Communications of the Korean Statistical Society Vol. 15, No. 3, 2008, pp. 343–351

Simple Graphs for Complex Prediction Functions

Myung-Hoe Huh[1], Yonggoo Lee[2]

Abstract

By supervised learning with p predictors, we frequently obtain a prediction function of the form $y = f(x_1, \ldots, x_p)$. When $p \geq 3$, it is not easy to understand the inner structure of f, except for the case the function is formulated as additive. In this study, we propose to use p simple graphs for visual understanding of complex prediction functions produced by several supervised learning engines such as LOESS, neural networks, support vector machines and random forests.

Keywords: Visualization; prediction function; LOESS; neural network model; support vector machine; random forest.

1. Introduction and Proposed Graphs

Suppose that we obtain a prediction function of the form $y = f(x_1, \ldots, x_p)$ as a byproduct of supervised learning with p predictors. It would be very valuable if we visualize the inner structure of f. Non-additive prediction functions with $p \geq 3$, however, are not directly visible since the manifolds are embedded in 4 or more dimensional Euclidean space. The aim of this study is to propose the use of p simple graphs, one for each predictor, for visual aids of complex prediction functions produced by several supervised learning methods such as LOESS, neural network models, support vector machines and random forests.

Suppose that we are given $y = f(x_1, \ldots, x_p)$, for $a_j \leq x_j \leq b_j$ $(j = 1, \ldots, p)$. To visually understand f, we propose the "conditional predictive graphs" of x_j versus y, one for each predictor.

Definition 1.1 We define a conditional predictive graph CPG(j) by the plot of trajectory curve

$$(x_j, f(x_j^0, \ldots, x_j, \ldots, x_p^0)), \quad \text{for all } x_j \text{ in } (a_j, b_j),$$

with fixed $\mathbf{x}_{(j)}^0 = (x_1^0, \ldots, x_{j-1}^0, x_{j+1}^0, \ldots, x_p^0)$. Thus, CPG($j$) varies as $\mathbf{x}_{(j)}^0$ assumes different values. Practically, we may draw CPG(j) for n observed or simulated cases of $\mathbf{x}_{(j)}^0$.

1) Professor, Departament of Statistics, Korea University, Anam-Dong 5-1, Sungbuk-Gu, Seoul 136-701, Korea. Correspondence: stat420@korea.ac.kr
2) Professor, Department of Statistics, Chung Ang University, Heukseok-Dong 221, Dongjak-Gu, Seoul 156-756, Korea. E-mail: leeyg@cau.ac.kr

