International Journal of Control and Automation
Vol.7, No.9 (2014), pp.45-54
http://dx.doi.org/10.14257/ijca.2014.7.9.04

Desirable Properties of Learning Function from Examples

Bodong Liang

School of Automotive and Transportation Engineering
Shenzhen Polytechnic
Shenzhen 518055, China
liangbodong@szpt.edu.cn

Abstract

In this paper, we argue that total three desirable properties should be satisfied so that a function to be learned or interpolated from a set of input-output examples can cross all the given examples with minimal oscillations among the examples. As long as the number of given examples exceeds the dimension of example and meanwhile none existing hyper-plane, in real vector space of example's dimension, passes through all the given examples exactly, we can construct one simple function learning solution, which is expressed as a sum of two terms: one is an example-influence term that consists of the outputs of a number of basis functions and another is a linear term, to allow all the three desirable properties to be satisfied exactly. Experiments show that the solution can simulate both continuous and discontinuous functions even with very sparse given examples.

Keywords: *function learning, input-output example set, desirable property*

1. Introduction

A powerful method to learn a function from a set of input-output examples is so-called *Example-Based Interpolation* (EBI) [11][15], which can be viewed as an input-output mapping to be interpolated from the collection of example data for the purpose of creating this mapping for novel input queries. Compared with artificial neural networks [4], EBI would automatically find the function that, without the need of adjusting any control parameter (like the number of neurons etc. in the case of artificial neural networks), crosses all the input examples exactly with minimal oscillations among the examples. It is particularly suitable for solving tasks for which there are no well-defined or computationally tractable algorithmic solution. Such tasks are found in a variety of domains including computer vision [6] [7] [1], object detection and character recognition [20] [13] [5], computer animation and graphics [10] [22] [18], and image processing [8][21] [9].

In this paper, we point out there are three desirable properties for EBI solution that should be satisfied so that the learned function can satisfy all the given examples with minimal oscillations. The function learned under the EBI scheme can be expressed as a sum of two terms: an example-influence term that consists of the outputs of a number of basis functions (BF), and a polynomial term [15] [19]. We show that, as long as the number of given examples exceeds the dimension of each example and meanwhile there does not exist one hyper-plane, in real vector space of example's dimension, passing through all the given examples, even with a linear form of the polynomial term the EBI function already has enough degrees of freedom to satisfy all the three desirable properties.

ISSN: 2005-4297 IJCA

EBI requires a BF to model the influence of each example to any arbitrary input. Under this mechanism, a weight would be assigned to each example and for any arbitrary input, the learning function would then be the weighted sum of the BF outputs, which could be radial basis function (RBF) outputs or outputs of any other function that models influence of an example point to its neighborhood, of all the given examples at the input. The function learning problem boils down to the design of the weights, one for each example. In the literature, RBF [14][19] [17][22], distance weighted regression [2][12][8], and other methods have been used as the BF. In our experiments, we choose RBF as the BF for illustration only, but the derived results are actually applicable to EBI coupled with any neighborhood-influence template. Through experiments, we testify that this form of EBI solution can satisfy all the above three desirable properties, even for learning discontinuous function as long as the given examples are adequate.

2. Desirable Properties in Function Learning

2.1. Problem Statement

Given S input-output pairs $\{(\mathbf{V}_s, f_{\mathbf{V}_s}) : s = 1,2,\ldots, S\}$ as examples of a function $f : \mathbf{R}^D \to \mathbf{R}$, the function learning problem is then to construct scalar function value $f_{\mathbf{V}} = f(\mathbf{V})$ for any arbitrary input $\mathbf{V}$ in real D-dimensional vector space $\mathbf{R}^D$ such that it satisfies $f_{\mathbf{V}_s} = f(\mathbf{V}_s)$ for all the S given examples. We shall refer to each $\mathbf{V}_s$ ($s \in \{1,2,\ldots, S\}$) as the sth *example point*, and $f_{\mathbf{V}_s}$ the sth *example value*.

