Function approximation on manifolds

H. N. Mhaskar
http://www.calstatela.edu/faculty/hmhaska

Question

Let $D \geq d \geq 1$ be integers, $M \subset \mathbf{R}^D$ be a d dimensional manifold, $d \ll D$, $f : M \to \mathbf{R}$. Given data of the form $(x_i, f(x_i))$, approximate f. We will write ρ_M for the geodesic distance on M.

Remark: Many algorithms and theoretical results are known for approximation on cube, sphere, ball, etc.

Novelty: M is not known.

A toy problem

Let $\mathbf{a}, \mathbf{b} \in \mathbf{R}^{100}$, $M \subset \mathbf{R}^{100}$ be defined by

$$\mathbf{x} = \exp(u\mathbf{a} + v\mathbf{b}) + \text{noise}, \qquad u, v \in [-1, 1].$$

Question Given data of the form $(\mathbf{x}_i, u_i + \text{noise})$, but not knowing the function generating $\mathbf{x}_i$, find u as a function of $\mathbf{x}$.

The heat kernel

Coifman, Jones, Lafon, Maggioni, $\cdots$
The (the heat kernel) K_t for the manifold is defined by

$$K_t(x, y) = \sum_{\ell=0}^{\infty} \exp(-\lambda_\ell t)\phi_\ell(x)\phi_\ell(y),$$

where λ_ℓ are eigenvalues of the Laplacian Δ.

Graph Laplacian

Given the data set $\{\mathbf{x}_i\}_{i=1}^N$, let

$$W_{i,j} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\epsilon),$$

$$L_{i,j} = W_{i,j}/\sum_k W_{i,k} - \delta_{i,j}.$$ (Graph Laplacian)

Lafon, Singer, Belkin, Niyogi: For C^∞ f,

$$\epsilon^{-1}\sum_j L_{i,j} f(\mathbf{x}_j) = (\Delta f)(\mathbf{x}_i) + \mathcal{O}(N^{-1/2}\epsilon^{-1/2-d/4}).$$

Δ is the manifold Laplacian.

Semi-supervised learning

- Given $\{(\mathbf{x}_i, u_i)\}_{i=1}^N$, compute the eigenfunctions $\{\phi_j\}_{j=1}^N$ of the graph Laplacian for a suitable ϵ.
- Choose a small subset of the data as training data, and find the least square fit for the first n eigenfunctions based on this data.
- Compute the error on the rest of the set.

Remarks

- A proper choice for ϵ and n can be made by further splitting the training data with one part used for obtaining the training error, and minimizing this error.
- The calculation of eigenvectors may be expensive.
- The calculation of eigenvectors depends also on the test data.
- New data points require a new computation, not clear how to generate new points

Heat triangulation theorem

Jones, Maggioni, Schul

Let $z \in M$, (U, v) be a chart around z, $R_z \leq 1$ be the maximum radius of a ball centered at $v(z)$ contained in $v(U)$, $p_1, \cdots, p_d$ be linearly independent directions, y_i be such that $y_i - z$ is in the direction of p_i, and $c_1 R_z \leq \rho_M(y_i, z) \leq c_2 R_z$, $t = c_3 R_z^2$. Let $\Phi(x) = (K_t(x, y_1), \cdots, K_t(x, y_d))$. Then for all x_1, x_2 with $\rho_M(x_1, z), \rho_M(x_2, z) \leq c_4 R_z$,

$$c_5 \rho_M(x_1, x_2) \leq R_z^{d+1} \|\Phi(x_1) - \Phi(x_2)\| \leq c_6 \rho_M(x_1, x_2).$$

Set up

Let $B_r := \{\mathbf{q} \in \mathbf{R}^d : \|\mathbf{q}\|_d \leq r\}$, $\mathbf{u} : B_1 \to \mathbf{R}^D$,
$J(\mathbf{q}) =$ Jacobian of $\mathbf{u}$, $\|J(\mathbf{q}) - J(\mathbf{0})\| \leq \kappa \|q\|_d$,