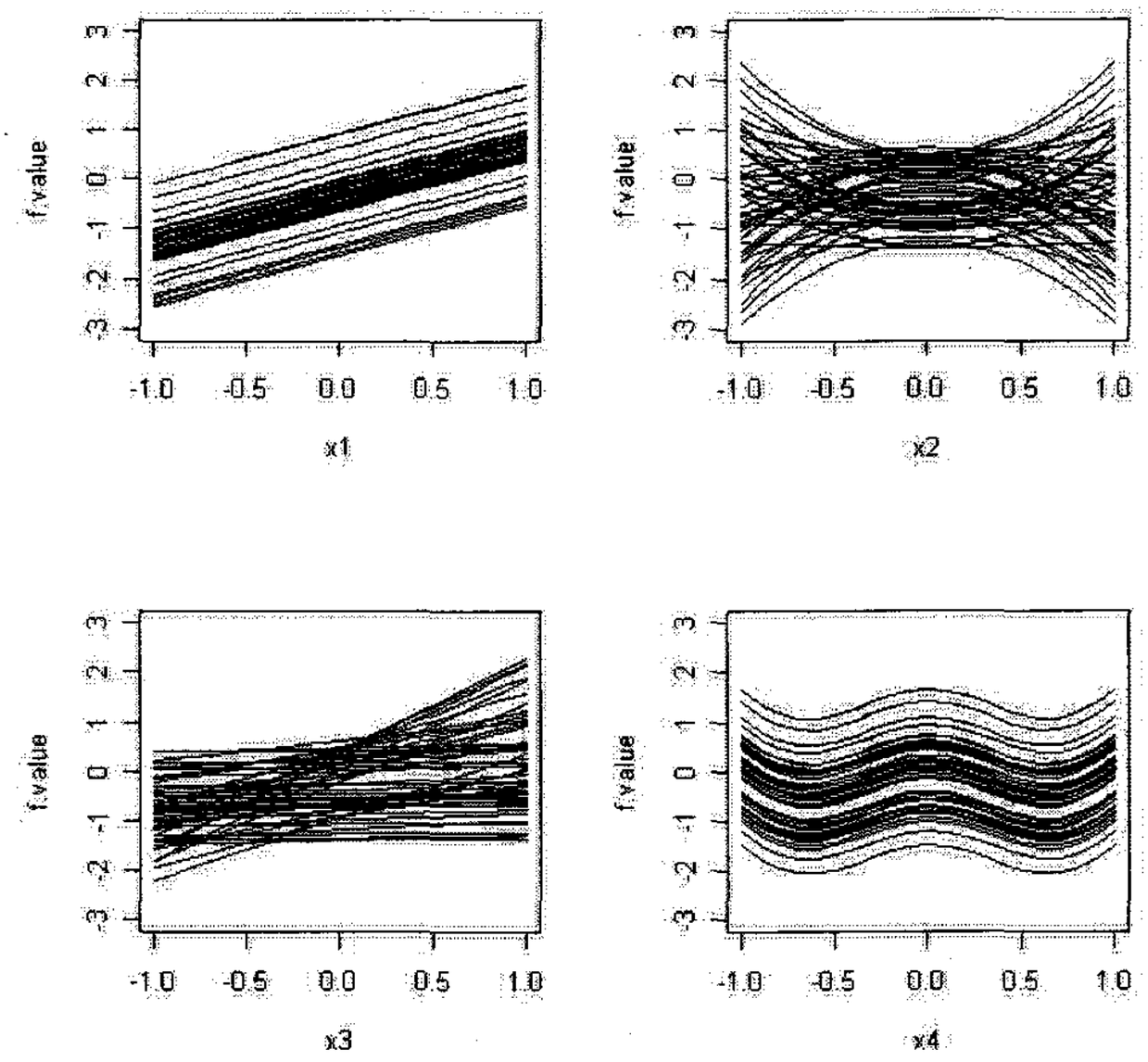

Figure 1.1: CPG's of $f(x_1, x_2, x_3, x_4) = x_1 + 2\,x_2^2\,x_3 - x_4\,\sin(\pi\,x_4)$

For the purpose of illustration of CPG's, we consider

$$f(x_1, x_2, x_3, x_4) = x_1 + 2\,x_2^2\,x_3 - x_4 \sin(\pi\,x_4), \quad -1 \le x_j \le 1 \ \ (j = 1, 2, 3, 4).$$

Figure 1.1 shows CPG's for x_1, x_2, x_3, x_4. For visual effect, we use 50 (=n) Monte Carlo generated cases from Uniform$(-1, 1)$. We observe that CPG(1)'s are parallel each other. Thus we may infer that f is the sum of two separable components, of which one is a function of x_1 and the other does not depend on x_1. Similarly, CPG(4)'s are parallel each other. Hence the role of x_4 is same as that of x_1. On the other hand, nonparallel CPG(2)'s and CPG(3)'s indicate that there exists the interaction between x_2 and x_3 in determining f. Thus we conclude that f can be expressed as

$$f(x_1, x_2, x_3, x_4) = s(x_1) + t(x_4) + u(x_2, x_3).$$

Figure 1.2 shows perspective plots for two components of $f(x_1, x_2, x_3, x_4)$:

$$f_1(x_1, x_4) = x_1 - x_4 \sin(\pi\,x_4) \ \text{ and } \ f_2(x_2, x_3) = 2\,x_2^2\,x_3.$$

In the left plot, we see that $f_1(x_1, x_4)$ are additive sum of component functions. In the right plot, we notice that $f_2(x_2, x_3)$ contains interactions.

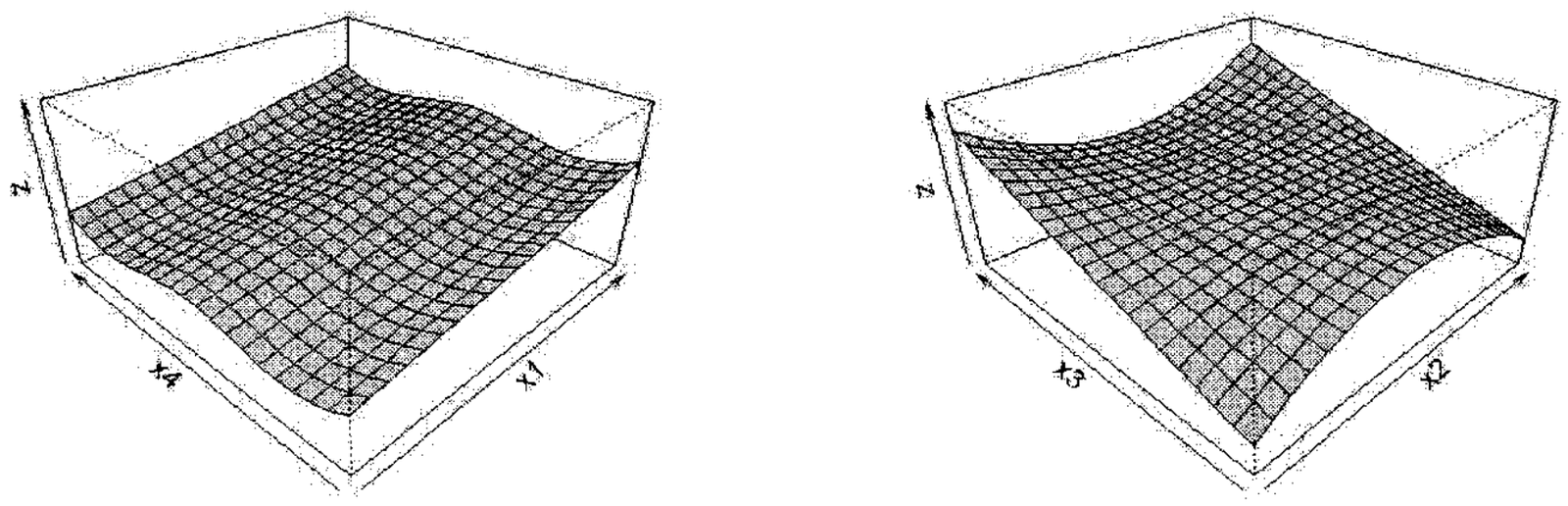

Figure 1.2: Perspective plots of two components of $f(x_1, x_2, x_3, x_4)(f_1(x_1, x_4) = x_1 - x_4 \sin(\pi x_4)$ (left) and $f_2(x_2, x_3) = 2\, x_2^2\, x_3$ (right))

The proposed graphs can be contrasted to the special case of Friedman and Popescu's plots (Friedman and Popescu, 2005) of partial dependency functions, which are equal to the pointwise average of CPG's for each predictor. By taking the average, however, the plots lose the information tag indicating whether the function is additive in that variable or not.

