

Multiscale Data Sampling and Function Extension

A. Bermanis¹, A. Averbuch^{2*}, R.R. Coifman³

¹Department of Applied Mathematics, School of Mathematical Sciences

Tel Aviv University, Tel Aviv 69978, Israel

² School of Computer Science

Tel Aviv University, Tel Aviv 69978, Israel

³Department of Mathematics, Program in Applied Mathematics

Yale University, New Haven, CT 06510, USA

October 9, 2011

Abstract

We introduce a multiscale scheme for sampling scattered data and extending functions defined on the sampled data points, which overcomes some limitations of the Nyström interpolation method. The multiscale extension (MSE) method is based on mutual distances between data points. It uses a coarse-to-fine hierarchy of the multiscale decomposition of a Gaussian kernel. It generates a sequence of subsamples, which we refer to as adaptive grids, and a sequence of approximations to a given empirical function on the data, as well as their extensions to any newly-arrived data point. The subsampling is done by a special decomposition of the associated Gaussian kernel matrix in each scale in the hierarchical procedure.

Keywords: Nyström extension, multiscale, subsampling, Gaussian kernel, diffusion maps, geometric harmonics.

1 Introduction

Many dimensionality reduction methods embed a given data from a metric space, where only the distances between the data points are given, into a lower dimension (vector) space where the data is analyzed.

*Amir Averbuch, e-mail: amir@math.tau.ac.il, Tel: +972-54-5694455, Fax: +972-3-6422020

Dimensionality reduction by Diffusion Maps [5] is a typical example. First, a diffusion operator is formed on the data. Then, by spectral decomposition of the operator, a family of maps $\{\Psi_t\}_{t>0}$ from the data into Euclidean spaces is produced. The Euclidean distances between the embedded data points approximate the diffusion distances between the data points in the genuine metric space, i.e. it becomes the transition probability in t time steps from one data point to another. A spectral decomposition of large matrices, whose dimensions are proportional to the size of the data, has high computational costs. Especially, this procedure can not be repeated frequently when data is accumulated over time. To avoid repeated application of such procedure, an extension method is required, which is called an out-of-sample extension.

The Nyström method [1, 13] is vastly used for an out-of-sample extension in dimensionality reduction methods. It is a numerical scheme for the extension of integral operator eigenfunctions. It finds a numerical approximation for the eigenfunction problem

$$\int_a^b G(x, y)\phi(y)dy = \lambda\phi(x) \quad (1.1)$$

where ϕ is an eigenfunction and λ is the corresponding eigenvalue. Given a set of equidistant points $\{x_j\}_{j=1}^n \subset [a, b]$, Eq. 1.1 can be approximated by a quadrature rule to become

$$\frac{b-a}{n} \sum_{j=1}^n G(x_i, x_j)\phi(x_j) = \lambda\phi(x_i).$$

Then, the Nyström extension of ϕ to a new data point x_* is

$$\hat{\phi}(x_*) \triangleq \frac{b-a}{n\lambda} \sum_{j=1}^n G(x_*, x_j)\phi(x_j). \quad (1.2)$$

If G is symmetric, then its normalized eigenfunctions $\{\phi_i\}_{i=1}^n$ constitute an orthonormal basis to \mathbb{R}^n . Thus, any vector $f = [f_1 f_2 \dots f_n]^T$, ($f_j = f(x_j)$, $j = 1, \dots, n$) can be decomposed into a superposition of its eigenvectors $f = \sum_{i=1}^n (f^T \cdot \phi_i) \phi_i$. Then, the Nyström extension of f to x_* becomes

$$f_* \triangleq \sum_{i=1}^n (f^T \cdot \phi_i) \hat{\phi}_i(x_*). \quad (1.3)$$

The Nyström extension method is strongly related to a Gaussian process regression (GPR) [14], which is an extension method in the field of statistical inference. The n observations $\{f_1, f_2, \dots, f_n\}$ are considered as a sample from some multivariate (n -variate) Gaussian distribution. Very often, it is assumed that the mean of this Gaussian distribu-

tion is zero. The observations are related to each other in these cases by the covariance function g . Thus, the conditional distribution of f_* given f is

$$f_*|f \sim \mathcal{N}(G_*G^{-1}f, G_{**} - G_*G^{-1}G_*^T), \quad (1.4)$$

where $\mathcal{N}(\mu, \sigma)$ is the normal distribution whose mean and variance are μ and σ , respectively,

$$G \triangleq \begin{bmatrix} g(x_1, x_1) & g(x_1, x_2) & \cdots & g(x_1, x_n) \\ g(x_2, x_1) & g(x_2, x_2) & \cdots & g(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ g(x_n, x_1) & g(x_n, x_2) & \cdots & g(x_n, x_n) \end{bmatrix}, \quad (1.5)$$

$$G_* \triangleq \begin{bmatrix} g(x_*, x_1) & g(x_*, x_2) & \cdots & g(x_*, x_n) \end{bmatrix}, \quad (1.6)$$

and

$$G_{**} \triangleq g(x_*, x_*). \quad (1.7)$$

As a consequence, the best estimate for f_* is the mean of this distribution

$$\bar{f}_* = G_*G^{-1}f, \quad (1.8)$$

and the uncertainty in this estimation is captured by its variance

$$\text{var}(f_*) = G_{**} - G_*G^{-1}G_*^T. \quad (1.9)$$

Note that Eqs. 1.2 and 1.3 are equivalent to Eq. 1.8.

In the field of geostatistics, where the Nyström extension is better known as Kriging, the covariance between any pair of observations is usually related directly to the geological nature of the data [11, 15], as opposed to our setup, where data comes from an unknown physical phenomenon.

When the covariance is unknown, an artificial covariance function has to be chosen. A Gaussian covariance is a popular choice as the covariance function, and it is given by

$$g_\epsilon(x, x') \triangleq \exp\left(-\|x - x'\|^2/\epsilon\right), \quad (1.10)$$

where $\|\cdot\|$ constitutes a metric on the space. The corresponding covariance (affinities) matrix is

$$(G_\epsilon)_{i,j} \triangleq g_\epsilon(x_i, x_j), \quad i, j = 1, 2, \dots, n. \quad (1.11)$$

Under this choice, if $x \approx x'$, then $g_\epsilon(x, x')$ approaches the maximum, which means

that $f(x)$ is nearly perfectly correlated with $f(x')$. The vise-versa is also true: if x is distant from x' , we have $g_\epsilon(x, x') \approx 0$ instead, i.e. the two data points cannot see each other. For example, during interpolation at new values, distant observations will have negligible effect. The effect of this separation will depend on the length parameter ϵ . A too large ϵ will result in ill-conditioned (i.e. numerically singular) covariance matrix, as proved in Section 3. On the other hand, due to Eqs. 1.8 and 1.9, interpolation with too small ϵ can be done only in a very small neighborhood of D , otherwise, the variance in Eq. 1.9 from a statistical point of view reaches near its maximum, and $f(x) \approx 0$.

The Nyström extension scheme has three significant disadvantages: (a) Diagonalization of G costs $O(n^3)$ operations ([9]). (b) G may be ill-conditioned due to fast decay of its spectrum, and (c) it is unclear how to choose the length parameter ϵ since the output is sensitive to the choice of ϵ .

We overcome these limitations by using a multiscale approach: we define a sequence of Gaussian kernel matrices G_s , $s = 0, 1, \dots$, whose entries are $(G_s)_{i,j} = g_{\epsilon_s}(x_i, x_j)$, where ϵ_s is a positive monotonic decreasing function of s , which tends to zero as the scale parameter s tends to infinity. For example, we can choose $\epsilon_s = 2^{-s}$, $s = 0, 1, \dots$. By the application of randomized interpolative decomposition (ID) to G_s , we identify a well-conditioned basis for its numerical range. In each scale, f (or its residual) is decomposed into a sum of its projections on this basis and it is extended in a similar fashion to Eq. 1.8. In addition, selection of the proper columns in G_s is equivalent to data sampling of the associated data points.

Our method requires no grid. It automatically generates a sequence of adaptive grids according to the data distribution (see Example 4.1). It is based on the mutual distances between the data points and on a continuous extension of Gaussian functions. In addition, most of the costly computations are done just once during the process, independently of the number of the extended data points since they depend only on the data and on the given function.

