S-truncated Functions and Rough Sets in Approximation and Classification of Bottleneck Polygons

Elisabeth Rakus-Andersson

School of Engineering, Blekinge Institute of Technology,
S-37179 Karlskrona, Sweden
Elisabeth.Andersson@bth.se

Abstract. Some collections of two-dimensional points form very irregular shapes, which cannot be approximated by standard curves without making large errors. We approximate the sets of points to introduce formal mathematical expressions giving rise for future predictions for other points, which are not placed in data sets. To accomplish the thorough approximation of finite point sets we test parametric *s*-truncated functions piecewise, which warrants a high accuracy of approximating. By operating with the functions, which represent samples of points obtained during experiments carried out, and by adopting the rough set technique, we attempt a classification of curves. Even if the curves are stretched and shaped differently we will divide them in classes gathering similar objects. To confirm availability and correctitude of the approximation and the classification proposed, we consider an examination of Internet packet streams, especially a bottleneck distribution based on throughput values.

1 Introduction

Some examinations of the behavior of two variables X and Y provide us with sequences of values x and y, which can be included in the pairs (x, y), treated further as the coordinates of points in the two-dimensional system. We suppose that a finite set A consists of the points (x, y); therefore it can be illustrated as a polygon in which the points are joined by segments of straight lines.

Certain experiments, in which $y \in [-1, 1]$, deliver the polygon (the set A) composed of parts looking like bells (or hills), which lie over and under the x-axis, e.g., like A, sketched in Fig. 1. The polygon, which ties a lot of straight-line bits, cannot constitute a spline interpolation of the points since a number of first degree polynomial equations is too large for further efficient analysis and, moreover, such interpolation is not smooth enough.

The most popular classical method of approximating applied to a set of points is known as the least-square regression that has plenty of modern variants developed lately [2]. Other algorithms of approximating adopt such technical tools as cubic polynomials based on four points [5], tangent curves [1], free algebras [4] or weighted approximations [13].

We consider an approximation of multi-shapes from Fig. 1 by s-truncated functions used piecewise as another approach to the numerical problem of a smooth curve fitting to point sets. Since the y-coordinates of the points constituting the elements of A belong to the interval [0, 1] or [–1, 0], then it will be desirable to approximate pieces of a polygon representing A by a function, which takes its values in [0, 1]. This property is characteristic of the s-function. The procedure of forming the approximation of A by truncated s-functions tied by pieces of straight lines, if needed, is developed in the second section of the paper. "The sampled truncated s", as we call an entire approximation curve, consists of first and second degree-polynomials. It follows the polygon's shape very closely and it cumulates a very low error measuring deviations between the approximating curve and the polygon. This should be regarded as an important advantage of the sampled truncated approximation method in comparison with other procedures. We should add that the approach proposed as the approximation of non-standard curve shapes contributes an original own solution in numerical mathematics.

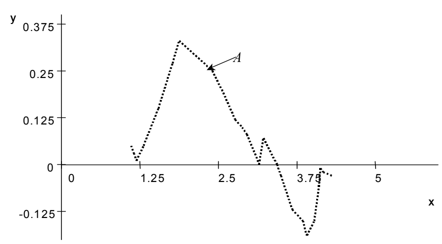

Fig. 1. The example of the multi-shaped polygon reflecting $A = \{(x, y)\}$

Let us assume that the y-values of curves, that are similar in shape to the set depicted in Fig. 1, are important indicators in the further classification process of the curves. These can resemble some letters, e.g., N, W, or M, and can occur in different places along the x-axis. In order to assign the curves to proper classes denoted by N, W or M we should compare their y-coordinates. It is not possible if the curves under consideration are scattered in different segments of the x-axis. To be able to make the curves comparable, which aims at estimating their deviations in y-values stated at the same x, we should move the curves over the interval [0, 1]. The own procedure transfers the curves over a common interval and preserves their original shapes. We outline all derived transformations in Section 3.

All assumptions made for the shapes of curves correspond to a problem of complex bottleneck recognition. The bottleneck data provided as sets of points result in obtaining curves, which are distinguished as three main shapes of the letters N, M and W. The three letters characterize the main bottleneck classes, which gather some objects possessing similar shapes. The patterns sometimes are mixed and they do not resemble basic letters but we still want to find for them an appropriate class, which has most of their attributes. We accomplish the bottleneck classification as a practical example that reveals the utility of a very thorough approximation of point sets combined with rough set techniques. The bottleneck classification, as one of the unsolved Internet problems, has stimulated us to develop the final solution, which appears in the fourth section.

2 Sampled Truncated *S*-functions in the Approximation of a Clock-like Polygon

The approach to approximation of irregular polygons presented below constitutes an own original solution [11, 12], which differs from other procedures of seeking approximation curves [1, 2, 4, 5, 13].

We discover that the x-values of pairs included in A belong to the interval $[x_{\min}(A), x_{\max}(A)]$, in which $x_{\min}(A)$ is the smallest and $x_{\max}(A)$ is the largest x-value in A. In the next step we divide the total interval in subintervals $[x_{\min(A_j)}, x_{\max(A_j)}]$, where A_j, j = 1, 2, …, Q, are parts of A. In the parts A_j we can experience either the growth or the decrease of the y-values corresponding to these x that are placed between the borders $x_{\min(A_j)}$ and $x_{\max(A_j)}$ standing for the smallest and, respectively, the largest value of x in A_j. *S*-functions or segments of straight lines attached to two adjacent *s*-curves approximate the A_j components.