In coming sections, we will demonstrate the conditional predictive graphing (CPG) method for visualizing prediction functions produced by various supervised learning engines such as LOESS, neural network, support vector machines and random forests.

2. LOESS

LOESS fits locally a polynomial surface by one or more numerical predictors (Cleveland *et al.* 1992). We will demonstrate our CPG method as applied to a LOESS model of `stackloss` data, which consists of 21 cases on the response variable `stack.loss` and three predictors '`Air Flow`', '`Water Temp`' and '`Acid Conc.`' (Brownlee, 1960).

Figure 2.1 shows three CPG's for LOESS prediction function with `span=1` corresponding to respective predictors. Observation numbers are plotted at (x, y), where x = realized predictor and y = predicted response. The third CPG's by '`Acid Conc.`' is peculiar in two respects. First, four cases (1, 2, 3, 21) are quite different from the others. Second, there exists a sharp cut at $x = 80$. Such roughness led us to try another LOESS with a larger span. Figure 2.2 contains three CPG's for LOESS prediction function with `span=5`. Now, the sharp cut at some value of '`Acid Conc.`' disappears. But, four cases (1, 2, 3, 21) still show peculiar patterns in CPG's corresponding to '`Water Temp`' and '`Acid Conc.`'.

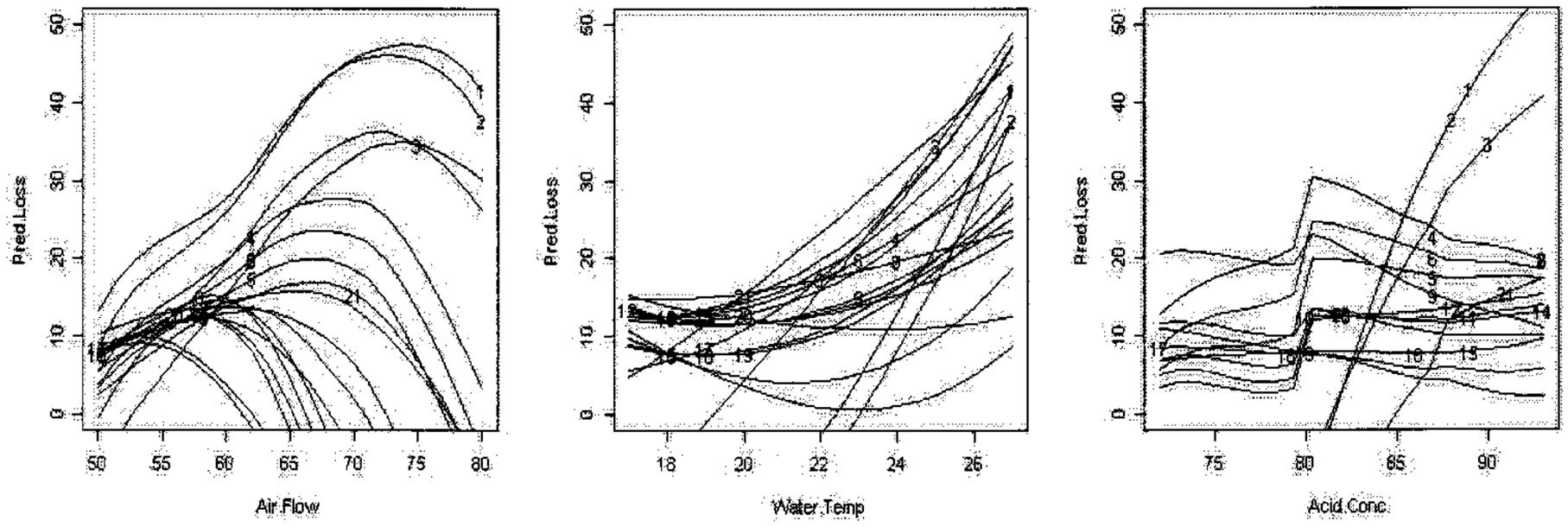

Figure 2.1: CPG's of LOESS (`span=1`) model for `stackloss` data

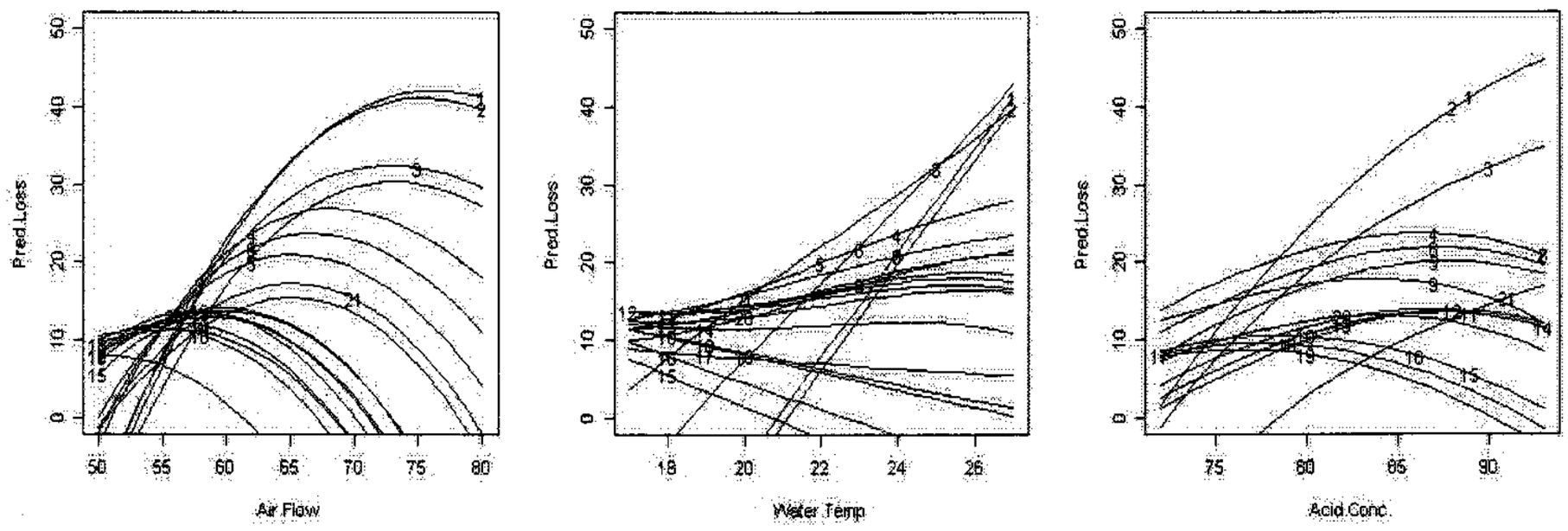

