Why Clustering in Function Approximation? Theoretical Explanation

Vladik Kreinovich[1] and Yeung Yam[2]

[1]Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968, USA
email vladik@utep.edu

[2]Department of Mechanical &
Automation Engineering
The Chinese University of Hong Kong
Shatin, NT, Hong Kong, China
email yyam@mae.cuhk.edu.hk

Abstract

Function approximation is a very important practical problem: in many practical applications, we know the exact form of the functional dependence $y = f(x_1, \ldots, x_n)$ between physical quantities, but this exact dependence is complicated, so we need a lot of computer space to store it, and a lot of time to process it, i.e., to predict y from the given x_i. It is therefore necessary to find a simpler *approximate* expression $g(x_1, \ldots, x_n) \approx f(x_1, \ldots, x_n)$ for this same dependence. This problem has been analyzed in numerical mathematics for several centuries, and it is, therefore, one of the most thoroughly analyzed problems of applied mathematics. There are many results related to approximation by polynomials, trigonometric polynomials, splines of different type, etc. Since this problem has been analyzed for so long, no wonder that for many reasonable formulations of the optimality criteria, the corresponding problems of finding the optimal approximations have already been solved.

Lately, however, new *clustering-related* techniques have been applied to solve this problem (by Yager, Filev, Chu, and others). At first glance, since for most traditional optimality criteria, optimal approximations are already known, clustering approach can only lead to non-optimal approximations, i.e., approximations of inferior quality. We show, however, that there exist new reasonable criteria with respect to which clustering-based

function approximation is indeed the optimal method of function approximation.

1 Informal Formulation of the Problem

1.1 Clustering: the choice of an optimal clustering method is important

In many practical problems, we have several points that represent different observations, and we must divide them into *clusters* so that points from one cluster are, in general, closer to each other than points belonging to different clusters.

There exist different clustering techniques; a survey of main non-fuzzy techniques is given, e.g., in [10]; the main ideas of fuzzy clustering are described in [1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 14, 15].

It is known that different clustering methods lead to results of different quality, so it is extremely important to find the *best* clustering technique.

1.2 Non-clustering applications of clustering techniques: before comparing different clustering methods, we must first check whether it makes sense to use clustering at all

Often, clustering methods are used not for clustering itself, but as an *auxiliary* tool to solve other practical problems.

In such applications, before we find the clustering technique which is the best for this particular application, it is desirable to first find out whether clustering-based methods are indeed good for the corresponding problem. Otherwise, if clustering-based methods are, in principle, much worse than other known techniques, then the search for an optimal clustering-based method makes no practical sense: we will spend quite some time and effort to select the "best" ("least bad") of bad methods, while much better non-cluster-based methods are readily available.

1.3 Function approximation: a recent application area of clustering techniques

One of such non-clustering application of clustering techniques is the problem of *function approximation.* This is a very important practical problem: in many practical applications, we know the exact form of the functional dependence $y = f(x_1, \ldots, x_n)$ between physical quantities, but this exact dependence is complicated, so we need a lot of computer space to store it, and a lot of time to process it, i.e., to predict y from the given x_i. It is therefore necessary to find a simpler *approximate* expression $g(x_1,\ldots,x_n) \approx f(x_1,\ldots,x_n)$ for this same dependence. This problem has been analyzed in numerical mathematics for several centuries, and it is, therefore, one of the most thoroughly analyzed problems of applied mathematics. There are many results related to approximation by polynomials, trigonometric polynomials, splines of different type, etc. (see, e.g., [11]). Since this problem has been analyzed for so long, no wonder that for many reasonable formulations of the optimality criteria, the corresponding problems of finding the optimal approximations have already been solved.