Example 1

The pairs, which create the polygon depicted in Fig. 1, are the members of A = {(1.1, 0.05), (1.15, 0.03), (1.19, 0.01), (1.3, 0.05), (1.54, 0.15), (1.76, 0.27), (1.87, 0.33), (2.4, 0.25), (2.55, 0.2), (2.76, 0.12), (2.87, 0.1), (2.96, 0.08), (3.1, 0.02), (3.14, 0), (3.21, 0.07), (3.48, 0), (3.49, –0.03), (3.67, –0.12), (3.84, –0.15), (3.9, –0.19), (4.02, –0.15), (4.09, –0.06), (4.12, –0.01), (4.16, –0.02), (4.3, –0.03)}. By measuring the direction of changes in the y-values, which point out extreme nodes in A's shape, we consider the subintervals [1.1, 1.19], [1.19, 1.3], [1.3, 1.87], [1.87, 3,14], [3.14, 3.21], [3.21, 3.43], [3.43, 3.9], [3.9, 4.12], [4.12, 4.16], [4.16, 4.3]. Over the intervals either *s*-functions or straight lines will be applied as approximation tools.

The *s*-function with the standard parameters α, β, γ, introduced by [6, 7, 14, 15] as

$$y = s(\alpha,\beta,\gamma) = \begin{cases} (1) \quad \varepsilon\left(2\left(\dfrac{x-\alpha}{\gamma-\alpha}\right)^2\right) & \text{for} \quad \alpha \le x < \beta \\ (2) \quad \varepsilon\left(1-2\left(\dfrac{x-\gamma}{\gamma-\alpha}\right)^2\right) & \text{for} \quad \beta \le x \le \gamma \end{cases} \tag{1}$$

is fitted best for the appearances of the "half-bells" A_j. Since the y-values of the classical s belong to the interval [0, 1] ($-s$ has its y-values in [–1, 0]) then we should insert an additional parameter ε in (1) in order to accommodate a height of the function to the real data existing in the set A_j, $j = 1, 2, \ldots, Q$. The parameter β is estimated as the arithmetic mean of α and γ. We have already introduced the partition of A by means of the subsets A_j, looking like "half-bells", then we should denote each s-function that approximates A_j by $s_{A_j}(x, \alpha_{A_j}, \beta_{A_j}, \gamma_{A_j}, \varepsilon_{A_j})$.

Example 2

We intend to recall in mind how the s-function (1) looks like. If we choose, e.g., $\alpha = 0$, $\gamma = 2.1$, $\beta = 1.05$ and $\varepsilon = 0.26$ as some casual numbers, then the function will have a graph drawn in Fig. 2.

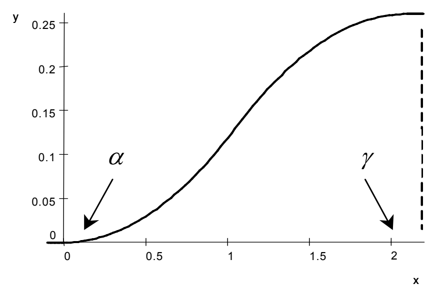

Fig. 2. The s-function for $\alpha = 0$, $\gamma = 2.1$ and $\varepsilon = 0.26$

In the next part of the chapter we discuss different cases of A_j's approximation, which is dependent on the sizes of y-coordinates in the set A_j.

Let us assume that the values of the y-coordinates in A_j associated with the x-values belonging to $\lfloor x_{\min(A_j)}, x_{\max(A_j)} \rfloor$ appear in the ascending order, and let us notice that no y-coordinate is equal to zero. The pair $(x_{\min(A_j)}, y(x_{\min(A_j)}))$ (($y(x_{\min(A_j)})$ corresponds to $x_{\min(A_j)}$) begins the set A_j but we cannot identify $x_{\min(A_j)}$ as α_{A_j}. Thus, the value of α_{A_j} in the s_{A_j}-function, which is expected to approximate A_j, is unknown. To find α_{A_j} we, at first, accept the value of ε_{A_j} as the largest y-coordinate in A_j, which corresponds to the x-coordinate taken as γ_{A_j}. We can now reconstruct the value of the remaining parameter α_{A_j}, according to the following patterns:

a) $\alpha_{A_j} = \dfrac{x_{\min(A_j)} - \gamma_{A_j}\sqrt{\frac{y(x_{\min(A_j)})}{2}}}{1 - \sqrt{\frac{y(x_{\min(A_j)})}{2}}}$ for $y(x_{\min(A_j)}) < \dfrac{\varepsilon}{2}$. This modifies (1) as

$$y = \begin{cases} (1) \quad \varepsilon_{A_j}\left(2\left(\dfrac{x-\alpha_{A_j}}{\gamma_{A_j}-\alpha_{A_j}}\right)^2\right) & \text{for} \quad x_{\min(A_j)} \le x < \beta_{A_j} \\ (2) \quad \varepsilon_{A_j}\left(1-2\left(\dfrac{x-\gamma_{A_j}}{\gamma_{A_j}-\alpha_{A_j}}\right)^2\right) & \text{for} \quad \beta_{A_j} \le x \le \gamma_{A_j}. \end{cases} \tag{2}$$