A preliminary version of the paper was presented in [2]

The paper has the following structure: Section 2 presents related works on (multiscale) data sampling and function extension. In Section 3, we prove that the numerical rank of a Gaussian kernel, which is defined on a dataset in \mathbb{R}^d , is proportional to the volume of the data. A multiscale scheme for data sampling and function extension is presented in Section 4. Experimental results are given in Section 5.

2 Related works

A multiscale scheme for scattered data sampling and interpolation is introduced in [8]. In each scale, Delaunay triangulations is employed to sample the data. Then, using the sampled data, the function (or its residual) is interpolated to the rest of the data points by using a radial basis functions (RBF) technique.

The method of geometric harmonics is introduced in [6]. First, the function, which is defined on a manifold, is decomposed into a superposition of the eigenfunctions of the manifold's Laplace-Beltrami operator. Then, these eigenfunction are extended using the Nyström extension of the eigenfunctions of a sequence of Bessel kernels. It is proved that the extension is optimal in the sense of maximal energy concentration of the data points. This method is strongly related to the the Kriging method, which was already mentioned in section 1.

3 Numerical rank of a Gaussian Kernel matrix

In this section, we prove that the number of numerically independent columns of a Gaussian kernel matrix (i.e., its numerical rank) is independent of its size. First, we prove it for the unit circle \mathbb{S}^1 . Then, we generalize the result to \mathbb{R} and \mathbb{R}^d . To formalize this assertion we will need the following definition:

Definition 3.1. The *numerical rank* of a matrix $K \in \mathbb{C}^{m \times n}$ up to precision $\delta > 0$ is

$$R_\delta(K) \triangleq \# \left\{ j : \frac{\sigma_j(K)}{\sigma_0(K)} \geq \delta \right\},$$

where $\sigma_j(K)$ denotes the j -th largest singular value of the matrix K .

3.1 Samples on \mathbb{S}^1

In this section, we prove that the numerical rank of a Gaussian kernel matrix, which is defined on the unit circle, is independent of the number of data points while it depends only on the length parameter ϵ .

Assume C is an $n \times n$ circulant matrix C , whose first row is $\vec{\gamma} = (c_0, \dots, c_{n-1})$. It is denoted by $C \triangleq \text{circ}(\vec{\gamma})$. The n -th principal root of unity is denoted by $\omega_n = e^{i\frac{2\pi}{n}}$. According to Theorem 3.2.2 in [7], the eigenvalues of C are

$$\lambda_j = \sum_{k=0}^{n-1} c_k \omega_n^{jk}, \quad j = 0, 1, \dots, n-1. \quad (3.1)$$

A special case occurs when C is also real and symmetric.

Lemma 3.2. *Let $C = \text{circ}(\vec{\gamma})$ be an $n \times n$ symmetric matrix, where $\vec{\gamma} = (c_0, \dots, c_{n-1}) \in \mathbb{R}^n$. Then, the eigenvalues of C are*

$$\lambda_j = \sum_{k=0}^{n-1} c_k \omega_n^{-jk}, j = 0, 1, \dots, n-1. \quad (3.2)$$

In addition, $\lambda_j = \lambda_{n-j}$ for any $j = 1, 2, \dots, n-1$.

Proof. C is symmetric, hence, its eigenvalues are all real. Since $\vec{\gamma} \in \mathbb{R}^n$ then by conjugating Eq. 3.1 we get Eq. 3.2. In addition, since $\omega_n^n = 1$, we get

$$\lambda_{n-j} = \sum_{k=0}^{n-1} c_k \omega_n^{(n-j)k} = \sum_{k=0}^{n-1} c_k \omega_n^{-jk} = \lambda_j, \quad (3.3)$$

for any $j = 1, \dots, n-1$. □

The right hand side of Eq. 3.2 is the discrete Fourier transform (DFT, see [16]) of c_0, \dots, c_{n-1} .

Suppose that $\{\theta_j\}_{j=0}^{n-1}$ are n equidistant samples on the unit circle and that $C_\epsilon^{(n)}$ is the associated $n \times n$ Gaussian kernel matrix, i.e.

$$(C_\epsilon^{(n)})_{i,j} \triangleq g_\epsilon(\theta_i, \theta_j), i, j = 1, \dots, n, \quad (3.4)$$

where g_ϵ was defined in Eq. 1.10 with the arc-length metric $|\cdot|_{\mathbb{S}^1}$. Due to Bochner's theorem (see [17]), $C_\epsilon^{(n)}$ is a positive definite matrix. Additionally, $C_\epsilon^{(n)}$ is symmetric, hence, its eigenvalues coincide with its singular values. Therefore, due to Lemma 3.2, the singular values of $C_\epsilon^{(n)}$ satisfy

$$\frac{1}{n} \sigma_j(C_\epsilon^{(n)}) = \hat{g}_\epsilon \left[\left[\frac{j}{2} \right] \right], j = 0, 1, \dots, n-1, \quad (3.5)$$

where \hat{g}_ϵ is the discrete Fourier transform of g_ϵ , i.e.

$$\hat{g}_\epsilon[\omega] = \frac{1}{n} \sum_{j=0}^{n-1} e^{-\frac{x_j^2}{\epsilon}} e^{-i\omega x_j}, x_j = -\pi + \frac{2\pi}{n}j, j = 0, 1, \dots, n-1. \quad (3.6)$$

Since g_ϵ is Riemann-integrable on $[-\pi, \pi]$ it satisfies

$$\lim_{n \rightarrow \infty} \hat{g}_\epsilon[\omega] = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-\frac{x^2}{\epsilon}} e^{-i\omega x} dx. \quad (3.7)$$

Lemma 3.3. Let $C_\epsilon^{(n)}$ be the $n \times n$ Gaussian kernel matrix defined by Eqs. 1.10 and 3.4 for a set of n equidistant points on the unit circle. Then, for any $\epsilon > 0$

$$\lim_{n \rightarrow \infty} R_\delta (C_\epsilon^{(n)}) \leq 4\sqrt{\epsilon^{-1} \ln(\delta^{-1})} + 1.$$

Proof. Due to Eqs. 3.5 - 3.7, for any $j = 0, 1, \dots, n - 1$

$$\lim_{n \rightarrow \infty} \frac{\sigma_j (C_\epsilon^{(n)})}{\sigma_0 (C_\epsilon^{(n)})} = \frac{\int_{-\pi}^{\pi} e^{-\frac{x^2}{\epsilon}} \cos\left(\left[\frac{j}{2}\right] x\right) dx}{\int_{-\pi}^{\pi} e^{-\frac{x^2}{\epsilon}} dx}. \quad (3.8)$$

In order to compute the quotient in Eq. 3.8, we use the Taylor expansion of $\cos(tx)$:

$$\begin{aligned} \int_{-\pi}^{\pi} e^{-\frac{x^2}{\epsilon}} \cos(tx) dx &= \int_{-\pi}^{\pi} e^{-\frac{x^2}{\epsilon}} \sum_{j=0}^{\infty} \frac{(-1)^j (tx)^{2j}}{(2j)!} dx \\ &= \sum_{j=0}^{\infty} \frac{(-1)^j t^{2j}}{(2j)!} \int_{-\pi}^{\pi} e^{-\frac{x^2}{\epsilon}} x^{2j} dx \\ &= \sum_{j=0}^{\infty} \frac{(-1)^j t^{2j}}{(2j)!} P_{2j}, \end{aligned} \quad (3.9)$$

where

$$P_{2j} \triangleq \int_{-\pi}^{\pi} e^{-\frac{x^2}{\epsilon}} x^{2j} dx, j = 0, 1, \dots$$

By using integration by parts we get

$$\begin{aligned} P_{2j} &= \int_{-\pi}^{\pi} e^{-\frac{x^2}{\epsilon}} x \cdot x^{2j-1} dx \\ &= -\frac{\epsilon}{2} e^{-\frac{x^2}{\epsilon}} x^{2j-1} \Big|_{-\pi}^{\pi} + \frac{\epsilon}{2} (2j-1) P_{2j-2}. \end{aligned} \quad (3.10)$$

Since the first term in eq. 3.10 is negative, we get

$$P_{2j} < P_0 \left(\frac{\epsilon}{2}\right)^j (2j-1)!!, \quad (3.11)$$

where $(2j-1)!! \triangleq (2j-1) \cdot (2j-3) \cdot \dots \cdot 1$.

