An Efficient Person Clustering Algorithm for Open Checkout-free Groceries

Junde Wu[1], Yu Zhang[2], Rao Fu[2], Yuanpei Liu[3], and Jing Gao[1⋆]

[1] Purdue University, West Lafayette, IN, 47907, USA jinggao@purdue.edu
[2] Harbin Institute of Technology, Harbin, 150080, China
[3] Beijing Institute of Technology, Beijing, 102213, China

Abstract. Open checkout-free grocery is the grocery store where the customers never have to wait in line to check out. Developing a system like this is not trivial since it faces challenges of recognizing the dynamic and massive flow of people. In particular, a clustering method that can efficiently assign each snapshot to the corresponding customer is essential for the system. In order to address the unique challenges in the open checkout-free grocery, we propose an efficient and effective person clustering method. Specifically, we first propose a Crowded Sub-Graph (CSG) to localize the relationship among massive and continuous data streams. CSG is constructed by the proposed Pick-Link-Weight (PLW) strategy, which **picks** the nodes based on time-space information, **links** the nodes via trajectory information, and **weighs** the links by the proposed von Mises-Fisher (vMF) similarity metric. Then, to ensure that the method adapts to the dynamic and unseen person flow, we propose Graph Convolutional Network (GCN) with a simple Nearest Neighbor (NN) strategy to accurately cluster the instances of CSG. GCN is adopted to project the features into low-dimensional separable space, and NN is able to quickly produce a result in this space upon dynamic person flow. The experimental results show that the proposed method outperforms other alternative algorithms in this scenario. In practice, the whole system has been implemented and deployed in several real-world open checkout-free groceries.

Keywords: Open Checkout-free Groceries, Person Clustering, Graph Convolutional Network

1 Introduction

Traditional checkout-free groceries are in small closed venues with limited commodities. Customers are required to register upon entry, which may cause privacy and security issues. Recently, a concept of open checkout-free grocery is proposed to address the existing issues. Open checkout-free grocery allows free entry of consumers without registering customer information. Customers can

⋆ Junde Wu and Yu Zhang equally contributed.
Jing Gao is the corresponding author.

walk in such grocery stores just like what they do in the traditional supermarkets while enjoying the benefit of automatic checkout. To achieve this goal, it is essential to automatically identify the customers in the grocery, which is a very challenging task in an open environment. The number of needed identifications is not accessible beforehand, and the identification process needs to work continuously and steadily for every customer. Therefore, many tools in computer vision to associate/track a person, e.g., human tracking, person re-identification, or face verification, cannot be directly applied in this scenario. Instead, a person clustering component is essential to cluster the customers based on features extracted from their video snapshots. Implementing an effective clustering algorithm in this scenario is non-trivial. Among many challenges, below we discuss the two major challenges of this problem.

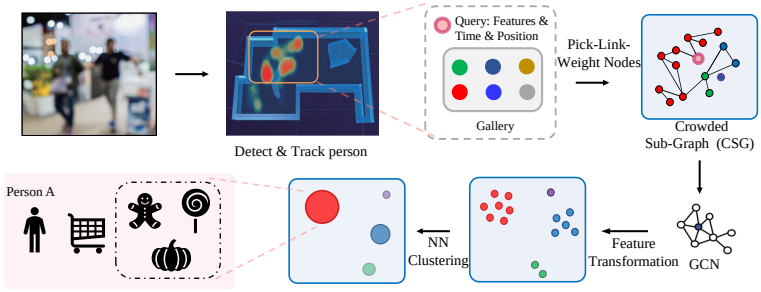

Fig. 1. Overview of the person clustering algorithm in open checkout-free groceries.

The first challenge comes from the difficulty of handling data streams [2], which are the input of the person clustering algorithm. The proposed clustering algorithm has to work over massive, unbounded sequences of data objects that are continuously generated at rapid rates. In addition, the algorithm must react immediately when a *query* comes, under an extremely harsh condition where the observed data stream is generally too large to store and too expensive to access.

The second challenge is that person clustering in this scenario is an open-world problem [5]. The data that the system needs to process is dynamic and unknown person flow. Therefore, the system is required to continuously distinguish and memorize the unseen new customer once an individual comes and could recognize the person in any other locations and views since then.

For the first challenge, the commonly applied method is to cluster locally on the summarized data (statistic summaries of data streams) [2]. In this paper, we propose a Crowded Sub-Graph (CSG) to efficiently collect the local clusters and provide the divergence of the summarized data. CSG constructs a local relationship network for each incoming *query* through the Pick-Link-Weight (PLW) strategy. Specially, we leverage time-space information to **pick** the nodes,

and utilize trajectory information to **link** the nodes, and propose a novelty von Mises-Fisher (vMF) similarity metric to **weigh** the links. With the CSG, we can compare locally to accelerate the system without degrading performance.

