Influence of feature scaling on convergence of gradient iterative algorithm

Xing Wan
Leshan Vocational and Technical College, Leshan 614000, China
krantson@163.com

Abstract. Feature scaling is a method to unify self-variables or feature ranges in data. In data processing, it is usually used in data pre-processing. Because in the original data, the range of variables is very different. Feature scaling is a necessary step in the calculation of stochastic gradient descent. This paper takes the computer hardware data set maintained by UCI as an example, and compares the influence of normalization method and interval scaling method on the convergence of stochastic gradient descent by algorithm simulation. The result of study has a certain value on feature scaling.

1. Introduction
The gradient descent algorithm [3] is the core algorithm in linear regression. The gradient algorithm is used to feature the data before the iterative calculation to accelerate the convergence speed of the algorithm. There are usually two methods for feature scaling: normalization methods and interval scaling. This paper tests the effect of different feature scaling methods on the random gradient iteration rate based on the computer hardware dataset provided by UCI. For a single CPU, its performance is determined by multiple factors. UCI maintains a set of computer hardware data [7]. The data set provides 209 configuration data for different CPU-related metrics and performance. Each data has 10 attributes, including 7 performance-related numerical attributes: machine cycle time in nanoseconds (MYCT), minimum main memory in kilobytes (MMIN), maximum main memory in kilobytes (MMAX), cache memory in kilobytes (CACH), minimum channels in units (CHMIN), maximum channels in units (CHMAX), published relative performance (PRP). Using this data, you can have an accurate assessment of the performance of the chip. CPU data set is as follows:

Table 1 CPU data set

| No. | MYCT  | MMIN | MMAX  | CACH | CHMIN | CHMAX | PRP |
|-----|-------|------|-------|------|-------|-------|-----|
| 1   | 125   | 256  | 6000  | 256  | 16    | 128   | 198 |
| 2   | 29    | 8000 | 32000 | 32   | 8     | 32    | 269 |
| 3   | 29    | 8000 | 32000 | 32   | 8     | 32    | 220 |
| ... | ...   | ...  | ...   | ...  | ...   | ...   | ... |
| 209 | 480   | 1000 | 4000  | 0    | 0     | 0     | 45  |

In this article $D$ represents the number of features and $\lambda$ represents the number of data instances, $i$ represents ith number of data and $j$ represents jth feature value.
2. Multivariate Gradient Descent Method Estimation

2.1. Multivariate gradient descent

In regression analysis, if there are two or more independent variables, it is called multiple regression. In fact, a phenomenon is often associated with multiple factors. It is more effective and practical to predict or estimate dependent variables by the optimal combination of multiple independent variables than by using only one independent variable. Therefore, multivariate linear regression is more practical than a single variable linear regression. In multivariate linear regression, use $\mathbf{w}$ represents a parameter vector that needs to be estimated, where $\mathbf{w} = (w_0, w_1, \ldots, w_D)$, the linear regression polynomial can be expressed as:

$$h(x, \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_D x_D$$  \hspace{1cm} (2.1)

Assuming $x_0 = 1$ at this point, our calculation formula can be rewritten as a vector form:

$$h(x; \mathbf{w}) = \mathbf{w}^T \mathbf{x}$$  \hspace{1cm} (2.2)

If there are multiple sample data sets, they can be written as a matrix:

$$\hat{\mathbf{y}} = \mathbf{Xw}$$  \hspace{1cm} (2.3)

Among them $\hat{\mathbf{y}}$ is an estimate of the expected output. The variable gradient descent method is similar to the multivariate linear regression. It is necessary to construct a cost function, which is the sum of the squares of the difference between the predicted and actual values. The cost of the multivariate linear regression problem is as follows:

$$J(\mathbf{w}) = \frac{1}{2N} \sum_{j=1}^{N} (h(x^j; \mathbf{w}) - y^j)^2$$  \hspace{1cm} (2.4)

The optimization goal is to find the cost function $J(\mathbf{w})$. Minimum parameter $\mathbf{w}$ Multivariate linear regression algorithm can use formulas, the difference is that you need to find the cost function for multiple parameters. $J(\mathbf{w})$ Partial derivative

$$\frac{\delta}{\delta w_i} J(\mathbf{w}) = \frac{\delta}{\delta w_i} \frac{1}{2N} \sum_{j=1}^{N} (h(x^{(j)}; w_0, w_1, \ldots, w_D) - y^{(j)})^2, i = 0, 1, 2, \ldots, D$$  \hspace{1cm} (2.5)

When $i = 0$ the formula (2.5) can be rewritten as:

$$\frac{\delta}{\delta w_0} J(\mathbf{w}) = \frac{1}{N} \sum_{j=1}^{N} \left( h(x^{(j)}; \mathbf{w}) - y^{(j)} \right) x_0^{(j)}$$  \hspace{1cm} (2.6)

When $i > 0$ the formula (2.5) can be rewritten as:

$$\frac{\delta}{\delta w_j} J(\mathbf{w}) = \frac{1}{N} \sum_{j=1}^{N} \left( h(x^{(j)}; \mathbf{w}) - y^{(j)} \right) x_j^{(j)}$$  \hspace{1cm} (2.7)

Combine the above two cases of cost functions $J(\mathbf{w})$, the partial derivative can be expressed by the following formula:

$$\frac{\delta}{\delta \mathbf{w}} J(\mathbf{w}) = \frac{1}{N} \sum_{j=1}^{N} \left( h(x^{(j)}; \mathbf{w}) - y^{(j)} \right) \mathbf{x}_j^{(j)}$$  \hspace{1cm} (2.8)

