Discrete thin-plate splines

Linda Stals

Mathematical Sciences Institute, Australian National University
Thin-Plate Splines

- 3D Image Recovery
- Finger Print Analysis
- Image Warping
- Medical Image Analysis
- Data Mining
Given a set of attributes vectors $\mathbf{x} = (x_1, x_2, \cdots, x_d)^T$, build a predictive model

$$\mathbf{y} = f(\mathbf{x}).$$

$$\mathbf{y} \approx f(\mathbf{x}).$$

To estimate $f$ by a 2nd-order smoothing spline minimise:

$$J_\alpha(f) = \frac{1}{n} \sum_{i=1}^{n} (f(\mathbf{x}^{(i)}) - y^{(i)})^2 + \alpha \int_{\Omega} \sum_{|\nu|=2} \binom{2}{\nu} (D^\nu f(\mathbf{x}))^2 d\mathbf{x}.$$  

The first term penalises lack of fit, the second penalises roughness.
The standard approach is to represent $f$ as a linear combination of radial basis functions

$$f(x) = \sum_{k=1}^{M} a_k \phi_k(x) + \alpha \sum_{i=1}^{n} w_i U(x, x^{(i)}),$$

where $\phi_k$ are monomials of order up to 1 and $U$ are suitable radial basis functions.

Favoured method as it gives an analytical solution.
Radial Basis Functions

2D Eg:

\[ U(x, x^{(i)}) = \frac{-1}{16\pi} r^2 \ln(r). \]
Thin Plate Splines

- Requires a solution of a **dense** system of matrices.
- System may be ill-conditioned.
- **Size increases with the number of data points.**

Not practical for large data sets.
Finite Element Approximation

Represent $f$ as a linear combination of linear finite elements. In vector notation $f$ will be of the form

$$f(x) = b(x)^T c.$$ 

Minimise $J_\alpha$ over all $f$ of this form

$$J_\alpha(f) = \frac{1}{n} \sum_{i=1}^{n} (f(x^{(i)}) - y^{(i)})^2 + \alpha \int_{\Omega} \sum_{|\nu|=2} \binom{2}{\nu} (D^\nu f(x))^2 \, dx.$$ 

[S. Roberts, M. Hegland, I. Altas; SINUM 2003]
The smoothing term (derivatives) is not defined for piecewise multi-linear functions.

Use non-conforming finite elements.

Represent the gradient of $f$ by $\mathbf{u} = (\mathbf{b}^T \mathbf{g}_1, \ldots, \mathbf{b}^T \mathbf{g}_d)$ where

$$
\int_{\Omega} \nabla f(\mathbf{x}) \cdot \nabla v(\mathbf{x}) \, d\mathbf{x} = \int_{\Omega} \mathbf{u}(\mathbf{x}) \cdot \nabla v(\mathbf{x}) \, d\mathbf{x},
$$

for all piecewise multi-linear function $v$. 
Non-Conforming Finite elements

\[
\int_{\Omega} \nabla f(x) \cdot \nabla v(x) \, dx = \int_{\Omega} u(x) \cdot \nabla v(x) \, dx,
\]

is equivalent to

\[
Lc = \sum_{s=1}^{d} G_s g_s,
\]

where \( L \) is a discrete approximation to the negative Laplace operator and \((G_1, \ldots, G_d)\) is a discrete approximation to the transpose of the gradient operator.
Finite Element Approximation

2nd-order smoothing spline: minimise

\[ J_\alpha(f) = \frac{1}{n} \sum_{i=1}^{n} (f(x^{(i)}) - y^{(i)})^2 + \alpha \int_\Omega \sum_{|\nu|=2}^2 (D^\nu f(x))^2 \, dx. \]