By substituting Eq. 3.11 into Eq. 3.9 we get

$$\begin{aligned}
\int_{-\pi}^{\pi} e^{-\frac{x^2}{\epsilon}} \cos(tx) dx &< P_0 \sum_{j=0}^{\infty} \frac{(-1)^j t^{2j}}{(2j)!} \left(\frac{\epsilon}{2}\right)^j (2j-1)!! \\
&= P_0 \sum_{j=0}^{\infty} \left(\frac{-t^2\epsilon}{4}\right)^j \frac{1}{j!} \\
&= P_0 e^{-\frac{t^2\epsilon}{4}}.
\end{aligned}$$

Obviously, the denominator in Eq. 3.8 equals to P_0 . Thus

$$\lim_{n \rightarrow \infty} \frac{\sigma_k \left(C_{\epsilon}^{(n)}\right)}{\sigma_0 \left(C_{\epsilon}^{(n)}\right)} < \exp\left(-\frac{\epsilon}{4} \left\lceil \frac{k}{2} \right\rceil^2\right).$$

The last quotient is less than δ if and only if $k > 4\sqrt{\epsilon^{-1} \ln(\delta^{-1})}$. □

Proposition 3.4 concludes the present section. It provides an upper bound for the numerical rank of a Gaussian kernel matrix associated with an arbitrary set on the unit circle.

Proposition 3.4. *Let $\{\tilde{\theta}_k\}_{k=1}^m$ be an arbitrary set on \mathbb{S}^1 and let G_{ϵ} be the associated Gaussian kernel matrix, i.e.*

$$(G_{\epsilon})_{i,j} = g_{\epsilon}(\tilde{\theta}_i, \tilde{\theta}_j), \quad i, j = 1, 2, \dots, n,$$

where g_{ϵ} was defined in Eq. 1.10 with the arc-length metric $|\cdot|_{\mathbb{S}^1}$. Then, for any $0 < \delta \leq 1$,

$$R_{\delta}(G_{\epsilon}) \leq 4\sqrt{\epsilon^{-1} \ln(\delta^{-1})} + 1.$$

Proof. Let $\Theta_n = \{\theta_i^n\}_{i=1}^n$ be a set of n ($n > m$) equidistant points in \mathbb{S}^1 that are indexed s.t.

$$\left|\tilde{\theta}_k - \theta_k^n\right|_{\mathbb{S}^1} = \min_{i=1, \dots, n} \left|\tilde{\theta}_k - \theta_i^n\right|_{\mathbb{S}^1}, \quad k = 1, \dots, m.$$

Obviously,

$$\lim_{n \rightarrow \infty} \theta_k^n = \tilde{\theta}_k, \quad k = 1, \dots, m. \tag{3.12}$$

Let $C_{\epsilon}^{(n)}$ be the Gaussian matrix defined by Eqs. 3.4 and 1.10, which is associated with Θ_n . In addition, let P_n be the $m \times n$ projection matrix on $\{\theta_k^n\}_{k=1}^m$, i.e.

$$P_n = \left[I_m \mid 0 \right],$$

where I_m is an $m \times m$ identity matrix and 0 is an $m \times (n - m)$ zeros matrix. Due to Eq. 3.12 and since g_ϵ is continuous,

$$\lim_{n \rightarrow \infty} \|P_n C_\epsilon^{(n)} P_n^T - G_\epsilon\|_{\max} = 0,$$

or equivalently,

$$\lim_{n \rightarrow \infty} \|P_n C_\epsilon^{(n)} P_n^T - G_\epsilon\|_2 = 0.$$

Thus, from Weyl's inequality (see [3])

$$\lim_{n \rightarrow \infty} \sigma_j(P_n C_\epsilon^{(n)} P_n^T) = \sigma_j(G_\epsilon), j = 1, \dots, m.$$

As a consequence,

$$\lim_{n \rightarrow \infty} R_\delta(P_n C_\epsilon^{(n)} P_n^T) = R_\delta(G_\epsilon).$$

For any $n \in \mathbb{N}$, $R_\delta(P_n C_\epsilon^{(n)} P_n^T) \leq R_\delta(C_\epsilon^{(n)})$ therefore, due to Lemma 3.3, we get

$$R_\delta(C_\epsilon^{(n)}) \leq 4\sqrt{\epsilon^{-1} \ln(\delta^{-1})} + 1.$$

□

Corollary 3.5. *If the dataset lies on a half circle, then the numerical rank of the associated Gaussian kernel matrix is less than $2\sqrt{\epsilon^{-1} \ln(\delta^{-1})} + 1$.*

3.2 Samples on \mathbb{R}

In this section, we generalize Proposition 3.4 to an interval in \mathbb{R} . We provide an upper bound to the numerical rank of a Gaussian kernel matrix associated with a dataset on the real line. This bound depends on the length of the minimal interval where the data points lie in and on the length parameter ϵ . It is independent of the number of data points. For a fixed length parameter ϵ of the Gaussian, length l of the minimal bounding interval and accuracy δ , the bound is given by the following constant

$$C(l, \epsilon, \delta) \triangleq \frac{2l}{\pi} \sqrt{\epsilon^{-1} \ln(\delta^{-1})} + 1, \quad (3.13)$$

as Proposition 3.6 proves:

Proposition 3.6. *Let $\{x_i\}_{i=1}^n \subset [a, b] \subset \mathbb{R}$, and let G_ϵ be the associated Gaussian kernel matrix, i.e.*

$$(G_\epsilon)_{i,j} = g_\epsilon(x_i, x_j), i, j = 1, \dots, n,$$

where g_ϵ is given in Eq. 1.10 with the standard Euclidean norm on \mathbb{R} . Then,

$$R_\delta(G_\epsilon) \leq C(b-a, \epsilon, \delta).$$

Proof. Define $y_j \triangleq \frac{x_j - a}{b-a} \pi$. Then, $y_j \in [0, \pi], j = 1, \dots, n$, hence, the Euclidean and the arc-length metrics coincide for $\{y_j\}_{j=1}^n$. We get

$$|x_i - x_j| = \frac{b-a}{\pi} |y_i - y_j| = \frac{b-a}{\pi} |y_i - y_j|_{\mathbb{S}^1}. \quad (3.14)$$

Then, $g_\epsilon(x_i, x_j) = \exp\left(-\frac{(b-a)^2}{\pi^2 \epsilon} |y_i - y_j|_{\mathbb{S}^1}^2\right)$. Therefore, due to Corollary 3.5, $R_\delta(G_\epsilon) \leq C(b-a, \epsilon, \delta)$. \square

3.3 Samples on \mathbb{R}^d

Our next goal is to generalize Proposition 3.6 to \mathbb{R}^d :

Proposition 3.7. *Let $Z = \{z_i\}_{i=1}^n \subset \mathbb{R}^d$ be a set bounded by a box $B = I_1 \times I_2 \times \dots \times I_d$, where I_1, I_2, \dots, I_d are intervals in \mathbb{R} , and let G_ϵ be the associated Gaussian kernel matrix, i.e. $(G_\epsilon)_{i,j} = g_\epsilon(x_i, x_j), i, j = 1, \dots, n$ where g_ϵ is given in Eq. 1.10 with the standard Euclidean norm in \mathbb{R}^d . Then,*

$$R_\delta(G_\epsilon) \leq \prod_{i=1}^d C(|I_i|, \epsilon, \delta). \quad (3.15)$$

Proposition 3.7 will be proved for $d = 2$. Generalization to any higher dimension is straightforward. The proof is based on Theorem 4.2.12 in [10], which states that for any two matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{m \times m}$ the eigenvalues of the $mn \times mn$ tensor product matrix $A \otimes B$ are

$$\sigma_{jk}(A \otimes B) = \sigma_j(A) \sigma_k(B). \quad (3.16)$$

Proof of Proposition 3.7 for $d = 2$. To simplify the proof, we assume that $Z = X \times Y$, where $X \subset I_1$ and $Y \subset I_2$. Then, $G_\epsilon = G_\epsilon^X \otimes G_\epsilon^Y$, where G_ϵ^X and G_ϵ^Y are the Gaussian kernel matrices, which are associated with X and Y , respectively. Suppose that r_x and r_y are the numerical ranks of G_ϵ^X and G_ϵ^Y , respectively. Thus, due to Eq. 3.16,

$$\frac{\sigma_{r_x \cdot r_y}(G_\epsilon)}{\sigma_0(G_\epsilon)} = \frac{\sigma_{r_x}(G_\epsilon^X)}{\sigma_0(G_\epsilon^X)} \cdot \frac{\sigma_{r_y}(G_\epsilon^Y)}{\sigma_0(G_\epsilon^Y)} \leq \delta^2 \leq \delta.$$

As a consequence, $R_\delta(G_\epsilon) \leq R_\delta(G_\epsilon^X) \cdot R_\delta(G_\epsilon^Y)$. \square

Since the box B does not have to be parallel to the axes, Proposition 3.7 states that the numerical rank of the Gaussian kernel is proportional to the volume of the *minimal* bounding box B and to $\epsilon^{-d/2}$. If the data lies on a \tilde{d} -dimensional hyperplane in \mathbb{R}^d ($\tilde{d} < d$), then the minimal bounding box is also \tilde{d} -dimensional i.e., there are $d - \tilde{d}$ intervals whose lengths are zero. In this case, due to Eqs. 3.13 and 3.15, the numerical rank of the Gaussian kernel is bounded by a constant that depends on the *intrinsic* dimension of the data. When the data lies on a \tilde{d} -dimensional manifold, where curvature is involved, the analysis is more complicated. This issue will be treated by us in a future work.