$\lambda_{min}\|\mathbf{y}\|_d \leq \|J(\mathbf{q})\mathbf{y}\|_D \leq \lambda_{max}\|\mathbf{y}\|_d$, $\mathbf{y} \in \mathbf{R}^d$, $\mathbf{q} \in \overline{B_{r^*}}$,

Let $\mathbf{p}_1, \cdots, \mathbf{p}_d \in \mathbf{R}^d$ satisfy

$$\left\| \sum_{\ell=1}^{d} y^\ell \mathbf{p}_\ell \right\|_d \geq \gamma \|\mathbf{y}\|_d, \quad \mathbf{y} \in \mathbf{R}^d.$$

Local coordinates

Let $\theta \in (0,1)$, $t_\ell \in [\theta c_1, c_1]$, and $\mathbf{q}_\ell = t_\ell \mathbf{p}_\ell$, $\ell = 1, \cdots, d$. Let $\Phi(\mathbf{w}) = (\|\mathbf{w} - \mathbf{u}(\mathbf{q}_\ell)\|_D)_{\ell=1}^d$, such that for $\mathbf{w}_1, \mathbf{w}_2 \in \mathbf{u}(B_{c_2})$,

$$c_3 \rho_M(\mathbf{w}_1, \mathbf{w}_2) \leq \|\Phi(\mathbf{w}_1) - \Phi(\mathbf{w}_2)\|_D \leq c_4 \rho_M(\mathbf{w}_1, \mathbf{w}_2).$$

For every $\mathbf{y}$ with $\|\mathbf{y} - \Phi(\mathbf{u}(\mathbf{0}))\|_d \leq c_5$, there exists unique $\mathbf{w} \in \mathbf{u}(B_{c_2})$ such that $\mathbf{y} = \Phi(\mathbf{w})$.
Remark The constants can be stated explicitly in terms of $\gamma, \lambda_{min}, \lambda_{max}, \kappa, \theta$.

Some consequences

- With $\mathbf{z}_0 = \mathbf{u}(\mathbf{0})$, we may use standard approximation theory techniques (with scattered data) on the ball $\| \circ - \Phi(\mathbf{z}_0)\|_d \le c_5$ to approximate functions on the image of this ball.
- If the manifold is compact, one can cover it by finitely many such patches.
- Guaranteed approximation rates from classical theory. In particular, the Laplacian can be computed arbitrarily well.

Some consequences

- With $\mathbf{z}_0 = \mathbf{u}(\mathbf{0})$, we may use standard approximation theory techniques (with scattered data) on the ball $\| \circ - \Phi(\mathbf{z}_0)\|_d \le c_5$ to approximate functions on the image of this ball.
- The approximations no longer depend upon the test data.
- New test data can be generated at will.

Quasi–interpolation

Chui–Diamond 1990 Let $\phi : \mathbf{R}^d \to \mathbf{R}$ be a compactly supported function, and λ be a local linear functional such that the operator $Q(f)(\mathbf{x}) := \sum_{\mathbf{k}\in\mathbf{Z}^s} \lambda(f(\circ + \mathbf{k}))\phi(\mathbf{x} - \mathbf{k})$ satisfies $Q(P) = P$ for all polynomials of total degree $\leq m$.

If $\sum_{\mathbf{k}\in\mathbf{Z}^s} |\mathbf{k}|^m |\hat{\phi}(\mathbf{k})| < \infty$ then for $0 \leq r \leq m$,

$$\max_{\mathbf{x}\in\mathbf{R}^s} \left| \partial_r f(\mathbf{x}) - \partial_r Q\Big(f(h\cdot); \mathbf{x}/h \Big) \right| = \mathcal{O}(h^{m+1-r}),$$

as $h \to 0+$.