The S example values $f_{\mathbf{V}_1}, f_{\mathbf{V}_2}, \cdots, f_{\mathbf{V}_S}$ could be expressed as a matrix $\mathbf{E}$, which we refer to as the *example value matrix*:

$$\mathbf{E} = [\, f_{\mathbf{V}_1}, f_{\mathbf{V}_2}, \cdots, f_{\mathbf{V}_S} \,] . \tag{1}$$

The problem could then be expressed as inferring function $f_{\mathbf{V}}$, for any input $\mathbf{V}$ in $\mathbf{R}^D$, from example value matrix $\mathbf{E}$.

A natural inference of $f_{\mathbf{V}}$ for any $\mathbf{V}$, as illustrated in Figure 1, is a weighted sum of the example values, *i.e.*,

$$f_{\mathbf{V}} = \mathbf{E}\mathbf{W}_{\mathbf{V}} , \tag{2}$$

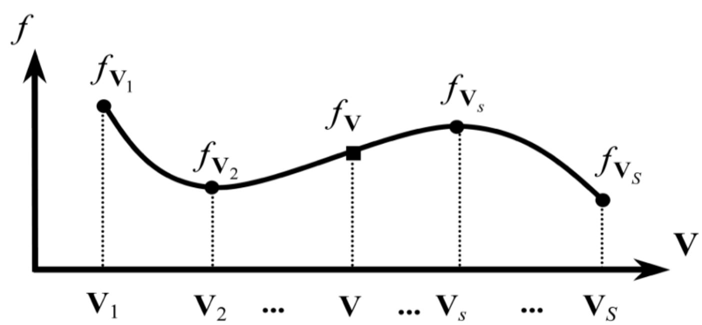


Figure 1. Learning a Scalar Function from Example Values at a Number of Example Points. Dots Represent the known Function Values at the Various Eample Points and Square Represents the Learning Result

where $\mathbf{w}_{\mathbf{v}} = [w_{1,\mathbf{v}}, w_{2,\mathbf{v}}, \ldots, w_{S,\mathbf{v}}]^{\mathrm{T}}$ referred to as the *weight matrix* represents, for computing $f_{\mathbf{v}}$ at any arbitrary input $\mathbf{v}$, the respective weights given to the S example values respectively.

The problem thus boils down to the design of the weights, one for each example.

2.2. Desirable Properties of Function Learning

The to be learned function need to cross all the example points with minimal oscillations among the examples, so the weight matrix $\mathbf{w}_{\mathbf{v}}$ is a function of $\mathbf{v}$ since naturally it varies with $\mathbf{v}$ in this way: the closer $\mathbf{v}$ is to any particular example point $\mathbf{v}_s$ ($s = 1,2, \cdots, S$), the larger the weight is allocated to the example value $f_{\mathbf{v}_s}$ relative to the other weights. Three natural properties for design of $\mathbf{w}_{\mathbf{v}}$ are as follows.

2.2.1. Property P1: Unity sum of all weights. For any arbitrary input $\mathbf{v}$, the sum of all S weights in $\mathbf{w}_{\mathbf{v}}$ is equal to 1, i.e.,

$$\sum_{s=1}^{S} w_{s,\mathbf{v}} = [1,1,\ldots, 1]\mathbf{w}_{\mathbf{v}} = 1 . \qquad (3)$$

This condition has the origin that the contributions of the example values at the various example points are relative, not absolute.

2.2.2. Property P2: Exact satisfaction to examples. If $\mathbf{v}$ happens to be one of the example point $\mathbf{v}_s$, where $s = 1,2, \cdots, S$, the sth entry of $\mathbf{w}_{\mathbf{v}}$ is 1 while all the other entries are 0, i.e., for all s and $\mathbf{v}_\alpha$, where $s, \alpha \in \{1,2,\ldots, S\}$, we have

$$w_{s,\mathbf{v}_\alpha} = \begin{cases} 1 & \text{if } s = \alpha \\ 0 & \text{otherwise} \end{cases} . \qquad (4)$$

This condition allows the constructed function to satisfy the given input-output pairs exactly.

