Deep Density Clustering of Unconstrained Faces
(Supplementary Material)

Wei-An Lin Jun-Cheng Chen Carlos D. Castillo Rama Chellappa
University of Maryland, College Park
walin@umd.edu pullpull@cs.umd.edu carlos@cs.umd.edu rama@umiacs.umd.edu

A. Mathematical Details

In this section, we first provide the two core mathematical formulations and then present detailed proofs for Lemma 1 and Theorem 1.

SVDD formulation:

$$\begin{aligned} \min_{\boldsymbol{c}, \bar{R}, \boldsymbol{\xi}} \quad & \bar{R} + \frac{1}{\nu \cdot n_V} \sum_{\boldsymbol{z} \in V(\boldsymbol{x})} \xi(\boldsymbol{z}) \\ \text{s.t.} \quad & \|\Psi_\theta(\boldsymbol{z}) - \mathbf{c}\|^2 \leq \bar{R} + \xi(\boldsymbol{z}), \\ & \xi \geq 0, \quad \forall \boldsymbol{z} \in V(\boldsymbol{x}), \end{aligned} \tag{1}$$

OC-SVM formulation:

$$\begin{aligned} \min_{\boldsymbol{w}, \rho, \boldsymbol{\xi}} \quad & \frac{1}{2} \|\boldsymbol{w}\|^2 + \frac{1}{\nu \cdot n_V} \sum_{\boldsymbol{z} \in V(\boldsymbol{x})} \xi_{\boldsymbol{z}} - \rho \\ \text{s.t.} \quad & \boldsymbol{w}^T \Psi_\theta(\boldsymbol{z}) \geq \rho - \xi_{\boldsymbol{z}}, \\ & \xi_{\boldsymbol{z}} \geq 0, \quad \forall \boldsymbol{z} \in V(\boldsymbol{x}). \end{aligned} \tag{2}$$

A.1. Proof of Lemma 1

Lemma 1. *If $1/n_V < \nu \leq 1$, the SVDD formulation in (1) is equivalent to the OC-SVM formulation in (2) when the evaluation functions for the two are given by*

$$h_{SVDD}(\boldsymbol{x}) = \bar{R}^* - \|\Psi_\theta(\boldsymbol{x}) - \boldsymbol{c}^*\|^2, \tag{3}$$

$$h_{OC\text{-}SVM}(\boldsymbol{x}) = \boldsymbol{w}^{*T} \Psi_\theta(\boldsymbol{x}) - \rho^*, \tag{4}$$

with the correspondence $\boldsymbol{w}^* = \boldsymbol{c}^*$, and $\rho^* = \boldsymbol{c}^{*T} \Psi_\theta(\boldsymbol{x}_s)$, where $\boldsymbol{x}_s$ is a support vector in (1) that lies on the learned enclosing sphere.

Proof. The condition corresponds to the case $1/n_V \leq C < 1$ in [1] with $C = 1/(\nu \cdot n_V)$. We introduce the kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Psi_\theta(\boldsymbol{x}_i)^T \Psi_\theta(\boldsymbol{x}_j)$. Since $K(\boldsymbol{x}_i, \boldsymbol{x}_i)$ is constant in our setting, the same dual formulation for (1) and (2) can be written as:

$$\min_{\boldsymbol{\alpha}} \sum_{ij} \alpha_i \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad \text{s.t.} \quad 0 \leq \alpha_i \leq C, \ \sum_{i=1}^{n_V} \alpha_i = 1.$$

Let $S = \{i \mid 0 < \alpha_i < C\}$. We have the following results:

$$\boldsymbol{c}^* = \sum_{i=1}^{n_V} \alpha_i \Psi_\theta(\boldsymbol{x}_i), \quad \bar{R}^* = \|\Psi_\theta(\boldsymbol{x}_s) - \boldsymbol{c}^*\|^2, \tag{5}$$

$$\boldsymbol{w}^* = \sum_{i=1}^{n_V} \alpha_i \Psi_\theta(\boldsymbol{x}_i), \quad \rho^* = \boldsymbol{w}^{*T} \Psi_\theta(\boldsymbol{x}_s), \tag{6}$$

where $s \in S$. Substituting into (3) and (4), we obtain

$$h_{SVDD}(\boldsymbol{x}) = 2 \cdot h_{OC\text{-}SVM}(\boldsymbol{x}) = 2 \left[\sum_{i=1}^{n_V} \alpha_i K(\boldsymbol{x_i}, \boldsymbol{x}) - \rho^* \right]. \tag{7}$$

