Consistent PCA and Spectral Clustering

Satoshi Hara
The University of Electro-Communications

Yuichi Yoshida
National Institute of Informatics

Abstract

Principal component analysis (PCA) and spectral clustering are representative methods for extracting and interpreting the inherent structure of data. However, if the output results significantly change upon the addition of new data points, it can lead to several issues such as instability in the downstream task or a lack of trust in the findings. To address these problems, we consider online variants of PCA and spectral clustering, and show that a natural subspace-preserving regularizer provides provable approximation and consistency guarantees. Here, an algorithm is said to have a high consistency if the output change, with respect to an appropriate distance metric, is small when new data points are added. We empirically confirm the superiority of the proposed methods using real-world data.

1 INTRODUCTION

Principal component analysis (PCA) (Abdi and Williams, 2010) and spectral clustering (Von Luxburg, 2007) are representative methods for extracting and interpreting the inherent structure of vector data and graph data, respectively, which have variety of applications ranging from data visualization and dimensionality reduction (Jolliffe and Cadima, 2016) to image analysis (Turk and Pentland, 1991), social network analysis (Von Luxburg, 2007), and recommendation systems (Sarwar et al., 2000). Their widespread adoption across various fields underscores their practical importance and effectiveness.

In real-world problems, practitioners frequently encounter scenarios where data arrive incrementally rather than in a batch, necessitating to update the results of PCA and spectral clustering over time. A critical but often overlooked issue arises when these methods produce significantly different outputs after incorporating new data points. Such inconsistency can severely impact downstream tasks, making statistical analysis unreliable, diminishing user trust, and complicating decision-making processes. For example, Figure 1 demonstrates PCA results on MNIST, highlighting how standard PCA (top row) suffers from considerable inconsistency where the positions of projected points drastically change as new data points, particularly those belonging to class $y = 4$, are incrementally added. This inconsistency fundamentally disrupts the utility and interpretability of data visualization.

To address this critical issue above, we explore online variants of PCA and spectral clustering specifically designed for scenarios where data points arrive sequentially. Specifically, we show that a natural subspace-preserving regularizer provides provable approximation and consistency guarantees; here, consistency implies that the output changes minimally in response to new data points, measured with an appropriate distance metric. The introduction of this regularizer ensures consistency of visualizations or clustering, which is crucial for applications where data grow over time. In particular, our theoretical results are valid even in the worst case where input sequences could be chosen arbitrarily.

Related work Online algorithm with low inconsistency (sometimes called *recourse* in the literature) have been studied for various problems including k-clustering (Lattanzi and Vassilvitskii, 2017; Fichtenberger et al., 2021; Yoshida and Ito, 2022; Łącki et al., 2024; Guo et al., 2021; Dong and Yoshida, 2024) and covering and packing problems (Gupta and Levin, 2020; Bhattacharya et al., 2023; Gupta et al., 2017). However, it is evident that their technique cannot be applied to PCA or spectral clustering, which involve the computation of eigenvalues and eigenvectors.

A related notion to inconsistency is *sensitivity* (Varma and Yoshida, 2023), where we say that an algorithm has a low sensitivity when the outputs of the algorithm on one input and a slightly perturbed input are close in, say, the Hamming distance or the ℓ_2 distance.

Proceedings of the 29[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

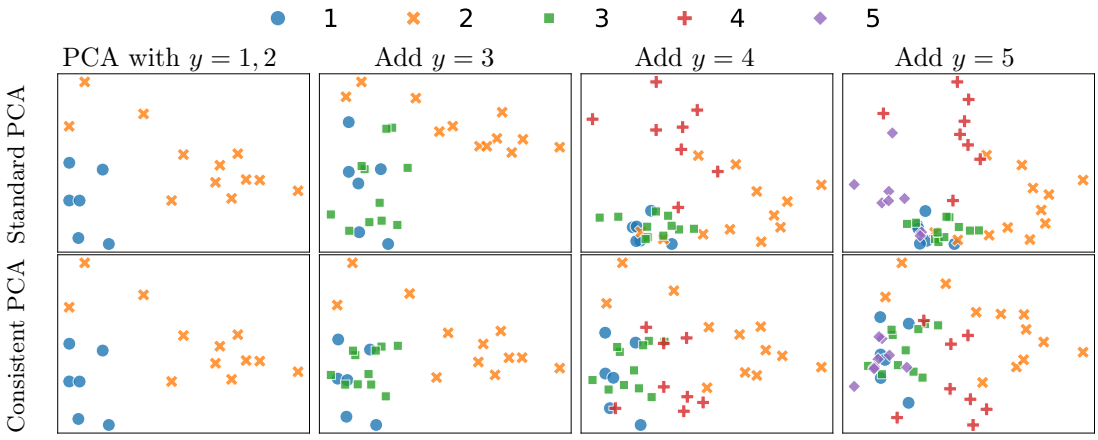

Figure 1: The results of standard PCA (top row) and consistent PCA (bottom row, proposed) on MNIST as new data points are incrementally added. Standard PCA exhibits drastic changes when the data point of the class $y = 4$ is newly added. Consistent PCA shows more stable results.

Bounding sensitivity is usually harder than bounding inconsistency because, in the former, the algorithm has no knowledge as to how the input was perturbed whereas, in the latter, we know that the last data point added to the input. Although the sensitivity of some spectral clustering algorithms have been investigated in Peng and Yoshida (2020), the bound is useful only when the input graph has a strong cluster structure.

Notations We use bold symbols to denote vectors. For vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^\top \boldsymbol{y}$ denote their inner product and $\boldsymbol{x} \otimes \boldsymbol{y} := \boldsymbol{x}\boldsymbol{y}^\top$ denote their outer product. For matrices A and B, we define $\langle A, B \rangle := A^\top B$ and $A \otimes B := AB^\top$. For a matrix A with orthonormal column vectors, let $\Pi_A = A \otimes A$ denote the projection matrix onto the space spanned by the column vectors of A. Let $A \in \mathbb{R}^{p \times p}$ be a symmetric matrix. Then, all the eigenvalues of A are real, and let $\lambda_i(A)$ denote the i-th largest eigenvalue of A. For an integer $1 \leq k \leq p$, the *Ky Fan k norm* of A is defined as $\|A\|_{F_k} = \sum_{i=1}^{k} \lambda_i(A)$. For two random variables X and Y, let $\mathrm{TV}(X, Y)$ denote the total variation distance between the distributions of X and Y.

All proofs are provided in Appendix A.

2 SUBSPACE-PRESERVING REGULARIZER

In this study, we focus on the following subspace-preserving regularizer for both PCA and spectral clustering.

$$\phi_{\mathrm{SP}}(W, W') = \frac{1}{2} \left\| \Pi_W - \Pi_{W'} \right\|_F^2, \tag{1}$$

where $W, W' \in \mathbb{R}^{p \times k}$ are orthonormal matrices, i.e., $W^\top W = I_k$ and $W'^\top W' = I_k$. This regularizer ensures that the projection matrices Π_W and $\Pi_{W'}$ are close in the Frobenius norm.

Note that $\phi_{\mathrm{SP}}(W, W')$ can be rewritten as

$$\begin{aligned}
\phi_{\mathrm{SP}}(W, W') &= \frac{1}{2} \left\| W \otimes W - W' \otimes W' \right\|_F^2 \\
&= \frac{1}{2} \cdot \mathrm{tr} \left((W \otimes W - W' \otimes W')^2 \right) \\
&= (k - \mathrm{tr}\,(W \otimes WW' \otimes W')) \\
&= \mathrm{tr}\langle W, (I - W' \otimes W')W \rangle.
\end{aligned}$$

This formulation enables us to solve PCA and spectral clustering with this regularizer using standard eigenvalue decomposition.

Chi et al. (2009) used this regularizer in the context of spectral clustering, under the name of *Preserving Clustering Membership* (PCM). They empirically showed that this regularizer helps to stabilize the clustering results over time, but the theoretical analysis remained unexplored. We show that this regularizer indeed provides provable approximation and inconsistency guarantees for both PCA and spectral clustering.

Connection to Davis-Kahan Theorem To see the intuition behind the regularizer, let $\mathcal{W}$ and $\mathcal{W}'$ be the subspaces spanned by the column vectors of W and W'. Then, the *principle angles* between $\mathcal{W}$ and $\mathcal{W}'$ are the angles $\theta_1, \ldots, \theta_k$ such that $\cos\theta_1, \ldots, \cos\theta_k$ are the singular values of $\langle W, W' \rangle$. Then by simple calculation, we can verify that the regularizer term $\left\| \Pi_W - \Pi_{W'} \right\|_F^2$ is equal to $2\sum_{i=1}^{k} \sin^2\theta_i$, which is the distance notion between subspaces used in the celebrated Davis-Kahan theorem (Davis and Kahan, 1970).

3 CONSISTENT PCA

In *principal component analysis* (PCA), given vectors $\boldsymbol{v}^{(1)}, \dots, \boldsymbol{v}^{(n)} \in \mathbb{R}^p$ and a positive integer k, the goal is to compute orthonormal vectors $\boldsymbol{w}_1, \dots, \boldsymbol{w}_k \in \mathbb{R}^p$ that maximizes the variance $\sum_{i=1}^k \langle \boldsymbol{w}_i, C\boldsymbol{w}_i \rangle$, where $C = \sum_{i=1}^n \boldsymbol{v}^{(i)} \otimes \boldsymbol{v}^{(i)}$ is the covariance matrix. It is clear that the maximum variance of $\|C\|_{F_k}$ is attained when $\boldsymbol{w}_1, \dots, \boldsymbol{w}_k$ are orthonormal eigenvectors corresponding to the top k eigenvalues of C, and those are called *principal components*.

In this section, we study an online variant of PCA from the viewpoint of inconsistency. In this problem, we are given a vector $\boldsymbol{v}^{(t)} \in \mathbb{R}^p$ at step t, and then for the current covariance matrix $C^{(t)} := \sum_{i=1}^t \boldsymbol{v}^{(i)} \otimes \boldsymbol{v}^{(i)}$, we want to compute orthonormal vectors $\boldsymbol{w}_1^{(t)}, \dots, \boldsymbol{w}_k^{(t)} \in \mathbb{R}^p$ such that

- the variance $\sum_{i=1}^k \langle \boldsymbol{w}_i^{(t)}, C^{(t)} \boldsymbol{w}_i^{(t)} \rangle$ is large, and
- the *inconsistency* $\sum_{t=2}^n \sum_{i=1}^k \|\boldsymbol{w}_i^{(t)} - \boldsymbol{w}_i^{(t-1)}\|^2$ is small.

Note that we do not assume anything about the inputs $\boldsymbol{v}^{(t)}$ such as i.i.d. We allow the inputs to be chosen arbitrarily.

