Greedy Subspace Clustering

Dohyung Park
Dept. of Electrical and Computer Engineering
The University of Texas at Austin
dhpark@utexas.edu

Constantine Caramanis
Dept. of Electrical and Computer Engineering
The University of Texas at Austin
constantine@utexas.edu

Sujay Sanghavi
Dept. of Electrical and Computer Engineering
The University of Texas at Austin
sanghavi@mail.utexas.edu

Abstract

We consider the problem of subspace clustering: given points that lie on or near the *union* of many low-dimensional linear subspaces, recover the subspaces. To this end, one first identifies sets of points close to the same subspace and uses the sets to estimate the subspaces. As the geometric structure of the clusters (linear subspaces) forbids proper performance of general distance based approches such as K-means, many model-specific methods have been proposed. In this paper, we provide new simple and efficient algorithms for this problem. Our statistical analysis shows that the algorithms are guaranteed exact (perfect) clustering performance under certain conditions on the number of points and the affinity between subspaces. These conditions are weaker than those considered in the standard statistical literature. Experimental results on synthetic data generated from the standard unions of subspaces model demonstrate our theory. We also show that our algorithm performs competitively against state-of-the-art algorithms on real-world applications such as motion segmentation and face clustering, but with much simpler implementation and lower computational cost.

1 Introduction

Subspace clustering is a classic problem where one is given points in a high-dimensional ambient space and would like to approximate them by a *union* of lower-dimensional linear subspaces. In particular, each subspace contains a subset of the points. This problem is hard because one needs to jointly find the subspaces, and the points corresponding to each; the data we are given are unlabeled. The unions of subspaces model naturally arises in settings where data from multiple latent phenomena are mixed together and need to be separated. Applications of subspace clustering include motion segmentation [23], face clustering [8], gene expression analysis [10], and system identification [22]. In these applications, data points with the same label (e.g., face images of a person under varying illumination conditions, feature points of a moving rigid object in a video sequence) lie on a low-dimensional subspace, and the mixed dataset can be modeled by unions of subspaces. For detailed description of the applications, we refer the readers to the reviews [10, 20] and references therein.

There is now a sizable literature on empirical methods for this particular problem and some statistical analysis as well. Many recently proposed methods, which perform remarkably well and have theoretical guarantees on their performances, can be characterized as involving two steps: *(a)* finding a "neighborhood" for each data point, and *(b)* finding the subspaces and/or clustering the points given these neighborhoods. Here, neighbors of a point are other points that the algorithm estimates to lie on the same subspace as the point (and not necessarily just closest in Euclidean distance).

		Subspace	Conditions for:	
Algorithm	What is guaranteed	condition	Fully random model	Semi-random model
SSC [4, 16]	Correct neighborhoods	None	$\frac{d}{p} = O(\frac{\log(n/d)}{\log(nL)})$	$\max \mathrm{aff} = O(\frac{\sqrt{\log(n/d)}}{\log(nL)})$
LRR [14]	Exact clustering	No intersection	-	-
SSC-OMP [3]	Correct neighborhoods	No intersection	-	-
TSC [6, 7]	Exact clustering	None	$\frac{d}{p} = O(\frac{1}{\log(nL)})$	$\max \mathrm{aff} = O(\frac{1}{\log(nL)})$
LRSSC [24]	Correct neighborhoods	None	$\frac{d}{p} = O(\frac{1}{\log(nL)})$	-
NSN+GSR	Exact clustering	None	$\frac{d}{p} = O(\frac{\log n}{\log(ndL)})$	$\max \mathrm{aff} = O(\sqrt{\frac{\log n}{(\log dL)\cdot\log(ndL)}})$
NSN+Spectral	Exact clustering	None	$\frac{d}{p} = O(\frac{\log n}{\log(ndL)})$	-

Table 1: Subspace clustering algorithms with theoretical guarantees. LRR and SSC-OMP have only deterministic guarantees, not statistical ones. In the two standard statistical models, there are n data points on each of L d-dimensional subspaces in $\mathbb{R}^p$. For the definition of $\max \mathrm{aff}$, we refer the readers to Section 3.1.

Our contributions: In this paper we devise **new algorithms** for each of the two steps above; *(a)* we develop a new method, Nearest Subspace Neighbor (NSN), to determine a neighborhood set for each point, and *(b)* a new method, Greedy Subspace Recovery (GSR), to recover subspaces from given neighborhoods. Each of these two methods can be used in conjunction with other methods for the corresponding other step; however, in this paper we focus on two algorithms that use NSN followed by GSR and Spectral clustering, respectively. Our main result is establishing **statistical guarantees for exact clustering with general subspace conditions**, in the standard models considered in recent analytical literature on subspace clustering. Our condition for exact recovery is weaker than the conditions of other existing algorithms that only guarantee *correct neighborhoods*[1], which do not always lead to correct clustering. We provide numerical results which demonstrate our theory. We also show that for the real-world applications our algorithm performs competitively against those of state-of-the-art algorithms, but the computational cost is much lower than them. Moreover, our algorithms are much simpler to implement.

1.1 Related work

The problem was first formulated in the data mining community [10]. Most of the related work in this field assumes that an underlying subspace is parallel to some canonical axes. Subspace clustering for unions of arbitrary subspaces is considered mostly in the machine learning and the computer vision communities [20]. Most of the results from those communities are based on empirical justification. They provided algorithms derived from theoretical intuition and showed that they perform empirically well with practical dataset. To name a few, GPCA [21], Spectral curvature clustering (SCC) [2], and many iterative methods [1, 19, 26] show their good empirical performance for subspace clustering. However, they lack theoretical analysis that guarantees exact clustering.

As described above, several algorithms with a common structure are recently proposed with both theoretical guarantees and remarkable empirical performance. Elhamifar and Vidal [4] proposed an algorithm called Sparse Subspace Clustering (SSC), which uses ℓ_1-minimization for neighborhood construction. They proved that if the subspaces have *no intersection*[2], SSC always finds a correct neighborhood matrix. Later, Soltanolkotabi and Candes [16] provided a statistical guarantee of the algorithm for subspaces with intersection. Dyer et al. [3] proposed another algorithm called SSC-OMP, which uses Orthogonal Matching Pursuit (OMP) instead of ℓ_1-minimization in SSC. Another algorithm called Low-Rank Representation (LRR) which uses nuclear norm minimization is proposed by Liu et al. [14]. Wang et al. [24] proposed an hybrid algorithm, Low-Rank and Sparse Subspace Clustering (LRSSC), which involves both ℓ_1-norm and nuclear norm. Heckel and Bölcskei [6] presented Thresholding based Subspace Clustering (TSC), which constructs neighborhoods based on the inner products between data points. All of these algorithms use spectral clustering for the clustering step.

The analysis in those papers focuses on neither exact recovery of the subspaces nor exact clustering in general subspace conditions. SSC, SSC-OMP, and LRSSC only guarantee correct neighborhoods which do not always lead to exact clustering. LRR guarantees exact clustering only when

[1] By correct neighborhood, we mean that for each point every neighbor point lies on the same subspace.

[2] By no intersection between subspaces, we mean that they share only the null point.

the subspaces have no intersections. In this paper, we provide novel algorithms that guarantee exact clustering in general subspace conditions. When we were preparing this manuscript, it is proved that TSC guarantees exact clustering under certain conditions [7], but the conditions are stricter than ours. (See Table 1)

1.2 Notation

There is a set of N data points in $\mathbb{R}^p$, denoted by $\mathcal{Y} = \{y_1, \ldots, y_N\}$. The data points are lying on or near a union of L subspaces $\mathcal{D} = \cup_{i=1}^{L} \mathcal{D}_i$. Each subspace $\mathcal{D}_i$ is of dimension d_i which is smaller than p. For each point y_j, w_j denotes the index of the nearest subspace. Let N_i denote the number of points whose nearest subspace is $\mathcal{D}_i$, i.e., $N_i = \sum_{j=1}^{N} \mathbb{I}_{w_j = i}$. Throughout this paper, sets and subspaces are denoted by calligraphic letters. Matrices and key parameters are denoted by letters in upper case, and vectors and scalars are denoted by letters in lower case. We frequently denote the set of n indices by $[n] = \{1, 2, \ldots, n\}$. As usual, $\mathrm{span}\{\cdot\}$ denotes a subspace spanned by a set of vectors. For example, $\mathrm{span}\{v_1, \ldots, v_n\} = \{v : v = \sum_{i=1}^{n} \alpha_i v_i, \alpha_1, \ldots, \alpha_n \in \mathbb{R}\}$. $\mathrm{Proj}_{\mathcal{U}} y$ is defined as the projection of y onto subspace $\mathcal{U}$. That is, $\mathrm{Proj}_{\mathcal{U}} y = \arg\min_{u \in \mathcal{U}} \|y - u\|_2$. $\mathbb{I}\{\cdot\}$ denotes the indicator function which is one if the statement is true and zero otherwise. Finally, $\bigoplus$ denotes the direct sum.

2 Algorithms

We propose two algorithms for subspace clustering as follows.

- NSN+GSR : Run Nearest Subspace Neighbor (NSN) to construct a neighborhood matrix $W \in \{0, 1\}^{N \times N}$, and then run Greedy Subspace Recovery (GSR) for W.
- NSN+Spectral : Run Nearest Subspace Neighbor (NSN) to construct a neighborhood matrix $W \in \{0, 1\}^{N \times N}$, and then run spectral clustering for $Z = W + W^\top$.

2.1 Nearest Subspace Neighbor (NSN)

NSN approaches the problem of *finding neighbor points most likely to be on the same subspace* in a greedy fashion. At first, given a point y without any other knowledge, the one single point that is most likely to be a neighbor of y is the nearest point of the line $\mathrm{span}\{y\}$. In the following steps, if we have found a few correct neighbor points (lying on the same true subspace) and have no other knowledge about the true subspace and the rest of the points, then the most potentially correct point is the one closest to the subspace spanned by the correct neighbors we have. This motivates us to propose NSN described in the following.

