Quasi-Monte Carlo Feature Maps for Shift-Invariant Kernels

Jiyan Yang

Stanford University

June 24th, 2014

ICML, 2014, Beijing

Joint work with Vikas Sindhwani, Haim Avron and Michael Mahoney

Brief Overview of Kernel Methods

Low-dimensional Explicit Feature Map

Quasi-Monte Carlo Random Feature

Empirical Results

Problem setting

We will start with the kernelized ridge regression problem,

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2 + \lambda \|f\|_{\mathcal{H}}^2. \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^d$, $\mathcal{H}$ is a nice hypothesis space (RKHS) and ℓ is a convex loss function.

- A symmetric and positive-definite kernel $k(\mathbf{x}, \mathbf{y})$ generates a unique RKHS $\mathcal{H}$.
- For example, RBF kernel, $k(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}$.
- Kernel methods are widely used in solving regression, classification or inverse problems raised in many areas as well as unsupervised learning problems.

Scalability

- By the Representer Theorem, the minimizer of (1) can be represented by
$$c = (K + \lambda n I)^{-1} Y.$$
- Above the Gram matrix K is defined as $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Forming $n \times n$ matrix K needs $\mathcal{O}(n^2)$ storage and typical linear algebra needs $\mathcal{O}(n^3)$ running time.
- This is an $n \times n$ dense linear system which is not scalable for large n.

Linear kernel and explicit feature maps

- Suppose we can find a feature map $\Psi : \mathcal{X} \to \mathbb{R}^s$ such that $k(\mathbf{x}, \mathbf{y}) = \mathbf{z}(\mathbf{x})^T \mathbf{z}(\mathbf{y})$. Then the Gram matrix $K = ZZ^T$, where the i-th row of Z is $\mathbf{z}(\mathbf{x}_i)$ and $Z \in \mathbb{R}^{n \times s}$.
- The solution to (1) can be expressed as

$$w = (Z^T Z + \lambda n I)^{-1} Z^T Y.$$

- This is an $s \times s$ linear system.
- It is attractive if $s < n$.
- Testing times reduces from $\mathcal{O}(nd)$ to $\mathcal{O}(s + d)$.

Brief Overview of Kernel Methods

Low-dimensional Explicit Feature Map

Quasi-Monte Carlo Random Feature

Empirical Results

Mercer's Theorem and explicit feature map

Theorem (Mercer)

For any positive definite kernel k, it can be expanded into

$$k(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N_F} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{y}).$$

- Can define $\Phi(\mathbf{x}) = (\sqrt{\lambda_1}\phi_1(\mathbf{x}), \ldots, \sqrt{\lambda_{N_F}}\phi_{N_F}(\mathbf{x}))$.
- For many kernels, such as RBF, $N_F = \infty$.
- **Goal**: Find explicit feature map $\mathbf{z}(\mathbf{x}) \in \mathbb{R}^s$ such that

$$k(\mathbf{x}, \mathbf{y}) \simeq \mathbf{z}(\mathbf{x})^T \mathbf{z}(\mathbf{y}),$$

where $s < n$. Then $K \simeq ZZ^T$.

Bochner's Theorem

Theorem (Bochner)

A continuous kernel $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y})$ on $\mathbb{R}^d$ is positive definite if and only if $k(\mathbf{x} - \mathbf{y})$ is the Fourier transform of a non-negative measure.

A Monte Carlo Approximation

- More specifically, given a shift-invariant kernel k, we have

$$k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y}) = \int_{\mathbb{R}^d} e^{-i\mathbf{w}^T(\mathbf{x}-\mathbf{y})} p(\mathbf{w}) \mathrm{d}\mathbf{w}.$$

- By standard Monte Carlo (MC) approach, the above can be approximated by

$$\tilde{k}(\mathbf{x}, \mathbf{y}) = \frac{1}{s} \sum_{j=1}^{s} e^{-i\mathbf{w}_j^T(\mathbf{x}-\mathbf{y})}, \tag{2}$$

where $\mathbf{w}_j$ are drawn from $p(\mathbf{w})$.

