Optimization in Function Fitting and Interpolation

Boda Devaki ,Chemikala Bhanu Teja Reddy, Muvva Thanmai,Dilip Kumar Reddy Nagireddygari, Narayana Banu Bharadwaj

Abstract: Analyzing data and studying its behavior is fundamental to any aspect of science. Its behavior and trend can be studied comfortably using the methods of curve fitting and interpolation.It helps in two ways. Large data requires a lot of storage. In this project we are going to mainly study the methods of convex optimization , mainly focused on the spline function fitting using linear combination of B spline function, used in Function fitting. Basically we have a n dimensional subspace (n variables) whose different set of values are given to us. On the basis of which we have to interpret a function which satisfies the given trends. We try to build a function by minimizing the norm of the function taken and given data. We basically try to understand the basic concepts of function fitting methods(such as minimum norm) and functions(such as spline, polynomial, piecewise) in the span of the project and try to solve some problems or applications based on what we have learnt.Interpolation,extrapolation , bounding are done based on the above process.Most of the real-life problems, although spread widely across many fields,we mainly concentrate on mechanical engineering problems and machine learning .

Index Terms: splines , b-splines, l1 regularization, l2 regularization, active and inactive knots, Genetic algorithm.

1. Introduction

Consider a family of functions $f_1, f_2, ...f_n : R^k \rightarrow R$ and a vector x belonging to the space R^n. Now, A function f can be built with the given functions $f(u) = x_1 f_1(u).. + x_n f_n(u)$ for the fitting problem. This is called a basis and the basis functions generate a subspace F of functions on D, where D is the intersection of domain of the functions in the basis and is the domain of the resultant function f.

The functions in the basis are chosen by using previous experiences, some of those are as follows

1) **Polynomial functions:**
   In polynomial functions, there are different kinds of basis. We can take the simplest one being $f_i(t) = t^{i-1}$, where i = 1,2,3,4.....n where each f_i is orthonormal to the others.
   Lagrange basis, trigonometric polynomials are some majorly used basis

2) **Piece-wise linear functions:**
   We divide the domain into m disjoint simplex's where Each simplex is the convex hull of k + 1 grid points, and we require that each grid point is a vertex of any simplex it lies in.
   for each of the simplex, a basis function is defined such as for any point in the simplex it outputs 1 else it outputs 0

3) **Piece-wise polynomials or Splines:**
The idea of triangularization of domain remains the same but the linearity of the function defined in each simplex.But the functions we choose as a basis function in each simplex are not linear. They are chosen to be continuous at the edges of simplex.Now additionally if we chose the basis functions such that their derivatives up to order k-1 are also continuous they are known as splines where k is the order of the piece wise polynomial considered

   a) **Splines:**
   A function s(x) defined on a finite interval $[x_{min}, x_{max}]$ whose degree k(≥ 0) with knots denoted by a strictly increasing sequence λ_i where i = 0,1,2,3.....,g+1.
   On each knot interval [λ_i, λ_{i+1}] s(x) is a polynomial of atmost degree k.
   $S_{[\lambda_i,\lambda_{i+1}]}$ ∈P$_k$ where i = 0,1,2....g.
   S(x) and it's derivatives upto order (k-1) are continuous on the interval $[x_{min}, x_{max}]$
   $S_{[\lambda_i,\lambda_{i+1}]}$ ∈C$^{k-1}_{[x_min,x_max]}$
   An internal knot λ_i is called active if the kth-order derivative of s(x) is discontinuous at λ_i and is inactive otherwise.

Splines are known to be greatly helpful in the function fitting and they are discussed in more detail further

The constraints on the function define the convexity of fitting problem :

1) Constraints on Function value:
The constraints on the function value(inequalities) such as $l \leq f(v) \leq u$. They are convex since they are half spaces.
The Lipschitz constraint and positivity constraint are other examples.
2) Derivative constriants:
Derivative of the given function is a combination of partial derivatives. $\nabla f(v) = \sum_{i=1}^{n} x_i \nabla f_i(v)$, $||\nabla f(v)|| = ||\sum_{i=1}^{n} x_i \nabla f_i(v)|| \leq M$,
3) Integral constraints:
Constraints in the form of integrals are also convex.

There are many methods of solving the function fitting problem using convex optimization :

1) Minimum Norm Function Fitting
2) Using Splines as Basis Functions
3) Gradient Descent Algorithms
4) Using Genetic Algorithm

This project mainly focuses on optimization using splines.