Lately, however, *clustering-related* techniques have been applied to solve this problem (see, e.g., [7, 8, 13]). Instead of applying traditional techniques of numerical mathematics, we approximate a given function $f(x_1,\ldots,x_n)$ as follows:

- we pick a small step $h > 0$ and make a rectangular grid based on this step;
- for each point $(x_1^{(p)},\ldots,x_n^{(p)})$ from this grid, we find the corresponding value $y^{(p)} = f(x_1^{(p)},\ldots,x_n^{(p)})$ and construct the corresponding point $z^{(p)} = (x_1^{(p)},\ldots,x_n^{(p)},y^{(p)})$ from the graph of the function $f(x_1,\ldots,x_n)$;
- then, we apply a clustering algorithm to the resulting points $z^{(p)}$; often, such algorithms work in several steps, leading to a *hierarchical* clusterization; for example, a bottom-up clusterization is performed as follows:
  - first, we pick some reasonably small distance d_1 and combine the points into a cluster only if they are d_1-close to each other; since d_1 is small, some points may be left off this classification;
  - then, we repeat this clusterization procedure for a larger value $d_2 > d_1$; as a result, some old clusters may combine into a single new cluster, and some point that were not previously clustered are now classified;
  - if necessary, we repeat this procedure for an even large value $d_3 > d_2$, etc.

After this, on each resulting cluster, we approximate the original function by a simple expression. This approximation can be a linear function, or it can be a non-linear function, as in fuzzy rules, etc.

The smaller h, the closer the neighboring points $z^{(p)}$. So, it makes sense to choose d_i depending on h, e.g., to choose the values d_i as $d_i = k_i \cdot h$ for some coefficients k_i.

1.4 Clustering in function approximation: hopeless?

This application of clusterization techniques is not only proposed as a practically useful tool, but it is also proposed as a case study for comparing the relative quality of different clustering techniques. But does this comparison make sense?

Since for most traditional optimality criteria, optimal approximations are already known, for these criteria, comparing two different clustering techniques means comparing two clearly non-optimal algorithms.

1.5 There is hope

Function-approximation comparison of clustering methods would make sense only if we can find some reasonable criteria with respect to which clustering-based function approximation is indeed the optimal method of function approximation.

Such criteria will be presented in this paper, for the simplest case of:

- smooth functions of one variable $y = f(x)$,
- the simplest possible clustering algorithm, and
- the simplest possible (linear) approximation on each cluster.

2 For some reasonable criteria, clustering leads to the optimal function approximation: formulation of the result

Definition 1. (of clustering) *Let $k = (1 < k_1 < k_2 < \ldots < \ldots)$ be a strictly increasing sequence of real numbers, let $h > 0$ be a real number, and let $z^{(1)}, z^{(2)}, \ldots, z^{(N)} \in R^2$ be points. By a (k,h)-clustering of the set $\{z^{(1)}, \ldots, z^{(N)}\}$, we mean the following procedure:*

- *First, we combine, into a single cluster, points $z^{(p)}$ and $z^{(q)}$ for which $d(z^{(p)}, z^{(q)}) \le k_1 \cdot h$ (where d denotes Euclidean distance). To be more precise, we consider the points $z^{(p)}$ and $z^{(q)}$ to be:*
  - *immediately close if $d(z^{(p)}, z^{(q)}) \le k_1 \cdot h$, and*
  - *belonging to a common cluster if there exists a sequence of indices $p = p_1, p_2, \ldots, p_{k-1}, p_k = q$ for which, for every i, $z^{(p_i)}$ is immediately close to $z^{(p_{i+1})}$.*
- *Then, we discard all points $z^{(p)}$ which have already been clusterized (i.e., which already have been assigned to a cluster). belong to one cluster with some other point. On the set of remaining (un-clustered) points, we apply a similar clustering procedure, but with $k_2 > k_1$ instead of k_1.*
- *Then, we again discard the clustered points. If there still are non-clustered points, we apply a similar procedure with k_3, etc.*
- *We stop when all points have been clustered.*

Definition 2. (of clustering-based function approximation)