3.1 Algorithm

To solve the online PCA problem with small inconsistency, for a parameter $\lambda > 0$, we introduce an objective function $f_\lambda^{(t)} : (\mathbb{R}^p)^k \to \mathbb{R}$ defined as

$$f_\lambda^{(t)}(\boldsymbol{w}_1, \dots, \boldsymbol{w}_k) := \mathrm{tr}\langle W, C^{(t)} W \rangle - \lambda \, \phi_{\mathrm{SP}}(W, W^{(t-1)}),$$

where $W \in \mathbb{R}^{p \times k}$ and $W^{(t-1)} \in \mathbb{R}^{p \times k}$ are the matrices whose i-th column is $\boldsymbol{w}_i$ and $\boldsymbol{w}_i^{(t-1)}$, respectively. We assume $\boldsymbol{w}_i^{(0)} = \boldsymbol{0}$ for every $i \in \{1, 2, \dots, k\}$. The subspace-preserving regularizer is added to ensure that W remains close to $W^{(t-1)}$.

At each step t, we solve the following problem:

$$\begin{array}{lll} \text{maximize} & f_\lambda^{(t)}(\boldsymbol{w}_1, \dots, \boldsymbol{w}_k) & \\ \text{subject to} & \|\boldsymbol{w}_i\|^2 = 1 & \forall i = 1, \dots, k, \\ & \langle \boldsymbol{w}_i, \boldsymbol{w}_j \rangle = 0 & \forall 1 \le i < j \le k. \end{array} \tag{2}$$

Note that $f_\lambda^{(t)}$ can be rewritten as

$$\begin{aligned} &\mathrm{tr}\langle W, C^{(t)} W \rangle - \lambda \cdot \mathrm{tr}\langle W, (I - W^{(t-1)} \otimes W^{(t-1)}) W \rangle \\ &= \mathrm{tr}\langle W, \{C^{(t)} - \lambda (I - W^{(t-1)} \otimes W^{(t-1)})\} W \rangle. \end{aligned}$$

Hence, we can solve (2) by computing the top k eigenvalues of the matrix $C^{(t)} - \lambda(I - W^{(t-1)} \otimes W^{(t-1)})$, which can be done in polynomial time.

Algorithm 1: Consistent PCA

$\boldsymbol{w}_1^{(0)}, \dots, \boldsymbol{w}_k^{(0)} \leftarrow \boldsymbol{0}$;
for $t = 1, \dots, n$ **do**
  Receive $\boldsymbol{v}^{(t)}$;
  Solve (2) and let $\tilde{\boldsymbol{w}}_1^{(t)}, \dots, \tilde{\boldsymbol{w}}_k^{(t)} \in \mathbb{R}^p$ be the obtained solution;
  Let $\tilde{W}^{(t)} \in \mathbb{R}^{p \times k}$ be the matrix whose i-th column is $\tilde{\boldsymbol{w}}_i^{(t)}$;
  Solve (3) and let $R^{(t)} \in \mathbb{R}^{k \times k}$ be the obtained rotation matrix;
  Let $W^{(t)} := \tilde{W}^{(t)} R^{(t)}$ and let $\boldsymbol{w}_i^{(t)}$ be the i-th column of $W^{(t)}$;

Let $\tilde{\boldsymbol{w}}_1^{(t)}, \dots, \tilde{\boldsymbol{w}}_k^{(t)} \in \mathbb{R}^p$ be the obtained solution, and let $\tilde{W}^{(t)} \in \mathbb{R}^{p \times k}$ be the matrix whose i-th column is $\tilde{\boldsymbol{w}}_i^{(t)}$. Then, we rotate it to align it with the previous matrix $W^{(t-1)}$. Specifically, we solve the following problem:

$$\text{minimize } \|\tilde{W}^{(t)} R - W^{(t-1)}\|_F^2 \quad \text{subject to} \quad R^\top R = I_k. \tag{3}$$

This problem is known as Procrustes alignment (Schönemann, 1966) whose solution is given by $R^{(t)} := U^\top V$, where U and V are the left and right singular matrices of $\langle \tilde{W}^{(t)}, W^{(t-1)} \rangle$. Then, we output the vectors $\boldsymbol{w}_1^{(t)}, \dots, \boldsymbol{w}_k^{(t)}$, where $\boldsymbol{w}_i^{(t)}$ is the i-th column of $W^{(t)} := \tilde{W}^{(t)} R^{(t)}$. We call this method ConsistentPCA. See Algorithm 1 for more details.

3.2 Theoretical Guarantees

We first show that the variance of the solution obtained by ConsistentPCA is close to the optimal, which is $\|C^{(t)}\|_{F_k}$.

Lemma 3.1. *We have*

$$\sum_{i=1}^k \langle \boldsymbol{w}_i^{(t)}, C^{(t)} \boldsymbol{w}_i^{(t)} \rangle \ge \|C^{(t)}\|_{F_k} - O(\lambda k).$$

Now, we analyze the inconsistency of ConsistentPCA.

Lemma 3.2. *We have*

$$\begin{aligned} \sum_{t=2}^n \sum_{i=1}^k \|\boldsymbol{w}_i^{(t)} - \boldsymbol{w}_i^{(t-1)}\|^2 &\le \sum_{t=2}^n \|\Pi_{W^{(t)}} - \Pi_{W^{(t-1)}}\|^2 \\ &\le \frac{2\mathrm{tr}\langle W^{(n)}, C^{(n)} W^{(n)} \rangle}{\lambda}. \end{aligned} \tag{4}$$

Consider setting $\lambda = \Theta(\varepsilon \|C^{(n)}\|_{F_k} / k)$. Then, Lemma 3.1 implies that $(\boldsymbol{w}_1^{(n)}, \dots, \boldsymbol{w}_k^{(n)})$ is a $(1 -$

ε)-approximate solution for the problem of maximizing the variance of $C^{(n)}$ and Lemma 3.2 implies that the inconsistency is bounded by $2k \cdot \mathrm{tr}\langle W^{(n)}, C^{(n)}W^{(n)}\rangle/(\varepsilon\|C^{(n)}\|_{F_k}) \leq O(\varepsilon^{-1}k)$, which is significantly smaller than the trivial upper bound of $O(kn)$ on inconsistency.

Actually, this upper-bound of inconsistency is tight when k is constant, which is typically the case for data visualization where $k = 2$. The following theorem shows that any algorithm for the online PCA that outputs $(1-\varepsilon)$-approximate solution must have inconsistency $\Omega(\varepsilon^{-1})$.

Theorem 3.3. *Let $\varepsilon > 0$ be sufficiently small. Consider any online PCA algorithm that for any $t = 1, \ldots, n$, after it has seen the first t vectors $\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(t)} \in \mathbb{R}^2$, outputs a unit vector $\boldsymbol{w}^{(t)} \in \mathbb{R}^2$ satisfying*

$$\langle \boldsymbol{w}^{(t)}, C^{(t)}\boldsymbol{w}^{(t)}\rangle \geq (1-\varepsilon) \max_{\|\boldsymbol{w}\|=1} \langle \boldsymbol{w}, C^{(t)}\boldsymbol{w}\rangle,$$

where $C^{(t)} := \sum_{i=1}^{t} \boldsymbol{v}^{(i)} \otimes \boldsymbol{v}^{(i)}$. Then we have

$$\sum_{t=2}^{n} \|\boldsymbol{w}^{(t)} - \boldsymbol{w}^{(t-1)}\|^2 = \Omega\left(\frac{1}{\varepsilon}\right).$$

4 CONSISTENT SPECTRAL CLUSTERING

Let $G = (V, E)$ be a graph. The *Laplacian* of G is defined as $L = D - A$, where $D \in \mathbb{R}^{V\times V}$ is the diagonal degree matrix and $A \in \mathbb{R}^{V\times V}$ is the adjacency matrix. It is known that the Laplacian is positive semidefinite and hence all the eigenvalues are nonnegative real numbers.

In *spectral clustering*, given a graph $G = (V, E)$ and an integer k, we compute the smallest k eigenvectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k \in \mathbb{R}^V$ of the Laplacian L, then we map each vertex $v \in V$ to a k-dimensional vector $\boldsymbol{p}_v = (\boldsymbol{w}_1(v), \ldots, \boldsymbol{w}_k(v))$, which is called the *spectral embedding*. Then, we apply some geometric clustering method to the set of points $P = \{\boldsymbol{p}_v\}_{v\in V}$ to compute a partition of V into k parts. Here, instead of typical k-means, we consider the *k-median problem* to find such a partition[1]. Specifically, given points P, we compute a set $\{c_1, \ldots, c_k\} \subseteq V$ of k vertices that minimizes

$$\sum_{v\in V} d(\boldsymbol{p}_v, \{\boldsymbol{p}_{c_1}, \ldots, \boldsymbol{p}_{c_k}\}), \tag{5}$$

where $d(\boldsymbol{p}, Q) := \min_{\boldsymbol{q}\in Q} \|\boldsymbol{p} - \boldsymbol{q}\|$ is the minimum distance between $\boldsymbol{p}$ and a point in Q. Then, we add each $v \in V$ to a vertex set V_i, where $i \in \{1, \ldots, k\}$ is such that the point closest to $\boldsymbol{p}_v$ among $\{\boldsymbol{p}_{c_1}, \ldots, \boldsymbol{p}_{c_k}\}$ is $\boldsymbol{p}_{c_i}$ (break ties arbitrarily), and we output the partition $(V_1, \ldots, V_k)$.

We consider an online counterpart of spectral clustering and show PCM (Chi et al., 2009) provides consistent solution. In the online setting, we are given the vertex set V in advance. Then at step $t \in \{1, 2, \ldots, m\}$, an edge $e^{(t)} \in \binom{V}{2}$ is given, where $\binom{V}{2} := \{\{u, v\} \in V \mid u \neq v\}$. Let $E^{(t)} = \{e^{(1)}, \ldots, e^{(t)}\}$ and $G^{(t)} = (V, E^{(t)})$. The goal is to output a partition $(V_1^{(t)}, \ldots, V_k^{(t)})$ of V into k parts such that

- the output clustering is close to the one obtained by applying spectral clustering on $G^{(t)}$.
- the inconsistency $\sum_{i=2}^{m} \mathrm{TV}(C^{(t)}, C^{(t-1)})$ is low, where $C^{(t)} = \{c_1^{(t)}, \ldots, c_k^{(t)}\}$ is the set of vertices corresponding to the output centroids for the t-th step.

Note that we do not assume anything about the new edge $e^{(t)}$, which can be chosen arbitrarily.

4.1 Algorithm of PCM (Chi et al., 2009)

Let $L^{(t)} \in \mathbb{R}^{V\times V}$ denote the Laplacian of $G^{(t)}$. We can compute the smallest k eigenvalues of $L^{(t)}$ by minimizing $\sum_{i=1}^{k}\langle \boldsymbol{w}_i, L^{(t)}\boldsymbol{w}_i\rangle$. To obtain consistent solution, for a parameter $\lambda > 0$, Chi et al. (2009) introduced an objective function $f_\lambda^{(t)} : (\mathbb{R}^V)^k \to \mathbb{R}$ defined as

$$f_\lambda^{(t)}(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k) := \mathrm{tr}\langle W, L^{(t)}W\rangle + \lambda\, \phi_{\mathrm{SP}}(W, W^{(t-1)}),$$

where $W, W^{(t-1)} \in \mathbb{R}^{V\times k}$ are the matrices whose i-th column is $\boldsymbol{w}_i$ and $\boldsymbol{w}_i^{(t-1)}$, respectively.