Minimum norm function fitting:
In this problem, a function f $\in F$ that matches the given data,
$(u_1, y_1), .., (u_m, y_m)$ with $u_i \in$ D and $y_i \in$ R
as closely as possible is obtained.
Number of constraints can be added to the problem like linear inequalities satisfying the function at various points, derivative constraints, monotonicity constraints.
For example, a simple least-square fitting problem in variable x is given by
minimize $\sum_{(i=1)}^{m} (f(u_i) - y_i)^2$

In these problems, the number of data points is much larger than the dimension of the subspace of functions i.e., m $\gg$ n.

Gradient Descent Algorithms:
In this problem of finding smooth curves to the given set of data points $p_0, p_1, ..., p_n$, on non-linear manifolds at distinct and ordered instants of time, the obtained smooth curve γ involves two goals of conflicting nature.

1) The curve best approximates the data.
$E_d(\gamma) = \sum_{i=0}^{n} d^2(\gamma(t_i), p_i)$
where d denotes the distance function on the Riemannian manifold M.
2) The curve should be sufficiently regular, i.e., the changes in the velocity or the acceleration is minimized.
Thus, it results in a optimization problem with two objective functions, fitting function E_d and regularity function E_s, whose domain is the suitable set of curves in the Riemannian manifold M.

Expressing splines as a linear combination of B-splines:
B-splines or Basis-splines are used as a basis to represent polynomial splines.
A B-spline of degree l having order (l+1) can be computed by the De-Boor recursive formula, given by:

$$B_{i,l+1}(x) = \frac{x-\lambda_i}{\lambda_{i+l}-\lambda_i} B_{i,l}(x) + \frac{\lambda_{i+l+1}-x}{\lambda_{i+l+1}-\lambda_{i+1}} B_{i+1,l}(x)$$

with the base case of degree 0 as:

$$B_{i,1}(x) = \begin{cases} 1, if x \in [\lambda_i, \lambda_{i+1}). \\ 0, \quad otherwise. \end{cases}$$

The number of knots is related to accuracy of the spline.
The vector space of a spline is of dimension g+k+1. The number of independent b-splines that can be constructed with the above defined equation is g-k+1. An additional 2k knots are required to obtain full set of basis function to cover the vector space of dimension g+k+1.
This can be done by introducing the boundary knots. $\lambda_{-k}, ..., \lambda_{-1}$ and $\lambda g + 2, ..., \lambda g + k + 1$, such that, $\lambda_{-k} = ... = \lambda_{-1} = \lambda_0 = x_{min}$ $\lambda_{g+1} = \lambda_{g+2} .... = \lambda_{g+k+1} = x_{max}$
so S(x) can be written as linear combination of the g+k+1 basis functions obtained and multiplying them with the spline coefficients c_i
s(x) = $\sum_{i=-k}^{g} c_i B_{i,k+1}(x)$
The vth derivative of s(x) is also a spline of degree k-v.

$$s^(v)(x) = \prod_{i=1}^{v}(k+1-i) \sum_{i=-k+v}^{g} c_i^{(v)} B_{i,k+1-v}(x)$$

with

$$c_j^{(i)} = \begin{cases} c_j if \ i = 0 \\ \frac{c_j^{(i-1)} - c_{(j-1)}^{(i-1)}}{\lambda_{(j+k+1-i)} - \lambda_j} \ if \ i > 0 \end{cases}$$

The basic function fitting problem that is formulated by Hayes as shown below:
Given values $y_r, r = 1, ..., m$, corresponding to values $x_r \in [x_{min}, x_{max}]$, determine a function $y(x) := y(x;)$ of known form but containing a vector of n disposable parameters to be determined such that $y(x_r) \approx y_r$.
The basis spline function changes its convexity across an active knot. This can be verified by a