3.4 From bounding box to ϵ -cover

In this section, we provide a finer bound for the numerical rank of the Gaussian kernel matrix. For the sake of demonstration, suppose that the data divided into two distant clusters. Then, most of the bounding box contains no data. In this case, it is more accurate to cover the dataset by two bounding boxes.

The conclusion of the present section is that the numerical rank is bounded from above by a constant, which is proportional to the minimal number of cubes whose side length is $\epsilon^{d/2}$, that is required to cover the data.

In order to prove the above, we will need Lemmas 3.8 and 3.9 and Definition 3.10. Lemma 3.8 provides a criterion for a matrix to have a certain numerical rank. In Lemma 3.9, we prove that the numerical rank of a block diagonal matrix is not bigger than the sum of the numerical rank of its blocks.

Lemma 3.8. $R_\delta(K) \leq l$ if and only if there exists a matrix M whose rank is l , s.t. $\|K - M\|_2 < \delta \|K\|_2$.

Proof. If $R_\delta(K) \leq l$ then $\sigma_{l+1}(K) < \delta \|K\|_2$. Define M to be the l -SVD of K . Then, M is of rank l and $\|K - M\|_2 \leq \sigma_{l+1}(K)$. Now, assume that M is a matrix of rank l s.t. $\|K - M\|_2 < \delta \|K\|_2$. Then, according to Weyl's inequality (see [3]), we get $\sigma_{l+1}(K) = |\sigma_{l+1}(K) - \sigma_{l+1}(M)| \leq \|K - M\|_2 < \delta \|K\|_2$, i.e., $\frac{\sigma_{l+1}(K)}{\|K\|_2} < \delta$ and particularly $R_\delta(K) \leq l$. \square

Lemma 3.9. *Let*

$$K = \left[\begin{array}{c|c} A & 0 \\ \hline 0 & B \end{array} \right], \quad (3.17)$$

where A and B are $l \times l$ and $m \times m$ matrices, respectively, then

$$R_\delta(K) \leq R_\delta(A) + R_\delta(B). \quad (3.18)$$

Proof. Let $r_A = R_\delta(A)$ and $r_B = R_\delta(B)$. According to Lemma 3.8, there exist two matrices \tilde{A} and \tilde{B} whose ranks are r_A and r_B , respectively, s.t. $\|A - \tilde{A}\|_2 < \delta\|A\|_2$ and $\|B - \tilde{B}\|_2 < \delta\|B\|_2$. Define

$$\tilde{K} = \left[\begin{array}{c|c} \tilde{A} & 0 \\ \hline 0 & \tilde{B} \end{array} \right].$$

Obviously, the rank of \tilde{K} is $r_A + r_B$. In addition, $\|K - \tilde{K}\|_2 \leq \max\{\|A - \tilde{A}\|_2, \|B - \tilde{B}\|_2\} < \delta \max\{\|A\|_2, \|B\|_2\} = \delta\|K\|_2$. Hence, according to Lemma 3.8, we get that $R_\delta(K) \leq r_A + r_B$. \square

Definition 3.10.

1. The distance between two sets A and B in a metric space whose metric is $m(\cdot, \cdot)$, is $dist(A, B) \triangleq \sup_{a \in A} \inf_{b \in B} m(a, b)$.
2. A connected component of a dataset $X \subset \mathbb{R}^d$, with respect to $\epsilon > 0$ and $\eta > 0$, is a nonempty subset $Y \subset X$, s.t. $dist(Y, X \setminus Y) \geq \sqrt{\epsilon \ln(\eta^{-1})}$. Thus, the affinities between X and $Y \setminus X$ (Eqs. 1.10 and 1.11) are less than η . Typically we use $\eta = 10^{-10}$.
3. The closure \bar{Y} of a connected component Y is a d -dimensional box that contains Y , whose volume is minimal. This volume is denoted by $|\bar{Y}|$.
4. The closure of a dataset X is $\bar{X} \triangleq \bigcup_{j=1}^c \bar{Y}_j$, where $\{Y_j\}_{j=1}^c$ are the connected components of X . Its volume is $|\bar{X}| \triangleq \sum_{j=1}^c |\bar{Y}_j|$.

Proposition 3.11. *Let $X = \bigcup_{j=1}^c Y_j \subset \mathbb{R}^d$, where $\{Y_j\}_{j=1}^c$ are the connected components of X and let G_ϵ be the associated Gaussian kernel. Then, $R_\delta(G_\epsilon) \leq \sum_{j=1}^c C(|\bar{Y}_j|, \epsilon, \delta)$.*

Proof. If $c = 1$, then, due to Proposition 3.6, $R_\delta(G_\epsilon) \leq C(|\bar{X}|, \epsilon, \delta)$. Otherwise, assume that Y_1 and Y_2 are two connected sets of X and $K_\epsilon^{(1)}$ and $K_\epsilon^{(2)}$ are the associated kernel matrices, respectively. Then, we get

$$G_\epsilon = \left[\begin{array}{c|c} K_\epsilon^{(1)} & D \\ \hline D^T & K_\epsilon^{(2)} \end{array} \right]. \quad (3.19)$$

Since $dist(Y_1, Y_2) \geq \sqrt{\epsilon \ln(\eta^{-1})}$, numerically $D = 0$. In this case, due to Lemma 3.9 and Theorem 3.6, we get $R_\delta(K) \leq C(|\bar{Y}_1|, \epsilon, \delta) + C(|\bar{Y}_2|, \epsilon, \delta)$. \square

We conclude the present section with Proposition 3.12. It generalizes Proposition 3.11 to the case where X consists of more than two connected components. The proof is the

same as the proof of Proposition 3.11 and it is based on the generalization of Lemma 3.9 to any number of blocks.

Proposition 3.12. *If $Z = \bigcup_{j=1}^c Y_j$, where $\{Y_j\}_{j=1}^c$ are the connected components of Z , then,*

$$R_\delta(G_\epsilon) \leq \sum_{j=1}^c \left\{ \prod_{i=1}^d C(|I_i^{(j)}|, \epsilon, \delta) \right\}, \quad (3.20)$$

where G_ϵ is the associated Gaussian kernel matrix of Z and $\bar{Y}_j = I_1^{(j)} \times I_2^{(j)} \dots \times I_d^{(j)}$, $j = 1, 2, \dots, c$, i.e., the numerical rank of G_ϵ is bounded by

$$\left(\frac{2}{\pi} \sqrt{\ln(\delta^{-1})} \right)^d \times \{\text{minimal cover of } \bar{Z} \text{ by cubes of volume } (\sqrt{\epsilon})^d\}.$$

4 Multiscale sampling and functions extension

Given a real function $f = [f_1, \dots, f_n]^T$ on a dataset $D = \{x_1, x_2, \dots, x_n\}$ in \mathbb{R}^d ($f_i = f(x_i)$, $i = 1, \dots, n$). Our goal is to extend f to any data point in \mathbb{R}^d by a superposition of Gaussians, which are centered at D . Due to Bochner's theorem [17], the Gaussian kernel G_ϵ is strictly positive-definite, therefore, theoretically we can use the Nyström extension by:

1. Calculate the coordinates vector $c = (c_1, c_2, \dots, c_n)^T$ of f in the basis of G_ϵ 's columns such that

$$c = G_\epsilon^{-1} f. \quad (4.1)$$

2. Extend f to $x_* \in \mathbb{R}^d$ by a natural extension of the Gaussians to x_* such that

$$f_* \triangleq \sum_{j=1}^N g_\epsilon(x_*, x_j) c_j. \quad (4.2)$$

As proved in Section 3, G_ϵ may be ill conditioned, i.e., numerically non-invertible. In addition, as previously mentioned, inversion of G_ϵ costs $\mathcal{O}(n^3)$ floating-point operations and it is unclear how to choose the length parameter ϵ .