$\square$

A.2. Proof of Theorem 1

Theorem 1. *If $1/n_V < \nu \leq 1$ and $\boldsymbol{c}^{*T} \Psi_\theta(\boldsymbol{x}_s) \neq 0$ for some support vector $\boldsymbol{x}_s$, $h_{SVDD}(\boldsymbol{x})$ defined in (3) is asymptotically a Parzen window density estimator in the feature space with Epanechnikov kernel.*

Proof. Given the condition, according to Lemma 1, $h_{SVDD}(\boldsymbol{x})$ is equivalent to $h_{OC\text{-}SVM}(\boldsymbol{x})$ with $\rho^* \neq 0$. From the results in [10] and the fact that $\sum \alpha_i = 1$, we obtain:

$$\begin{aligned} h_{OC\text{-}SVM}(\boldsymbol{x}) &= \sum_{i=1}^{n_V} \alpha_i \left[1 - \frac{1}{2} \|\Psi_\theta(\boldsymbol{x}) - \Psi_\theta(\boldsymbol{x}_i)\|^2 \right] - \rho^* \\ &= \frac{8}{3} \sum_{i=1}^{n_V} \alpha_i K_E \left(\frac{\|\Psi_\theta(\boldsymbol{x}) - \Psi_\theta(\boldsymbol{x}_i)\|}{2} \right) - \rho^* - 1, \end{aligned}$$

where $K_E(u) = \frac{3}{4}(1 - u^2)$, $|u| \leq 1$ is the Epanechnikov kernel. As a consequence of Proposition 4 in [10] and the proof of Proposition 1 in [11], as $n_V \to \infty$, the fraction of support vector is ν, and the fraction of points with $0 < \alpha_i < 1/(\nu \cdot n_V)$ vanishes. Therefore, either $\alpha_i = 0$ or $\alpha_i = 1/(\nu \cdot n_V)$. We introduce the notation $\bar{S} = \{i \mid \alpha_i =$

$1/(\nu \cdot n_V)\}$. Then asymptotically,

$$h_{OC\text{-}SVM}(\boldsymbol{x}) = \frac{8}{3\nu n_V} \sum_{s \in \bar{S}} K_E \left(\frac{\|\Psi_\theta(x) - \Psi_\theta(x_s)\|}{2} \right) - \rho^* - 1,$$
$$= \frac{2^{d+3}}{3} \hat{f}\left(\Psi_\theta(\boldsymbol{x})\right) - \rho^* - 1, \quad (8)$$

where $\hat{f}(\boldsymbol{z}) = \frac{1}{\nu \cdot n_V \cdot 2^d} \sum_{s \in \bar{S}} K_E \left(\frac{\|\boldsymbol{z}_s - \boldsymbol{z}\|}{2} \right)$ is a density estimator. As a result, $h_{SVDD}(\boldsymbol{x})$ is equivalent to a Parzen window density estimator with Epanechnikov kernel of bandwidth 2. By scaling properly, Parzen window estimator with different bandwidths can be obtained. □

B. Implementation Details

We adopt the network architecture presented in [16]. The network is first trained on the CASIA-WebFace dataset [14] using SGD for 750K iterations with a standard batch size 128 and momentum 0.9. The learning rate is set to 0.01 initially and is halved every 100K iterations. The weight decay rates of all the convolutional layers are set to 0, and the weight decay of the final fully connected layer is set to 5×10^{-4}. Then, the model is finetuned with the MSCeleb-1M dataset [6] using the learning rate 1×10^{-4} for all the convolutional layers, and 1×10^{-2} for the fully connected layers. The network is then trained with additional 230K iterations. The inputs to the networks are $100 \times 100 \times 3$ RGB images. Data augmentation is performed by randomly cropping and horizontally flipping face images. Given a face image, the deep representation is extracted from the `pool5` layer with dimension 320.

