Learning Self-Expression Metrics for Scalable and Inductive Subspace Clustering

Julian Busch* **Evgeniy Faerman** **Matthias Schubert** **Thomas Seidl**

Ludwig-Maximilians-Universität München
Munich Center for Machine Learning (MCML)
`{busch, faerman, schubert, seidl}@dbs.ifi.lmu.de`

Subspace clustering has established itself as a state-of-the-art approach to clustering high-dimensional data. In particular, methods relying on the self-expressiveness property have recently proved especially successful. However, they suffer from two major shortcomings: First, a quadratic-size coefficient matrix is learned directly, preventing these methods from scaling beyond small datasets. Secondly, the trained models are transductive and thus cannot be used to cluster out-of-sample data unseen during training. Instead of learning self-expression coefficients directly, we propose a novel metric learning approach to learn instead a subspace affinity function using a siamese neural network architecture. Consequently, our model benefits from a constant number of parameters and a constant-size memory footprint, allowing it to scale to considerably larger datasets. In addition, we can formally show that out model is still able to exactly recover subspace clusters given an independence assumption. The siamese architecture in combination with a novel geometric classifier further makes our model inductive, allowing it to cluster out-of-sample data. Additionally, non-linear clusters can be detected by simply adding an auto-encoder module to the architecture. The whole model can then be trained end-to-end in a self-supervised manner. This work in progress reports promising preliminary results on the MNIST dataset. In the spirit of reproducible research, me make all code publicly available. [1] In future work we plan to investigate several extensions of our model and to expand experimental evaluation.

1 Introduction

Subspace clustering [Vidal, 2011] assumes the data to be sampled from a union of low-dimensional subspaces of the full data space. The goal is to recover these subspaces and to correctly assign each data point to its respective subspace cluster. As a state-of-the-art approach to clustering high-dimensional data, it enables a multitude of applications, including image segmentation [Ma et al., 2007, Yang et al., 2008], motion segmentation [Kanatani, 2001, Elhamifar and Vidal, 2009, Ji et al., 2016], image clustering [Ho et al., 2003, Elhamifar and Vidal, 2013] and clustering gene expression profiles [McWilliams and Montana, 2014]. For instance, face images of a subject under fixed pose and varying lighting conditions [Basri and Jacobs, 2003] or images of hand-written digits with different rotations, translations and other natural transformations [Hastie and Simard, 1998] have been shown to lie in low-dimensional subspaces.

Recently, *self-expressiveness*-based methods [Elhamifar and Vidal, 2009, Liu et al., 2010, Lu et al., 2012, Elhamifar and Vidal, 2013, Liu et al., 2013, Wang et al., 2013, Feng et al., 2014, Ji et al., 2014, Vidal and Favaro, 2014, Ji et al., 2015, You et al., 2016a] have proved especially successful. The main idea is that each point can be expressed by a linear combination of points from the same subspace. This property is used to learn a quadratic-size coefficient matrix from which cluster labels can be extracted in a post-processing step using spectral clustering. The quadratic number of parameters prevents these methods from scaling beyond small datasets and makes them transductive and thus

[1] `https://github.com/buschju/sscn`

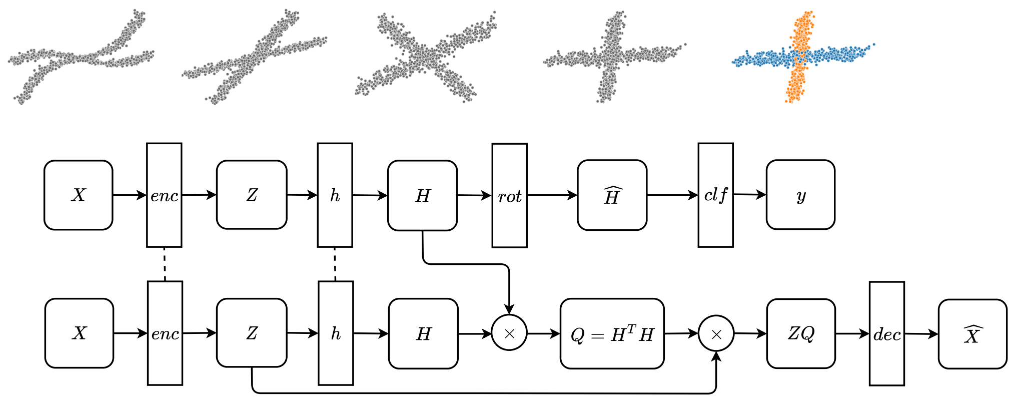

Figure 1: Data flow within our model. Square boxes denote tensors, rectangular boxes denote functions, and dashed lines indicate parameter sharing. After being mapped into a space in which the data fits better into independent linear subspaces using an encoder function enc, the data is mapped into a space using a function h in which independent clusters are orthogonal and dot-products corresponds to self-expression coefficients. Afterwards, the data is rotated into pre-defined axis-aligned subspaces and assigned to clusters based on the orthogonal projection distance (upper path). The original data is reconstructed from the self-expressed data in latent space by a decoder function dec to ensure that the learned embeddings are compatible with the original data (lower path).

