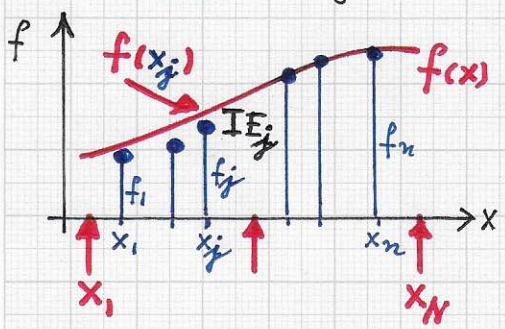
■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks:...

Definition of Hardy's MC method for the univariate setting:



$\sum E_j^2 \rightarrow \min$

• Given data: $\{(x_j, f_j)\}_{j=1,\dots,n}$

• Knots: $\{x_i\}_{i=1}^{N}$, $N \ll n$

• N radial MC basis functions:

$$r_i(x) = \left(R^2 + d_i^2(x)\right)^{1/2}, \quad i = 1,\dots,N$$

• Commonly suggested value for R^2:

$$R^2 = \max\left\{|x_{j_1} - x_{j_2}|\right\}/100, \quad j_1, j_2 \in \{1,\dots,n\}$$

• Squared distance:

$$d_i^2(x) = (x - x_i)^2$$

• Overdetermined linear system:

$$f(x_j) = \sum_{i=1}^{N} c_i r_i(x_j), \quad j = 1,\dots,n.$$

One can select the needed "scattered data approximation" method for a model function P_{cl} from a multitude of techniques developed for multivariate approximation. Hardy's multiquadric (MC) method is another method one can adopt to establish a local model for a probability function P_{cl} based on scattered, unorganized p_{cl}^{sc} data. Before adapting the standard MC method to our classification problem, we review its basic definition. The prototypical Hardy MC approximation function is the expansion

$$f(x) = \sum_{i=1}^{N} c_i \left(R^2 + d_i^2(x)\right)^{1/2}.$$

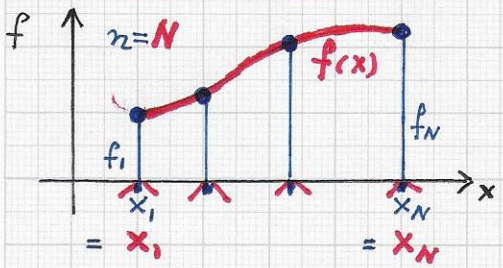
The function $f(x)$ can be constructed as a function that approximates given data (x_j, f_j), $j = 1,\dots,n$, $N \ll n$. The function $f(x)$ is defined as the function minimizing the sum of squared errors, $\sum_{j=1}^{n} (f_j - f(x_j))^2$.

...

OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

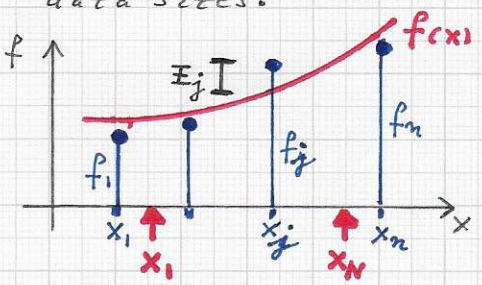
• Laplacian eigenfunctions and neural networks:...

Hardy's MC method used for interpolation:



In the interpolation scenario, the data sites x_i are equal to the knots x_i, and the number of sites is equal to the number of knots, i.e., $i = 1, \ldots, N$.

Our objective is the computation of a **MODEL** function by using Hardy's MC method to define a function $f(x)$ that uses only "a relatively small number of knots / radial basis functions - but large enough to achieve an acceptable error of approximation at the given data sites:



Ideally, one can use a MODEL function $f(x)$ for which $N \ll n$ and the site-specific approximation errors E_j satisfy some set ε threshold criterion.