Finite element approximation: minimise

\[ J_\alpha(c, g_1, g_2, \ldots, g_d) = \frac{1}{n} \sum_{i=1}^{n} (f(x^{(i)}) - y^{(i)})^2 + \alpha \sum_{s=1}^{d} g_s^T L g_s, \]

subject to

\[ Lc = \sum_{s=1}^{d} G_s g_s. \]
2D Formulation

2nd-order smoothing spline: minimise

\[ J_\alpha(f) = \frac{1}{n} \sum_{i=1}^{n} (f(x^{(i)}) - y^{(i)})^2 \]

\[ + \alpha \int_{\Omega} \left( (\partial_1^2 f(x))^2 + 2 (\partial_1 \partial_2 f(x))^2 + (\partial_2^2 f(x))^2 \right) dx, \]

\[ J_\alpha(c, g_1, g_2) = \frac{1}{n} \sum_{i=1}^{n} (b(x^{(i)})^T c - y^{(i)})^2 \]

\[ + \alpha \int_{\Omega} \nabla b^T(x) g_1 \cdot \nabla b^T(x) g_1 + \nabla b^T(x) g_2 \cdot \nabla b^T(x) g_2 \ dx \]
2D Formulation

Minimise:

\[ J_\alpha(c, g_1, g_2) = c^T A c - 2d^T c + \|y\|^2/n + \alpha(g_1^T L g_1 + g_2^T L g_2) \]

subject to

\[ Lc = G_1 g_1 + G_2 g_2. \]

Where

\[ A = \frac{1}{n} \sum_{i=1}^{n} b(x^{(i)}) b(x^{(i)})^T, \]

and

\[ d = \frac{1}{n} \sum_{i=1}^{n} b(x^{(i)}) y^{(i)}. \]
Recall:

Minimise:

\[
J_\alpha(c, g_1, g_2) = c^T A c - 2d^T c + \|y\|^2/n + \alpha(g_1^T L g_1 + g_2^T L g_2)
\]

subject to

\[
L c = G_1 g_1 + G_2 g_2.
\]

\[
\begin{bmatrix}
A & 0 & 0 & L \\
0 & \alpha L & 0 & -G_1^T \\
0 & 0 & \alpha L & -G_2^T \\
L & -G_1 & -G_2 & 0
\end{bmatrix}
\begin{bmatrix}
c \\
g_1 \\
g_2 \\
\hat{w}
\end{bmatrix}
= \begin{bmatrix}
d \\
0 \\
0 \\
0
\end{bmatrix}
- \begin{bmatrix}
h_1 \\
h_2 \\
h_3 \\
h_4
\end{bmatrix},
\]

\(\hat{w}\) is a Lagrange multiplier.

The vectors \(h_1, \ldots, h_4\) store the boundary information. The boundary conditions can be Dirichlet or Neumann.
Boundary Condition Examples - Dirichlet

\[ \mathbf{x}^{(1)} = (0.25, 0.25), \quad \mathbf{x}^{(2)} = (0.75, 0.25), \quad \mathbf{x}^{(3)} = (0.25, 0.75), \]
\[ \mathbf{x}^{(4)} = (0.75, 0.75); \quad y^{(1)} = 1, \quad y^{(2)} = 0, \quad y^{(3)} = 0 \text{ and } y^{(4)} = 1 \]

\[ h_f(\mathbf{x}) = \text{tps fit.} \]
\[ h_f(\mathbf{x}) = 0. \]
Boundary Condition Examples - Neumann

\( x^{(1)} = (0.25, 0.25), \ x^{(2)} = (0.75, 0.25), \ x^{(3)} = (0.25, 0.75), \ x^{(4)} = (0.75, 0.75); \ y^{(1)} = 1, \ y^{(2)} = 0, \ y^{(3)} = 0 \) and \( y^{(4)} = 1 \)
Further Model Problems - Peak

\[
\exp \left( -50 \left( 0.5 - x_1^{(j)} \right)^2 \right) \exp \left( -50 \left( 0.5 - x_2^{(j)} \right)^2 \right) + \epsilon,
\]

where \( \epsilon \) is a normally distributed random variable with mean 0 and a variance of 0.05

Boundary conditions: \( h_f = \tilde{f}_y \), \( h_u = \nabla \tilde{f}_y \) and \( h_\lambda = -\alpha \Delta \tilde{f}_y \).
Further Model Problems - Sin

\[ \sin(4\pi x_1) \sin(4\pi x_2) \]

Finite element grid of size \( m = 4225 \) with different values of \( \alpha \).

Boundary conditions: \( h_f = \tilde{f}_y, h_u = \nabla \tilde{f}_y \) and \( h_{\lambda} = -\alpha \Delta \tilde{f}_y \).  