- *Let $f(x)$ be a continuously differentiable function on an interval $[a, b]$. This function will be called an approximated function.*
- *Let $h > 0$ be a real number. This number will be called a step.*
- *By a linear approximation method M, we mean a mapping which, given a finite set of points $z^{(1)}, \ldots, z^{(N)} \in R^2$, produces a linear function $y = a \cdot x + b$.*
- *By an M-based k-clustering-based h-step approximation $f_{M,k,h}(x)$ to $f(x)$, we mean the following piecewise-linear function:*
  - *First, we consider the points $z^{(p)} = (x^{(p)}, f(x^{(p)}))$, where $x^{(1)} = a$, $x^{(2)} = a + h$, ..., $x^{(p)} = a + (p - 1) \cdot h$, ..., until we reach b.*
  - *Then, we apply the k-clustering algorithm to these points.*
  - *Finally, for each of the resulting clusters, we use the method M to generate a linear function.*

  These linear functions, taken together, form the desired approximation $f_{M,k,h}$.

Definition 3. *Let $\varepsilon > 0$ be a real number. We say that the functions $f(x)$ and $g(x)$ are ε-close if $J(f, g) \leq \varepsilon$, where $J(f, g) = \max_x |f'(x) - g'(x)|$.*

Definition 4. *Let $f(x)$ be a continuously differentiable function on an interval $[a, b]$. We say that a piecewise-linear function $f_{\approx}(x)$ is an optimal ε-approximation to $f(x)$ if:*

- *$f_{\approx}(x)$ is an ε-approximation to f, and*
- *among all the piece-wise linear functions $g(x)$ which are ε-approximations to f, the function $f_{\approx}(x)$ has the smallest possible number of linear pieces.*

We say that a piecewise-linear function $f_{\approx}(x)$ is an almost optimal ε-approximation to $f(x)$ if:

- *$f_{\approx}(x)$ is an ε-approximation to f, and*
- *it is either optimal, or it has one more linear piece than an optimal approximation.*

Comment. This optimality criterion makes sense, e.g., if the main goal of our function approximation is to *extrapolate* from the experimentally observed values or pre-computed values x_0 of the function.

Indeed, if the extrapolation interval $[x_0, x]$ is small enough, then, on this interval, the function can be well approximated by a linear expression $f(x) \approx$

$f(x_0)+f'(x_0)\cdot(x-x_0)$. We can then use the approximate expression $f_\approx(x)$ for $f(x)$ to estimate $f'(x_0)$ as $f'_\approx(x_0)$. Since we know $f(x_0)$ exactly, the accuracy of the resulting extrapolation is determined by the accuracy with which we can estimate the derivative $f'(x_0)$. So, if we, e.g., want to be able to predict the value $f(x)$ with an accuracy δ on all extrapolation intervals of the length $\le L$, then we must be able to estimate the derivative $f'(x_0)$ with the accuracy δ/L, i.e., we must have $|f'(x)-f'_\approx(x)|\le\delta/L$ for all x. In our notation, we must have $J(f,f_\approx)\le\delta/L$.

Theorem. (clustering leads to optimal function approximation) *For every $\varepsilon>0$, there exists a sequence k and a linear approximation method M for which, as $h\to 0$, the M-based k-clustering based h-step approximations to $f(x)$ tend to an almost optimal ε-approximation $\lim_{h\to 0} f_{M,k,h}$ to f.*

Comment. So, for an appropriate clustering-based function approximation, for sufficiently small h, the M-based k-clustering based h-step approximation $f_{M,k,h}$ to $f(x)$ is practically identical to an almost optimal ε-approximation to f. Thus, clustering indeed leads to an (almost) optimal function approximation.

3 Proof

Let us first describe the above clustering in analytical terms. For small h, we have $f(x^{(p+1)})=f(x^{(p)}+h)=f(x^{(p)})+h\cdot f'(x^{(p)})+o(h)$, and therefore, the distance $d(z^{(p)},z^{(p+1)})$ between the points $z^{(p)}=(x^{(p)},f(x^{(p)}))$ and $z^{(p+1)}=(x^{(p)}+h,f(x^{(p)}+h))$ is equal to

$$d(z^{(p)},z^{(p+1)})=\sqrt{h^2+(f(x^{(p)}+h)-f(x^{(p)}))^2}=$$

$$\sqrt{h^2+h^2\cdot(f'(x^{(p)}))^2+o(h^2)}=h\cdot\sqrt{1+(f'(x^{(p)}))^2}+o(h).$$

Thus, for sufficiently small h, points x for which $\sqrt{1+(f'(x))^2}\le k_1$ get clustered together with their neighbors, while other points are left out.

