ELSEVIER

Knowledge-Based Systems 15 (2002) 169–175

Knowledge-Based SYSTEMS

www.elsevier.com/locate/knosys

Clustered linear regression

Bertan Ari[a], H. Altay Güvenir[b,*]

[a]16021 NE 36th WAY, Redmond, WA 98052, USA
[b]Department of Computer Engineering, Bilkent University, Ankara 06533, Turkey

Received 1 November 2000; accepted 2 February 2001

Abstract

Clustered linear regression (CLR) is a new machine learning algorithm that improves the accuracy of classical linear regression by partitioning training space into subspaces. CLR makes some assumptions about the domain and the data set. Firstly, target value is assumed to be a function of feature values. Second assumption is that there are some linear approximations for this function in each subspace. Finally, there are enough training instances to determine subspaces and their linear approximations successfully. Tests indicate that if these approximations hold, CLR outperforms all other well-known machine-learning algorithms. Partitioning may continue until linear approximation fits all the instances in the training set — that generally occurs when the number of instances in the subspace is less than or equal to the number of features plus one. In other case, each new subspace will have a better fitting linear approximation. However, this will cause *over fitting* and gives less accurate results for the test instances. The stopping situation can be determined as no significant decrease or an increase in relative error. CLR uses a small portion of the training instances to determine the number of subspaces. The necessity of high number of training instances makes this algorithm suitable for data mining applications. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Clustering Linear regression; Machine learning algorithm; Eager approach

1. Introduction

Approximating the values of continuous functions is called *regression* and it is one of the main research issues in machine learning, while approximating the values of functions that have categorical values is called as *classification*. In that respect, classification is a subcategory of regression. Some researchers emphasized this relation by describing regression as 'learning how to classify among continuous classes' [12].

For both these problems, we have also two types of solutions: *eager learning* and *lazy learning*. In eager learning, models are constructed according to the given training instances in training part. Such methods can obtain the interpretation of the underlying data. Constructing models in training leads long training times for eager learning methods. On the other hand, in lazy learning methods, all the work is done during testing, so they require much longer test times. Lazy learning methods do not construct models by using training data, so they cannot enable interpretation of training data. CLR, an extension of *linear regression*, is an eager learning approach.

Although most of the real life applications are classification problem, there are also very important regression problems such as problems involving time-series. Regression techniques can also be applicable to the classification problems. For example, neural networks are often applied to classification problems [14]

$$\mathrm{Est}(b) = \frac{\mathrm{Covariance}(x, y)}{\mathrm{Variance}(x)}$$

The traditional approach for regression problem is the classical *linear least-squares regression*. This old, yet effective, method has been widely used in real-world applications. However, this simple method has deficiency of linear methods in general. Advances in computational technology bring us the advantage of using new sophisticated non-linear regression algorithms. Among eager learning regression systems, CART [4], RETIS [9], M5 [12] and DART/HYESS [7] induces regression trees; FORS [3] uses inductive logic programming for regression and RULE [14] induces regression rules, projection pursuit regression [6], neural network models and MARS [5] produces mathematical models. Among lazy learning methods, locally weighted regression (LWR) [2] produces local parametric

[*] Corresponding author. Tel.: +90-312-290-1252; fax: +90-312-266-4126.

E-mail addresses: bari@microsoft.com (B. Ari), guvenir@cs.bilkent.edu.tr (H.A. Güvenir).

PII: S0950-7051(01)00154-X

functions according to the query instances, and *k-NN* [1,10,11] algorithm is the most popular non-parametric instance-based approach for regression problems [13]. Regression by feature projections (RFP) method is an advanced *k-NN* method that uses feature projections based knowledge representation. This research uses the local weight and feature projection concepts and combines them with the traditional *k-NN* method. Using local weight with feature projection may cause losing the relation between the features; however, this new method eliminates the most common problems of regression, such as curse of dimensionality, dealing with missing feature values, robustness (dealing with noisy feature values), information loss because of disjoint partitioning of data, irrelevant features, computational complexity of test and training, missing local information at query positions and requirement for normalization.