[M. F. Hutchinson; Communications in Statistics 1989]
\[ y(x, y) = \sin(2\pi x) \sin(2\pi y), \text{ such that } y(x, y) < 0. \]

\[ n = 179401, \ m = 4229 \text{ with } \alpha = 10^{-6} \]

Boundary: \( h_f(x) = y(x), \ h_u = \nabla h_f(x), \ h_\lambda(x) = -\alpha \Delta h_f. \)
Use of FEM Grid

\[(0.25 - \|x^{(j)} - [0.5, 0.5]^T\|_1) \times (0.5 - \|x^{(j)} - [0.5, 0.5]^T\|_1)\]
Use of FEM Grid

Examples of initial FEM grid, with the centre.
Sphere Example

Grid - 189 Nodes, $\alpha = 10^{-3}$.
Sphere Example

Grid - 68705 Nodes, $\alpha = 10^{-3}$. 

![Sphere Grid Example](image-url)
Semi Sphere Example

Grid - 68705 Nodes, $\alpha = 10^{-3}$.
Two Sphere Example

Grid - 68705 Nodes, $\alpha = 10^{-7}$. 
Convergence on Smooth Problem

\[ y^{(i)} = \tilde{f}_y(x^{(i)}) \text{ where } \nabla^4 \tilde{f}_y = 0. \]

\[ \tilde{f}(x) = \tilde{f}_y(x) = \left\| x + \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \right\|_2 \]

\[ \tilde{f}(x) = \tilde{f}_y(x) = \cosh(2\pi x_1) \sin(2\pi x_2) \]
Recall:

\[
\begin{bmatrix}
A & 0 & 0 & L \\
0 & \alpha L & 0 & -G_1^T \\
0 & 0 & \alpha L & -G_2^T \\
L & -G_1 & -G_2 & 0
\end{bmatrix}
\begin{bmatrix}
c \\
g_1 \\
g_2 \\
\hat{w}
\end{bmatrix}
=
\begin{bmatrix}
d \\
0 \\
0 \\
0
\end{bmatrix}
- \begin{bmatrix}
h_1 \\
h_2 \\
h_3 \\
h_4
\end{bmatrix}.
\]

Scale by $1/\alpha$ to get

\[
\begin{bmatrix}
A/\alpha & 0 & 0 & L \\
0 & L & 0 & -G_1^T \\
0 & 0 & L & -G_2^T \\
L & -G_1 & -G_2 & 0
\end{bmatrix}
\begin{bmatrix}
c \\
g_1 \\
g_2 \\
\hat{w}
\end{bmatrix}
=
\begin{bmatrix}
d/\alpha \\
0 \\
0 \\
0
\end{bmatrix}
- \begin{bmatrix}
h_1/\alpha \\
h_2/\alpha \\
h_3/\alpha \\
h_4
\end{bmatrix}.
\]

The vectors $h_1, \cdots, h_4$ store the boundary information. (Assume Dirichlet for now.)
Data Projection Matrix

The matrix $A = BB^T$, where

$$B^T = \frac{1}{\sqrt{n}} \begin{bmatrix} b_1(x^{(1)}) & \cdots & b_m(x^{(1)}) \\ \vdots & \ddots & \vdots \\ b_1(x^{(n)}) & \cdots & b_m(x^{(n)}) \end{bmatrix}$$

is symmetric, nonnegative and sparse.

$B : \text{Data points} \rightarrow \text{FEM.}$

The properties of the $A$ matrix varies with the distribution of the data points.

- If no data points lie in the support of basis function $b_i$, the entries in the $i$th column are zero. In which case $A$, has a non-trivial null-space.
- If any triangles has less than two data points, the system is under-determined system and thus $A$ has a non-trivial null-space.
Let $P = \begin{bmatrix} 0 & I_p \end{bmatrix}$ be a projection onto the region of the domain containing data points. ($I_p$ be an identity matrix).

The matrix $A_p := PAP^T$ is defined on the region of the domain where there are data points.