Random Fourier feature

- The random Fourier feature map can be defined as

$$\psi(\mathbf{x}) = \frac{1}{\sqrt{s}}(g_{\mathbf{w}_1}(\mathbf{x}), \ldots, g_{\mathbf{w}_s}(\mathbf{x})),$$

  where $g_{\mathbf{w}_j}(\mathbf{x}) = e^{-i\mathbf{w}_j^T\mathbf{x}}$. [Rahimi and Recht 07].

- So

$$\tilde{k}(\mathbf{x}, \mathbf{y}) = \frac{1}{s}\sum_{j=1}^{s} e^{-i\mathbf{w}_j^T(\mathbf{x}-\mathbf{y})} = \psi(\mathbf{x})^T\bar{\psi}(\mathbf{y}).$$

Motivation

- We want to use less random features while maintaining the same approximation accuracy.
- MC method has a convergence rate of $\mathcal{O}(1/\sqrt{s})$.
- To gain a faster convergence, quasi-Monte Carlo method will be a better choice since it has a convergence rate of $\mathcal{O}((\log s)^d/s)$.

Quasi-Monte Carlo method

Goal

To approximate an integral over the d-dimensional unit cube $[0,1]^d$,

$$I_d(f) = \int_{[0,1]^d} f(x)\mathrm{d}x_1 \cdots \mathrm{d}x_d.$$

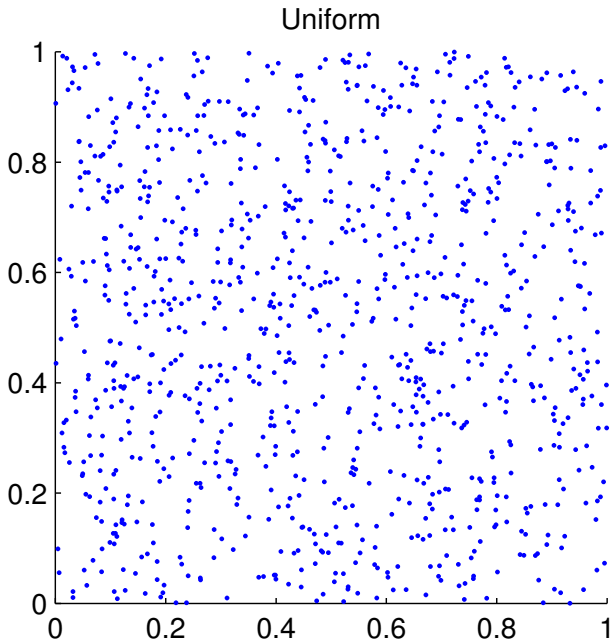
Quasi-Monte Carlo methods usually take the following form,

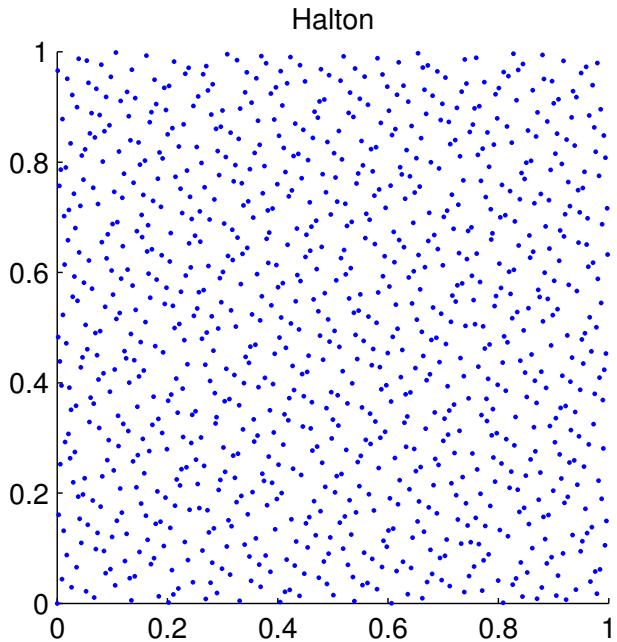
$$Q_s(f) = \frac{1}{s}\sum_{i=1}^{s} f(t_i),$$

where $t_1, \ldots, t_s \in [0,1]^d$ are pseudo-random points chosen deterministically with low-discrepancy.

