Energy-aware adaptive bi-Lipschitz embeddings

Ali Sadeghian
LIONS, EPFL, Switzerland

Bubacarr Bah
LIONS, EPFL, Switzerland

Volkan Cevher
LIONS, EPFL, Switzerland

***Abstract*—We propose a dimensionality reducing matrix design based on training data with constraints on its Frobenius norm and number of rows. Our design criteria is aimed at preserving the distances between the data points in the dimensionality reduced space as much as possible relative to their distances in original data space. This approach can be considered as a deterministic Bi-Lipschitz embedding of the data points. We introduce a scalable learning algorithm, dubbed AMUSE, and provide a rigorous estimation guarantee by leveraging game theoretic tools. We also provide a generalization characterization of our matrix based on our sample data. We use compressive sensing problems as an example application of our problem, where the Frobenius norm design constraint translates into the sensing energy.**

I. Introduction

Embedding of high dimensional data into lower dimensions is almost a classical subject. Random projections is one way of doing such embeddings and this method rely on the famous Johnson-Lindenstrauss (JL) lemma [1]. Recently, JL mappings have also found use in compressed sensing (CS), which is a promising alternative to Nyquist sampling [2]. The current CS theory uses random, non-adaptive matrices and provide recovery guarantees for highly under sampled signals. An key component in the analysis of CS recovery is the restricted isometry property (RIP), [3], [4].

Definition 1 ([3]): A matrix $\mathbf{\Phi}$ satisfies the RIP of order k if the following holds for all vectors $\mathbf{z}$, which has at most k nonzero entries (i.e., k-sparse):

$$(1-\delta_k)\|\mathbf{z}\|_2^2 \leq \|\mathbf{\Phi z}\|_2^2 \leq (1+\delta_k)\|\mathbf{z}\|_2^2. \tag{1}$$

The RIP constant (RIC) of $\mathbf{\Phi}$ of order k is the smallest δ_k for which (1) holds. In the sequel, we use δ without explicit reference to k for the RIC.

In this paper, we consider *adaptivity* in matrix design. Our setting is as follows: we are given a representative data set which can well-approximate an unknown signal. Using this data set, we would like to design a CS matrix that incorporates time and energy constraints while trying to approximate the best RIP matrix. We provide that the embedding we learn is also generalizable to some extent, that is, if a signal is drawn within ϵ of the data set, then the matrix will have good RIC. We formulate the matrix learning problem into a semidefinite program (SDP) and propose an algorithm leveraging tools from game theory.

This work was supported in part by the European Commission under Grant MIRG-268398, ERC Future Proof, SNF 200021-132548.

The main contribution of this work is that, to the best of our knowledge, it is the first deterministic design that is adaptive to data, uses RIP and gives provable approximation guarantees. A salient feature of our approach is that the design has the *digital fountain* property, which makes it nested, that is, if the measurements are not enough, we can still increase the measurements without changing the previous rows of the matrix. In addition, our approach incorporates an important criteria: the *energy constraint*, which may also be important for applications beyond CS. The algorithm we propose is also highly *scalable*, that is, it works in linear space in the matrix size because it only keep the matrix factors. Experimentally, using the matrices we design for CS seems promising as our matrices outperform those of random projections.

Notation: We define the set of k-sparse vectors as $\Sigma_k := \{\mathbf{z} \in \mathbb{R}^n : \|\mathbf{z}\|_0 \leq k\}$; and the set $\Xi_r := \left\{\mathbf{X} \in \mathbb{S}_+^{n\times n} : \text{rank}(\mathbf{X}) \leq r \text{ and } \|\mathbf{X}\|_{\text{tr}} \leq \lambda\right\}$ for scalars $r > 0$ and $\lambda > 0$, where $\mathbb{S}_+^{n\times n}$ is the set of positive semidefinite (PSD) matrices. We denote the n-dimensional simplex by Δ^n.

