Subspace Memory Clustering

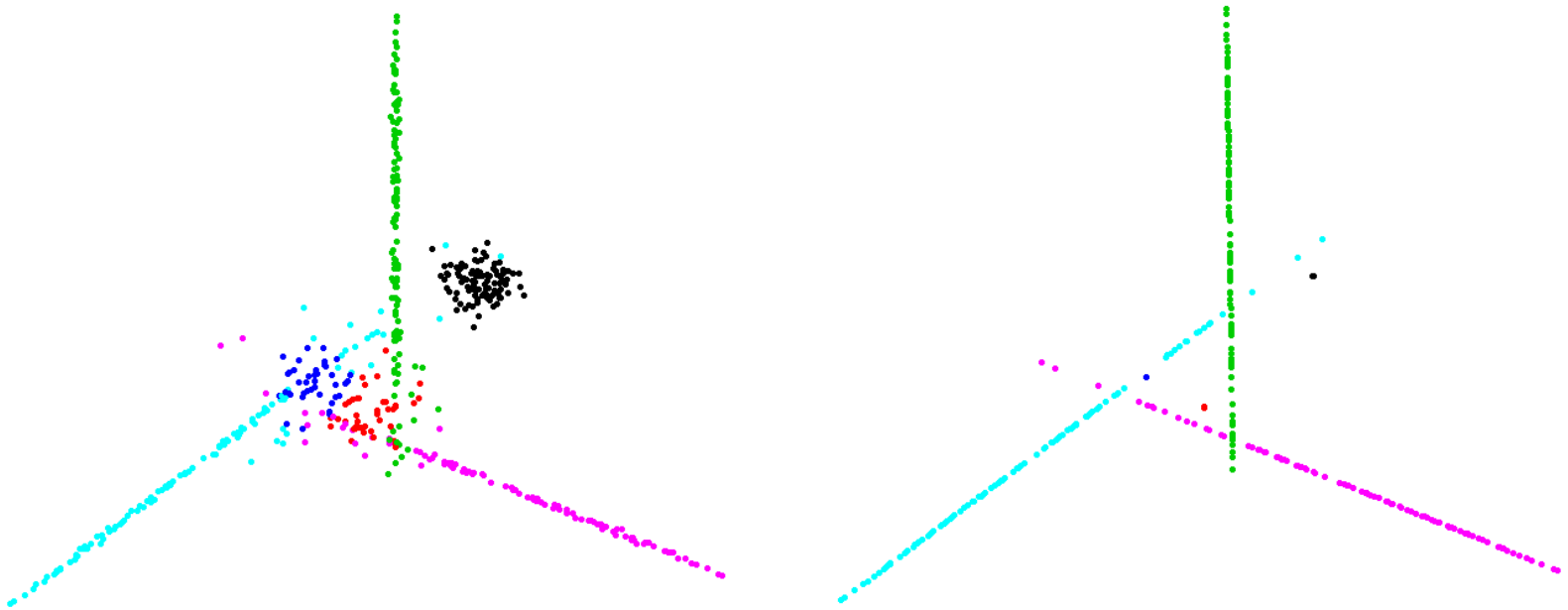
Łukasz Struski, Jacek Tabor and Przemysław Spurek

Faculty of Mathematics and Computer Science,
Jagiellonian University (Poland)

TFML 2015, Będlewo, Poland

1. What is subspace clustering?
2. Our method
3. Experiments
4. Conclusions

Subspace clustering can be viewed as generalization of PCA.



(a) All points of a given dataset together with its clusters.

(b) Projections onto the subspaces.

Figure: Dividing a dataset into two zero-dimensional (points) and three one-dimensional subspaces (lines).

Why do we use subspace clustering?

We use subspace clustering in order to:

1. reduce the high number of data to several basic categories that can be used as objects for further analysis,
2. obtain a homogeneous test subjects,
3. reduce the workload and time analysis,
4. ...

Why do we use subspace clustering?