The open-world nature of the scenario leads to dynamic changes in the distribution of person features. Thus, some complex clustering techniques may consume too much time, while some simple techniques may lead to performance degradation. In this paper, we propose Graph Convolutional Network (GCN) with a simple Nearest Neighbor (NN) strategy to accurately cluster the instances of CSG. GCN is adopted to project the features into low-dimensional separable space, and NN is able to quickly produce a result in this space upon dynamic person flow. To our knowledge, our work is the first towards a comprehensive strategy to identify a person in this data-stream & open-world environment.

In brief, the contributions of the paper are as follows.

- We are the first to define the People Clustering task in an open checkout-free grocery scenario, and we propose an effective framework to address this important problem. As the first research report of this scene, we believe it will be an essential starting point for future research and practical applications.
- We propose Crowded Sub-Graph (CSG), a local relationship graph constructed by PLW strategy. PLW strategy can model the distance of nodes through the lens of the probability distribution, so as to construct a subgraph that fairly represents the relevance of local nodes.
- Given CSG as input data, we apply Graph Convolutional Network (GCN) on it following a simple NN (Nearest Neighbor) strategy to cluster the nodes, which is able to quickly adapt to the dynamic person distribution in groceries. Experimental results show the proposed algorithm considerably outperforms best alternative methods.

2 Related Works

2.1 Data-stream clustering

For the last decade, we have seen an increasing interest in managing the data streams [2, 14]. Clustering on data streams requires a process to continuously cluster data objects within memory and time restrictions [14].

In the data abstraction step, the data structures to summarize the data are also diversely proposed for the different tasks [25], like feature vectors [41], prototype arrays [10], coreset trees [1], and data grids [20]. However, most of them are bound with particular clustering methods, which narrows their applications.

In the clustering step, many k-means variants have been presented to deal with summarized features to cluster in data streams in real-time. Bradley et al. [6] proposed Scalable k-means, which uses the CF vectors of the processed and new data objects as input to find k clusters. The ClusTree algorithm [20] proposes to use a weighted CF vector, which is kept into a hierarchical tree (R-tree family). These well-designed data-stream clustering methods, on the one hand, to date, are limited to Euclidean spaces, and on the other hand, hard to take a balance between efficiency and performance.

2.2 GCN on clustering

Recently, with a part of the various applications of deep neural networks [16, 32–34, 36], Graph Convolutional Network (GCN) has shown outstanding performance on data clustering. GCN can extract high-level node representations, thus simplifying the sensitive discrimination step [35].

To apply GCN on the data which naturally has the graph structure seems straightforward, such as graph-based recommendation systems [27], point clouds classification [21], and molecular properties prediction [22], etc. However, the graph nature of some other data may not be so explicit. In this case, researchers have to construct the graph of the data. For example, [40] addressed the traffic prediction problem using STGNNs. [29] applied the GCN to text classification based on the syntactic dependency tree of a sentence. [30] proposed Instance Pivot Subgraph (IPS) to construct the sub-graph for person face features. However, despite the outstanding performance of GCN on data clustering, there is still a research gap between GCN and data-stream clustering.

3 Preliminary

3.1 Data-stream clustering

A data stream S is a massive sequence of data objects $x_1, x_2, ..., x_K$, that is, $S = \{x_i\}_{i=1}^{K}$, which is potentially unbounded ($K \to \infty$). Each data object is described by a n-dimensional attribute vector $x_i = [x_i^j]_{j=1}^{n}$ belonging to an attribute space Ω that can be continuous, categorical, or mixed.

It is impossible to store and get access to each data object in the data stream. Developing suitable data structures to store statistic summaries of data streams is indispensable in the data-stream clustering tasks. Cluster Feature vector (CF vector) is a commonly used data structure for summarizing large amounts of data. The CF vector has three components: K, the number of data objects, LS, the linear sum of the data objects, and SS, the sum of squared data objects. The structures LS and SS are n-dimensional vectors. These three components allow to compute cluster measures, such as cluster mean μ and radius σ (Eq. (1)).

$$\mu = \frac{LS}{K},\ \sigma = \sqrt{\left(\frac{SS}{K} - \left(\frac{LS}{K}\right)^2\right)}, \tag{1}$$

where $(\cdot)^2$ and $\sqrt{\cdot}$ represent element-wise square and square root. Obviously, the three components of the CF vector have incrementality and additivity properties, which make the CF vector widely used in clustering (More details can be found in the supplementary material, or [25]).