Definition 2 ([5]): Given $\mathbf{x}_l \in \mathcal{X} \subset \Sigma_k$ we define the set of normalized secants vectors of $\mathcal{X}$ as:

$$\mathcal{S}(\mathcal{X}) := \left\{\mathbf{v}_{ij} = \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|_2} \text{ for } i \neq j\right\}. \tag{2}$$

Ouline: Section II is problem statement with a bit of background; while Section III formulates the problem and presents the algorithm. We analyse the algorithm and give generalization bounds in Section IV, followed by empirical results from simulations and conclusions in Sections V and VI respectively.

II. Background and problem description

The CS literature heavily relies on random matrices in establishing recovery guarantees. There has also been also progress in obtaining structured matrices via randomization. However, for CS to live up to its promise, real applications must be able to use data adaptive matrices. Attempts have been made in this direction that include what is referred to as optimizing projection matrices which entails reducing the correlation between normalised data points (dictionary) of the given data set, see [6], [7]. Our work is in this direction as is [5]. Precisely, this work build on what was done in [5] by learning a projection (embedding) matrix from a given data set via the RIP. However, in sharp contrast to [5], our solution provides rigorous approximation guarantees.

To set up the problem, let us assume that we are given a set of $p \gg n$ sample points (training set) $\mathcal{X} = \{\mathbf{x}_j\}_{j=1}^{p}$. Then we impose that the embedding matrix we are learning $\mathbf{\Phi}$ satisfy RIP on the pairwise distances of the points in $\mathcal{X}$, that is $\mathbf{\Phi}$ satisfies (1) with $\mathbf{z}$ replaced by $\mathbf{x}_i - \mathbf{x}_j$ for all $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$ where $i \neq j$. $\mathbf{\Phi}$ is bi-Lipschitz due to the RIP construct. Theoretical guarantees for this approach relies on results from differential geometry, see [5] and the references therein.

If we replaced $\mathbf{z}$ in (1) by $\mathbf{x}_i - \mathbf{x}_j$ and normalized the pairwise distances, then the RIP condition (1) on $\mathcal{S}(\mathcal{X})$ becomes $(1-\delta) \leq \mathbf{v}_{ij}^T \mathbf{\Phi}^T \mathbf{\Phi} \mathbf{v}_{ij} \leq (1-\delta)$. This expression simplifies to $|\mathbf{v}_{ij}^T \mathbf{\Phi}^T \mathbf{\Phi} \mathbf{v}_{ij} - 1| \leq \delta$ for each $i \neq j$. Re-indexing the $\mathbf{v}_{ij}$ to $\mathbf{v}_l$ for $l = 1, \ldots, M$, where $M = \binom{p}{2}$, we form the M secant vectors $\mathcal{S}(\mathcal{X}) = \{\mathbf{v}_1, \ldots, \mathbf{v}_M\}$ into an $n \times M$ matrix $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_M]$ and let $\mathbf{B} = \mathbf{\Phi}^T \mathbf{\Phi}$. Then we define a linear transform $\boldsymbol{\mathcal{A}} : \mathbb{S}_+^{n \times n} \to \mathbb{R}^M$ as:

$$\boldsymbol{\mathcal{A}}(\mathbf{B}) := \operatorname{diag}\left(\mathbf{V}^T \mathbf{B} \mathbf{V}\right), \tag{3}$$

where $\operatorname{diag}(\mathbf{H})$ denotes a vector of the entries of the principal diagonal of the matrix $\mathbf{H}$. Note that the rank of $\mathbf{B}$ is the same as that of $\mathbf{\Phi}$ and $\mathbf{B}$ is a PSD self-adjoint matrix. In addition, we place a constraint on the energy of $\mathbf{B}$ to be a fixed budget, say b, which adds a trace constraint to our problem and in practice may translates for example to having the entries of $\mathbf{\Phi}$ to all have a certain magnitude range. So our problem of adaptively learning an energy-aware RIP matrix $\mathbf{\Phi}$ and an RIC δ is equivalent to the following trace constrained affine rank minimization (ARM) problem:

$$\begin{aligned} \min_{\mathbf{B}} \quad & \|\boldsymbol{\mathcal{A}}(\mathbf{B}) - \mathbf{1}_M\|_\infty \\ \text{s.t.} \quad & \mathbf{B} \succeq 0, \quad \operatorname{rank}(\mathbf{B}) = r, \quad \operatorname{trace}(\mathbf{B}) = b. \end{aligned} \tag{4}$$

In [5], they solve a different problem by constraining the value of δ. Then they use eigen-decomposition to reach a number of samples. We directly take the constraints, design the matrix and give approximation guarantees. In our case, our algorithm returns the factors directly, which reduces the post processing costs such as taking eigendecompositions.