Figure 2.2: CPG's of LOESS (`span=5`) model for `stackloss` data

3. Neural Networks

We will assume a single-hidden-layer neural network which contains two neurons in the hidden layer to the `PimaIndiansDiabetes2` data (Ripley, 1996; R's `mlbench` library). Response variable is `dichotomous` diabetes (`pos` or `neg`) and predictors are log transform of (1+`pregnant`), `glucose`, `pressure`, `triceps`, `mass`, `pedigree` and `age`. Only the complete cases with no missing values are fitted to the model ($n = 532$).

Figure 3.1 shows CPG's for singe-hidden-layer neural network's predictions which are softmax transform of combined signals from hidden neurons. We see that the figures are too dark, caused by over-plotting more than 500 curves. Hence we need "thinning". Figure 3.2 shows systematically sampled CPG's (with fraction rate 1/20). Now, pictures look more nice compared to original Figure 3.1 We notice that three predictors are important in determining the response: `glucose`, `mass` and `pedigree`. These predictors are positively associated with `diabetes`, the response.

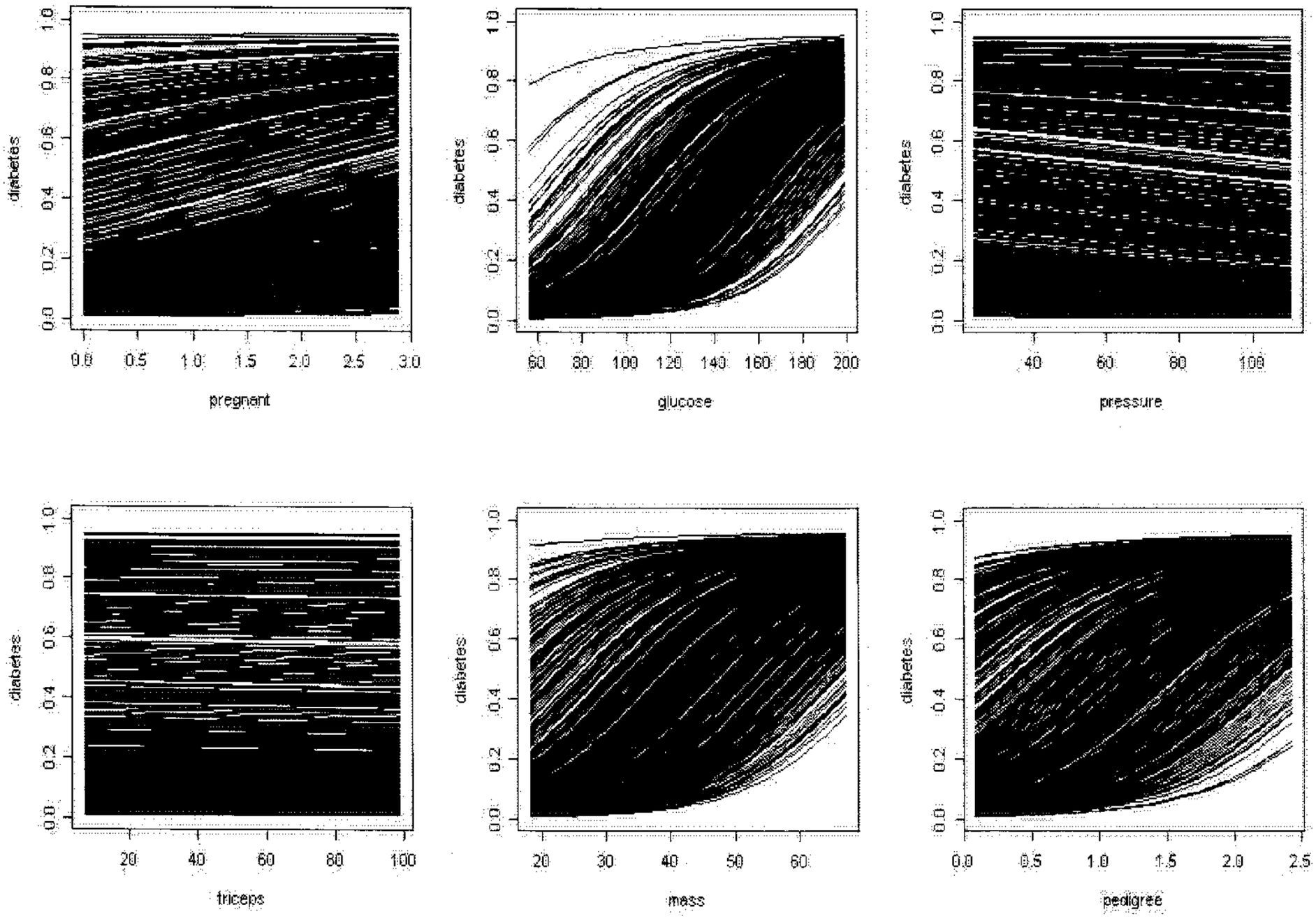

Figure 3.1: CPG's of neural network model for `PimaIndiansDiabetes2` data

Also, CPG's in Figure 3.2 overall indicate that the interactions among predictors are not strong: Individual curves do not cross each other. According to Jiang and Owen (2002) who analyzed the same data, the additivity contributes 78–85% of the total variation.