Algorithm 1 Nearest Subspace Neighbor (NSN)

```
Input: A set of $N$ samples $\mathcal{Y} = \{y_1, \ldots, y_N\}$, The number of required neighbors $K$, Maximum
  subspace dimension $k_{\max}$.
Output: A neighborhood matrix $W \in \{0,1\}^{N \times N}$
  $y_i \leftarrow y_i / \|y_i\|_2, \forall i \in [N]$                          ▷ Normalize magnitudes
  for $i = 1, \ldots, N$ do                                       ▷ Run NSN for each data point
    $\mathcal{I}_i \leftarrow \{i\}$
    for $k = 1, \ldots, K$ do          ▷ Iteratively add the closest point to the current subspace
      if $k \le k_{\max}$ then
        $\mathcal{U} \leftarrow \mathrm{span}\{y_j : j \in \mathcal{I}_i\}$
      end if
      $j^* \leftarrow \arg\max_{j \in [N] \setminus \mathcal{I}_i} \|\mathrm{Proj}_{\mathcal{U}} y_j\|_2$
      $\mathcal{I}_i \leftarrow \mathcal{I}_i \cup \{j^*\}$
    end for
    $W_{ij} \leftarrow \mathbb{I}_{j \in \mathcal{I}_i \text{ or } y_j \in \mathcal{U}}, \ \forall j \in [N]$         ▷ Construct the neighborhood matrix
  end for
```

NSN collects K neighbors sequentially for each point. At each step k, a k-dimensional subspace $\mathcal{U}$ spanned by the point and its $k - 1$ neighbors is constructed, and the point closest to the subspace is newly collected. After $k \geq k_{\max}$, the subspace $\mathcal{U}$ constructed at the $k_{\max}$th step is used for collecting neighbors. At last, if there are more points lying on $\mathcal{U}$, they are also counted as neighbors. The subspace $\mathcal{U}$ can be stored in the form of a matrix $U \in \mathbb{R}^{p \times \dim(\mathcal{U})}$ whose columns form an orthonormal basis of $\mathcal{U}$. Then $\|\mathrm{Proj}_{\mathcal{U}} y_j\|_2$ can be computed easily because it is equal to $\|U^\top y_j\|_2$. While a naive implementation requires $O(K^2pN^2)$ computational cost, this can be reduced to $O(KpN^2)$, and the faster implementation is described in Section A.1. We note that this computational cost is much lower than that of the convex optimization based methods (e.g., SSC [4] and LRR [14]) which solve a convex program with N^2 variables and pN constraints.

NSN for subspace clustering shares the same philosophy with Orthogonal Matching Pursuit (OMP) for sparse recovery in the sense that it incrementally picks the point (dictionary element) that is the most likely to be correct, assuming that the algorithms have found the correct ones. In subspace clustering, that point is the one closest to the subspace spanned by the currently selected points, while in sparse recovery it is the one closest to the residual of linear regression by the selected points. In the sparse recovery literature, the performance of OMP is shown to be comparable to that of Basis Pursuit (ℓ_1-minimization) both theoretically and empirically [18, 11]. One of the contributions of this work is to show that this high-level intuition is indeed born out, provable, as we show that NSN also performs well in collecting neighbors lying on the same subspace.

2.2 Greedy Subspace Recovery (GSR)

Suppose that NSN has found correct neighbors for a data point. How can we check if they are indeed correct, that is, lying on the same true subspace? One natural way is to count the number of points close to the subspace spanned by the neighbors. If they span one of the true subspaces, then many other points will be lying on the span. If they do not span any true subspaces, few points will be close to it. This fact motivates us to use a greedy algorithm to recover the subspaces. Using the neighborhood constructed by NSN (or some other algorithm), we recover the L subspaces. If there is a neighborhood set containing only the points on the same subspace for each subspace, the algorithm successfully recovers the unions of the true subspaces exactly.

Algorithm 2 Greedy Subspace Recovery (GSR)

```
Input: $N$ points $\mathcal{Y} = \{y_1, \ldots, y_N\}$, A neighborhood matrix $W \in \{0,1\}^{N \times N}$, Error bound $\epsilon$
Output: Estimated subspaces $\hat{\mathcal{D}} = \cup_{l=1}^{L} \hat{\mathcal{D}}_l$. Estimated labels $\hat{w}_1, \ldots, \hat{w}_N$
  $y_i \leftarrow y_i / \|y_i\|_2, \forall i \in [N]$    ▷ Normalize magnitudes
  $\mathcal{W}_i \leftarrow \text{Top-}d\{y_j : W_{ij} = 1\}, \forall i \in [N]$    ▷ Estimate a subspace using the neighbors for each point
  $\mathcal{I} \leftarrow [N]$
  while $\mathcal{I} \neq \emptyset$ do    ▷ Iteratively pick the best subspace estimates
      $i^* \leftarrow \arg\max_{i \in \mathcal{I}} \sum_{j=1}^{N} \mathbb{I}\{\|\mathrm{Proj}_{\mathcal{W}_i} y_j\|_2 \geq 1 - \epsilon\}$
      $\hat{\mathcal{D}}_l \leftarrow \hat{\mathcal{W}}_{i^*}$
      $\mathcal{I} \leftarrow \mathcal{I} \setminus \{j : \|\mathrm{Proj}_{\mathcal{W}_{i^*}} y_j\|_2 \geq 1 - \epsilon\}$
  end while
  $\hat{w}_i \leftarrow \arg\max_{l \in [L]} \|\mathrm{Proj}_{\hat{\mathcal{D}}_l} y_i\|_2, \ \forall i \in [N]$    ▷ Label the points using the subspace estimates
```

Recall that the matrix W contains the labelings of the points, so that $W_{ij} = 1$ if point i is assigned to subspace j. Top-$d\{y_j : W_{ij} = 1\}$ denotes the d-dimensional principal subspace of the set of vectors $\{y_j : W_{ij} = 1\}$. This can be obtained by taking the first d left singular vectors of the matrix whose columns are the vector in the set. If there are only d vectors in the set, Gram-Schmidt orthogonalization will give us the subspace. As in NSN, it is efficient to store a subspace $\mathcal{W}_i$ in the form of its orthogonal basis because we can easily compute the norm of a projection onto the subspace.

Testing a candidate subspace by counting the number of near points has already been considered in the subspace clustering literature. In [25], the authors proposed to run RANdom SAmple Consensus (RANSAC) iteratively. RANSAC randomly selects a few points and checks if there are many other points near the subspace spanned by the collected points. Instead of randomly choosing sample points, GSR receives some candidate subspaces (in the form of sets of points) from NSN (or possibly some other algorithm) and selects subspaces in a greedy way as specified in the algorithm above.

3 Theoretical results

We analyze our algorithms in two standard noiseless models. The main theorems present sufficient conditions under which the algorithms cluster the points exactly with high probability. For simplicity of analysis, we assume that every subspace is of the same dimension, and the number of data points on each subspace is the same, i.e., $d \triangleq d_1 = \cdots = d_L, \quad n \triangleq N_1 = \cdots = N_L$. We assume that d is known to the algorithm. Nonetheless, our analysis can extend to the general case.

3.1 Statistical models

We consider two models which have been used in the subspace clustering literature:

- Fully random model: The subspaces are drawn iid uniformly at random, and the points are also iid randomly generated.
- Semi-random model: The subspaces are arbitrarily determined, but the points are iid randomly generated.

Let $D_i \in \mathbb{R}^{p \times d}, i \in [L]$ be a matrix whose columns form an orthonormal basis of $\mathcal{D}_i$. An important measure that we use in the analysis is the *affinity* between two subspaces, defined as

$$\mathrm{aff}(i,j) \triangleq \frac{\|D_i^\top D_j\|_F}{\sqrt{d}} = \sqrt{\frac{\sum_{k=1}^{d} \cos^2 \theta_k^{i,j}}{d}} \in [0,1],$$

where $\theta_k^{i,j}$ is the kth principal angle between $\mathcal{D}_i$ and $\mathcal{D}_j$. Two subspaces $\mathcal{D}_i$ and $\mathcal{D}_j$ are identical if and only if $\mathrm{aff}(i,j) = 1$. If $\mathrm{aff}(i,j) = 0$, every vector on $\mathcal{D}_i$ is orthogonal to any vectors on $\mathcal{D}_j$. We also define the maximum affinity as

$$\max \mathrm{aff} \triangleq \max_{i,j \in [L], i \neq j} \mathrm{aff}(i,j) \in [0,1].$$

There are $N = nL$ points, and there are n points exactly lying on each subspace. We assume that each data point y_i is drawn iid uniformly at random from $\mathbb{S}^{p-1} \cap \mathcal{D}_{w_i}$ where $\mathbb{S}^{p-1}$ is the unit sphere in $\mathbb{R}^p$. Equivalently,

$$y_i = D_{w_i} x_i, \quad x_i \sim \mathrm{Unif}(\mathbb{S}^{d-1}), \quad \forall i \in [N].$$

As the points are generated randomly on their corresponding subspaces, there are no points lying on an intersection of two subspaces, almost surely. This implies that *with probability one the points are clustered correctly provided that the true subspaces are recovered exactly.*

3.2 Main theorems

The first theorem gives a statistical guarantee for the fully random model.

Theorem 1 *Suppose L d-dimensional subspaces and n points on each subspace are generated in the fully random model with n polynomial in d. There are constants $C_1, C_2 > 0$ such that if*

$$\frac{n}{d} > C_1 \left(\log \frac{ne}{d\delta}\right)^2, \quad \frac{d}{p} < \frac{C_2 \log n}{\log(ndL\delta^{-1})}, \tag{1}$$

then with probability at least $1 - \frac{3L\delta}{1-\delta}$, NSN+GSR[3] clusters the points exactly. Also, there are other constants $C_1', C_2' > 0$ such that if (1) with C_1 and C_2 replaced by C_1' and C_2' holds then NSN+Spectral[4] clusters the points exactly with probability at least $1 - \frac{3L\delta}{1-\delta}$. e is the exponential constant.

[3] NSN with $K = k_{max} = d$ followed by GSR with arbitrarily small ϵ.

[4] NSN with $K = k_{max} = d$.

Our sufficient conditions for exact clustering explain when subspace clustering becomes easy or difficult, and they are consistent with our intuition. For NSN to find correct neighbors, the points on the same subspace should be many enough so that they look like lying on a subspace. This condition is spelled out in the first inequality of (1). We note that the condition holds even when n/d is a constant, i.e., n is linear in d. The second inequality implies that the dimension of the subspaces should not be too high for subspaces to be distinguishable. If d is high, the random subspaces are more likely to be close to each other, and hence they become more difficult to be distinguished. However, as n increases, the points become dense on the subspaces, and hence it becomes easier to identify different subspaces.

