document is a template for Microsoft Word versions 6.0 or later, its content provides authors with most of the formatting specifications needed for preparing electronic versions of their papers. All standard paper components have been specified for three reasons: First, ease of use when formatting individual papers; second, automatic compliance to electronic requirements that facilitate the concurrent or later production of electronic products; third, conformity of style throughout a journal

proceedings.

Roughly speaking, intelligence at least includes computing ability, memory ability and learning ability. Current computers have stronger computing and memory than human beings, at least for well-defined computational problems and rote learning. However, people always have such a feeling that machine learning ability is too much weaker than that of human beings. Up to now, no theoretical analysis for this intuition has been published. This paper will focus on this issue. First, we recall the conventional definition of machine learning. In machine learning, a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E, (see [9]). Therefore, machine learning is comprised of four components: experience, task, performance, and computer program. In computer science, experience is often represented by data, task and performance are usually integrated into a model, and computer program often represents an algorithm. Obviously, the relations among data, model and algorithm are very complex.

Nowadays, extensive research has been focus on the relations between data and model, and a great progress has been made. For instance, Minsky [8] has proved that a too simple model cannot fit any data well, like linear perceptions. Therefore, people have designed many models with enough complexity, like neural networks, decision tree and Bayesian network. In the last century, it was theoretically proved that neural networks are complex enough to fit any data in a perfect way, (see [4]). When studying the relation between data and model, we often take it for granted that the model may not perfectly fit the data. As noted by Poggio, Rifkin, Mukherjee, & Niyogi [10], the basic requirement for a learning algorithm is that its performance on the training data is as the same as that on the future real data. Therefore, sample complexity and generalization ability are the main concern of current learning theory, like PAC theory, and statistical learning theory.

In current learning theory like statistical learning theory, it is supposed that a learning algorithm can perfectly implement its corresponding model, in other words, the learning algorithm can perfectly realize its expectation – the original task and performance. If the model is very simple, the above assumption is reasonable. When the

Corresponding Author: Jian Yu is with the Department of Computer Science, Beijing Jiaotong University, Beijing, 100044.

E-mail: jianyu@center.njtu.edu.cn

Manuscript received 12 Nov. 2007; accepted 1 Dec. 2007.

model is very complex, it is not always feasible to develop a learning algorithm to realize the expectation of the model. Usually, the model is described by minimizing the objective function. Generally speaking, the computational complexity of minimizing the objective function is NP hard, (see [1], [11]). Therefore, it often uses heuristic algorithms like gradient descent or greedy research to find a local minimum instead of the global minimum, (see [5]). Even so, it still has two questions to answer given that the model is described by minimizing the objective function:

Any local minimum of the objective function can be the stable output of heuristic algorithms?

Any stable outputs of the learning algorithm like heuristic methods can be the local minimum of the objective function?

In other words, the learning algorithm can exactly realize the expectation of its corresponding model? What is the relation between a model and the corresponding algorithm with respect to learning? This paper is devoted to studying this issue.

In order to illustrate our idea clearly, we go back to human learning first. In our education system, a teacher hopes his students to learn something specific from his demonstrations and textbook, his students try to understand his teacher's teaching by their own learning methods. Obviously, the aims of the teacher may not be achievable for all students' learning methods. Therefore, students' final learning results are generally different from teacher's expectation, which is called the gap between teaching and learning (in short, GTL) in this paper. More specifically, if that the student has learned is just as the same as that the teacher expected, we say, the teaching is consistent with the learning; if that the student has learned is less than that the teacher expected, we say, the teaching is stronger than the learning; if that the student has learned is more than that the teacher expected, we say, the teaching is weaker than the learning; etc.

Notice that teacher is usually supposed to be an interpreter of the textbook, the role that the teacher plays in human education is very similar to that the model in machine learning. Likely, textbook vs. data and student vs. algorithm play the similar roles. As noted above, the gap between teaching and learning occurs during human learning process, it is natural to ask whether or not the similar phenomena happen in machine learning.

It is easy to guess that the similar phenomena occur in machine learning. In the standard language of machine learning, the results outputted by the algorithm may be inconsistent with those expected by the model, in other words, the algorithm maybe not perfectly implement its corresponding model. In the following, we will give a detailed proof of this fact.

