2010 International Conference on Technologies and Applications of Artificial Intelligence

Adaptive Knowledge Transfer Based on Locally Weighted Learning

Lei Han, Jianying Wu, Ping Gu, Kunqing Xie,
Guojie Song, Shiwei Tang, Dongqing Yang, Bingli Jiao
Key Laboratory of Machine Perception
(Ministry of Education)
Peking University
Beijing, China
Email: gjsong@pku.edu.cn

Feng Gao
Software school Fudan University
Fudan University
Shanghai, China
Email: gaofeng@fudan.edu.cn

***Abstract*—Locally weighted learning (LWL), which is an effectual and flexible method for prediction problems, is widely used in many regression scenarios. The training data samples, referring to the history experience knowledge base, are required to help do regression by new queries. However, sometimes, the knowledge base tends to be helpless due to the lake of information, such as inadequate training data. In such cases, traditional locally weighted learning will be powerless due to less history or inappropriate experience if there are not an adaptive mechanism or other learning methods like knowledge transfer to assistant. In this paper, we propose an adaptive transfer learning mechanism to assistant LWL to do prediction. As there are many auxiliary training sets, we assign different optimal local models to take each training set as the learning basic, and combine those models into an integrated one adaptively to give the final prediction value by allocating weights for each model dynamically with the feedback prediction error. Importantly, this learning process is assigned for multi-domain knowledge bases transference and multi-locally-weighted-model integration. Moreover, we also give an analysis about how the selection of additional training domains affects the regression result. Experimental studies are based on climate data which contains the monthly average of global land air temperature from 1901 to 2002 on grids divided by 0.5 latitude and 0.5 longitude. Knowledge transfer is taken out from neighbor grids to a center. The results show that our mechanism performs much better than traditional LWL.**

***Keywords*-Locally Weighted Learning; Transfer Learning; Adaptive; Multi-model;**

I. INTRODUCTION

We focus on transfer learning when the original source domain is unsatisfied with the demand of calculations in test domain, and the learning process needs more knowledge from additional training dataset that are different but related with the test domain. The motivation is obvious that when we face with a new challengeable difficult, we first attempt to gain help from the similar scenario that have already been solved in our experience, which is actually attributed to the artificial intelligent learning mechanism. This knowledge transfer conception has already taken place in clustering and labeling where transfer learning begins to show its superiority [1][2].

As the same with the conception above, we transfer this transfer learning from clustering to lazy learning for local regression. Lazy learning is a learning method in which generalization beyond the training data is delayed until a query comes as opposed to in eager learning where the system tries to generalize the training data before receiving queries, and wins the favor in some learning problems such as prediction. A typical lazy learning method is locally weighted learning (LWL) which contains some flexible parts such as parameter selection. Widely used parameter selection methods are, for example, Global Bandwidth Selection(GBS), providing a global optimal bandwidth to show its simplicity and universality, and Query-based Bandwidth Selection(QBS), using a bandwidth associated with each query point that will allow rapid or asymmetric changes in the behavior of the data. For lazy learning, the training set is the most important part because such lazy behavior completely depends on the abundant memory as a strong supporter. Unfortunately, in some cases, this knowledge memory is not that strong and even poor to support lazy learning due to the lack of information such as inadequate training data and new queries. Especially, traditional lazy learning method work well only when the training and testing data are drawn from a same data source and feature space. Once there comes a new query and the training memory is accumulated from similar but different data source, the learning model has to be rebuilt which is expensive or even impossible. In such cases, traditional locally weighted learning will lose its ability and some adaptive strategies should be introduced to assistant regression. Some related work in [3][4] focus on adaption to make LWL more robust, but those mechanisms talk about how to select optimal model parameters on the premise that there are abundant helpful training samples.