Then, Chi et al. (2009) considered the following minimization variant of (2).

$$\begin{array}{lll} \text{minimize} & f_\lambda^{(t)}(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k) & \\ \text{subject to} & \|\boldsymbol{w}_i\|^2 = 1 & \forall i = 1, \ldots, k, \\ & \langle \boldsymbol{w}_i, \boldsymbol{w}_j\rangle = 0 & \forall 1 \leq i < j \leq k. \end{array} \tag{6}$$

The details of our modified version of PCM is given in Algorithm 2. Let $\boldsymbol{w}_1^{(t)}, \ldots, \boldsymbol{w}_k^{(t)} \in \mathbb{R}^V$ be the obtained vectors by solving (6). Define $\boldsymbol{p}_v^{(t)} = (\boldsymbol{w}_1^{(t)}(v), \ldots, \boldsymbol{w}_k^{(t)}(v))$ for each $v \in V$. Different from Chi et al. (2009), we apply the D^1-sampling algorithm (Arthur and Vassilvitskii, 2007) to solve the k-median problem, which achieves $O(\log k)$-approximation to the k-median problem (5). In this algorithm, we first sample $c_1^{(t)}$ uniformly from V. Then, for each $i = 2, \ldots, k$, we sample $c_i^{(t)} \in V$ with probability proportional to $d(\boldsymbol{p}_v^{(t)}, \{\boldsymbol{p}_{c_i^{(t)}}^{(t)}, \ldots, \boldsymbol{p}_{c_{i-1}^{(t)}}^{(t)}\})$.

[1] We consider the setting where centroids are chosen from P. We can also analyze k-means though the inconsistency bound is slightly worse. See Appendix A.6 for details.

Algorithm 2: Preserving Cluster Membership

Procedure D^1-SAMPLING($V, \{\boldsymbol{p}_v\}_{v\in V}$)
  Sample $c_1 \in V$ uniformly at random;
  for $i = 2, \dots, k$ **do**
    Let $\mathcal{D}$ be the distribution over V such that $v \in V$ is sampled with probability $\propto d(\boldsymbol{p}_v, \{\boldsymbol{p}_{c_1}, \dots, \boldsymbol{p}_{c_{i-1}}\})$;
    Sample $c_i \in V$ from $\mathcal{D}$;
  return $\{c_1, \dots, c_k\}$.
Procedure PCM(V)
  $\boldsymbol{w}_1^{(0)} \leftarrow \mathbf{0}, \dots, \boldsymbol{w}_k^{(0)} \leftarrow \mathbf{0}$;
  for $t = 1, \dots, m$ **do**
    Receive $e^{(t)} \in \binom{V}{2}$;
    Let $L^{(t)} \in \mathbb{R}^{V\times V}$ be the Laplacian of the graph $G^{(t)} = (V, \{e^{(1)}, \dots, e^{(t)}\})$;
    Compute $\boldsymbol{w}_1^{(t)}, \dots, \boldsymbol{w}_1^{(t)}$ by solving (6);
    $\boldsymbol{p}_v^{(t)} \leftarrow (\boldsymbol{w}_1^{(t)}(v), \dots, \boldsymbol{w}_k^{(t)}(v))$ for $v \in V$;
    $\{c_1^{(t)}, \dots, c_k^{(t)}\} \leftarrow$ D^1-SAMPLING($V, \{\boldsymbol{p}_v^{(t)}\}_{v\in V}$);
    Compute the partition $(V_1^{(t)}, \dots, V_k^{(t)})$ of V using $c_1^{(t)}, \dots, c_k^{(t)}$.

4.2 Theoretical Guarantees

For a positive semidefinite matrix, let $\lambda_i(L)$ denote the i-th smallest eigenvalue of L. Then by modifying the proof of Lemma 3.1 for minimization, we obtain the following approximation guarantee:

Lemma 4.1. *We have*

$$\sum_{i=1}^{k} \langle \boldsymbol{w}_i^{(t)}, L^{(t)} \boldsymbol{w}_i^{(t)} \rangle \le \sum_{i=1}^{k} \lambda_i(L^{(t)}) + O(\lambda k).$$

As the minimum of $\sum_{i=1}^{k} \langle \boldsymbol{w}_i, L^{(t)} \boldsymbol{w}_i \rangle$ subject to $\boldsymbol{w}_1, \dots, \boldsymbol{w}_k$ being orthonormal is $\sum_{i=1}^{k} \lambda_i(L^{(t)})$, this lemma shows that the obtained vectors are close to the optimum. Combined with the fact that D^1-sampling achieves $O(\log k)$-approximation to the k-median problem, PCM outputs a partition with a reasonable quality.

We now turn to the inconsistency of the spectral embedding.

Lemma 4.2. *We have*

$$\sum_{t=1}^{m} \sum_{v\in V} \|\boldsymbol{p}_v^{(t)} - \boldsymbol{p}_v^{(t-1)}\|^2 \le \frac{1}{\lambda} \sum_{i=1}^{k} \lambda_i(L^{(m)}) + O(k).$$

Next, we analyze the inconsistency of the centroids selected by PCM. We note that existing consistent algorithms for the k-median problem (Lattanzi and Vassilvitskii, 2017; Fichtenberger et al., 2021; Yoshida and Ito, 2022; Guo et al., 2021; Dong and Yoshida, 2024) cannot be directly applied in our setting, as they assume a new data point arrives at each step, whereas in our case, the positions of existing points are perturbed.

Lemma 4.3. *Let $C^{(t)} = \{c_1^{(t)}, \dots, c_k^{(t)}\}$ and $\mathrm{OPT}^{(t)}$ be the optimal value of the k-median problem for $P^{(t)} = \{\boldsymbol{p}_v^{(t)} : v \in V\}$. Then, we have*

$$\sum_{t=1}^{m} \mathrm{TV}(C^{(t)}, C^{(t-1)}) = O\left(kn \sqrt{\sum_{i=1}^{k} \frac{\lambda_i(L^{(m)})}{\lambda} + O(k)} \sqrt{\sum_{t=1}^{m} \left(\frac{1}{\mathrm{OPT}^{(t)}} \right)^2} \right).$$

Consider setting $\lambda = \Theta(\varepsilon \sum_{i=1}^{k} \lambda_i(L^{(m)})/k)$ for $\varepsilon > 0$. By Lemma 4.1, we can obtain a $(1+\varepsilon)$-approximate spectral embedding. Note that $\mathrm{OPT}^{(t)}$ can be as large as $\Omega(n)$, and in such a case, the bound in Lemma 4.3 reduces to $O\left(kn\sqrt{\frac{k}{\varepsilon}}\sqrt{\frac{m}{n^2}}\right) = O\left(k\sqrt{\frac{km}{\varepsilon}}\right)$, which is sublinear in the number of edges. Note that the trivial bound on inconsistency is $O(km)$.

4.3 Implementation Details

In practice, we want $C^{(t)}$ and $C^{(t-1)}$ to be exactly equal with high probability (if we share the randomness) rather than just having a small total variation distance. To this end, we adopted the sampling algorithm similar to SEEDEDSTABLEDR of Hara and Yoshida (2023) (Algorithm 4), which converts the latter guarantee to the former guarantee.

5 EXPERIMENTS: PCA

We demonstrate that the proposed CONSISTENTPCA has a higher consistency than the existing methods using some benchmark datasets. For all the experiments, we used a workstation with 256 cores of AMD EPYC processors and 758GB of RAMS.

All the codes and results are available at `github.com/sato9hara/consistent-pca-sc`.

5.1 Setups

Baseline Methods As the naive baseline, we adopted vanilla PCA. Vanilla PCA computes the top-k eigenvectors of the covariance matrix $C^{(t)}$ for each step t without any cares on the consistency.

A concept closely related to the consistency of PCA is robustness, which aims to estimate principal components accurately even when data contains outliers. As the baseline method from the robustness literature, we selected Spherical PCA (Locantore et al., 1999). Spherical PCA applies Vanilla PCA to data instances that have been scaled to have unit length $\boldsymbol{v}/\|\boldsymbol{v}\|$, making it robust against outliers.[2]

As the advanced methods, we adopted two Online PCA algorithms from Nie et al. (2016). The first algorithm, GD, updates a matrix P by iterating the following descent step (D) and projection step (P):

$$\text{(D)} \quad \hat{P}^{(t+1)} = P_t - \eta C^{(t)},$$
$$\text{(P)} \quad P^{(t+1)} = \operatorname*{argmin}_{P \in \mathcal{P}_{d-k}} \|P - \hat{P}^{(t+1)}\|_F^2,$$

where $\mathcal{P}_{d-k}$ is the set of positive semidefinite matrices whose trace is $d-k$ and the maximum eigenvalue is upper-bounded by one. We can obtain the estimated principal components $W^{(t+1)}$ as the k-smallest eigenvectors of $P^{(t+1)}$.

The second algorithm, MEG, updates P by using the quantum relative entropy instead of Frobenius norm, which results in the following two steps:

$$\text{(D)} \quad \hat{P}^{(t+1)} = \mathrm{expm}(\mathrm{logm} P^{(t)} - \eta C^{(t)}),$$
$$\text{(P)} \quad P^{(t+1)} = \operatorname*{argmin}_{P \in \mathcal{P}_{d-k}} D(P, \hat{P}^{(t+1)}),$$

where expm and logm denotes matrix exponential and logarithm, respectively, and $D(P, \hat{P}^{(t+1)}) = \mathrm{tr}(P(\mathrm{logm} P - \mathrm{logm} \hat{P}^{(t)}) + \hat{P}^{(t)} - P)$.

In both algorithms, the learning rate $\eta > 0$ controls the magnitude of the update. Smaller η results in small changes in the update and thus the estimated principal components tend to be consistent, while the converse is true for large η. We observed that too large η can spoil the update, and thus the careful tuning of η is essential for these methods.

Datasets We evaluated the effectiveness of ConsistentPCA using real-world datasets. First, we conducted experiments using the face image dataset from `sklearn.datasets.fetch_lfw_people`, which contains face images from multiple individuals. We set `min_faces_per_person=10` to include only those individuals who have at least ten images. All other settings remained at their default values, resulting in each face image being represented as a $d = 2919$ dimensional vector. During each experimental run, we fed images for each individual incrementally to each PCA algorithm.

Additionally, we evaluated PCAs using several classification datasets, `gas-drift`, `har`, `micro-mass`, and `mnist_784` from OpenML. For each dataset with the size of N, we randomly selected a subset containing $\min\{1,000, N\}$ data points. We then incrementally fed the data points to each PCA algorithm in a classwise manner.

Metrics We employed the following two metrics to assess the performance of PCAs.