Note that $A = P^T A_p P$, since $A$ is zero in the region where there are no data points.
Furthermore, $A_p$ may be decomposed as follows

$$A_p = \begin{bmatrix} B_{oo} & B_{ou} \\ 0 & B_{uu} \end{bmatrix} \begin{bmatrix} B_{oo}^T & 0 \\ B_{ou}^T & B_{uu}^T \end{bmatrix} = Q^T \begin{bmatrix} A_{oo}^T & B_{ou} \\ B_{ou}^T & I_u \end{bmatrix} Q,$$

where $I_u$ is an identity matrix, $A_{oo} = B_{oo}B_{oo}^T + B_{ou}B_{ou}^T$ and

$$Q = \begin{bmatrix} I_o & 0 \\ 0 & B_{uu}^T \end{bmatrix}$$

with $I_o$ being an identity matrix.
Let

\[ F = \begin{bmatrix} A_{oo}^T & B_{ou} \\ B_{ou}^T & I_u \end{bmatrix}. \]

Then \( A_p = Q^T F Q \) and \( A = (P^T Q^T) F (QP) \). Note that \( F \) is SPD.
Example Data Distribution

Figure: Example distribution of data in a uniform finite element grid corresponding to the $A$ matrix
Example Data Distribution

**Figure**: Example distribution of data in a uniform finite element grid corresponding to the $A_p$ matrix. The empty triangles have been removed.
Figure: Example distribution of data in a uniform finite element grid corresponding to the $F$ matrix. Work directly on the data points in regions where there are not a lot of data points.
Recall:

\[
S = \begin{bmatrix}
L & -G_1 & -G_2 & 0 \\
0 & L & 0 & -G_1^T \\
0 & 0 & L & -G_2^T \\
A/\alpha & 0 & 0 & L
\end{bmatrix}
\]

If \( P_1 = \begin{bmatrix} 0 & \cdots & 0 & P \end{bmatrix} \) and \( P_2 = \begin{bmatrix} P & 0 & \cdots & 0 \end{bmatrix} \) then \( M + P_1^T A_p P_2 = M + P_1^T Q^T FQP_2 \) is equivalent to above matrix where

\[
M = \begin{bmatrix}
L & -G_1 & -G_2 & 0 \\
0 & L & 0 & -G_1^T \\
0 & 0 & L & -G_2^T \\
0 & 0 & 0 & L
\end{bmatrix}.
\]

This is an upper triangular matrix with the Laplacian matrix sitting along the main diagonal. As the \( L \) matrices are SPD the above system can be readily inverted by using back substitution.
The SMW formula is given by:

\[
\left( M + X_1 F X_2^T \right)^{-1} = M^{-1} - M^{-1} X_1 \left[ F^{-1} + X_2^T M^{-1} X_1 \right]^{-1} X_2^T M^{-1}.
\]

According to the Sherman–Morrison–Woodbury formula:

\[
S^{-1} = M^{-1} - M^{-1} P_1^T Q^T \left[ F^{-1} + Q P H P^T Q^T / \alpha \right]^{-1} Q P_2 M^{-1},
\]

where \( H = L^{-1} \left( \sum_{s=1}^{d} G_s L^{-1} G_s^T \right) L^{-1} \).

- \( M \) is an upper triangular matrix and is easily inverted.
- The Laplacian matrix \( L \) is SPD and \( L^{-1} \) is applied to a vector by using an optimal solver such as the multigrid method.
- Most of the work goes into solving systems of the form \( E_\alpha v = \alpha f \), where \( E_\alpha := \left( \alpha F^{-1} + Q P H P^T Q^T \right) \). The matrix \( E_\alpha \) is SPD so a PCG solver is used.
To measure the conditioning we need to determine the eigenvalues of $L$, $F$ and $G_s$.

We firstly look at a uniform finite element grid defined on the one dimensional domain $[0, 1]$ with grid spacing $h = 1/(m + 1)$. If the domain contains a lot of data spread throughout,

$$E_\alpha = \alpha A_u^{-1} + L^{-1} \left( G_1 L^{-1} G_1^T \right) L^{-1}.$$

Laplacian Matrix

For a uniform finite element grid in one dimension, the stencils for $L$ is

$$L : \frac{1}{h} \begin{bmatrix} -1 & 2 & -1 \end{bmatrix}.$$

Following Hackbusch, we can show the eigenvalues of $L$ are

$$\lambda^\mu(L) = \frac{4}{h} \sin^2 \left( \frac{\pi \mu h}{2} \right),$$

and the eigenvectors are $u^\mu$ where $u^\mu_j = \sin(\pi \mu j h)$ for $1 \leq \mu \leq m$ and $1 \leq j \leq m$. 
Recall:

\[ A = \frac{1}{n} \sum_{j=1}^{n} b(x^{(j)})b(x^{(j)})^T. \]

