基于正交匹配追踪的可扩展稀疏子空间聚类

摘要

本文对《Scalable sparse subspace clustering by orthogonal matching pursuit》进行研究和复现。这篇论文提出的基于正交匹配追踪的可扩展稀疏子空间聚类 (SSC - OMP) 算法旨在解决传统基于 ℓ_1 优化的子空间聚类算法计算复杂问题，其通过特定优化问题利用正交匹配追踪 (OMP) 寻找稀疏表示。理论上，在子空间独立、充分分离或随机等多种情况下，能给出子空间保持性的自表达系数矩阵。复现实验使用合成数据集和手写数字数据集，对比 SSC - BP 等算法，结果表明 SSC - OMP 虽在子空间保持表示误差和连通性方面可能逊于 SSC - BP，但随着数据点密度增加，聚类精度差距缩小，且在大规模数据处理上速度优势显著，在准确性和效率间实现良好平衡，为高维数据聚类提供有效工具，不过在处理超高维复杂数据时仍有改进空间。

关键词：子空间聚类；SSC - OMP；稀疏子空间

1 引言

子空间聚类作为一种有效的数据分析工具，在计算机视觉、机器学习和信号处理等众多领域都有广泛的应用。比如在图像识别中，通过子空间聚类可以将不同类别的图像进行区分，有助于提高识别准确率；在信号处理中，能够对不同特征的信号进行分类和处理。来自不同类别的数据可以通过低维子空间的并集来近似表示，在这种情况下，任务就是根据数据点在子空间中的关系，将数据聚类成组，使得每个组只包含来自同一子空间的数据点。

在过去的十年中，这个问题受到了极大的关注，并开发出了许多子空间聚类算法，包括基于迭代、代数、统计和谱聚类的方法。在现有技术中，基于对通过求解包含 ℓ_1、ℓ_2 或核规范正则化的优化问题而得到的亲和矩阵应用谱聚类的方法得到了广泛关注，其简单性、理论正确性和经验上的成功而极为流行。这些方法一般包括两个基本步骤:(1) 从数据中学习一个亲和矩阵;(2) 基于这个亲和矩阵做谱聚类 [5]。自表达 (Self-Expression) 模型 [3] 是子空间聚类中用于计算亲和矩阵的有效工具。给定数据矩阵 $X = [x_1, \ldots, x_N] \in \mathbb{R}^{D\times N}$, 其列向量对应的数据点采样自 n 个子空间中。自表达模型把一个数据点 x_j 用其他数据点的线性组合来表示:

$$x_j = Xc_j + e_j, \quad c_{jj} = 0. \tag{1}$$

其中 c_j 表示数据点 x_j 所对应的自表达系数。

对于 c_j，方程1 可能没有唯一解，因此，为了得到具有特定性质的自表达系数，我们通常使用不同的正则化约束，比如 ℓ_1 范数，ℓ_2 范数，核范数等。对于一个数据点 x_j 来说，如果我们能找到一组系数 c_j 使得 $x_j = Xc_j$ 成立，这里 X 是所有数据点组成的数据矩阵，并且

c_j 的第 j 个元素为 0（即 $c_{jj} = 0$），则称 c_j 为 x_j 的一个表达向量。若此表达向量中的非零元素仅对应于与 x_j 同属一个子空间的数据点，则称该表达是子空间保持的。简单来说，就是在这个表示中，非零系数所对应的点 x 都在点所属的子空间 $\mathbb{R}^{D\times N}$ 中。如果得到自表达系数且系数具有子空间保持性质，那么在构建亲和矩阵 W(其中元素 $w_{ij} = |c_{ij}| + |c_{ij}|$) 时，就可以使得亲和矩阵更准确地反应子空间内和子空间之间的关系。亲和矩阵也是后续进行谱聚类等操作的重要基础，准确的亲和矩阵能够提高聚类的准确性。然而，只有当子空间独立且数据未被破坏时，所得到的系数矩阵才是子空间保留的。因此，我们需要一种既能在广泛条件下保证子空间保留亲和性，又能提高计算效率的方法。