Let us compare our result with the conditions required for success in the fully random model in the existing literature. In [16], it is required for SSC to have correct neighborhoods that n should be superlinear in d when d/p fixed. In [6, 24], the conditions on d/p becomes worse as we have more points. On the other hand, our algorithms are guaranteed exact clustering of the points, and the sufficient condition is order-wise at least as good as the conditions for correct neighborhoods by the existing algorithms (See Table 1). Moreover, exact clustering is guaranteed even when n is linear in d, and d/p fixed.

For the semi-random model, we have the following general theorem.

Theorem 2 *Suppose L d-dimensional subspaces are arbitrarily chosen, and n points on each subspace are generated in the semi-random model with n polynomial in d. There are constants $C_1, C_2 > 0$ such that if*

$$\frac{n}{d} > C_1 \left(\log \frac{ne}{d\delta}\right)^2, \quad \max \mathrm{aff} < \sqrt{\frac{C_2 \log n}{\log(dL\delta^{-1}) \cdot \log(ndL\delta^{-1})}}. \tag{2}$$

then with probability at least $1 - \frac{3L\delta}{1-\delta}$, NSN+GSR[5] clusters the points exactly.

In the semi-random model, the sufficient condition does not depend on the ambient dimension p. When the affinities between subspaces are fixed, and the points are exactly lying on the subspaces, the difficulty of the problem does not depend on the ambient dimension. It rather depends on $\max \mathrm{aff}$, which measures how close the subspaces are. As they become closer to each other, it becomes more difficult to distinguish the subspaces. The second inequality of (2) explains this intuition. The inequality also shows that if we have more data points, the problem becomes easier to identify different subspaces.

Compared with other algorithms, NSN+GSR is guaranteed exact clustering, and more importantly, the condition on $\max \mathrm{aff}$ improves as n grows. This remark is consistent with the practical performance of the algorithm which improves as the number of data points increases, while the existing guarantees of other algorithms are not. In [16], correct neighborhoods in SSC are guaranteed if $\max \mathrm{aff} = O(\sqrt{\log(n/d)}/\log(nL))$. In [6], exact clustering of TSC is guaranteed if $\max \mathrm{aff} = O(1/\log(nL))$. However, these algorithms perform empirically better as the number of data points increases.

4 Experimental results

In this section, we empirically compare our algorithms with the existing algorithms in terms of clustering performance and computational time (on a single desktop). For NSN, we used the fast implementation described in Section A.1. The compared algorithms are K-means, K-flats[6], SSC, LRR, SCC, TSC[7], and SSC-OMP[8]. The numbers of replicates in K-means, K-flats, and the K-

[5] NSN with $K = d - 1$ and $k_{max} = \lceil 2 \log d \rceil$ followed by GSR with arbitrarily small ϵ.

[6] K-flats is similar to K-means. At each iteration, it computes top-d principal subspaces of the points with the same label, and then labels every point based on its distances to those subspaces.

[7] The MATLAB codes for SSC, LRR, SCC, and TSC are obtained from `http://www.cis.jhu.edu/~ehsan/code.htm`, `https://sites.google.com/site/guangcanliu`, and `http://www.math.duke.edu/~glchen/scc.html`, `http://www.nari.ee.ethz.ch/commth/research/downloads/sc.html`, respectively.

[8] For each data point, OMP constructs a neighborhood for each point by regressing the point on the other points up to 10^{-4} accuracy.

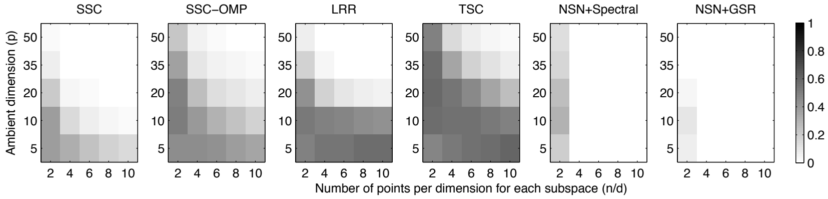

Figure 1: CE of algorithms on 5 random d-dimensional subspaces and n random points on each subspace. The figures shows CE for different numbers of n/d and ambient dimension p. d/p is fixed to be $3/5$. Brighter cells represent that less data points are clustered incorrectly.

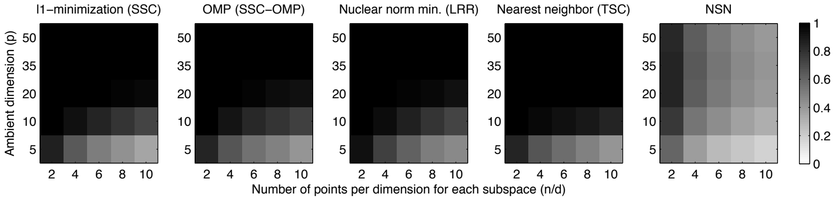

Figure 2: NSE for the same model parameters as those in Figure 1. Brighter cells represent that more data points have all correct neighbors.

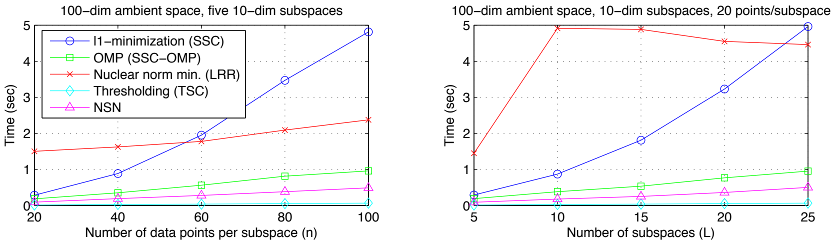

Figure 3: Average computational time of the neighborhood selection algorithms

means used in the spectral clustering are all fixed to 10. The algorithms are compared in terms of *Clustering error (CE)* and *Neighborhood selection error (NSE)*, defined as

$$(\text{CE}) = \min_{\pi \in \Pi_L} \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(w_i \neq \pi(\hat{w}_i)), \quad (\text{NSE}) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\exists j : W_{ij} \neq 0, w_i \neq w_j)$$

where Π_L is the permutation space of $[L]$. CE is the proportion of incorrectly labeled data points. Since clustering is invariant up to permutation of label indices, the error is equal to the minimum disagreement over the permutation of label indices. NSE measures the proportion of the points which do not have all correct neighbors.[9]

4.1 Synthetic data

We compare the performances on synthetic data generated from the fully random model. In $\mathbb{R}^p$, five d-dimensional subspaces are generated uniformly at random. Then for each subspace n unit-norm points are generated iid uniformly at random on the subspace. To see the agreement with the theoretical result, we ran the algorithms under fixed d/p and varied n and d. We set $d/p = 3/5$ so that each pair of subspaces has intersection. Figures 1 and 2 show CE and NSE, respectively. Each error value is averaged over 100 trials. Figure 1 indicates that our algorithm clusters the data points better than the other algorithms. As predicted in the theorems, the clustering performance improves

[9] For the neighborhood matrices from SSC, LRR, and SSC-OMP, the d points with the maximum weights are regarded as neighbors for each point. For TSC, the d nearest neighbors are collected for each point.

L	Algorithms	K-means	K-flats	SSC	LRR	SCC	SSC-OMP(8)	TSC(10)	NSN+Spectral(5)
2	Mean CE (%)	19.80	13.62	1.52	2.13	2.06	16.92	18.44	3.62
	Median CE (%)	17.92	10.65	0.00	0.00	0.00	12.77	16.92	0.00
	Avg. Time (sec)	-	0.80	3.03	3.42	1.28	0.50	0.50	0.25
3	Mean CE (%)	26.10	14.07	4.40	4.03	6.37	27.96	28.58	8.28
	Median CE (%)	20.48	14.18	0.56	1.43	0.21	30.98	29.67	2.76
	Avg. Time (sec)	-	1.89	5.39	4.05	2.16	0.82	1.15	0.51

Table 2: CE and computational time of algorithms on Hopkins155 dataset. L is the number of clusters (motions). The numbers in the parentheses represent the number of neighbors for each point collected in the corresponding algorithms.

L	Algorithms	K-means	K-flats	SSC	SSC-OMP	TSC	NSN+Spectral
2	Mean CE (%)	45.98	37.62	1.77	4.45	11.84	1.71
	Median CE (%)	47.66	39.06	0.00	1.17	1.56	0.78
	Avg. Time (sec)	-	15.78	37.72	0.45	0.33	0.78
3	Mean CE (%)	62.55	45.81	5.77	6.35	20.02	3.63
	Median CE (%)	63.54	47.92	1.56	2.86	15.62	3.12
	Avg. Time (sec)	-	27.91	49.45	0.76	0.60	3.37
5	Mean CE (%)	73.77	55.51	4.79	8.93	11.90	5.81
	Median CE (%)	74.06	56.25	2.97	5.00	33.91	4.69
	Avg. Time (sec)	-	52.90	74.91	1.41	1.17	5.62
10	Mean CE (%)	82.68	62.72	9.43	15.32	39.48	9.82
	Median CE (%)	82.97	62.89	8.75	17.11	39.45	9.06
	Avg. Time (sec)	-	134.0	157.5	5.26	3.17	14.73

Table 3: CE and computational time of algorithms on Extended Yale B dataset. For each number of clusters (faces) L, the algorithms ran over 100 random subsets drawn from the overall 38 clusters.

as the number of points increases. However, it also improves as the dimension of subspaces grows in contrast to the theoretical analysis. We believe that this is because our analysis on GSR is not tight. In Figure 2, we can see that more data points obtain correct neighbors as n increases or d decreases, which conforms the theoretical analysis.

We also compare the computational time of the neighborhood selection algorithms for different numbers of subspaces and data points. As shown in Figure 3, the greedy algorithms (OMP, Thresholding, and NSN) are significantly more scalable than the convex optimization based algorithms (ℓ_1-minimization and nuclear norm minimization).

4.2 Real-world data : motion segmentation and face clustering

We compare our algorithm with the existing ones in the applications of motion segmentation and face clustering. For the motion segmentation, we used Hopkins155 dataset [17], which contains 155 video sequences of 2 or 3 motions. For the face clustering, we used Extended Yale B dataset with cropped images from [5, 13]. The dataset contains 64 images for each of 38 individuals in frontal view and different illumination conditions. To compare with the existing algorithms, we used the set of 48×42 resized raw images provided by the authors of [4]. The parameters of the existing algorithms were set as provided in their source codes.[10] Tables 2 and 3 show CE and average computational time.[11] We can see that NSN+Spectral performs competitively with the methods with the lowest errors, but much faster. Compared to the other greedy neighborhood construction based algorithms, SSC-OMP and TSC, our algorithm performs significantly better.