2.2.3. Property P3: Smoothly passing through examples. For any input $\mathbf{v}$, the contributions of the various example values $f_{\mathbf{v}_1}, f_{\mathbf{v}_2}, \cdots, f_{\mathbf{v}_S}$ to $f_{\mathbf{v}}$ are in accordance with the relative proximity of their examples points $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_S$ to $\mathbf{v}$. Same as (2), this implies the weighted sum of all example points $\mathbf{v}_s$'s in accordance with $\mathbf{w}_{\mathbf{v}}$ should return $\mathbf{v}$ itself, *i.e.*,

$$\mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_S]\mathbf{w}_{\mathbf{v}} . \qquad (5)$$

Finally the function learning problem is much about the design of $\mathbf{w}_{\mathbf{v}}$ that conforms to Properties in (3), (4) and (5) as much as possible. With this sense, we call them three desirable properties of learning function.

If $\mathbf{A}_{\mathbf{v}} = [1, \mathbf{v}]^{\mathrm{T}}$, $\mathbf{A}_{\mathbf{v}_s} = [1, \mathbf{v}_s]^{\mathrm{T}}$, and $\mathbf{A} = [\mathbf{A}_{\mathbf{v}_1}, \mathbf{A}_{\mathbf{v}_2}, \ldots, \mathbf{A}_{\mathbf{v}_S}]$, where $s = 1,2, \cdots, S$, then we can combine Property P1 and Property P3 to:

$$\mathbf{A}_{\mathbf{V}} = \mathbf{A}\mathbf{W}_{\mathbf{V}} . \quad (6)$$

3. A Simple Solution of Learning Function by EBI Mechanism

3.1. EBI with Basis Function and Polynomial

A natural design for $\mathbf{W}_{\mathbf{V}}$ is that the *s*th weight $w_{s,\mathbf{V}}$ inside it, which is for the *s*th example value $f_{\mathbf{V}_s}$, is related to how distant the corresponding example point $\mathbf{V}_s$ is from query point $\mathbf{V}$ relative to all other example points. The farther is $\mathbf{V}_s$ away from $\mathbf{V}$ in comparison with all other example points, the smaller should be the contribution of the example value $f_{\mathbf{V}_s}$ to $f_{\mathbf{V}}$. On this, basis functions (BF), one kind of neural networks, around each of the example points could give help [15][19]:

$$\mathbf{W}_{\mathbf{V}} = \mathbf{r}\mathbf{R}_{\mathbf{V}} + \mathbf{p}_m(\mathbf{V}) , \quad (7)$$

where $\mathbf{r}$ and $\mathbf{R}_{\mathbf{V}} = [R_1(\mathbf{V}), R_2(\mathbf{V}), \cdots, R_S(\mathbf{V})]^{\mathrm{T}}$ are the basis functions coefficients and basis functions themselves. $\mathbf{p}_m(\mathbf{V})$ is a polynomial term of degree *m* that assures a certain polynomial precision and only depends upon $\mathbf{V}$.

We use scalar function $R_s(\mathbf{V})$ at any arbitrary input $\mathbf{V}$ to denote the BF that is of a fixed radius in the $\mathbf{V}$-space and centered at the *s*th example point $\mathbf{V}_s$ ($s \in \{1,2,\cdots,S\}$), *i.e.*,

$$R_s(\mathbf{V}) = \Phi(c(\mathbf{V}_s), d(\mathbf{V},\mathbf{V}_s), \theta(\mathbf{V},\mathbf{V}_s)) , \quad (8)$$

where $c(\mathbf{V}_s)$ represents the center of this BF is $\mathbf{V}_s$, $d(\mathbf{V},\mathbf{V}_s)$ represents the distance between $\mathbf{V}_s$ and $\mathbf{V}$, $\theta(\mathbf{V},\mathbf{V}_s)$ represents the orientation from $\mathbf{V}_s$ to $\mathbf{V}$.

Note that $\mathbf{R}_{\mathbf{V}}$ is known for us through selecting the BF of $R_s(\mathbf{V})$'s center *c*, distance *d* and orientation θ. In later section of this paper, we shall show a method to determine the each term of $R_s(\mathbf{V})$ in detail.

3.2. Simple Solution to EBI: Simplifying Polynomial Term to Linear Form

In this section, we will show that it is sufficient to determine $\mathbf{W}_{\mathbf{V}}$ with the three desirable properties described in (3), (4) and (5) if we choose linear polynomials, where $m = 1$, in (7), *i.e.*,

$$\mathbf{W}_{\mathbf{V}} = \mathbf{r}\mathbf{R}_{\mathbf{V}} + \mathbf{a}\mathbf{A}_{\mathbf{V}} , \quad (9)$$

where $\mathbf{a}$ is linear functions coefficient.