Low-discrepancy sequences

- Many pseudo-random sequences $\{\mathbf{t}_i\}_{i=1}^{\infty}$ with low-discrepancy are available, such as Halton sequence and Sobol' sequence.
- They tend to be more "uniform" than sequence drawn uniformly.
- Notice the clumping and the space with no points in the left subplot.





Quasi-random features

- By setting $\mathbf{w} = \Phi^{-1}(\mathbf{t})$, $k(\mathbf{x}, \mathbf{y})$ can be rewritten as

$$\begin{aligned} \int_{\mathbb{R}^d} e^{-i(\mathbf{x}-\mathbf{y})^T \mathbf{w}} p(\mathbf{w}) d\mathbf{w} &= \int_{[0,1]^d} e^{-i(\mathbf{x}-\mathbf{y})^T \Phi^{-1}(\mathbf{t})} d\mathbf{t} \\ &\approx \frac{1}{s} \sum_{j=1}^{s} e^{-i(\mathbf{x}-\mathbf{y})^T \Phi^{-1}(\mathbf{t}_j)} . \quad (3) \end{aligned}$$

- After generating the low discrepancy sequence $\{\mathbf{t}_j\}_{j=1}^s$, the quasi-random features can be represented by $\frac{1}{s} \sum_{j=1}^s g_{\mathbf{t}_i}(\mathbf{x})$, where

$$g_{\mathbf{t}_j}(\mathbf{x}) = e^{-i\mathbf{x}^T \Phi^{-1}(\mathbf{t}_j)}.$$

Algorithm: Quasi-Random Fourier Features

Input: Shift-invariant kernel k, size s.
Output: Feature map $\hat{\Psi}(\mathbf{x}) : \mathbb{R}^d \mapsto \mathbb{C}^s$.

1: Find p, the inverse Fourier transform of k.
2: Generate a low discrepancy sequence $\mathbf{t}_1, \ldots, \mathbf{t}_s$.
3: Transform the sequence: $\mathbf{w}_j = \Phi^{-1}(\mathbf{t}_j)$.
4: Set $\hat{\Psi}(\mathbf{x}) = \sqrt{\frac{1}{s}} \left[e^{-i\mathbf{x}^T\mathbf{w}_1}, \ldots, e^{-i\mathbf{x}^T\mathbf{w}_s} \right]$.

Quality of Approximation

- Given a pair of points $\mathbf{x}, \mathbf{y}$, let $\mathbf{u} = \mathbf{x} - \mathbf{y}$. The approximation error is

$$\epsilon[f_{\mathbf{u}}] = \int_{\mathbb{R}^d} f_{\mathbf{u}}(\mathbf{w}) p(\mathbf{w}) d\mathbf{w} - \frac{1}{s} \sum_{i=1}^{s} f_{\mathbf{u}}(\mathbf{w}_i),$$

  where $f_{\mathbf{u}}(\mathbf{w}) = e^{i\mathbf{u}^T\mathbf{w}}$.
- Want to characterize the behavior of $\epsilon[f_{\mathbf{u}}]$ when $u \in \bar{\mathcal{X}}$ and $\bar{\mathcal{X}} = \{\mathbf{x} - \mathbf{z} | \mathbf{x}, \mathbf{z} \in \mathcal{X}\}$.
- Consider a broader class of integrands,

$$\mathcal{F}_{\square\mathbf{b}} = \{f_{\mathbf{u}} | \mathbf{u} \in \square\mathbf{b}\}.$$

  Here $\square\mathbf{b} = \{\mathbf{u} \in \mathbb{R}^d \mid |u_j| \leq b_j\}$ and $\bar{\mathcal{X}} \in \square\mathbf{b}$.