Acknowledgments

The authors would like to acknowledge NSF grants 1302435, 0954059, 1017525, 1056028 and DTRA grant HDTRA1-13-1-0024 for supporting this research. This research was also partially supported by the U.S. Department of Transportation through the Data-Supported Transportation Operations and Planning (D-STOP) Tier 1 University Transportation Center.

[10] As SSC-OMP and TSC do not have proposed number of parameters for motion segmentation, we found the numbers minimizing the mean CE. The numbers are given in the table.

[11] The LRR code provided by the author did not perform properly with the face clustering dataset that we used. We did not run NSN+GSR since the data points are not well distributed in its corresponding subspaces.

References

[1] P. S. Bradley and O. L. Mangasarian. K-plane clustering. *Journal of Global Optimization*, 16(1):23–32, 2000.

[2] G. Chen and G. Lerman. Spectral curvature clustering. *International Journal of Computer Vision*, 81(3): 317–330, 2009.

[3] E. L. Dyer, A. C. Sankaranarayanan, and R. G. Baraniuk. Greedy feature selection for subspace clustering. *The Journal of Machine Learning Research (JMLR)*, 14(1):2487–2517, 2013.

[4] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2765–2781, 2013.

[5] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23 (6):643–660, 2001.

[6] R. Heckel and H. Bölcskei. Subspace clustering via thresholding and spectral clustering. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2013.

[7] R. Heckel and H. Bölcskei. Robust subspace clustering via thresholding. *arXiv preprint arXiv:1307.4891v2*, 2014.

[8] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[9] T. Inglot. Inequalities for quantiles of the chi-square distribution. *Probability and Mathematical Statistics*, 30(2):339–351, 2010.

[10] H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(1):1, 2009.

[11] S. Kunis and H. Rauhut. Random sampling of sparse trigonometric polynomials, ii. orthogonal matching pursuit versus basis pursuit. *Foundations of Computational Mathematics*, 8(6):737–763, 2008.

[12] M. Ledoux. *The concentration of measure phenomenon*, volume 89. AMS Bookstore, 2005.

[13] K. C. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 27(5):684–698, 2005.

[14] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):171–184, 2013.

[15] V. D. Milman and G. Schechtman. *Asymptotic Theory of Finite Dimensional Normed Spaces: Isoperimetric Inequalities in Riemannian Manifolds*. Lecture Notes in Mathematics. Springer, 1986.

[16] M. Soltanolkotabi and E. J. Candes. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 40(4):2195–2238, 2012.

[17] R. Tron and R. Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[18] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666, 2007.

[19] P. Tseng. Nearest q-flat to m points. *Journal of Optimization Theory and Applications*, 105(1):249–252, 2000.

[20] R. Vidal. Subspace clustering. *Signal Processing Magazine, IEEE*, 28(2):52–68, 2011.

[21] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[22] R. Vidal, S. Soatto, Y. Ma, and S. Sastry. An algebraic geometric approach to the identification of a class of linear hybrid systems. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 1, pages 167–172. IEEE, 2003.

[23] R. Vidal, R. Tron, and R. Hartley. Multiframe motion segmentation with missing data using power factorization and GPCA. *International Journal of Computer Vision*, 79(1):85–105, 2008.

[24] Y.-X. Wang, H. Xu, and C. Leng. Provable subspace clustering: When LRR meets SSC. In *Advances in Neural Information Processing Systems (NIPS)*, December 2013.

[25] A. Y. Yang, S. R. Rao, and Y. Ma. Robust statistical estimation and segmentation of multiple subspaces. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[26] T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Hybrid linear modeling via local best-fit flats. *International journal of computer vision*, 100(3):217–240, 2012.

A Discussion on implementation issues

A.1 A faster implementation for NSN

At each step of NSN, the algorithm computes the projections of all points onto a subspace and find one with the largest norm. A naive implementation of the algorithm requires $O(pK^2N^2)$ time complexity.

In fact, we can reduce the complexity to $O(pKN^2)$. Instead of finding the maximum norm of the projections, we can find the maximum squared norm of the projections. Let $\mathcal{U}_k$ be the subspace $\mathcal{U}$ at step k. For any data point y, we have

$$\|\mathrm{Proj}_{\mathcal{U}_k} y\|_2^2 = \|\mathrm{Proj}_{\mathcal{U}_{k-1}} y\|_2^2 + |u_k^\top y|^2$$

where u_k is the new orthogonal axis added from $\mathcal{U}_{k-1}$ to make $\mathcal{U}_k$. That is, $\mathcal{U}_{k-1} \perp u_k$ and $\mathcal{U}_k = \mathcal{U}_{k-1} \bigoplus u_k$. As $\|\mathrm{Proj}_{\mathcal{U}_{k-1}} y\|_2^2$ is already computed in the $(k-1)$'th step, we do not need to compute it again at step k. Based on this fact, we have a faster implementation as described in the following. Note that P_j at the kth step is equal to $\|\mathrm{Proj}_{\mathcal{U}_k} y_j\|_2^2$ in the original NSN algorithm.

Algorithm 3 Fast Nearest Subspace Neighbor (F-NSN)

```
Input: A set of $N$ samples $\mathcal{Y} = \{y_1, \ldots, y_N\}$, The number of required neighbors $K$, Maximum
  subspace dimension $k_{max}$.
Output: A neighborhood matrix $W \in \{0,1\}^{N \times N}$
  $y_i \leftarrow y_i/\|y_i\|_2, \forall i \in [N]$
  for $i = 1, \ldots, N$ do
    $\mathcal{I}_i \leftarrow \{i\}, u_1 \leftarrow y_i$
    $P_j \leftarrow 0, \forall j \in [N]$
    for $k = 1, \ldots, K$ do
      if $k \le k_{max}$ then
        $P_j \leftarrow P_j + \|u_k^\top y_j\|^2, \ \forall j \in [N]$
      end if
      $j^* \leftarrow \arg\max_{j \in [N], j \notin \mathcal{I}_i} P_j$
      $\mathcal{I}_i \leftarrow \mathcal{I}_i \cup \{j^*\}$
      if $k < k_{max}$ then
        $u_{k+1} \leftarrow \frac{y_{j^*} - \sum_{l=1}^k (u_l^\top y_{j^*}) u_l}{\|y_{j^*} - \sum_{l=1}^k (u_l^\top y_{j^*}) u_l\|_2}$
      end if
    end for
    $W_{ij} \leftarrow \mathbb{I}_{j \in \mathcal{I}_i \text{ or } P_j = 1}, \ \forall j \in [N]$
  end for
```

A.2 Estimation of the number of clusters

When L is unknown, it can be estimated at the clustering step. For Spectral clustering, a well-known approach to estimate L is to find a knee point in the singular values of the neighborhood matrix. It is the point where the difference between two consecutive singular values are the largest. For GSR, we do not need to estimate the number of clusters a priori. Once the algorithms finishes, the number of the resulting groups will be our estimate of L.

A.3 Parameter setting

The choices of K and $k_{\max}$ depend on the dimension of the subspaces d. If data points are lying exactly on the model subspaces, $K = k_{\max} = d$ is enough for GSR to recover a subspace. In practical situations where the points are near the subspaces, it is better to set K to be larger than d. $k_{\max}$ can also be larger than d because the $k_{\max} - d$ additional dimensions, which may be induced from the noise, do no intersect with the other subspaces in practice. For Extended Yale B dataset and Hopkins155 dataset, we found that NSN+Spectral performs well if K is set to be around $2d$.

B Proofs

B.1 Proof outline

We describe the first few high-level steps in the proofs of our main theorems. Exact clustering of our algorithms depends on whether NSN can find all correct neighbors for the data points so that the following algorithm (GSR or Spectral clustering) can cluster the points exactly. For NSN+GSR, exact clustering is guaranteed when there is a point on each subspace that have all correct neighbors which are at least $d-1$. For NSN+Spectral, exact clustering is guaranteed when each data point has only the $n-1$ other points on the same subspace as neighbors. In the following, we explain why these are true.

Step 1-1: Exact clustering condition for GSR

The two statistical models have a property that for any d-dimensional subspace in $\mathbb{R}^p$ other than the true subspaces $\mathcal{D}_1, \dots, \mathcal{D}_L$ the probability of any points lying on the subspace is zero. Hence, we claim the following.

Fact 3 (Best d-dimensional fit) *With probability one, the true subspaces $\mathcal{D}_1, \dots, \mathcal{D}_L$ are the L subspaces containing the most points among the set of all possible d-dimensional subspaces.*

Then it suffices for each subspace to have one point whose neighbors are $d-1$ all correct points on the same subspace. This is because the subspace spanned by those d points is almost surely identical to the true subspace they are lying on, and that subspace will be picked by GSR.

Fact 4 *If NSN with $K \geq d-1$ finds all correct neighbors for at least one point on each subspace, GSR recovers all the true subspaces and clusters the data points exactly with probability one.*

In the following steps, we consider one data point for each subspace. We show that NSN with $K = k_{max} = d$ finds all correct neighbors for the point with probability at least $1 - \frac{3\delta}{1-\delta}$. Then the union bound and Fact 4 establish exact clustering with probability at least $1 - \frac{3L\delta}{1-\delta}$.

Step 1-2: Exact clustering condition for spectral clustering

It is difficult to analyze spectral clustering for the resulting neighborhood matrix of NSN. A trivial case for a neighborhood matrix to result in exact clustering is when the points on the same subspace form a single fully connected component. If NSN with $K = k_{max} = d$ finds all correct neighbors for every data point, the subspace $\mathcal{U}$ at the last step ($k = K$) is almost surely identical to the true subspace that the points lie on. Hence, the resulting neighborhood matrix W form L fully connected components each of which contains all of the points on the same subspace.

In the rest of the proof, we show that if (1) holds, NSN finds all correct neighbors for a fixed point with probability $1 - \frac{3\delta}{1-\delta}$. Let us assume that this is true. If (1) with C_1 and C_2 replaced by $\frac{C_1}{4}$ and $\frac{C_2}{2}$ holds, we have

$$n > C_1 d \left(\log \frac{ne}{d(\delta/n)} \right)^2, \quad \frac{d}{p} < \frac{C_2 \log n}{\log(ndL(\delta/n)^{-1})}.$$

Then it follows from the union bound that NSN finds all correct neighbors for all of the n points on each subspace with probability at least $1 - \frac{3L\delta}{1-\delta}$, and hence we obtain $W_{ij} = \mathbb{I}_{w_i = w_j}$ for every $(i,j) \in [N]^2$. Exact clustering is guaranteed.