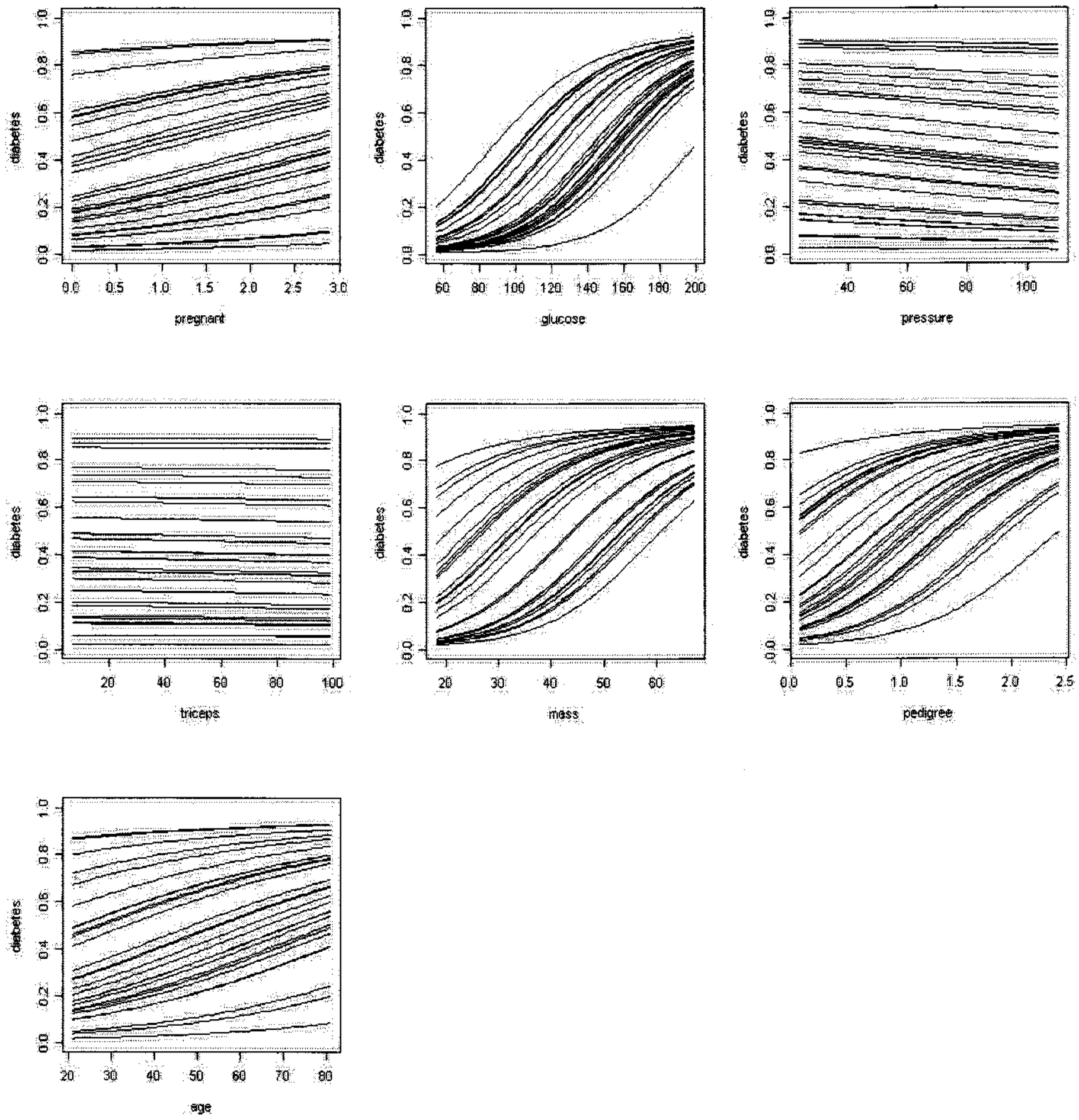

Figure 3.2: Systematically sampled CPG's (from Figre 3.1)

4. Support Vector Machines

We fitted the support vector machine(SVM) which allows probability predictions for famous `iris` data, which consists of four predictors (`sepal.length`, `sepal.width`, `petal.length`, `petal.width`) for three different kinds of species (`setosa`, `versicolor`, `virginica`). Figure 4.1 shows CPG's for the predicted probability of "`versicolor`". We may see that there are two bundles of curves. Without difficulty, we could link one bundle of curves to "`setosa`" and the other bundle to either "`versicolor`" or "`virginica`".

