

Copyright © 2003.
This work is reproduced in good faith. Every reasonable effort has been made to trace the copyright owner. For more information please contact researchbank@swin.edu.au.
Polynomial Splines and Data Approximation

Nadejda Soukhoroukova

Centre for Informatics and Applied Optimization,
School of Information Technology and Mathematical Sciences,
University of Ballarat, Victoria 3353, Australia

Abstract

The problem of data approximation is of great interest. There are a lot of approaches to solve this problem. One of them is a polynomial spline approximation. In this paper we propose a new algorithm for polynomial spline approximation based on nonsmooth optimization techniques. Numerical experiments using this algorithm have been carried out. The results are presented and discussed.

1 Introduction.

The problem of approximation for given a function of one variable or data in two-dimensional space was studied by many authors (see for example [1], [11]-[16] and references therein). Several methods based on quite different approaches were proposed and studied, but the problem of constructing new algorithms is still of great interest.

With the growth of new technologies, the datasets in the future are going to be bigger and bigger. The content of each dataset is very different and it is quite difficult to find an approach, which works efficiently for all datasets. A variety of numerical approaches allows us to choose the most appropriate for the dataset under consideration.

There are a lot of approaches to construct polynomial approximations. It could be reasonable to consider instead of polynomial approximations piece-wise polynomial approximations. We split the interval $[a, b]$ into several segments and consider functions which are continuous on $[a, b]$ and polynomial on each of the segments. Such kind of approximation is called polynomial spline approximation.

The polynomial spline approximation is more flexible and it allows us to avoid large oscillation observed for high-degree polynomial approximation. Several approaches for polynomial spline approximation have been developed (see [11]-[14] for details). The main obstacle of polynomial spline approximation is that we have to determine two groups of parameters (polynomial coefficients and boundaries for each interval).

The objective function has a lot of local minima if the boundaries are not fixed, but the appropriate choice of segments within the interval $[a, b]$ could give much better approximation than the equidistant grid with the same number of segments (see [2], [11], [13] for more information).

In this paper we propose a new algorithm for polynomial spline approximation. We present the results obtained by this algorithm and compare them with the results obtained in [2]. We discuss the results and give some comments on a new approach.
2 Continuous function and Discrete Data Approximation

Assume that we study a continuous function \( f(x) \), defined on the interval \([a, b]\), or data (collection of points in two-dimensional space)

\[
D = \{(x_i, f(x_i))\}_{i=1}^{N},
\]

where \( x_i \in [a, b], i = 1, N \). Our task is to construct an approximation of the function \( f(x) \) (data \( D \)) by a curve. We can consider a curve from a certain class \( G(B) = \{g(B, x)\}_{x \in [a, b]} \), where \( B \in \mathbb{R}^p \) is a vector of parameters. Each particular vector \( B \) determines a particular curve from the class \( G(B) \). We have to choose an appropriate collection of parameters \( B_0 = (b_0^1, \ldots, b_0^p) \), such that the approximation \( g(x) = g(B_0, x) \) is satisfactory enough.

The well-known example of such a class is plots of polynomials

\[
P_{p-1}(x) = \sum_{i=1}^{p} b_i x^{i-1},
\]

defined on the interval \([a, b]\), the degree is less or equal to \( p - 1 \). Our task is to determine parameters \( b_1, b_2, \ldots, b_p \) to make the approximation as precisely as possible.

There are several criteria to measure a quality of approximation. We use two of them: the best uniform approximation criterion and the least squares criterion. We have two kinds of problem: continuous function approximation and discrete data approximation.