Main Theoretical Result

Theorem (Average Case Error)

Let $\mathcal{U}(\mathcal{F}_{\square\mathbf{b}})$ denote the uniform distribution on $\mathcal{F}_{\square\mathbf{b}}$. That is, $f \sim \mathcal{U}(\mathcal{F}_{\square\mathbf{b}})$ denotes $f = f_{\mathbf{u}}$ where $f_{\mathbf{u}}(\mathbf{x}) = e^{-i\mathbf{u}^T\mathbf{x}}$ and $\mathbf{u}$ is randomly drawn from a uniform distribution on $\square\mathbf{b}$. We have,

$$\mathbb{E}_{f\sim\mathcal{U}(\mathcal{F}_{\square\mathbf{b}})}\left[\epsilon_{S,p}[f]^2\right] = \frac{\pi^d}{\prod_{j=1}^d b_j} D_p^{\square\mathbf{b}}(S)^2 \ .$$

Box discrepancy

Suppose that $p(\cdot)$ is a probability density function, and that we can write $p(\mathbf{x}) = \prod_{j=1}^{d} p_j(x_j)$ where each $p_j(\cdot)$ is a univariate probability density function as well. Let $\phi_j(\cdot)$ be the characteristic function associated with $p_j(\cdot)$. Then,

$$
\begin{aligned}
D_{\mathrm{sinc}_{\mathbf{b}},p}(S)^2 \;=\; & (\pi)^{-d} \prod_{j=1}^{d} \int_{-b_j}^{b_j} |\phi_j(\beta)|^2 d\beta - \\
& \frac{2(2\pi)^{-d}}{s} \sum_{l=1}^{s} \prod_{j=1}^{d} \int_{-b_j}^{b_j} \phi_j(\beta) e^{i w_{lj} \beta} d\beta + \\
& \frac{1}{s^2} \sum_{l=1}^{s} \sum_{j=1}^{s} \mathrm{sinc}_{\mathbf{b}}(\mathbf{w}_l, \mathbf{w}_j) \; . \qquad (4)
\end{aligned}
$$

Proof techniques

- Consider integrands to be in some Reproducing Kernel Hilbert Space (RKHS). Uniform bound for approximating error can be derived by standard arguments.
- Here we consider the space of functions that admit an integral representation over $\mathcal{F}_{\square\mathbf{b}}$ of the form,

$$f(\mathbf{x}) = \int_{\mathbf{u}\in\square\mathbf{b}} \hat{f}(\mathbf{u}) e^{-i\mathbf{u}^T\mathbf{x}} d\mathbf{u} \text{ where } \hat{f}(\mathbf{u}) \in L_2(\square\mathbf{b}) \ . \tag{5}$$

  These spaces are called *Paley-Wiener spaces* $PW_{\mathbf{b}}$ and they constitute a RKHS.
- The damped approximations of the integrands in $\mathcal{F}_{\square\mathbf{b}}$ of form $\tilde{f}_{\mathbf{u}}(\mathbf{x}) = e^{-i\mathbf{u}^T\mathbf{x}} \operatorname{sinc}(T\mathbf{x})$ are members of $PW_{\mathbf{b}}$ with $\|\tilde{f}\|_{PW_{\mathbf{b}}} = \frac{1}{\sqrt{T}}$. Hence, we expect $D_p^{\square\mathbf{b}}$ to provide a discrepancy measure for integrating functions in $\mathcal{F}_{\square\mathbf{b}}$.