inapplicable to out-of-sample data unseen during training. In contrast, our model requires only a constant number of parameters to provably provide the same expressive power. A classifier leveraging the unique geometric properties of our model further makes it inductive and enables it to cluster out-of-sample data.

In practice, we can usually not assume that the data exactly fits in linear subspaces, e.g., due to noise or additional non-linearities in the underlying data generating process which was not accounted for. Instead, we might rather assume the data to be situated on different *non-linear* sub-manifolds. Some methods [Chen and Lerman, 2009, Patel et al., 2013, Patel and Vidal, 2014, Xiao et al., 2016, Yin et al., 2016, Ji et al., 2017a] rely on the *kernel trick* to account for non-linearity. However, it is usually not clear whether a particular pre-defined kernel function is particularly suitable for subspace clustering. More recent methods [Peng et al., 2016, Ji et al., 2017b, Zhou et al., 2018, Zhang et al., 2019a, Seo et al., 2019, Kheirandishfard et al., 2020] learn a suitable feature transformation explicitly in an *end-to-end* differentiable model. In particular, *Deep Subspace Clustering Networks (DSC-Net)* [Ji et al., 2017b] introduced the idea of modeling the coefficient matrix as a dense neural network layer, called self-expressive layer, and training it jointly with an auto-encoder. As a result, the encoder represents a feature transformation which has been optimized w.r.t. linear cluster structure in latent space. *Self-Supervised Convolutional Subspace Clustering Networks (S^{2}ConvSCN)* [Zhang et al., 2019a] learn an additional classifier which can be applied to out-of-sample data but still rely on the full coefficient matrix and spectral clustering. Our model on the other hand offers both, applicability to out-of-sample data and scalability.

Several works have addressed the challenge of *scalability* but are either only able to detect linear clusters [You et al., 2016b, Rahmani and Atia, 2017], rely on a k-means-like procedure which requires good initialization and is sensitive to outliers [Zhang et al., 2018] or still fully parametrize coefficient matrices which need to be re-learned from scratch for each new data batch and come with no theoretical guarantees [Zhang et al., 2019b]. In contrast, our model is suitable to detect non-linear clusters, can be trained end-to-end with back-propagation and provides a theoretical foundation.

In summary, we propose, to the best of our knowledge, the first metric learning approach to subspace clustering, which enables a quadratic reduction of the number of parameters and memory footprint compared to existing methods while maintaining theoretical performance guarantees. Our model is applicable to out-of-sample data, suitable to detect non-linear clusters and can be trained end-to-end with back-propagation.

2 Siamese Subspace Clustering Networks

In subspace clustering, we are given a set set of points $\{x_i\}_{i=1}^N \subseteq \mathbb{R}^{d_X}$ sampled from a union of subspaces $\{S_i\}_{i=1}^K$ of unknown dimensions and arranged as columns of a data matrix $X \in \mathbb{R}^{d_X \times N}$. The goal is to recover these subspaces and to correctly assign each data point to its respective subspace cluster. While there exist many different variants of self-expressive subspace clustering, we focus here on a relaxed noise-aware version of *Efficient Dense Subspace Clustering (EDSC)* [Ji et al., 2014]:

Definition 1 (Efficient Dense Subspace Clustering [Ji et al., 2014])**.**

$$\min_{C \in \mathbb{R}^{N \times N}} \frac{1}{2} \|C\|_F^2 + \frac{\lambda}{2} \|X - XC\|_F^2 . \tag{1}$$

where the i-th column of the $N \times N$ coefficient matrix C contains the coefficients for expressing x_i. Regularization of C ensures that x_i is expressed using only points from the same subspace. Given the learned coefficient matrix, cluster assignments can be extracted in a post-processing step by applying spectral clustering to the subspace affinity matrix $A = |C| + |C|^T$. The unique solution to this problem can be expressed in closed-form as the solution C^* of the linear system $\left(I + \lambda X^T X\right) C = \lambda X^T X$ [Ji et al., 2014]. Let $r := \operatorname{rank}(X) = \dim\left(\bigoplus_{i=1}^K S_i\right)$ denote the rank of X. In a noise-free setting, if the subspaces are *independent*, i.e., if $r = \sum_{i=1}^K \dim(S_i)$, then C^* is guaranteed to be block-diagonal with $C_{ij}^* = 0$ if x_i and x_j originate from different subspaces [Vidal et al., 2008]. The corresponding solution is called *subspace-preserving*.