PCA Objective: We used the average PCA objective $\mathrm{Obj}^{\mathrm{PCA}} = \frac{1}{n}\sum_{t=1}^{n} \mathrm{tr}\langle W^{(t)}, C^{(n)} W^{(t)}\rangle$ at the last n-th update to assess the quality of the estimated principal components $W^{(t)}$. A large value of $\mathrm{Obj}^{\mathrm{PCA}}$ indicates that a PCA algorithm could find nearly optimal principal components in the early stage of the PCA updates by observing only a few examples. Hence, that algorithm is considered to be effective in term of the PCA objective.

Average Inconsistency: The average inconsistency $\mathrm{IC}^{\mathrm{PCA}} = \frac{1}{n-1}\sum_{t=2}^{n} \|W^{(t)} - W^{(t-1)}\|_F^2$ measures the average squared difference in the principal components with each update.

5.2 Results

Figure 2 shows the results on the face and OpenML data experiments. The results on the other configurations could be found in Appendix B. Because some of the baseline methods exhibit significantly worse objectives or inconsistencies, we only show the plots around the tradeoff curves of ConsistentPCA.

In the experiments, we varied the regularization λ of ConsistentPCA and the learning rate η of GD/MEG algorithms. We repeated the experiments 30 times and plotted the averages of $\mathrm{Obj}^{\mathrm{PCA}}$ and $\mathrm{IC}^{\mathrm{PCA}}$ of each method in the figures.

There are two important implications. First, the proposed ConsistentPCA achieves a wide and meaningful range of tradeoffs between the $\mathrm{Obj}^{\mathrm{PCA}}$ and the average inconsistency $\mathrm{IC}^{\mathrm{PCA}}$, and it naturally coincides with vanilla PCA in the limit $\lambda \to 0$. This flexibility is notably absent in GD and MEG, where adjusting the learning rate η typically fails to yield similarly reasonable tradeoffs.

Second, the performance of GD and MEG is highly

[2]Although more advanced robust PCA methods exist (Lerman and Maunu, 2018), we did not select them as baselines due to computational scalability concerns. The proposed method, Vanilla PCA, Spherical PCA, and the Online PCA methods we adopted can all be executed efficiently using just a few eigenvalue decompositions. In contrast, advanced robust PCA methods require tens or even hundreds of eigenvalue decompositions until convergence, making them impractical for repeated use every time new instaces are added.

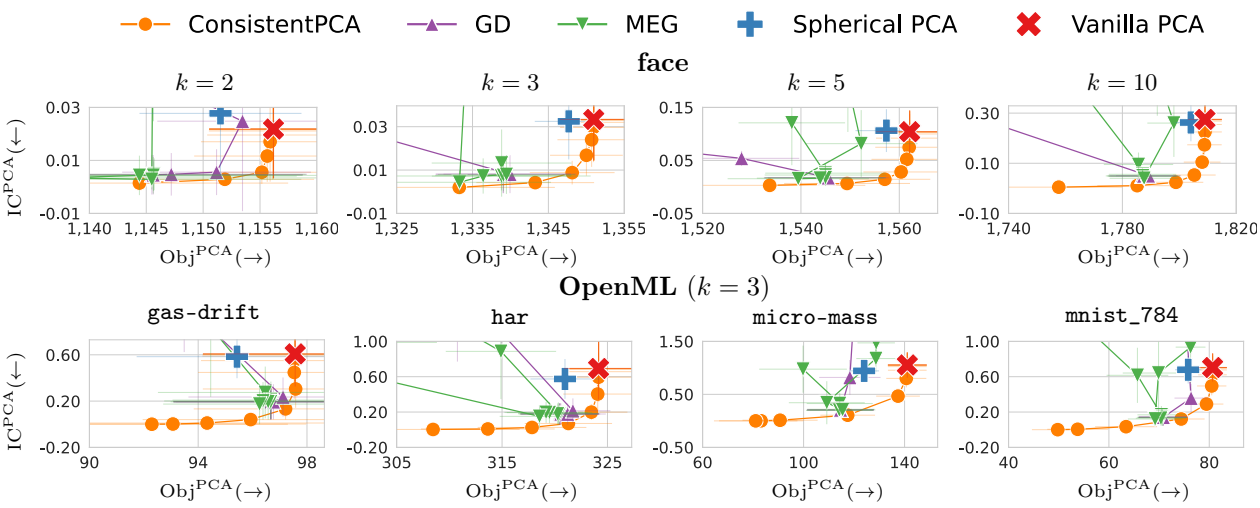

Figure 2: Results of PCA on the face and OpenML datasets. The error bars denote standard deviations. The arrows in the axis labels indicate favorable directions.

sensitive to the choice of η. These methods produce acceptable results only for a limited range of carefully tuned η values, but even small deviations from optimal η cause their performance to degrade significantly, resulting in considerably worse inconsistencies or objectives. Such sensitivity makes GD and MEG difficult to apply reliably in practice. In contrast, CONSISTENT-PCA exhibits reasonable performance across a broader range of λ, consistently outperforming GD, MEG, and Spherical PCA in terms of the tradeoffs.

Spherical PCA also fails to demonstrate significant improvements over Vanilla PCA. This limitation arises because the robustness against outliers does not necessarily indicate consistency when data points are incremental added. Outlier robustness alone cannot prevent drastic shifts in PCA.

6 EXPERIMENTS: SPECTRAL CLUSTERING

6.1 Setups

In the experiments, we demonstrate that PCM (Chi et al., 2009) has a high consistency.

Baseline Methods We adopted vanilla spectral clustering (SC) as the baseline. Vanilla SC simply computes the smallest-k eigenvectors of the Laplacian $L^{(t)}$ for each step t and applies D^1-SAMPLING.

We also adopted two Online SC algorithms. The first algorithm uses the dynamic sparsifier (DSp) (Sun and Zanetti, 2019; Laenen and Sun, 2024), which we refer to as DSp-SC. DSp-SC maintains a sparsified graph instead of the original graph. DSp-SC ensures that the clustering result on the sparsified graph can well approximate the result of Vanilla SC. DSp-SC also enables us to update its sparsified graph every time a new edge is added.

The second algorithm, *Preserving Cluster Quality* (PCQ) (Chi et al., 2009; Xu et al., 2014) applys SC to $\alpha L^{(t)} + (1-\alpha)L^{(t-1)}$ instead of $L^{(t)}$ so that the cluster structure in the previous time step is taken into consideration. In PCQ, the parameter $\alpha \in [0, 1]$ controls the balance between the current and previous Laplacians.

Datasets[3] We evaluated the effectiveness of PCM on an email dataset[4], a real-world temporal graph. This temporal graph consists of a set of edges (u, v) and their timestamps t, where a node u sent an email to another node v at time t. We split the edges to 20 consecutive chunks based on the timestamps. We focused on the largest connected subgraph consisting of 684 nodes appeared in the first chunk and tracked the evolution of edges between these nodes. We first construct a graph from the first chunk where the nodes are connected by an edge if they communicate with an email within the first chunk. In every chunk, we add edges to the graph when the nodes communicate within the chunk. In this way, we constructed a temporal graph with 20 time steps. The graph in the first time step contains 6,412 edges, while the graph in the last time step contains 26,994 edges.

We also used the Facebook dataset[5] for evaluation. We used the six out of ten community graphs with more than 100 nodes in the dataset. For each graph, we randomly assigned ten distinct timestamps to each edge

[3] Synthetic data experiments with various graph sizes can be found in Appendix D.

[4] `https://snap.stanford.edu/data/email-Eu-core-temporal.html`

[5] `https://snap.stanford.edu/data/ego-Facebook.html`

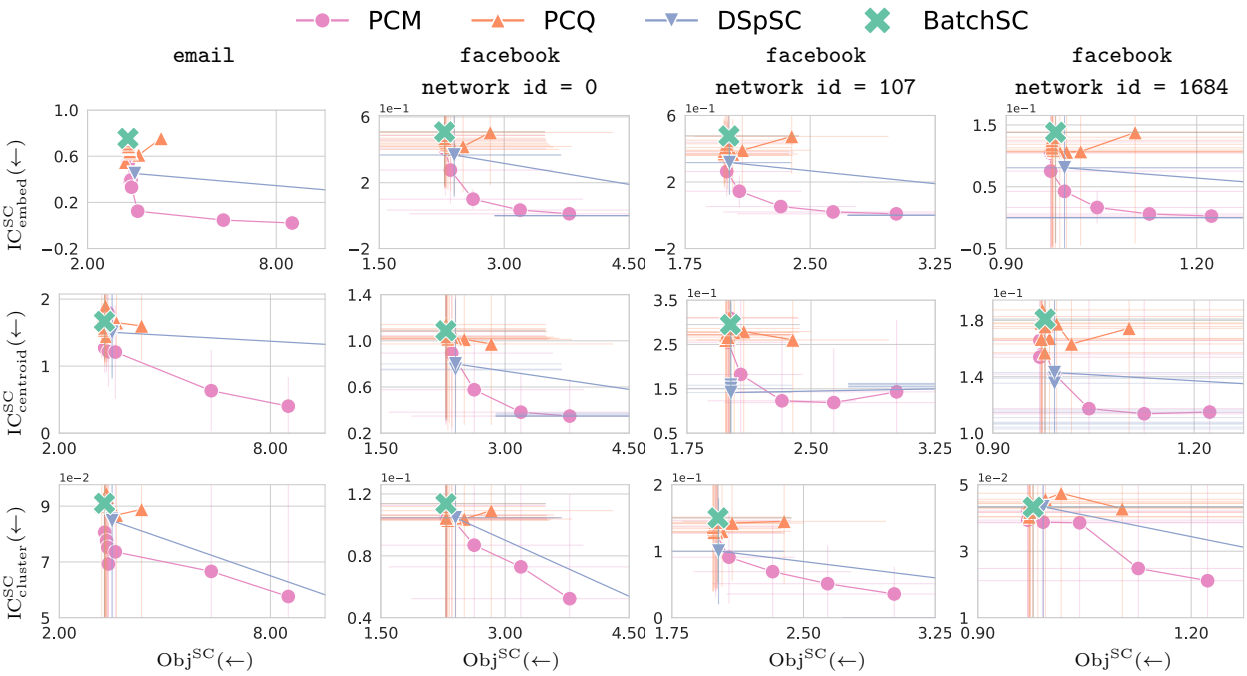

Figure 3: Results of Spectral Clustering ($k = 3$) on the email and facebook datasets. The error bars denote standard deviations. The arrows in the axis labels indicate favorable directions.

so that the edges are added to the graph over the ten rounds of evolution. Similar to the email dataset experiment, we focused on the largest connected subgraph in the first timestamp and tracked the evolution of edges between these nodes.

Metrics We employed the following four metrics:

SC Objective: We used the average SC objective $\mathrm{Obj}^{\mathrm{SC}} = \frac{1}{m}\sum_{t=1}^{m} \mathrm{tr}\langle W^{(t)}, L^{(m)}W^{(t)}\rangle$ at the last m-th update to assess the quality of the spectral embedding W. A small value of $\mathrm{Obj}^{\mathrm{SC}}$ indicates that the algorithm could find nearly optimal embeddings in the early stage of the SC updates by observing only a few edges.