Step 2: Success condition for NSN

Now the only proposition that we need to prove is that for each subspace $\mathcal{D}_i$ NSN finds all correct neighbors for a data point (which is a uniformly random unit vector on the subspace) with probability at least $1 - \frac{3\delta}{1-\delta}$. As our analysis is independent of the subspaces, we only consider $\mathcal{D}_1$. Without loss of generality, we assume that y_1 lies on $\mathcal{D}_1$ ($w_1 = 1$) and focus on this data point.

When NSN finds neighbors of y_1, the algorithm constructs k_{max} subspaces incrementally. At each step $k = 1, \ldots, K$, if the largest projection onto $\mathcal{U}$ of the uncollected points on the same true subspace $\mathcal{D}_1$ is greater than the largest projection among the points on different subspaces, then NSN collects a correct neighbor. In a mathematical expression, we want to satisfy

$$\max_{j:w_j=1, j\notin\mathcal{I}_1} \|\text{Proj}_{\mathcal{U}} y_j\|_2 > \max_{j:w_j\neq 1, j\notin\mathcal{I}_1} \|\text{Proj}_{\mathcal{U}} y_j\|_2 \tag{3}$$

for each step of $k = 1, \ldots, K$.

The rest of the proof is to show (1) and (2) lead to (3) with probability $1 - \frac{3\delta}{1-\delta}$ in their corresponding models. It is difficult to prove (3) itself because the subspaces, the data points, and the index set $\mathcal{I}_1$ are all dependent of each other. Instead, we introduce an Oracle algorithm whose success is equivalent to the success of NSN, but the analysis is easier. Then the Oracle algorithm is analyzed using stochastic ordering, bounds on order statistics of random projections, and the measure concentration inequalities for random subspaces. The rest of the proof is provided in Sections B.3 and B.4.

B.2 Preliminary lemmas

Before we step into the technical parts of the proof, we introduce the main ingredients which will be used. The following lemma is about upper and lower bounds on the order statistics for the projections of iid uniformly random unit vectors.

Lemma 5 *Let $x_1, \ldots, x_n$ be drawn iid uniformly at random from the d-dimensional unit ball $\mathbb{S}^{d-1}$. Let $z_{(n-m+1)}$ denote the m'th largest value of $\{z_i \triangleq \|Ax_i\|_2, 1 \le i \le n\}$ where $A \in \mathbb{R}^{k\times d}$.*

a. Suppose that the rows of A are orthonormal to each other. For any $\alpha \in (0,1)$, there exists a constant $C > 0$ such that for $n, m, d, k \in \mathbb{N}$ where

$$n - m + 1 \ge Cm\left(\log \frac{ne}{m\delta}\right)^2 \tag{4}$$

we have

$$z^2_{(n-m+1)} > \frac{k}{d} + \frac{1}{d}\cdot \min\left\{2\log\left(\frac{n-m+1}{Cm\left(\log\frac{ne}{m\delta}\right)^2}\right), \alpha\sqrt{d-k}\right\} \tag{5}$$

with probability at least $1 - \delta^m$.

b. For any $k \times d$ matrix A,

$$z_{(n-m+1)} < \frac{\|A\|_F}{\sqrt{d}} + \frac{\|A\|_2}{\sqrt{d}}\cdot\left(\sqrt{2\pi} + \sqrt{2\log\frac{ne}{m\delta}}\right) \tag{6}$$

with probability at least $1 - \delta^m$.

Lemma 5b can be proved by using the measure concentration inequalities [12]. Not only can they provide inequalities for random unit vectors, they also give us inequalities for random subspaces.

Lemma 6 *Let the columns of $X \in \mathbb{R}^{d\times k}$ be the orthonormal basis of a k-dimensional random subspace drawn uniformly at random in d-dimensional space.*

a. For any matrix $A \in \mathbb{R}^{p\times d}$.

$$E[\|AX\|_F^2] = \frac{k}{d}\|A\|_F^2$$

b. [15, 12] If $\|A\|_2$ is bounded, then we have

$$\Pr\left\{\|AX\|_F > \sqrt{\frac{k}{d}}\|A\|_F + \|A\|_2\cdot\left(\sqrt{\frac{8\pi}{d-1}} + t\right)\right\} \le e^{-\frac{(d-1)t^2}{8}}.$$

B.3 Proof of Theorem 2

Following Section B.1, we show in this section that if (2) holds then NSN finds all correct neighbors for y_1 (which is assumed to be on $\mathcal{D}_1$) with probability at least $1 - \frac{3\delta}{1-\delta}$.

Step 3: NSN Oracle algorithm

Consider the Oracle algorithm in the following. Unlike NSN, this algorithm knows the true label of each data point. It picks the point closest to the current subspace among the points with the same label. Since we assume $w_1 = 1$, the Oracle algorithm for y_1 picks a point in $\{y_j : w_j = 1\}$ at every step.

Algorithm 4 NSN Oracle algorithm for y_1 (assuming $w_1 = 1$)

```
Input: A set of $N$ samples $\mathcal{Y} = \{y_1, \dots, y_N\}$, The number of required neighbors $K = d - 1$,
  Maximum subspace dimension $k_{max} = \lceil 2 \log d \rceil$
  $\mathcal{I}_1^{(1)} \leftarrow \{1\}$
  for $k = 1, \dots, K$ do
      if $k \leq k_{max}$ then
          $\mathcal{V}_k \leftarrow \text{span}\{y_j : j \in \mathcal{I}_1^{(k)}\}$
          $j_k^* \leftarrow \arg\max_{j \in [N] : w_j = 1, j \notin \mathcal{I}_1^{(k)}} \|\text{Proj}_{\mathcal{V}_k} y_j\|_2$
      else
          $j_k^* \leftarrow \arg\max_{j \in [N] : w_j = 1, j \notin \mathcal{I}_1^{(k)}} \|\text{Proj}_{\mathcal{V}_{k_{max}}} y_j\|_2$
      end if
      if $\max_{j \in [N] : w_j = 1, j \notin \mathcal{I}_i^{(k)}} \|\text{Proj}_{\mathcal{V}_k} y_j\|_2 \leq \max_{j \in [N] : w_j \neq 1} \|\text{Proj}_{\mathcal{V}_k} y\|_2$ then
          Return FAILURE
      end if
      $\mathcal{I}_1^{(k+1)} \leftarrow \mathcal{I}_1^{(k)} \cup \{j_k^*\}$
  end for
  Return SUCCESS
```

Note that the Oracle algorithm returns failure if and only if the original algorithm picks an incorrect neighbor for y_1. The reason is as follows. Suppose that NSN for y_1 picks the first incorrect point at step k. By the step $k-1$, correct points have been chosen because they are the nearest points for the subspaces in the corresponding steps. The Oracle algorithm will also pick those points because they are the nearest points among the correct points. Hence $\mathcal{U} \equiv \mathcal{V}_k$. At step k, NSN picks an incorrect point as it is the closest to $\mathcal{U}$. The Oracle algorithm will declare failure because that incorrect point is closer than the closest point among the correct points. In the same manner, we see that NSN fails if the Oracle NSN fails. Therefore, we can instead analyze the success of the Oracle algorithm. The success condition is written as

$$\begin{aligned} \|\text{Proj}_{\mathcal{V}_k} y_{j_k^*}\|_2 &> \max_{j \in [N] : w_j \neq 1} \|\text{Proj}_{\mathcal{V}_k} y\|_2, \quad \forall k = 1, \dots, k_{max}, \\ \|\text{Proj}_{\mathcal{V}_{k_{max}}} y_{j_k^*}\|_2 &> \max_{j \in [N] : w_j \neq 1} \|\text{Proj}_{\mathcal{V}_{k_{max}}} y\|_2, \quad \forall k = k_{max} + 1, \dots, K. \end{aligned} \tag{7}$$

Note that $\mathcal{V}_k$'s are independent of the points $\{y_j : j \in [N], w_j \neq 1\}$. We will use this fact in the following steps.

Step 4: Lower bounds on the projection of correct points (the LHS of (7))

Let $V_k \in \mathbb{R}^{d \times k}$ be such that the columns of $D_1 V_k$ form an orthogonal basis of $\mathcal{V}_k$. Such a V_k exists because $\mathcal{V}_k$ is a k-dimensional subspace of $\mathcal{D}_1$. Then we have

$$\|\text{Proj}_{\mathcal{V}_k} y_{j_k^*}\|_2 = \|V_k^\top D_1^\top D_1 x_{j_k^*}\|_2 = \|V_k^\top x_{j_k^*}\|_2$$

In this step, we obtain lower bounds on $\|V_k^\top x_{j_k^*}\|_2$ for $k \leq k_{max}$ and $\|V_{k_{max}}^\top x_{j_k^*}\|_2$ for $k > k_{max}$.

It is difficult to analyze $\|V_k^\top x_{j_k^*}\|_2$ because V_k and $x_{j_k^*}$ are dependent. We instead analyze another random variable that is stochastically dominated by $\|V_k^\top x_{j_k^*}\|_2^2$. Then we use a high-probability lower bound on that variable which also lower bounds $\|V_k^\top x_{j_k^*}\|_2^2$ with high probability.

Define $P_{k,(m)}$ as the m'th largest norm of the projections of $n-1$ iid uniformly random unit vector on $\mathbb{S}^{d-1}$ onto a k-dimensional subspace. Since the distribution of the random unit vector is isotropic, the distribution of $P_{k,(m)}$ is identical for any k-dimensional subspaces independent of the random unit vectors. We have the following key lemma.