Similarly, on the second step, we get all points for which

$$k_1\le\sqrt{1+(f'(x))^2}\le k_2$$

into clusters, and on each step i, we get points for which

$$k_{i-1}\le\sqrt{1+(f'(x))^2}\le k_i$$

into clusters. In terms of the derivatives, this inequality is equivalent to $(k_i)^2\le 1+(f'(x))^2\le(k_{i+1})^2$, or to $\sqrt{(k_i)^2-1}\le|f'(x)|\le\sqrt{(k_{i+1})^2-1}$. Let us show that for appropriate values k_i, the resulting clustering indeed leads to an almost optimal approximation.

Approximating a function $f(x)$, in the sense of Definition 2, by a piecewise-linear function $f_{\approx}(x)$ means that we approximate the derivative $f'(x)$ of the original function by a piecewise-constant function $f'_{\approx}(x)$. We want this approximation to consist of as few pieces as possible. This means that the function $f'_{\approx}(x)$ should take as few constant levels as possible.

Due to the definition of ε-approximation, each constant v of the approximating piecewise-constant derivative $f'_{\approx}(x)$ can serve as a good approximator for all values x for which $f'(x) \in [v-\varepsilon, v+\varepsilon]$. Therefore, the intervals $[v-\varepsilon, v+\varepsilon]$ corresponding to all constant values v of $f'_{\approx}$ must cover the whole range of the function $f'(x)$.

Since the approximated function $f(x)$ is assumed to be continuously differentiable, its derivative $f'(x)$ is a continuous function and therefore, the range of this derivative is a closed interval. So, if $v+\varepsilon$ does not cover the upper endpoint of the derivative's range, then there should be another interval of this type $[v'-\varepsilon, v'+\varepsilon]$ which covers all points close to $v+\varepsilon$, i.e., for which $v'-\varepsilon \leq v+\varepsilon$. If $v'-\varepsilon < v+\varepsilon$, then we can shift all further constant values higher and still get an ε-approximation (and maybe even get fewer segments this way). So, without losing generality, we can assume that in the optimal approximation, the two neighboring intervals $[v-\varepsilon, v+\varepsilon]$ and $[v'-\varepsilon, v'+\varepsilon]$ intersect only in their boundary points, i.e., $v'-\varepsilon = v+\varepsilon$, and $v' = v+2\varepsilon$.

If we denote the values of the constant pieces of the approximating derivative by $v_1 < v_2 < \ldots < v_m$, then we can conclude that $v_2 = v_1 + 2\varepsilon$, $v_3 = v_2 + 2\varepsilon = v_1 + 4\varepsilon$, $\ldots$, and $v_k = v_1 + 2(k-1)\cdot\varepsilon$ for all k. As soon as the values v_i are fixed, the approximation itself is easy to describe: for each x, we pick the value v_i for which $f'(x) \in [v_i - \varepsilon, v_i + \varepsilon]$.

We cannot use this optimal approximation for our purposes because the corresponding values v_i may depend on the approximating function $f(x)$, while we want the values which will work for all functions $f(x)$. Let us therefore use the above *optimal* approximation to design a new, *almost optimal* approximation in which each constant value w_i is equal to $2k\cdot\varepsilon$ for some integer k. Indeed, let us take an arbitrary value v_i from the original optimal approximation. Since intervals $[2k\cdot\varepsilon, 2(k+1)\cdot\varepsilon]$ cover the entire real line, the value v_i must belong to one of these intervals, i.e., $v_i \in [2k\cdot\varepsilon, 2(k+1)\cdot\varepsilon]$ for some integer k. The desired construction of an almost optimal approximation depends on whether v_i in the lower or in the upper half of this interval.