In practical and general case of examples locating randomly over whole example set, as long as the number of examples is larger than the dimension of example set, *i.e.*, $S \geqslant (D+1)$, and in geometry, there does not exist a hyper-plane, in real D-dimensional vector space $\mathbf{R}^D$, passing through all the S given example points, *i.e.*, $\mathrm{rank}(\mathbf{A}) = (D+1)$, we can have a least-squares-error solution to $\mathbf{W}_{\mathbf{V}}$ as follow.

3.2.1. Step 1 – a-Determination Step: Fitting a least squares hyper-plane to approximate the over-determined condition (just-determined while $S = (D+1)$) where one example has a value of 1 and the rest have a value of 0, i.e., $\mathbf{I} = \mathbf{aA}$. The least-squares-error solution for **a** is then given by

$$\mathbf{a} = \mathbf{A}^{T}(\mathbf{AA}^{T})^{-1}. \tag{10}$$

It must be noticed that with this least-squares-error solution of (10), $\mathbf{aA}$ might only be close but not exact to $\mathbf{I}$.

3.2.2. Step 2 – r-Determination Step: Calculating the residuals from the hyper-plane approximation by

$$\mathbf{rR} = \mathbf{I} - \mathbf{aA}, \tag{11}$$

where $\mathbf{R} = [\mathbf{R}_{\mathbf{V}_1}, \mathbf{R}_{\mathbf{V}_2}, \ldots, \mathbf{R}_{\mathbf{V}_S}]$. This is a just-determined system of equation for $\mathbf{r}$ as long as $\mathbf{R}$ is nonsingular, such that the above will be led to

$$\mathbf{r} = (\mathbf{I} - \mathbf{aA})\mathbf{R}^{-1}. \tag{12}$$

For the particular case in which $\mathbf{R}$ is singular, Equation (11) is under-determined for $\mathbf{r}$, then there are infinite solutions for $\mathbf{r}$.

3.3. Satisfaction of Desirable Properties in the Least-Squares-Error Solution

Notice that Equations (11) and (12) imply $\mathbf{I} = \mathbf{rR} + \mathbf{aA}$, which is just the matrix form of the Property P2 listed out in (4) combined with the design for $\mathbf{W}_{\mathbf{V}}$ in Equation (9). In addition, $\mathbf{AW}_{\mathbf{V}} = \mathbf{A}_{\mathbf{V}}$, shown in Equation (6) and combined the Property P1 and P3 listed out in Equations (3) and (5), is always satisfied as long as Equations (9), (10), and (12) are given because

$$\begin{aligned}
\mathbf{AW}_{\mathbf{V}} &= \mathbf{A}(\mathbf{rR}_{\mathbf{V}} + \mathbf{aA}_{\mathbf{V}}) = \mathbf{ArR}_{\mathbf{V}} + \mathbf{AaA}_{\mathbf{V}} \\
&= \mathbf{A}(\mathbf{I} - \mathbf{aA})\mathbf{R}^{-1}\mathbf{R}_{\mathbf{V}} + \mathbf{AA}^{T}(\mathbf{AA}^{T})^{-1}\mathbf{A}_{\mathbf{V}} \\
&= (\mathbf{A} - \mathbf{AaA})\mathbf{R}^{-1}\mathbf{R}_{\mathbf{V}} + \mathbf{A}_{\mathbf{V}} \\
&= (\mathbf{A} - \mathbf{AA}^{T}(\mathbf{AA}^{T})^{-1}\mathbf{A})\mathbf{R}^{-1}\mathbf{R}_{\mathbf{V}} + \mathbf{A}_{\mathbf{V}} \\
&= (\mathbf{A} - \mathbf{A})\mathbf{R}^{-1}\mathbf{R}_{\mathbf{V}} + \mathbf{A}_{\mathbf{V}} \\
&= \mathbf{A}_{\mathbf{V}}.
\end{aligned}$$

Till now, even for the very practical case in which basis functions $\mathbf{R}$ is nonsingular, we have proved that the least-squares-error solution, shown in (9), (10), and (12), could allow the three desirable properties, shown in (3), (4) and (5), be satisfied exactly.