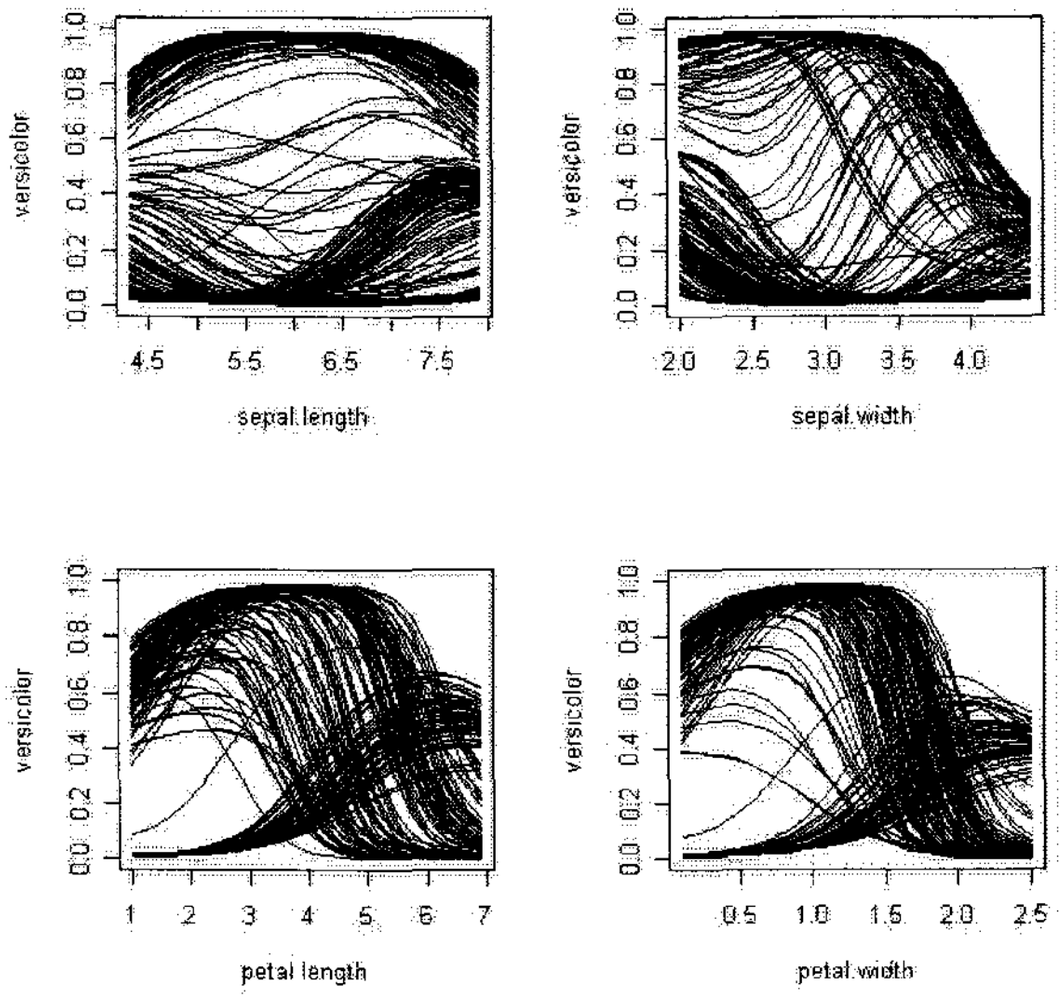

Figure 4.1: CPG's of support vector machine for `iris` data

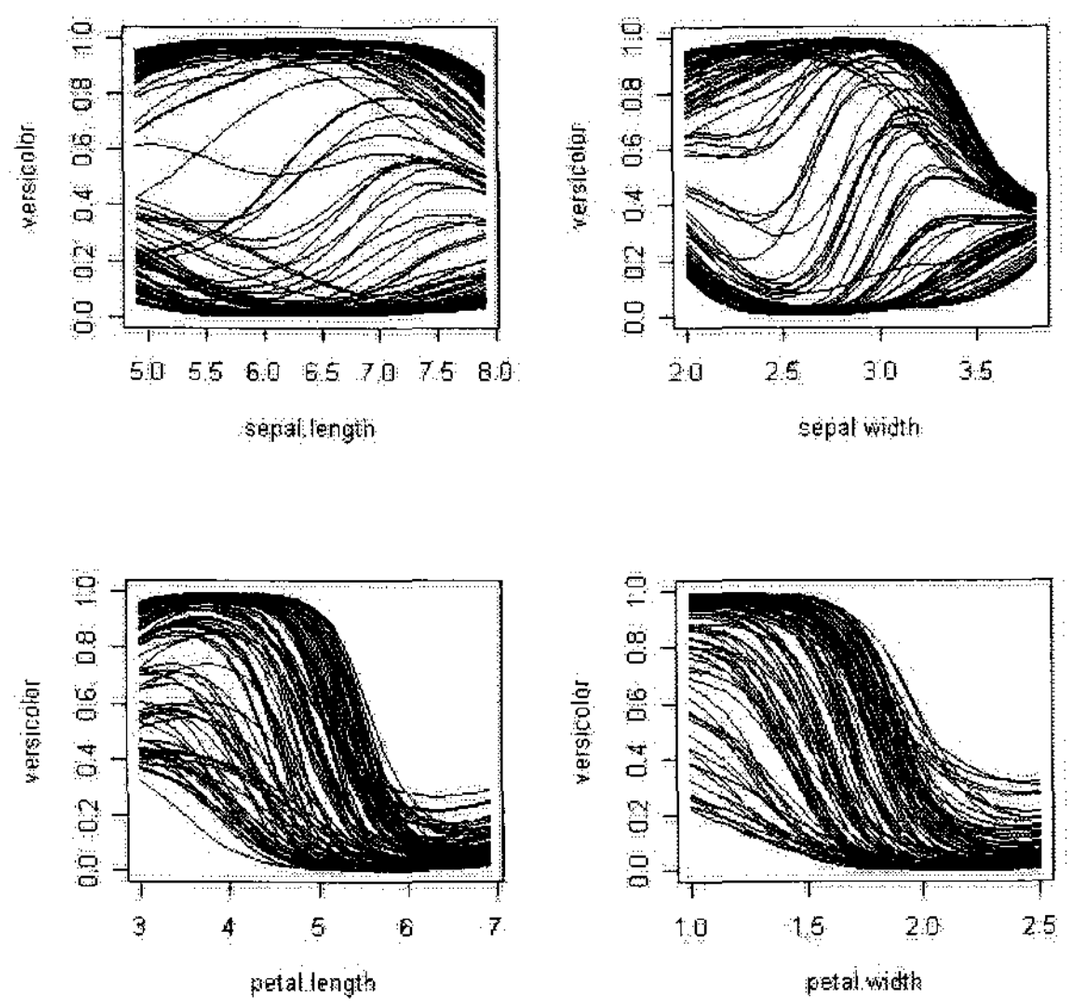

Figure 4.2: CPG's of support vector machine for `iris` data. `Versicolor` *vs.* `Virginica`

As a next step, we imposed the SVM for a subset data of `iris` belonging to "`versicolor`" and "`virginica`". Figure 4.2 shows the resulting CPG's. Now, SVM as binary classifier becomes more easily understandable. 