If v_i is in the *lower* half, i.e., if $v_i \in [2k\cdot\varepsilon, (2k+1)\cdot\varepsilon]$, then we take $\Delta = v_i - 2k\cdot\varepsilon$ (so that $0 \leq \Delta \leq \varepsilon$), use $w_1 = v_1 - \Delta, \ldots, w_m = v_m - \Delta$, and $w_{m+1} = w_m + 2\varepsilon$. Let us show that these values cover $f'(x)$ for all x (if we show this, then, since this approximation has one more segment than the optimal one, it is almost optimal). Indeed, the original intervals $[v_i - \varepsilon, v_i + \varepsilon]$ covered the range $[v_1 - \varepsilon, v_m + \varepsilon]$. The new intervals $[w_i - \varepsilon, w_i + \varepsilon]$ cover the range $[w_1 - \varepsilon, w_{m+1} + \varepsilon]$. Let us show that the old range is thus covered:

- Since $w_1 = v_1 - \Delta$, with $\Delta \geq 0$, we have $w_1 \leq v_1$ and $w_1 - \varepsilon \leq v_1 - \varepsilon$.
- Similarly, since $w_{m+1} = w_m + 2\varepsilon = v_m - \Delta + 2\varepsilon$ and $\Delta \leq \varepsilon$, we conclude that $w_{m+1} \geq v_m$ and therefore, $w_{m+1} + \varepsilon \geq v_m + \varepsilon$.

So, $[v_1 - \varepsilon, v_m + \varepsilon] \subseteq [w_1 - \varepsilon, w_{m+1} + \varepsilon]$, and all values are covered by the new approximation. Thus, in this case, the new approximation is indeed almost optimal.

If v_i is in the *upper* half, i.e., if $v_i \in [(2k+1)\cdot\varepsilon, (2k+2)\cdot\varepsilon]$, then we take $\Delta = (2k+2)\cdot\varepsilon - v_i$ (so that $0 \leq \Delta \leq \varepsilon$), use $w_1 = v_1 + \Delta, \ldots, w_m = v_m + \Delta$, and $w_0 = w_1 - 2\varepsilon$. Let us show that these values cover $f'(x)$ for all x (if we show this, then, since this approximation has one more segment than the optimal one, it is almost optimal). Indeed, the original intervals $[v_i - \varepsilon, v_i + \varepsilon]$ covered the range $[v_1 - \varepsilon, v_m + \varepsilon]$. The new intervals $[w_i - \varepsilon, w_i + \varepsilon]$ cover the range $[w_0 - \varepsilon, w_m + \varepsilon]$. Let us show that the old range is thus covered:

- Since $w_m = v_m + \Delta$, with $\Delta \geq 0$, we have $w_m \geq v_m$ and $w_m + \varepsilon \geq v_m + \varepsilon$.
- Similarly, since $w_0 = w_1 - 2\varepsilon = v_1 + \Delta - 2\varepsilon$ and $\Delta \leq \varepsilon$, we conclude that $w_0 \leq v_1$ and therefore, $w_0 - \varepsilon \geq v_1 - \varepsilon$.

So, $[v_1 - \varepsilon, v_m + \varepsilon] \subseteq [w_0 - \varepsilon, w_m + \varepsilon]$, and all values are covered by the new approximation. Thus, in this case, the new approximation is also almost optimal.

In this almost optimal approximation, the values v_i are equal to 0, 2ε, 4ε, etc. Points for which $f'(x) \in [v_i - \varepsilon, v_i + \varepsilon]$ are approximated by the same piece. Therefore, the points x where one piece is changing to another correspond to values $v_i \pm \varepsilon = \pm\varepsilon, \pm 3\varepsilon, \pm 5\varepsilon, \ldots$ In other words, we group together points for which $f'(x) \in [-\varepsilon, \varepsilon]$, for which $f'(x) \in [\varepsilon, 3\varepsilon$, for which $f'(x) \in [3\varepsilon, 5\varepsilon]$, etc.