III. Proposed design and our algorithm

Problem (4), as is common practice for ARM problems, can be relaxed as follows:

$$\begin{aligned} \min_{\mathbf{B}} \quad & \|\mathbf{y} - \boldsymbol{\mathcal{A}}(\mathbf{B})\|_\infty \\ \text{s.t.} \quad & \operatorname{rank}(\mathbf{B}) \leq r \quad \text{and} \quad \|\mathbf{B}\|_{\text{tr}} \leq b. \end{aligned} \tag{5}$$

where $\mathbf{y} = \mathbf{1}_M$ and $\|\mathbf{B}\|_{\text{tr}} \leq b$ captures the PSD and the trace constraints. Based on the work in [8], we reformulate (5) as a minimax game next.

A. Reformulation of (5)

We first define a linear map $\boldsymbol{\mathcal{A}}_+ : \mathbb{S}^{n \times n} \to \mathbb{R}^{2M}$ where $\boldsymbol{\mathcal{A}}_+(\mathbf{B})$ is a concatenation of $\boldsymbol{\mathcal{A}}(\mathbf{B})$ and $-\boldsymbol{\mathcal{A}}(\mathbf{B})$ that is: $\boldsymbol{\mathcal{A}}_+(\mathbf{B}) = [\boldsymbol{\mathcal{A}}(\mathbf{B})^T, -\boldsymbol{\mathcal{A}}(\mathbf{B})^T]^T$, and correspondingly set $\mathbf{f} = [\mathbf{y}^T, -\mathbf{y}^T]^T$. Therefore, we have

$$\begin{aligned} \|\mathbf{y} - \boldsymbol{\mathcal{A}}(\mathbf{B})\|_\infty = \max_{i \in [2M]} |\, [\boldsymbol{\mathcal{A}}_+(\mathbf{B}) - \mathbf{f}]_i \,| = \\ \max_{i \in [2M]} \mathbf{e}_i^T \left(\boldsymbol{\mathcal{A}}_+(\mathbf{B}) - \mathbf{f}\right) = \max_{\mathbf{N} \in \Delta^{2M}} \mathcal{L}(\mathbf{N}, \mathbf{B}), \end{aligned} \tag{6}$$

where $\mathcal{L}(\mathbf{N}, \mathbf{B}) := \langle \mathbf{N}, (\boldsymbol{\mathcal{A}}_+(\mathbf{B}) - \mathbf{f}) \rangle$ and $\mathbf{e}_i$ is the canonical basis vector. The last equality in (6) is due to the fact that the maximum of a linear program occurs at a boundary point of the simplex Δ^{2M}. This reduces problem (5) to a minimax problem:

$$\min_{\mathbf{B} \in \Xi_r} \quad \max_{\mathbf{N} \in \Delta^{2M}} \quad \mathcal{L}(\mathbf{N}, \mathbf{B}) \tag{7}$$

where Ξ_r is the primal set, Δ^{2M} is the dual set and the mapping $\mathcal{L} : \Xi_r \times \Delta^{2M} \to \mathbb{R}$ is referred to as the *loss function* in game theory. We would need the following $\mathcal{L}_{\max} := \max_{\mathbf{N},\mathbf{B}} |\mathcal{L}(\mathbf{N}, \mathbf{B})| = \|\boldsymbol{\mathcal{A}}(\mathbf{B})\|_\infty + \|\mathbf{y}\|_\infty$. Note that $\boldsymbol{\mathcal{A}}_+^* : \mathbb{R}^{2M} \to \mathbb{S}^{n \times n}$, which is the adjoint of $\boldsymbol{\mathcal{A}}_+$, can be expressed in terms of the adjoint of $\boldsymbol{\mathcal{A}}$, denoted by $\boldsymbol{\mathcal{A}}^*$, precisely $\boldsymbol{\mathcal{A}}_+^*(\mathbf{w}) = \boldsymbol{\mathcal{A}}^*(\mathbf{w}_1 - \mathbf{w}_2)$ for $\mathbf{w} = [\mathbf{w}_1, \mathbf{w}_2]^T$ where $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^M$.