To overcome those difficulties described above, transfer learning should be introduced to LWL to help knowledge transfer from similar source domains to the test domain. In this paper, we propose an adaptive transfer learning mechanism based on locally weighted learning. For the lack of training samples, we take the similar training domains, which are abundant for normal local regression, as the

978-0-7695-4253-9/10 $26.00 © 2010 IEEE
DOI 10.1109/TAAI.2010.69

auxiliary knowledge. We notice that there are usually many different local models available such as constant, linear or quadratic model. Each model has their own advantages in different learning scenarios, and it is definite that no single model can perform well globally. Especially, even for the same type of the models, dissimilar parameters will lead to significant differences in regression results [3][4]. This makes different optimal local models suit for different training domains. Under a transformation background, we hope that all of the models can play its advantages in their compatible regions. A directly way is that we can create optimal models for each training domain with the same query come from the test domain and mix all the models together by averaging weighted mechanism. Instead of a global selection to give each model a fixed weight before local regression, we use a local way to allocate the weights adaptively by tuning the weights with the regression error. Moreover, we pay much attention on how the selection of additional training domains will affect the final integrated model. In line with this philosophy, we take an experiment to test the effectiveness of the integrated model with the increasing number of additional training domains and draw an explicit analysis.

The main contributions of this paper are: (1) as different from former transfer learning application in discrete clustering or labeling, we combine this conception with lazy learning method for continuous regression and demonstrate its superiority in real climate dataset; (2) according with the transfer knowledge, we create multi-models for the knowledge domains and mix the models together; (3) by using the feedback regression errors, we allocate the weights of each model adaptively when we focus on a certain test domain; (4) we also give an analysis about how the selection of training domains effects the regression result.

The rest paragraphs are organized as follows. The second part gives preliminaries. Section three talks about the methodology of adaptive knowledge transfer via LWL. Experimental studies are implemented in the fourth part. The last section gives a conclusion.

II. Preliminaries

A. Locally Weighted Learning

The standard formation of locally weighted learning can be represented as:

$$\hat{y} = f(q) + \varepsilon \tag{1}$$

Where q is the input query, and $\hat{y}$ the prediction output of q. $f()$ is the regression model, with various types such as constant, linear and quadratic style. A kernel function, involved in f() and includes two parameters known as bandwidth h and number of neighbors K, calculates the weights of neighbors of input q.

As the complexity of the quadratic formation, we take constant model and linear model as a illustration. A constant model can be represented as an averaging weighted equation:

$$\hat{y} = \frac{\sum_i^K y_i \cdot G(\frac{xi,q}{h})}{\sum_i^K G(\frac{d(xi,q)}{h})} \tag{2}$$

where x_i and y_i are the sample input and output respectively. $G()$ is a kernel function. h is a fit parameter called bandwidth which controls neighborhood scale and can be selected by many methods such as Global Bandwidth Selection (GBS) and Query-based Bandwidth Selection (QBS) [5].

For a linear model,

$$Y = \beta X \tag{3}$$

the training inputs and outputs are formalized in input matrix X and output matrix Y. β is the parameter matrix here [6][7].

B. Transfer Learning

Knowledge transfer happens when there are a source domain with less training samples and a test domain with the same feature space and same distribution as the former. However, some other source domains with the similar but not same feature space are abundant. To assistant the learning work, knowledge should be transferred from the abundant similar source domain to the test domain. A transfer learning problem is often represented as:

Given a source domain D_S, and learning domain T_S, a test domain D_T and learning task T_T , transfer learning aims to help improve the learning of the target predictive function $f_T()$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$, or $T_S \neq T_T$ [8].

In many machine learning problems, the traditional leaning methods often face the difficulties described above. Beside the clustering and labelling problems, lazy learning often needs some assistant leaning mechanism to overcome such embarrassing situation similarly. Thus, we focus on the cooperation of transfer learning and LWL. A detailed discussion will be given in the following section.