The central idea of our approach is to view subspace clustering from a metric learning perspective. To this end, we employ a siamese neural network [Bromley et al., 1994] consisting of two identical branches with shared weights and mirrored parameter updates which is optimized such that dot-products in latent space correspond to self-expression coefficients:

Definition 2 (Siamese Dense Subspace Clustering)**.**

$$\begin{aligned} \min_{\theta_h} \quad & \frac{1}{2} \|Q\|_F^2 + \frac{\lambda}{2} \|X - XQ\|_F^2 \\ \text{s.t.} \quad & Q = H^T H, \quad H = h\left(X; \theta_h\right) \end{aligned} \tag{2}$$

where $Q \in \mathbb{R}^{N \times N}$ contains the self-expression coefficients corresponding to dot-products of the embeddings $H \in \mathbb{R}^{d_H \times N}$ computed by the embedding function h. Note that weight sharing leads to symmetric coefficient matrices. Even though the reduction of parameters compared to (1) is quadratic, we can show that this model is able to recover the exact solution to the original subspace clustering problem, even when h consists of only a single linear layer with a sufficient number of neurons. Note that (2) is convex in this case.

Theorem 1. *Let $h(X) = WX$, $W \in \mathbb{R}^{d_H \times d_X}$, $d_H \geq r$, then (2) attains its global minimum at $W^* = R\sqrt{\lambda\left(I - \lambda\left(\Sigma_r^{-2} + \lambda I\right)^{-1}\right)} U_r^T$ where $X = U_r \Sigma_r V_r^T$ is the reduced SVD of X and $R \in \operatorname{St}(d_H, r)$ is an arbitrary orthonormal matrix. The unique optimal coefficient matrix Q^* of (2) corresponds to the unique solution of (1).*

Above, $\operatorname{St}(n,p) = \left\{X \in \mathbb{R}^{n \times p} \mid X^T X = I\right\}$ for $n \geq p$ denotes the *Stiefel manifold* which is composed of all $n \times p$ orthonormal matrices. Since (2) leads to a well-studied optimal solution, it can be analyzed directly within existing theory. In particular, it is guaranteed that under the independence assumption and in a noise-free setting, (2) yields a subspace-preserving solution. Also note that we don't need to know the exact rank of X, it is sufficient to have an upper bound. Since $r \leq \sum_{i=1}^K \dim(S_i)$, we can simply estimate the number of clusters K and the maximum cluster dimension q and set $d_H = Kq$ and $R \in \operatorname{St}(d_H, d_H)$.

Since we can choose R arbitrarily from $\operatorname{St}(d_H, d_H)$ and still obtain the same optimal coefficient matrix Q^*, we are able to optimize R on the Stiefel manifold w.r.t. to a cluster assignment objective where we take advantage of the observation that points from independent clusters will have orthogonal embeddings in H. To this end, we compute rotated embeddings $\widehat{H} = RH$ and then classify points by assigning them to their closest subspace w.r.t. orthogonal projection distance and applying the

	ACC	ARI	NMI	#Parameters	GPU-Memory (GB)
DSC-Net	63.54 ± 0.00	57.42 ± 0.00	$\mathbf{72.34 \pm 0.00}$	$100,014,991$	2.71
SSCN	$\mathbf{67.98 \pm 3.40}$	$\mathbf{58.53 \pm 3.34}$	69.48 ± 2.38	$\mathbf{66,291}$ $(-\mathbf{99.93}\%)$	$\mathbf{0.19}$ $(-\mathbf{92.96}\%)$
SSCN-OoS	67.39 ± 3.38	57.10 ± 3.27	67.16 ± 2.34	$66,291$	0.19

Table 1: Results on the MNIST dataset. Upper part: Transductive clustering of the 10,000 test images. Lower part: Inductive clustering of the 60,000 out-of-sample training images using our previously trained model. Note that our model did not see these images during training and that DSC-Net does not support clustering out-of-sample data and would require more than 4.9B parameters and 39GB of GPU-memory to cluster the whole dataset. All results are aggregated over 10 independent runs with different random initializations. For better comparability, all models use the same pre-trained auto-encoder. Results for DSC-Net exhibit no variation since the model uses constant initialization.