Approximation of Laplacian

- Get data on the ball of radius c_5.
- Constructions for λ based on scattered data given by M., Narcowich, Ward, 2000.
- Quasi-interpolation gives
  - (i) Other points on the manifold,
  - (ii) Ability to approximate Laplacian

Toy problem set up

$$\mathbf{x} = \exp(u\mathbf{a} + v\mathbf{b}) + \text{noise},$$

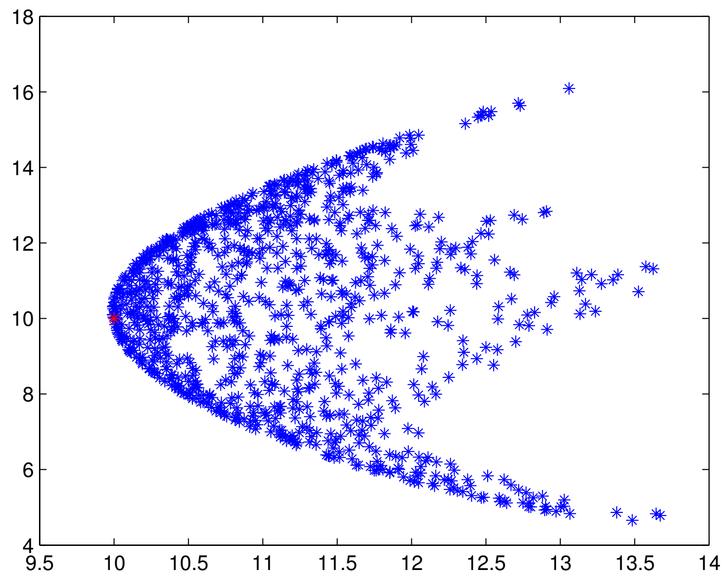
$u, v \in [-0.7, 0.7]$, $\mathbf{a}, \mathbf{b}$, noise chosen uniformly in $[-0.5, 0.5]^{100}$.
Size of the data set =1024, noise in the u values is added before training, uniform in the range $[-0.5, 0.5]$.
training data size= 102, 30 test runs.

Eigenprojections

$\epsilon \longrightarrow, n \downarrow$	0.0010	0.1000	1.0000	10.0000
2	0.4151	0.3596	0.3476	0.3991
10	0.4420	0.1153	0.1036	0.1063
20	0.4962	0.1979	0.1516	0.1616
30	0.5786	0.6547	0.2441	0.4113
60	1.1283	17.1105	2.1603	6.7598

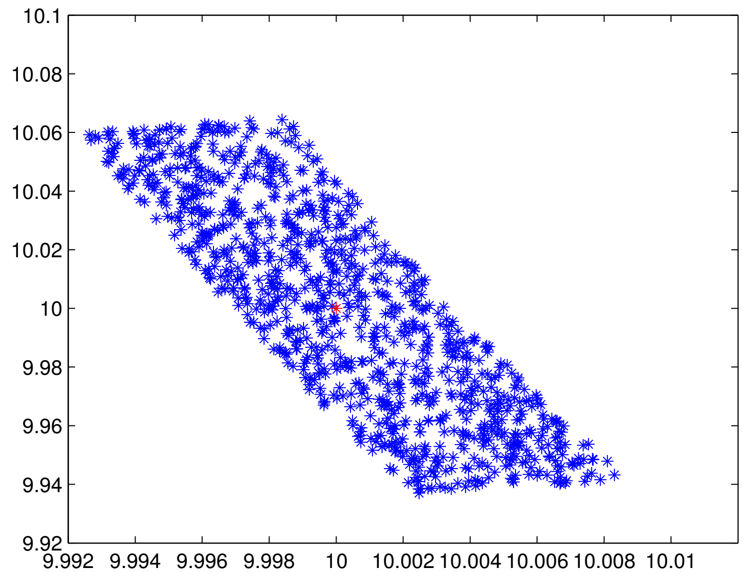
Coordinate region for the toy example

Taking 102 training examples, we used the first 2 singular vectors for this data as the independent directions $\mathbf{u}(\mathbf{q}_\ell)$, $\ell = 1, 2$.



Coordinate region for the toy example

A close up view



Results for the toy example

We just computed the linear least square regression. The mean square error was 0.2405. The time requirement was about 0.6942 seconds vs an average of 4.6783 for the same experiment using any one of the different ϵ's and eigenfunctions.