Embedding Inconsistency: The embedding inconsistency $\mathrm{IC}^{\mathrm{SC}}_{\mathrm{embed}} = \frac{1}{m-1}\sum_{t=2}^{m} \|W^{(t)} - W^{(t-1)}\|_F^2$ measures the average squared difference in the spectral embedding with each update.

Centroid Inconsistency: The centroid inconsistency $\mathrm{IC}^{\mathrm{SC}}_{\mathrm{centroid}} = \frac{1}{m-1}\sum_{t=2}^{m}\sum_{i=1}^{k} \|c_i^{(t)} - c_i^{(t-1)}\|^2$ measures the change of centroids over the updates. This metric can be interpreted as an empirical analogue of Lemma 4.3.

Clustering Inconsistency: The clustering inconsistency $\mathrm{IC}^{\mathrm{SC}}_{\mathrm{cluster}} = \frac{1}{m-1}\sum_{t=2}^{m}\frac{1}{|V|}\sum_{v\in V} 1[y_v^{(t)} \neq y_v^{(t-1)}]$, where $y_v^{(t)}$ denotes the clustering label of the node v at time t, measures the fraction of the nodes whose clustering assignment has changed.

6.2 Results

Figure 3 shows the results on the email and three graphs (id = 0, 107, 1684) of the facebook dataset experiments with the number of clusters set to $k = 3$. The results on the other configurations could be found in Appendix C. In the experiments, we varied the regularization λ of PCM, α of PCQ, and sampling parameter of DSp-SC.

The figures underline three key points. First, PCM successfully provides a comprehensive and effective range of tradeoffs between the objective $\mathrm{Obj}^{\mathrm{SC}}$ and several inconsistencies $\mathrm{IC}^{\mathrm{SC}}_{*}$. In contrast, the baseline methods are significantly constrained in their ability to balance these two factors effectively.

Second, PCM consistently demonstrates superior performance compared to the baselines across all the inconsistency measures. Both DSp-SC and PCQ either incur substantially higher inconsistencies or fail to maintain acceptable objective performance levels.

Third, while DSp-SC occasionally achieves favorable consistency, i.e., small $\mathrm{IC}^{\mathrm{SC}}_{*}$, this typically comes at the cost of severely degraded clustering quality, i.e., higher $\mathrm{Obj}^{\mathrm{SC}}$, as evidenced by the tradeoff curves extending toward the bottom-right region of the plots. Such a tradeoff indicates that while DSp-SC may maintain cluster assignments more consistently across incremental updates, it fundamentally compromises the overall clustering quality.

In summary, the failure modes of the baseline methods

underscore the critical advantage of PCM in achieving consistent and high quality clustering.

7 CONCLUSIONS

We showed that the subspace-preserving regularizer (1) is helpful for both online PCA and spectral clustering to better balance the quality and consistency. Additionally, we empirically demonstrate their superiority over baseline methods using real-world data.

Potential open problems include establishing lower bounds on the trade-off between approximation ratio and consistency. Another interesting question is whether a consistent k-means clustering algorithm can be designed for spectral clustering, in which the positions of existing points are perturbed at each step.

Acknowledgements

We thank the reviewers for the helpful comments that greatly improved our paper. SH and YY were supported by JSPS KAKENHI Grant Number JP24K02903. SH was supported in part by JSPS KAKENHI Grant Number JP23K28146.

References

Emmanuel Abbe. Community detection and stochastic block models: recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018.

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1027–1035, 2007.

Sayan Bhattacharya, Niv Buchbinder, Roie Levin, and Thatchaphol Saranurak. Chasing positive bodies. In *Proceedings of the IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1694–1714, 2023.

Yun Chi, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle L Tseng. On evolutionary spectral clustering. *ACM Transactions on Knowledge Discovery from Data*, 3(4):1–30, 2009.

Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

Jing Dong and Yuichi Yoshida. A batch-to-online transformation under random-order model. *Advances in Neural Information Processing Systems*, 36, 2024.

Hendrik Fichtenberger, Silvio Lattanzi, Ashkan Norouzi-Fard, and Ola Svensson. Consistent k-clustering for general metrics. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2660–2678, 2021.

Xiangyu Guo, Janardhan Kulkarni, Shi Li, and Jiayi Xian. Consistent k-median: Simpler, better and robust. In *International Conference on Artificial Intelligence and Statistics*, pages 1135–1143, 2021.

Anupam Gupta and Roie Levin. Fully-dynamic submodular cover with bounded recourse. In *Proceedings of the IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1147–1157, 2020.

Anupam Gupta, Ravishankar Krishnaswamy, Amit Kumar, and Debmalya Panigrahi. Online and dynamic algorithms for set cover. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 537–550, 2017.

Satoshi Hara and Yuichi Yoshida. Average sensitivity of decision tree learning. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

Ian T Jolliffe and Jorge Cadima. Principal component analysis: A review and recent developments. *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

Steinar Laenen and He Sun. Dynamic spectral clustering with provable approximation guarantee. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Silvio Lattanzi and Sergei Vassilvitskii. Consistent k-clustering. In *International Conference on Machine Learning*, pages 1975–1984. PMLR, 2017.

Gilad Lerman and Tyler Maunu. An overview of robust subspace recovery. *Proceedings of the IEEE*, 106(8): 1380–1410, 2018.

Jakub Łącki, Bernhard Haeupler, Christoph Grunau, Rajesh Jayaram, and Václav Rozhoň. Fully dynamic consistent k-center clustering. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3463–3484, 2024.

N. Locantore, J. S. Marron, D. G. Simpson, N. Tripoli, J. T. Zhang, and K. L. Cohen. Robust principal component analysis for functional data. *Test*, 8:1–73, 1999.

Jiazhong Nie, Wojciech Kotlowski, and Manfred K Warmuth. Online PCA with optimal regret. *Journal of Machine Learning Research*, 17(173):1–49, 2016.

Pan Peng and Yuichi Yoshida. Average sensitivity of spectral clustering. In *Proceedings of the 26th ACM*

SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD), pages 1132–1140, 2020.

Badrul Sarwar, George Karypis, Joseph Konstan, John Riedl, et al. Application of dimensionality reduction in recommender system-a case study. In *ACM WebKDD Workshop*, volume 1625, pages 285–295, 2000.

Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31 (1):1–10, 1966.

He Sun and Luca Zanetti. Distributed graph clustering and sparsification. *ACM Transactions on Parallel Computing*, 6(3):1–23, 2019.

Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

Nithin Varma and Yuichi Yoshida. Average sensitivity of graph algorithms. *SIAM Journal on Computing*, 52(4):1039–1081, 2023.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007.

Kevin S Xu, Mark Kliger, and Alfred O Hero III. Adaptive evolutionary clustering. *Data Mining and Knowledge Discovery*, 28:304–336, 2014.

Yuichi Yoshida and Shinji Ito. Average sensitivity of euclidean k-clustering. *Advances in Neural Information Processing Systems*, 35:32487–32498, 2022.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Appendix

A Proofs

A.1 Proof of Lemma 3.1

Proof. Let $\hat{\boldsymbol{w}}_1^{(t)}, \dots, \hat{\boldsymbol{w}}_k^{(t)} \in \mathbb{R}^p$ be the top-k eigenvectors of $C^{(t)}$, and let $\hat{W}^{(t)} \in \mathbb{R}^{p \times k}$ be the matrix whose i-th column is $\hat{w}_i^{(t)}$. Note that $\{\hat{\boldsymbol{w}}_1^{(t)}, \dots, \hat{\boldsymbol{w}}_k^{(t)}\}$ is orthonormal. Then, we have

$$
\begin{aligned}
&\sum_{i=1}^{k} \langle \boldsymbol{w}_i^{(t)}, C^{(t)} \boldsymbol{w}_i^{(t)} \rangle \\
&\geq \sum_{i=1}^{k} \langle \boldsymbol{w}_i^{(t)}, C^{(t)} \boldsymbol{w}_i^{(t)} \rangle - \frac{\lambda}{2} \|\Pi_{W^{(t-1)}} - \Pi_{W^{(t)}}\|_F^2 \\
&\geq \sum_{i=1}^{k} \langle \hat{\boldsymbol{w}}_i^{(t)}, C^{(t)} \hat{\boldsymbol{w}}_i^{(t)} \rangle - \frac{\lambda}{2} \|\Pi_{W^{(t-1)}} - \Pi_{\hat{W}^{(t)}}\|_F^2 && \text{(Optimality of } \tilde{W}^{(t)} \text{ and } W^{(t)}\text{)} \\
&= \sum_{i=1}^{k} \langle \hat{\boldsymbol{w}}_i^{(t)}, C^{(t)} \hat{\boldsymbol{w}}_i^{(t)} \rangle - \lambda (\|\Pi_{W^{(t-1)}}\|_F^2 + \|\Pi_{\hat{W}^{(t)}}\|_F^2) \\
&\geq \sum_{i=1}^{k} \langle \hat{\boldsymbol{w}}_i^{(t)}, C^{(t)} \hat{\boldsymbol{w}}_i^{(t)} \rangle - O(\lambda k). && \square
\end{aligned}
$$

A.2 Proof of Lemma 3.2

Proof. We first prove the second inequality of the lemma. From the optimality of $W^{(t)}$, we have

$$
\begin{aligned}
&\mathrm{tr}\langle W^{(t)}, C^{(t)} W^{(t)} \rangle - \frac{\lambda}{2} \|\Pi_{W^{(t)}} - \Pi_{W^{(t-1)}}\|_F^2 \\
&\geq \mathrm{tr}\langle W^{(t-1)}, C^{(t)} W^{(t-1)} \rangle - \frac{\lambda}{2} \|\Pi_{W^{(t-1)}} - \Pi_{W^{(t-1)}}\|_F^2 \\
&= \mathrm{tr}\langle W^{(t-1)}, C^{(t)} W^{(t-1)} \rangle,
\end{aligned}
$$

which implies

$$
\begin{aligned}
&\mathrm{tr}\langle W^{(t)}, C^{(t)} W^{(t)} \rangle - \mathrm{tr}\langle W^{(t-1)}, C^{(t)} W^{(t-1)} \rangle \\
&\geq \frac{\lambda}{2} \|\Pi_{\langle W^{(t)}} - \Pi_{W^{(t-1)}}\|_F^2.
\end{aligned}
$$

Summing up this inequality for $t = 2$ to n, we obtain

$$
\begin{aligned}
&\sum_{t=2}^{n} \left(\mathrm{tr}\langle W^{(t)}, C^{(t)} W^{(t)} \rangle - \mathrm{tr}\langle W^{(t-1)}, C^{(t)} W^{(t-1)} \rangle \right) \\
&\geq \frac{\lambda}{2} \sum_{t=2}^{n} \|\Pi_{W^{(t)}} - \Pi_{W^{(t-1)}}\|_F^2.
\end{aligned}
$$

Noting that $\mathrm{tr}\langle W^{(1)}, C^{(2)} W^{(1)} \rangle \geq 0$ because $C^{(2)}$ is positive semidefinite, this implies

$$
\mathrm{tr}\langle W^{(n)}, C^{(n)} W^{(n)} \rangle \geq \frac{\lambda}{2} \sum_{t=2}^{n} \|\Pi_{W^{(t)}} - \Pi_{W^{(t-1)}}\|_F^2.
$$

Rearranging the inequality, we obtain the second inequality.