在复现的论文中，Chong You 等人研究了基于自表达性的子空间聚类方法，该方法使用正交匹配追踪（OMP）来寻找稀疏表示，以代替基于 ℓ_1 的基追踪（BP）方法。这种方法被称为 SSC-OMP，因为它与最初的 SSC（论文中称为 SSC-BP）具有亲缘关系。该论文的主要贡献在于找到了 SSC-OMP 所产生的亲和矩阵具有子空间保持性的理论条件，并证明了它在处理大规模问题时的效率。具体来说，它证明了 (1) 当子空间和数据是确定的时，如果子空间是独立的，或者如果子空间充分分离且数据分布良好，SSC-OMP 会给出一个子空间保持性的自表达系数矩阵 C。(2) 当子空间和数据都是均匀随机抽取时，如果子空间的维度相对于环境维度足够小，并且其系数受样本密度和子空间数量的控制，那么 SSC-OMP 就能给出一个子空间保持性的自表达系数矩阵 C。(3)SSC-OMP 比原始的 SSC-BP 快几个数量级，可处理多达 100,000 个数据点。所以复现工作主要在于 (1) 复现 SSC-OMP 算法。(2) 使用随机生成的合成数据集和手写数字数据集进行测试，并与其他算法进行效率对比。

2 相关工作

稀疏子空间聚类算法 (SSC) [11] 使用 ℓ_1 范数来追求自表达系数矩阵 C 的稀疏性，先前的研究表明，如果子空间是独立的 [3, 11]，或者如果来自不同子空间的数据满足某些分离条件，且来自同一子空间的数据分布良好 [2,3,9,14]，SSC 就能给出子空间保留解。类似的结果也适用于数据被噪声 [10, 12] 和异常值 [9] 干扰的情况。其他基于自我表达的方法对系数矩阵 C 采用不同的正则化处理。低秩表示（LRR）[6, 7] 和低秩子空间聚类（LRSC）[4, 7] 使用核规范最小化鼓励 C 为低秩。在此基础上，[8] 研究了 ℓ_1 和 ℓ_2 混合的正则化，[13] 提出了 ℓ_1 和核规范混合的正则化。具体来说，使用 ℓ_1 优化虽然可以提供良好的稀疏性和理论保证，但它涉及到求解大规模凸优化问题，这在实际应用中会非常耗时，尤其是在数据量很大的情况下。

与稀疏正则化 SSC 相比，ℓ_2 正则化 LSR 和核规范正则化 LRR 和 LRSC 的优势在于，C 的解可以通过（无噪声）数据矩阵 X 的 SVD 以封闭形式计算出来，因此它们在计算上更具吸引力。然而，只有当子空间独立且数据未被破坏时，所得到的 C 才是子空间保留的。因此，我们需要一种既能在广泛条件下保证子空间保留亲和性，又能提高计算效率的方法。

值得注意的是，[1] 中已经考虑过将正交匹配追踪（OMP）用于稀疏子空间聚类（SSC）这一想法。复现论文的工作的核心贡献在于，针对任意子空间的情况，为使亲和矩阵具有子空间保持性质提供了弱得多但更简洁且更易于解释的条件。特别地，复现论文提出的条件与基于基追踪（BP）的稀疏子空间聚类（SSC - BP）的条件存在自然联系，这揭示了这两种基于稀疏性的子空间聚类方法之间的关系。

此外，复现的论文中的实验结果对大规模问题下 SSC - OMP 的表现进行了更为详细的

评估。同时，[14] 也研究了 OMP 给出子空间保持性的条件。但是复现论文通过给出在确定性独立子空间模型、确定性任意子空间模型和随机子空间模型下的结果，对 OMP 在子空间聚类问题上进行了更为全面的研究。尤其是在确定性任意模型方面，得到的结果比 [14] 中的主要结果要强得多。

复现的论文将 SSC-OMP 的结果与使用 ℓ_1 范数作为正则的 SSC-BP [2, 3, 9] 的结果进行比较。我们还将我们的结果与 [1] 的 SSC-OMP 结果进行了比较。比较的标准是这些替代算法给出的解是否具有子空间保持性。

3 本文方法

3.1 算法实现

在实际应用中，当数据点的数量 N 很大时，对于 ℓ_1 的优化问题：

$$c_j^* = \arg\min_{c_j} \|c_j\|_1 \quad \text{s.t.} \quad x_j = X c_j,\ c_{jj} = 0. \tag{2}$$

求解 N 个关于 N 个变量 ℓ_1 的最小化问题可能是难以实现的。这是因为这个问题是一个大规模凸优化问题，随着 N 的增大，计算量会急剧增加，可能导致计算资源不足或计算时间过长等问题。复现的论文引入了一种替代方案，即考虑下面的优化问题：

$$c_j^* = \arg\min_{c_j} \|x_j - X c_j\|_2^2 \quad \text{s.t.} \quad \|c_j\|_0 \le k,\ c_{jj} = 0. \tag{3}$$