III. Knowledge Transfer via LWL

To do knowledge transfer, we first need the training domains and test domain. We can draw this scheme in a 2D plate in Table. I. The center region D_c is considered as the combination of learning domain T_S and test domain D_T from where queries come. Due to the weak training samples in T_S, it is difficult to create a local model M_c to do accurate regression for queries from D_c without assistant knowledge base. Therefore, we need additional training domains having the similar features with the center domain D_c, which appear here as the neighbors D_i to D_j (the source domains D_S). To transfer the knowledge from the neighbor domains, we create optimal models for them to do regressions(the learning task T_T).

Without considering the knowledge transfer firstly, the regression of the focused center region D_c can be represented

Table I
KNOWLEDGE DOMAINS FOR TRANSFERRING

(D_i, M_i)	...	...
...	(D_c, M_c)	...
...	...	(D_j, M_j)

as:

$$\hat{y}_c^t = f(q, M_c) \tag{4}$$

where M_c is the optimal model created for D_c. Then, we assume that there are N neighbor regions are similar with the center D_c, taken a example from Table. I. Those regions are the assistant training domains. When we transfer the knowledge from neighbor regions to D_c for the same query q, optimal models will be created for each neighbor region. Then we can get N models based on different training domain and a same query. The regression model for each training domain can be therefore represented as:

$$\hat{y}_c^i = f(q, M_i, D_i) \tag{5}$$

Thus, the prediction error of each model is:

$$e_c = |\hat{y}_c - y_c| \tag{6}$$

$$e_i = |\hat{y}_c^i - y_c| \tag{7}$$

To combine all the models, $\hat{y}_c^t$ and each $\hat{y}_c^i$ have to be weighted and contribute to the final prediction as an averaging weighted formation:

$$\hat{y}_c = \frac{w_c \cdot \hat{y}_c^t + \sum w_i \cdot \hat{y}_c^i}{w_c + \sum w_i} \tag{8}$$

A. Adaptive Weights Allocation for Training Domains

Let w_c be the weight of model created on T_S and w_i be the weight of model created on D_i, we assign w_c and w_i with the prediction errors e_c and e_i dynamically. To explain the motivation, we consider a practical climate data background and aim to predict the temperature of a given local regions using knowledge transfer based on LWL. The region from where the queries come is considered as the test domain. Besides the training samples from the centre region, there must be several neighbor regions holding the similar climate features, and those regions will be treated as the training domains. We aim to transfer the neighbors' knowledge to the center, thus we allocate the weights of each model in those regions dynamically with each model's prediction error. As being geographic information, climate has its spatial-temporal feature that the changes over time or space are continues. That means the neighbor regions of the center or during narrow time duration, the climate features remain similar. So allocating weights with the prediction errors is not a lag to catch up with the changes, but an accommodation with the continuous variation. This is the essential motivation of the adaption.

Table II
VARIABLES DESCRIPTION

D_c	The centre domain
T_S	The learning domain with the same origin as queries, $T_S \subset D_c$
D_T	Test domain, $D_T \subset D_c, D_T \cup T_S = D_c$
$\mathbf{q}$	A vector denotes the given sequence of queries which come from D_T.
l	The length of $\mathbf{q}$.
q_k	The kth query in $\mathbf{q}$.
j	Number of selected neighbor domains.
$\hat{y}_c$	The output prediction result.
h	A parameter in the kernel function of a local model.
$[h_{min}, h_{max}]$	Adjustment interval for h.
$\boldsymbol{w}$	Weight vector for the neighbor domains.
w_i	An element in $\boldsymbol{w}$.
y_c^k	The real output of q_k.
$\hat{y}_c^t$	Prediction of the model created on T_S.
$\hat{y}_c^i$	Prediction of the model created on D_i.
e_c	Error of the model created on D_c^t.
$G()$	Kernel function, a Gaussian Kernel here $G(x) = 2.718^{-x}$.
w_c	Weight of D_c^t.
$\boldsymbol{e}$	Error vector for the neighbor domains.