The reminder of this paper is organized as follows: Section 2 describes the roles the data, the model and the algorithm play in machine learning. Section 3 introduces the definition of the gap between teaching and learning in machine learning, and classifies the learning algorithms into four types based on GTL. Moreover, we prove that GTL indeed occurs in machine learning. As an application of GTL, we study two learning algorithms: general c-means clustering model and perceptron, and we prove that perceptron algorithm is type-I learning algorithm when the training set is separable. In section 4, we present a perspective of the GTL theory.

2. Data, Model and Algorithm

In order to clearly show our idea introduced in Section 1, we give the formal definitions in machine learning.

Data space: observed data, or transformed data of original observed data

Hypothesis space (HS): the knowledge that can be obtained from the observed data in the data space, which is usually represented by a function.

Model: the criterion on how to choose a concrete element in the hypothesis space according to the specific data in the data space.

Learning algorithm (LA): a learning algorithm is such a map from the data space to the hypothesis space that its output is an approximation of the hypothesis that defined by its corresponding model as sharply as possible.

A machine learning problem can be divided into the four components: a data space, a hypothesis space, a model and a learning algorithm. The above four components plays a very important role in machine learning.

Data space need not be the original observed data space since the original data may be not inappropriate for further processing, so we can apply some transforms to the original data to change them into the wanted data structure. For instance, dimensionality reduction or feature selection is very important technique for machine learning, like principle component analysis (PCA), multidimensional scaling, independent component analysis (ICA) and locally linear embedding, etc.

Hypothesis space represents the knowledge that we expected to obtain from the data, and also limits the search scope that the learning algorithm need scan. Naturally, different hypothesis spaces have a great impact on designing the corresponding model and algorithm; hence it is a pivotal problem to choose an appropriate hypothesis space. If the hypothesis space is not properly chose, we have no chance to find the real knowledge hidden behind the data no matter how learning algorithm are designed. Commonly, hypothesis space are some space of functions, see

A model is not only a hint for developing the learning algorithm, but also a guide for evaluating the performance of the learning algorithm. Accurately speaking, a model not only defines the expected output that we hope to obtain from the hypothesis space in theory, but also explains the physical meaning of the learning algorithm and sometimes provides a tool to prove the convergence of the learning algorithm. Commonly, all the prior knowledge about the given data is integrated into the model as totally as possible. Transparently, the prior knowledge about the data changes, the model also changes.

Learning algorithm describes the process to find the output from hypothesis space according to the model, and is a run-time architecture to extract the practical knowledge from the given data. In practice, that learning has time limit means that applicable learning algorithm should be polynomial time.

Sometimes, a model can be considered to coincide with the hypothesis space like perceptron, and sometimes model is isomorphic to the learning algorithm like Alternating Cluster Estimation [12], more detail can be seen in Section 3. Strictly speaking, a model focuses on describing a learning problem in mathematical language as exact and simple as possible; a learning algorithm is devoted to execute the process introduced by the model as fast and accurate as possible. Therefore, the requirements of designing a model are different from that of developing the corresponding learning algorithm with respect to a specific learning problem. Thus, a model is different from a learning algorithm in general cases.

When the results that a learning algorithm can output is the totally same as defined by the model, we can take them as one isomorph. Such cases indeed exist, for example, in a K-nearest neighbor method in supervised learning, its model and algorithm can be considered as one isomorph. Most previous research in machine learning takes an learning algorithm and its corresponding model as one isomorph, for example, empirical risk minimization (ERM) algorithms are defined in [10] as those satisfying:

$$l_s[f_s] = \min_{f \in HS} l_s[f] \quad (1)$$

Consequently, one research focus of machine learning is to determine the conditions under which a learning algorithm will generalize from its finite training set to novel examples, as indicated by Cucker & Smale [3]. About this issue, people have made several great achievements, for example, on the conditions for predictivity or generalization for many specific leaning algorithms, like Vapnik [13]; Poggio, Rifkin, Mukherjee, & Niyogi [10]; Bousquet & Elisseeff [2]; Evgeniou, Pontil, & Elisseeff [6]; Freund, & Schapire [7]; Zhou [15].