Lemma 7 $\|V_k^\top x_{j_k^*}\|_2$ *stochastically dominates* $P_{k,(k)}$*, i.e.,*

$$\Pr\{\|V_k^\top x_{j_k^*}\|_2 \geq t\} \geq \Pr\{P_{k,(k)} \geq t\}$$

for any $t \geq 0$*. Moreover,* $P_{k,(k)} \geq P_{\hat{k},(k)}$ *for any* $\hat{k} \leq k$*.*

The proof of the lemma is provided in Appendix B.5. Now we can use the lower bound on $P_{k,(k)}$ given in Lemma 5a to bound $\|V_k^\top x_{j_k^*}\|_2$. Let us pick α and C for which the lemma holds. The first inequality of (2) with $C_1 = C+1$ leads to $n-d > Cd\left(\log\frac{ne}{d\delta}\right)^2$, and also

$$n-k > Ck\left(\log\frac{ne}{k\delta}\right)^2, \quad \forall k = 1,\dots,d-1. \tag{8}$$

Hence, it follow from Lemma 5a that for each $k=1,\dots,k_{max}$, we have

$$\begin{aligned}\|V_k^\top x_{j_k^*}\|_2 &\geq \frac{k}{d} + \frac{1}{d}\min\left\{2\log\left(\frac{n-k+1}{Ck\left(\log\frac{ne}{k\delta}\right)^2}\right), \alpha\sqrt{d-k}\right\}\\ &\geq \frac{k}{d} + \frac{1}{d}\min\left\{2\log\left(\frac{n-d}{Cd\left(\log\frac{ne}{\delta}\right)^2}\right), \alpha\sqrt{d-2\log d}\right\}\end{aligned} \tag{9}$$

with probability at least $1-\delta^k$.

For $k > k_{max}$, we want to bound $\|\mathrm{Proj}_{\mathcal{V}_{k_{max}}} y_{j_k^*}\|_2$. We again use Lemma 7 to obtain the bound. Since the condition for the lemma holds as shown in (8), we have

$$\begin{aligned}\|V_{k_{max}}^\top x_{j_k^*}\|_2 &\geq \frac{2\log d}{d} + \frac{1}{d}\min\left\{2\log\left(\frac{n-k+1}{Ck\left(\log\frac{ne}{k\delta}\right)^2}\right), \alpha\sqrt{d-2\log d}\right\}\\ &\geq \frac{2\log d}{d} + \frac{1}{d}\min\left\{2\log\left(\frac{n-d}{Cd\left(\log\frac{ne}{\delta}\right)^2}\right), \alpha\sqrt{d-2\log d}\right\}\end{aligned} \tag{10}$$

with probability at least $1-\delta^k$, for every $k = k_{max}+1,\dots,d-1$.

The union bound gives that (9) and (10) hold for all $k = 1,\dots,d-1$ simultaneously with probability at least $1-\frac{\delta}{1-\delta}$.

Step 5: Upper bounds on the projection of incorrect points (the RHS of (7))

Since we have $\|\mathrm{Proj}_{\mathcal{V}_k} y_j\|_2 = \|V_k^\top D_1^\top D_{w_j} x_j\|_2$, the RHS of (7) can be written as

$$\max_{j:j\in[N],w_j\neq 1} \|V_k^\top D_1^\top D_{w_j} x_j\|_2 \tag{11}$$

In this step, we want to bound (11) for every $k=1,\dots,d-1$ by using the concentration inequalities for V_k and x_j. Since V_k and x_j are independent, the inequality for x_j holds for any fixed V_k.

It follows from Lemma 5b and the union bound that with probability at least $1-\delta$,

$$\begin{aligned}
&\max_{j:j\in[N],w_j\neq 1} \|V_k^\top D_1^\top D_{w_j} x_j\|_2 \\
&\le \frac{\max_{l\neq 1}\|V_k^\top D_1^\top D_l\|_F}{\sqrt{d}} + \frac{\max_{l\neq 1}\|V_k^\top D_1^\top D_l\|_2}{\sqrt{d}} \cdot \left(\sqrt{2\pi} + \sqrt{2\log\frac{n(L-1)e}{\delta/d}}\right) \\
&\le \frac{\max_{l\neq 1}\|V_k^\top D_1^\top D_l\|_F}{\sqrt{d}} \cdot \left(5 + \sqrt{2\log\frac{ndL}{\delta}}\right)
\end{aligned}$$

for all $k = 1,\dots,d-1$. The last inequality follows from the fact $\|V_k^\top D_1^\top D_l\|_2 \le \|V_k^\top D_1^\top D_l\|_F$. Since $\|V_k^\top D_1^\top D_{w_j} x_j\|_2 \le \|V_k^\top D_1^\top D_{w_j}\|_F \le \max_{l\neq 1}\|V_k^\top D_1^\top D_l\|_F$ for every j such that $w_j \neq 1$, we have

$$\max_{j:j\in[N],w_j\neq 1} \|V_k^\top D_1^\top D_{w_j} x_j\|_2 \le \frac{\max_{l\neq 1}\|V_k^\top D_1^\top D_l\|_F}{\sqrt{d}} \cdot \min\left\{5 + \sqrt{2\log\frac{ndL}{\delta}}, \sqrt{d}\right\}. \quad (12)$$

Now let us consider $\max_{l\neq 1}\|V_k^\top D_1^\top D_l\|_F$. In our statistical model, the new axis added to $\mathcal{V}_k$ at the kth step (u_{k+1} in Algorithm 3) is chosen uniformly at random from the subspace in $\mathcal{D}_1$ orthogonal to $\mathcal{V}_k$. Therefore, V_k is a random matrix drawn uniformly from the $d \times k$ Stiefel manifold, and the probability measure is the normalized Haar (rotation-invariant) measure. From Lemma 6b and the union bound, we obtain that with probability at least $1 - \delta/dL$,

$$\begin{aligned}
\|V_k^\top D_1^\top D_l\|_F &\le \sqrt{\frac{k}{d}}\|D_1^\top D_l\|_F + \|D_1^\top D_l\|_2 \cdot \left(\sqrt{\frac{8\pi}{d-1}} + \sqrt{\frac{8}{d-1}\log\frac{dL}{\delta}}\right) \\
&\le \|D_1^\top D_l\|_F \cdot \left(\sqrt{\frac{k}{d}} + \sqrt{\frac{8\pi}{d-1}} + \sqrt{\frac{8}{d-1}\log\frac{dL}{\delta}}\right) \\
&\le \max \operatorname{aff} \cdot \sqrt{d} \cdot \left(\sqrt{\frac{k}{d}} + \sqrt{\frac{8\pi}{d-1}} + \sqrt{\frac{8}{d-1}\log\frac{dL}{\delta}}\right). \qquad (13)
\end{aligned}$$

The union bound gives that with probability at least $1-\delta$, $\max_{l\neq 1}\|V_k^\top D_1^\top D_l\|_F$ is also bounded by (13) for every $k = 1,\dots,k_{max}$.

Putting (13) and (12) together, we obtain

$$\begin{aligned}
&\max_{j:j\in[N],w_j\neq 1} \|V_k^\top D_1^\top D_{w_j} x_j\|_2 \\
&\le \max \operatorname{aff} \cdot \left(\sqrt{\frac{k}{d}} + \sqrt{\frac{8\pi}{d-1}} + \sqrt{\frac{8}{d-1}\log\frac{dL}{\delta}}\right) \cdot \min\left\{5 + \sqrt{2\log\frac{ndL}{\delta}}, \sqrt{d}\right\} \qquad (14)
\end{aligned}$$

for all $k = 1,\dots,d-1$ with probability at least $1-2\delta$.

Final Step: Proof of the main theorem

Putting (9), (10), and (14) together, we obtain that if

$$\max \operatorname{aff} < \min_{1\le k\le d-1} \frac{\sqrt{\min\{k, 2\log d\} + \min\left\{2\log\left(\frac{n-d}{Cd}\right) - 4\log\log\frac{ne}{\delta}, \alpha\sqrt{d-2\log d}\right\}}}{\left(\sqrt{\min\{k, 2\log d\}} + \sqrt{\frac{8\pi d}{d-1}} + \sqrt{\frac{8d}{d-1}\log\frac{dL}{\delta}}\right) \cdot \min\left\{5 + \sqrt{2\log\frac{ndL}{\delta}}, \sqrt{d}\right\}}, \quad (15)$$

then (7) holds, and hence NSN finds all correct neighbors for y_1 with probability at least $1 - \frac{3\delta}{1-\delta}$. The RHS of (15) is minimized when $k \ge 2\log d$, and consequently the condition (15) is equivalent to

$$\max \operatorname{aff} < \frac{\sqrt{2\log d + \min\left\{2\log\left(\frac{n-d}{Cd}\right) - 4\log\log\frac{ne}{\delta}, \alpha\sqrt{d-2\log d}\right\}}}{\left(\sqrt{2\log d} + \sqrt{\frac{8\pi d}{d-1}} + \sqrt{\frac{8d}{d-1}\log\frac{dL}{\delta}}\right) \cdot \min\left\{5 + \sqrt{2\log\frac{ndL}{\delta}}, \sqrt{d}\right\}}. \quad (16)$$

As n is polynomial in d, there is a constant $C_3 > 0$ such that

$$\text{(RHS of (16))} > \frac{C_3\sqrt{\log(n-d) - \log\log\frac{ne}{\delta}}}{\sqrt{\log\frac{dL}{\delta}\cdot\log\frac{ndL}{\delta}}}$$

This completes the proof.

B.4 Proof of Theorem 1

As we did in Section B.3, we prove in this section that if (1) holds then NSN finds all correct neighbors for y_1 with probability at least $1 - \frac{3\delta}{1-\delta}$.

The only difference between the semi-random model and the fully random model is the statistical dependence between subspaces. We can follow Step 3 in Section B.3 because they do not take any statistical dependence between subspaces into account. We assert that (7) is the success condition also for the fully random model. However, as $K = k_{max} = d$, there is no case where $k > k_{max}$ in this proof.

Now we provide a new proof of the last three steps for the fully random model.

Step 4: Lower bounds on the projection of correct points (the LHS of (7))

We again use Lemma 7. For $k > d/2$, we use the fact that $\|V_k^\top x_{j_k^*}\|_2$ stochastically dominates $P_{\lfloor d/2\rfloor,(k)}$. Then it follows from Lemma 5a that

$$\|V_k^\top x_{j_k^*}\|_2 \geq \frac{k}{2d} + \frac{1}{d}\min\left\{2\log\left(\frac{n-k+1}{Ck\left(\log\frac{ne}{k\delta}\right)^2}\right), \alpha\sqrt{d/2}\right\} \tag{17}$$

for all $k = 1, \ldots, d-1$ simultaneously with probability at least $1 - \frac{\delta}{1-\delta}$.