1) `Petal.length` and `petal.width`

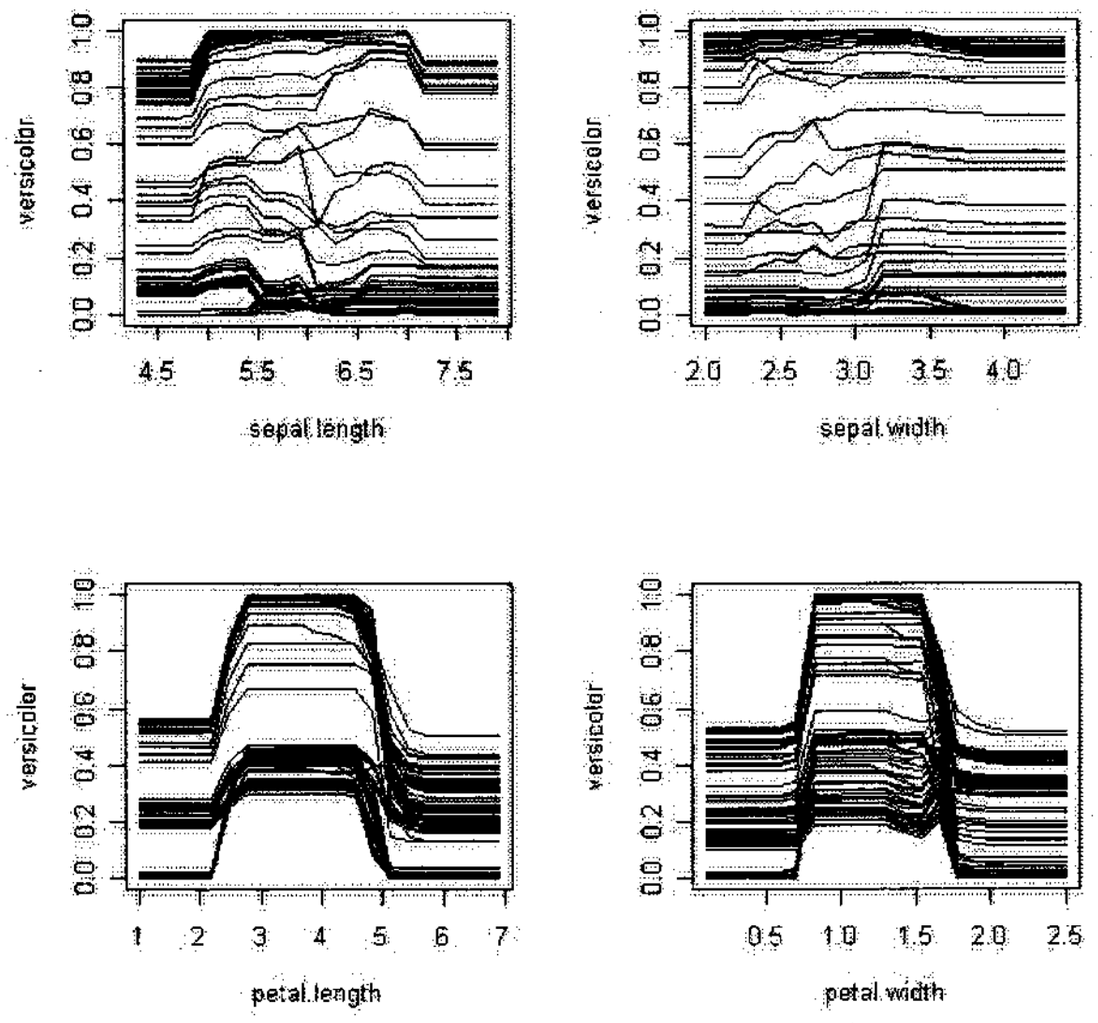

Figure 5.1: CPG's of random forest model for `iris` data

are important predictors, while the role of `sepal.length` and `sepal.width` is limited. 2) Small `petal.length` and `petal.width` predicts "`versicolor`", while large `petal.length` and `petal.width` predicts "`virginica`".

