International Journal of Grid and Distributed Computing
Vol. 9, No. 12 (2016), pp.223-232
http://dx.doi.org/10.14257/ijgdc.2016.9.12.20

Research on Locally Weighted Linear Regression in Cloud Computing

Chuanying Lu

JiLin Communications Polytechnic, Changchun 130000, China
1963171234@qq.com

Abstract

Massive calculation tasks always show as a regular problem in the area of data mining. Many traditional data mining algorithms can only deal with small-scale input data and will run slower or even collapse when the input data increase. The problem above is always a bottleneck of traditional data mining algorithm. Better performance can be achieved if we can transplant these algorithms in cloud computing platform and make them run in parallel. Thus, whether the algorithm can be run in parallel properly or not becomes the key to solve the problem mentioned above. By analyzing the process of local linear regression algorithm, the bottleneck and the aspect which can be parallelized in these algorithms corresponding MapReduced algorithms are proposed, which handle the key problem of efficiency successfully. The research achievements gained in this paper provide a solution for MapReducing algorithms of data mining, and the experiment results show the effectiveness of the solution.

Keywords: *Sports Video, Cloud computing, MapReduce*

1. Introduction

Regression analysis comes firstly with single-element regression, then extends to multiple regression [1-8]. Local weighted linear regression algorithm has main ideas: fetch a number of local data; then match polynomial regression curve from local dataset; so it's possible to observe the rule and tendency that data show locally [9-15]. Common regression analysis creates model based on global data because it can describe the overall trend, but in reality, rules are not always of straight lines as described in the book[16-25]. We move forward the local range from left to right and finally one consecutive curved line is figured out. Obviously, the curve smoothness is closely associated with the selection of local data: local data is fewer, fitting is less smooth; conversely, fitting is smoother [26-35]. However the algorithm has a serious drawback. With more and more data waiting for regression, finding a certain quantity of local data from enormous data is a matter involving big amount of calculations. In order to solve the problem of high calculation of big dataset, we deploy the algorithm to implement in Hadoop cloud computing platform [36-40]. Experimental results show that if allocate pending tasks equally into several times of the number of Mapper in the cluster, and make Mapper end at the same time, the proposed algorithm can reach good effect of linear acceleration [41-44].

2. Local Weighted Linear Regression Algorithm

Robust locally weighted regression algorithm (LWLR) was proposed by Cleveland [45], utilizing local observational data to perform polynomial weighted fitting of points waiting for fitting. It integrates traditional local polynomial fitting, local weighted regression, very robust fitting process and stepwise regression algorithm, its speed faster

ISSN: 2005-4262 IJGDC

than general linear regression; besides, its operator is chosen progressively. LWLR algorithm is widely applied in statistics and forecast field. When data amount is big, LWLR algorithm's calculation is also big.

It has advantages like:

1. fast training speed;
2. learning complicated objective function;
3. information not easily lost;

It has shortcomings:

1. slow search speed;
2. easily mix with irrelevant attributes;
3. very big calculating amount, especially when the number of both user and item is huge, stand-alone machine requires several or more days to do calculations. Considering MapReduce frame allows user to disassemble a big problem into many small problems, let them solved through parallel operation in computer cluster, which helps improve greatly the operation efficiency; then deploy LWLR algorithm in MapReduce to run in Hadoop cloud computing platform.

Local weighted linear regression is a method based on internal memory. It does regression analysis of points close to samples in the training set. We'll investigate from the following parts:

1. For the calculated local data, do weighting of them in different ways and compare the effect;
2. Let LWLR algorithm MapReduce framed and deploy it to run in Hadoop cluster;
3. Compare performance difference of LWLR algorithm implemented in Hadoop MapReduce frame and single computer, and the relationship between the improvement of operational performance and DataNode number in Hadoop cluster.

3. Steps Included in the Locally Weighted Linear Regression Algorithm

LWLR is widely used in statistics and forecast field. LWLR algorithm's idea can be abstracted in these steps:

1. Calculate neighboring data points around predicted point from global data;
2. Do weighting processing of acquired local data points (through the most classical weighting pattern –Gaussian weighting model);
3. For data points after weighting, calculate regression coefficient and decide regression equation;
4. Make prediction calculation and do error checking of predicted data; determine regression effect to help user make decision.