From now on we will use a terminology of scales rather than length parameters. Thus, we define

$$G^{(s)} \triangleq G_{\epsilon_s}, \quad (4.3)$$

where s denotes the scale and $\{\epsilon_s\}_{s=0}^\infty$ is a decreasing positive sequence that tends to zero as s tends to infinity.

In order to overcome the limitations mentioned above, we use the following multiscale two-phase scheme:

1. Sampling: a well-conditioned basis of $G^{(s)}$'s columns is identified. Accordingly, the sampled dataset is the set of data points, which are associated with these columns. This phase overcomes the problem arises from the numerical singularity of $G^{(s)}$.
2. Extension: f is projected on this basis. Then, $f^{(s)}$, which is the projection of f on this basis, is extended by a continuous extension of the involved Gaussians to x_* in a similar way to Eq. 4.2.

Of course, f does not have to be equal to its projection $f^{(s)}$. In this case, we apply the procedure to $f - f^{(s)}$ with $G^{(s+1)}$ whose numerical rank (i.e., its number of numerically independent columns) is bigger than the numerical rank of $G^{(s)}$, as was proved in Section 3. Thus, we get a multiscale scheme for data sampling and function extension.

4.1 Phase 1: Single-scale data sampling

Suppose that $l^{(s)}$ is the numerical rank of the $n \times n$ Gaussian kernel matrix $G^{(s)}$ for a fixed scale s . Our goal is to identify the $l^{(s)}$ columns of $G^{(s)}$, which constitute a well-conditioned basis for its numerical range. In other words, we are looking for an $n \times l^{(s)}$ matrix $B^{(s)}$, whose columns constitute a subset of the columns of $G^{(s)}$ and an $l^{(s)} \times n$ matrix $P^{(s)}$, s.t. $l^{(s)}$ of its columns make up the identity matrix and $B^{(s)}P^{(s)} \approx G^{(s)}$. Such matrix factorization is called *interpolative decomposition* (ID). The data points $D_s = \{x_{s_1}, x_{s_2}, \dots, x_{s_{l^{(s)}}}\}$, which are associated with the columns of $B^{(s)}$, constitute the sampled dataset at scale s .

For that purpose we use Algorithm 2, which is a randomized ID algorithm [12]. It produces an ID for a general given matrix $m \times n$ matrix A and an integer $l < \min\{m, n\}$, s.t.

$$\|A - BP\|_2 \lesssim l\sqrt{mn}\sigma_{l+1}(A) \quad (4.4)$$

that costs $\mathcal{O}(l^2n \log(n))$ floating-point operations. Algorithm 2 uses Algorithm 1, which is a deterministic ID algorithm that costs $\mathcal{O}(mn^2)$ operations for an $m \times n$ matrix [4].

Algorithm 1: Deterministic interpolative decomposition

Input: An $m \times n$ matrix A and an integer k , s.t. $k < \min \{m, n\}$.

Output: An $m \times k$ matrix B , whose columns constitute a subset of the columns of A , and an $k \times n$ matrix P s.t. $\|A - BP\|_2 \leq \sqrt{4k(n-k) + 1}\sigma_{k+1}(A)$

1: Apply the pivoted QR routine to W (Algorithm 5.4.1 in [9]),

$$AP_R = QR,$$

where P_R is an $n \times n$ permutation matrix, Q is an $m \times m$ orthogonal matrix and R is an $m \times n$ upper triangular matrix, where the absolute values on the diagonal are decreasingly ordered.

2: Split R and Q s.t.

$$R = \left[\begin{array}{c|c} R_{11} & R_{12} \\ \hline 0 & R_{22} \end{array} \right],$$

$$Q = \left[\begin{array}{c|c} Q_1 & Q_2 \end{array} \right],$$

where R_{11} is $k \times k$, R_{12} is $k \times (n - k)$, R_{22} is $(m - k) \times (n - k)$, Q_1 is $m \times k$ and Q_2 is $m \times (m - k)$.

3: Define the $m \times k$ matrix

$$B = Q_1 R_{11} \tag{4.5}$$

4: Define the $k \times n$ matrix

$$P = \left[\begin{array}{c|c} I_k & R_{11}^{-1} R_{12} \end{array} \right],$$

where I_k is the $k \times k$ identity matrix.

Algorithm 2: Randomized interpolative decomposition

Input: An $m \times n$ matrix A and two integers $l < k$, s.t. $k < \min\{m, n\}$ (for example, $k = l + 8$).

Output: An $m \times l$ matrix B and an $l \times n$ matrix P that satisfy Eq. 4.4.

- 1: Use a random number generator to form a real $k \times m$ matrix G whose entries are i.i.d Gaussian random variables of zero mean and unit variance. Compute the $k \times n$ product matrix

$$W = GA.$$

- 2: Using Algorithm 1, form a $k \times l$ matrix S , whose columns constitute a subset of the columns of W , and a real $l \times n$ matrix P , such that

$$\|SP - W\|_2 \leq \sqrt{4l(n-l) + 1}\sigma_{l+1}(W).$$

- 3: From Step 2, the columns of S constitute a subset of the columns of W . In other words, there exists a finite sequence i_1, i_2, \dots, i_l of integers such that, for any $j = 1, 2, \dots, l$, the j -th column of S is the i_j -th column of W . The corresponding columns of A are collected into a real $m \times l$ matrix B , so that, for any $j = 1, 2, \dots, l$, the j -th column of B is the i_j -th column of A . Then, the sampled dataset is $D_s = \{x_{i_1}, x_{i_2}, \dots, x_{i_l}\}$.
-

Application of Algorithm 2 to an $n \times n$ Gaussian kernel matrix $G^{(s)}$, whose numerical rank is $l^{(s)}$, results in a well conditioned $n \times l^{(s)}$ matrix $B^{(s)}$, whose columns constitute a subset of $G^{(s)}$ columns in the associated sampled set D_s .

Example 4.1. Figure 4.1 demonstrates the data sampling in six different scales. The data consists of 1469 data points, which are scattered on the square $[0, 2\pi] \times [0, 2\pi]$. It is represented by the white dots in Fig.4.1(f). For each scale s ($s = 0, 1, \dots, 5$) we formed a Gaussian kernel $G^{(s)}$ (see Eq. 4.3), where $\epsilon_s = 4^{-s}$. By the application of Algorithm 2 to $G^{(s)}$, we produced a sequence of sampled datasets $D_s = \{x_{s_1}, x_{s_2}, \dots, x_{s_{l^{(s)}}}\}$, $s = 0, 1, \dots, 5$, which are represented by the white dots in Figs. 4.1((a)-(f)). The associated Gaussians constitute a maximal set of (numerically) linearly independent Gaussians on the data at scale s .

4.2 Phase 2: Single-scale function extension

Once Algorithm 2 was applied to $G^{(s)}$, the columns of $B^{(s)}$ constitute a well-conditioned basis for the columns of $G^{(s)}$. Algorithm 3 describes a procedure to extend the orthogonal projection of $f = [f_1 \ f_2 \ \dots \ f_n]^T$ on $B^{(s)}$ to a new data point $x_* \in \mathbb{R}^d \setminus D$. For this we need