4. Experiments of Learning Nonlinear Function

4.1. Selection of Basis Function

Here we make use of radial basis function (RBF) instead of BF in our real procedure.

RBF, a special case of BF, does not consider influence of orientation function $\theta(\mathbf{V}, \mathbf{V}_s)$ shown in (8). RBF defined so that it decreases as the distance increases provides a global approximation to the target function. The value for any given RBF is non-negligible only when the input query falls into the region defined by its particular center and width. RBF is discussed in [11], and surveyed in [15]. There are number of choices to select the RBF including Gaussian, multi-quadric, thin-plate splines, B-splines [3][16], and other forms.

Below we shall finalize and select each term of $R_s(\mathbf{V})$ shown in (8) for our following experiments.

4.1.1. Center Selection: For $R_s(\mathbf{V})$ in (8), the center $c(\mathbf{V}_s)$ is at $\mathbf{V}_s$ (the *s*th example point) and location is at $\mathbf{V}$.

4.1.2. Distance Selection: For $R_s(\mathbf{V})$ in (8), the distance $d(\mathbf{V},\mathbf{V}_s)$ between $\mathbf{V}_s$ and $\mathbf{V}$ is measured by the Euclidean norm, *i.e.*, $d(\mathbf{V},\mathbf{V}_s) = \|\mathbf{V} - \mathbf{V}_s\|_2$.

4.1.3. Orientation Selection: For $R_s(\mathbf{V})$ in (8), since the symmetrical RBF would be used, the orientation $\theta(\mathbf{V},\mathbf{V}_s)$ from $\mathbf{V}_s$ to $\mathbf{V}$ is always set to zero, *i.e.*, $\theta(\mathbf{V},\mathbf{V}_s) = 0$.

4.1.4. Function Selection: There are many choices for function selection. Rose *et al.* [17] chose a basis with a cross section of a cubic B-spline centered on the example and with radius twice the Euclidean distance to the nearest other examples. We adopt same structure in this paper such that, for a cubic B-spline centered on $\mathbf{V}_s$ ($s = 1,2,\cdots,S$), the radius r_s is defined as:

$$r_s = 2 \times \underset{n=1,2,\cdots,S}{\text{median}} \left\{ \|\mathbf{V}_n - \mathbf{V}_s\|_2 \right\}.$$

Let $r = \frac{1}{2} r_s$, $d = d(\mathbf{V},\mathbf{V}_s) = \|\mathbf{V} - \mathbf{V}_s\|_2$, and $\theta(\mathbf{V},\mathbf{V}_s) = 0$, the function $R_s(\mathbf{V})$ in (8) with center at $\mathbf{V}_s$ and radius of r_s then has the form of

$$R_s(\mathbf{V}) = \begin{cases} A\left[r^3 + 3r^2(r-d) + 3r(r-d)^2 - 3(r-d)^3\right] & \text{if } d \in [0,r] \\ A(2r-d)^3 & \text{if } d \in [r,2r] \\ 0 & \text{otherwise} \end{cases}.$$

where A is coefficient and will be cancelled in determining final result of $\mathbf{w}_{\mathbf{V}}$. In our real processing, as illustrated in Figure 2, we choose $A = \frac{1}{4r^3}$ from the viewpoint of having $R_s(\mathbf{V}_s) = 1$.

4.2. Experimental Results

Here we are to use the EBI mechanism described in (2), (9), (10) and (12) to simulate two set of nonlinear functions: one set includes continuous sine and cosine functions, and another is a customer-defined discontinuous function.

4.2.1. Learning of Sine and Cosine Functions: Here we only selected 7 input angles and their sine and cosine outputs, from 0° to 360° every 60°, as our examples and chose RBF as same as that defined in the previous section.

The function learning result, shown in Figure 3, tightly overlapped the ground truth of both sine and cosine functions in small bias.