这个优化方法的目标是找到一个系数向量 c_j，使得 x_j 与 Xc_j 的 ℓ_2 范数平方最小，同时满足 ℓ_0 范数小于等于 k 且 c_j 的对角元素 $c_{jj} = 0$(即不允许数据点自身表示自身)。上述问题可以使用论文中提出的正交匹配追踪 (OMP) 算法来求解，OMP 算法通过利用前一轮迭代的残差来估计稀疏矩阵中非零元素的位置，并求解其值，为每个数据点找到一个稀疏表示，使得该表示仅由同一子空间中的其他数据点构成，从而实现子空间聚类。通过正交匹配追踪算法，逐步选择最能代表当前数据点的字典列，以构建稀疏表示，从而贪婪地解决 $\min_c \|Ac - b\|_2^2 \quad \text{s.t.} \quad \|c\|_0 \le k$ 问题。计算出 C^* 后，通过对亲和矩阵 $W = \|C^*\| + \|C^{*\top}\|$ 进行谱聚类，找到数据的分段。

Algorithm 1 Orthogonal Matching Pursuit (OMP)

Input: $A = [a_1, \ldots, a_M] \in \mathbb{R}^{m \times M}$, $b \in \mathbb{R}^m$, $k_{\max}$, ϵ.

1: Initialize $k = 0$, residual $q_0 = b$, support set $T_0 = \emptyset$.
2: **while** $k < k_{\max}$ and $\|q_k\|_2 > \epsilon$ **do**
3: $\quad T_{k+1} = T_k \cup \{i^*\}$, where $i^* = \arg\max_{i=1,\ldots,M} |a_i^T q_k|$.
4: $\quad q_{k+1} = (I - P_{T_{k+1}})b$, where $P_{T_{k+1}}$ is the projection onto the span of the vectors $\{a_j, j \in T_{k+1}\}$.
5: $\quad k \leftarrow k + 1$.
6: **end while**

Output: $c^* = \arg\min_{c:\text{Supp}(c) \subseteq T_k} \|b - Ac\|_2$.

Algorithm 2 Sparse Subspace Clustering by Orthogonal Matching Pursuit (SSC - OMP)

Input: Data $X = [x_1, \ldots, x_N]$, parameters $k_{\max}, \epsilon$.

1: Compute c_j^* from OMP(X_{-j}, x_j) using Algorithm 1.

2: Set $C^* = [c_1^*, \ldots, c_N^*]$ and $W = |C^*| + |C^{*\top}|$.

3: Compute segmentation from W by spectral clustering.

Output: Segmentation of data X.

3.2 理论部分

这部分主要是论文对基 SSC - OMP 算法进行理论分析。其目的是从理论层面研究在何种充分条件下，SSC - OMP 能够给出子空间保留表示。这是理解和评估 SSC - OMP 算法有效性的关键，因为子空间保留表示能够确保聚类结果准确反映数据点所属的子空间结构，从而实现有效的子空间聚类。例如，在图像聚类应用中，只有保证子空间保留，才能将属于同一类别的图像（如同一数字的手写图像或同一人的人脸图像）准确地聚在一起。这里只给出结论，不过多赘述。

分析假设数据是无噪声的，这是为了简化理论分析，先在理想数据情况下研究 SSC - OMP 的性能。在实际应用中，虽然数据往往存在噪声，但无噪声假设下的分析可以为后续处理噪声数据提供基础和参考。算法1中的终止参数 $\epsilon = 0$ 和 k_{max} 足够大。$\epsilon = 0$ 表示追求精确解，即希望残差尽可能小直至为 0；k_{max} 足够大则是为了给算法足够的迭代次数来寻找最优解，确保算法有足够的机会选择合适的列向量来构建稀疏表示。同时假设数据矩阵 X 的列向量都被归一化为单位 ℓ_2 范数。这样做的目的是使数据在某种程度上具有可比性，避免因向量长度差异过大而影响算法对向量重要性的判断。例如，在计算内积选择列向量时，归一化后的向量能更准确地反映其方向上的相关性，而不是被长度因素主导。

对于独立子空间，首先考虑子空间固定、数据点固定且子空间相互独立的情况。定义了独立子空间的概念（如两个子空间独立当且仅当它们仅在原点相交，且独立子空间集合的子集也独立），并证明了在此情况下 OMP 能够给出每个数据点的子空间保留表示。证明思路是基于独立子空间的性质，当计算 $OMP(X_{-j}, s_j)$ 时，由于 S_i 和 $S_{(-i)}$ 独立，与 x_j 不在同一子空间的数据点的系数最终会为 0。