Consequently, we take such adaption, and the weight of each model is proportional to the difference between its prediction result and real output:

$$w_c \propto |e_c| = |\hat{y}_c^t - y_c| \tag{9}$$

$$w_i \propto |e_i| = |\hat{y}_c^i - y_c| \tag{10}$$

where y_c is the real output. Incorporated into a kernel function, the weights can be outlined in equations:

$$w_c = G(e_c) \tag{11}$$

$$w_i = G(e_i) \tag{12}$$

where G() use a Gaussian Kernel $G(x) = e^{-x}$ here.

B. Algorithm Description

Table. II gives the explanation of the variables used in this paper. The main conception of the knowledge transfer is summarized in Algorithm 1. We first provide the regression with the queries from a centre domain D_c and the training domains including the centre training domain T_S and the neighbor domains $D_1, \cdots, D_j$. After the initialization of model set and parameters, we train the optimal model and parameters for each training domain. In the regression part, for each query, we first select an optimal model for T_S, and calculate its prediction error. Then, optimal models are selected for each neighbor domains and we can finally get

an error vector $\boldsymbol{e}$. By the end of each round of a query, the weight vector $\boldsymbol{w}$ will be updated by the error vector $\boldsymbol{e}$. Then the adjustment with the feedback errors will take effects in the next round of regression.

Importantly, how to choose the neighbor domains is necessary and very important. The choice of the assistant knowledge base affects the effectiveness of the integrated model significantly. To analyze the correlation between knowledge domain selection and the effectiveness, we change the number of neighbor domains increasingly and implement Algorithm 1 in pace with the changes. Results are shown in the following section.

Algorithm 1 : Knowledge Transfer via LWL

Input:

(1) A sequence of given queries $\boldsymbol{q} = [q_1, \ldots, q_l]$ form a centre domain D_c

(2) A centre training set T_S from D_c and j neighbor training sets $D_1, \cdots, D_j$

Output:

The output prediction vector $\hat{y}_c$ of the given queries.

Algorithm:

1. Initialization:

(1) Initial the model selection set with constant model, linear model and quadratic model.

(2) Given the model parameters a range to adjust such as Interval $[h_{min}, h_{max}]$ for bandwidth h.

(3) Assign the weight vector $\boldsymbol{w} = [w_1, , wj], w_1 = \cdots = w_j = 1$.

2. Training:

Train optimal model and parameters for each training set D_i and centre D_c.

3. Regression:

For each query q_k

- Do regression by selecting the optimal model and parameters for q_k using the centre training set T_S, and get $\hat{y}_c^t$.
- Calculate the prediction error $e_c = |y_c^k - \hat{y}_c^t|$, and the weight for T_S, $w_c = G(e_c)$.
- **For** each neighbor training set D_i
  - Select the optimal model for q_k with D_i and get $\hat{y}_c^i$.
  - Calculate the prediction error $e_i = |y_c^k - \hat{y}_c^i|$.
- Make the final prediction by weighting $\hat{y}_c^i$ with w_i,
  $$\hat{y}_c = (w_c \cdot \hat{y}_c^t + \sum w_i \cdot \hat{y}_c^i)/(w_c + \sum w_i).$$
- Adjust $\boldsymbol{w}$ by the current round of $\boldsymbol{e}$.
- **For** each w_i
  - Update the weights $w_c = G(e_c)$, $w_i = G(e_i)$.

4. Return.

IV. Experiments

A. Data Source

This section talks about experimental studies. The data used in the experiments are from Climatic Research Unit (CRU TS 2.0), Tyndall Centre [9]. It contains the monthly average of global land air temperature from 1901 to 2002 on grids divided by 0.5 latitude and 0.5 longitude. Thus, there are 102×12 samples in a grid. To construct a query sequence for testing, we choose the samples from 1991 to 2002 as the queries and others as training samples (a query will be added to the training set after its prediction has done). We take the grid, at where Beijing locates, as the focused centre domain (Latitude: $39°54'50''$N, Longitude: $116°23'30''$E). Grids nearby are selected as the neighbor domains.