As noted in Section 1, it is impossible to design an

algorithm to find the global minimum of the objective function $l_s[f]$ within polynomial time. It is proved that globally fitting the weights of a neural network or seeking the smallest decision tree are NP-complete problems, (see [1], [11]). Such analysis tells us that we have to lessen our expectation of finding the global minimum. Usually, it is satisfactory in practice to obtain a local minimum within polynomial time. Even after such a trade-off, not all algorithms can meet such expectation of its corresponding model. In Section 3, we will study this case in detail.

3. Gap between teaching and

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the reference list. Use letters for table footnotes.

As noted in Section 1, GTL phenomenon occurs during a human learning process. It is natural to ask if GTL occurs in machine learning. In order to answer this question, we give a deep analysis of GTL in a human learning process. For simplicity, we denote the teacher's expectation as A, that some students have learned as B. Usually, a teacher meets four cases: $A=B$; $A \supseteq B$; $A \subseteq B$; $A-B \neq \emptyset$ and $B-A \neq \emptyset$. In fact, the above four cases respectively represent four types of students: learning perfectly matches teaching; learning is weaker than teaching; learning is stronger than teaching; learning does not match teaching. Inspired by such an analysis, we first give several relevant definitions in machine learning as follows:

Teachable set (TS): the elements in the hypothesis space that is defined by the model.

Learned set (LS): the stable outputs of the learning algorithm.

Obviously, the operative determinations of a teachable set and a learned set depend on the specific learning problems. Sometimes, it is very challenging to determine the proper teachable set and the learned set. In the later, we will address this issue. Basically, we suppose: $TS \subseteq HS$, $LS \subseteq HS$.

If the teachable set and the learned set are determined for a given learning algorithm, we can judge the relation between the teachable set and the learned set as follows:

Type-I learning algorithm: $TS=LS$

Type-II learning algorithm: $TS \supseteq LS$

Type-III learning algorithm: $TS \subseteq LS$

Type-IV learning algorithm:
 $TS - LS \neq \emptyset, LS - TS \neq \emptyset$

Furthermore, if $TS=LS$ only holds for a given data set S, the learning algorithm is called type I with respect to S. Similarly, we can define type II, III and IV with re-

spect to S.

Equation $TS \neq \emptyset$ does not always hold. Sometimes, the model maybe consist of contradictory constraints, therefore it is possible that $TS = \emptyset$. Similarly, $LS \neq \emptyset$ does not always hold. Traditionally, $|TS|=1$ means that the model is well posed, and $|LS|=1$ means that the learning algorithm is well posed. Thus $|LS|=1$ means that the learning algorithm is independent of its initialization. Generally speaking, different initializations usually lead to different outputs as for a learning algorithm. $|LS|=1$ is usually impossible for a learning algorithm. In particular, if $TS = \emptyset$, the learning algorithm is Type-III; if $LS = \emptyset$, the learning algorithm is Type-II.

The above analysis offers a novel framework to study learning algorithms. In other words, we can judge what type a learning algorithm belongs to, which is very helpful to understand the properties of a learning algorithm. If a learning algorithm is Type-I, it is impossible to be improved in the learning sense. If a learning algorithm is not Type-I, it has some room to improve in theory, at least from the view of machine learning.

As the designer of any learning algorithm always hope that the learning algorithm can realize the expectation of the model as exact as possible, therefore we call Type-I learning algorithm a perfect learning algorithm. As for machine learning, we hope that any successful learning algorithm had better belong to Type-I. However, such a hope is one dream for machine learning, which cannot always come true. As a matter of fact, it is well known that it is impossible to develop a learning algorithm to realize the expectation of a too complex model, although it is a common assumption when studying generalization in learning theory.

In the following, we use two learning algorithms to show how to use GTL theory. Example 1 is from unsupervised learning, and Example 2 from supervised learning.