**Continuous Function Approximation.** Assume that we have to approximate a continuous function \( f(x) \) by a function \( g(x, B) \in G(B) \). We use the uniform optimization criterion to find a set of parameters \( B \), that is a solution of the problem:

\[
\text{minimize } \max_{x \in [a,b]} |g(x, A) - f(x)|, \quad \text{subject to } B \in \mathbb{R}^p. \tag{2.1}
\]

**Discrete Data Approximation.** Assume that we obtained some data

\[
D = \{(x_i, f(x_i))\}_{i=1}^{N},
\]

where \( x_i \in [a, b], i = 1, N \). We have to approximate this data by a function \( g(x, B) \in G(B) \). We can use several optimization criteria, for example uniform optimization criterion, we solve the following problem:

\[
\text{minimize } \max_{i=1,N} |g(x_i, B) - f(x_i)|, \quad \text{subject to } B \in \mathbb{R}^p \tag{2.2}
\]

and the least squares optimization criterion, we solve the following problem:

\[
\text{minimize } \left( \sum_{i=1,N} (g(x_i, B) - f(x_i))^2 \right)^{\frac{1}{2}}, \quad \text{subject to } B \in \mathbb{R}^p. \tag{2.3}
\]

3 Polynomial Splines.

Assume that we want to find an appropriate approximation for a continuous function of one variable \( f(x) \) or two-dimensional data. We study polynomial spline approximation.

There are a lot of approaches to construct polynomial splines (see, for example, [1], [2]). We have to choose the most convenient for solving our approximation problems.
Assume that the interval \([a, b]\) consists of \(n\) segments \([t_{i-1}, t_i]\), \(i = 1, n\), such that
\[
a = t_0 < t_1 < \cdots < t_{n-1} < t_n = b.
\] (3.1)

Consider the following function
\[
S(A, T, x) = a_0 + \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}(x - t_{i-1})^+_j,
\] (3.2)
where \(A = (a_0, a_{11}, \ldots, a_{1m}, a_{21}, \ldots, a_{2m}, \ldots, a_{nm}) \in R^{mn+1}\) — vector of coefficients of the spline,
\(T = (t_1, \ldots, t_{n-1})\), \(T \in T_{[a,b]} \in R^{n-1}\) — vector of knots of the spline, \(t_0 = a\),
the set \(T_{[a,b]}\) is a set of vectors whose coordinates are satisfied the inequalities in (3.1);
\[
(x - t_\alpha)_+^\beta = \begin{cases} 
0, & x < t_\alpha, \\
(x - t_\alpha)_+^\beta, & x \geq t_\alpha,
\end{cases}
\]
\(
\alpha = \overline{1}, n - \overline{1}, \beta = \overline{1}, m.
\)

It is evident that the function \(S(A, T, x)\), if \(A\) and \(T\) are fixed, is a piecewise polynomial function, the degree of a polynomial on each particular segment is less than or equal to \(m\).
The highest degree of polynomials, composing the spline, is called the order of the spline.

**Uniform approximation.** In the case under consideration the continuous function approximation problem has the form:
\[
\text{minimize } \max_{x \in [a,b]} |g(A, T, x) - f(x)|, \text{ subject to } A \in R^{mn+1}, T \in T_{[a,b]},
\] (3.3)
and the discrete data approximation problem has the form:
\[
\text{minimize } \max_{i=1,N} |g(A, T, x_i) - f(x_i)| \text{ subject to } A \in R^{mn+1}, T \in T_{[x_1,x_N]},
\] (3.4)
where the components of the vector \(A \in R^{mn+1}\) are coefficients of the spline, the components of the vector \(T \in T_{[a,b]}\) are knots of the spline.

If all knots are fixed, the problems (3.3) and (3.4) are as follows:
\[
\text{minimize } \max_{x \in [a,b]} |g(A, x) - f(x)|, \text{ subject to } A \in R^{mn+1}
\] (3.5)
and
\[
\text{minimize } \max_{i=1,N} |g(A, x_i) - f(x_i)|, \text{ subject to } A \in R^{mn+1}.
\] (3.6)

The objective functions in problems (3.5) and (3.6) are convex and they can be minimized, using methods of nonsmooth optimization, proposed, for example, in [3]-[6].

If the knots of a spline are not fixed, the problem is much more complicated, because the objective functions in problems (3.3) and (3.4) are nonconvex, nonsmooth and have a lot of local minima.

**REMARK.** In our numerical experiments we represented the interval \([a, b]\) as a collection of equidistant points (grid), which are sufficiently close to each other, and considered the values of the function \(f(x)\) in the points of this grid. The problem of continuous function approximation was transformed to a discrete data approximation problem.