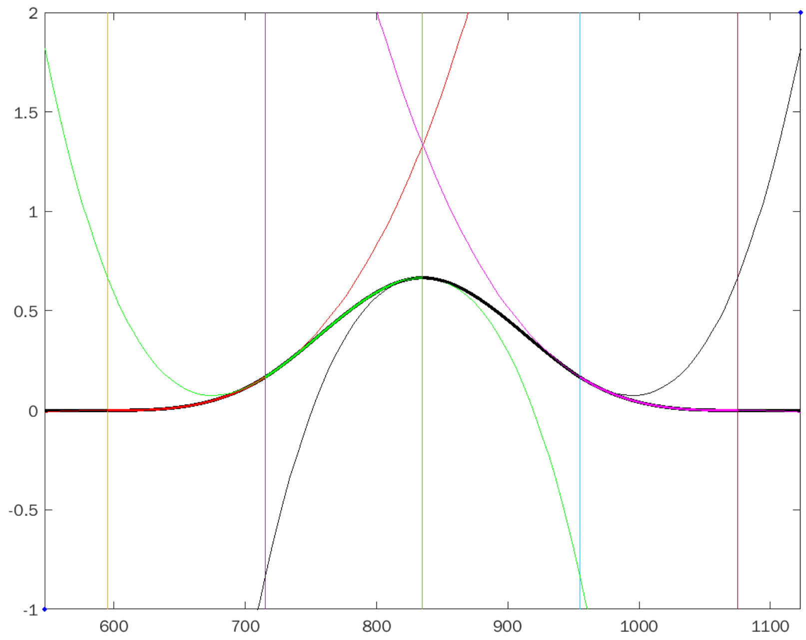

Fig. 1. Cubic Basis splines with 4 knots

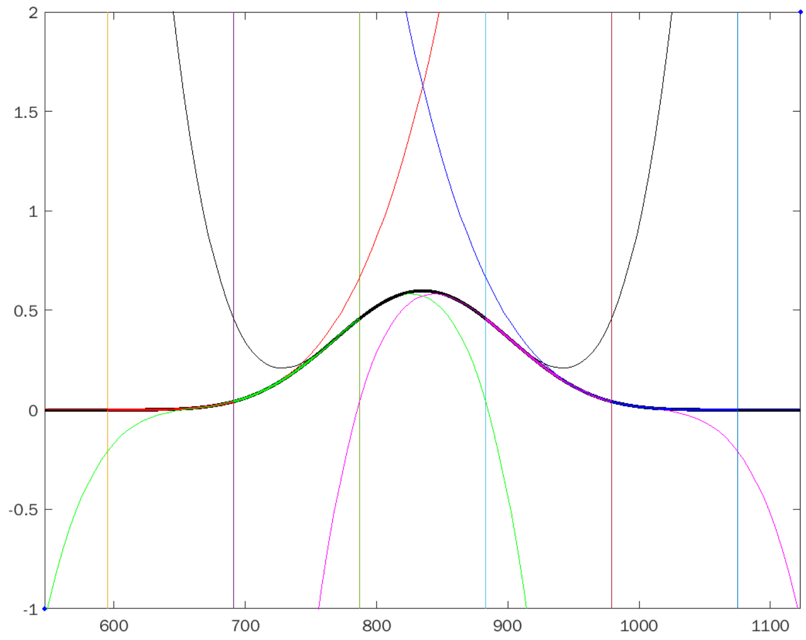

Fig. 2. Cubic Basis spline with 5 knots

sign change at the second-order derivative of the function.
This problem can be tackled by adopting the indirect spline knot optimization approach proposed by Demeulenaere et al. Since the candidate knots are already predefined, splines coefficients are to be determined in such a way that the active knots are minimal. This process is done using l1 regularization. This results in a convex problem.
The coefficients of the derivatives of the spline function along with the coefficients of the spline function are considered optimization variables for the ease of problem-solving.
The dimension of this optimization variable vector(c) is $(k+1)(g+1+k/2)$.

$$c = [c_{-k}^{(0)} ... c_{g}^{(0)} c_{-k+1}^{(1)} ... c_{g}^{(1)} ..... c_{0}^{(k)} ... c_{g}^{(k)}]^T$$

Constraints on the derivative coefficients are defined in the above equation and are $k(g+\frac{k+1}{2}$ in number.

Clearly, it is observed that $k(g+\frac{k+1}{2})+(g+k+1)=(k+1)(g+1+\frac{k}{2})$.