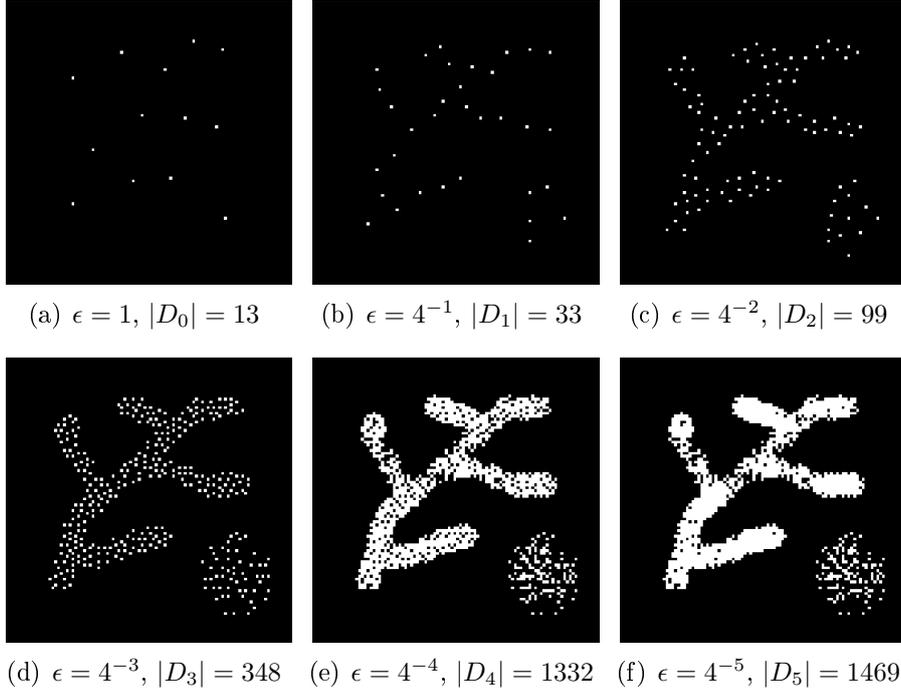


Figure 4.1: Sampled data points at different scales where each $|D_i|, i = 1, \dots, 5$ represents the number of sampled data points in each scale.

the following notation:

$$G_*^{(s)} \triangleq \begin{bmatrix} g_{\epsilon_s}(x_*, x_{s_1}) & g_{\epsilon_s}(x_*, x_{s_2}) & \cdots & g_{\epsilon_s}(x_*, x_{s_{l(s)}}) \end{bmatrix}, \quad (4.6)$$

where g_{ϵ_s} is given in Eq. 1.10 with the standard Euclidean norm in \mathbb{R}^d .

Algorithm 3: Single-scale extension

Input: An $n \times l^{(s)}$ matrix $B^{(s)}$, the associated sampled data $D_s = \{x_{s_1}, x_{s_2}, \dots, x_{s_{l^{(s)}}}\}$, a new data point x , a function $f = [f_1 f_2 \dots f_n]^T$ to be extended and a length parameter ϵ_s .

Output: The projection $f^{(s)} = [f_1^{(s)} f_2^{(s)} \dots f_n^{(s)}]^T$ of f on $B^{(s)}$ and its extension $f_*^{(s)}$ to x .

- 1: Apply SVD to $B^{(s)}$, s.t. $B^{(s)} = U\Sigma V^*$.
- 2: Calculate the pseudo-inverse $(B^{(s)})^\dagger = V\Sigma^{-1}U^*$ of $B^{(s)}$.
- 3: Calculate the coordinates vector of the orthogonal projection of $f^{(s)}$ on the range of $B^{(s)}$ in the basis of the $B^{(s)}$'s columns $c = (B^{(s)})^\dagger f$.
- 4: Calculate the orthogonal projection of f on the columns of $B^{(s)}$, $f^{(s)} = B^{(s)}c$.
- 5: Form the matrix $G_*^{(s)}$ from Eq. 4.6.
- 6: Calculate the extension $f_*^{(s)}$ of $f^{(s)}$ to x_* :

$$f_*^{(s)} \triangleq G_*^{(s)}c. \quad (4.7)$$

Remark 4.2.

1. Due to step 1, the complexity of Algorithm 3 is in $\mathcal{O}(n(l^{(s)})^2)$.
2. Due to Eq. 4.7, $f^{(s)}$ is a linear combination of $l^{(s)}$ Gaussians with a fixed length parameter ϵ_s . Hence, $f^{(s)} \in C^\infty(\mathbb{R}^d)$. Moreover, $f_*^{(s)} \rightarrow 0$ as $\text{dist}(x_*, D_s) \rightarrow \infty$ (see Definition 3.10).

4.3 Multiscale data sampling and function extension

Algorithm 4 is a multiscale scheme that extends f to x_* while producing a sequence of sampled datasets. It is based on Algorithms 2 and 3. For $s = 0$, Algorithm 3 is applied to f . If $\|f - f^{(0)}\|$ is not sufficiently small (the criterion for this and the norm $\|\cdot\|$ are determined by the user) then Algorithm 3 is applied again to the difference $f - f^{(0)}$ with $s = 1$, and so on. Thus, we use the data points in D as test set for our extension.

Algorithm 4: Multiscale data sampling and function extension

Input: A dataset $D = \{x_1, \dots, x_n\}$ in \mathbb{R}^d , a positive number $T > 0$, a new data point $x_* \in \mathbb{R}^d \setminus D$, a function $f = [f_1 f_2 \dots f_n]^T$ to be extended and an error parameter $err \geq 0$.

Output: An approximation $F = [F_1 F_2 \dots F_n]^T$ of f on D and its extension F_* to x_* .

- 1: Set the scale parameter $s = 0$, $F^{(-1)} = 0 \in \mathbb{R}^n$ and $F_*^{(-1)} = 0$.
 - 2: **while** $\|f - F^{(s-1)}\| > err$ **do**
 - 3: Form the Gaussian kernel $G^{(s)}$ on D (see Eqs. 1.11 and 4.3), with $\epsilon_s = \frac{T}{2^s}$.
 - 4: Estimate the numerical rank $l^{(s)}$ of $G^{(s)}$ using Eq. 3.20.
 - 5: Apply Algorithm 2 to $G^{(s)}$ with the parameters $l^{(s)}$ and $l^{(s)} + 8$ to get an $n \times l^{(s)}$ matrix $B^{(s)}$ and the sampled dataset D_s .
 - 6: Apply Algorithm 3 to $B^{(s)}$ and f . We get the approximation $f^{(s)}$ to $f - F^{(s-1)}$ at scale s , and its extension $f_*^{(s)}$ to x_* .
 - 7: Set $F^{(s)} = F^{(s-1)} + f^{(s)}$, $f_*^{(s)} = F_*^{(s-1)} + f_*^{(s)}$, $s = s + 1$.
 - 8: **end while**
 - 9: $F = F^{(s-1)}$ and $F_* = F_*^{(s-1)}$.
-

Remark 4.3. Choice of the algorithm's parameters:

1. T is the length parameter of the Gaussian kernel matrix at the first scale of the algorithm. Therefore, in order to capture x_* , we set T to be $T = \max \{dist(x_*, D), \kappa(D)\}$, where $\kappa(D) = 2 \left(\frac{diameter(D)}{2} \right)^2$. $diameter(D)$ is the distance between the most distant pair in D . This choice of T ensures that in the first scale the influence of D on x_* is significant and that D is covered by a single Gaussian.
2. err is a user defined accuracy parameter. If we take $err = 0$ than $F = f$, i.e. we have a multiscale interpolation scheme. A too big err may result in inaccurate approximation of f .
3. Estimation of $l^{(s)}$, which is the numerical rank of $G^{(s)}$, is required in step 4. This is done by Eq. 3.20, where the parameter δ is involved. Typically, we take $\delta = 0.1$, which guarantees that $B^{(s)}$ is well conditioned and, as a consequence, it guarantees the robustness of Algorithm 4.
4. $F \in C^\infty(\mathbb{R}^d)$ as a finite sum of $C^\infty(\mathbb{R}^d)$ functions.

5 Experimental results

Examples 5.1-5.3 below demonstrate the application of the MSE procedure. In the first example, we apply the MSE to a univariate function. The results are compared with the results from the application of the Nyström extension. The second example demonstrates an extension of a function from the unit circle to \mathbb{R}^2 . The third example shows an interpolation of a linear function.

Example 5.1. We demonstrate the advantages of the MSE procedure by the following example: We applied the MSE procedure to 50 random samples of the function

$$h(x) = \begin{cases} \sin(x) & 0 \leq x \leq \pi \\ \sin(5x) & \pi \leq x \leq 2\pi \end{cases}.$$

Figures 5.1(a)-5.1(f) show the evolution of the MSE outputs (with $err = 0$ and $T = 2\pi^2$) through six scales. The given data point (the samples of $h(x)$) are denoted by +, the sampled dataset at each scale, namely D_s , are denoted by circles and the extension is denoted by a continuous line. The last three shown evolution levels (Figs. 5.1(d)-5.1(f)) were compared to the corresponding Nyström extensions with the same length parameters $\epsilon = \frac{\pi^2}{27}, \frac{\pi^2}{29}, \frac{\pi^2}{210}$ (Figs. 5.1(g)-5.1(i)). Extension by Nyström with too small length parameter resulted in oscillatory extension with artifacts as shown in Figs. 5.1(h)-5.1(i). On the other hand, the Gaussian kernel matrix is ill-conditioned for large length parameters, as was proved in Section 3.3. The condition numbers of the Gaussian kernel matrices, which correspond to the first three shown evolution levels (Figs. 5.1(a)-5.1(c)), are above 10^{18} . This is the reason why we cannot compare between our and the Nyström extension results for these length parameters. Nyström extension with $\epsilon = \frac{\pi^2}{27}$ (Fig. 5.1(g)) is both well-conditioned and free of artifacts, but still, there is no way to know the proper length parameter a priori. Detailed results are summarized in Table 5.1.1.