Step 5: Upper bounds on the projection of incorrect points (the RHS of (7))

We again use the notion of $X_k \in \mathbb{R}^{d\times k}$ which is defined in the proof of Theorem 2. Since $\|\mathrm{Proj}_{\mathcal{V}_k} y_j\|_2 = \|V_k^\top D_1^\top y_j\|_2$, the RHS of (7) can be written as

$$\max_{j:j\in[N], w_j\neq 1} \|V_k^\top D_1^\top y_j\|_2 \tag{18}$$

Since the true subspaces are independent of each other, y_j with $w_j \neq 1$ is also independent of D_1 and V_k, and its marginal distribution is uniform over $\mathbb{S}^{p-1}$. It follows from Lemma 5b that with probability at least $1 - \delta/d$,

$$\begin{aligned}(18) &\leq \frac{\|V_k^\top D_1^\top\|_F}{\sqrt{p}} + \frac{\|V_k^\top D_1^\top\|_2}{\sqrt{p}}\cdot\sqrt{2\log\frac{n(L-1)e}{\delta/d}} \\ &\leq \sqrt{\frac{k}{p}} + \sqrt{\frac{2}{p}\log\frac{ndLe}{\delta}}. \end{aligned}\tag{19}$$

The last inequality is obtained using the facts $\|D_1V_k\|_F = \sqrt{k}$ and $\|D_1V_k\|_2 \leq 1$. The union bound provides that (19) holds for every $k = 1, \ldots, d-1$ with probability at least $1 - \delta$.

Final Step: Proof of the main theorem

Now it suffices to show that (17) > (19) for every $k = 1, 2, \ldots, d-1$, i.e.,

$$\sqrt{\frac{k}{2d} + \frac{1}{d}\min\left\{2\log\left(\frac{n-k+1}{Ck\left(\log\frac{ne}{k\delta}\right)^2}\right), \alpha\sqrt{\frac{d}{2}}\right\}} > \sqrt{\frac{k}{p}} + \sqrt{\frac{2}{p}\log\frac{ndLe}{\delta}}, \quad \forall k = 1, 2, \ldots, d-1. \tag{20}$$

where α, C are the constants described in Lemma 5a. (20) is equivalent to

$$\frac{d}{p} < \min_{1 \le k \le d-1} \frac{k/2 + \min\left\{2\log\left(\frac{n-k+1}{Ck}\right) - 4\log\left(\log\frac{ne}{k\delta}\right), \alpha\sqrt{d/2}\right\}}{\left(\sqrt{k} + \sqrt{2\log(ndL\delta^{-1}e)}\right)^2}. \tag{21}$$

As n is polynomial in d, the numerator can be replaced by $O(k + \log(n-k+1))$. The RHS is minimized when $k = O(\log(ndL\delta^{-1}))$. Hence, the above condition is satisfied if (1) holds.

B.5 Proof of Lemma 7

We construct a generative model for two random variables that are equal in distribution to $\|V_k^\top x_{j_k^*}\|_2^2$ and $P_{k,(k)}^2$. Then we show that the one corresponding to $\|V_k^\top x_{j_k^*}\|_2^2$ is greater than the other corresponding to $P_{k,(k)}^2$. This generative model uses the fact that for any isotropic distributions the marginal distributions of the components along any orthogonal axes are invariant.

The generative model is given as follows.

1. For $k = 1, \ldots, k_{max}$, repeat 2.
2. Draw $n-1$ iid random variables $Y_1^{(k)}, \ldots, Y_{n-1}^{(k)}$ as follows.

   $$Y_j^{(k)} = \left(1 - \sum_{i=1}^{k-1} Y_j^{(i)}\right) \cdot (X_{j1}^{(k)})^2, \; X_j^{(k)} \sim \text{Unif}(\mathbb{S}^{d-k}), \quad \forall j = 1, \ldots, n-1.$$

   where $X_{j1}^{(k)}$ is the first coordinate of $X_j^{(k)}$. Define

   $$\pi_k \triangleq \arg\max_{j : j \neq \pi_1, \ldots, \pi_{k-1}} \left(\sum_{i=1}^{k} Y_j^{(i)}\right).$$

3. For $k = k_{max}+1, \ldots, d-1$, repeat

   $$\pi_k \triangleq \arg\max_{j : j \neq \pi_1, \ldots, \pi_{k-1}} \left(\sum_{i=1}^{k_{max}} Y_j^{(i)}\right).$$

We first claim that $(\sum_{i=1}^{k} Y_{\pi_k}^{(i)})$ is equal in distribution to $\|V_k^\top x_{j_k^*}\|_2^2$. Consider the following two sets of random variables.

$$A_k \triangleq \left(\sum_{i=1}^{k} Y_j^{(i)} : j \in [n-1], j \neq \pi_1, \ldots, \pi_{k-1}\right),$$
$$B_k \triangleq \left(\|V_k^\top x_j\|_2^2 : w_j = 1, j \neq 1, j_1^*, \ldots, j_{k-1}^*\right).$$

Each set contains $(n-k)$ random variables. We prove by induction that the joint distribution of the random variables of A_k is equal to those of B_k. Then the claim follows because $(\sum_{i=1}^{k} Y_{\pi_k}^{(i)})$ and $\|V_k^\top x_{j_k^*}\|_2^2$ are the maximums of A_k and B_k, respectively.

- Base case : As $V_1 = x_1$, $B_1 = (\|V_1^\top x_j\|_2^2 : w_j = 1, j \neq 1)$ is the set of squared inner products with x_1 for the $n-1$ other points. Since the $n-1$ points are iid uniformly random unit vectors independent of x_1, the squared inner products with x_1 are equal in distribution to $Y_j^{(1)} = (X_{j1}^{(1)})^2$. Therefore, the joint distribution of $B_1 = (\|V_1^\top x_j\|_2^2 : w_j = 1, j \neq 1)$ is equal to the joint distribution of $A_1 = (Y_j^{(1)} : j = 1, \ldots, n-1)$.
- Induction : Assume that the joint distribution of A_k is equal to the joint distribution of B_k. It is sufficient to show that given $A_k \equiv B_k$ the conditional joint distribution of $A_{k+1} = (\sum_{i=1}^{k+1} Y_j^{(i)} : j \in [n-1], j \neq \pi_1, \ldots, \pi_k)$ is equal to the conditional joint distribution of $B_{k+1} = (\|V_{k+1}^\top x_j\|_2^2 : w_j = 1, j \neq 1, j_1^*, \ldots, j_k^*)$. Define

  $$v_k = \frac{x_{j_k^*} - V_k V_k^\top x_{j_k^*}}{\|x_{j_k^*} - V_k V_k^\top x_{j_k^*}\|_2}.$$

v_k is the unit vector along the new orthogonal axis added on $\mathcal{V}_k$ for $\mathcal{V}_{k+1}$. Since we have

$$\|V_{k+1}^\top x_j\|_2^2 = \|V_k^\top x_j\|_2^2 + (v_k^\top x_j)^2, \quad \forall j : w_j = 1,$$

The two terms are independent of each other because $V_k \perp v_k$, and x_j is isotropically distributed. Hence, we only need to show that $((v_k^\top x_j)^2 : w_j = 1, j \neq 1, j_1^*, \ldots, j_k^*)$ is equal in distribution to $(Y_j^{(k+1)} : j \in [n-1], j \neq \pi_1, \ldots, \pi_k)$.

Since v_k is a normalized vector on the subspace $\mathcal{V}_k^\perp \cap \mathcal{D}_1$, and $x_{j_k^*}$ is drawn iid from an isotropic distribution, v_k is independent of $V_k^\top x_{j_k^*}$. Hence, the marginal distribution of v_k given $\mathcal{V}_k$ is uniform over $(\mathcal{V}_k^\perp \cap \mathcal{D}_1) \cap \mathbb{S}^{p-1}$. Also, v_k is also independent of the points $\{x_j : w_j = 1, j \neq 1, j_1^*, \ldots, j_k^*\}$. Therefore, the random variables $(v_k^\top x_j)^2$ for j with $w_j = 1, j \neq 1, j_1^*, \ldots, j_k^*$ are iid equal in distribution to $Y_j^{(k+1)}$ for any j.

Second, we can see that the k'th maximum of $\{\sum_{i=1}^k Y_j^{(i)} : j \in [n-1]\}$ is equal in distribution to $P_{k,(k)}^2$. This is because each $\sum_{i=1}^k Y_j^{(i)}$ can be seen as the norm of the projection of a uniformly random unit vector in $\mathbb{R}^d$ onto a k-dimensional subspace.

Now we are ready to complete the proof. Since $\left(\sum_{i=1}^k Y_{\pi_k}^{(i)}\right)$ is the maximum of the $n-k$ variables of A_k, it is greater than or equal to the k'th maximum of $\left(\sum_{i=1}^k Y_j^{(i)} : j \in [n-1]\right)$. Therefore, $\|V_k^\top x_{j_k^*}\|_2^2$ stochastically dominates $P_{k,(k)}^2$.

The second claim is clear because $\mathcal{V}_{\hat{k}} \subseteq \mathcal{V}_k$, and hence the norm of the projection onto $\mathcal{V}_k$ is always larger than the norm of the projection onto $\mathcal{V}_{\hat{k}}$.

B.6 Proof of Lemma 5a

Let x be an unit vector drawn uniformly at random from $\mathbb{S}^{d-1}$. Equivalently, x can be drawn from

$$x = \frac{w}{\|w\|_2}, \quad w \sim \mathcal{N}(0, I_{d\times d}).$$

Define $A^\perp \in \mathbb{R}^{(d-k)\times d}$ as a matrix with orthonormal rows such that $\|w\|_2^2 = \|Aw\|_2^2 + \|A^\perp w\|_2^2$ for any $w \in \mathbb{R}^d$. We have

$$\begin{aligned}
\Pr\left\{\|Ax\|_2^2 > \frac{k}{d}(1+\epsilon)\right\} &= \Pr\left\{\frac{\|Aw\|_2^2}{\|w\|_2^2} > \frac{k}{d}(1+\epsilon)\right\} \\
&= \Pr\left\{\frac{\|Aw\|_2^2}{\|Aw\|_2^2 + \|A^\perp w\|_2^2} > \frac{k}{d}(1+\epsilon)\right\} \\
&\geq \Pr\left\{\|Aw\|_2^2 > k(1+\epsilon), \|A^\perp w\|_2^2 < (d-k) - k\epsilon\right\} \\
&= \Pr\left\{\|Aw\|_2^2 > k(1+\epsilon)\right\} \cdot \Pr\left\{\|A^\perp w\|_2^2 < (d-k) - k\epsilon\right\}, \quad (22)
\end{aligned}$$

where the last equality follows from that $\|Aw\|_2$ and $\|A^\perp w\|_2$ are independent of each other because $w \sim \mathcal{N}(0, I_{d\times d})$. Note that $\|Aw\|_2^2$ and $\|A^\perp w\|_2^2$ are Chi-square random variables with degrees of freedom k and $d-k$, respectively.