b) $\alpha_{A_j} = \gamma_{A_j} - \dfrac{\gamma_{A_j} - x_{\min(A_j)}}{\sqrt{\frac{1-y(x_{\min(A_j)})}{2}}}$ for $y(x_{\min(A_j)}) \ge \dfrac{\varepsilon}{2}$. The $s_{A_j}(x)$ formula appears as

$$y = \begin{cases} (1) \quad 0 & \text{for} \quad x < x_{\min(A_j)} \\ (2) \quad \varepsilon_{A_j}\left(1-2\left(\dfrac{x-\gamma_{A_j}}{\gamma_{A_j}-\alpha_{A_j}}\right)^2\right) & \text{for} \quad x_{\min(A_j)} \le x \le \gamma_{A_j}. \end{cases} \tag{3}$$

It happens that the position of pairs in the set A_j introduces the descending order among points with respect to the *y*-coordinate values. We assume that none of them is equal to zero. The pair $(x_{\max(A_j)}, y(x_{\max(A_j)}))$ will end the set A_j, but $x_{\max(A_j)} \neq \gamma_{A_j}$. Let us assign the largest value of *y* in A_j, regarded as ε_{A_j}, to the *x*-coordinate $x_{\min(A_j)} = \alpha_{A_j}$. Then it is possible to restore the missing value of γ_{A_j}, which is one of the parameters included in a function $1 - s_{A_j}(x, \alpha_{A_j}, \beta_{A_j}, \gamma_{A_j}, \varepsilon_{A_j})$ applied to approximate A_j.

We make the following distinction between two different cases of adjusting the parameter γ_{A_j} to the data set A_j:

c) $\gamma_{A_j} = \dfrac{x_{\max(A_j)} - \alpha_{A_j}\sqrt{\frac{y(x_{\max(A_j)})}{2}}}{1-\sqrt{\frac{y(x_{\max(A_j)})}{2}}}$ for $y(x_{\max(A_j)}) < \dfrac{\varepsilon}{2}$. We suggest the following changes in (1) to adapt it to the new assumptions

$$y = \begin{cases} (1) \quad \varepsilon_{A_j}\left(1-2\left(\dfrac{x-\alpha_{A_j}}{\gamma_{A_j}-\alpha_{A_j}}\right)^2\right) & \text{for} \quad \alpha_{A_j} \le x < \beta_{A_j} \\ (2) \quad \varepsilon_{A_j}\left(2\left(\dfrac{x-\gamma_{A_j}}{\gamma_{A_j}-\alpha_{A_j}}\right)^2\right) & \text{for} \quad \beta_{A_j} \le x \le x_{\max(A_j)}. \end{cases} \tag{4}$$

d) $\gamma_{A_j} = \alpha_{A_j} + \dfrac{x_{\max(A_j)} - \alpha_{A_j}}{\sqrt{\frac{1-y(x_{\max(A_j)})}{2}}}$ for $y(x_{\max(A_j)}) \geq \dfrac{\varepsilon}{2}$. We adjust the $s_{A_j}(x)$ formula as

$$y = \begin{cases} (1) \quad \varepsilon_{A_j}\left(1 - 2\left(\dfrac{x - \alpha_{A_j}}{\gamma_{A_j} - \alpha_{A_j}}\right)^2\right) & \text{for} \quad \alpha_{A_j} \leq x \leq x_{\max(A_j)} \\ (2) \quad 0 & \text{for} \quad x > x_{\max(A_j)}. \end{cases} \tag{5}$$

The modified s_{A_j} constitutes a section of the classical *s*-function; therefore we will name it a truncated *s*-function. By selecting the minimal and the maximal *x*-value and the maximal *y*-value, which exist in the set A_j, we prepare the mathematical apparatus with (2)–(5) for computing unknown parameters α_{A_j} or γ_{A_j}. The point, in which the *y*-coordinate takes the ε_{A_j}-value and the *x*-coordinate is equal to the γ_{A_j} value for $s_{A_j}(\alpha_{A_j}, \beta_{A_j}, \gamma_{A_j}, \varepsilon_{A_j})$ respectively the α_{A_j} value for $1 - s_{A_j}(\alpha_{A_j}, \beta_{A_j}, \gamma_{A_j}, \varepsilon_{A_j})$, is one of the vertices in *A* and it constitutes the common element of A_j and the function s_{A_j}, $j = 1, \ldots, Q$. The total approximation s_A of *A* is called "sampled truncated *s*".

To preserve the right shape of the approximating curve, it is advisable to tie two adjacent functions s_{A_j}, $s_{A_{j+1}}$ between the points $(x_{\max(A_j)}, y(x_{\max(A_j)}))$, $(x_{\min(A_{j+1})}, y(x_{\min(A_{j+1})}))$ by the equation of a straight line in the form

$$y = line_{A_j}(x) = k_{A_j} x + l_{A_j} \quad \text{for} \quad x_{\max(A_j)} \leq x < x_{\min(A_{j+1})}. \tag{6}$$

Example 3
The "sampled truncated *s*", accommodated to the data from Ex.1, is shown in Fig. 3.