Brief Overview of Kernel Methods

Low-dimensional Explicit Feature Map

Quasi-Monte Carlo Random Feature

Empirical Results

Approximation error on Gram matrix

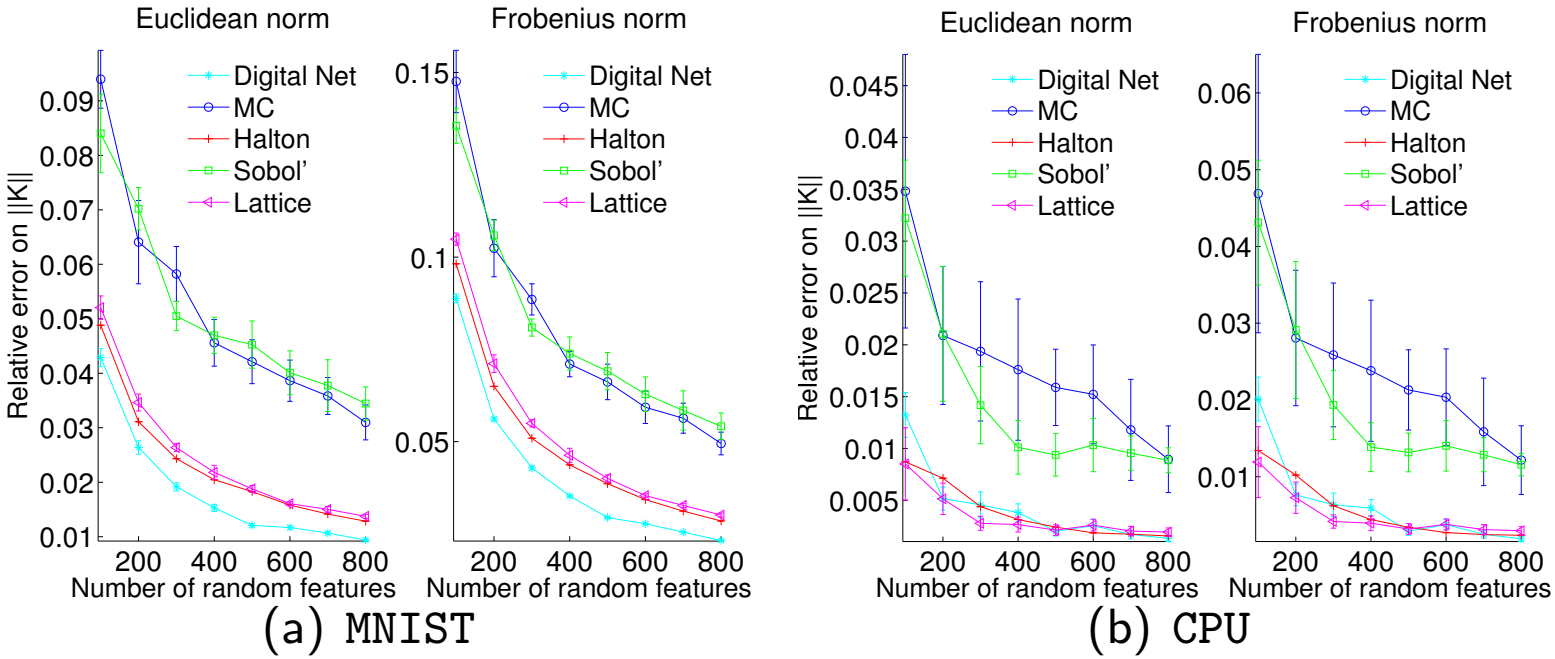


Figure : Relative error on approximating the Gram matrix measured in Euclidean norm and Frobenius norm, i.e. $\|\mathbf{K} - \tilde{\mathbf{K}}\|_2 / \|\mathbf{K}\|_2$ and $\|\mathbf{K} - \tilde{\mathbf{K}}\|_F / \|\mathbf{K}\|_F$, for various s. For each kind of random feature and s, 10 independent trials are executed, and the mean and standard deviation are plotted.

Generalization error

	s	Halton	Sobol'	Lattice	Digit	MC
CPU	100	**0.0367**	0.0383	0.0374	0.0376	0.0383
		(0)	(0.0015)	(0.0010)	(0.0010)	(0.0013)
	500	**0.0339**	0.0344	0.0348	0.0343	0.0349
		(0)	(0.0005)	(0.0007)	(0.0005)	(0.0009)
	1000	**0.0334**	0.0339	0.0337	0.0335	0.0338
		(0)	(0.0007)	(0.0004)	(0.0003)	(0.0005)
CENSUS	400	**0.0529**	0.0747	0.0801	0.0755	0.0791
		(0)	(0.0138)	(0.0206)	(0.0080)	(0.0180)
	1200	**0.0553**	0.0588	0.0694	0.0587	0.0670
		(0)	(0.0080)	(0.0188)	(0.0067)	(0.0078)
	1800	**0.0498**	0.0613	0.0608	0.0583	0.0600
		(0)	(0.0084)	(0.0129)	(0.0100)	(0.0113)

Table : Regression error, i.e. $\|\hat{\mathbf{y}} - \mathbf{y}\|_2 / \|\mathbf{y}\|_2$ where $\hat{\mathbf{y}}$ is the predicted value and $\mathbf{y}$ is the ground truth.