Minimize: $\sum_{i=1}^{g} w_i|s^{(k)}(\lambda_i+)-s^{(k)}(\lambda_i-)|$

subject to $\sum_{r=1}^{m}(v_r(y_r-s(x_r)))^2 \leq S$

hspace*2 cm$C.\mathbf{c}=0$

$$g(s(x),....,s^{(k)}(x))=0$$
$$h(s(x),....,s^{(k)}(x))\leq 0$$

w_i, v_i and S are the fixed parameters. g and h are the additional constraint functions and are generally obtained from the prior knowledge.
The above defined problem is similar to the Dierckx smoothing criterion. The difference is that instead of using the l1 norm the Dierckx problem uses l2 norm .
l1 norm produces sparse solutions because of discontinuity in derivatives.

$$\sum_{i=1}^{g}(s^{(k)}(\lambda_i+)-s^{(k)}(\lambda_i-))^2$$

l1 norm is more suitable than l2 norm because

1) it measures the non smoothness of the spline function
2) l1 norm minimization produces sparse solutions implies very few active knots $(s^{(k)}(\lambda_i+)-s^{(k)}(\lambda_i-))$(optimal knot locations are found automatically where as in other algorithms knot values are chosen before hand)

Candès et al proposed that if even after l1 regularization the number of active knots are significant the weights are updated to

$$w_i^{(j+1)} = \frac{1}{|s^{(k)}(\lambda_i+)-s^{(k)}(\lambda_i-)|+\epsilon}$$

so that, the weight of the active knots at which non-smoothness are relatively less are given more prominence to be reduced to convert them into inactive knots.

Using Genetic Algorithm:

Genetic Algorithms are the efficient search algorithms working with the bit strings based on the mechanics of natural genetics. These GAs are used in curve fitting to avoid the complicated and unreliable process of finding the fitting parameters.
Its goal is to minimize the curvature integral in order to reach an optimal shape. A real-coded genetic algorithm is developed in to find good knots of a fitting spline.
These genetic algorithms are run simultaneously and cooperatively, consisting of a coevolutionary genetic algorithm. The final solution of the fitting algorithm is derived by combining the partial solutions of the GAs.

Examples:

This section deals with two examples of how our approach can be used for a better fit:

1) **Unconstrained optimisation:**
Here, the titanium heat data (from de boor and rice) is fitted. This turns out to be so difficult with the conventional methods like minimum norm and gradient descent due to high fluctuations in the data. Hence, this is approached by spline fitting algorithms in a way just similar to the above mentioned problem. This problem is solved by cubic splines(degree, k = 3) and S(our parameter to control quality of fit) = 0.0073, and the weights v_r are taken as 1, and 1000 equidistant candidtae knots. This value of S corresponds to the one used in Dierckx[1975] and is close to the minimum quadratic error for the final 5 active knots[Jupp, 1978].
For the same titanium heat data, here are the number of final active knots(g) and the value of non-smoothness for the three methods:

Method	g	non-smoothness
dierckx	8	0.0899
Jupp	5	0.0742
Convex approach	7	0.0433

Clearly, non-smoothness is lesser in our fit than in the other two methods implying better fit of our method. Also, it is not needed to provide initial guesses of the knot sequence in this approach which is mainly needed in the other methods for a better fit.

If possible, it is preferred to add pre-known constraints, if we know any, to the problem. This helps the algorithm to produce a still more better fit.