B. AMUSE algorithm

We now propose an algorithm that solves the minimax game (7) with provable theoretical guarantees: see **Algorithm 1**. It is important to note that the algorithm works with rank-1 updates B^t (in a matter similar to the conditional gradient descent algorithms [9]). As a result, after r iterations, our algorithm returns an estimator $\widehat{\mathbf{B}} = \frac{1}{r} \sum_{t=1}^{r} \mathbf{B}^t$. As we do not explicitly compute the product of the factors, the algorithm is scalable since each factor corresponds to 1 measurement. Moreover, we bound the recovery error as thus:

$$\|\boldsymbol{\mathcal{A}}(\widehat{\mathbf{B}}) - \mathbf{y}\|_\infty \leq \min_{\mathbf{B} \in \Xi_r} \|\boldsymbol{\mathcal{A}}(\mathbf{B}) - \mathbf{y}\|_\infty + \mathcal{O}\left(\frac{1}{\sqrt{r}}\right).$$

This is the first approximation bound for obtaining such sensing matrices.

Essentially, the MUSE for ARM (AMUSE) algorithm we propose is a modification of the Multiplicative Update Selector and Estimator (MUSE) algorithm for learning to play repeated games proposed in [8]. The MUSE itself can be thought of as a restatement of the Multiplicative Weights Algorithm (MWA), which in turn uses the Weighted Majority Algorithm, see [8] and references therein. We also point out also that the multiplicative updating has connections to Frank-Wolfe and related algorithms [10].

Steps 2 and 3 of the loop of AMUSE performs the multiplicative update of the dual variable $\mathbf{N}$ and the update is exactly the same as in MUSE for a given primal variable at iteration t, $\mathbf{B}^t$. Therefore the step size η remains the same as in the MUSE algorithm, [8]; that is $\eta = \ln\left(1 + \sqrt{2\ln(2M)/r}\right)$. As a result, the theoretical guarantees given in [8] for MUSE also holds for AMUSE. Basically, for a fixed matrix at iteration t, $\mathbf{B}^t$, the proof for the multiplicative update in [8] for the vector case remains the same.

Algorithm 1 MUSE for ARM (AMUSE)

Input: $\mathbf{y}$, η

Output: $\widehat{\mathbf{B}} \approx \mathbf{B}^*$ with $\text{rank}(\widehat{\mathbf{B}}) \leq r$

Initialize $\mathbf{N}^1 = \frac{1}{2M}\mathbf{1}_{2M}$

For $t = 1, \ldots, r$ **do**

1. Find $\mathbf{B}^t = \text{argmin}_{\|\mathbf{B}\|_{\text{tr}} \leq 1} \ \mathcal{L}(\mathbf{N}^t, \mathbf{B})$
2. Set $\mathbf{Q}_j^{t+1} = \mathbf{N}_j^t \cdot e^{\frac{\eta \cdot \mathcal{L}(\mathbf{e}_j, \mathbf{B}^t)}{\mathcal{L}_{\max}}}$ for $j \in [2M]$
3. Update $\mathbf{N}^{t+1} = \frac{\mathbf{Q}^{t+1}}{\sum_{j=1}^{2M} \mathbf{Q}_j^{t+1}}$

End for

Return $\widehat{\mathbf{B}} = \frac{1}{r}\sum_{t=1}^{r} \mathbf{B}^t$

Note that the main and crucial difference between AMUSE and MUSE is the first step of the loop where we update our primal variable $\mathbf{B}$ given our dual variable at iteration t, $\mathbf{N}^t$, by $\mathbf{B}^t = \text{argmin}_{\|\mathbf{B}\|_{\text{tr}} \leq 1} \ \mathcal{L}(\mathbf{N}^t, \mathbf{B})$. These updates have rank 1 and hence their linear combination, $\widehat{\mathbf{B}}$, has rank at most r, since rank is sub-additive.