3.1. Determine Neighboring Data Points

The first step of LWLR algorithm is to determine local data points, i.e. determine the nearest data points around the predicted point. An often used approach to determine local data points is KNN(k-Nearest Neighbor) algorithm [46], with the following ideas: compute the distance from predicted point to all data points in feature space; from them find certain point sets which are the nearest.

K-nearest neighboring classification algorithm is a theoretically mature method and one of the simplest machine learning algorithms. It has one assumption: if most of k most similar samples (the nearest neighbor) of one sample in feature space belong to a class, the sample belongs to the class. In KNN algorithm, selected neighbors are correctly classified objects. When making decision in classification, the algorithm decides its belonging category depending on one or more samples which are nearest to the sample awaiting classification. Theoretically KNN method relies on limit theorem; but in

classification decision-making, it's relevant with very few adjacent samples. Since KNN method determines pending sample's category by depending on limited and neighboring samples around instead of class field discriminant method. So for sample collection awaiting for classification whose class fields are intersected or heavily overlapping, KNN method works better than other classifying approaches.

However the algorithm is weak in classifying:

1. When samples are out of balance, if sample size of one class is very big and others' are small, it may have the problem: when a sample inputs, samples belonging to big-size class in the K neighbors are in the majority. In that case, we can take the method of adjusting weight to correct it, i.e. make bigger the weight of neighbors whose distance from the sample is small.

2. Next the algorithm requires big calculation amount, because it has to calculate the distance between pending sample and all known samples so as to get its K nearest neighboring points; a question of this kind can be solved rather by means of cloud computing platform.

The specific description of the KNN algorithm is as follows:

```
1. N =Φ
2. For each  d ∈T  Do  BEGIN
3. IF  | N|≤k  THEN
4. N =N ∪{u}
5. ELSE
6. IF  ∃u ∈N  such  that  sim(t,u) <sim(t, d)  THEN
7. BEGIN
8.  N =N −{u}
9. N =N ∪{d}
10. END
11.END
12.c =class  related to such u ∈N  which  has  the most  number ;
```

T is input the training data; K is the nearest number of samples; t is to be classified in the tuple; c is type of output.

Got k data points to calculate the distance between the test point to other points, assuming that any instance x with $x = \{a1(x), a2(x), ..., an(x)\}$ to describe. The distance of the two instances can be shown as a formula 1.

$$d = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2} \qquad (1)$$

d is distance between two points; $a_r(x_i)$ is related value of instance X in the dimension i, which is used to calculate the Euler distance between two instances.

3.2 Local Data Point Weighted Processing

By distance to do weighted calculation, the distance of the weighted function is shown in formula 2.

$$\hat{f} \leftarrow \arg\max \sum_{i=1}^{k} \omega_i \delta(v.f(x_i)) \qquad (2)$$

Where, ω_i shows the size of the weights calculated from the distance, it is shown in formula 3.

$$\omega_i = \frac{1}{d(x_q, x_i)^2} \qquad (3)$$

x_q is predicted point; x_i is neighboring point of x_q; calculate the reciprocal value of distance between them, which is the value of weight; sure we can utilize other ways to get it. It is shown in Figure1.