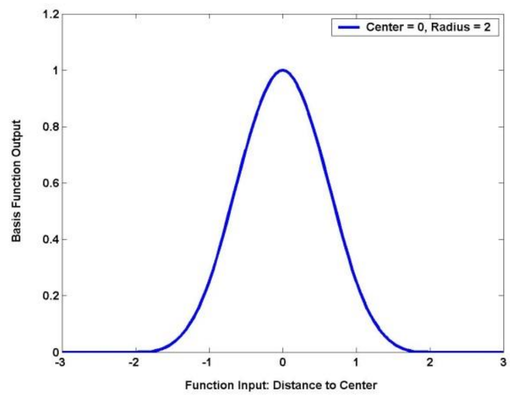


Figure 2. Radial Basis Function with Center at Zero and Radius of Two.

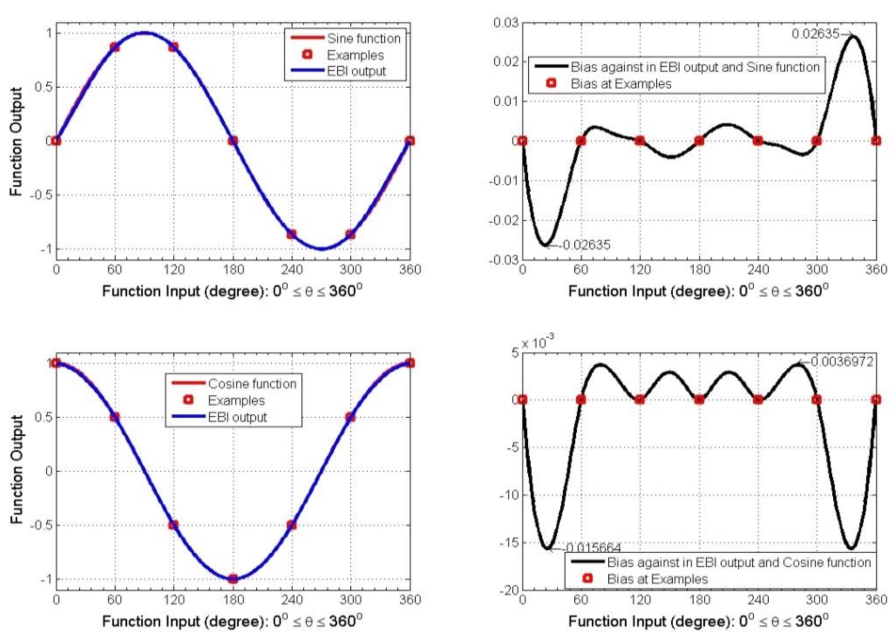
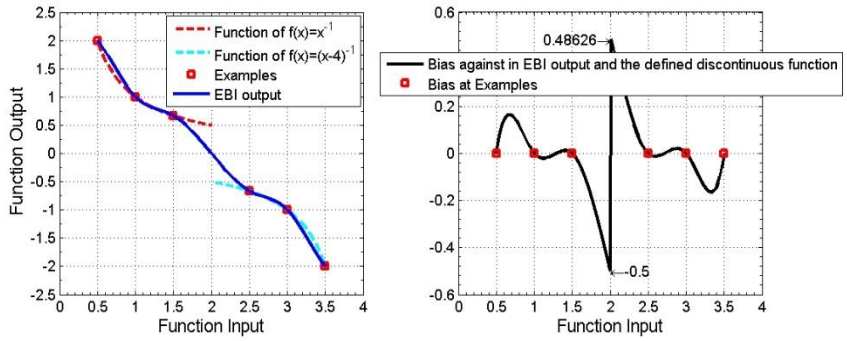