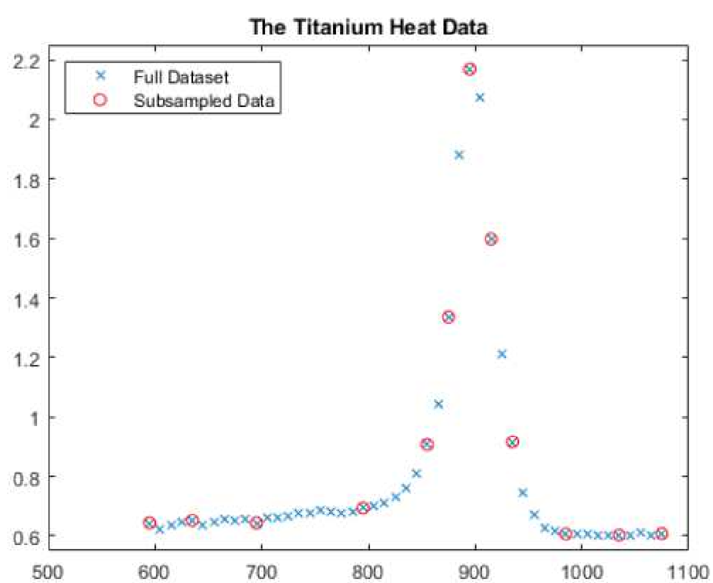

Fig. 3. Available data points

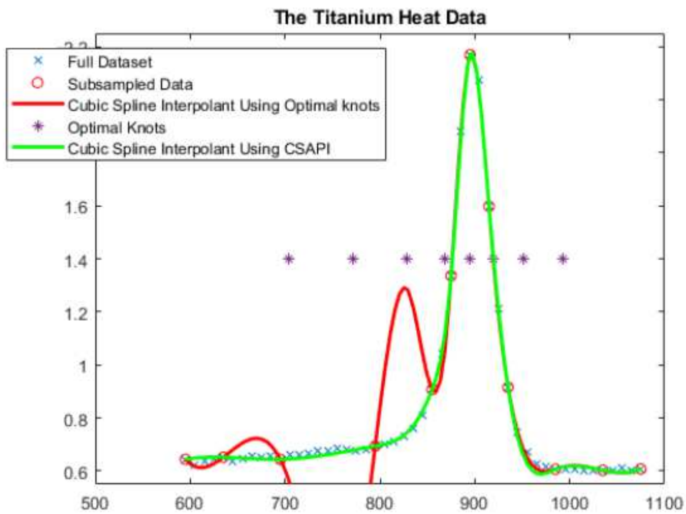

Fig. 4. Cubic spline interpolant using csapi

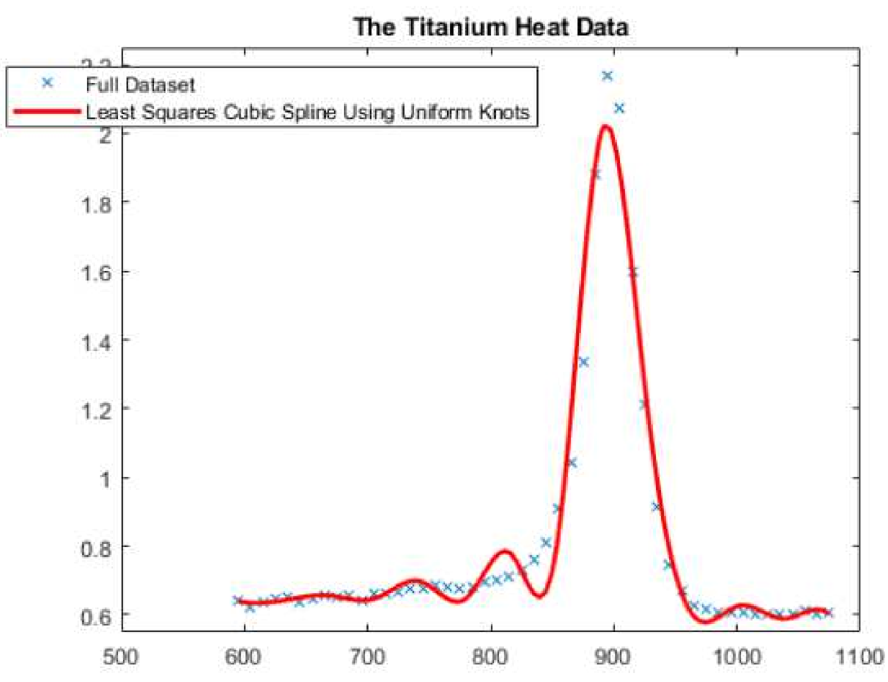

Fig. 5. Least sqaures cubic spline using uniform knots

2) **Fitting with constraints between curves**
Consider measurements of n marker positions P_i i = 1,2,3..n as ends of a rigid body moving in 3D let the co ordinates be x_i, y_i, z_i of a point i and the function $S_{x_i}, S_{y_i}, S_{z_i}$ are spline functions of fitted values of x_i, y_i, z_i as a function of time.
measurement noise is considered $\pm\delta$

The constraint becomes a inequality as

$$(\mu_{ij} - \delta)^2 \leq \Sigma(S_{l_i} - S_{l_j})^2 \leq (\mu_{ij} + \delta)^2$$

observe that the lower bound on the constraint is not a convex constriant and hence it is converted into linear constraint before solving
Conversion into the linear :

$$(\Delta x, \Delta y) = (x_i - x_j, y_i - y_j)$$

$$(dx, dy) = (x_i(t_k) - x_j(t_k), y_i(t_k) - y_j(t_k))$$

We can say that from the least square constraint that $(x_i(t_k) - x_j(t_k), y_i(t_k) - y_j(t_k))$ is close to $(S_{x_i}(t_k) - S_{x_j}(t_k), S_{y_i}t_k)$