Then the gradient descent algorithm can be written as follows:

$$w_i = w_i - \alpha \frac{1}{N} \sum_{j=1}^{N} \left( h(x^{(j)}; \mathbf{w}) - y^{(j)} \right) x_j^{(j)}$$  \hspace{1cm} (2.9)

Above formula can be rewritten as matrix form:

$$\mathbf{w} = \mathbf{w} - \alpha \frac{1}{N} \mathbf{x}^T (\mathbf{Xw} - \mathbf{y})$$  \hspace{1cm} (2.10)
The algorithm flow is as follows:

\[ \alpha_1 = -\frac{1}{N} X^T (X'w - y) \]  

(2.11)

By continuously iterating, the parameters are finally converged. In the actual operation, a termination condition is generally set, and the calculation is stopped as soon as the termination condition is satisfied.

2.2. Normalized processing

Data normalization is a basic work of data mining. Different evaluation indicators often have different dimensions, and the difference of numerical values may be very large. Without processing, the results of data analysis may be affected. In order to eliminate the influence of dimension and range differences between indicators, standardized processing is needed. The data is scaled to a specific area to facilitate comprehensive analysis. In multivariate linear regression, it is necessary to ensure that all features have similar ranges of values for better gradient descent performance. For the CPU data set, the minimum value of the CHMIN attribute is 0, the maximum value is 52, and the minimum value of the MMAX attribute is 64, and the maximum value is 64000. If left unprocessed, the gradient descent method takes a long time to converge. The solution to this problem is feature scaling, which converts data of different scales into the same scale. Common methods include standardization and interval methods. The premise of the normalization method is that the eigenvalues obey the normal distribution, and each genus is transformed into a standard positive distribution with a mean of 0 and a variance of 1 by translation and scaling data transformation. The standardized transformation formula is as follows:

\[ x'_i = \frac{x_i - \mu_i}{s_i} \]  

(2.12)

2.3. Interval scaling

The interval method utilizes the boundary information to scale the range of features to a range of features. For example, the commonly used interval scaling methods such as [0, 1] use two extreme values (maximum and minimum values) for scaling. The formula is expressed as:

\[ x'_i = \frac{2(x_i - \min(x_i))}{\max(x_i) - \min(x_i)} \]  

(2.13)

Among them, \( x_i \) is the ith transformed eigenvector, \( \max \) and \( \min \) are functions for finding the maximum and minimum, respectively.
In principle, as long as the learning rate is small enough, it can guarantee convergence to the local minimum, but it cannot guarantee convergence to the minimum. The value of the learning rate depends only on the speed at which the cost function converges to a minimum value, and does not depend on the magnitude of the minimum value. If the learning rate is too large, it is possible to skip the local minimum. There is no calculation formula for learning rate selection, which is closely related to specific models and data. Therefore, it is generally based on experience that the learning rate is selected by mistake.

It can be seen from the results of Fig. 1 that when the value of the feature component is large, a faster iteration rate can be obtained by the normalization method. This is because when the variance of a certain feature component is large, the convergence rate may be significantly reduced. Therefore, if the data is approximately positively distributed, the normalization method is recommended.

3. Conclusion
There is no standard process for data preprocessing, which is usually different for different tasks and data set attributes. The common processes of data preprocessing are regularization, interval scaling, data whitening and etc. Based on the results and discussions presented above, the conclusions are obtained as below:

1. In performing linear regression calculations, it is usually necessary to normalize the feature data to speed up the iteration.
2. When the feature data value is large, the normalization method can be used to obtain better convergence.
3. When the feature data value is small, the normalization method can be selected according to the specific situation.
4. Feature scaling can make machine learning algorithm work better and different machine learning algorithms can accept different range of input values.

Acknowledgments
This work was financially supported by the Applied Research Project of Education Department of Sichuan Province (18ZB0280).

References
[1] Allwein, E. L., R. E. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers.[J].Journal of Machine Learning Research, 2000,1:113-141.
[2] Cramer, K. and Y. Singer.. On the algorithmic implementation of multiclass kernel-based vector machines.[J].Journal of Machine Learning Research, 2001, 2:265-292.
[3] Dietterich, T.G. and G. Bakiri.. Solving multiclass learning problems via error-correcting ouput
codes. [J]. Journal of Artificial Intelligence Research, 1995, 2:263-286.

[4] Bishop, C. M. Pattern Recognition and Machine Learning. [M].Spring, NewYork, NY. 2006.

[5] Alpaydin, E. Introduction to Machine Learning. MIT Press, 2004.

[6] Carbonell, J. G., ed. Machine Learning: Paradigms and Methods. MIT Press. 1990.

[7] Computer Hardware Data Set. [EB/OL]. https://archive.ics.uci.edu/ml/datasets/

[8] A. Peled and A. Ruiz. Frequency domain data transmission using reduced computational complexity algorithms [C]. Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP-80, 1980, vol. 5: 964-967

[9] B. Muquet, Z. Wang, G. B. Giannakis, M. de Courville, and P. Duhamel. Cyclic prefixing or zero padding for wireless multicarrier transmissions [J]. IEEE Trans. Commun., vol. 50, no. 12: 2136–2148, Dec. 2002

[10] M. R. B. Shankar and K. V. S. Hari, Reduced Complexity Equalization Schemes for Zero padded OFDM Systems [J]. IEEE Signal Processing Letters, vol. 11, no. 9, September 2004

[11] Andrea Goldsmith. Wireless Communications [M]. Cambridge: Cambridge University Press, 2005. 8