We now prove the first inequality of the lemma. By von Neumann's trace inequality, we have

$$\max_R \mathrm{tr}\langle W^{(t)}R, W^{(t-1)}\rangle = \sum_{i=1}^k \sigma_i,$$

where σ_i is the i-th largest singular value of $\langle W^{(t)}, W^{(t-1)}\rangle$. Note that the maximizer is equal to $R^{(t)}$ in the algorithm. Also we have $\sigma \in [0,1]$ because for any $x \in \mathbb{R}^k$, we have

$$\|\langle W^{(t)}, W^{(t-1)}\rangle x\| \le \|(W^{(t)})^T\| \cdot \|W^{(t-1)}\| \cdot \|x\| \le \|x\|.$$

Note that

$$\begin{aligned}\sum_{i=1}^k \|\boldsymbol{w}_i^{(t)} - \boldsymbol{w}_i^{(t-1)}\|^2 &= 2k - 2\sum_{i=1}^k \langle \boldsymbol{w}_i^{(t)}, \boldsymbol{w}_i^{(t-1)}\rangle \\ &= 2k - 2\mathrm{tr}(W^{(t)}R^{(t)}, W^{(t-1)}) = 2k - 2\sum_{i=1}^k \sigma_i.\end{aligned}$$

On the other hand, we have

$$\begin{aligned}&\|\Pi_{W^{(t)}} - \Pi_{W^{(t-1)}}\|_F^2 \\ &= 2k - 2\mathrm{tr}\langle W^{(t)} \otimes W^{(t)}, W^{(t-1)} \otimes W^{(t-1)}\rangle \\ &= 2k - 2\mathrm{tr}(\langle W^{(t)}, W^{(t-1)}\rangle^\top \langle W^{(t)}, W^{(t-1)}\rangle) \\ &= 2k - 2\sum_{i=1}^k \sigma_i^2\end{aligned}$$

Then the claim holds because $\sigma_i \in [0,1]$ for every $i \in \{1, \dots, k\}$. □

A.3 Proof of Theorem 3.3

Proof. Consider the input sequence

$$n := \left\lfloor \frac{1}{4\varepsilon} \right\rfloor \quad \text{(assume } n \text{ is even)}, \qquad \boldsymbol{v}^{(t)} := \begin{cases} (1,0)^\top & t \text{ odd}, \\ (0,1)^\top & t \text{ even}, \end{cases}$$

Because every $\boldsymbol{v}^{(t)}$ is axis-aligned, each $C^{(t)}$ is diagonal:

$$C^{(t)} = \begin{pmatrix} o_t & 0 \\ 0 & e_t \end{pmatrix}, \text{ where } o_t := \left\lceil \frac{t}{2} \right\rceil \text{ and } e_t := \left\lfloor \frac{t}{2} \right\rfloor.$$

Hence $\lambda_t := \max_{\|\boldsymbol{w}\|=1} \boldsymbol{w}^\top C^{(t)} \boldsymbol{w} = \max\{o_t, e_t\}$. As n is even, we have $\lambda_n = e_n = n/2$.

Write $\boldsymbol{w}^{(t)} = (\cos\theta_t, \sin\theta_t)^\top$. Then, we have

$$\lambda_t - \langle \boldsymbol{w}^{(t)}, C^{(t)}\boldsymbol{w}^{(t)}\rangle = \begin{cases} (o_t - e_t)\sin^2\theta_t = \sin^2\theta_t & (t \text{ odd}), \\ (e_t - o_t)\cos^2\theta_t = \cos^2\theta_t & (t \text{ even}). \end{cases}$$

The multiplicative guarantee yields $\lambda_t - \langle \boldsymbol{w}^{(t)}, C^{(t)}\boldsymbol{w}^{(t)}\rangle \le \varepsilon\lambda_n$. Using $\lambda_n = \frac{n}{2} = \frac{1}{8\varepsilon}$ we obtain

$$\sin^2\theta_t \le \tfrac{1}{8} \quad (t \text{ odd}), \qquad \cos^2\theta_t \le \tfrac{1}{8} \quad (t \text{ even}). \tag{7}$$

Define

$$\delta := \arcsin\left(\tfrac{1}{\sqrt{8}}\right) \approx 0.361$$

Conditions (7) say

$$\theta_t \in [-\delta, \delta] \cup [\pi - \delta, \pi + \delta], \quad \theta_{t+1} \in \left[\tfrac{\pi}{2} - \delta, \tfrac{\pi}{2} + \delta\right] \cup \left[-\tfrac{\pi}{2} - \delta, -\tfrac{\pi}{2} + \delta\right].$$

Thus consecutive admissible sectors are separated by the angle $\phi := \frac{\pi}{2} - 2\delta$. Note that $\cos\phi = \sin(2\delta) = \frac{\sqrt{7}}{4}$. For two unit vectors whose angular difference is ϕ, the squared Euclidean distance is

$$2\big(1 - \cos\phi\big) = 2\left(1 - \frac{\sqrt{7}}{4}\right) = 2 - \frac{\sqrt{7}}{2}.$$

Hence each transition $t-1 \to t$ ($t \geq 2$) contributes at least $2 - \frac{\sqrt{7}}{2}$ to $\|\boldsymbol{w}^{(t)} - \boldsymbol{w}^{(t-1)}\|^2$.
The stream alternates parity in every step, so the lower bound applies to all $n-1$ transitions. Hence, we have

$$\sum_{t=2}^{n} \big\|\boldsymbol{w}^{(t)} - \boldsymbol{w}^{(t-1)}\big\|^2 \geq (n-1)\left(2 - \frac{\sqrt{7}}{2}\right) = \Omega(n) = \Omega\left(\frac{1}{\varepsilon}\right). \qquad \square$$

A.4 Proof of Lemma 4.2

Proof. Let $W^{(m)}$ be the matrix whose i-th column is $\boldsymbol{w}_i^{(m)}$. Then by Lemmas 3.2 and 4.1, we have

$$\begin{aligned}
&\sum_{t=1}^{m}\sum_{i=1}^{k} \|\boldsymbol{w}_i^{(t)} - \boldsymbol{w}_i^{(t-1)}\|^2 \leq \frac{\mathrm{tr}\langle W^{(m)}, L^{(m)} W^{(m)}\rangle}{\lambda} \\
&\leq \frac{1}{\lambda}\left(\sum_{i=1}^{k} \lambda_i(L^{(m)}) + O(\lambda k)\right) = \frac{1}{\lambda}\sum_{i=1}^{k} \lambda_i(L^{(m)}) + O(k).
\end{aligned}$$

The claim follows observing that $\sum_{v\in V} \|\boldsymbol{p}_v^{(t)} - \boldsymbol{p}_v^{(t-1)}\|^2 = \sum_{i=1}^{k} \|\boldsymbol{w}_i^{(t)} - \boldsymbol{w}_i^{(t-1)}\|^2$. $\square$

A.5 Proof of Lemma 4.3

Proof. Let $\delta^{(t)} = \sqrt{\sum_{v\in V} \|\boldsymbol{p}_v^{(t)} - \boldsymbol{p}_v^{(t-1)}\|^2}$. Note that $\sum_{t=1}^{m} (\delta^{(t)})^2 \leq \lambda^{-1} \sum_{i=1}^{k} \lambda_i(L^{(m)}) + O(k)$ by Lemma 4.2. Let $C_i^{(t)} = \{c_1^{(t)}, \ldots, c_i^{(t)}\}$, $P_i^{(t)} = \{\boldsymbol{p}_v : v \in C_i^{(t)}\}$, and $Z_i^{(t)} = \sum_{u\in V} d(\boldsymbol{p}_u^{(t)}, P_i^{(t)})$. Then, we have

$$\Pr[c_i^{(t)} = v] = \frac{d(\boldsymbol{p}_v^{(t)}, P_{i-1}^{(t)})}{Z_{i-1}^{(t)}}.$$

Conditioned on $C_i^{(t)} = C_i^{(t-1)}$, we have

$$\begin{aligned}
& d(\boldsymbol{p}_u^{(t)}, P_i^{(t)}) = d(\boldsymbol{p}_u^{(t-1)}, P_i^{(t-1)}) \pm \delta^{(t)}, \\
\Rightarrow \quad & \sum_{u\in V} d(\boldsymbol{p}_u^{(t)}, P_i^{(t)}) = \sum_{u\in V} d(\boldsymbol{p}_u^{(t-1)}, P_i^{(t-1)}) \pm \delta^{(t)} n,
\end{aligned}$$

where $a = b \pm c$ is a shorthand for $b - c \leq a \leq b + c$. For a set of $i-1$ vertices $C \subseteq V$, let $P_C^{(t)} = \{\boldsymbol{p}_v^{(t)} : v \in C\}$, and $Z_C^{(t)} = \sum_{u\in V} d(\boldsymbol{p}_u^{(t)}, P_C^{(t)})$. Note that $Z_C^{(t)} \geq \mathrm{OPT}_i^{(t)}$, where $\mathrm{OPT}_i^{(t)}$ is the optimal value of the i-median problem for $P^{(t)} = \{\boldsymbol{p}_v : v \in V\}$.

Then for any $i \geq 2$, we have

$$\begin{aligned}
&\Pr[c_i^{(t)} = v \mid C_{i-1}^{(t)} = C] - \Pr[c_i^{(t-1)} = v \mid C_{i-1}^{(t-1)} = C] \\
&= \frac{d(\boldsymbol{p}_v^{(t)}, P_C^{(t)})}{Z_C^{(t)}} - \frac{d(\boldsymbol{p}_v^{(t-1)}, P_C^{(t-1)})}{Z_C^{(t-1)}} \\
&= \frac{Z_C^{(t-1)} d(\boldsymbol{p}_v^{(t)}, P_C^{(t)}) - Z_C^{(t)} d(\boldsymbol{p}_v^{(t-1)}, P_C^{(t-1)})}{Z_C^{(t)} \cdot Z_C^{(t-1)}} \\
&= \frac{1}{Z_C^{(t)} \cdot Z_C^{(t-1)}} \Big(Z_C^{(t-1)} (d(\boldsymbol{p}_v^{(t-1)}, P_C^{(t-1)}) \pm \delta^{(t)}) \\
&\qquad - (Z_C^{(t-1)} \pm \delta^{(t)} n) d(\boldsymbol{p}_v^{(t-1)}, P_C^{(t-1)}) \Big) \\
&= \frac{\pm Z_C^{(t-1)} \delta^{(t)} \pm n d(\boldsymbol{p}_v^{(t-1)}, P_C^{(t-1)}) \delta^{(t)}}{Z_C^{(t)} \cdot Z_C^{(t-1)}}.
\end{aligned}$$