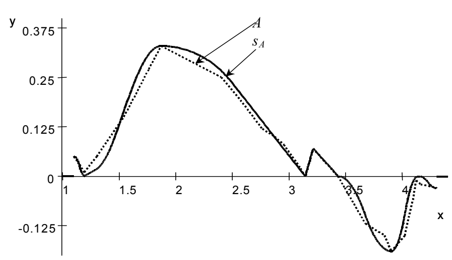

Fig. 3. The approximation of *A* by "sampled truncated *s*"

The first set of points $A_1 \subset A$, in which the y-coordinates establish the descending order, is placed over [1.1, 1.19]. Since no y-value is equal to zero we will reconstruct $\gamma_{A_1} = \frac{1.19 - 1.1\sqrt{\frac{0.01}{2}}}{1-\sqrt{\frac{0.01}{2}}} = 1.1968$ for $\varepsilon_{A_1} = 0.05$, $\alpha_{A_1} = 1.1$, $x_{\max(A_1)} = 1.19$ and $y(x_{\max(A_1)}) = 0.01$ in accordance with c).

The formula of s_A for A is expanded as the following split definition

$$y = s_A(x) = \begin{cases} 0.05(1-2(\frac{x-1.1}{1.1-1.1968})^2) & \text{for } 1.1 \le x < 1.1484 \\ 0.05(2(\frac{x-1.1968}{1.1-1.1968})^2) & \text{for } 1.1484 \le x < 1.19 \\ 0.15886x - 0.18855 & \text{for } 1.19 \le x < 1.3 \\ 0.33(2(\frac{x-1.0043}{1.87-1.0043})^2)^2 & \text{for } 1.3 \le x < 1.4372 \\ \ldots & \ldots \quad \ldots \\ -0.03(1-2(\frac{x-4.3}{4.1444-4.3})^2) & \text{for } 4.2222 \le x < 4.3. \end{cases}$$

We can prove some additional operations on the s-function values, e.g., $y = (s(x))^2$ or $y = (s(x))^{\frac{1}{2}}$ to match a shape of the function to the given polygon in the best way.

It is worth noticing that the total error that collects the deviations of $s_A(x)$ from A is very small. This is entailed in further analysis of curves, where even small differences between approximating curves and point sets can lead to wrong conclusions.

3 Sampled *S*-functions over the Interval [0, 1]

The curve created for A has a particular pattern since it resembles the letter *N*. In some technical problems we obtain the sets of points, which will be approximated by three shapes of letters, namely, *N*, *M* and *W*. The shapes of mentioned letters can be disturbed or vague, which makes difficult to classify them properly, i.e., we do not know exactly how to include the curves in classes determined by *N*, *M* and *W*. In order to ensure if a vague or unknown object can belong to the considered class or not, we accomplish a classification according to the rules of Rough Set Theory. Definitely, we can model some other shapes as well but we need the mentioned letters in the Internet unsolved problem, which inspires us to lead the further discussion.

If we are given several polygons then we, at the first stage, want to collect all approximated objects over a common interval [0, 1] to measure their deviations in y-values with respect to the same x value.

Suppose that we have obtained different shapes of the curves originating from the point sets A^1–A^5. Each of them is approximated by a continuous function, which consists of s-sections and pieces of straight lines that link the parts of s-functions if it is necessary. Figure 4 constitutes a picture, which reveals the polygons and the approximating functions over their original intervals along the x-axis. We describe the polygon memberships in the following way: A^1, A^3 and A^5 belong to the "*N*" class, A^4 is a member of the "*W*" class while the origin of A^2 is unknown.

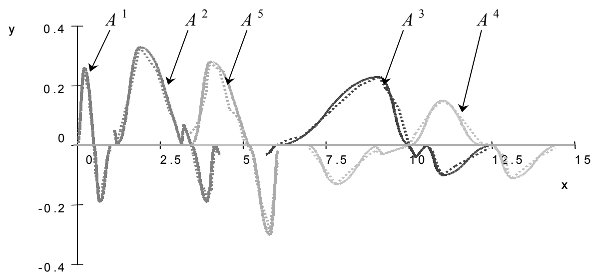

Fig. 4. The approximated polygons A^1–A^5

In the further analysis we use only the continuous curves, also named A^1–A^5.

To move all curves to the same start point settled as the origin of the *xy* coordinate system, we suggest the following transformations.

Suppose that the A^i-curve, $i = 1, \ldots, 5$, is placed in the *x*-interval $\left[x_{\min}(A^i), x_{\max}(A^i)\right]$. We move the j^{th} segment $s_{A^i_j}$, approximating the subset A^i_j of A^i, $i = 1, \ldots, 5, j = 1, \ldots, Q$, to a position close to the origin by introducing the formula

$$y = \begin{cases} (1)\ \ \varepsilon_{A^i_j}\left(2\left(\dfrac{x-(\alpha_{A^i_j}-x_{\min}(A^i))}{\gamma_{A^i_j}-\alpha_{A^i_j}}\right)^2\right) & \text{for } x_{\min(A^i_j)} - x_{\min}(A^i) \le x < \beta_{A^i_j} - x_{\min}(A^i) \\ (2)\ \ \varepsilon_{A^i_j}\left(1-2\left(\dfrac{x-(\gamma_{A^i_j}-x_{\min}(A^i))}{\gamma_{A^i_j}-\alpha_{A^i_j}}\right)^2\right) & \text{for } \beta_{A^i_j} - x_{\min}(A^i) \le x \le \gamma_{A^i_j} - x_{\min}(A^i). \end{cases} \tag{7}$$

The function (7), in which $x_{\min(A^i_j)}$ is the least *x*-value in the set of pairs A^i_j, should be previously suited to A^i_j by applying (2) (or (3)). The choice of the (1–*s*)-function induces the application of (4) (or (5)).