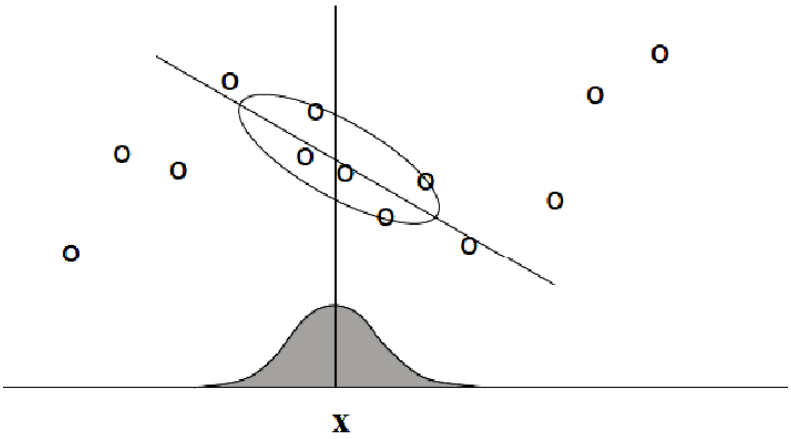

Figure 1. Euler Distance Local Model Map

In local weighted regression, point's weight is calculated by distance; regression calculates the cost with weighted points; point's weight can be obtained in various ways, of which the most often used is Gaussian model.

$$h_i \equiv h(x - x_i) = \exp(-k(x - x_i)^2) \qquad (4)$$

Where k is a smoothing parameter, determined by the following steps:

Make $n = \sum_i h_i$ to calculate the expectation and variance of the point:

$$\mu = \frac{\sum_i h_i x_i}{n} \qquad (5)$$

Formula 5 is used to calculate the average mathematical expectation of the surrounding neighborhood.

$$\sigma_x^2 = \frac{\sum_i h_i (x_i - \mu_x)^2}{n} \qquad (6)$$

The formula 6 is used to calculate the variance of the neighboring points.

$$\sigma_{xy} = \frac{\sum_i h_i (x_i - \mu_x)(y_i - \mu_y)}{n} \qquad (7)$$

The formula7 is used to calculate the covariance of the two points around the neighborhood.

$$\mu_y = \frac{\sum_i h_i y_i}{n}$$
$$\sigma_y^2 = \frac{\sum_i h_i (y_i - \mu_y)^2}{n} \qquad (8)$$
$$\sigma_{y|x} = \sigma_y^2 - \frac{\sigma_{xy}^2}{\sigma_x^2}$$

We use the covariance of the data to calculate and describe the expected and variance of the data, and the expectation is calculated as the formula 9:

$$\hat{y} = \mu_y + \frac{\sigma_{xy}}{\sigma_x^2}(x - \mu_x) \qquad (9)$$

Formula of variance is shown the formula 10.

$$\sigma_{\hat{y}}^2 = \frac{\sigma_{y|x}^2}{n^2}\left(\sum_i {}^2_{hi} + \frac{(x-\mu_x)}{\sigma_x^2}\sum_i {}^2_{hi} + \frac{(x_i-\mu_x)^2}{\sigma_x^2}\right) \quad (10)$$

As indicated from calculated results, the variance-based method can reach best result, i.e. value of smoothing coefficient k which can minimize $\sigma_{\hat{y}}^2$ around the reference point.

3.3. Determination of Linear Regression Function and Regression Coefficient

Regression function is shown in formula 11.

$$\hat{f}(x) = \beta_0 + \beta_1 a_1(x) + ... + \beta_n a_n(x) \quad (11)$$

β_0 Beta is the constant term of the regression, $\beta_1, \beta_2, .., \beta_{n-1}$ is the regression coefficient, $\hat{f}(x)$ is the predictive value of regression.

$$E_1(x_q) \equiv \frac{1}{2}\sum_{x \in k\_nearest\_nbrs\_of\_x_q} (f(x) - \hat{f}(x))^2 \quad (12)$$

Combined $E_1(x_q)$ and $E_2(x_q)$ were obtained.

$$E_2(x_q) \equiv \frac{1}{2}\sum_{x \in D} (f(x) - \hat{f}(x))^2 K(d(x_q, x)) \quad (13)$$

$$E_3(x_q) \equiv \frac{1}{2}\sum_{x \in k\_nearest\_nbrs\_of\_x_q} (f(x) - \hat{f}(x))^2 K(d(x_q, x)) \quad (14)$$

Weighted processing for each point:

$$\sum_{x \in k\_nearest\_nbrs\_of\_x_q} w_k (f(x) - \beta^T x_i)^2 \to \min \quad (15)$$

3.4. Prediction Calculation

First of all, find out neighbors around the point (using KNN algorithm for calculation; for acquired neighboring point collection, do weighting treatment with mixed Gaussian model based on near and far distance to get the weight of each point nearby; then do weighting with relative weight and each instance point to obtain the actual value of points after weighting; next substitute points after weighting into regression equation to have predicted value. It is shown in Figure2.