If there are a lot of data points throughout the whole domain, i.e. \( n \) is large,

\[ A_{k,l} \approx \int_{\Omega} b_k(x)b_l(x) \, dx. \]

We construct a matrix \( A_u \), where \( (A_u)_{k,l} = \int_{\Omega} b_k b_l \, dx \), that is defined by the stencil

\[ A_u : \quad \frac{h}{6} \begin{bmatrix} 1 & 4 & 1 \end{bmatrix}. \]

Consequently, \( A_u \) can be used as an approximation to \( A \).
By using the same eigenvectors that we used for $L$, $u^\mu$, we get

$$\lambda^\mu (A_u) = \frac{h}{3} \left( 2 \cos^2 \left( \frac{\pi \mu h}{2} \right) + 1 \right).$$
We have $u^\mu$ is also an eigenvector for $G_1 L^{-1} G_1^T$ with corresponding eigenvalues

$$
\lambda^\mu \left( G_1 L^{-1} G_1^T \right) = h \cos^2 \left( \frac{\pi \mu h}{2} \right).
$$
Theorem

If

\[ \mathcal{E}_\alpha = \alpha A_u^{-1} + L^{-1} \left( G_1 L^{-1} G_1^T \right) L^{-1}, \]

then

\[ \lambda^\mu(\mathcal{E}_\alpha) = \frac{3\alpha}{h} \frac{1}{2 \cos^2 \left( \frac{\pi \mu h}{2} \right) + 1} + \frac{h^3 \cos^2 \left( \frac{\pi \mu h}{2} \right)}{16 \sin^4 \left( \frac{\pi \mu h}{2} \right)}, \]

\[ \lambda^\mu(A_u \mathcal{E}_\alpha) = \alpha + \frac{h}{3} \left( 2 \cos^2 \left( \frac{\pi \mu h}{2} \right) + 1 \right) \frac{h^3 \cos^2 \left( \frac{\pi \mu h}{2} \right)}{16 \sin^4 \left( \frac{\pi \mu h}{2} \right)}, \]

and

\[ \lambda^\mu(L \mathcal{E}_\alpha L) = \frac{48\alpha}{h^3} \frac{\sin^4 \left( \frac{\pi \mu h}{2} \right)}{2 \cos^2 \left( \frac{\pi \mu h}{2} \right) + 1} + h \cos^2 \left( \frac{\pi \mu h}{2} \right), \]

where \(1 \leq \mu \leq m.\)
Recall:

We want to solve

\[ S^{-1} = M^{-1} - M^{-1} P_1^T Q^T \left[ F^{-1} + QPHP^T Q^T / \alpha \right]^{-1} QP_2 M^{-1}, \]

where \( H = L^{-1} \left( \sum_{s=1}^{d} G_s L^{-1} G_s^T \right) L^{-1} \).

Expensive part:

\[ E_\alpha := \left( \alpha F^{-1} + QPHP^T Q^T \right). \]

(Eg. \( E_\alpha = \alpha A_u^{-1} + L^{-1} (G_1 L^{-1} G_1^T) L^{-1} \). )
Saddle Point Formulation

Recall: (yet again)
\[
\begin{bmatrix}
A/\alpha & 0 & 0 & L \\
0 & L & 0 & -G_1^T \\
0 & 0 & L & -G_2^T \\
L & -G_1 & -G_2 & 0
\end{bmatrix}
\begin{bmatrix}
c \\
g_1 \\
g_2 \\
w
\end{bmatrix}
= \begin{bmatrix}
d/\alpha \\
0 \\
0 \\
0
\end{bmatrix}
- \begin{bmatrix}
h_1/\alpha \\
h_2/\alpha \\
h_3/\alpha \\
h_4
\end{bmatrix}.
\]

Swap the first and last rows to get
\[
\begin{bmatrix}
L & -G_1 & -G_2 & 0 \\
0 & L & 0 & -G_1^T \\
0 & 0 & L & -G_2^T \\
A/\alpha & 0 & 0 & L
\end{bmatrix}
\begin{bmatrix}
c \\
g_1 \\
g_2 \\
w
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
0 \\
d/\alpha
\end{bmatrix}
- \begin{bmatrix}
h_4 \\
h_2/\alpha \\
h_3/\alpha \\
h_1/\alpha
\end{bmatrix}.
\]
Matrix Properties

- Symmetric Indefinite for Dirichlet
- Symmetric and Singular for Neumann;

\[
\text{null space} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.
\]
Conditioning = $O(h^2)$, but large constant.