The straight line (6) is transferred nearby the origin by the action of an equation

$$
\begin{aligned}
y &= line_{A^i_j}(x) = \\
&= k_{A^i_j} x + l_{A^i_j} + k_{A^i_j} \cdot x_{\min}(A^i) \quad \text{for} \quad x_{\max(A^i_j)} - x_{\min}(A^i) \le x < x_{\min(A^i_{j+1})} - x_{\min}(A^i) \\
&= K_{A^i_j} x + L_{A^i_j}.
\end{aligned} \tag{8}
$$

Example 4
Figure 5 shows A^1–A^5 attached to the origin after performing (7), (8).

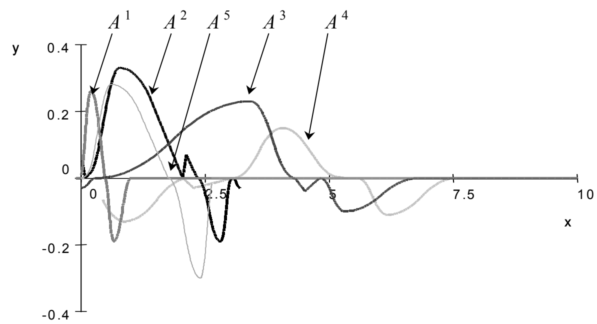

Fig. 5. The curves A^1–A^5 with their start points at the origin

In Fig. 5 we recognize A_2 as A from Ex. 1. We decide $x_{\min}(A^2) = 1.1$ and modify "sampled truncated s" for $A_2 = A$, as a function

$$
y = s_{A^2}(x) = \begin{cases}
0.05(1 - 2(\frac{x-(1.1-1.1)}{1.1-1.1968})^2) & \text{for} \quad 1.1 - 1.1 \le x < 1.1484 - 1.1 \\
0.05(2(\frac{x-(1.1968-1.1)}{1.1-1.1968})^2) & \text{for} \quad 1.1484 - 1.1 \le x < 1.19 - 1.1 \\
0.15886x - 0.18855 + 0.15886 \cdot 1.1 & \text{for} \quad 1.19 - 1.1 \le x < 1.3 - 1.1 \\
0.33(2(\frac{x-(1.0043-1.1)}{1.87-1.0043})^2)^2 & \text{for} \quad 1.3 - 1.1 \le x < 1.4372 - 1.1 \\
\ldots & \ldots \quad \ldots \\
-0.03(1 - 2(\frac{x-(1.43-1.1)}{4.1444-4.3})^2) & \text{for} \quad 4.2222 - 1.1 \le x < 4.3 - 1.1
\end{cases}
$$

which displaces A_2's beginning to the origin.

The comparison of all curves will be successful if we can observe them at a common interval. Let us determine the interval [0, 1] as a new domain for all split-functions A^1–A^5. Each piece $s_{A^i_j}$ or $line_{A^i_j}$, $i = 1, \ldots, 5, j = 1, \ldots, Q$, should be shrunk or widened proportionally to fit it for the interval [0, 1] together with other pieces.

In order to achieve the required movements of s^i_j over [0, 1], we initiate the parameter $\delta_{A^i} = \dfrac{1}{x_{\max}(A^i) - x_{\min}(A^i)}$ in (7), which generates a new formula [6]

$$y = \begin{cases} (1) \quad \varepsilon_{A_j^i}\left(2\left(\dfrac{x-(\alpha_{A_j^i}-x_{\min}(A^i))\delta_{A^i}}{(\gamma_{A_j^i}-\alpha_{A_j^i})\delta_{A^i}}\right)\right) \\ \text{for} \quad (x_{\min(A_j^i)}-x_{\min}(A^i))\delta_{A^i} \le x < (\beta_{A_j^i}-x_{\min}(A^i))\delta_{A^i} \\ (2) \quad \varepsilon_{A_j^i}\left(1-2\left(\dfrac{x-(\gamma_{A_j^i}-x_{\min}(A^i))\delta_{A^i}}{(\gamma_{A_j^i}-\alpha_{A_j^i})\delta_{A^i}}\right)\right) \\ \text{for} \quad (\beta_{A_j^i}-x_{\min}(A^i))\delta_{A^i} \le x \le (\gamma_{A_j^i}-x_{\min}(A^i))\delta_{A^i}. \end{cases} \tag{9}$$

Before equipping Eq. (8) with the parameter δ_{A^i} we should find another form of (8), adapted to the range [0, 1] as

$$y = K_{A_j^i}x + L_{A_j^i} = \frac{x + \dfrac{L_{A_j^i}}{K_{A_j^i}}}{\dfrac{1}{K_{A_j^i}}} \quad \text{for} \quad x_{\max(A_j^i)} - x_{\min}(A^i) \le x < x_{\min(A_{j+1}^i)} - x_{\min}(A^i). \tag{10}$$