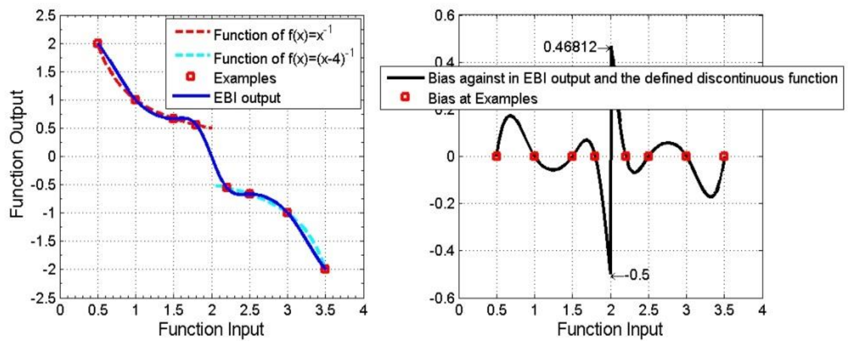
Figure 3. Learning Sine and Cosine Functions from a Number of Examples. The 7 Squares Represent 7 Examples, which are the Sine or Cosine Function Outputs at 7 Example Points, from 0° to 360° every 60°, respectively. The Left Curves are Learning Results of Sine and Cosine Functions with the EBI Mechanism, the Right Curves are Results of EBI Output Minus Sine and Cosine Function, Respectively, in which the Biases Against all Examples in EBI Output and Sine and Cosine Function are Zero Exactly

4.2.2. Learning of a discontinuous function: The to be simulated discontinuous function is defined as

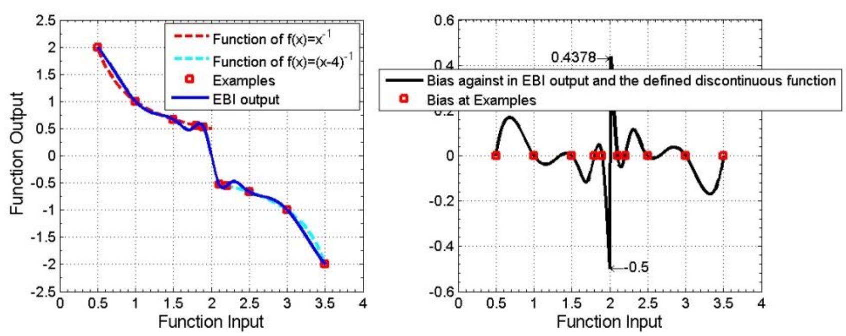
$$f(x) = \begin{cases} x^{-1} & \text{if } x \in (0,2] \\ (x-4)^{-1} & \text{if } x \in (2,4) \end{cases}.$$



(a). 6 examples



(b). 8 examples



(c). 10 examples

Figure 4. Learning a Defined Discontinuous Function from (a) 6 Examples, (b) 8 Examples, and (c) 10 Examples, Respectively. All Learning Results Cross all Examples with Minimal Oscillations among the Examples. In Addition, the more Examples at Discontinuous Area are given, the Smaller Biases are Achieved

The function learning result is shown in Figure 4. Notice that, at discontinuous area around $x = 2$, the gaps between EBI output and the defined function are clear. However, by increasing the examples (see Figure 4(a)-(c)) around the discontinuous area, we can easily decrease the bias between EBI output and the defined function. Moreover, we can still see EBI output crosses each example point exactly and tries to pass through all examples smoothly.

Both experiments show that the EBI scheme succeeds in learning function that (i) changes only gradually while the input changes gradually, and (ii) always returns the given example value exactly when the input just falls upon their example point.

Although the above design has $\mathbf{W}_{\mathrm{v}}$ appearing in a linear form in (9), with the use of nonlinear BFs for $\mathbf{R}_{\mathrm{v}}$'s the design is capable of modeling nonlinear functions.

5. Conclusions

In this paper, we pointed out three desirable properties for EBI solution. We also showed that, as long as the number of given examples exceeds the dimension of each example and none existing a hyper-plane, in real vector space of example's dimension, passes through all the given examples, for the EBI solution to satisfy exactly all these three desirable properties, it needs only be in a simple format: the sum of a basis function part plus a linear part. Experimental results showed that the function learning solution can cross all the given examples with minimal oscillations among the examples and is capable of simulating both continuous and discontinuous nonlinear functions even with very sparse given examples.

Acknowledgements

The work described in this paper was partially supported by grants from Shenzhen Polytechnic (No. 2210K3100042 and No. 2613K3100021).

References

[1] J. Ackermann, M. Ritz, A. Stork and M. Goesele, "Removing the example from example-based photometric stereo", Lecture Notes in Computer Science, vol. 6554, **(2012)**, pp. 197-210.

[2] C.G. Atkeson, A.W. Moore and S. Schaal, "Locally weighted learning", Artificial Intelligence Review, vol. 11, no. 1, **(1997)**, pp. 11-73.