softmin function: $y_{ij} = {\exp\left(-\|\hat{h}_i - S_j S_j^T \hat{h}_i\|_2^2\right)}/{\sum_{k=1}^{K} \exp\left(-\|\hat{h}_i - S_k S_k^T \hat{h}_i\|_2^2\right)}$. The subspaces are fixed a-priori to be axis-aligned and don't need to be optimized. The matrix R is optimized such that classifications agree with self-expression affinities. For now, we compute the coefficient matrix of the training set using our trained model and then apply the same post-processing as in [Ji et al., 2017b] to obtain pseudo-labels which are used to train the classifier using *cross-entropy* loss and the *Cayley-Adam* algorithm [Li et al., 2020]. In future work, we plan to employ a triplet-loss [Hermans et al., 2017] which does not rely on spectral clustering and to additionally optimize the remaining model parameters with feedback from the classifier.

To account for non-linearity, we can simply add an auto-encoder to our model with the task of mapping the original data into a d_Z-dim. latent space in which the data better fits in linear subspaces and additionally the independence assumption can be better satisfied. This non-linear transformation is learned together with the rest of the model. We formalize our complete model in Definition 3.

Definition 3 (Siamese Subspace Clustering Network (SSCN))**.**

$$\begin{aligned} \min_{\theta_e, \theta_d, \theta_h, R} \quad & \frac{1}{2}\|Q\|_F^2 + \frac{\lambda_1}{2}\|Z - ZQ\|_F^2 + \frac{\lambda_2}{2}\left\|X - \widehat{X}\right\|_F^2 + \lambda_3 \mathcal{L}_{clf}(R; X) \\ \text{s.t.} \quad & Q = H^T H, \quad H = h\left(Z; \theta_h\right), \\ & Z = enc\left(X; \theta_e\right), \quad \widehat{X} = dec\left(ZQ; \theta_d\right) \end{aligned} \tag{3}$$

Above, $Z \in \mathbb{R}^{d_Z \times N}$ are non-linear embeddings of the input X computed by the encoder function enc. After self-expression in latent space, X is reconstructed as $\widehat{X} \in \mathbb{R}^{d_X \times N}$ using dec, a decoder function matching enc. The reconstruction loss ensures that the learned embeddings are actually compatible with the original data and prevents trivial solutions. Note that training with the full data batch X would lead to materialization of the full $N \times N$ coefficient matrix Q. This is not an issue for our model, however, since it can be trained with mini-batches and thus scale to large datasets. The only requirements are that batches need to be sampled uniformly at random and that the batch-size needs to be sufficiently large so that we sample enough instances from each class on average and thus obtain a representative sample. An illustration of the data flow is provided in Figure 1.

3 Experiments

As a first proof of concept, we compare our model with *DSC-Net* [Ji et al., 2017b] on the *MNIST* dataset [LeCun, 1998]. By default, we use a small convolutional auto-encoder and the same parameter settings for both models wherever possible. For SSCN, we model h as a single linear layer without bias as motivated above and train with a batch-size of 1000. The results are summarized in Figure 1. We can see that our model provides competitive performance while drastically reducing the required number of model parameters and GPU-memory. Even large amounts of out-of-sample data can be clustered reliably without any memory overhead. All hyper-parameter values and complete code for reproducing the reported results are provided in our public code repository. In future work we plan to train our model end-to-end using a triplet-loss and additional feedback from the classifier to the encoder and self-expression module. We further plan to evaluate on more datasets, with different architectural choices and against more baselines.

Acknowledgments and Disclosure of Funding

This work has been partially funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibilities for its content.

References

R Basri and DW Jacobs. Lambertian reflectance and linear subspaces. *TPAMI*, 2003.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *NeurIPS*, 1994.

Guangliang Chen and Gilad Lerman. Spectral curvature clustering (scc). *IJCV*, 2009.

Ehsan Elhamifar and René Vidal. Sparse subspace clustering. *CVPR*, 2009.

Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *TPAMI*, 2013.

Jiashi Feng, Zhouchen Lin, Huan Xu, and Shuicheng Yan. Robust subspace segmentation with block-diagonal prior. *CVPR*, 2014.

Trevor Hastie and Patrice Y Simard. Metrics and models for handwritten character recognition. *Statistical Science*, 1998.

Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

J Ho, Ming-Husang Yang, Jongwoo Lim, Kuang-Chih Lee, and D Kriegman. Clustering appearances of objects under varying illumination conditions. *CVPR*, 2003.