对于任意确定性子空间，即子空间不必是独立或不相交的。通过定义 OMP 残差方向（W_j^i 和 W^i）和相干性等概念，得出 OMP 给出子空间保留表示的充分条件。直观上，这个条件要求子空间的残差方向与其他子空间数据点的点积小于与本子空间内除 x_j 外数据点的点积，意味着数据点应在子空间内分布良好且不同子空间数据点应充分分离（角度上）。同时，通过解释为子空间间相干性小（数据点充分分离）且子空间内点分布良好（最小内半径大），并与 SSC - BP 算法的相关条件进行了比较。

对于任意随机子空间，即子空间的基元素和数据点都是随机生成的情况。论文证明在一定条件下（子空间维度 d、环境维度 D、子空间数量 n 和子空间数据点密度 ρ 满足特定关系），OMP 以高概率给出子空间保留表示，并且随着数据点密度增加，条件更易满足，成功概率也增加。这部分分析展示了 SSC - OMP 在随机数据场景下的性能特点，为其在实际应用中的广泛适用性提供了理论支持。例如，在处理随机分布的传感器数据或大规模随机采样的数据时，该理论结果可用于判断 SSC - OMP 算法的有效性和可靠性。

3.3 评价指标

论文验证了 SSC-OMP 的理论结果，所以复现工作中通过使用随机模型对合成数据进行实验，将其与 SSC-BP 的结果进行比较。具体来说，论文中证明了即使子空间不是独立的，OMP 的解也具有子空间保持性的概率随数据点密度的增加而增加。其次，复现工作在对手写数字数据集进行聚类时测试了所提方法的性能，并以 KMeans、SSC-BP 方法为基线进行对比，得出结论：SSC-OMP 在准确性和效率之间实现了最佳权衡。

论文中的评价指标使用两个指标来评估子空间保持性的满足程度。第一个指标是直接衡量解是否为保持子空间。子空间保持表示的百分比（$p\%$）：这是表示为子空间保持的点的百分比。由于求解器不精确，绝对值小于 10^3 的系数被视为零。子空间保持解的 p 为 100%。不过，为了与输出结果通常不保留子空间的先进方法进行比较，第二个指标衡量的是系数与保留子空间的接近程度。子空间保持表示误差（$e\%$）：对于每个系数 c_j，我们计算其 ℓ_1 中来自其他子空间的部分，然后求所有 j 的平均值，其中 ω_{ij} $0, 1$ 是真正的亲和性。当自表示系数矩阵是具有子空间保持性质的时候，其子空间保持误差是 0。

现在，子空间聚类的性能不仅取决于子空间保持性，还取决于相似性图的连通性，即每个聚类中的数据点是否构成图的连通部分。一个比较重要的指标是连通性 (c)：对于权重 $W \in \mathbb{R}^{N\times N}$ 且度矩阵 $D = diag(W \cdot 1)$，论文使用归一化拉普拉斯矩阵 L 的第二最小特征值 λ_2 来衡量图的连通性；λ_2 的范围为 $[0, \frac{n-1}{n}]$，并且只有当图不连通时才为零。在论文的例子中，计算每个簇的代数连通性 λ_2^i，并将数量 $c = min_i\lambda_2^i$ 作为连通性的度量。

最后，用以下两个指标来评价子空间聚类方法的性能。一是聚类准确率 ($a\%$)：这是正确标注数据点的百分比。它是通过匹配估计标签和真实标签计算得出的，即 $a = \max_\pi \frac{100}{N}\sum_{ij} Q^{\text{est}}_{\pi(i)j} Q^{\text{true}}_{ij}$，其中 是 n 个群组的置换，Q^{est} 和 Q^{true} 分别是数据的估计标签和地面真实标签，如果点 j 属于簇 i，则它们的 $i\,j$ 项等于 1，否则为 0。二是运行时间 (t)：使用 python 完成每个聚类任务的运行时间。

4 复现细节

4.1 与已有开源代码对比

本论文已开源部分源代码，源代码来自 https://github.com/ChongYou/subspace-clustering。该项目主要由 python 语言编写，核心算法文件 selfrepresentation.py，其中包括了自表示矩阵的数据结构 SelfRepresentation，弹性网络支撑集 (active-support-elastic-net)，SSC-BP 算法、SSC-OMP 算法、LSR 算法的代码实现，第二部分为数据集，包含了随机生成的合成数据集和 MNIST 手写数字图像数据集，第三部分为多种评价指标的实现，代码文件为 accuracy.py，第四部分为对两个数据集的测试与分析。算法执行的主要流程如下：(1) 初始化 SparseSubspaceClusteringOMP 类，参数包括指定的簇数量 n_clusters、构建亲和矩阵的方式 affinity、与谱聚类相关参数 random_state 和 n_init、残差阈值 thr、最大迭代次数 n_nonzero。(2) 对预处理好的数据进行训练，训练时会计算出子空间自表示矩阵 C^*，并对 C^* 进行谱聚类，返回一个 csr_matrix 类型的稀疏矩阵。