To confirm the neighborhood, we first use KNN clustering method to generate a cluster with similar temperature for the centre domain. This step just gives a rough partition of the spatial neighborhoods which are clustered on the whole training samples. Thus, once clustered, the rough neighborhoods will be available for any queries from different grids within a certain time interval. The clusters have to be updated until the spatial feature changes considerably. The result of clustering is shown in Table. III where D_c is the centre domain, 1 denotes the neighborhood and 0 the irrelevant grids. Thus, we can select the training domains from those grids labeled 1.

Table III
SPATIAL GRIDS

0	0	0	0	0	0	0
0	0	0	0	0	0	1
0	0	0	0	**1**	1	1
0	0	1	**1**	D_c	**1**	1
0	0	1	1	**1**	1	1
1	1	1	1	1	1	1
0	1	1	1	1	1	0
0	0	0	0	0	0	0

B. Performance Evaluation

Instead of attempting to find the most beneficial grids, we select the nearest four from the cluster of centre domain, highlighted in Table. III, as the additional training domains to assistant the prediction. The reasons why we concentrated on these four grids are explained as following. Figure. 1 and Table. IV give the comparison of Knowledge Transfer via LWL (KTLWL), LWL with Global Bandwidth Selection (GBS) and LWL with Query-based Bandwidth Selection (QBS). The horizontal coordinates of Figure. 1 denote the testing years 1991$\sim$2002, and the longitudinal coordinates denote the mean error of all prediction errors in the corresponding year. Because the monthly average temperature of the centre region (Beijing, China) ranges from $-9°$C to $29°$C, it is difficult to use Mean Relative Error, especially when the actual value approximates zero. Thus, we directly use the Mean Absolute Error measured in Degree Celsius which is understandable.

Table IV
A COMPARISON OF PREDICTION MEAN ERROR OF ALL THE QUERIES

Methods	LWL with GBS	LWL with QBS	KTLWL
Mean Error of all the queries (Degree Celsius)	1.23	1.15	0.93

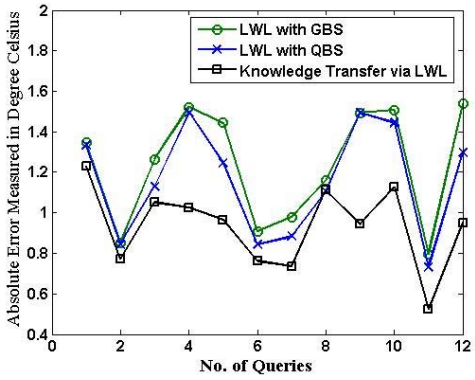

Figure 1. Comparison of Absolute Mean Errors.

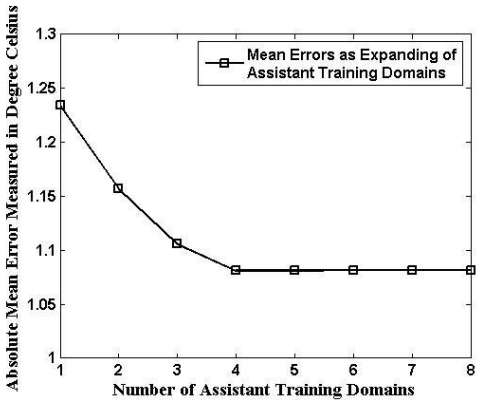

Figure 2. Relationship of Assistant Training Domain Selection and The Effectiveness of Integrated Model.

From the figure and table, we can easily find that knowledge transfer from neighbor domains to assistant LWL is necessary and this integrated model performs much better than the traditional locally weighted learning. However, it can also be found from Figure. 1 that at the 8th query, KTLWL does not take any advantages than the other two methods. This is mainly because the good quality of the learning domain T_S in D_c. In this case, the training samples in the centre domain have the ability to provide abundant helpful information that even if knowledge transfer does not take place, the regression with traditional LWL will also performs well. Actually, this scenario is not belong to the description of transfer learning.