Example 1. $X = \{x_1, x_2, \dots, x_n\}$ is a s-dimensional data set, $v = \{v_1, v_2, \dots, v_c\}$ is cluster prototype. $v = \{v_1, v_2, \dots, v_c\}$ is obtained by minimizing the objective function as below:

$$Q = \frac{1}{n} \sum_{k=1}^n a_k f \left(\sum_{i=1}^c \alpha_i g(d_{ik}) \right) \quad , \quad \text{where} \quad (2)$$

$$\sum_{i=1}^c \alpha_i = 1, \forall i, \alpha_i \geq 0, \quad \forall k, a_k \geq 0 \quad , \quad d_{ik} = \|x_k - v_i\|^2$$

According to (Yu, 2005), minimizing (2) leads to the update equation for cluster centers in the general

c-means clustering algorithm (GCM) ² as follows:

$$v_i = \frac{\sum_{k=1}^n a_k f' \left(\sum_{i=1}^c \alpha_i g(d_{ik}) \right) g'(d_{ik}) x_k}{\sum_{k=1}^n a_k f' \left(\sum_{i=1}^c \alpha_i g(d_{ik}) \right) g'(d_{ik})} \quad (3)$$

Then the procedure of the GCM clustering algorithm can be described as follows:

Initialize: set $v^{(0)} = (v_1^{(0)}, v_2^{(0)}, \dots, v_c^{(0)})$, and terminal limit ε , and the maximal iteration number T, t=0

Step 1: update $v^{(t)} = (v_1^{(t)}, v_2^{(t)}, \dots, v_c^{(t)})$ with update equation (3) with $v^{(t-1)} = (v_1^{(t-1)}, v_2^{(t-1)}, \dots, v_c^{(t-1)})$ (For brevity, this step can be expressed by $v^{(t)} = \theta(v^{(t-1)})$).

Step 2: If $t > T$ or $\|v^{(t)} - v^{(t-1)}\| \leq \varepsilon$, then output $v^{(t)} = (v_1^{(t)}, v_2^{(t)}, \dots, v_c^{(t)})$, stop; otherwise, $t=t+1$, go to Step 1.

Obviously, in example 1, the data space is R^s , the hypothesis space is $R^{c \times s}$, the model is $\min_v Q$, the learning algorithm is the above programming. If the teachable set of GCM is defined as the global minimum of the objective function Q, it is easy to know that GCM is type-III learning algorithm, as it has a very low probability that GCM outputs the global minimum of Q.

However, it is a common sense that it is impossible to find out the global minimum of Q using alternating optimization. Naturally, it is impossible that GCM has always output the global minimum of Q, even in a high probability. Therefore, the usual expectation for GCM is to find out the local minimum of Q instead. If so, the teachable set of this problem is composed of the local minimum of Q.

By stability analysis of dynamic systems, we known that the output of GCM should be the stable fixed point of dynamic systems described by (3) when it is convergent, i.e. the learned set of this problem consists of the stable fixed point of dynamic systems described by (3). In the following, we will prove that even under such a circumstance, GCM is not a type-I learning algorithm.

According to the above analysis, the teachable set and the learned set can be defined respectively in mathematical language as follows:

$$TS = \left\{ v \in R^{c \times s} \left| \frac{\partial Q}{\partial v_i} = 0; \left[\frac{\partial^2 Q}{\partial v_i \partial v_j} \right]_{c \times c} \text{ is a positive matrix} \right. \right\}$$

where:

$$\frac{\partial Q}{\partial v_i} = -2 \sum_{k=1}^n a_k f' \left(\sum_{i=1}^c \alpha_i g(d_{ik}) \right) g'(d_{ik}) (x_k - v_i)$$

²The reason we choose the GCM as an example is as follows: GCM is a comprehensive framework of partitional clustering, more detail can be seen in [14].