**Least squares approximation.** This optimization criterion in the case of polynomial spline approximation for discrete data can be represented as the following problem:
\[
\text{minimize } \sum_{i=1}^{N} (f(x_i) - S(A, T, x_i))^2, \text{ subject to } A \in R^{mn+1}, T \in T_{[x_1,x_N]}.
\] (3.7)
If knots in (3.7) are not fixed, the objective function has a lot of local minima.
Table 1. Initial points search.

<table>
<thead>
<tr>
<th>Initial point</th>
<th>A</th>
<th>T</th>
<th>Objective function value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1, 0.1, 0.1, 0.1, 0.1</td>
<td>0.1, 0.6, 0.9</td>
<td>148.31</td>
<td></td>
</tr>
<tr>
<td>1, 1, 1, 1, 1</td>
<td>0.1, 0.6, 0.9</td>
<td>154.02</td>
<td></td>
</tr>
<tr>
<td>0.2, 0.2, 0.2, 0.2, 0.2</td>
<td>0.1, 0.6, 0.9</td>
<td>148.88</td>
<td></td>
</tr>
<tr>
<td>-0.1, 0.1, -0.1, 0.1, 0.2</td>
<td>0.1, 0.6, 0.9</td>
<td>0.40</td>
<td></td>
</tr>
<tr>
<td>0.23, 0.23, 0.23, 0.23, 0.23</td>
<td>0.1, 0.24, 0.95</td>
<td>0.52</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Results</th>
<th>A</th>
<th>T</th>
<th>Objective function value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0035, -0.0025, -0.0182, 0.0172, 0.1506</td>
<td>0.000, 0.000, 0.034</td>
<td>0.15</td>
<td></td>
</tr>
<tr>
<td>0.000, -0.0118, -0.0287, 0.0005, 0.15</td>
<td>0.973, 0.986, 0.999</td>
<td>0.15</td>
<td></td>
</tr>
<tr>
<td>0.075, 0.067, 0.060, -0.401, 0.097</td>
<td>0.122, 0.150, 0.499</td>
<td>0.12</td>
<td></td>
</tr>
<tr>
<td>-1.000, 2.000, -2.000, 2.000, 0.2000</td>
<td>0.200, 0.500, 0.800</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>0.230, 0.335, -0.669, 0.334, 0.100</td>
<td>0.201, 0.500, 0.800</td>
<td>0.1</td>
<td></td>
</tr>
</tbody>
</table>

4 Splines with Free Knots. Local optimization techniques.

4.1 Local optimization method: computational aspects

Most local optimization methods work faster than global methods (especially with the growing of the dimension of the optimization problem). It makes difficult to find a global minimum for both groups of variables simultaneously, but we can use some local optimization techniques to find a minimum subject to all variables. The choice of an initial point is very important for local optimization methods to solve the problem under consideration. It is reasonable to develop new algorithms based on a combination of local and global methods.

We use the Discrete Gradient method to minimize the objective functions. This method is a method of nonsmooth optimization (see [4] for details). Numerical experiments (see [5], [6], [9], [10]) show, that this method avoids many saddle points and some shallow local minima and leads to a local minimum, which is deep enough. In order to check the efficiency of this method for the problem of polynomial spline approximation, we applied the method to study some simple problems.

4.2 Numerical experiments

Let us consider a continuous piecewise linear function

\[ g(x) = \max_{x \in [0, 1]} \{|x - 0.2|, |x - 0.8|\} \]

on an interval [0, 1]. This function is a linear spline with three entire knots, such that \(T = (0.2, 0.5, 0.8)\). The vector of coefficients of the spline is \(A = (-1, 2, -2, 2, 0.2)\). We present the interval as a collection of 501 equidistant points \(x_1, \ldots, x_{501}\) the plot of function \(g(x)\) as a collection of points in two-dimensional space \(\{(x_i, g(x_i))\}_{i=1}^{501}\). We run the program starting from five different initial points. We present the results in Table 1.