Evaluation of box discrepancy

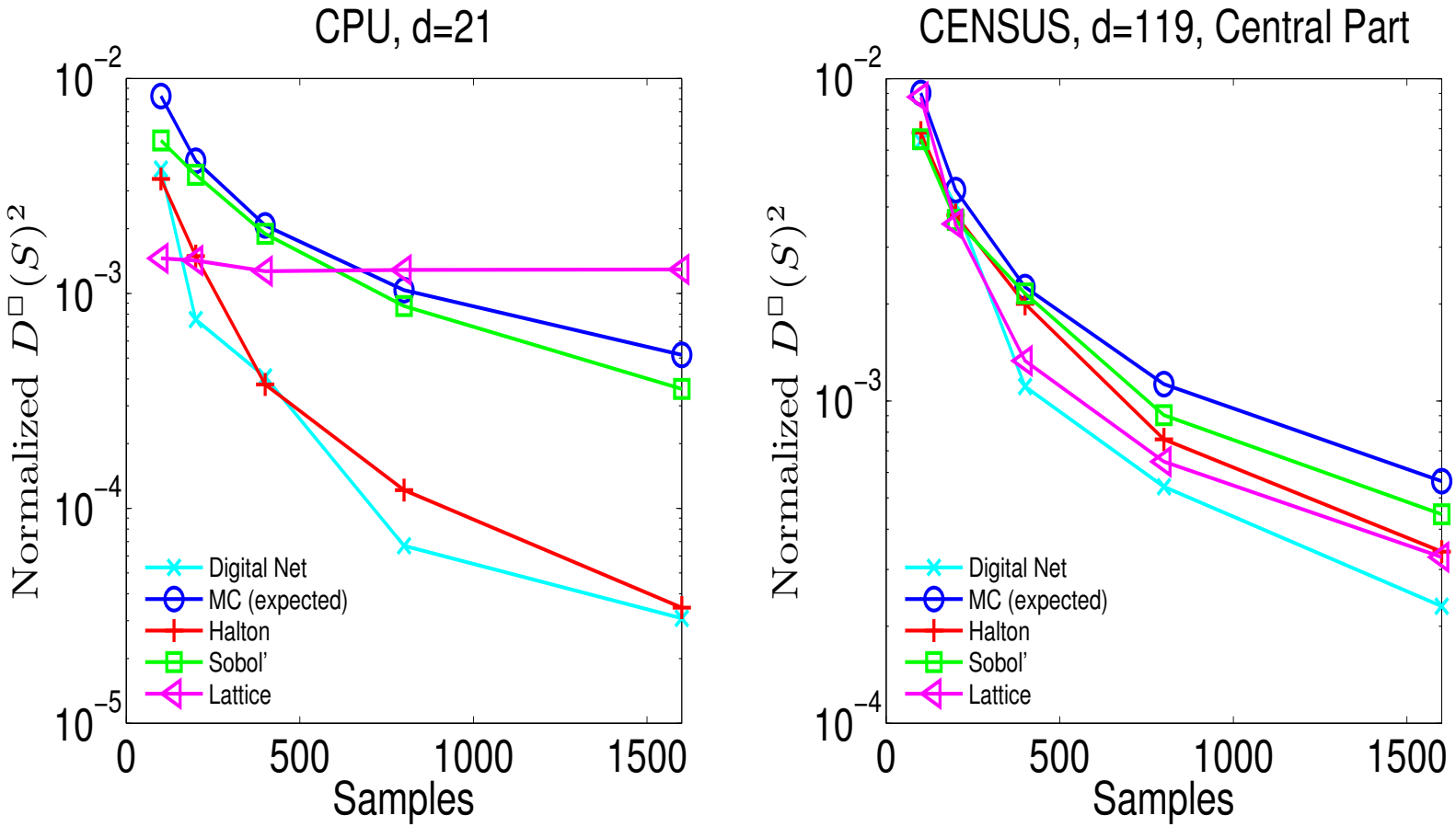


Figure : Discrepancy values ($D^{\square}$) for the different sequences on cpu and census. For census we measure the discrepancy on the central part of the bounding box (we use $\square\mathbf{b}/2$ in the optimization instead of $\square\mathbf{b}$).

Adaptively learning low-discrepancy sequence

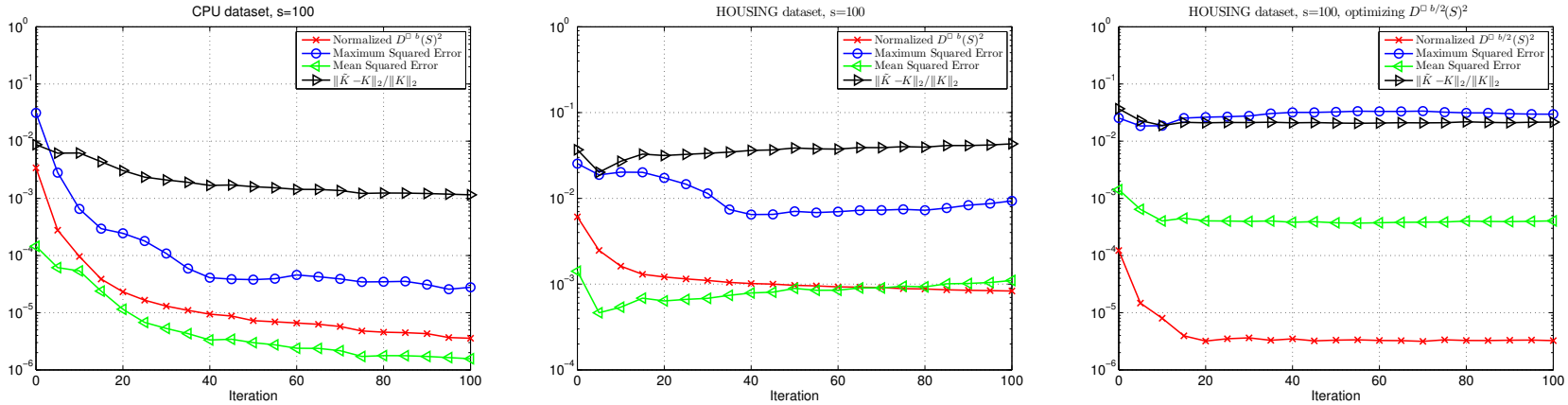


Figure : Examining the behavior of learning *Global Adaptive* sequences. Various metrics on the Gram matrix approximation are plotted.

Generalization error

	s	HALTON	GLOBAL$_{\mathbf{b}}$	GLOBAL$_{\mathbf{b}/4}$	GREEDY$_{\mathbf{b}}$	GREEDY$_{\mathbf{b}/4}$
CPU	100	0.0304	0.0315	**0.0296**	0.0307	**0.0296**
	300	0.0303	0.0278	0.0293	0.0274	**0.0269**
	500	0.0348	0.0347	0.0348	0.0328	**0.0291**
CENSUS	400	**0.0529**	0.1034	0.0997	0.0598	0.0655
	800	0.0545	0.0702	0.0581	0.0522	**0.0501**
	1200	0.0553	0.0639	**0.0481**	0.0525	0.0498
	1800	0.0498	0.0568	**0.0476**	0.0685	0.0548
	2200	0.0519	**0.0487**	0.0515	0.0694	0.0504

Table : Regression error, i.e. $\|\hat{\mathbf{y}} - \mathbf{y}\|_2 / \|\mathbf{y}\|_2$ where $\hat{\mathbf{y}}$ is the predicted value and $\mathbf{y}$ is the ground truth.