AMUSE is used to approximate problem (4) by rescaling to meet the trace constraint. The parameter η remain the same and $\mathcal{L}_{\max} = 1 + \max_i \max_j v_{ij}^2$ where v_{ij} is the (i,j) entry of $\mathbf{V}$.

IV. Analysis

A. AMUSE guarantees

The following theorem formalizes our claim that the AMUSE algorithm outputs an approximate solution $\widehat{\mathbf{B}}$ with $\text{rank}(\widehat{\mathbf{B}}) \leq r$ with a bounded ℓ_∞ loss in the measurement domain after r iterations. The proof of this theorem use Lemma 4.1 of [8].

Theorem 1: Let AMUSE return $\widehat{\mathbf{B}}$ after r iterations. Then $\text{rank}(\widehat{\mathbf{B}}) \leq r$ and $\|\boldsymbol{\mathcal{A}}(\widehat{\mathbf{B}}) - \mathbf{y}\|_\infty$ is at most

$$\|\mathbf{e}\|_\infty + \left(1 + \sqrt{2}\right) \cdot \left(2\|\boldsymbol{\mathcal{A}}(\widehat{\mathbf{B}})\|_\infty + \|\mathbf{e}\|_\infty\right) \sqrt{\frac{\ln(2M)}{r}},$$

where $\mathbf{e}$ measures the perturbation of the linear model.

Proof: We sketch the proof as follows, for details see [8]. By the definition of $\boldsymbol{\mathcal{A}}$, $\mathbf{y}$ and $\mathcal{L}$, $\|\boldsymbol{\mathcal{A}}(\widehat{\mathbf{B}}) - \mathbf{y}\|_\infty = \max_{\mathbf{N}} \mathcal{L}(\mathbf{N}, \widehat{\mathbf{B}})$. Then we first show that $\min_{\mathbf{B}} \max_{\mathbf{N}} \mathcal{L}(\mathbf{N}, \mathbf{B}) + (1+\sqrt{2})\mathcal{L}_{\max}\sqrt{\frac{\ln(2M)}{r}}$ upper bounds $\max_{\mathbf{N}} \mathcal{L}(\mathbf{N}, \widehat{\mathbf{B}})$, a key ingredient of which is the min-max theorem. Next we deduce that $\min_{\mathbf{B}} \max_{\mathbf{N}} \mathcal{L}(\mathbf{N}, \mathbf{B}) = \min_{\mathbf{B}} \|\boldsymbol{\mathcal{A}}(\mathbf{B}) - \mathbf{y}\|_\infty \leq \|\mathbf{e}\|_\infty$. Then, using the triangle inequality we bound $\mathcal{L}_{\max}$ by bounding $\|\mathbf{y}\|_\infty$ as thus: $\mathcal{L}_{\max} = \|\boldsymbol{\mathcal{A}}(\mathbf{B})\|_\infty + \|\mathbf{y}\|_\infty$ which is upper bounded by $2\|\boldsymbol{\mathcal{A}}(\mathbf{B})\|_\infty + \|\mathbf{e}\|_\infty$. ■

Furthermore, we bound the error of the output of AMUSE for the RIP matrix learning problem in Corollary 1 which follows from Theorem 1.