• **Note.** In the context of scattered data INTERPOLATION, Hardy's MC method (and also Shepard's method) is used to interpolate f_i-values at their associated sites. In this case, the number of basis functions equals the number of function values to be reproduced exactly, i.e.,

$$f(x_j) = \sum_{i=1}^{N} c_i \, r_i(x_j), \quad j = 1, \ldots, N.$$

The consequence of exact interpolation is an N-by-N matrix that must be inverted for c_i-value computation:

$$\begin{bmatrix} r_1(x_1) & \cdots & r_N(x_1) \\ \vdots & & \vdots \\ r_1(x_N) & \cdots & r_N(x_N) \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix}$$

$$\Leftrightarrow \quad \mathcal{R} \, \mathcal{C} = \mathcal{F}$$

For our application, we use Hardy's MC method for approximation (best approximation) purposes. The two main reasons for this use are:

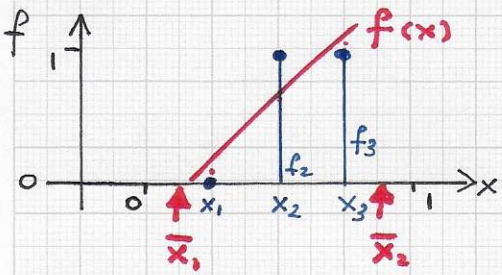
(i) The given data in our classification setting are NOT exact, precise, error-free data. (ii) By keeping the number of radial basis functions $r_i(x)$ relatively small, one can compute the $\{c_i\}$ and $f(x)$ efficiently.

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...

Simple example of the use of Hardy's MC method for best approximation:



• Given data: $x_1 = \frac{1}{4}, x_2 = \frac{1}{2}, x_3 = \frac{3}{4}$
$f_1 = 0, f_2 = f_3 = 1$

• Knots: $\bar{x}_1 = \frac{1}{8}, \bar{x}_2 = \frac{7}{8}$

• Constant $R^2 = 1/64$

• Compute $r_i(x_j)$ values:

$$r_i(x_j) = \left(1/64 + (x_j - \bar{x}_i)^2\right)^{1/2}$$

$$\Rightarrow R = \frac{\sqrt{2}}{8}\begin{bmatrix} 1 & \sqrt{13} \\ \sqrt{5} & \sqrt{5} \\ \sqrt{13} & 1 \end{bmatrix}$$

Compute $\mathbb{C}$ as $(R^T R)^{-1} R^T \mathbb{f}$:

$$\begin{bmatrix} 19 & 5+2\sqrt{13} \\ 5+2\sqrt{13} & 19 \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = 4\sqrt{2}\begin{bmatrix} \sqrt{5}+\sqrt{13} \\ 1+\sqrt{5} \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \approx \begin{bmatrix} 1.9082 \\ -0.2629 \end{bmatrix}$$

$$\Rightarrow f(x) = 1.9082\, r_1(x) - 0.2629\, r_2(x)$$

• Note. The value of R^2 used in this example is a "poor choice" for practical purposes.

When using Hardy's MC method for best approximation, the dependent function values are f_j, $j = 1, \ldots, n$, given at associated sites x_j. The expansion $f(x) = \sum_{i=1}^{N} c_i r_i(x)$, $N < n$, is defined by n equations:

$$\begin{bmatrix} r_1(x_1) & \cdots & r_N(x_1) \\ \vdots & & \vdots \\ r_1(x_n) & \cdots & r_N(x_n) \end{bmatrix}\begin{bmatrix} c_1 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}$$

$$\Leftrightarrow \quad R \quad \mathbb{C} = \mathbb{f}.$$

The coefficient vector $\mathbb{C}$ of the best approximation that minimizes the sum of the squared error values is given by the least-squares solution:

$$\mathbb{C} = \left(R^T R\right)^{-1} R^T \mathbb{f}.$$

To avoid confusion, the knots associated with the radial basis functions $r_i(x)$ will be called $\bar{x}_i$. Generally, knot values/locations $\bar{x}_i$ are different from site values/locations x_j. In fact, determining the best knots $\bar{x}_i$ for a given N is yet another optimization problem.

...

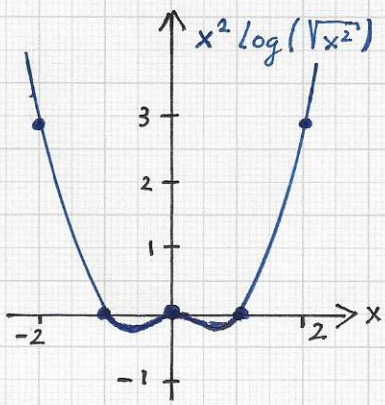
■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks:...