在进行 MINST 数据集的测试中，除了论文中的聚类准确率指标，我还采用了 NMI(Normalized Mutual Information) 指标和 ARI 指标来从不同方面反映聚类结果的优劣，提高评估准确性。

NMI 是一种用于衡量两个聚类结果之间相似性或一致性的指标，衡量两个随机变量之间的相互依赖程度，它表示由于知道其中一个变量的值而导致另一个变量的不确定性减少的程度。$NMI(C,K) = \frac{MI(C,K)}{\sqrt{H(C)H(K)}}$，其中 $H(C)$ 和 $H(K)$ 分别是聚类 C 和 K 聚类的信息熵，信息熵是衡量随机变量不确定性的指标。NMI 值越接近 1，表示两个聚类结果越相似，一致性越高；NMI 值越接近 0，表示两个聚类结果越不相似，一致性越低。当 $NMI = 1$ 时，说明两个聚类结果完全一致；当 $NMI = 0$ 时，说明两个聚类结果是完全独立的，没有任何相似性。在聚类分析中，NMI 用于评估不同聚类算法对同一数据集的聚类结果的优劣，帮助选择最合适的聚类算法。

ARI 是调整兰德指数（Adjusted Rand Index），是一种用于评估聚类结果与真实标签之间相似度的指标。它通过混淆矩阵和组合数得出，其中 ARI 的取值范围为-1 到 1。其中 $ARI = 1$，表示聚类结果与真实标签完全一致，是最理想的情况；$ARI = -1$ 表示聚类结果与真实标签完全相反，是最差的情况；ARI 接近 0 时，表示聚类结果与随机聚类的结果差不多，聚类效果比较差。ARI 值越大，表示聚类结果与真实类别越相似，聚类效果越好。在比较不同聚类算法对同一数据集的聚类效果时，通过计算 ARI，可以直观地判断哪种算法的聚类结果更接近真实情况，从而选择更优的聚类算法。

4.2 实验环境搭建

搭建稳定且适配的实验环境是算法复现的关键环节。本研究的实验环境主要基于 Python 编程语言，并借助了一些常用的科学计算与数据分析库。硬件方面，采用了一台配备 Intel Core i5 - 14600K 处理器、32GB 内存以及 NVIDIA GeForce RTX 4070 GPU 的台式计算机。该硬件配置能够满足算法在处理中等规模数据集时的计算需求，确保实验过程的流畅性。软件环境搭建如下：首先，安装了 Anaconda 集成开发环境，用于管理 Python 版本及相关依赖库。创建了一个独立的 conda 虚拟环境，避免不同项目之间的库冲突。在虚拟环境中，安装了 Python 3.8 版本，确保能够兼容大多数最新的库函数。对于核心的科学计算库，安装了 NumPy 用于高效的数值计算，尤其是矩阵运算；SciPy 提供了丰富的科学算法工具，如优化算法、线性代数求解等，在正交匹配追踪算法的实现中发挥了重要作用；Scikit - learn 库则用于实现聚类算法以及数据预处理中的一些常规操作，如标准化、K - means 聚类等。此外，为了可视化实验结果，安装了 Matplotlib 库，用于绘制数据分布、聚类结果等图形，直观展示算法的性能表现。SPAMS 库是一个用于稀疏优化的 Python 库，在机器学习和数据分析中，用于特征选择、模型正则化等。当数据特征较多且存在冗余时，利用 SPAMS 库求解稀疏模型，可自动选择出对目标变量最具影响力的特征，提高模型的泛化能力和解释性。在处理大规模数据集时，利用了 PyTorch 库的 GPU 加速功能，将手写数字数据集中部分矩阵运算和迭代计算过程转移到 GPU 上执行，显著提高了算法的运行速度。通过精心搭建上述实验环境，为基于正交匹配追踪的稀疏子空间聚类算法的复现与实验提供了坚实的基础，确保了实验结果的准确性与可靠性。

4.3 界面分析与使用说明

本复现算法的实现提供了简洁易用的命令行界面，方便研究人员快速上手并进行实验。在打开项目后，用户首先需要通过要确定指定数据集的路径。用户还需根据数据集的特点，设

置一些关键的算法参数，如正交匹配追踪算法中的最大迭代次数、残差阈值，以及谱聚类算法中的预估子空间个数（聚类数）等。这些参数的合理设置对于算法的性能至关重要，不同数据集可能需要不同的参数组合来达到最佳聚类效果。