[3] R. H. Bartels, J. C. Beatty and B. A. Barsky, "An Introduction to Splines for Use in Computer Graphics and Geometric Modeling", Morgan Kaufmann Publishers, **(1987)**.

[4] C. M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, **(1995)**.

[5] M. Fisher, D. Ritchie, M. Savva, T. Funkhouser and P. Hanrahan, "Example-based synthesis of 3D object arrangements", In Proceedings of ACM SIGGRAPH Asia, vol. 31, no. 6, **(2012)**, Article No. 135.

[6] B. Heisele, A. Verri and T. Poggio, "Learning and vision machines", Proceedings of the IEEE, vol. 90, no. 7, **(2002)**, pp. 1164-1177.

[7] Z. Hui and K.M. Lam, "Eigentransformation-based face super-resolution in the wavelet domain", Pattern Recognition Letters, vol. 33, no. 6, **(2012)**, pp. 718-727.

[8] K. I. Kim and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 6, **(2010)**, pp. 1127-1133.

[9] E. Konukoglu, A. Kouwe, M. R. Sabuncu and B. Fischl, "Example-based restoration of high-resolution magnetic resonance image acquisitions", Lecture Notes in Computer Science, vol. 8149, **(2013)**, pp. 131-138.

[10] S. Martin, B. Thomaszewski, E. Grinspun and M. Cross, "Example-based elastic materials", In Proceedings of ACM SIGGRAPH, vol. 30, no. 4, **(2011)**, Article No. 72.

[11] C. A. Micchelli, "Interpolation of scattered data: distance matrices and conditionally positive definite functions", Constructive Approximation, vol. 2, **(1986)**, pp. 11-22.

[12] T. M. Mitchell, "Machine Learning", McGraw-Hill, **(1997)**.
[13] C. Papageorgiou and T. Poggio, "A trainable system for object detection", International Journal of Computer Vision, vol. 38, no. 1, **(2000)**, pp. 15-33.
[14] T. Poggio and F. Girosi, "Networks for approximation and learning", Proceedings of the IEEE, vol. 78, no. 11, **(1990)**, pp. 1481-1497.
[15] M. J. D. Powell, "A review of methods for multivariable interpolation at scattered data points", in the State of the Art in Numerical Analysis, I.S. Duff and G.A. Watson, Eds., Oxford University Press, **(1997)**, pp. 283-309.
[16] D. F. Rogers, "An Introduction to NURBS: with Historical Perspective, Morgan Kaufmann Publishers", **(2001)**.
[17] C. F. III Rose, M. F. Cohen and B. Bodenheimer, "Verbs and adverbs: multidimensional motion interpolation", IEEE Computer Graphics and Applications, vol. 18, no. 7, **(1998)**, pp. 32-40.
[18] R. Ruiters, C. Schwartz, and R. Klein, "Example-based interpolation and synthesis of bidirectional texture functions", Computer Graphics Forum, vol. 32, no. 2, **(2013)**, pp. 361-370.
[19] D. Ruprecht and H. Muller, "Image warping with scattered data interpolation", IEEE Computer Graphics and Applications, vol. 15, no. 2, **(1995)**, pp. 37-43.
[20] K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 1, **(1998)**, pp. 39-51.
[21] K. Zhang, X. Gao, X. Li and D. Tao, "Partially supervised neighbor embedding for example-based image super-resolution", IEEE Journal of Selected Topics in Signal Processing, vol. 5, no. 2, **(2011)**, pp. 203-239.
[22] J. S. Zurdo, J. P. Brito and M. A. Otaduy, "Animating wrinkles by example on non-skinned cloth", IEEE Transactions on Visualization and Computer Graphics, vol. 19, no. 1, **(2013)**, pp. 149-158.

Authors

Bodong Liang has been with School of Automotive and Transportation Engineering at Shenzhen Polytechnic, China, since 2010. He received his Ph.D. degree from the Chinese University of Hong Kong in 2006. He was Technical Marketing Manager with Zoran Corporation in 2006-2009, and served as FAE at C-Cube Microsystems Inc. and LSI Logic Corporation in 1997-2002. His current research interests are computer vision, artificial intelligence, intelligent transportation systems.