We use subspace clustering in order to:

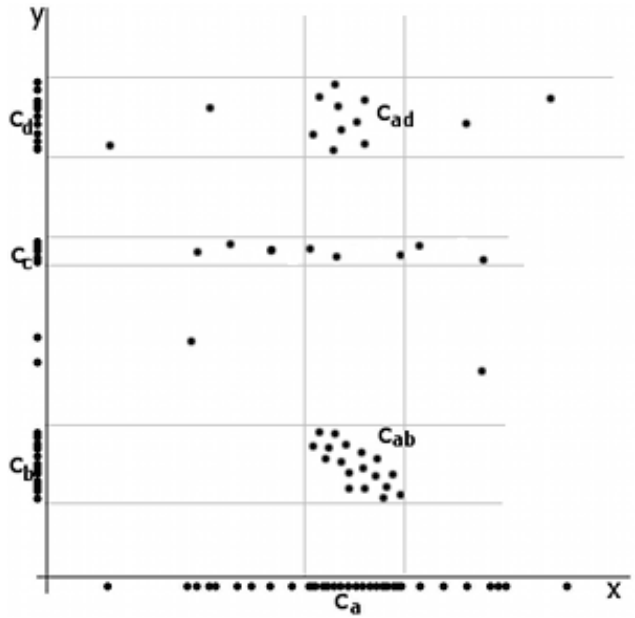
1. reduce the high number of data to several basic categories that can be used as objects for further analysis,
2. obtain a homogeneous test subjects,
3. reduce the workload and time analysis,
4. ...

Applications:

1. image processing (e.g., image segmentation, representation and compression),
2. systems theory (e.g., hybrid system identification).

In general, subspace clustering algorithms can be divided into two main classes:

1. constructing axis parallel subspaces – this approach is used in methods: ENCLUS, MAFIA, CLTree, DOC, PROCLUS;



2. those who are able to build arbitrary subspaces – is used: ORCLUS, DBSCAN, PCA.

Our goal is to divide dataset $D \subset \mathbb{R}^N$, $N \in \mathbb{N}$ into $k \in \mathbb{N}$ pairwise disjoint clusters $D_1, \ldots, D_k$ $(D = D_1 \cup \ldots \cup D_k)$ such that each cluster is well represented by an affine subspace.

How can we do this?

For a fixed allowed memory we minimize the projection error.

Our goal is to divide dataset $D \subset \mathbb{R}^N$, $N \in \mathbb{N}$ into $k \in \mathbb{N}$ pairwise disjoint clusters $D_1, \ldots, D_k$ $(D = D_1 \cup \ldots \cup D_k)$ such that each cluster is well represented by an affine subspace.

How can we do this?

For a fixed allowed memory we minimize the projection error.

We need to answer on two questions:

1. How to calculate error?
2. What is memory?

Error:

Suppose that we have one cluster (group) $D \subset \mathbb{R}^N$ and an affine space $V \subset \mathbb{R}^N$ of dimension $n \leq N$. The total squared-error in cluster is given by

$$E[D; V] := \sum_{x \in D} \mathrm{dist}(x, V)^2,$$

where $\mathrm{dist}(x, V)$ describe distance between point and its orthogonal projection on subspace V.

We use the PCA method to construct the subspace V which minimizes the total squared-error with fixed level of allowed memory. Hence the optimal subspace is given by

$$V = \mathrm{mean}_D + \mathrm{span}(v_1, \ldots, v_n),$$

where mean_D is a mean of D and $v_1, \ldots, v_n$ are the n consecutive eigenvectors of the covariance matrix cov_D of D (ordered decreasingly with respect to eigenvalues).

This allows us to define the minimal projection error by

$$E[D, n] := E[D, \mathrm{mean}_D + \mathrm{span}(v_1, \ldots, v_n)]. \quad (1)$$

This allows us to define the minimal projection error by

$$E[D, n] := E[D, \mathrm{mean}_D + \mathrm{span}(v_1, \ldots, v_n)]. \quad (1)$$

Proposition 1 (*[I. Jolliffe. Principal component analysis. Wiley Online Library, 2005]*)

Let $D \subset \mathbb{R}^N$ be a given dataset, where $N \in \mathbb{N}$, and let $\lambda_1, \ldots, \lambda_N$ be the decreasingly ordered eigenvalues of covariance matrix of D. Then the squared error is given by

$$E[D, n] = |D| \cdot \sum_{i \in \mathbb{N},\ i > n} \lambda_i.$$

Memory:

Memory is understood as the number of scalars, which we need to describe a point after projection. We define the compression level of memory (measured by the number of used scalars) by

$$p := \frac{|D|n}{|D|N} = \frac{n}{N} \in [0, 1],$$

where $|D|$ is the cardinality of the dataset D.