Pan Ji, Mathieu Salzmann, and Hongdong Li. Efficient dense subspace clustering. *WACV*, 2014.

Pan Ji, Mathieu Salzmann, and Hongdong Li. Shape interaction matrix revisited and robustified: Efficient subspace clustering with corrupted and incomplete data. *ICCV*, 2015.

Pan Ji, Hongdong Li, Mathieu Salzmann, and Yiran Zhong. Robust multi-body feature tracker: a segmentation-free approach. *CVPR*, 2016.

Pan Ji, Ian Reid, Ravi Garg, Hongdong Li, and Mathieu Salzmann. Adaptive low-rank kernel subspace clustering. *arXiv preprint arXiv:1707.04974*, 2017a.

Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian Reid. Deep subspace clustering networks. *NeurIPS*, 2017b.

Ken-ichi Kanatani. Motion segmentation by subspace separation and model selection. *ICCV*, 2001.

Mohsen Kheirandishfard, Fariba Zohrizadeh, and Farhad Kamangar. Multi-level representation learning for deep subspace clustering. *WACV*, 2020.

Yann LeCun. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

Jun Li, Li Fuxin, and Sinisa Todorovic. Efficient riemannian optimization on the stiefel manifold via the cayley transform. *ICLR*, 2020.

Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. *ICML*, 2010.

Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *TPAMI*, 2013.

Can-Yi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan. Robust and efficient subspace segmentation via least squares regression. *ECCV*, 2012.

Yi Ma, Harm Derksen, Wei Hong, and John Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *TPAMI*, 2007.

Brian McWilliams and Giovanni Montana. Subspace clustering of high-dimensional data: a predictive approach. *DMKD*, 2014.

Vishal M Patel and René Vidal. Kernel sparse subspace clustering. *ICIP*, 2014.

Vishal M Patel, Hien Van Nguyen, and René Vidal. Latent space sparse subspace clustering. *ICCV*, 2013.

Xi Peng, Shijie Xiao, Jiashi Feng, Wei-Yun Yau, and Zhang Yi. Deep subspace clustering with sparsity prior. *IJCAI*, 2016.

Mostafa Rahmani and George Atia. Innovation pursuit: A new approach to the subspace clustering problem. *ICML*, 2017.

Junghoon Seo, Jamyoung Koo, and Taegyun Jeon. Deep closed-form subspace clustering. *ICCV Workshops*, 2019.

René Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 2011.

René Vidal and Paolo Favaro. Low rank subspace clustering (lrsc). *Pattern Recognition Letters*, 2014.

René Vidal, Roberto Tron, and Richard Hartley. Multiframe motion segmentation with missing data using powerfactorization and gpca. *IJCV*, 2008.

Yu-Xiang Wang, Huan Xu, and Chenlei Leng. Provable subspace clustering: When lrr meets ssc. *NeurIPS*, 2013.

Shijie Xiao, Mingkui Tan, Dong Xu, and Zhao Yang Dong. Robust kernel low-rank representation. *IEEE transactions on neural networks and learning systems*, 2016.

Allen Y Yang, John Wright, Yi Ma, and S Shankar Sastry. Unsupervised segmentation of natural images via lossy data compression. *CVIU*, 2008.

Ming Yin, Yi Guo, Junbin Gao, Zhaoshui He, and Shengli Xie. Kernel sparse subspace clustering on symmetric positive definite manifolds. *CVPR*, 2016.

Chong You, Chun-Guang Li, Daniel P Robinson, and René Vidal. Oracle based active set algorithm for scalable elastic net subspace clustering. *CVPR*, 2016a.

Chong You, Daniel Robinson, and René Vidal. Scalable sparse subspace clustering by orthogonal matching pursuit. *CVPR*, 2016b.

Junjian Zhang, Chun-Guang Li, Chong You, Xianbiao Qi, Honggang Zhang, Jun Guo, and Zhouchen Lin. Self-supervised convolutional subspace clustering network. *CVPR*, 2019a.

Tong Zhang, Pan Ji, Mehrtash Harandi, Richard Hartley, and Ian Reid. Scalable deep k-subspace clustering. *ACCV*, 2018.

Tong Zhang, Pan Ji, Mehrtash Harandi, Wenbing Huang, and Hongdong Li. Neural collaborative subspace clustering. *ICML*, 2019b.

Pan Zhou, Yunqing Hou, and Jiashi Feng. Deep adversarial subspace clustering. *CVPR*, 2018.