C. Baseline Methods

- Agglomerative Hierarchical Clustering (AHC) [5]: The conventional hierarchical clustering algorithm.
- K-means [8]: The classic K-means algorithm.
- Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [2]: A well-known density-based clustering method. We set the parameter *MinPts* $= 5$.
- Affinity Propagation (AP) [3]: Affinity Propagation groups data points based on the concept of "message passing". It automatically finds exemplars and determines the number of clusters.
- Sparse Subspace Clustering using Orthogonal Matching Pursuit (SSC-OMP) [15]: SSC-OMP is a competitive method and runs faster than the classic SSC.
- Joint Unsupervised Learning of deep representations and clusters (JULE) [13]: JULE initializes each image as a cluster. It then iteratively merges images in feature space and updates network parameters.
- Deep Embedded Regularized Clustering (DEPICT) [4]: DEPICT is an efficient image clustering method that runs faster than JULE while attaining comparable performance.
- Proximity-Aware Hierarchical Clustering (PAHC) [7]: PAHC exploits neighborhood similarity based on linear SVMs. This method achieves high clustering performance on several face datasets.
- Approximate Rank-Order Clustering (ARO) [9]: ARO measures pairwise similarity based on shared nearest neighbors. The method is computationally efficient and is highly scalable.
- Conditional Pairwise Clustering (ConPaC) [12]: ConPaC builds a discriminative conditional random field model on the adjacency matrix, and then infers the parameters using the loopy belief propagation. This method outperforms several clustering algorithms on challenging face datasets.

D. Additional Evaluations on the IJB-B dataset

Table 1 reports the F-measure and NMI comparisons on the three subtasks in IJB-B. Table 2 summarizes the statistics of these subtasks.

Dataset	*IJB-B-32*		*IJB-B-64*		*IJB-B-512*	
	F	NMI	F	NMI	F	NMI
K-means [8]	0.659	0.806	0.677	0.837	0.555	0.839
AHC [5]	0.845	0.915	0.814	0.912	0.746	**0.918**
AP [3]	0.513	0.814	0.508	0.831	0.422	0.847
DBSCAN [2]	0.825	0.896	0.751	0.885	0.696	0.888
SSC-OMP [15]	0.361	0.575	0.275	0.539	0.111	0.521
PAHC [7]	0.798	0.891	0.786	0.898	0.650	0.882
ARO* [9]	0.667	-	0.574	-	0.410	-
ConPaC* [12]	0.751	-	0.656	-	0.481	-
DDC	0.827	0.906	0.783	0.903	0.733	0.909
DDC-NEG	**0.851**	**0.919**	**0.818**	**0.915**	**0.761**	**0.918**

Table 1: BCubed F-measure and NMI performance comparisons. Results reported from the original papers are marked by asterisks (*). The best performance is reported in **bold**.

Dataset	# Samples	# Subjects
IJB-B-32	1,026	32
IJB-B-64	2,080	64
IJB-B-512	18,251	512

Table 2: Statistics for the three IJB-B subtasks.

References

[1] W.-C. Chang, C.-P. Lee, and C.-J. Lin. A revisit to support vector data description (svdd), 2013. 1

[2] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD, pages 226–231, 1996. 2

[3] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315, 2007. 2

[4] K. Ghasedi Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[5] K. C. Gowda and G. Krishna. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognition*, 10(2):105–112, 1978. 2

[6] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large scale face recognition. In *European Conference on Computer Vision*. Springer, 2016. 2

[7] W.-A. Lin, J.-C. Chen, and R. Chellappa. A proximity-aware hierarchical clustering of faces. In *IEEE Conference on Automatic Face and Gesture Recognition (FG)*, 2017. 2

[8] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. 2

[9] C. Otto, D. Wang, and A. K. Jain. Clustering millions of faces by identity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (99), 2017. 2

[10] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7), July 2001. 1

[11] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, May 2000. 1

[12] Y. Shi, C. Otto, and A. K. Jain. Face clustering: Representation and pairwise constraints. *CoRR*, abs/1706.05067, 2017. 2

[13] J. Yang, D. Parikh, and D. Batra. Joint unsupervised learning of deep representations and image clusters. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[14] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 2

[15] C. You, D. Robinson, and R. Vidal. Scalable sparse subspace clustering by orthogonal matching pursuit. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[16] J. Zheng, J.-C. Chen, N. Bodla, V. M. Patel, and R. Chellappa. Vlad encoded deep convolutional features for unconstrained face verification. In *IEEE International Conference on Pattern Recognition*, 2016. 2