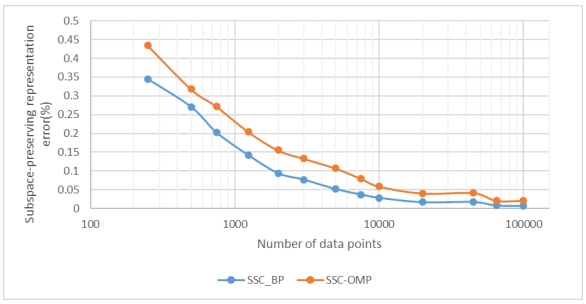
复现的两个工作可以在 run_mnist.py 和 run_synthetic.py 中得到。程序运行过程中，会在命令行实时输出一些关键信息，包括数据加载进度、正交匹配追踪算法的迭代过程信息，以及谱聚类算法中的特征分解进度等，这些功能通过一个叫 progress2 的库实现。这些信息有助于用户了解算法的执行流程，及时发现可能出现的问题，如数据加载错误、算法不收敛等。当算法完成聚类后，会将聚类结果以文本文件的形式输出到指定目录。同时，利用 Matplotlib 库绘制聚类结果的可视化图形，展示数据点在低维空间中的分布以及所属的聚类类别，用户可以直观地评估聚类的准确性。对于初次使用的用户，提供了详细的 README 文件，涵盖了从环境安装、参数设置到结果解读的全过程说明。还针对常见问题提供了故障排除指南，确保用户在使用过程中遇到问题能够及时解决，顺利完成实验任务。

5 实验结果分析

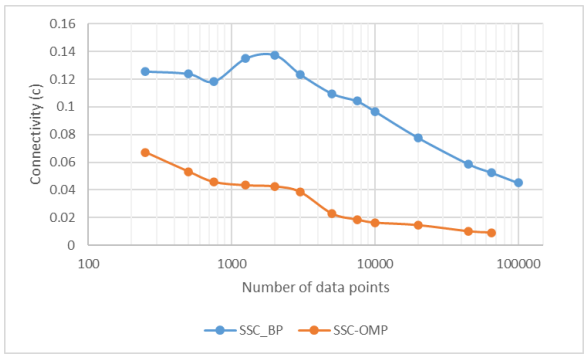
5.1 合成数据测试

论文通过随机生成的合成数据集验证了 SSC-OMP 的理论结果 (即使子空间不是独立的，OMP 的解决方案也是子空间保持的，其误差随着数据点密度的增加而降低)，并通过使用随机模型对合成数据进行实验将其与 SSC-BP 进行比较。测试在维数为 $D = 9$ 的环境空间中随机生成 $n = 5$ 个子空间，每个子空间的维数为 $d = 6$。每个子空间包含在单位球面上随机生成的 $N_i = \rho d$ 个样本点，其中 ρ 的范围从 250 到 100000 ，因此点的数量从 150 到 150,000 。对于 SSC-OMP，在算法 1中设定 ϵ 为 10^{-3}，k_{max} 为 6。对于 SSC-BP，测试使用 ℓ_1-Magic 求解器。

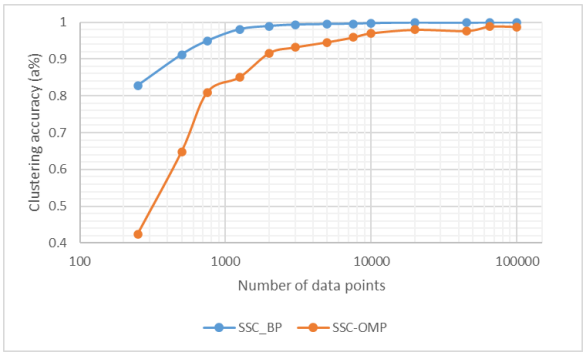
图 1a中绘制了子空间保持表示误差。可以看出，随着数据点密度的增加，SSC-OMP 得到子空间保留解的误差也在减小。与 SSC-OMP 相比，我们可以看到 SSC-BP 的误差更优。这与论文的理论分析部分相吻合，即 SSC-OMP 得到子空间保留表示的条件更强 (即更难满足)。从子空间聚类的角度来看，论文更感重视的是该方法在聚类精度方面的表现，以及该方法在运行时间方面的效率。图 1b和 1c以及图 1d中的连线绘制了这些结果。首先发现，SSC-OMP 的连接性不如 SSC-BP。部分原因可能是，正如子空间保留百分比所显示的那样，它的正确连接较少。在聚类精度方面，SSC-OMP 在小规模数据时比 SSC-BP 差。这是因为 SSC-OMP 生成的稀疏表示不像 SSC-BP 那样具有子空间保留性或良好的连接性。不过，观察到随着数据点密度的增加，聚类精度的差异也在减小，SSC-OMP 似乎可以在大 N 的情况下实现任意好的聚类精度。此外，从图 1d中可以明显看出，SSC-OMP 的速度明显更快：在对 7500 个点进行聚类时，它比 SSC-BP 快 3 到 4 个数量级。复现工作的结论是，随着 N 的增加，SSCOMP 和 SSC-BP 在聚类精度上的差距会缩小，但 SSC-OMP 的速度明显更快，因此它更适合处理大规模问题，以上复现得出的结论与论文中的结论一致，数值因机器硬件的不同而有所差异。