Now we use the following lemma.

Lemma 8 (Chi-square upper-tail lower-bound) *For any $k \in \mathbb{N}$ and any $\epsilon \geq 0$, we have*

$$\Pr\{\chi_k^2 \geq k(1+\epsilon)\} \geq \frac{1}{3\sqrt{k\epsilon + 6}} \exp\left(-\frac{k\epsilon}{2}\right).$$

where χ_k^2 is the chi-square random variable with k degrees of freedom.

Suppose $0 \leq \epsilon \leq \alpha \frac{(d-k)^{\frac{1}{2}}}{k}$ for some $\alpha \in (0,1)$. It follows from Lemma 8 and the central limit theorem that

$$\begin{aligned}(22) &\geq \Pr\left\{\|Aw\|_2^2 > k(1+\epsilon)\right\} \cdot \Pr\left\{\|A^{\perp}w\|_2^2 - (d-k) < -\alpha(d-k)^{\frac{1}{2}}\right\} \\ &\geq \frac{f(\alpha)}{3k\epsilon+6}\exp\left(-\frac{k\epsilon}{2}\right)\end{aligned}$$

where $f(\alpha) \in (0,1)$ is some constant depending only on α.

Then it follows that

$$\begin{aligned}\Pr\left\{z^2_{(n-m+1)} < \frac{k}{d}(1+\epsilon)\right\} &= \Pr\left\{\exists I \subset [n], |I| = n-m+1 : z_i^2 < \frac{k}{d}(1+\epsilon), \forall i \in I\right\} \\ &\leq \binom{n}{m-1} \cdot \Pr\left\{z_1^2 < \frac{k}{d}(1+\epsilon)\right\}^{n-m+1} \\ &\leq \left(\frac{ne}{m}\right)^m \cdot \left(1 - \frac{f(\alpha)}{3k\epsilon+6}\exp\left(-\frac{k\epsilon}{2}\right)\right)^{n-m+1} \\ &\leq \exp\left\{m \log\frac{ne}{m} - \frac{f(\alpha)\cdot(n-m+1)}{3k\epsilon+6}\exp\left(-\frac{k\epsilon}{2}\right)\right\} \qquad (23)\end{aligned}$$

where we use the facts $\binom{n}{m} \leq \left(\frac{ne}{m}\right)^m$ and $1 + x \leq e^x, \forall x$.

Set $C = \frac{6}{f(\alpha)}$, and choose ϵ such that

$$\epsilon = \frac{1}{k}\min\left\{2\log\left(\frac{n-m+1}{Cm\left(\log\frac{ne}{m\delta}\right)^2}\right), \alpha\sqrt{d-k}\right\}.$$

This ϵ is valid because $0 \leq \epsilon \leq \alpha\frac{(d-k)^{\frac{1}{2}}}{k}$. Then we obtain

$$\begin{aligned}(23) &\leq \exp\left\{m\log\frac{ne}{m} - \frac{f(\alpha)\cdot(n-m+1)}{6\log\left(\frac{n-m+1}{Cm\left(\log\frac{ne}{m\delta}\right)^2}\right)+6} \cdot \frac{Cm\left(\log\frac{ne}{m\delta}\right)^2}{n-m+1}\right\} \\ &= \exp\left\{m\log\frac{ne}{m} - \frac{6\log\frac{ne}{m\delta}}{6\left(1+\log\left(\frac{f(\alpha)}{6}\cdot\frac{n-m+1}{m}\cdot(\log\frac{ne}{m\delta})^{-2}\right)\right)} \cdot m\log\frac{ne}{m\delta}\right\} \\ &\leq \exp\left\{m\log\frac{ne}{m} - \frac{6(1+\log\frac{n}{m})}{6(1+\log\frac{f(\alpha)}{6}+\log\frac{n}{m})} m\log\frac{ne}{m\delta}\right\} \\ &\leq \exp\left\{m\log\frac{ne}{m} - m\log\frac{ne}{m\delta}\right\} \\ &\leq \delta^m.\end{aligned}$$

This completes the proof.

B.7 Proof of Lemma 5b

We use a special case of Levy's lemma for this proof.

Lemma 9 ([12]) *For $x \sim$ Unif$(\mathbb{S}^{d-1})$,*

$$\Pr\{\|Ax\|_2 > m\|Ax\|_2 + t\} \leq \exp\left(-\frac{dt^2}{2\|A\|_2^2}\right),$$

$$\Pr\{\|Ax\|_2 < m\|Ax\|_2 - t\} \leq \exp\left(-\frac{dt^2}{2\|A\|_2^2}\right).$$

for any matrix $A \in \mathbb{R}^{p\times d}$ and $t > 0$. $m\|Ax\|_2$ is the median of $\|Ax\|_2$.

It follows from the lemma that

$$|E\|Ax\|_2 - m\|Ax\|_2| \le E\left[|\|Ax\|_2 - m\|Ax\|_2|\right] \le \int_0^\infty 2e^{-\frac{dt^2}{2\|A\|_2^2}} dt = \sqrt{\frac{2\pi}{d}}\|A\|_2.$$

Then we have

$$\begin{aligned}\Pr\left\{\|Ax_i\|_2 > \sqrt{\frac{\|A\|_F^2}{d}} + \sqrt{\frac{2\pi}{d}}\|A\|_2 + t\right\} &= \Pr\left\{\|Ax_i\|_2 > \sqrt{E\|Ax_i\|_2^2} + \sqrt{\frac{2\pi}{d}}\|A\|_2 + t\right\} \\ &\le \Pr\left\{\|Ax_i\|_2 > E\|Ax_i\|_2 + \sqrt{\frac{2\pi}{d}}\|A\|_2 + t\right\} \\ &\le \Pr\left\{\|Ax_i\|_2 > m\|Ax_i\|_2 + t\right\} \\ &\le \exp\left(-\frac{dt^2}{2\|A\|_2^2}\right).\end{aligned}$$

It follows that

$$\begin{aligned}&\Pr\left\{z_{(n-m+1)} > \sqrt{\frac{\|A\|_F^2}{d}} + \sqrt{\frac{2\pi}{d}}\|A\|_2 + t\right\} \\ &\le \Pr\left\{\exists I \subset [n], |I| = m : \|Ax_i\|_2 > \sqrt{\frac{\|A\|_F^2}{d}} + \sqrt{\frac{2\pi}{d}}\|A\|_2 + t, \forall i \in I\right\} \\ &\le \binom{n}{m} \cdot \Pr\left\{\|Ax_1\|_2 > \sqrt{\frac{\|A\|_F^2}{d}} + \sqrt{\frac{2\pi}{d}}\|A\|_2 + t\right\}^m \\ &\le \left(\frac{ne}{m}\right)^m \cdot \exp\left(-\frac{mdt^2}{2\|A\|_2^2}\right) \\ &= \exp\left\{m \log \frac{ne}{m} - \frac{mdt^2}{2\|A\|_2^2}\right\}.\end{aligned}$$

Replacing t with $\sqrt{\frac{2\|A\|_2^2}{d} \log \frac{ne}{m\delta}}$, we obtain the desired result.

B.8 Proof of Lemma 6a

Let $A = U\Sigma V^\top$ be the singular value decomposition of A. Then we have

$$E[\|AX\|_F^2] = E[\|U\Sigma V^\top X\|_F^2] = E[\|\Sigma X\|_F^2] = \sum_{i=1}^{\min(p,d)} \sigma_i^2 \cdot \left(\sum_{j=1}^{k} E[X_{ij}^2]\right) = \sum_{i=1}^{\min(p,d)} \sigma_i^2 \cdot \frac{k}{d} = \frac{k}{d}\|A\|_F^2.$$

where the second last equality follows from that X_{ij} is a coordinate of a uniformly random unit vector, and thus

$$E[X_{ij}^2] = \frac{1}{d}, \quad \forall i, j.$$

B.9 Proof of Lemma 6b

Consider the Stiefel manifold $V_k(\mathbb{R}^d)$ equipped with the Euclidean metric. We see that X is drawn from $V_k(\mathbb{R}^d)$ with the normalized Harr probability measure. We have

$$\|AX\|_F - \|AY\|_F \le \|AX - AY\|_F = \|A(X-Y)\|_F \le \|A\|_2\|X - Y\|_F$$

for any $X, Y \in \mathbb{R}^{d\times k}$. Since $\|A\|_2 \le 1$, $\|AX\|_F$ is a 1-Lipschitz function of X. Then it follows from [15, 12] that

$$\Pr\{\|AX\|_F > m\|AX\|_F + t\} \le e^{-\frac{(d-1)t^2}{8}},$$

where $m\|AX\|_F$ is the median of $\|AX\|_F$. Also, we have

$$\Pr\{|\|AX\|_F - m\|AX\|_F| > t\} \leq 2e^{-\frac{(d-1)t^2}{8}},$$

and then it follows that

$$|E\|AX\|_F - m\|AX\|_F| \leq E\left[|\|AX\|_F - m\|AX\|_F|\right] \leq \int_0^\infty 2e^{-\frac{(d-1)t^2}{8}} dt = \sqrt{\frac{8\pi}{d-1}}.$$

It follows from Jensen's inequality and Lemma 6a that

$$E\|AX\|_F \leq \sqrt{E\|AX\|_F^2} = \sqrt{\frac{k}{d}}\|A\|_F$$

Putting the above inequalities together using the triangle inequality, we obtain the desired result.

B.10 Proof of Lemma 8

For $k \geq 2$, it follows from [9, Proposition 3.1] that

$$\begin{aligned}
\Pr\{\chi_k^2 \geq k(1+\epsilon)\} &\geq \frac{1-e^{-2}}{2} \frac{k(1+\epsilon)}{k\epsilon + 2\sqrt{k}} \exp\left(-\frac{1}{2}(k\epsilon - (k-2)\log(1+\epsilon) + \log k)\right) \\
&\geq \frac{1}{3\sqrt{k}\epsilon + 6} \exp\left(-\frac{k}{2}(\epsilon - \log(1+\epsilon))\right) \\
&\geq \frac{1}{3\sqrt{k}\epsilon + 6} \exp\left(-\frac{k\epsilon}{2}\right).
\end{aligned}$$

For $k = 1$, we can see numerically that the inequality holds.