Example 5.2. Figure 5.2 demonstrates the multiscale function extension of $f(\theta) = \sin(\theta) + \sin(5\theta) + \sin(10\theta)$ from 1000 equally-spaced data points on the unit circle to the square $[-3, 3] \times [-3, 3]$. The sampled dataset in each scale is shown on the left column. It is represented by the red asterisks, where the full dataset is represented by the black dots. The multiscale extension of f is shown on the right column. It is represented by the surface. f is drawn by the black line.

Example 5.3. Figure 5.3 demonstrates an embedding of linear data: the scattered data points represent the original data (20 data points, which are distributed uniformly in $[0, 1]$). The filled points are the sampled data points in each scale. The embedded data

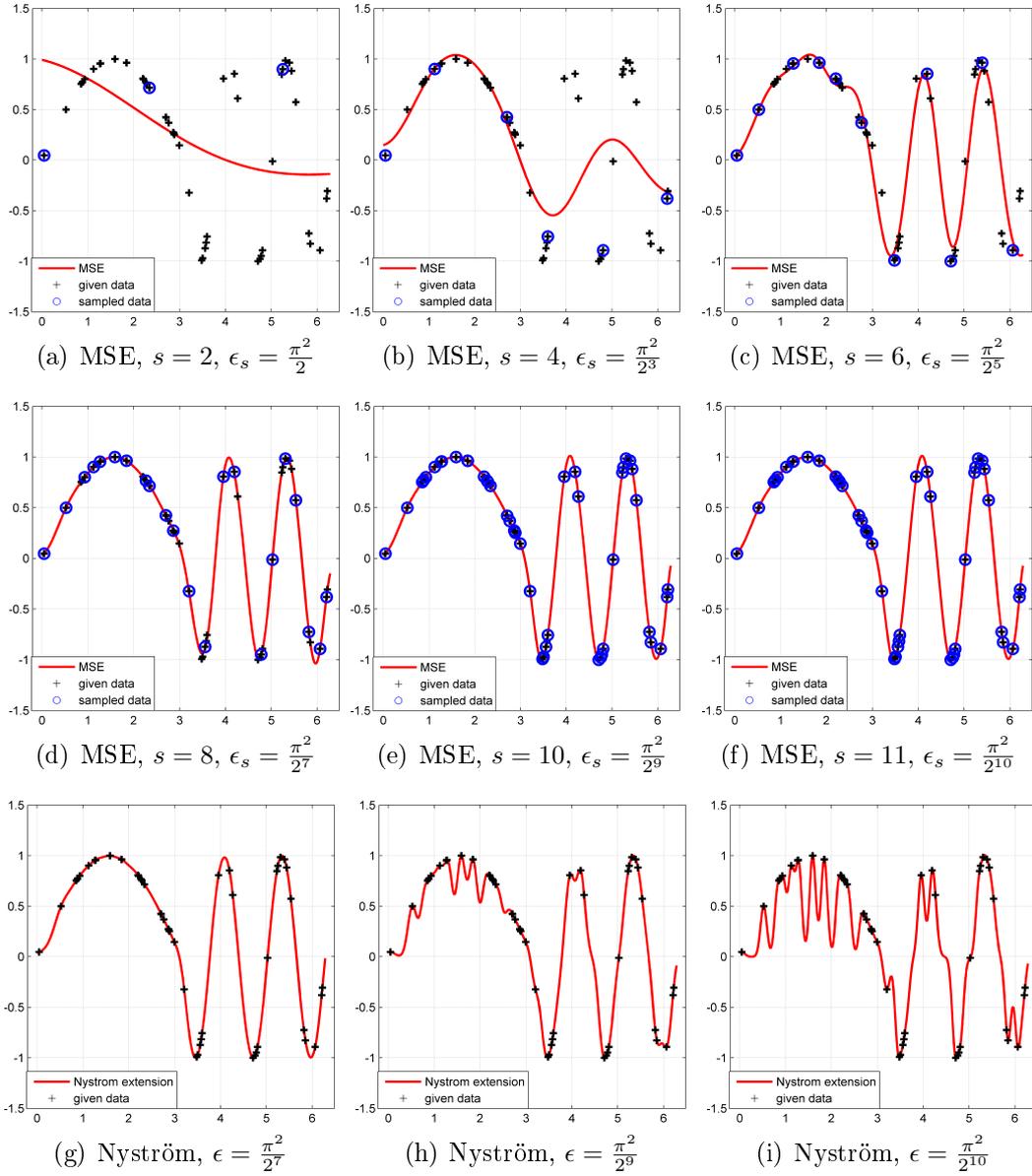
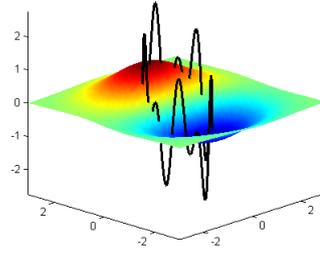
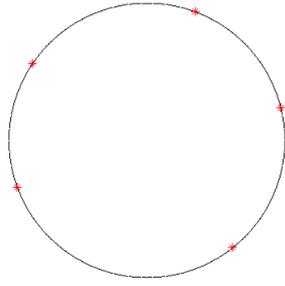


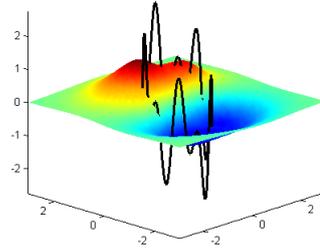
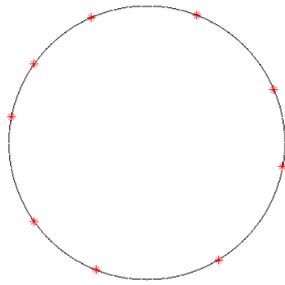
Figure 5.1: Comparison between the outputs from the MSE procedure and the Nyström extension: Figures (a)-(f) show the evolution of the MSE procedure through several scales. Figures (g)-(i) show the Nyström extension applied to the same samples of $h(x)$ with the same length parameters ϵ as in scales 8, 10 and 11 (Figs. (d)-(f)). The Nyström extension, which corresponds to scales 2, 4 and 6 (Figs. (a)-(c)), is ill-conditioned and cannot be implemented to the given data.

MSE					Nyström	
scale s	ϵ_s	$l^{(s)}$	CN of $B^{(s)}$	$error$	ϵ	CN of $G^{(s)}$
0	2π	2	$.36E + 01$	$.27E + 02$	2π	$.12E + 20$
1	π	2	$.29E + 01$	$.27E + 02$	π	$.27E + 19$
2	$\pi \times 2^{-1}$	3	$.35E + 01$	$.27E + 02$	$\pi \times 2^{-1}$	$.15E + 19$
3	$\pi \times 2^{-2}$	4	$.39E + 01$	$.27E + 02$	$\pi \times 2^{-2}$	$.49E + 19$
4	$\pi \times 2^{-3}$	6	$.56E + 01$	$.23E + 02$	$\pi \times 2^{-3}$	$.90E + 19$
5	$\pi \times 2^{-4}$	8	$.85E + 01$	$.22E + 02$	$\pi \times 2^{-4}$	$.44E + 19$
6	$\pi \times 2^{-5}$	11	$.78E + 01$	$.92E + 01$	$\pi \times 2^{-5}$	$.21E + 19$
7	$\pi \times 2^{-6}$	16	$.12E + 02$	$.21E + 01$	$\pi \times 2^{-6}$	$.29E + 18$
8	$\pi \times 2^{-7}$	22	$.24E + 02$	$.59E + 00$	$\pi \times 2^{-7}$	$.10E + 13$
9	$\pi \times 2^{-8}$	31	$.19E + 03$	$.83E - 01$	$\pi \times 2^{-8}$	$.35E + 09$
10	$\pi \times 2^{-9}$	44	$.17E + 05$	$.44E - 03$	$\pi \times 2^{-9}$	$.60E + 07$
11	$\pi \times 2^{-10}$	50	$.59E + 06$	0	$\pi \times 2^{-10}$	$.59E + 06$