5. Random Forests

Figures 5.1 and 5.2 show CPG's of Breiman's random forest models (Breiman, 2001) fitted to the `iris` data with three species (`setosa`, `versicolor`, `virginica`) and two species (`versicolor` vs. `virginica`), respectively. We may verify visually that Figures 5.1 and 5.2 from random forests correspond to Figures 4.1 and 4.2 originated from SVM. However, there is one apparent difference: The random forest is more additive than the SVM in this case.

6. Concluding Remarks

By conditional predictive graphs (CPG's), we can capture overall features of the complex function of p predictors produced by supervised learning engines. Nevertheless, CPG's do not show all details of the prediction function. But CPG's could be useful in diagnosing the additivity of variables for the predictive function at hand, as Kim's graphical method (Kim, 2008) for generalized linear models. We need a further research in that direction.

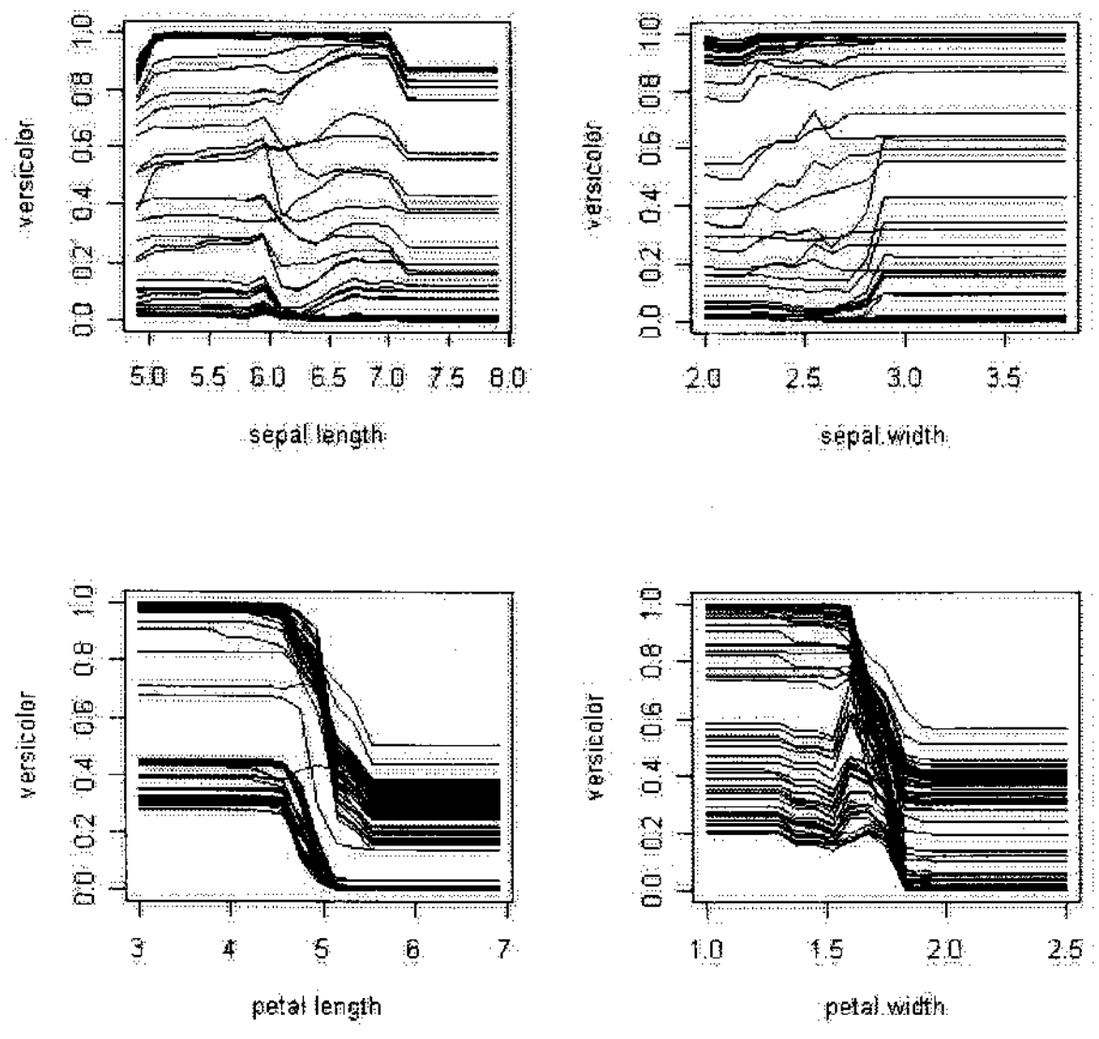

Figure 5.2: CPG's of random forest model for `iris` data. `Versicolor` *vs.* `Virginica`

References

Breiman, L. (2001). Random forests, *Machine Learning*, **45**, 5–32.

Brownlee, K. A. (1960). *Statistical Theory and Methodology in Science and Engineering*, Jhon Wiley & Sons, New York.

Cleveland, W. S., Grosse, E. and Shyu, W. M. (1992). Local regression models, Chapter 8 of *Statistical Models in S* (eds by J.M. Chambers and T.J. Hastie), Wadsworth & Brooks/Cole.

Friedman, J. H. and Popescu, B. E. (2005). *Predictive learning via rule ensembles*, Technical Report, Department of Statistics, Stanford University.

Jiang, T. and Owen, A. B. (2002). *Quasi-regression for visualization and interpretation of black box functions*, Technical Report, Department of Statistics, Stanford University.

Kim, J. H. (2008). A graphical method of checking the adequacy of linear systematic component in generalized linear models, *Communications of the Korean Statistical Society*, **15**, 27–41.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge.

[Received February 2008, Accepted March 2008]