4. Implementation of Local Weighted Linear Regression Algorithm in MapReduce

Looking back at the introduction and analysis of LWLR algorithm in the above, we can find easily the algorithm has huge amount of calculation and it's very complex. When input data is tremendous, single machine running is extremely time-consuming. So, single computer processing enormous data with the algorithm is hardly implemented. Next we describe and analyze how to use LWLR algorithm to solve the bottleneck by advantage of cloud computing cluster after its implementation in MapReduce. LWLR algorithm is carried out in MapReduce in following steps:

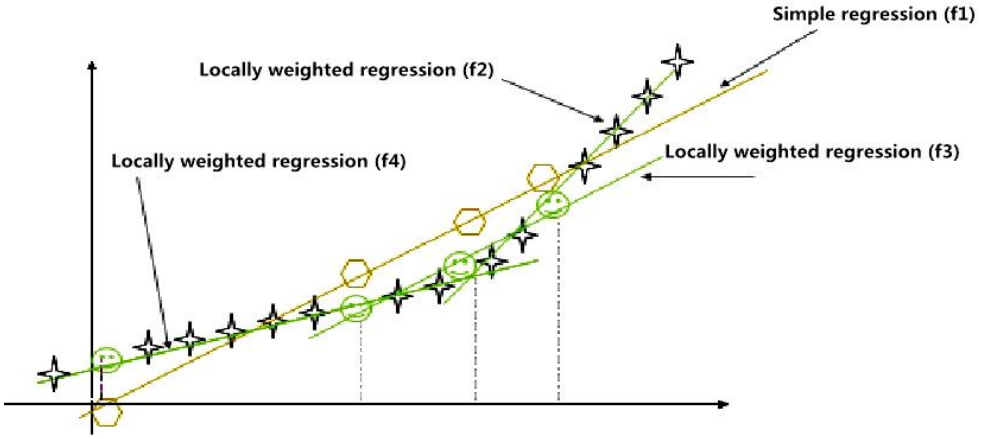

Figure 2. Predicted Calculation Example Diagram

4.1. Partition of Datanode

One of the core objectives of MapReduce frame is to maximize Datanode's effective calculation time and make it do operation processing with possibly more time, which is also the main principle of data division in this part. Most of running time of MapReduce should be spent in service computing, rather than in frequent initialization and system communication. In the case of limited network bandwidth, we need to reduce as much as possibly communication capacity between mappers. We need determine the size of optimal data blocks according to given network bandwidth. Only by doing that can effective calculation time been increased and network communication be decreased.

The practical implementation of data division has direct connection with network attribute, and platform operation work property. So far no quantitative formula can describe it but merely an empirical equation. Guided by the above principles, the size of partition blocks adjusted as per real effect, the experiment can achieve the best effect.

4.2. Map Stage

At this stage, system needs assign the number of Mapper for initialization for each DataNode according to algorithm's consumption of internal memory etc. First of all, Map function in Map class invokes distance to calculate modules, does distance calculation of data well partitioned in the above part, gets the distance from pending sample to each sample in the training sample set, and from it chooses K neighboring sample points which are closest to sample point; then, Map function calls calculation module to do weighting calculation of neighboring points to get weighted value of each adjacent point; finally, Map function calls matrix calculation module to calculate regression coefficient; with regression equation, it gets prediction value; then it does hypothesis testing, significance testing and calculation of confidence interval regarding testing data.

4.3. Reduce Stage

At this stage, post processing is done by data output format, no great significances. Without specific requirements, processing of output results can omit.