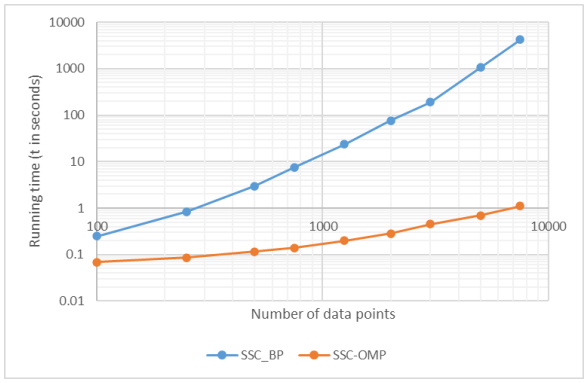
(a) 子空间保持表示误差



(b) 连通性



(c) 聚类准确度



(d) 计算时间

图 1. SSC-OMP 和 SSC-BP 在合成数据上的性能。数据来自环境维度为 9、维度为 6 的 5 个子空间。每个子空间包含相同数量的点，总点数从 250 到 100000 ，并以对数比例显示。请注意，图 1d的 y 轴也使用了对数标度。

5.2 手写数字图像聚类

在本次复现实验中评估了不同子空间聚类方法对手写数字图像进行聚类的性能。论文中使用的是 MNIST 数据集，其中包含手写数字 0-9 的灰度图像。在每次实验中为 10 位数字中的每一位随机选择 $N_i \in \{50, 100, 200, 400, 600\}$ 幅图像。对于每幅图像，使用散射卷积网络计算一组特征向量。特征向量是网络各层系数的串联，具有平移不变性和变形稳定性。每个特征向量的大小为 3472。然后使用 PCA 算法将所有图像的特征向量投影到 500 维，然后将子空间聚类技术应用于投影特征。结果见表 1。

数据显示，SSC-OMP 和 SSC-BP 的子空间表示误差都比 LSR 小得多，其中 SSC-BP 比 SSC-OMP 更优。这与论文中的理论分析是一致的，因为 LSR 无法保证对非独立子空间给出子空间保留表示，而 SSC-BP 比 SSC-OMP 更有可能给出子空间保留表示。就聚类准确度而言，SSC-OMP 优于 SSC-BP，而 SSC-BP 又优于 LSR；以 nmi 和 ari 的角度看，SSC-BP 略优于 SSC-OMP，而 LSR 的指标相较差，这与论文中得出的 SSC-BP 的子空间保持性误差较小有关。考虑到各种方法的运行时间，复现出的计算时间因为机器硬件的原因与论文中的时间有所出入，但是结论相同，SSC-BP 需要更多的计算量，尤其是当点的数量较多时。这再次证明论文所提出 SSC-OMP 方法适用于大规模问题。

表 1. 子空间聚类方法在 MNIST 数据集上的性能。数据由随机选择的数字 $N_i \in \{50\ 100\ 200\ 400\ 600\}$ 组成，每个数字（即 0-9）的图像都是从散射网络中提取的特征，并使用 PCA 投影到维数 500。