We can now place δ_{A^i} in (10) according to a pattern

$$y = \frac{x + \dfrac{L_{A_j^i}\delta_{A^i}}{K_{A_j^i}}}{\dfrac{\delta_{A^i}}{K_{A_j^i}}} \quad \text{for} \quad (x_{\max(A_j^i)} - x_{\min}(A^i))\delta_{A^i} \le x < (x_{\min(A_{j+1}^i)} - x_{\min}(A^i))\delta_{A^i}. \tag{11}$$

Example 5

The applications of (9) and (11) to every *s*-section and every line segment that takes place in A^1–A^5, yields the effect of collecting all curves over the *x*-domain [0, 1]. The curves A^1–A^5, lying in [0, 1], are plotted in Fig. 6. The new formula of A^2

$$y = s_{A^2}(x) = \begin{cases} 0.05(1-2(\frac{x-(1.1-1.1)0.31}{(1.1-1.1968)0.31})^2) & \text{for} \quad (1.1-1.1)0.31 \le x < (1.1484-1.1)0.31 \\ 0.05(2(\frac{x-(1.1968-1.1)0.31}{(1.1-1.1968)0.31})^2) & \text{for} \quad (1.1484-1.1)0.31 \le x < (1.19-1.1)0.31 \\ \dfrac{x+(\frac{-1.3804\cdot 10^{-2}}{0.15886})0.31}{(\frac{1}{0.15886})0.31} & \text{for} \quad (1.19-1.1)0.31 \le x < (1.3-1.1)0.31 \\ 0.33(2(\frac{x-(1.0043-1.1)0.31}{(1.87-1.0043)0.31})^2)^2 & \text{for} \quad (1.3-1.1)0.31 \le x < (1.4372-1.1)0.31 \\ \cdots & \cdots \quad \cdots \\ -0.03(1-2(\frac{x-(1.43-1.1)0.31}{(4.1444-4.3)0.31})^2) & \text{for} \quad (4.2222-1.1)0.31 \le x < (4.3-1.1)0.31, \end{cases}$$

in which $\delta_{A^2} = \frac{1}{4.3-1.1} \approx 0.31$, gives an idea of executed transformations.

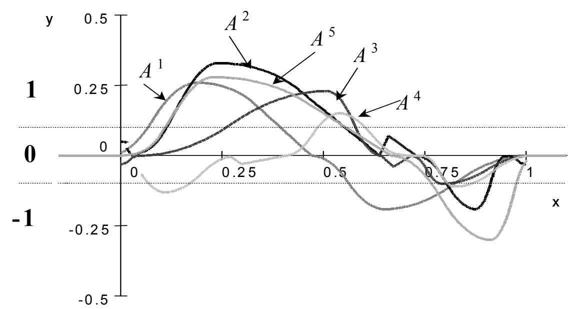

Fig. 6. The curves A^1–A^5 over the common interval [0, 1]

The mathematical tools used for polygons result in the creation of a common collection of curves, which represent the polygons over a substantial part of the *x*-axis determined as the interval [0, 1]. By researching the *y*-values of continuous functions corresponding to selected *x*-values, we can compare data saved in an indiscernibility relation. The relation is a crucial part of further empirical examinations in which the selected elements of Rough Set Theory constitute a foundation.

4 Rough Set Theory in Polygon Classification

The item of bottleneck identification and classification has given rise to mathematical investigations and developments discussed in former sections. Let us shortly describe the nature of bottleneck occurrence without deep discussions, which can be found in technical literature [3].

Advanced IP (Internet Protocol) network applications such as IP videoconferences, Voice-over-IP, or on-line games, involve IP network operations. These generate data streams, which are sensitive to specific delay and throughput requirements. Different packets in the networks can experience significant delays and interruptions. In the fluid flow model, the delays or other disturbances lead to variations in the throughput streams. The noisy variations, called bottlenecks, are marked by sudden changes of the linear flow. Since this paper presents a mathematical solution of the bottleneck problem, then we, without interpreting technical experiments that collect bottleneck data, should focus on final objects of telecommunication investigations. These result in *Throughput Difference Plots* [3]. The plots determine finite sets of pairs of the type $A = \{(x, y)\}$ = {(*bit rates in a time interval, differences between output and input flow bottleneck for observed bit* *rates*)}, $y \in [-1, 1]$.

The "shape" of a bottleneck contains information of its nature. We assign the letter of "*M*" to the shared bottleneck, the "*W*" letter to a shaping bottleneck and, finally, "*N*" stands for an overloaded bottleneck [3]. In order to improve the quality of streams the unfavorable occurrence of bottlenecks should be removed. Operators try to diminish the negative effects of bottlenecks, e.g., by overloading them or by choosing special parameter systems that are applied to streams sent to a receiver. In all cases, however, the bottleneck must be identified since an appropriate method of eliminating depends on its sort. The shapes of some bottlenecks often are undisturbed and we can include them to the classes *N*, *M*, *W* without difficulties. Other curves representing bottlenecks are interacted by noisy signals. These constitute confusing elements in the clear recognition of the bottlenecks' origins.

In order to include unknown sets defined as "*bottleneck*" within classes *N*, *M* and *W* already possessing the declared members, we apply some elements of Rough Set Theory [7, 8, 9, 10], which have proven useful in the process of a polygon classification [12].

