ARTICLE IN PRESS
Information Sciences xxx (2014) xxx–xxx
Contents lists available at ScienceDirect
Information Sciences
journal homepage: www.elsevier.com/locate/ins

# Control of the minimum action reasoning
# Soft computing by multi dimension optic geometry

Germano Resconi

*Faculty of Mathematics and Physics, Catholic University, Brescia, Italy*

ARTICLE INFO

*Article history:*
Received 13 February 2009
Received in revised form 1 November 2013
Accepted 5 February 2014
Available online xxxx

*Keywords:*
Reasoning
Geodesic
Soft-computing
Extension of linear regression
Projection operator
Geometry of fuzzy reasoning

ABSTRACT

Given a $n+1$ dimensional space $S$ with $(y, x_1, x_2, \ldots, x_n) \in S$, we use the model (hyper-plane) in $S$ $y = \beta_1 f_1(x_1, x_2, \ldots, x_n) + \cdots + \beta_q f_q(x_1, x_2, \ldots, x_n)$ to control the transformation of the points $Y = \{(y_1, x_1, x_2, \ldots, x_n), (y_2, x_1, x_2, \ldots, x_n), \ldots, (y_k, x_1, x_2, \ldots, x_n)\}$ into $Y' = \{(y'_1, x_1, x_2, \ldots, x_n), (y'_2, x_1, x_2, \ldots, x_n), \ldots, (y'_k, x_1, x_2, \ldots, x_n)\}$ where the values $y'$ have the minimum distance from $y$. The algorithm to compute the parameters $(\beta_1, \beta_2, \ldots, \beta_q)$ and the values $y'$ is denoted minimum action reasoning. The operation is the geometric projection of $y$ into the hyper-plane in $S$. With the different models or hyper-planes we can control many different geometric transformations as reflection, rotation, refraction. With a chain of transformations we generate the minimum path in $S$ that joins one point to another (geodesic). One ray in the space $S$ is controlled by models as a special environment that guides the ray to have the task with minimum distance. The minimum action reasoning can be used to create software by models for different applications The coordinates of the space $S$ can be real numbers, logic values, fuzzy sets or any other set that we can define. Classical linear or non-linear regression is part of this minimum action reasoning. Also classical logic, many value logic and fuzzy logic are included in the minimum action reasoning.



## 1. Introduction

The paper studies the possibility to implement minimum action reasoning in a multidimensional space $S$. The first step of the paper is to create an algorithm to project a random set of points $y$ in $S$ into a set of points $y'$ that are in the best model within a family of models. The algorithm chooses among all models in the family the best model for which $y$ and $y'$ have the minimum distance. The action to choose the best model is named minimum action (geodesic). Any chain of minimum action is denoted reasoning and therefore the algorithm is denoted minimum action reasoning. More complex geometric transformations are possible as projection, reflection, rotation, refraction and so on in the space $S$. For a family of linear models we can use the geometric projection to compute the best linear parameters in the linear regression. We can also extend the linear regression to non-linear regression or to fuzzy number transformations. The space $S$ can be a space of logic values, as classical logic values, many value logic and fuzzy logic.

*E-mail address:* resconi@numerica.it

http://dx.doi.org/10.1016/j.ins.2014.02.015

## 2. Linear regression, projection operator, minimum action reasoning

Specialized literature on regression analysis (Gujarati 2003 [2]) and more generally on linear and non-linear models (Ryan 1997 [2]) offered many solutions to study the dependence between two types of variables $y$ and $\{x_1, x_2, \dots, x_p\}$ where $y$ is a quantitative dependent variable and $\{x_1, x_2, \dots, x_p\}$ are independent variables. Regression analysis studies the dependence of $y$ with respect to the variables $\{x_1, x_2, \dots, x_p\}$ when we have samples of $y$ and samples of $\{x_1, x_2, \dots, x_p\}$. This requires the choice of a suitable model and the related parameters estimation. Given the generic model:

$$y = f(x_1, \dots, x_p; \beta) + \varepsilon \tag{1}$$

the statistical regression aims to find the set of unknown parameters so that

$$\tilde{y} = f(x_1, x_2, \dots, x_p; \widetilde{\beta}) \tag{2}$$

where $\tilde{y}$ are the values of $y$ in agreement with the model and with the minimum errors respect to the given samples of y. The term $\varepsilon$ indicates the deviation of $y$ from the model. The most widely used regression model is the Multiple Linear Regression Model (MLRM), as well as the Least Squares (LS) is the most widespread estimation procedure. In the MLRM the dependent variable $y$ would be expressed as the weighted sum of the independent variables $\{x_1, x_2, \dots, x_p\}$, with the unknown parameters

$$\{\beta_1, \beta_2, \dots, \beta_p\}$$

Formally we have the hyper-plane

$$y_n = \beta_0 + \beta_1 x_{n,1} + \dots + \beta_p x_{n,p} + \varepsilon_n \tag{3}$$

where $\beta_0$ is the parameter related to intercept term. In a matrix form the model is expressed as:

$$y = X\beta + \varepsilon$$

where

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_q \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_p \end{bmatrix}, \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \dots \\ \varepsilon_q \end{bmatrix}, \quad X = \begin{bmatrix} 1 & x_{1,1} & \dots & x_{1,p} \\ 1 & x_{2,1} & \dots & x_{2,p} \\ \dots & \dots & \dots & \dots \\ 1 & x_{q,1} & \dots & x_{q,p} \end{bmatrix} \tag{4}$$

LS is based on the minimization of the sum of squared deviations:

$$\begin{aligned} &\min_{\beta} D = (y - X\beta)^T (y - X\beta) \\ &\textit{where } ()^T \textit{ is the matrix transpose} \end{aligned} \tag{5}$$

The optimal solution $\beta$ of the minimization problem is obtained in this way

$$D = (y - X\beta)^T (y - X\beta) = y^T y - y^T X\beta - (X\beta)^T y + (X\beta)^T (X\beta)$$

To compute the minimum value we make the derivatives of the previous form

$$\frac{\partial D}{\partial \beta_j} = -y^T X \frac{\partial \beta}{\partial \beta_j} - \frac{\partial \beta^T}{\partial \beta_j} X^T y + \frac{\partial \beta^T}{\partial \beta_j} X^T X \beta + \beta^T X^T X \frac{\partial \beta}{\partial \beta_j}$$

where

$$\beta = \begin{bmatrix} \beta_1 \\ \dots \\ \beta_j \\ \beta_{j+1} \\ \dots \\ \beta_p \end{bmatrix} \text{ and } \frac{\partial \beta}{\partial \beta_j} = \begin{bmatrix} 0 \\ \dots \\ 1 \\ 0 \\ \dots \\ 0 \end{bmatrix} = v_j, \quad \frac{\partial \beta^T}{\partial \beta_j} = [0 \quad \dots \quad 1 \quad 0 \quad \dots \quad 0] = v_j^T$$

We have

$$\frac{\partial D}{\partial \beta_j} = 0$$

for $y^T X v_j + v_j^T X^T y = v_j^T X^T X \beta + \beta^T X^T X v_j$

But because we have the following scalar property