In this problem the objective function is as follows:
\[
F(A, T) = \max_{x \in [0, 1]} |g(x) - S(A, T, x)|, \quad \text{subject to} \quad A \in \mathbb{R}^5, \quad T \in T[0,1] \subset \mathbb{R}^3,
\]

where

\[
g(x) = \max_{x \in [0, 1]} \{|x - 0.2|, |x - 0.8|\},
\]

\[
S(A, T, x) = a_0 + a_1x + \sum_{i=2}^{4} a_i(x - t_{i-1})_+,
\]

\[
A = (a_1, \ldots, a_4, a_0), \quad T = (t_1, \ldots, t_3), \quad (x - t_\alpha)_+ = \max\{0, x - t_\alpha\}, \alpha = 1, 2, 3.
\]

**Comments to Table 1**

- The first initial point leads to the construction of the best polynomial approximation. Instead of three intervals we obtained one. The knots move to the left boundary of the interval [0, 1].
- The second initial point also leads to the construction of the best polynomial approximation. The knots move to the right boundary of the interval [0, 1].
- The third point allows to catch the knot 0.5 only.
- The forth point leads to a solution, which is not optimal.
- The last row in Table 1 gives a solution, which is not optimal, but the distribution of knots is already close to an optimum.

We can conclude that even for this simple problem the choice of initial points for the Discrete Gradient method is very important.

5 **Splines with Free Knots. Comparison of Algorithms**

5.1 **Beliakov’s Approach**

In [2] the problem was split into two parts (linear and nonlinear).

**Algorithm 1.**

- **STEP I.** The knots are fixed, the problem in (3.7) could be transformed to a linear problem (see, for example, [14]);
- **STEP II.** Coefficients of the spline, found on the first step, are fixed and a method could be applied in order to find the knots of the spline.

The procedure, described above, was applied many times (50-1500 times) to reach solutions.

In order to solve the problem appeared on the **STEP I** in [2] method of QR decomposition was used. The method of QR (see [12]) decomposition is a method to study over determined linear systems \(Ax = y\) with respect to the least square criterion:

\[
\min \|Ax - y\|, \quad \text{subject to} \quad x \in \mathbb{R}^n.
\]

On the **STEP II** the Cutting angle method was applied. The Cutting angle method is a method of global optimization. It was developed for IPH functions, determined on
the unit simplex. The domain of the objective function, if the knots are free and the
coefficients are fixed, could be easily transformed to a unit simplex (see [2], for example).
This function is Lipschitz and could be transformed to a restriction of a certain Increasing
Positively Homogeneous of degree one function (IPH) to the unit simplex (see [7], [8] for
more information).

The algorithm allows us to overcome the computational difficulties induced by the
high dimension, but the problem was split and the solutions for each group of variables
were found independently. It means the solution obtained in [2] is not necessarily a global
minimizer.

5.2 New Algorithm for Discrete Data Approximation.
The problem is formulated as follows. Let

\[ D = \{(x_i, f(x_i))\}_{i=1}^{N}, \quad a = x_1 < x_2 < \cdots < x_N = b \]

be a collection of points. We have to find an approximation curve. This curve should
be found in a class of polynomial splines. We propose a new algorithm to construct a
polynomial spline approximation. This approach contains three stages.

Conceptual algorithm for new approach

- **STEP I.** We use a local optimization method to calculate a minimum subject
to both groups of coefficients (free knots polynomial spline approximation). The
number of knots is big enough.

- **STEP II.** We reduce the number of intervals if some knots obtained on **STEP I**.
We choose one of them if they are close to each other with respect to some tolerance
or eliminate them if they are close to the boundaries of the interval.

- **STEP III.** We fix knots obtained on the previous step and solve a problem of
polynomial spline approximation with fixed knots to determine the coefficients of a
spline.

There are several ways to specify the algorithm. We use the following procedure in order
to solve the approximation problem.

**Algorithm 2.**

- **STEP I.** We use D.G. method to find a linear spline approximation under uniform
optimization criterion, the number of knots is big enough.

- **STEP II.** If some knots are close to each other (with respect to some tolerance
\( \varepsilon > 0 \)) we choose one of them, if they are close to the boundaries of the interval we
eliminate them.

- **STEP III.** We fix knots found before in **STEP I** and apply the QR decomposition
to determine the coefficients of a spline.