The *y*-axis in Fig. 6 is divided in three regions. After analyzing the nature of noisy signals that appear in bottlenecks, we consider three intervals for *y*. The values of *y* belong to the interval (–0.3, 0.35) in the recognized case. The signals, noted as the *y*-values, occurring from –0.1 to 0.1 cannot provide us with essential information about the nature of a bottleneck and they are ignored. As a consequence, the code assigned to the *y*-value belonging to [–0.1, 0.1] is equal to 0. For decisive, positive *y*-values the code of 1 is reserved, while the negative *y*-values of a deterministic character obtain the code stated as –1.

Each considered polygon has an envelope created by a continuous function that approximates it over [0, 1]. For every value *x* belonging to [0, 1] we can establish the association between it and one of the codes. We should at first compute the *s*(*x*) value and then place the value in one of the intervals associated with the codes, already announced by Fig. 6.

Let us introduce a universe set $U=\{A^1, A^2, \ldots, A^5\}$ composed of the polygons representing the sets of the "*bottleneck*" type. The objects of *U* are determined by two groups of attributes, so called condition and decision attributes presented by the sets *B* and *D* respectively. We assume that the set *B* consists of sizes $x_k \in [0,1]$, $k = 1, \ldots, m$, associated with values $code_{A^i}(x_k)$, $i = 1, \ldots, 5$, which are equal to the integers –1, 0 and 1.

Since we want to assign some members to the "*N*" class, then the set *D* obtains an attribute stated as "the membership of a polygon in "*N*"", where the membership is expressed as "*yes*", "*no*", "*unknown*".

The triple $I=(U, B, D)$ forms the decision table, which is treated as the data basis for an equivalence relation $I(B)$ called the indiscernibility relation and defined by a relationship

$$I(B) = \{(A^i, A^s) : code_{A^i}(x_k) = code_{A^s}(x_k)\} \text{ for each size } x_k , \qquad (12)$$

where $k = 1, 2, \ldots, m,\ i,\ s = 1, 2, \ldots, 5$.

We find the equivalence classes of the relation $I(B)$, i.e., the blocks $IB(A^i)$ as the sets

$$IB(A^i) = \{A^s : (A^i, A^s) \in I(B)\}. \tag{13}$$

By following a general rough set procedure we create a set $X = \{A^i$: which have the decision "*yes*" assigned}.

The first decision set (the lower approximation of X)

$$B_*(X) = \{A^i : IB(A^i) \subseteq X\} \tag{14}$$

reveals the curves (thus polygons), which match the "*N*" class without doubts (they are sure members of *N*).

The other decision set (the upper approximation of X)

$$B^*(X) = \{A^i : IB(A^i) \cap X \neq 0\} \tag{15}$$

contains the members of *U*, which may belong to the considered class "*N*".

The elements of a boundary set

$$B_{border}(X) = B^*(X) - B_*(X) \tag{16}$$

are the members of "*N*" in a certain grade.

The membership degree of A^i, interpreted as a degree of being a member in "*N*", is computed as

$$\mu_{"N"}(A^i) = \frac{|X \cap IB(A^i)|}{|IB(A^i)|}. \tag{17}$$

Example 6

We regard the data concerning A^1–A^5 and sampled in Fig. 6 as pictures of different bottleneck cases. We state $U = \{A^1, A^2, A^3, A^4, A^5\}$. The decision triple $I = (U, B, D)$ is expanded in Table 1.

Table 1. The decision table $I = (U, B, D)$

$A^i \backslash x_k$	0.125	0.250	0.375	0.500	0.625	0.750	0.875	Class"*N*"
A^1	1	1	1	0	–1	–1	0	*yes*
A^2	1	1	1	1	0	0	–1	*unknown*
A^3	0	1	1	1	0	0	0	*yes*
A^4	–1	0	0	1	0	0	0	*no*
A^5	1	1	1	1	0	0	–1	*yes*

The equivalence relation $I(B)$, provided in accordance with (12), is formed by a set of pairs $I(B) = \{(A^1, A^1), (A^2, A^2), (A^3, A^3), (A^4, A^4), (A^5, A^5), (A^2, A^5), (A^5, A^2)\}$.

The equivalence classes of $I(B)$ are decided as the sets

$IB(A^1) = \{A^1\},\ IB(A^2) = \{A^2, A^5\},\ IB(A^3) = \{A^3\},\ IB(A^4) = \{A^4\},\ IB(A^5) = \{A^2, A^5\}$.

The value of the decision attribute "*N*" = "*yes*" generates the set $X = \{A^1, A^3, A^5\}$, which in turn is an essential factor implementing $B_*(X) = \{A^1, A^3\}$, $B^*(X) = \{A^1, A^2, A^3, A^5\}$ and $B_{border}(X) = \{A^2, A^5\}$.

The polygon membership degrees, whose sizes confirm the membership in the "*N*" class, are obtained as $\mu_{"N"}(A^1) = 1,\ \mu_{"N"}(A^2) = \frac{1}{2},\ \mu_{"N"}(A^3) = 1,\ \mu_{"N"}(A^4) = 0, \mu_{"N"}(A^5) = \frac{1}{2}$.