$$\begin{aligned} \frac{\partial Q^2}{\partial v_i \partial v_j} &= 4\alpha_i \alpha_j \sum_{k=1}^n f''(S_k) g'(d_{jk}) g'(d_{ik}) (x_k - v_i)(x_k - v_j)^T \\ &+ 4\delta_{ij} \alpha_j \sum_{k=1}^n f'(S_k) g''(d_{ik}) (x_k - v_i)(x_k - v_j)^T \\ &+ 2\delta_{ij} \alpha_i \sum_{k=1}^n f'(S_k) g'(d_{ik}) I_s, \quad 1 \leq i, j \leq c \end{aligned}$$

$$LS = \left\{ v \in R^{c \times s} \left| \begin{array}{l} \|\lambda\| < 1, \lambda \text{ is any eigenvalue} \\ \text{of } \Theta, \Theta = \left[\frac{\partial \theta_i}{\partial v_j} \right]_{c \times c}, v = \theta(v) \end{array} \right. \right\}$$

Where:

$$\begin{aligned} \frac{\partial \theta_i}{\partial v_j} &= -2\alpha_j \frac{\sum_{k=1}^n a_k f''(S_k) g'(d_{ik}) g'(d_{jk}) (x_k - v_i)(x_k - v_j)^T}{\sum_{k=1}^n a_k f'(S_k) g'(d_{ik})} \\ &- 2\delta_{ij} \frac{\sum_{k=1}^n a_k f'(S_k) g''(d_{ik}) (x_k - v_i)(x_k - v_j)}{\sum_{k=1}^n a_k f'(S_k) g'(d_{ik})} \end{aligned}$$

If the GCM is I type learning algorithm, then it implies that $LS=TS$. In general cases, it is very difficult to prove that $LS=TS$ or $LS \neq TS$. We just give the proof in a special case below.

Noticing that the condition $v = \theta(v)$ is equivalent to $\forall i, \frac{\partial Q}{\partial v_i} = 0$, we will choose a specific point v satisfying $v = \theta(v)$ to check whether $v \in TS$ and $v \notin LS$.

It can be proved that $\forall i, v_i = \bar{x}_a = \frac{\sum_{k=1}^n a_k x_k}{\sum_{k=1}^n a_k}$ satisfying $v = \theta(v)$ or $\forall i, \frac{\partial Q}{\partial v_i} = 0$. And $\forall i, v_i = \bar{x}_a$ is a

strict local minimum of Q if $\left[\frac{\partial Q^2}{\partial v_i \partial v_j} \right]_{c \times c} \Big|_{\forall i, v_i = \bar{x}_a}$ is a posi-

tive matrix. It follows that $\lambda_{\max}(C_{\bar{x}_a}^g)$ is less than 1, where $\lambda_{\max}(C_{\bar{x}_a}^g)$ be the maximum eigenvalue of the matrix $C_{\bar{x}_a}^g = -\sum_{k=1}^n \frac{2a_k g''(d_{k\bar{x}_a})}{g'(d_{k\bar{x}_a}) \sum_{k=1}^n a_k} (x_k - \bar{x}_a)(x_k - \bar{x}_a)^T$ by analysis of

Hessian matrix, more details can be seen in (Yu, 2005). In particular, we can prove that $\forall x, f''(x) < 0, \forall i, v_i = \bar{x}_a$ is a strict local minimum of Q, it follows that $\forall i, v_i = \bar{x}_a$ belongs to TS when $\forall x, f''(x) < 0$.

By stability analysis of dynamical system, we know that \bar{x}_a is a stable fixed point of the dynamical system described by (3) if the modular of $\lambda(C_{\bar{x}_a}^g)$ is less than 1, where $\lambda(C_{\bar{x}_a}^g)$ be any eigenvalue of the matrix $C_{\bar{x}_a}^g$. The above analysis shows that when $\forall x, f''(x) < 0$ and the

modular of $\lambda(C_{\bar{x}_a}^g)$ is greater than 1, it is impossible for the GCM to output \bar{x}_a , i.e. \bar{x}_a is not in LS. Moreover, the above analysis is helpful for parameter selection in GCM.

In this way, we have not only proved that $LS \neq TS$ under some condition, but also show that GCM cannot output the original expected result under some condition, although a general condition for $LS = TS$ ($TS \subseteq LS, TS \supseteq LS$, or $LS \neq TS$) may be a challenging but interesting question as for this learning problem.

Example 1 clearly shows that GTL indeed occurs in machine learning field.