5. Experimental Analysis and Results

The experiment uses self-designed data, of which training sample set has 1000 samples and 100 classes; testing dataset size is respectively 300M and 580M, including thousands of testing samples. The dimension of the algorithm is open. The highest testing dimension

in the experiment is 20. But to make data be representative, we take 10-dimensional sample here. Testing sample data is shown in following mode:

1.2, 2.0 , 4.3, 2.1 , 6.0, 1.0, 9.2, 3.0, 7.3, 5.5, 8.1
10.0, 11.4, 35.6, 10.7, 15.8, 17.1, 20.2, 30.5, 19.1, 18.3, 12.7
……
119.0, 19.7, 124.6, 11.3, 251.0, 28.4, 36.1, 233.6, 79.5, 56.7, 9.8

Note: the training sample data is one more than the above data, and the last one is the sample size;

What's to be noted is: since the least square method is used, the number K of neighboring points cannot be too big or small; small K can lead to lower fitness; big K will cause the algorithm to global regression, losing the original significance.

We compare performance with same data dimension in single computer and cluster. In the cluster, DataNode counts 2-8 units in seven situations. In the experiment, to confirm cluster superior over single machine, with fixed DataNode and experimental dataset, we divide each kind of dataset in the way of conforming to data partition principle. Then get mean value of consumed time by different partitioning ways and use as running time of the cluster performing with current node number and experimental dataset. Compare the running time and that of single computer with the used dataset to evaluate superiority of cluster. We test with sample size 300M and 580M in Hadoop cluster, performance speed-up ratio shown in Figure 3.

From testing with different data dimensions in single machine and cluster, we note speed-up rate is linearly dependent. The point is when there's only one DataNode in single machine and cluster, cluster's performance may decline due to spending of communication compared with single machine.

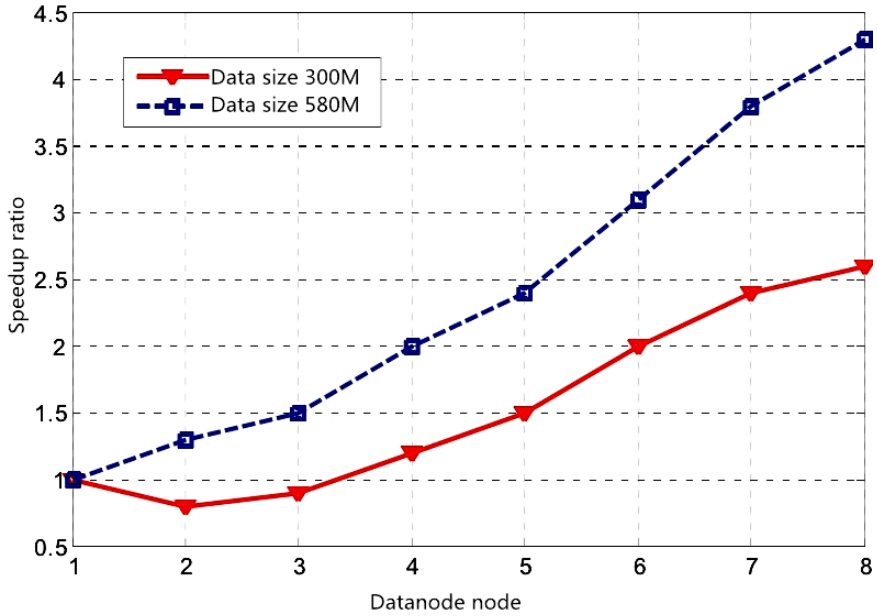

Figure 3. LWLR Algorithm Speedup Ratio Sample Map

Through comparisons, we see with growing data for processing, parallel computing frame works far faster than single machine. We also verified that to increase the efficiency of parallel computing system, it requires to improve its concurrency as much as possibly if system hardware allows.

The above analyses were made with simple data settings in the experiment. In the future, we need to investigate with more complicated data of higher dimension. Intensive studies will focus on statistical analysis and error checking of calculating data and accurate analysis of effect.

6. Conclusion

In this paper, the principle, realization process and main defects of local weighted linear regression algorithm are introduced.

A method of MapReduce is based on this algorithm is proposed, and the MapReduce of the algorithm is demonstrated by the cluster experiment, and the performance of the ideal speedup is obtained.