$$n_x = sgn(dx)(\mu_{ij} - \delta)\sqrt{\frac{dx^2}{dx^2 + dy^2}}$$

$$n_y = sgn(dy)(\mu_{ij} - \delta)\sqrt{\frac{dy^2}{dx^2 + dy^2}}$$

This can be explained as resolving components of the vector $(S_{x_i}(t_k) - S_{x_j}(t_k), S_{y_i}t_k - S_{y_j}t_k))$
$\Rightarrow$n$_x(S_{x_i}(t_k) - S_{x_j}(t_k)) + n_y(S_{y_i}(t_k) - S_{y_j}(t_k)) \geq (\mu_{ij} - \delta)^2$
The given constraint is linear. The splines for each of the function are added in the objective at minimization and the least square fit.

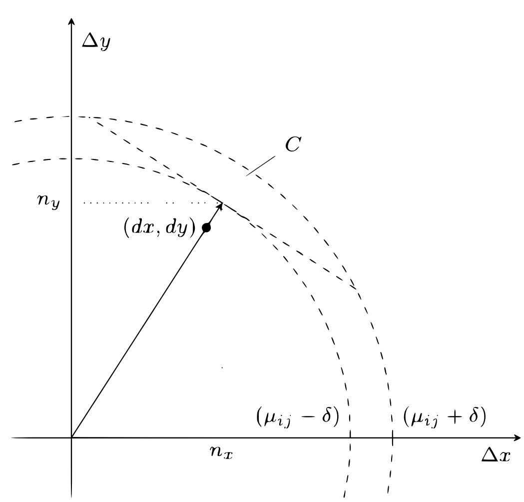

Fig. 6. Linearization of concave domain.

Psuedo code for the alogirthm:

Initialise:

knots = [input()]

y$_r = [input()]$

$w_i = 1, v_r = 1$

$S = somedesiredvalueoftheparametertocontrolthefit$

$g = initialnumberofknots$

$k = degreeofthesplinefunction$

$z = desiredminimumvalueofnon-smoothness$

Initialising b-spline functions:

B = cp.Variable((g, k+1))

//Initialising b-spline functions(when degree = 0)

for i in range(1000):

if $x \in [knots[i], knots[i+1])$:

$B[i][0](x) = 1$

else:

$B[i][0](x) = 0$

//de-boor recursive formula

for j in range(g):

for k in range(1, k+1):

$B[j][k](x) = \frac{x-knots[j]}{knots[j+1]-knots[j]}B[j][k-1](x) + \frac{knots[j+k]-x}{knots[j+k]-knots[j+1]}B[j+1][k-1](x)$

//kth derivative of our spline function

def s$_k(x)$:

$return\prod_{i=1}^{k}(k+1-i)\sum_{i=0}gc_i^{(k)}B[i][1](x)$

//The optimisation problem

while $\sum_{i=1}^{g} w_i|s^{(k)}(knots_i+) - s^{(k)}(knots_i-)| \geq z$:

objective = cp.Minimize($\sum_{i=1}^{g} w_i|s^{(k)}(knots_i+) - s^{(k)}(knots_i-)|$)

constraints = [$\sum_{r=1}^{m}(v_r(y_r - s(x_r)))^2 \leq S$,

$g(s(x), ..., s^{(k)}(x)) = 0$,

$h(s(x), ..., s^{(k)}(x)) \leq 0$]

//g and h are problem specific constraints

//c's of b-splines recursively using c's of previous order of b-splines:

for i in range(k):

for j in range(g + k + 1 - i):

if $i == 0$:

constraints.append($c_j^i = c_j$)

else if $i > 0$:

constraints.append($c_j^i = \frac{c_j^(i-1) - c_(j-1)^(i-1)}{knots[j+k+1-i]-knots[j]}$)

//weights are updated

$w_i = \frac{1}{|s^{(k)}(knots[i]+) - s^{(k)}(knots[i]-)| + \epsilon}$

References:

1.A Convex Optimization Approach to Curve Fitting with B-Splines by W.Van Loock,G.Pipeleers, J.De Schutter and J. Swevers, Proceedings of the 18th World Congress,IFAC,Milano(Italy), August 28-September 2,2011.

2.Interactive Spline Approximation by Marian Merchant,Simon Fraser University,1971.

3.Least Squares Cubic Spline Approximation I-Fixed Knots by Carl de Boor and John R.Rice, April 1968, Purdue University.
4.Chapter 6.5 in Convex Optimization by Stephen Boyd,Stanford University