Then we have

$$\begin{aligned}
&\sum_{v \in V} \Big| \Pr[c_i^{(t)} = v \mid C_{i-1}^{(t)} = C] - \Pr[c_i^{(t-1)} = v \mid C_{i-1}^{(t-1)} = C] \Big| \\
&\leq \frac{n Z_C^{(t-1)} \delta^{(t)} + n Z_C^{(t-1)} \delta^{(t)}}{Z_C^{(t)} \cdot Z_C^{(t-1)}} = O\left(\frac{n \delta^{(t)}}{\mathrm{OPT}_{i-1}^{(t)}} \right).
\end{aligned}$$

Observing that $\mathrm{TV}(C_1^{(t)}, C_1^{(t-1)}) = 0$, by induction on i, we can show that $\mathrm{TV}(C^{(t)}, C^{(t-1)}) = \sum_{i=2}^{k} O(n\delta^{(t)}/\mathrm{OPT}_{i-1}^{(t)})$. Hence, we have

$$\begin{aligned}
&\sum_{t=1}^{m} \mathrm{TV}(C^{(t)}, C^{(t-1)}) = O\left(\sum_{t=1}^{m} \sum_{i=2}^{k} \frac{n\delta^{(t)}}{\mathrm{OPT}_{i-1}^{(t)}} \right) \\
&= O\left(kn \sum_{t=1}^{m} \frac{\delta^{(t)}}{\mathrm{OPT}^{(t)}} \right) \\
&= O\left(kn \sqrt{\sum_{t=1}^{m} (\delta^{(t)})^2} \sqrt{\sum_{t=1}^{m} \left(\frac{1}{\mathrm{OPT}^{(t)}} \right)^2} \right) && \text{(Cauchy-Schwarz)} \\
&= O\left(kn \sqrt{\sum_{i=1}^{k} \frac{\lambda_i(L^{(m)})}{\lambda} + O(k)} \sqrt{\sum_{t=1}^{m} \left(\frac{1}{\mathrm{OPT}^{(t)}} \right)^2} \right).
\end{aligned}$$

□

A.6 Inconsistency of Spectral Clustering with k-Means Clustering

In Algorithm 2, we solved the k-median problem using D^1-sampling. One can use D^2-sampling to solve the k-means problem as well. Algorithm 3 shows the D^2-sampling version of PCM. For this version of PCM, we have the following inconsistency guarantee, which is worse than Lemma 4.3 by the factor $\sqrt{k}$.

Lemma A.1. *Let $C^{(t)} = \{c_1^{(t)}, \ldots, c_k^{(t)}\}$ and $\mathrm{OPT}^{(t)}$ be the optimal value of the k-means problem for $P^{(t)} = \{\boldsymbol{p}_v^{(t)} : v \in V\}$. Then, we have*

$$\begin{aligned}
&\sum_{t=1}^{m} \mathrm{TV}(C^{(t)}, C^{(t-1)}) \\
&= O\left(k\sqrt{k} n \sqrt{\sum_{i=1}^{k} \frac{\lambda_i(L^{(m)})}{\lambda} + O(k)} \sqrt{\sum_{t=1}^{m} \left(\frac{1}{\mathrm{OPT}^{(t)}} \right)^2} \right).
\end{aligned}$$

Algorithm 3: Preserving Cluster Membership with D^2-Sampling

Procedure D^2-SAMPLING$(V, \{\boldsymbol{p}_v\}_{v\in V})$
  Sample $c_1 \in V$ uniformly at random;
  for $i = 2, \dots, k$ **do**
    Let $\mathcal{D}$ be the distribution over V such that $v \in V$ is sampled with probability $\propto d(\boldsymbol{p}_v, \{\boldsymbol{p}_{c_1}, \dots, \boldsymbol{p}_{c_{i-1}}\})^2$;
    Sample $c_i \in V$ from $\mathcal{D}$;
  return $\{c_1, \dots, c_k\}$.
Procedure PCM-$D^2(V)$
  $\boldsymbol{w}_1^{(0)} \leftarrow \mathbf{0}, \dots, \boldsymbol{w}_k^{(0)} \leftarrow \mathbf{0}$;
  for $t = 1, \dots, m$ **do**
    Receive $e^{(t)} \in \binom{V}{2}$;
    Let $L^{(t)} \in \mathbb{R}^{V\times V}$ be the Laplacian of the graph $G^{(t)} = (V, \{e^{(1)}, \dots, e^{(t)}\})$;
    Compute $\boldsymbol{w}_1^{(t)}, \dots, \boldsymbol{w}_1^{(t)}$ by solving (6);
    $\boldsymbol{p}_v^{(t)} \leftarrow (\boldsymbol{w}_1^{(t)}(v), \dots, \boldsymbol{w}_k^{(t)}(v))$ for $v \in V$;
    $\{c_1^{(t)}, \dots, c_k^{(t)}\} \leftarrow D^2$-SAMPLING$(V, \{\boldsymbol{p}_v^{(t)}\}_{v\in V})$;
    Compute the partition $(V_1^{(t)}, \dots, V_k^{(t)})$ of V using $c_1^{(t)}, \dots, c_k^{(t)}$.

Proof. The proof is almost identical to that of Lemma 4.3. Below, we discuss the required changes to handle k-means.

First, conditioned on $C_i^{(t)} = C_i^{(t-1)}$, we have

$$\begin{aligned} &d(\boldsymbol{p}_u^{(t)}, P_i^{(t)})^2 - d(\boldsymbol{p}_u^{(t-1)}, P_i^{(t-1)})^2 = \left(d(\boldsymbol{p}_u^{(t)}, P_i^{(t)}) - d(\boldsymbol{p}_u^{(t-1)}, P_i^{(t-1)})\right) \cdot \left(d(\boldsymbol{p}_u^{(t)}, P_i^{(t)}) + d(\boldsymbol{p}_u^{(t-1)}, P_i^{(t-1)})\right) \\ &= O(\delta\sqrt{k}). \end{aligned}$$

This implies

$$\sum_{u\in V} d(\boldsymbol{p}_u^{(t)}, P_i^{(t)}) = \sum_{u\in V} d(\boldsymbol{p}_u^{(t-1)}, P_i^{(t-1)}) \pm O(\delta^{(t)}\sqrt{k}n),$$

Then we have

$$\sum_{v\in V} \left| \Pr[c_i^{(t)} = v \mid C_{i-1}^{(t)} = C] - \Pr[c_i^{(t-1)} = v \mid C_{i-1}^{(t-1)} = C] \right| = O\left(\frac{\sqrt{k}n\delta^{(t)}}{\mathrm{OPT}_{i-1}^{(t)}}\right).$$

and hence

$$\begin{aligned} \sum_{t=1}^{m} \mathrm{TV}(C^{(t)}, C^{(t-1)}) &= O\left(\sum_{t=1}^{m}\sum_{i=2}^{k} \frac{\sqrt{k}n\delta^{(t)}}{\mathrm{OPT}_{i-1}^{(t)}}\right) \\ &= O\left(k\sqrt{k}n \sum_{t=1}^{m} \frac{\delta^{(t)}}{\mathrm{OPT}^{(t)}}\right) \\ &= O\left(k\sqrt{k}n \sqrt{\sum_{t=1}^{m} (\delta^{(t)})^2} \sqrt{\sum_{t=1}^{m} \left(\frac{1}{\mathrm{OPT}^{(t)}}\right)^2}\right) && \text{(Cauchy-Schwarz)} \\ &= O\left(k\sqrt{k}n \sqrt{\sum_{i=1}^{k} \frac{\lambda_i(L^{(m)})}{\lambda} + O(k)} \sqrt{\sum_{t=1}^{m} \left(\frac{1}{\mathrm{OPT}^{(t)}}\right)^2}\right). \end{aligned}$$

□

B Results of PCA

In the PCA experiments, we set the hyperparameter λ of CONSISTENTPCA, η of GD and MEG as follows. For CONSISTENTPCA, we varied λ from 10^0 to 10^6 in a logarithmic scale. For GD, we varied η from 10^{-5} to 10^0 in a

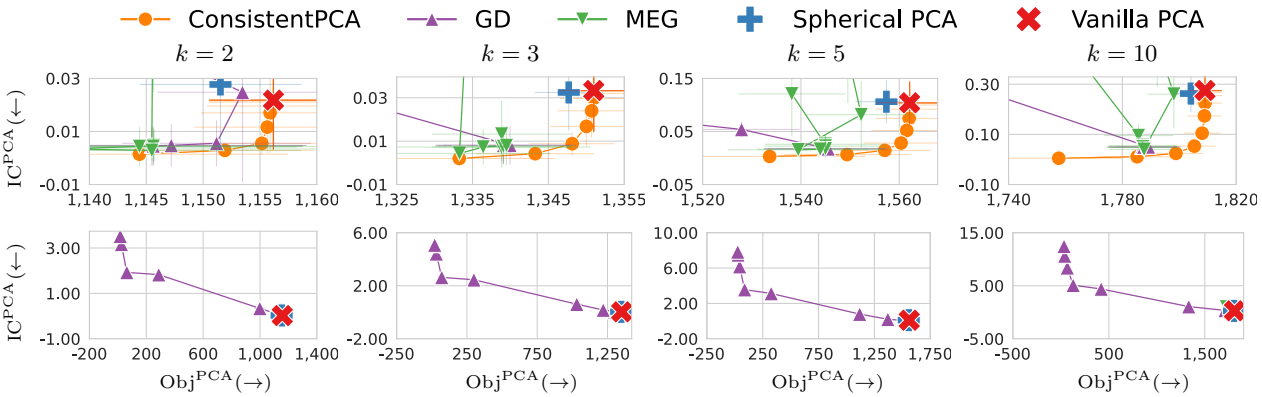

Figure 4: [PCA Results on Face Dataset] (top row) The zoomed plots appeared in the main body of the paper. (bottom row) The entire plots without zooming.

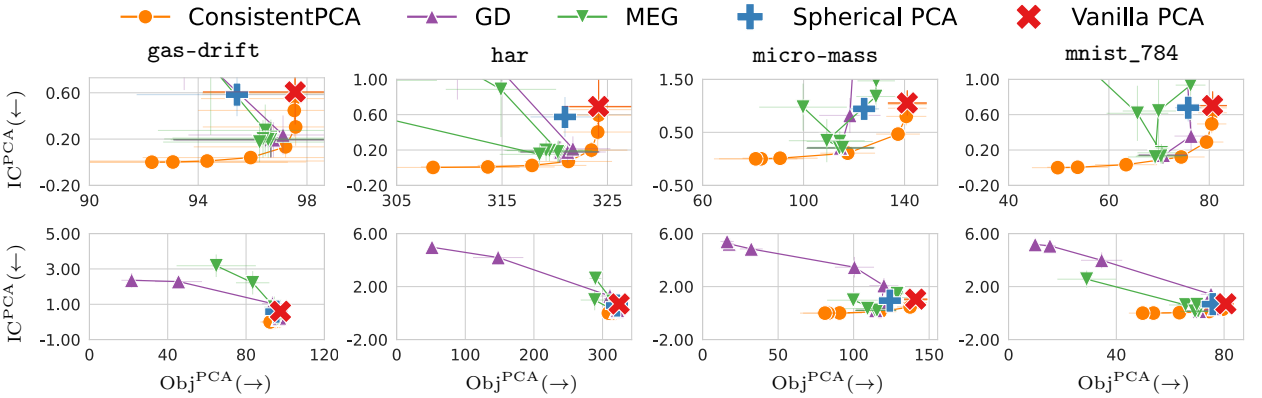

Figure 5: [PCA Results on OpenML Dataset with $k = 3$] (top row) The zoomed plots appeared in the main body of the paper. (bottom row) The entire plots without zooming.

logarithmic scale. For MEG, we varied η from 10^{-5} to 10^{-1} in a logarithmic scale.