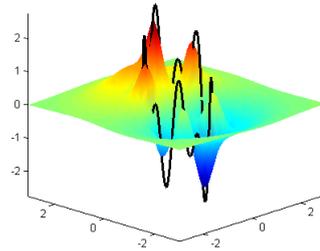
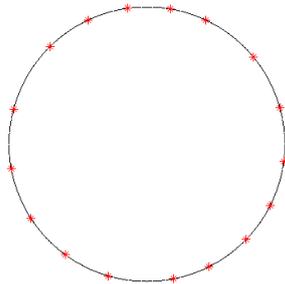
Table 5.1: Summary of Example 5.1. The table summarizes the results of application MSE to 50 random samples of $h(x)$, compared with Nyström extension, through 12 scales. The second column consists the length scales corresponding to each scale. $l^{(s)}$ is the approximation of the numerical rank of the corresponding Gaussian kernel matrix (see Step 4 in Algorithm 4), which is the size of sampled dataset in each scale. The fourth and the seventh columns consist the condition numbers (CNs) of the matrices $B^{(s)}$ and $G^{(s)}$ in MSE and Nyström extension, respectively. The $error$ in the fifth column was measured by the l_2 norm of $F - f^{(s)}$ from Algorithm 4. Of course, the error of the Nyström extension, when it applicable, is zero on the data, as it is an interpolation scheme.



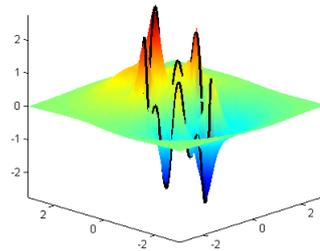
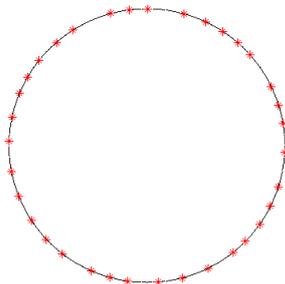
(a) $\epsilon = 2$, 5 sample points



(b) $\epsilon = 0.5$, 9 sample points



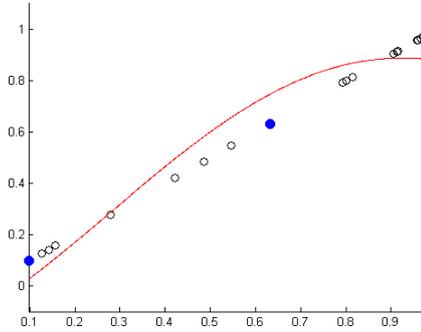
(c) $\epsilon = 0.125$, 17 sample points



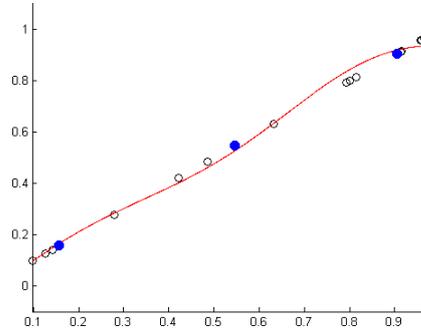
(d) $\epsilon = 0.0313$, 35 sample points

Figure 5.2: Multiscale extension of $f(\theta) = \sin(\theta) + \sin(5\theta) + \sin(10\theta)$ from 1000 equally-spaced data points on the unit circle (left) to the square $[-3, 3] \times [-3, 3]$ (right)

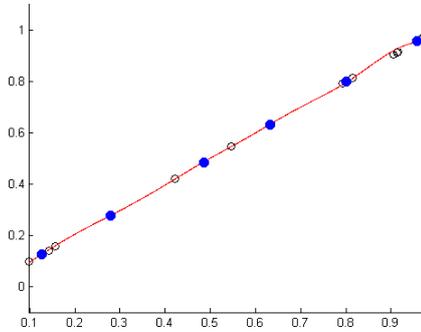
points are represented by red dots (1000 equally-spaced data points in $[0, 1]$). It is shown that if the dataset lies in \mathbb{R} , then the multiscale extension scheme embeds any newly-arrived data point to its exact location.



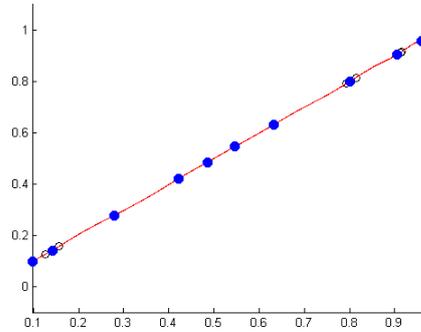
(a) $\epsilon = 0.873$, 2 sample points



(b) $\epsilon = 0.109$, 3 sample points



(c) $\epsilon = 0.013$, 6 sample points



(d) $\epsilon = 0.001$, 10 sample points

Figure 5.3: Linear embedding

Conclusions

We introduce a numerically-stable multiscale scheme to perform efficiently out-of-sample extension for function on high-dimensional space. The scheme overcomes the limitations of the Nyström extension. It is based on mutual distances between data points while utilizing IDs of a sequence of Gaussian kernel matrices. The method requires no grid. It automatically generates a sequence of adaptive grids according to the data points distribution.

The following topics and more extensions will be investigated by us next: 1. Can the extension smoothness be related to the smoothness of f ? 2. Do we get better energy concentration if the Gaussians are replaced by prolates? 3. How the L_1 minimization is

compared with the presented scheme? 4. How the results regarding the numerical rank of the Gaussian kernel matrix can be generalized to manifolds? In other words, how to perform multiscale data sampling and function extension that is based on the intrinsic dimension of the data that has a lower dimension than its ambient space.

Acknowledgments

This research was partially supported by the Israel Science Foundation (Grant No. 1041/10). Ronald R. Coifman and Amit Bermanis were partially supported by DOE grant DE-SC0002097

We would like to thank Yoel Shkolnisky for helpful discussions.

References

- [1] C.T.H. Baker. The numerical treatment of integral equations. Oxford: Clarendon Press, 1977.
- [2] A. Bermanis, A. Averbuch, and R.R. Coifman. Multiscale data sampling and function extension. *SampTA 2011 proceedings*, 2011.
- [3] R. Bhatia. Perturbation bounds for matrix eigenvalues. SIAM, 2007.
- [4] H. Cheng, Z. Gimbutas, P.-G. Martinsson, and V. Rokhlin. On the compression of low rank matrices. *SIAM J. Sci. Comput.*, 26:1389–1404, 2005.
- [5] R.R. Coifman and S. Lafon. Diffusion maps. *Appl. Comput. Harmon. Anal.*, 21:5–30, 2006.
- [6] R.R. Coifman and S. Lafon. Geometric harmonics: A novel tool for multiscale out-of-sample extension of empirical functions. *Appl. Comput. Harmon. Anal.*, 21:31–52, 2006.
- [7] P. J. Davis. Circulant matrices. A Wiley-Interscience Publication, 1979.
- [8] M.S. Floater and A. Iske. Multistep scattered data interpolation using compactly supported radial basis functions. *J. Comp. Appl. Math.*, 73:65–78, 1996.
- [9] G. H. Golub and C. F. Van Loan. Matrix computations. The Johns Hopkins University Press, 3rd edition, 1996.

- [10] R.A. Horn and C.R. Johnson. Topics in matrix analysis. Cambridge University Press, 3rd edition, 1991.
- [11] C.G. Journel and CH.J. Huijbregts. Mining geostatistics. Academic Press Inc, 1978.
- [12] P.G. Martinsson, V. Rokhlin, and M. Tygert. A randomized algorithm for the decomposition of matrices. *Appl. Comput. Harmon. Anal.*, 2010.
- [13] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. Numerical recipes in c. Cambridge Univ. Press, 2nd edition, 1992.
- [14] C. Rasmussen and C. Williams. Gaussian processes for machine learning. MIT Press, 2006.
- [15] H. Wackernagel. Multivariate geostatistics. Springer-Verlag, 2003.
- [16] J.S. Walker. Fourier analysis. Oxford University Press, 1988.
- [17] H. Wendland. Scattered data approximation. Cambridge University Press, 2005.