When the model truly reflects the given data, it is a reasonable hope that a learning algorithm is type-I. Such condition is not easy to be satisfied. In history, too many efforts have been devoted to solving this issue. Sometimes, a model can only be designed by intuition, has almost nothing to do with the given data. In [12], a novel clustering model (ACE) is proposed as follows:

The update equation for cluster centers can be randomly defined as $v^{(t)} = \theta(v^{(t-1)})$, where $\theta(\cdot)$ is a map by user.

According to GTL theory, the model of ACE can be defined as $v = \theta(v)$, the learned set with respect to ACE is defined as that in GCM. In this framework, the teachable set can be considered the same as the learned set as for ACE. Therefore, ACE is type-I. However, it can not be evaluated as a good clustering algorithm. In the following, we prove that ACE can not work as a clustering in extreme case.

For brevity, we set $\bar{x} = (\sum_{k=1}^n x_k)/n, \forall k, a_k = 1, \forall i, \alpha_i = 1/c, g(x) = x, h(x) = \cos(x/c), r(x) = x$, and the update equation for cluster prototype is defined as follows:

$$v_i = \frac{\sum_{k=1}^n \cos\left(d_{ik} / \sum_{w=1}^c d_{wk}\right) x_k}{\sum_{k=1}^n \cos\left(d_{ik} / \sum_{w=1}^c d_{wk}\right)} \quad (4)$$

Therefore, we know the Jacobian matrix of (5) is as follows:

$$\left. \frac{\partial \theta_i}{\partial v_j} \right|_{\forall i, v_i = \bar{x}} = -2 \left(\frac{1}{c} - \delta_{ij} \right) t g \left(\frac{1}{c} \right) \sum_{k=1}^n \frac{(x_k - \bar{x})(x_k - \bar{x})^T}{n \|x_k - \bar{x}\|^2} \Big|_{x=d_{ik}} \quad (5)$$

In order to compute the eigenvalues of $\left. \frac{\partial \theta_i}{\partial v_j} \right|_{\forall i, v_i = \bar{x}}$, we need prove the following Lemma 1.

Lemma 1: if $G = \begin{bmatrix} (1-\alpha_1)H & -\alpha_1H & \cdots & -\alpha_1H \\ -\alpha_2H & (1-\alpha_2)H & \cdots & -\alpha_2H \\ \cdots & \cdots & \cdots & \cdots \\ -\alpha_cH & -\alpha_cH & \cdots & (1-\alpha_c)H \end{bmatrix}$

where $\sum_{i=1}^c \alpha_i = 1, \forall i, \alpha_i \geq 0$, H is a $s \times s$ matrix, $c > 1$, then $|G - \lambda I| = \lambda^s |H - \lambda I|^{c-1}$.

Proof. As $\sum_{i=1}^c \alpha_i = 1, \forall i, \alpha_i \geq 0$, if we add all the block rows together to make the first row for the determinant, we have

$$|G - \lambda I| = \begin{vmatrix} \lambda I & \lambda I & \cdots & \lambda I \\ -\alpha_2H & (1-\alpha_2)H & \cdots & -\alpha_2H \\ \cdots & \cdots & \cdots & \cdots \\ -\alpha_cH & -\alpha_cH & \cdots & (1-\alpha_c)H \end{vmatrix}$$

Then, by using the first row to triangularize the block matrix, we have

$$|G - \lambda I| = \begin{vmatrix} \lambda I & \lambda I & \cdots & 0 \\ 0 & H - \lambda I & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & H - \lambda I \end{vmatrix} = |\lambda I_{s \times s}| \cdot |H - \lambda I|^{c-1} = \lambda^s |H - \lambda I|^{c-1}$$

Thus, the proof is completed.

Set $H = \sum_{k=1}^n \frac{(x_k - \bar{x})(x_k - \bar{x})^T}{n \|x_k - \bar{x}\|^2}$. As the trace of H is one, it

is easy to prove that all the eigenvalues of H is positive and not greater than 1. Hence, we know that all the

modular of the eigenvalues of $\left. \frac{\partial \theta_i}{\partial v_j} \right|_{\forall i, v_j = \bar{x}}$ (13) is less

than $2tg\left(\frac{1}{c}\right)$ by Lemma 1.