For clustering to lead to this almost optimal approximation, we must choose the values $k_i = \sqrt{1 + (f'(x))^2}$ which correspond to these thresholds for $f'(x)$, i.e., $k_1 = \sqrt{1+\varepsilon^2}$, $k_2 = \sqrt{1 + 9\varepsilon^2}$, and in general, $k_i = \sqrt{1 + (2i-1)^2 \cdot \varepsilon^2}$.

The corresponding approximating procedure M is straightforward: for a piece on which $(2k-1)\cdot\varepsilon \leq f'(x) \leq (2k+1)\cdot\varepsilon$, we take $2k\cdot\varepsilon$ as the approximating value for the derivative and thus, $2k\cdot\varepsilon\cdot x + C$ for some constant C for $f_{\approx}(x)$; we can determine this constant C, e.g., by the least squares method. The theorem is proven.

Acknowledgments. This work was supported in part by NASA under cooperative agreement NCC5-209, by NSF grant No. DUE-9750858, by the United Space Alliance, grant No. NAS 9-20000 (PWO C0C67713A6), by the Future Aerospace Science and Technology Program (FAST) Center for Structural Integrity of Aerospace Systems, effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-95-1-0518, by the National Security Agency under Grant No. MDA904-98-1-0564, and by the Hong Kong grant RGC4138/97E.

Part of this work was conducted while one of the authors (V.K.) was visiting the Department of Mechanical and Automation Engineering at the Chinese University of Hong Kong under the support of grant RGC4138/97E.

References

[1] J. C. Bezdek, "Numerical taxonomy with fuzzy sets", *Journal of Mathematical Biology*, 1974, Vol. 1, pp. 57–71.

[2] J. C. Bezdek, "Cluster validity with fuzzy sets", *Journal of Cybernetics*, 1973, Vol. 3, No. 3, pp. 58–71.

[3] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*, Plenum, NY, 1981.

[4] J. C. Bezdek, R. Hathaway, M. Sabin, and W. Tucker, "Convergence theory for fuzzy C-Means: counterexample and repairs", *IEEE Trans. Systems, Man, and Cybernetics*, 1987, Vol. SMC-17, pp. 873–877.

[5] J. C. Bezdek, R. Hathaway, M. Sabin, and W. Tucker, "Convergence theory for fuzzy C-Means: counterexample and repairs", In: J. Bezdek (ed.), *The Analysis of Fuzzy Information*, CRC Press, 1987, Vol. 3, Chapter 8.

[6] J. C. Bezdek and S. K. Pal (eds.) *Fuzzy models for pattern recognition*, IEEE Press, N.Y., 1992.

[7] S. Chiu, "Fuzzy model identification based on cluster estimation", *J. of Intelligent and Fuzzy Systems*, 1994, Vol. 2, No. 3, pp. 267–278.

[8] S. Chiu, "Selecting input variables for fuzzy models", *J. of Intelligent and Fuzzy Systems*, 1996, Vol. 4, pp. 243–256.

[9] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters", *Journal of Cybernetics*, 1973, Vol. 3, pp. 32–57.

[10] K. Fukunaga, *Introduction to statistical pattern recognition*, Academic Press, San Diego, CA, 1990.

[11] C. F. Gerald and P. O. Wheatley, *Applied Numerical Analysis*, Addison-Wesley, Reading, MA, 1992.

[12] A. Kandel, *Fuzzy techniques in pattern recognition*, Wiley-Interscience, NY, 1982.

[13] R. R. Yager and D. P. Filev, "Approximate clustering via the mountain method", *IEEE Trans. Systems, Man and Cybernetics*, 1994, Vol. 24, No. 8, pp. 1279–1284.

[14] R. R. Yager and D. P. Filev, "Generation of fuzzy rules by mountain clustering", *Journal of Intelligent and Fuzzy Systems*, 1994, Vol. 2, No. 3, pp. 209–219.

[15] Y. Yam, "Extraction of sparse rule base by Cartesian representation and clustering", In: B. Papadopoulos and A. Syropoulos (eds.), *Current Trends and Developments in Fuzzy Logic, Proceedings of the First International Workshop, Thessaloniki, Greece, October 16–20, 1998* (to appear).