Behavior of a TPS basis function in the univariate case:

$$r_i(x) = d_i^2(x) \cdot \log(d_i(x))$$
$$= (x-x_i)^2 \cdot \log\left((x-x_i)^2\right)^{1/2}$$
$$\overset{x_i=0}{=} x^2 \cdot \log\left((x^2)^{1/2}\right)$$

The value $r_i(0)$ is defined as 0.



This specific basis function has 0 as the value of its knot. It is important to note that $\log(0) = -\infty$ and the radial basis function $x^2 \log(\sqrt{x^2})$ has the value 0 for $x = 0$. In addition, this function also has value 0 for $x = +/-1$.

Thus, the basis function $r_i(x)$, and any multiple of it, does not contribute to the value of an expansion $f(x)$ for the x-values $x = x_i$, $x = x_i - 1$ and $x = x_i + 1$, where x_i is r_i's associated knot.

Another method widely used and studied in the context of scattered data approximation and interpolation is the **thin plate spline (TPS)** method. Just like a natural cubic spline, a TPS has a physics-related bending energy minimization property. A TPS can also be employed for the purposes of our driving data classification application. The TPS also uses radial basis functions; they are defined as

$$r_i(x) = d_i^2(x) \cdot \log(d_i(x)),$$

where $d_i^2(x)$ is the squared distance between x and the knot/center $\bar{x}_i$ associated with the i^{th} basis function. Thus, a TPS data approximation also has the representation

$$f(x) = \sum_{i=1}^{N} c_i r_i(x).$$

The knot/center of basis function $r_i(x)$ is $\bar{x}_i$; thus, $r_i(x)$ is

$$r_i(x) = (x-\bar{x}_i)^2 \cdot \log\left((x-\bar{x}_i)^2\right)^{1/2}.$$

The linear system to solve is again

$$f(x_j) = \sum_{i=1}^{N} c_i r_i(x_j) = f_j,$$
$$j = 1, \ldots, n.$$

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks:...

References for thin plate splines (fundamental literature for scattered data approximation):

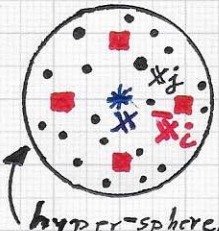
1) Richard Franke, 1982, "Smooth interpolation of scattered data by local thin plate splines"

2) John R. McMahon, Richard Franke, 1992, "Knot selection for least squares thin plate splines"

3) Richard Franke, Hans Hagen, Gregory M. Nielson, 1994, "Least squares surface approximation to scattered data using multiquadric functions"

4) Richard Franke, 1985, "Thin plate splines with tension"

5) Mohammadreza Amirian, Friedhelm Schwenker, 2020, "Radial basis function networks to learn similarity distance metric and improve interpretability"



2D ball example:

$\{x_j\}$ = data sites

$\{\bar{x}_i\}$ = knots of RBFs

$\{x\}$ = evaluation location

hyper-sphere, bounding the local neighborhood used for computations

$$f(x) = \sum_{i=1}^{N} c_i r_i(x) + \ldots$$

• **Note.** If it is desirable that an expansion has so-called "polynomial precision," then it will be possible to achieve this property by including additional – polynomial – basis functions in the expansion $f(x)$ to be computed. For example, $f(x)$ should have "linear precision," i.e., if the data values to be approximated all were lying on a line / coming from a linear polynomial, then this line / linear polynomial would be reproduced. In this scenario, the function $f(x)$ is

$$f(x) = \sum_{i=1}^{N} c_i r_i(x) + a + bx.$$

The set of equations to be used for the best approximation problem is:

$$f(x_j) = \sum_{i=1}^{N} c_i r_i(x_j) + a + bx_j = f_j$$

AND

$$\sum_{i=1}^{N} c_i = 0 \,, \quad \sum_{i=1}^{N} c_i \bar{x}_i = 0 \,.$$

Again, $\bar{x}_i$, $i = 1 \ldots N$, are the knots, and x_j, $j = 1 \ldots n$, are the sites with the data values f_j to be approximated, where $N \ll n$. In our classification application, the (probabilities of) class labels must change over short distances; thus, high-degree polynomials might be needed to model "cliffs."

...