Eigenvalues/eigenvectors approximation via CS

Edem Boahen[a], Mark Iwen[a,b]

[a]Department of Mathematics, Michigan State University, East Lansing, USA

[b]Department of Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, USA

boahened@msu.edu, markiwen@math.msu.edu

Abstract

The purpose of this article is to describe how the various experiments performed to empirically validate the MAM* method were made.

Keywords: low rank; sparsity; sublinear; recovery error, RIP

1 Experimental Setup

In this section, we outline the experimental setup used to evaluate the accuracy of the eigenvector approximation ($MAM*$) method under various conditions. Specifically, we vary the rank r of the target signal matrix while keeping the ambient dimension and sparsity level fixed.

1.1 Construction of the PSD Matrix A

The matrix $A \in R^{N\times N}$ is designed to be symmetric and positive semidefinite (PSD), with low-rank and sparse eigenvectors. The construction depends on three parameters:

- N: the ambient dimension of the signal,
- r: the desired rank of A (i.e., the number of eigenvectors),
- s: the sparsity level of each eigenvector.

We consider two different models for constructing the eigenvectors of A:

Model 1: Disjoint Support Eigenvectors In this model, each eigenvector $u_j \in R^N$ has support on a unique, nonoverlapping subset of $\{0,\dots,N-1\}$:

- The support of u_1 is $\{0,1,\dots,s-1\}$,
- The support of u_2 is $\{s,s+1,\dots,2s-1\}$, and so on.

Each u_j is then normalized to unit norm. The matrix A is formed as:

$$A = \sum_{j=1}^{r} \lambda_j u_j u_j^\top,$$

where the values λ_j define the decay of the eigenvalue (see below).

Model 2: Overlapping support eigenvectors In this case, we first generate r orthonormal vectors in R^s, then embed each into a common subset $S \subset [N]$ (of size s) selected uniformly at random. Each orthogonal vector is placed in the positions indexed by S, yielding orthonormal vectors r in R^N with overlap support. The matrix A is constructed again as:

$$A = \sum_{j=1}^{r} \lambda_j u_j u_j^\top.$$

It should be noted that the entries of the eigenvectors are first sampled independently from a standard normal distribution $\mathcal{N}(0,1)$ before any normalization.

Eigenvalue Decay Models. Two decay models for the eigenvalues $\{\lambda_j\}_{j=1}^r$ are considered:

- **Exponential decay:** $\lambda_j = 2^{-j}$,
- **Linear decay:** $\lambda_j = 1 - 0.1j$.

Both constructions ensure that A is symmetric, positive semidefinite, and low rank, with control over the structure of the eigenvector and the spectral decay.

1.2 Construction of the Matrix M and $\tilde{M}A\tilde{M}^*$

The matrix M is not explicitely constructed in our experiment. For the sake of optimizing the space usage of our code we only store the list of primes that the matrix M is based on. By exploiting the structure of M and the sparsity of the extended Bit testing matrix B'_N we then efficiently compute the vectors $\tilde{u}_j = \tilde{M}u_j = (B'_N \circ M)u_j$ where u_j is the j^{th} singular vector of the PSD matrix A. We then construct the $\tilde{M}A\tilde{M}^*$ matrix as follow:

$$\tilde{M}A\tilde{M}^* = \sum_{j=1}^{r} \lambda_j \frac{\tilde{u}_j \tilde{u}_j^\top}{\|\tilde{u}_j\|^2}.$$

1.3 Evaluation Metrics

We evaluated our method using four error metrics.

- Relative ℓ_2 error before inversion: $\min\left\{\frac{\|\tilde{u}_j - \hat{u}_j\|_2}{\|\tilde{u}_j\|}, \frac{\|\tilde{u}_j + \hat{u}_j\|_2}{\|\tilde{u}_j\|}\right\}$
- ℓ_∞ error before inversion: $\min\{\|\tilde{u}_j - \hat{u}_j\|_\infty, \|\tilde{u}_j + \hat{u}_j\|_\infty\}$
- Relative ℓ_2 error after inversion: $\min\left\{\frac{\|u_j - \hat{z}_j\|_2}{\|u_j\|}, \frac{\|u_j + \hat{z}_j\|_2}{\|u_j\|}\right\}$
- ℓ_∞ error before inversion: $\min\{\|u_j - \hat{z}_j\|_\infty, \|u_j + \hat{z}_j\|_\infty\}$

Here we denote by $\hat{u}_j$ the j^{th} eigenvector of $\tilde{M}A\tilde{M}$ and $\hat{z}_j$ the approximation to u_j using the Recovery algorithm with the approximate measurement $\mathbf{y} = \tilde{u}_j$.
These metrics are computed both before and after applying our inverse approximation method over a number of runs. We then plot the average of these errors in three different cases.

1.4 Varying Sparsity of Eigenvectors

In a single instance of this experiment, we vary the sparsity s of the eigenvectors in the matrix A.The rank r of the matrix A and the number of rows in the matrix M remain fixed. We then compute the four errors as defined above. Finally we run this experiment 20 times and plot the average errors against the sparsity level for the first four eigenvectors. We observe that as the sparsity of the eigenvectors increases both the errors before and after recovery quickly becomes worse.

1.5 Varying Rank of Matrix A

In a single instance of this experiment, we vary the rank r of the matrix A.The sparsity level s of the eigenvalues of the matrix A and the number of rows in the matrix M remain fixed. We then compute the four errors as defined above. Finally we run this experiment 20 times and plot the average error against the rank for the first four eigenvectors. We observe that as the rank is increasing, the error after recovery slowly increases with the value of r even though the error before recovery do not vary much.

1.6 Varying Number of Rows in M

We investigate the role of the number of rows in M i.e., the number of prime numbers used in its construction. Increasing the number of rows corresponds to increasing the number of linear measurements used to approximate the eigenvectors. In this experiment we both the sparsity s of the eigenvectors and the rank r of the matrix A are fixed. We then vary the number of rows of M. We then compute the four errors as defined above and plot the average errors against the number of rows for the first four eigenvectors.