Consider that $2tg\left(\frac{1}{c}\right) < 1$ when $c > 2$, we can draw a conclusion that all cluster centers outputted by the clustering algorithm induced by (5) will be stably coincided into one point based on the stability analysis of dynamic system, no matter what the given data is. A good clustering algorithm is never expected to output coincided cluster centers for any given data. Therefore, we clearly illustrate that a type-I learning algorithm maybe have a worse performance if the model is designed without properly reflecting the nature of a specific learning task. Speaking accurately, it is meaningless to lessen the requirements of designing a proper model with respect to a specific learning task, in order to seek a type-I learning algorithm.

Certainly, if the model perfectly reflects the nature of

the given data, type-I learning algorithm has the best performance. In the following, we will use perceptron to show this conclusion.

Example 2. Perceptron. The training set is $\{(x_k, y_k), 1 \leq k \leq n, x_k \in R^s, y_k \in \{-1, 1\}\}$, we hope to find a straight line $y = w \bullet x + b$ to separate the points in the training set.

In mathematical language, we seek $y = w \bullet x + b$ such that $y_i(w \bullet x_i + b) \geq 1$. The learning algorithm to solve the above problem is called perceptron algorithm, which can be seen in any standard textbook of pattern recognition.

It has proved that perceptron do converge when the training set are linear separable. Therefore, it is easy to prove that perceptrons is type-I learning algorithm, given that training set is linear separable. In other words, perceptrons is I-type learning algorithm with respect to the linear separable training set. If the training set is not linear separable, perceptrons does not converge. Therefore, perceptrons is not I-type learning algorithm. If we can transform the given data to linear separable, perceptrons is still available. This is just one starting point for developing kernel methods.

From the above analysis, we know that GTL theory indeed gives a novel insightful into learning theory. And GTL provides an approach to studying learning algorithms. The main goal of GTL theory is to bridge the gap between the model and its corresponding learning algorithm. However, if a model cannot well represent the given data, it has a little value to design a type-I learning algorithm is type-I. Therefore, all four components of machine learning should be well chose.

4. Discussions

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

In this paper, we propose a novel approach to studying learning algorithms-GTL theory. Usually, to solve a learning problem includes four components: data space, hypothesis space, model and learning algorithm. Different from other learning theories, GTL theory focuses on studying the relation between models and learning algorithms.

We have to prove that not all expectations of a model can be realized by its corresponding learning algorithm in this paper, and plan to prove that not all stable outputs of a learning algorithm can be included in the expecta-

tion of its corresponding algorithm. Speaking clearly, we have proved that GTL indeed occurs in machine learning.

According to GTL theory, learning algorithms are categorized into four types. When the model is well designed, a type-I learning algorithm is the most expected for a specific learning task. The reason is as follows: other than type-I learning algorithm can not be easier controlled and evaluated than a type-I learning algorithm.

Dating back to history, people always hope to find an exact solution for any scientific problem (including learning problem). For instance, we hope that we can obtain a generic formula to calculate the root of any polynomial equation. In 18 century, it is proved that such formula does not exist. Therefore, people devoted to finding a way to numerically approximate the accurate solution, which is called algorithm. Obviously, the best hope is that the algorithm converges to the exact solution. When the model is not too complex, such hope can be realized. With the complexity of the model increasing, the probability of realizing such hope becomes more and more small. In the literature, it has been proved that it is impossible to design an algorithm to find an accurate solution of any NP hard problem within polynomial time. In fact, type-I learning algorithm is a representative of finding the exact solution.

As noted in Section 3, a type-I algorithm perhaps has a little practical value if the model is not properly designed. When the model is correctly designed, sometimes other than type-I learning algorithm is acceptable in practice. For example, it is easy to guess that learning algorithm corresponding to neural networks usually is not type-I although it is a very challenging task to prove whether a learning algorithm is type-I or not. In the near future, we will prove that the decision tree, the K-nearest neighbor method are type-I. As for neural networks or evolutionary computation, it is very easy to define the teachable set and very challenging to give a proper criterion on determining the learned set.