Optimization problem. *Let dataset $D \subset \mathbb{R}^N$, the number of cluster k and the compression level $p \in [0,1]$ be given. Our goal is to find the splitting of D into k-clusters $D_1, \ldots, D_k$ and numbers $n_1, \ldots, n_k$, which minimize the total squared error*

$$E[D_1, n_1; \ldots; D_k, n_k] := E[D_1, n_1] + \ldots + E[D_k, n_k],$$

under the condition

$$\frac{n_1|D_1| + \ldots + n_k|D_k|}{N|D|} \leq p.$$

Algorithm

Optimization procedure:

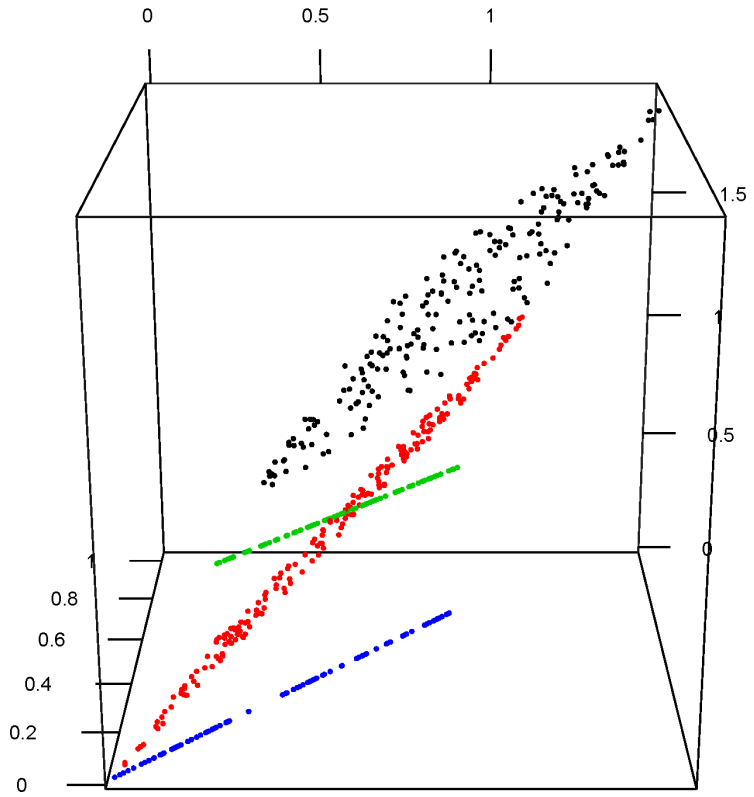
1. assign each point of D randomly to one of k clusters $D_1, \ldots, D_k$ and calculate the amount of memory $M_i = p \cdot |D_i| \cdot N$ for $i = 1, \ldots, k$,
2. proceed through all elements of the dataset D.
   a) switch position of point from the cluster where belongs $x \in D_i$ to another clusters D_j $(j \neq i)$, calculate square error of $E[\tilde{D}_i, \tilde{n}_i] + E[\tilde{D}_j, \tilde{n}_j]$ for all $j \neq i$ and we choose minimum from them (assume that it is realized by some $t \in \{1, \ldots, N\} \setminus \{i\}$),
   b) compare this minimum from $E[D_i, n_i] + E[D_t, n_t]$ and if we obtain smaller error by changing the belonging of point than we switch in 'original' cluster.
3. We repeat Step 2 until we do not switch the position of any point from our dataset.

We create two datasets $X_1, X_2 \subset \mathbb{R}^3$, which contain two 1-dimensional and two 2-dimensional clusters.