Corollary 1: Let AMUSE learn an RIP matrix $\widehat{\mathbf{B}}$ from a given data set $\mathcal{X}$ after r iterations with RIC $\hat{\delta}$. Assume that the optimal RIP matrix for that $\mathcal{X}$ has RIC δ^*. Then $\widehat{\mathbf{B}}$ has $\text{rank}(\widehat{\mathbf{B}}) \leq r$ and

$$\|\boldsymbol{\mathcal{A}}(\widehat{\mathbf{B}}) - \mathbf{1}_M\|_\infty \leq \delta^* + 2(1+\sqrt{2})\sqrt{\frac{\ln(2M)}{r}}.$$

This implies that if the optimal solution $\boldsymbol{\Phi}^*$ has RIC δ^* on the training set, then our approximation, $\widehat{\boldsymbol{\Phi}}$, of $\boldsymbol{\Phi}^*$ also satisfies RIP on these data points but with a slightly larger constant $\hat{\delta} \leq \delta^* + \mathcal{O}\left(1/\sqrt{r}\right)$. As the dimensions increase, we approximate the best RIP constant for the given dataset.

B. Generalization bounds

Interestingly, we can provably approximate the optimal RIC even for points that are outside our sample points as stated in the following proposition.

Proposition 1: Given the pair δ and $\boldsymbol{\Phi}$ as the optimal solution to (4), $\boldsymbol{\Phi}$ applied to any $\boldsymbol{z}$ with $\|\boldsymbol{z} - \mathbf{x}\|_2 \leq \epsilon$ for all $\mathbf{x} \in \mathcal{X}$ and $\epsilon \in [0, 1)$ gives an RIC, $\bar{\delta}$, bounded as follows:

$$\bar{\delta} \leq (\delta + \epsilon)/(1 - \epsilon). \tag{8}$$

Proof: Since $\boldsymbol{\Phi}$ is linear w.l.o.g let $\|\mathbf{x}\|_2 = 1$. For any $\boldsymbol{z}$ such that $\|\boldsymbol{z} - \mathbf{x}\|_2 \leq \epsilon$ and $\|\boldsymbol{z}\|_2 = 1$ then $\|\boldsymbol{\Phi}\boldsymbol{z}\|_2$ can be written as:

$$\|\boldsymbol{\Phi}\left(\mathbf{x} - (\boldsymbol{z} - \mathbf{x})\right)\|_2 \leq \|\boldsymbol{\Phi}\mathbf{x}\|_2 + \|\boldsymbol{\Phi}(\boldsymbol{z} - \mathbf{x})\|_2$$

using the triangle inequality. Let α_1 be the smallest constant such that $\|\boldsymbol{\Phi}\boldsymbol{z}\|_2 \leq (1 + \alpha_1)\|\boldsymbol{z}\|_2$ then with the definition of δ from the above inequality we have

$$\|\boldsymbol{\Phi}\boldsymbol{z}\|_2 \leq (1 + \delta)\|\mathbf{x}\|_2 + (1 + \alpha_1)\|\boldsymbol{z} - \mathbf{x}\|_2.$$

Evaluating and upper bounding the norms and using the definition of α_1 gives

$$(1 + \alpha_1) \leq (1 + \delta) + (1 + \alpha_1)\epsilon.$$

This simplifies to $\alpha_1 \leq (\delta + \epsilon)/(1 - \epsilon)$. Similarly, we lower bound $\|\boldsymbol{\Phi}\boldsymbol{z}\|_2$ and have an α_2 to be the largest constant such that $\|\boldsymbol{\Phi}\boldsymbol{z}\|_2 \geq (1 - \alpha_1)\|\boldsymbol{z}\|_2$, this leads to a bound on α_2 as thus: $\alpha_2 \leq (\delta + \epsilon)/(1 + \epsilon)$. The RIC, $\bar{\delta}$, is therefore given by $\max(\alpha_1, \alpha_2)$ and for the values of ϵ considered this is α_1, hence (8). ■

V. Empirical results

We use the synthetic data set of images of translations of white squares in a black background from [5]. In the first experiment we investigate the dependence of RIC we learn on the number of rows (or rank) of the $\boldsymbol{\Phi}$ we learn. Here, we use $M = 2000$ number of secants vectors. We use the same for PCA projected to meet the trace constraint of our problem (4) and also generate a random Gaussian matrix also constrained to have trace as our problem. Figure 1 displays this comparison, where our method clearly outperforms PCA and random designs.