References

[1] G.-F. Feng, M. Wang, L. Li and M. Chen, "Test method for the performance of shared memory MapReduce cloud computing", Computer Engineering, vol. 38, no. 40106, **(2012)**, pp. 50-52.
[2] W.-M. Lang and D.-P. Yang, "MapReduce technology in cloud computing", Telecommunications express, no. 48903, **(2012)**, pp. 3-5.
[3] H. Chang, "MapReduce and load balance in cloud computing", Journal of Taiyuan University, vol. 13, no. 4901, **(2012)**, pp. 131-133.
[4] J. Ming, "Modified Mapreduce model for cloud computing. Computer measurement and control", vol. 20, no. 16405, **(2012)**, pp. 1417-1419.
[5] D.-P. Zhang and G.-Q. Tan, "Research on MapReduce algorithm based on Apriori in the cloud computing environment", Jiangxi communication science and technology, no. 11802, **(2012)**, pp. 16-19.
[6] Q.-H. Ceng and J.-B. Yuan, "A cloud computing model based on MapReduce and GPU dual parallel computing", Computer and digital engineering, vol. 41, no. 28103, **(2013)**, pp. 333-336.
[7] Y.-Y. Qiao, F. Liu, Y. Ling and J.-S. Yin, "Resource modeling and performance prediction of MapReduce in cloud computing environment", Journal of Beijing University of Posts and Telecommunications, vol. 37, no. 1, **(2014)**, pp. 115-119.
[8] L. Li, "Optimization Research of Apriori algorithm based on MapReduce in the cloud computing environment", Automation and instrumentation, no. 17707, **(2014)**, pp. 1-4.
[9] L.-M. Ding and J.-Y. Lv, "Study on the MapReduce processing method of big data in cloud computing", Internet of things, vol. 4, no. 4309, **(2014)**, pp. 86-88.
[10] W. Pan, Z.-H. Li, S. Wu and Q. Chen, "Research on MapReduce graph algorithm based on message passing mechanism", Journal of computer science, vol. 34, no. 34610, **(2011)**, pp. 1768-1784.
[11] W.-J. Jin and C.-Z. Wang, "An iterative MapReduce framework for evolutionary algorithms", Computer applications, vol. 33, no. 28012, **(2013)**, pp. 3591-3595.
[12] L. Yu, "Research on the performance of regression algorithm based on Hadoop platform", University of South China, **(2012)**.
[13] P.-Q. Zhang, "Research on the recommendation algorithm based on regression strategy", Beijing Jiaotong University, **(2014)**.
[14] G.-Q. Wang, "Evaluation of aging power equipment based on robust locally weighted regression algorithm", North China Electric Power University (Beijing), **(2009)**.
[15] F.-D. Zheng, "Support vector regression algorithm and its application research", Beijing University of Technology, **(2012)**.
[16] C.-G. Yuan, "Least squares support vector regression and its application in water quality prediction", Guangdong University of Technology, **(2012)**.
[17] L.-C. Su, X. Liu, X.-L. Li and W.-P. Feng, "An online analysis algorithm for lubricating oils based on locally weighted regression", Failure analysis and prevention, vol. 9, no. 3501, **(2014)**, pp. 6-10.
[18] C.-N. Yan, "Local optimization weighted regression algorithm in electric power equipment failure prediction application", Computer measurement and control, vol. 22, no. 18401, **(2014)**, pp. 279-281.
[19] Z.-H. Wang and P.-Q. Zhang, "A network regression algorithm based on iterated weighted linear model", Computer Engineering, vol. 40, no. 43906, **(2014)**,pp. 166-170.
[20] X.-L. Chen, S.-L. Li, J. Liu and Z.-M. Liu, "Based on the application of locally weighted decision tree algorithm in porosity prediction", Journal of Engineering Geophysics, vol. 1105, **(2014)**, pp. 736-742.
[21] L.-T. Jiang, G.-Z. Xu and L.-L. Zhou, "Software aging trend extraction based on strong local weighted regression algorithm", Journal of Shanghai Jiao Tong University, vol. 11, **(2006)**, pp. 1951-1954.
[22] L. Yu and J.-Y. Xiao, "Data mining robust locally weighted regression algorithm", Computer knowledge and technology, vol. 807, **(2012)**, pp. 1493-1495.
[23] L.-Y. Duan, W. Tian, M.-T. Xu and Y.-Z. Chen, "Medical image enhancement algorithm based on context quantization and local linear regression", Journal of Shanghai Dian Ji University, vol. 15, no. 9705, **(2012)**, pp. 325-331.