No.points	**500**	**1000**	**2000**	**4000**	**6000**
$e\%$: subspace-preserving representation error					
SSC-OMP	31.91	28.34	27.05	25.21	24.45
SSC-BP	25.13	24.74	22.73	19.04	17.72
LSR	83.39	83.37	83.55	83.49	83.65
$a\%$: average clustering accuracy					
SSC-OMP	87.78	87.81	89.31	91.63	90.32
SSC-BP	82.38	83.47	83.92	84.27	84.52
LSR	77.38	77.68	77.87	78.48	78.86
$nmi\%$: normalized mutual information					
SSC-OMP	76.91	77.92	78.73	81.74	82.83
SSC-BP	87.76	87.86	86.79	88.15	89.04
LSR	76.38	77.24	77.20	77.84	78.39
$ari\%$: adjusted rand index					
SSC-OMP	75.59	75.53	78.17	80.64	81.75
SSC-BP	78.55	79.57	81.62	80.82	81.79
LSR	67.21	68.42	68.92	70.30	71.67
t(sec.): running time					
SSC-OMP	5.14	5.76	8.02	14.16	23.17
SSC-BP	26.46	29.52	37.25	57.09	91.28
LSR	2.10	2.47	3.32	4.80	6.88

6 总结与展望

本研究成功复现了基于正交匹配追踪的稀疏子空间聚类算法，并在多个方面取得了显著成果。通过深入剖析算法原理，细致阐述复现步骤，搭建稳定实验环境，实现了算法从理论到实践的落地。在与已有开源代码的对比学习中，汲取优化思路，提升了算法性能，同时添加了 nmi，ari 等多种聚类指标，有效提高评估准确性和稳定性，使其在随机合成数据、MNIST 手写数字数据集上均展现出较高的聚类准确率，验证了算法在不同领域数据处理中的有效性和适应性，为高维数据的聚类分析提供了有力工具，具有重要的理论与实践意义。

尽管取得了一定成果，但研究过程中也暴露出一些问题。复现过程中发现，SSC-OMP 算法在处理超高维、大规模且复杂结构的数据时，聚类性能波动较大，内存占用较大。虽然引入了一些优化策略，但在极端复杂的数据分布下，如多子空间高度相交、噪声分布不均匀等情

况，算法的聚类准确性还有待进一步提高。对于参数的选择，目前仍依赖于经验和反复试验，缺乏自适应的参数优化机制，降低了算法的易用性。未来的研究将聚焦于算法的改进与拓展。一方面，计划探索更高效的稀疏求解算法和近似计算方法，以降低计算复杂度，结合分布式计算框架，提升算法对大规模数据的处理能力。另一方面，深入研究数据的内在结构特性，挖掘更多有效的特征信息，进一步优化相似度矩阵构建和谱聚类过程，提高算法在复杂数据分布下的鲁棒性。尝试引入深度学习技术，自动学习数据的最优表示，实现参数的自适应调整，增强算法的自适应性。在应用拓展方面，将积极探索 SSC - OMP 算法在视频分析、生物医学大数据、物联网数据挖掘等新兴领域的应用，充分发挥其优势，为解决实际问题提供更精准、高效的解决方案。

参考文献

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

[2] Ehsan Elhamifar and René Vidal. Clustering disjoint subspaces via sparse representation. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1926–1929. IEEE, 2010.

[3] Ehsan Elhamifar and René Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781, 2013.

[4] Paolo Favaro, René Vidal, and Avinash Ravichandran. A closed form solution to robust subspace estimation and clustering. In *CVPR 2011*, pages 1801–1807. IEEE, 2011.

[5] Chun-Guang Li, Chong You, and René Vidal. Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework. *IEEE Transactions on Image Processing*, 26(6):2988–3001, 2017.

[6] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):171–184, 2012.

[7] Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 663–670, 2010.

[8] Canyi Lu, Jiashi Feng, Zhouchen Lin, and Shuicheng Yan. Correlation adaptive subspace segmentation by trace lasso. In *Proceedings of the IEEE international conference on computer vision*, pages 1345–1352, 2013.

[9] Mahdi Soltanolkotabi and Emmanuel J Candes. A geometric analysis of subspace clustering with outliers. 2012.

[10] Mahdi Soltanolkotabi, Ehsan Elhamifar, and Emmanuel J Candes. Robust subspace clustering. 2014.

[11] Ehsan Elhamifar René Vidal et al. Sparse subspace clustering. In *2009 IEEE conference on computer vision and pattern recognition (CVPR)*, volume 6, pages 2790–2797, 2009.

[12] Yu-Xiang Wang and Huan Xu. Noisy sparse subspace clustering. *Journal of Machine Learning Research*, 17(12):1–41, 2016.

[13] Yu-Xiang Wang, Huan Xu, and Chenlei Leng. Provable subspace clustering: When lrr meets ssc. *Advances in Neural Information Processing Systems*, 26, 2013.

[14] Chong You and René Vidal. Geometric conditions for subspace-sparse recovery. In *International conference on machine learning*, pages 1585–1593. PMLR, 2015.