**REMARK.** If we do not eliminate any knots, the number of intervals chosen on the
**STEP I** is not big enough.

**REMARK.** It is very difficult to know a priori, how many intervals we need for
approximation. This algorithm allows to determine the number of knots automatically.

**REMARK.** Within the algorithm we use different optimization criteria (**STEP I**
and **STEP III**). We studied several versions of the algorithm (with different optimization
criterion on each stage) and numerical experiments show that this combination is the most
efficient.
Table 2. Numerical experiments comparison.

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of intervals</th>
<th>Error</th>
<th>Order of spline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Titanium Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Algorithm 1</td>
<td>4</td>
<td>0.034</td>
<td>cubic</td>
</tr>
<tr>
<td>Algorithm 2</td>
<td>4</td>
<td>0.029</td>
<td>cubic</td>
</tr>
<tr>
<td>Pezzack’s Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Algorithm 1</td>
<td>4</td>
<td>0.031</td>
<td>cubic</td>
</tr>
<tr>
<td>Algorithm 2</td>
<td>4</td>
<td>0.031</td>
<td>cubic</td>
</tr>
<tr>
<td>Algorithm 1</td>
<td>4</td>
<td>0.042</td>
<td>quintic</td>
</tr>
<tr>
<td>Algorithm 2</td>
<td>4</td>
<td>0.013</td>
<td>quintic</td>
</tr>
</tbody>
</table>

5.3 Numerical experiments with Algorithm 1 and Algorithm 2: comparison of results.

We studied two datasets: the Titanium heart data (49 observations) and the Pezzack’s data (142 observations). The datasets, we used in numerical experiments, were studied in [2], using Algorithm 1. Cubical (for both datasets) and quintic (for Pezzack’s data) splines with different number of knots were constructed. In Algorithm 2 the number of knots is determined on the first step, but we can construct polynomial splines of different order.

To compare the results, obtained by these two algorithms, we computed the error $\delta(A, T)$ as it was done in [2]:

$$
\delta(A, T) = \left( \frac{1}{N-1} \sum_{i=1}^{N} w_i (f(x_i) - S(x_i))^2 \right)^{\frac{1}{2}},
$$

where $w_1 = w_N = \frac{1}{2}, w_i = 1, i = 2, N - 1,$

where $S(x)$ is a polynomial spline, we constructed to approximate the data.

Numerical implementation for Algorithm 2

- **STEP I.** We use D.G. method to find a linear spline approximation under uniform optimization criterion. In our experiments we used 9 equidistant interior knots.

- **STEP II.** We reduce the number of knots from 9 to 3 for both datasets ($\epsilon = 0.08$).

- **STEP III.** We fix knots, found before (STEP I, $T = (0.5144, 0.63, 0.827)$ for Titanium heart and $T = (0.33, 0.7, 0.85)$ for Pezzack’s) and apply the QR decomposition to determine coefficients of a spline.

We present a comparison of results, obtained by Algorithm 1 and Algorithm 2 (cubic and quintic splines).

When we applied Algorithm 2 twice (considering the results, obtained before as an initial point for STEP 1), there was no significant improvement, and it is not reasonable.
to repeat this procedure (it was necessary to apply Algorithm 1 many times to find a solution).

The Algorithm 1 exploited global optimization techniques, it requires much more memory and could meet some difficulties, if the dimension of the problem is high.

The approximation accuracy, reached by Algorithm 2 is higher, than the accuracy, reached by Algorithm 1. We can conclude, that the Algorithm 2 works efficiently with all examples of data approximation, we considered.

Figures 1-3 present results of numerical experiments.

Conclusions

- We proposed a new algorithm to find a polynomial spline approximation for discrete data.
- This algorithm determines the number of intervals, needed for approximation.
- Within this algorithm we use a combination of the Discrete Gradient method and QR decomposition method.
- The results, obtained by a combination of this method with QR decomposition method (Algorithm 2) are better than the results, obtained by Algorithm 1.

Acknowledgment

I would like to tell thanks to my supervisors Prof. A. Rubinov and Prof. A. Bagirov for their helpful advices. This research was supported by the Victorian Partnership for Advanced Computing, Victoria, Australia.

References


Figure 3: Cubic spline for Titanium heart data