CLR is an extension of *linear regression* algorithm. CLR approximates on the subspaces, and therefore, it can give accurate results for non-linear regression functions. Also, irrelevant features are eliminated easily. Robustness can be achieved by having large number of training instances. CLR can eliminate effects of noise as well.

2. Linear least-squares regression

Linear regression is the traditional approach for regression problems. There are two main classes of linear regression: *univariate linear regression* and *multivariate linear regression*.

2.1. Univariate linear regression

A set of data consisting of n series of x and y values is given, where x is the independent variable and y is the dependent variable. In other words, there is only one unique feature that is represented by x, and the target is represented by y. Assume that there is a linear relation between variable x and variable y:

$$y = bx + a$$

Here, b is the slope of the line, while a is the intercept at the y-axis. In reality, because of noise or mismatch between data and model, there is an error ε:

$$y_i = bx_i + a + \varepsilon_i$$

Finding a suitable model that minimizes the sum of squared errors for the given data set is the aim of *univariate linear regression*. Since *univariate linear regression* problem searches for a suitable model in the form of $y = bx + a$, a candidate slope, b and intercept a are chosen first. For each recorded (x, y) pair, square of $(y - bx - a)$, which is equal to square of e, is added to the total error. The line having the smallest total error is the best-fit line and so is the best model for univariate linear regression. The value of b can be estimated as follows:

$$b = \frac{\text{Covariance}(x, y)}{\text{Variance}(x)}$$

Note that if the variance of x is zero, then we cannot estimate b. This occurs when the x variable has the same value for all values of y. Once the value of b is determined, the value of a can be found easily.

2.2. Multivariate linear regression

Generally, the number of features in a data set is more than one. Finding a linear regression for data sets with more than one feature is called as *multivariate linear regression*. The model for multivariate linear regression can be represented in the index notation as follows:

$$y_i = a + b_1x_{1i} + b_2x_{2i} + b_3x_{3i}\ldots b_mx_{mi} + \varepsilon$$

By extending this representation, a system of n equations and m dependent variable can be shown as follows:

$$y_1 = a + b_1x_{11} + b_2x_{21} + b_3x_{31}\ldots b_mx_{m1} + \varepsilon_1$$

$$y_2 = a + b_1x_{12} + b_2x_{22} + b_3x_{32}\ldots b_mx_{m2} + \varepsilon_2$$

$$\vdots$$

$$y_i = a + b_1x_{1i} + b_2x_{2i} + b_3x_{3i}\ldots b_mx_{mi} + \varepsilon_i$$

$$\vdots$$

$$y_n = a + b_1x_{1n} + b_2x_{2n} + b_3x_{3n}\ldots b_mx_{mn} + \varepsilon_n$$

We can estimate the values of unknown variables b_m, ε_i and a if we have sufficient training data. Since there are $n + (m + 1)$ unknowns and fewer equations than unknowns, there is no unique solution to this system of equations. The *least-squares* solution minimizes the sum of squares of the errors. Linear algebra was developed to facilitate the solution of systems of linear equations. Following the conventions of linear algebra, the multivariate linear regression model can be rewritten as

$$\begin{matrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{matrix} = \begin{bmatrix} 1x_{11} & x_{12} & x_{13} & \cdots & x_{1m} \\ 1x_{21} & x_{22} & x_{23} & \cdots & x_{2m} \\ \vdots & & & & \vdots \\ 1x_{i1} & x_{i2} & x_{i3} & \cdots & x_{im} \\ \vdots & & & & \vdots \\ 1x_{n1} & x_{n2} & x_{n3} & \cdots & x_{nm} \end{bmatrix} \times \begin{bmatrix} a \\ b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_m \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_i \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

or more compactly:

$$\mathbf{y} = \mathbf{Xb} + \mathbf{e}$$

Then, we have a calculus problem to find a vector $\mathbf{b}$ that