In the second experiment we learn a $\boldsymbol{\Phi}$ from the data and use it to encode a randomly selected subset of $\mathcal{X}$ corrupted with Gaussian noise of varying signal-to-noise ratio (SNR).

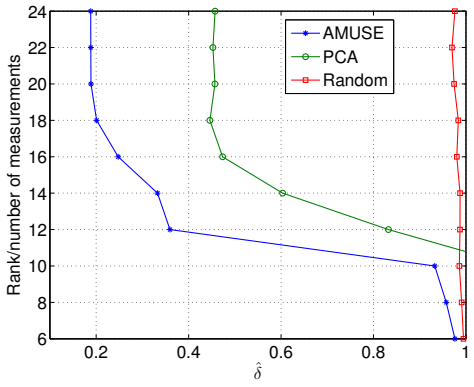

Fig. 1. A plot of the number of measurements (or rank of the $\widehat{\mathbf{\Phi}}$) as a function of the RIC $\widehat{\delta}$ for data points with an ambient dimension $n = 256$.

We then do Basis Pursuit denoising to decode these points. For comparison we use a Gaussian matrix with the trace-constrained and compute the mean-square error (MSE) over the subset. The results are displayed in Figure 2, which show that our approach outperforms the random projections due to its adaptivity to the underlying data manifold. Note that in this experiment, we simply searched over Frobenius norm constraint to approximate the RIC without any energy constraint.

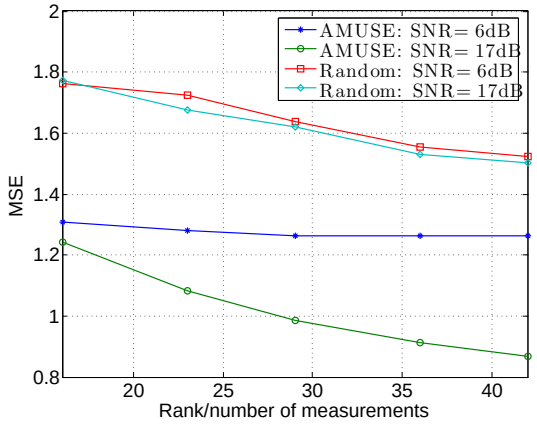

Fig. 2. CS recovery performance of our adaptive approach compared to energy constrained random projections.

VI. Conclusions

We reformulate the adaptive learning of a data embedding into an optimization problem and propose an algorithm that approximately solves this problem with provable guarantees. We show generalizability of our embedding to a test data set ϵ away from the training set in terms of the RIC of the embedding matrix learnt. Our experiments show better performance of our derived matrices as compared to random designs with regard to the empirical RIC and CS recovery.

References

[1] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Contemporary mathematics*, vol. 26, no. 189-206, p. 1, 1984.

[2] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[3] E. Candes and T. Tao, "Decoding by linear programming," *Information Theory, IEEE Transactions on*, vol. 51, no. 12, pp. 4203–4215, 2005.

[4] H. Rauhut, "Compressive sensing and structured random matrices," *Theoretical foundations and numerical methods for sparse recovery*, vol. 9, pp. 1–92, 2010.

[5] C. Hegde, A. Sankaranarayanan, W. Yin, and R. Baraniuk, "A convex approach for learning near-isometric linear embeddings," *preparation, August*, 2012.

[6] M. Elad, "Optimized projections for compressed sensing," *Signal Processing, IEEE Transactions on*, vol. 55, no. 12, pp. 5695–5702, 2007.

[7] J. M. Duarte-Carvajalino and G. Sapiro, "Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization," *Image Processing, IEEE Transactions on*, vol. 18, no. 7, pp. 1395–1408, 2009.

[8] S. Jafarpour, R. Schapire, and V. Cevher, "Compressive sensing meets game theory," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3660–3663.

[9] D. P. Bertsekas, "Nonlinear programming," 1999.

[10] K. L. Clarkson, "Coresets, sparse greedy approximation, and the frank-wolfe algorithm," *ACM Transactions on Algorithms (TALG)*, vol. 6, no. 4, p. 63, 2010.