![Graph showing condition number for different values of $\alpha$]
Matrix Properties

- Use SQMR method as solver directly on saddle point problem (but must be careful about breakdowns.)
- SQMR same as MINRES for SPD preconditioners. Unlike MINRES, SQMR allows the use of an indefinite preconditioner. Not so much theory for SQMR.
Model Problem

\[ y(x) = \exp \left( -50 (x - 0.5)^2 \right). \]

**Figure:** Spline fit to example test problem with \( \alpha = 10^{-2}, 10^{-5}, 10^{-8} \) and grid spacing \( h = 2^{-6} \).
Dirichlet top Neumann did not converge
(Similar for $\alpha = 10^{-5}$ and $\alpha = 10^{-8}$.)
The boundary conditions on each subdomain is Dirichlet, so can use SMW approach to solve problem on each subdomain. I did not find much theory on the use of DD methods with indefinite problems, so did testing with Python on 1D model problem before moving to proper parallel code.
Convergence Rate DD ($\alpha = 10^{-2}$)

DD Preconditioner ($\alpha = 10^{-2}$)

Dirichlet top Neumann did not converge
Convergence Rate DD ($\alpha = 10^{-5}$)

Dirichlet top Neumann did not converge

DD Preconditioner ($\alpha = 10^{-5}$)

Number of Iterations vs. level for different levels and DD Preconditioners.
Convergence Rate DD \( (\alpha = 10^{-8}) \)

Dirichlet top Neumann did not converge
The coarse grid is designed to move the small eigenvalues away from zero.
Convergence Rate DD+Coarse ($\alpha = 10^{-2}$)

![Graph showing convergence rate](image)

Dirichlet top Neumann bottom
Convergence Rate DD+Coarse ($\alpha = 10^{-5}$)

![Graph showing the convergence rate for Dirichlet top and Neumann bottom conditions with coarse preconditioner ($\alpha = 10^{-5}$).](image)

Dirichlet top Neumann bottom
Convergence Rate DD+Coarse ($\alpha = 10^{-8}$)

Dirichlet top Neumann bottom

Linda Stals (Mathematical Sciences Institute, Australian National University)
\[ Q = R_0^T \left( R_0 S R_0^T \right)^{-1} R_0. \]

\[ R = I - QS. \]

\[ P_{BNN}^{-1} = R^T P_{DD}^{-1} R + Q. \]

This technique is similar to coarse grid correction, but removes accumulation of errors.
Convergence Rate DD+Projection ($\alpha = 10^{-2}$)

Dirichlet top Neumann bottom

Linda Stals (Mathematical Sciences Institute, Australian National University)

Discrete thin-plate splines
Convergence Rate DD+Projection ($\alpha = 10^{-5}$)

Dirichlet top Neumann bottom
Convergence Rate DD+Projection \( (\alpha = 10^{-8}) \)

**Projection Preconditioner \( (\alpha = 10^{-8}) \)**

- **Level**: 2, 4, 8, 16, 32, 64
- **Number of Iterations**: 0, 10, 20, 30, 40, 50

**Projection Preconditioner \( (\alpha = 10^{-5}) \)**

- **Level**: 2, 4, 8, 16, 32, 64
- **Number of Iterations**: 0, 10, 20, 30, 40, 50

Dirichlet top Neumann bottom
Original Data Set (641601 data points)
2D Example

FEM fit with 4225 nodes and $\alpha = 10^{-8}$. 
Future Work

- Finish looking into parallel solver.
- Run some large 2D and 3D codes.
Adaptive refinement

- Error indicator
- Reduce/Remove reading data in from files
- Load balancing: take into account connection between FEM grid and data points.
Thank You