Then, we come to explain the reason of the selection of the four nearest grids. We concern our methodology with a question that how the selection of additional training domains affects the integrated model. To catch up with this consideration, we take an experiment to detect the relationship between training domain selection and the effectiveness of the integrated model. Figure. 2 shows the relationship curve. We run the KTLWL on the increasing number of assistant training domains in the cluster, and $x = 1, 2, \cdots,$ at the horizontal coordinates means that the vertical value at x is an average prediction value of all possible combinations of the nearest x neighbor domains limited by the cluster.

From the curve we find that as the increasing of assistant training domains, the integrated model will firstly improve its performance. But when the assistant training domains increase to a certain scale, the integrated model will not take any more improvement and even get worse. The reason of this property can be explained by the local similarity. Due to the inadequate knowledge in the centre learning domain, with the increasing of nearest similar training domains, the helpful knowledge become richer thus lead to the performance improvement. But when the training domains continuously expand, the knowledge in the newly selected domains are helpless because the further the knowledge domain from the center, the less similar the features they share, even some bad knowledge will participate in the regression to interfere. So as the neighbor training domains expanding, the performance of the integrated model will first improve and finally keep stationary or even begin to get worse.

V. Conclusion

This paper proposes an adaptive knowledge transfer mechanism based on locally weighted learning. After the statement of the scenarios that some lazy learning method need knowledge transfer to assistant, a conception of combination of locally weighted learning and transfer learning is advanced. Then we concentrated on multi-domain knowledge transferring and create optimal models for each assistant training domain to finally get an integrated model. An adaptive updating mechanism is introduced into the combination of multiply models that we assign the weights of each model dynamically with the feedback regression errors. Importantly, we also give an analysis about the relationship between the training domain selection and the effectiveness of the integrated model and explain the phenomenon that as the increasing of training domains, the performance of the integrated model will first improve and finally keep stationary or even begin to get worse. Experimental studies are assigned to demonstrate the superiority and necessity of knowledge transfer in locally weighted learning and to analyze how the selection of additional training domains affects the integrated model. The results provide a convictive proof of our approach.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No. 60703066 and No. 60874082; Supported by Beijing Municipal Natural Science Foundation No. 4102026; supported by the National High Technology Research and Development Program of China (863 Key Program) No.2007AA120502.

REFERENCES

[1] J. Gao, W. Fan, J. Jiang and J. Han. "Knowledge Transfer via Multiple Model Local Structure Mapping," KDD08, August 24C27, 2008.

[2] W. Dai, Q. Yang, G. Xue and Y. Yu. "Boosting for Transfer Learning. Proceedings of the 24th international conference on machine learning," Corvallis: Oregon State University Press, 2007.

[3] Han Lei, Xie Kunqing, Song Guojie, "Adaptive Fit Parameters Tuning with Data Density Changes in Locally Weighted Learning," in International Symposium on Neural Networks, 2010.

[4] Shuai Meng, Han Lei, Xie Kunqing, Song Guojie, Ma Xiujun, Chen Guanhua, "An Adaptive Traffic Flow Prediction Mechanism Based on Locally Weighted Learning," ACTA SCIENTIARUM NATURALIUM UNIVERSITATIS PEKINENSIS, vol. 46(1), pp. 64-68, 2010.

[5] C. Atkeson, A. Moore, S. Schaal, "Locally Weighted Learning," Artificial Intelligence Review, 11–73(1997).

[6] S. Vijayakumar, A.D. Souza, S. Schaal, "Incremental Online Learning in High Dimensions," in Neural Computation, vol. 17. MIT Press, 2005.

[7] M. Shuai, K. Xie, W. Pu, G. Song, X. Ma. "An Online Approach Based on Locally Weighted Learning for Real Time Traffic Flow Prediction," The 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Irvine, CA, USA. November 5-7, 2008.

[8] S. Pan, Q. Yang. "A Survey on Transfer Learning," Hong Kong University of Science and Technology Press, 2008. http://www.cse.ust.hk/ sinnopan/SurveyTL.htm.

[9] Climate Dataset, available online at: http://www.cru.uea.ac.uk/