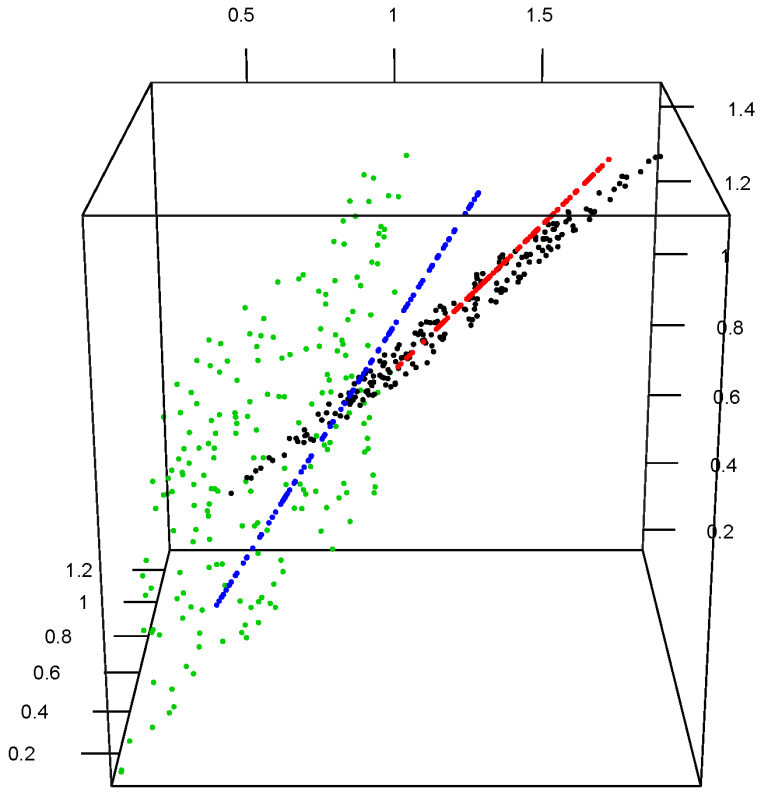
Dataset		X_1		X_2	
Dim. of clusters		1	2	1	2
Compression level for SuMC		0.33	0.66	0.33	0.66
Rand index	SuMC	0.8728381	1	0.7586923	1
	ORCLUS	1	0.7374569	0.9950417	0.8364218

Table: Comparison of SuMC and ORCLUS algorithms on synthetic datasets $X_1, X_2 \subset \mathbb{R}^3$, which have two 1-dimensional and two 2-dimensional subspaces (we divide on four clusters).

Our method allows to detect the optimal dimensions of the clusters.



(a) Clustering of dataset X_1 for $p = 0.63$.



(b) Clustering of dataset X_2 for $p = 0.56$.

In the next example we compare methods on datasets form the uci-repository `http://archive.ics.uci.edu/ml`. We consider three datasets from this repository: *glass*, *wine* and *yeast*.

Dim.	Rand index								
	Glass			Wine			Yeast		
	p	SuMC	ORCLUS	p	SuMC	ORCLUS	p	SuMC	ORCLUS
1	0.11	0.6715809	0.4869027	0.07	0.5330413	0.5588142	0.12	0.7325302	0.6405561
2	0.22	0.6973367	0.590935	0.15	0.629277	0.5567828	0.25	0.7277601	0.4985987
3	0.33	0.6814971	0.5919881	0.23	0.643433	0.5339935	0.37	0.7062958	0.6327916
4	0.44	0.6872888	0.6412619	0.31	0.6604456	0.5365962	0.5	0.6856921	0.6980632
5	0.55	0.6803563	0.6620157	0.38	0.5821748	0.5584333	0.62	0.6272326	0.7041102
6	0.66	0.6391558	0.6520995	0.46	0.5590681	0.5677649	0.75	0.6363158	0.6994391
7	0.77	0.6410425	0.669036	0.54	0.556275	0.5484035	0.87	0.7035431	0.7031342
8	0.88	0.5298144	0.6895704	0.61	0.5622421	0.6423538	1	0.7215641	0.7008795
9	1	0.6546444	0.6442017	0.69	0.5571002	0.5890307	—	—	—
10	—	—	—	0.77	0.5509427	0.6494001	—	—	—
11	—	—	—	0.85	0.568971	0.6478131	—	—	—
12	—	—	—	0.92	0.5589412	0.6065511	—	—	—
13	—	—	—	1	0.5534819	0.6621596	—	—	—

The main advantage of our methods:

1. automatically finds the optimal dimensions of the clusters,
2. we do not need to explicitly specify dimensions of groups.