In Figure 2, we only show the plots around the tradeoff curve of ConsistentPCA. Figure 4 and Figure 5 show the entire plots as well. The figures show that GD and MEG exhibit significantly worse results when η is not chosen appropriately.

All the codes and results are available at `github.com/sato9hara/consistent-pca-sc`.

C Results of Spectral Clustering

In the Spectral Clustering experiments, we set the hyperparameter λ of PCM, α of PCQ, and `sampling_constant` of DSp-SC as follows. For PCM, we varied λ from 10^{-2} to 10^{1} in a logarithmic scale. For PCQ, we varied α in $0.0, 0.1, 0.2, \ldots, 0.9, 1.0$. For DSp-SC, we varied `sampling_constant` in $5, 7, 9, \ldots, 23, 25$.

Figure 6 shows the results on the three remaining graphs (id = 1912, 3437, 348) of the facebook dataset experiments with the number of clusters set to $k = 3$. Similar to Figure 3, PCM realizes a wide range of tradeoffs between $\mathrm{Obj}^{\mathrm{SC}}$ and $\mathrm{IC}^{\mathrm{SC}}_{\mathrm{embed}}$; PCM tends to outperform the baselines; and DSp-SC can be a favorable choice in terms of consistency, which is however significantly worse in terms of the SC objective.

All the codes and results are available at `github.com/sato9hara/consistent-pca-sc`.

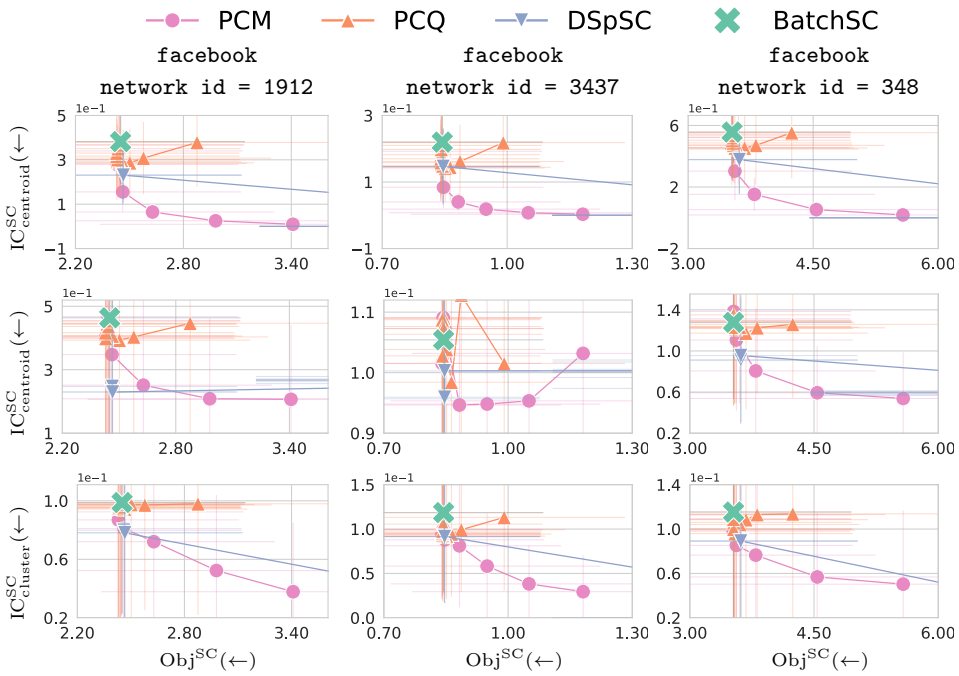

Figure 6: Results of Spectral Clustering ($k = 3$) on the facebook datasets with id = 1912, 3437, and 348.

D Synthetic Data Experiments with Stochastic Block Model

We generated synthetic graphs using Stochastic Block Model (SBC) (Abbe, 2018) to evaluate the performance of PCM for various graph sizes. The synthetic graph is generated by SBC with k-blocks. The nodes in each block are connected by an edge with probability 0.6, and the nodes in different blocks are are connected with probability 0.2. In each time step, one edge is added in a block with probability 0.8 and an edge between blocks is added otherwise. We generated graphs with the number of nodes 50, 100, 500, and 1000.

Figure 7[6] shows the results on the SBM experiments for the case when we set the number of nodes $|V|$ to $50, 100, 500$, and 1000, the number of clusters to $k = 5$, and the number of updates to $m = 100$. In the experiments, we varied the regularization λ of PCM, α of PCQ, and sampling parameter of DSp-SC. We repeated the experiments 100 times and plotted the averages of $\mathrm{Obj}^{\mathrm{SC}}$, $\mathrm{IC}^{\mathrm{SC}}_{\mathrm{embed}}$, $\mathrm{IC}^{\mathrm{SC}}_{\mathrm{centroid}}$, and $\mathrm{IC}^{\mathrm{SC}}_{\mathrm{cluster}}$ of each method in the figures.

There are two implications in the figures. First, PCM could realize a wide rage of tradeoffs between $\mathrm{Obj}^{\mathrm{SC}}$ and $\mathrm{IC}^{\mathrm{SC}}_{\mathrm{embed}}$ which naturally coincides with vanilla SC in the limit $\lambda \to 0$. Second, PCM tends to outperform vanilla SC in terms of inconsistencies $\mathrm{IC}^{\mathrm{SC}}_{\mathrm{embed}}$, $\mathrm{IC}^{\mathrm{SC}}_{\mathrm{centroid}}$, and $\mathrm{IC}^{\mathrm{SC}}_{\mathrm{cluster}}$. This result shows that, with appropriately chosen λ, practitioners can enjoy smooth evolution of spectral clustering with PCM every time new edges are observed.

To assess the clustering qualities with respect to the ground truth clusters, we computed Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) to assess the quality of the clustering results. Table 1 shows the average scores over 100 repeated experiments. As shown in the tables, there are only tiny differences between the methods in ARI and NMI, irrespective of the choice of regularization weights λ for PCM and α for PCQ. This demonstrates that PCM with the subspace-preserving regularizer can improve consistency without compromising the quality of the clustering structure relative to the ground truth.

[6]All the codes and results are available at `github.com/sato9hara/consistent-pca-sc`.

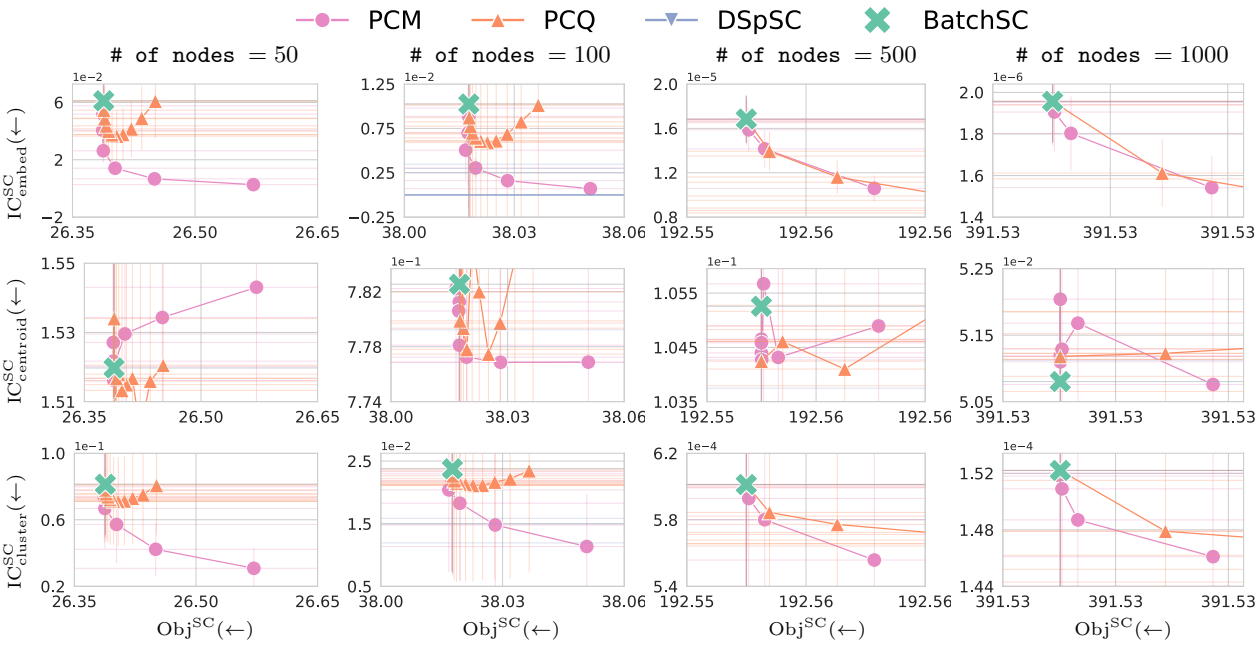

Figure 7: Results of Spectral Clustering ($k = 5$) on the SBM datasets. The error bars denote standard deviations. The arrows in the axis labels indicate favorable directions.

BatchSC	
$\mathrm{IC}^{\mathrm{SC}}_{\mathrm{cluster}}$	0.024
$\mathrm{Obj}^{\mathrm{SC}}$	38.018
ARI	0.500
NMI	0.638

PCM	$\lambda = 0.00$	0.01	0.03	0.10	0.32	1.00	3.16	10.00
$\mathrm{IC}^{\mathrm{SC}}_{\mathrm{cluster}}$	0.024	0.023	0.023	0.022	0.020	0.018	0.015	0.011
$\mathrm{Obj}^{\mathrm{SC}}$	38.018	38.018	38.018	38.017	38.017	38.019	38.028	38.051
ARI	0.500	0.500	0.500	0.500	0.500	0.500	0.503	0.505
NMI	0.638	0.638	0.638	0.638	0.638	0.639	0.640	0.641

PCQ	$\alpha = 0.0$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$\mathrm{IC}^{\mathrm{SC}}_{\mathrm{cluster}}$	0.023	0.022	0.022	0.021	0.021	0.021	0.021	0.022	0.022	0.023	0.024
$\mathrm{Obj}^{\mathrm{SC}}$	38.037	38.032	38.028	38.025	38.023	38.021	38.020	38.019	38.018	38.018	38.018
ARI	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500
NMI	0.638	0.638	0.638	0.638	0.638	0.638	0.638	0.638	0.638	0.638	0.638

Table 1: Results of Spectral Clustering ($k = 5$) on the SBM datasets. We computed Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) to assess the quality of the clustering results, averaged over 100 repeated experiments.