[24] N. Pan, X. Wu, Y. Liu, J.-S. Yang and Z. Yi, "Light, Linear traces of laser detection signal adaptive matching algorithm research", Chinese Journal of scientific instrument, vol. 3606, **(2015)**, pp. 1372-1380.
[25] C.-D. Xu, F. Yun and W.-W. Tong, "Weighted linear regression model in plateau mountainous area spatial precipitation interpolation research", Geo information science, no. 4701, **(2008)**, pp. 14-19.
[26] B.-Z. Zhao, S.-X. Zhang, Wang wind and rain, Z. Dong, "Short term forecasting of power load under massive data", Proceedings of the Chinese society of electrical engineering, **(2015)**.
[27] W. Quan, "Tumor motion in radiotherapy algorithm real-time tracking based on the prediction of respiratory", Southern Medical University, **(2015)**.
[28] W.-Z. Tan, "Basic theory and application of geographical weighted regression", Tongji University, **(2007)**.
[29] Xin, "Robust regression technique and its application in spectral analysis", Zhejiang University, **(2010)**.
[30] Z. Zhang, "The weighted technique of logistic regression and its application in the synthesis of mineral resources information", China University of Geosciences, **(2015)**.
[31] X.-Q. Yang, "System identification method for linear parameter variation", Harbin Institute of Technology, **(2014)**.
[32] Q. Li, "Parameter estimation and its application in the time weighted regression model", Lanzhou University of Technology, **(2014)**.
[33] Z. Qian, "Research on geographical weighted regression model of mean square error", Northeast Forestry University, **(2012)**.
[34] L. Yu, "Research on the performance of regression algorithm based on Hadoop platform", University of South China, **(2012)**.
[35] Y.-Z. Wu, "Research on normalization strategy of expression data for gene chip", Third Military Medical University, **(2004)**.
[36] X.-R. Chen, "Modeling and application of quantile regression in complex data", Yunnan University, **(2012)**.
[37] "Multi view measurement and regression learning methods and application research", Harbin Institute of Technology, **(2014)**.
[38] C.-D. Xu, "Spatial interpolation method of precipitation based on the linear weighted regression model", Henan University, **(2008)**.
[39] P.-Q. Zhang, "Research on the recommendation algorithm based on regression strategy", Beijing Jiaotong University, **(2014)**.
[40] Z.-Y. Duan, "Parameter estimation and statistical diagnosis of partial linear model with variable coefficients", Hunan University, **(2010)**.
[41] "R of Meiping geographically weighted generalized linear model based on the theory and application", Jinan University, **(2011)**.
[42] Q.-L. Huang, X.-Y. Tang, H.-X. Zhou, L. Qiao and X.-Q. Qiu, "Four kinds of spatial regression model in the influencing factors of the spatial data of the disease screening in the comparative study", Chinese health statistics, vol. 03, **(2013)**, pp. 334-338.
[43] H. J. Zhou, C.-L. Mei and C.-F. Wu, "Generalized linear model with variable coefficients and its estimation", System science and mathematics, vol. 01, **(2004)**, pp. 41-50.
[44] G. Gao, "Research on finger joint pattern recognition based on feature encoding and regression analysis', Nanjing University of Science and Technology, **(2014)**.
[45] W. S. Cleveland, "Robust locally weighted regression and smoothing scatterplots". Journal of Amer Stat Assoc, vol. 74, **(1979)**, pp. 829-836.
[46] D. Bremner, E. Demaine and J. Erickson, "Output-sensitive algorithms for computing nearest-neighbor decision boundaries", Discrete and Computational Geometry, vol. 33, no. 4, **(2005)**, pp. 593-604.

Author

Chuanying Lu. She got her M.S degree in Jilin University. She is an associate professor at JiLin Communications Polytechnic. Her research interests include computer application.