By looking at the results of the accomplished analysis we can conclude that A^1 and A^3 are the true members of "*N"*-class in *U*, while A^2 and A^5 may belong to the investigated class to certain grades. We can also notice that A^2 affects a status of A^5 negatively and, on the contrary, we can see that A^5 upgrades an importance of A^2 in the "*N*"-class. A_2, which has not been recognized at the first stage of classification, has joined the group of possible members of "*N*". The recognition of the bottleneck nature aims at the special treatment of all sure and possible objects belonging to "*N*" to get rid of their negative effects in a flow of signals. Operators who deal with parameter sets applied to disturbances know how to eliminate the members of the "*N*" group, on condition that they are aware of the class contents.

5 Conclusions

Some finite sets of pairs are often interpolated by polygons, which seldom have convenient equations mathematically expanded. Although there exists a large number of approximation methods applied to point sets, especially the different variations of least square regressions, we suggest applying a new procedure of approximation. This originates from the standard *s*-functions in truncated forms that approximate the irregular parts of the polygons very smoothly.

The functions, called by us "the sampled, truncated *s*" are composed of the first and second degree-polynomials in the form of split definitions. The low degrees of approximating functions make further operations on them rather easy, which is an essential advantage of the method. One truncated *s*-segment can approximate many nodes belonging to the point set, which reduces a number of piecewise functions involved in the general definition of an approximating collection. But most of all we notice that "the sampled, truncated *s*" follows the changes of the polygon's pattern very sensitively, which guarantees the high thoroughness of approximation results.

A new process of approximation sometimes is invented in mathematics as an interesting theoretical item without greater practical validity. To prove the empirical aspect of "sampled truncated *s*" we want to consider an Internet bottleneck praxis having been solved statistically. The probabilistic solutions could not give answers to all posed questions about classification of imprecise objects. The objects, collected as sets of points and approximated by *s*-functions are moved over the interval [0, 1]. The set [0, 1] is approved as an appropriate domain for all functions in order to compare them. The operations on functions, which transfer the curves over [0, 1], are the own contributions in the solution.

The accomplishment of a successful classification of unknown objects, possessing only some features typical of the considered class, is not an easy task. By applying the Rough Set Theory combined with earlier achievements in approximation, we could classify polygons within the same class even if they have an unknown origin. Two introduced sets B_* and B^* act as a lower and an upper approximation of the investigated class. This makes possible to assign its sure members as well as such ones that have most of the properties characteristic of the class. Moreover, we can easily exclude the polygons, which do not satisfy the class's attributes.

If we need a pattern for another classification of objects obtained as some point sets, we can return to the discussed model and adapt it to other assumptions.

References

1. Agadi, A., Penot, J-P.: A comparative Study of Various Notions of Approximation of Sets. Journal of Approximation Theory (2005) in Press
2. Cherkassky, V., Gehring, D., Mulier, F.: Comparison of Adaptive Methods for Function Estimation for Samples. IEEE Transactions on Neural Networks, vol. 7, no 4 (1996) 969-984
3. Fiedler, M., Tutschku, K., Carlsson P.: Identification of Performance Degradation in IP Networks Using Throughput Statistics. In: Charzinski, J., Lehnert, R., Tran-Gia, P. (eds.): Providing Quality of Service in Heterogeneous Environments (2003) 399-408
4. Haan, O.: A free Algebraic Solution for the Planar Approximation. Nuclear Physics B 705 (2005) 563-575
5. Lavoue, G., Dupont, F., Baskurt, A.: A New Subdivision Based Approach for Piecewise Smooth Approximation of 3D Polygonal Curves. Journal of Pattern Recognition Society (2005) in Press
6. Novák, V., Perfilieva, I.: Evaluating of Linguistic Expressions and Functional Fuzzy Theories in Fuzzy Logic. In: Zadeh, L. A., Kacprzyk, J. (eds.): Computing with Words in Information – Intelligent Systems 2, vol. 33, Studies in Fuzziness and Soft Computing, Physica verlag (1999) 383-406
7. Pal, S. K., Mitra, P.: Case Generation Using Rough Sets with Fuzzy Representation. IEEE Transactions on Knowledge and Data Engineering, vol. 16, no 3 (2004) 292-300
8. Pawlak, Z.: On Rough Sets. Bulletin of the EATCS 24, (1984) 94-108
9. Pawlak, Z.: Vagueness - a Rough Set View. Structures in Logic and Computer Science (1997) 106-117
10. Pawlak, Z.: Decision Networks. Proceedings of Rough Sets and Current Trends in Computing, Uppsala, Sweden (2004) 1-7
11. Rakus-Andersson, E., Salomonsson, M.: The Truncated π-functions in Approximation of Multi-shaped Polygons. Current Issues in Data and Knowledge Engineering, Proceedings of EUROFUSE 2004 – EURO WG on Fuzzy Sets, Warszawa, 22-25 September (2004) 444-452
12. Rakus-Andersson, E., Salomonsson, M.: Π-truncated functions and Rough Sets in Classification of Internet Protocols. Accepted by IFSA 2005

13. Salmeri, M., Mencattini, A., Rovatti, R.: Function Approximation Using Non-normalized SISO Fuzzy Systems. International Journal of Approximate Reasoning 26 (2001) 211-231
14. Zadeh, L. A.: Calculus of Fuzzy Restrictions. Fuzzy Sets and Their Applications to Cognitive and Decision Processes, London, Academic Press (1975)
15. Zadeh, L. A.: Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. IEEE Trans. Systems, vol. 3 (1973) 28-44