The results in Section 3 also shows that GTL can help us to get a deep insight into the specific learning algorithm, sometimes can offer some hints on parameter selection.

If we return to human learning, a student is considered as the most wanted by the teacher only when he has learned the more than that expected by the teacher. Generally speaking, the more he has learned than that expected by the teacher, the more excellent he can be considered. According to GTL theory, such a student is type-III. If a student can only learn the same as that expected by his teacher (similarly, such a student is type-I), he is considered a good one but not the most excellent.

As for machine learning, we have different views than human learning. We never think that type-III is better than type-I in machine learning. Roughly speaking, a teacher always thinks that a type-III student is better than a type-I student just because the teacher can learn something new from a type-III student. In other words, it is very useful that the information between teacher and student exchanges from each other in human learning process. In machine learning, the transfer of information from a model to its learning algorithm is a single way, model and learning algorithm lie in different control levels. When the output of a learning algorithm does not lie in the teachable set defined by its corresponding model, it is evaluated as not a breakthrough but a failure. Therefore, the best performance of a learning algorithm never surpasses that defined by the model, and a student can surpass his teacher's expectation in learning processing. Consequently, it is not reasonable to expect the learning ability of a machine to surpass human being. The above analysis support the following intuition: artificial intelligence can never surpass human intelligence from the learning point of view.

Acknowledgment

The work is partially supported by the Fok Ying Tung Education Foundation under Grant No. 101068, Program for New Century Excellent Talents in University in 2006, Grant NCET-06-0078, The Special Research Fund of Doctoral Program of Higher Education of China under Grant 20050004008, 973 Program under Grant No. 2006CB303105, 2007CB311002.

References

- [1] Blum, A. & Rivest, R.L. (1989). *Training a 3-node neural net is NP-Complete*. In *Advances in Neural Information Processing Systems I*, pages 494-501. Morgan Kaufmann.
- [2] Bousquet, O. & Elisseeff, A. (2001). Stability and generalization. *Journal of Machine Learning Research*, 2, 499-526.
- [3] Cucker, F. & Smale, S. (2001). The mathematical foundations of learning. *Bulletin of American Mathematical Society*, 39, 1-49
- [4] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematical Control Signals Systems*, 2, 303-314
- [5] Dietterich, T. G. (1995). Overfitting and under-computing in machine learning. *Computing Surveys*, 27(3), 326-327.

- [6] Evgeniou, T., Pontil, M. & Elisseeff, A. (2004) Leave one out error, stability, and generalization of voting combinations of classifiers. *Machine Learning*, 55(1), 71-97.
- [7] Freund, Y. & Schapire, R. (1997). A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences*, 55, 119-139
- [8] Minsky, M. L., & Papert S. A. (1969). *Perceptrons*, MIT
- [9] Mitchell, T. (1997). *Machine learning*. New York: McGraw-Hill.
- [10] Poggio, T., Rifkin, R., Mukherjee, S., & Niyogi, P. (2004). General conditions for predictivity in learning theory. *Nature*, 428, 419-422.
- [11] Quinlan, J.R. & Rivest, R.L. (1989). Inferring decision trees using the minimum Description Length Principle. *Information and Computation*, 80(3), 227-248.
- [12] Runkler, T. A. and Bezdek, J. C. (1999). Alternating Cluster Estimation: A New Tool for Clustering and Function Approximation. *IEEE Transactions on Fuzzy Systems*, 7(4), 377-393.
- [13] Vapnik, V.N. (1998). *Statistical learning theory*. New York: Wiley.
- [14] Yu, J. (2005). General c-means clustering model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, in press.
- [15] Zhou, D. (2002). The covering number in learning theory. *Journal of Complexity*. 18, 739-767.



Jian Yu received the B.S. degree in Applied Mathematics, M.S. degree in Mathematics, and Ph.D. degree in Applied Mathematics from Peking University, Beijing, P.R.China, in 1991, 1994 and 2000 respectively. At present, he is a full Professor and Head of Dept. of Computer Science, Beijing Jiaotong University. His current research interests include fuzzy systems, machine learning, pattern recognition, etc.