Learning to Cluster Faces via Confidence and Connectivity Estimation
Supplementary Material

Lei Yang\textsuperscript{1}, Dapeng Chen\textsuperscript{2}, Xiaohang Zhan\textsuperscript{1}, Rui Zhao\textsuperscript{2}, Chen Change Loy\textsuperscript{3}, Dahua Lin\textsuperscript{1}
\textsuperscript{1}The Chinese University of Hong Kong
\textsuperscript{2}SenseTime Group Limited, \textsuperscript{3}Nanyang Technological University
\{y1016, zx017, dhlin\}@ie.cuhk.edu.hk, \{chendapeng, zhaorui\}@sensetime.com, ccloy@ntu.edu.sg

1. Pseudo-code of the proposed algorithm

We provide a pseudo-code to illustrate the steps of the proposed method.

\begin{algorithm}[H]
\caption{Clustering via Confidence and Connectivity Estimation}
\textbf{Input:} Graph $G$, portion of vertices using GCN-E $\rho$, number of connections $M$, cut-off threshold $\tau$
\textbf{Output:} Clusters $C$
1: Vertex confidence $V = \text{GCN-E}(G)$
2: $S = \text{getCandidateSet}(V)$
3: $H = \text{getHighConfidenceVertexSet}(V, \rho)$
4: for $i \in H$ do
5: \hspace{1em} Edge connectivity $E_i = \text{GCN-E}(S_i, M)$
6: end for
7: for $i \in V \setminus H$ do
8: \hspace{1em} Edge connectivity $E_i = \text{Max}(E(S_i), M)$
9: end for
10: Clusters $C = \text{connectToClusters}(E, \tau)$
11: return $C$
\end{algorithm}

2. Detailed settings of compared methods

(1) \textbf{K-means} \cite{2}, minimizes the total intra-cluster variance with a given number of clusters. For $N = 584K$ of MS-Celeb-1M or DeepFashion, we employ K-means by adopting the ground-truth number of clusters. For $N \geq 1.74M$, we use mini-batch K-means with batch size 1,000.

(2) \textbf{HAC} \cite{6}, adopts \textit{single} strategy for bottom-up merging in our experiments. The distance threshold is set to 0.72 for different scale of MS-Celeb-1M. For DeepFashion, we tune the distance threshold from 0.1 to 0.9 with a step 0.1 and find 0.4 gives the best result.

(3) \textbf{DBSCAN} \cite{3}, has two important hyper-parameters, namely, $\text{radius}$ and $\text{minPts}$. For higher efficiency, we apply $K$NN DBSCAN, which only considers its $K$ nearest neighbors for density computation. We set $K = 80, \text{radius} = 0.25, \text{minPts} = 1$ for $584K, 1.74M$ and $2.89M$ of MS-Celeb-1M. When the number of unlabeled images is larger than 4.05M, we have to decrease the distance threshold $\tau$ from 0.25 to 0.2, otherwise the pairwise precision will go down to 1.46\%. For DeepFashion, we set $K = 4, \text{radius} = 0.1, \text{minPts} = 2$.

(4) \textbf{MeanShift} \cite{2}, fails to yield results in a reasonable time even on $584K$ of MS-Celeb-1M. Therefore, we only apply the approach in DeepFashion. We tune the bandwidth from 0.1 to 0.9 and find 0.5 gives the best result.

(5) \textbf{Spectral} \cite{5}, has $N \times N$ space complexity, incurring excessive memory demands even on the smallest setting of MS-Celeb-1M ($584K$). We employ spectral clustering on DeepFashion by setting the number of clusters to 3,991, which is the ground-truth number of clusters.

(6) \textbf{ARO} \cite{1}, depends on the number of nearest neighbors $K$. For the reported results of MS-Celeb-1M, we use $K = 80$ for all scales. When increasing $K$ to 500, it takes 21h to yield $F_p = 54.47$ on $584K$ of MS-Celeb-1M. For DeepFashion, we vary $K$ from 5 to 30 and the best result appears when $K = 10$.

(7) \textbf{CDP} \cite{9}, adopts a dynamic threshold algorithm to partition the affinity graph efficiently, which relies on an initial threshold $\tau$, a threshold step $\Delta \tau$, maximum size of clusters $s_{max}$ and $K$ for constructing $K$NN affinity graph. For all scales of MS-Celeb-1M, we set $\tau = 0.7, \Delta \tau = 0.05, s_{max} = 300$ and $K = 80$. For DeepFashion, we set $\tau = 0.5, \Delta \tau = 0.05, s_{max} = 200$ and $K = 2$.

(8) \textbf{L-GCN} \cite{7}, adopts the pseudo label propagation algorithm of CDP. In addition to $\tau, \Delta \tau$ and $s_{max}$, it requires $K$ at each hop $K_h$ to construct instance pivot graph and active connections $c$ for aggregating the predictions. For $584K$ and $1.74M$ of MS-Celeb-1M, we set $K_0 = 80, K_1 = 10, c = 10, \tau = 0.6, \Delta \tau = 0.05$ and $s_{max} = 300$. For $N \geq 2.89M$, we increase $\tau$ to 0.7 and $s_{max}$ to 900, while keeping other hyper-parameters the same. For DeepFashion, we set $K_0 = 5, K_1 = 5, c = 5, \tau = 0.5, \Delta \tau = 0.05$ and $s_{max} = 300$.

(9) \textbf{LTC} \cite{8}, for $N = 584K$ of MS-Celeb-1M, we adopt the same strategy of LTC, which sets different $K$ and $\tau$. 

generating a large number of proposals iteratively. For \( N \geq 1.74M \), to control the computational budget, we set \( K = 80, s_{\text{max}} = 300, \Delta \tau = 0.05 \) and generate cluster proposals using 5 thresholds ranging from 0.55 to 0.75 with a step of 0.05, without resorting to the iterative scheme. For DeepFashion, we set \( K = 5, s_{\text{max}} = 100, \tau = [0.55, 0.6] \). Adding proposals generated with \( \tau = [0.65, 0.7] \) only increases the \( F_P \) from 29.14 to 29.5, while increasing the runtime from 13s to 27s.

(10) **Ours (V)**, the proposed method mainly relies on two hyper-parameters, namely \( K \) and cut off threshold \( \tau_c \). For all settings, we set \( \tau_c = 0.8 \). To construct the \( K\)NN graph, we set \( K = 80 \) for MS-Celeb-1M and \( K = 5 \) for DeepFashion, respectively. For GCN-V, one hidden layer is adopted with a hidden dimension of 512.

(11) **Ours (V + E)**, introduces GCN-E module to select top \( \rho \) vertices for connectivity estimation and top-\( M \) prediction for connection. For both MS-Celeb-1M and DeepFashion, we set \( \rho = 0.7 \) for training and \( \rho = 0.8 \) for inference. \( M \) is set to 1 for all settings. To better evaluate the neighborhood of each vertex, we can use different \( K \) nearest neighbors for GCN-V and GCN-E. For MS-Celeb-1M, we use \( K = 80 \) for both GCN-V and GCN-E. For DeepFashion, we use \( K = 5 \) for GCN-V and \( K = 80 \) for GCN-E.

References