In the open checkout-free groceries, a complete *person record* p is represented as a set (Eq. (2))

$$p = \{\{\tilde{t}_i\}_{i=1}^{K}, \{z_i\}_{i=1}^{K}, \{v_i\}_{i=1}^{K}, CF[K]\}, \tag{2}$$

where $\tilde{t}_i = t_i^s + t_i^e$, t_i^s and t_i^e are the time stamps of the person appeared in the camera view and left the camera view, respectively. z_i is a two dimensional point records the plane coordinates of the camera, v_i is a two-dimensional normalized vector to denote the direction of the pedestrian's walking. It has $v_i = (\cos(\theta), \sin(\theta))$, where θ is the angle between the last straight pedestrian path in the camera and the horizontal line. The pedestrian path is got from the move path of the bounding box. $CF[K]$ represents incremented CF vectors of person features updated K times. $K \geq 1$ is the number of pieces of data incremented in p. Each coming data was a *query*, which is represented as:

$$q = \{\tilde{t}, z, v, CF[1]\}, \tag{3}$$

where $CF[1]$ represents the initial tracked person features.

4 Methodology

The complete abstracted features in open checkout-free grocery combined several different sensors, and the vision system is one of the most important parts. The visual features are obtained through a flow of object tracking [24], person detection [7], image deblurring [31], image enhancement [42] and deep learning-based person feature abstraction [3]. When the camera captures a customer, it tracks the customer until the individual is out of view. We would sample several frames from this track and use a person detection algorithm to abstract a set of images of the person. These pictures will then be sent to the pre-trained neural networks to abstract a set of visual features. These visual features, combined with the appeared time-space information and person walking track, will be sent to the clustering algorithm to be identified. We call this piece of data sent to the clustering algorithm a *query*.

The overflow of our algorithm contains two steps, which are shown in Fig. 1. In the first step, when a *query* comes, we construct a Crowded Sub-Graph for this *query*. CSG contains N nodes, and one node is the *query*, and the other N-1 nodes are *person records*. The N-1 *person records* are **picked** depending on the time-space constraint to assume a person is impossible to appear in a far place in a short time. Then we **link** the nodes depending on the person walking track and **weight** these links based on the vMF divergence of the person features. In the second step, we propose Graph Convolutional Network (GCN) with a simple Nearest Neighbor (NN) strategy to accurately cluster the instances of CSG. GCN is adopted to project the features into low-dimensional separable space, and NN is able to quickly produce a result in this space upon dynamic person flow.

4.1 Crowded Sub-Graph

In this section, we introduce Crowded Sub-Graph (CSG) to construct a graph based on the raw data of open checkout-free groceries. Constructing a CSG

contains three steps: to **pick** the nodes, **link** the nodes, and **weight** the links. In this way, CSG constructed a local sub-graph for the *query*. The association of each pair of nodes in the sub-graph can be well represented.

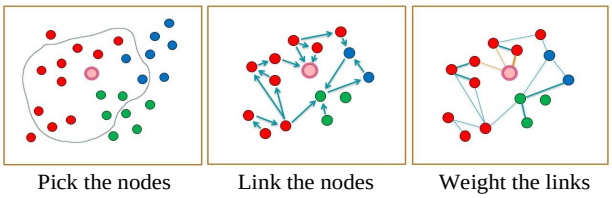

Fig. 2. Construct Crowded Sub-Graph through the proposed Pick-Link-Weight (PLW) strategy

Pick the nodes Our algorithm is required to react immediately once a *query* comes. However, clustering globally on the nearly unbounded data stream, i.e., clustering the *query* based on all of the recorded person in the grocery, is impossible. Fortunately, a customer would not move far within the two captures of the cameras. This allows us to cluster locally based on time and space constraints. Considering a *query* q captured by a camera at location z^q in time t^q, other *person records* that have a smaller time-space distance with the *query* q would have a higher possibility to be contained to the sub-graph. For a *person record* p with time and location sets $\{\tilde{t}_i^p\}_{i=1}^{K^p}$ and $\{z_i^p\}_{i=1}^{K^p}$, we define the time-space distance between p and q is:

$$\chi(p;q) = \min \sqrt{\frac{E(z_i^p, z^q)^2}{s^2} + (\tilde{t}_i^p - \tilde{t}^q)^2}, \ i \in [1, K^p], \tag{4}$$

where E denotes Euclidean distance, s is the standard human walking speed, which is set as 3 miles per hour. Then we can collect the nodes of CSG. The number of the nodes we collect depends on the setting size of CSG.

Link the nodes In the graph, two linked nodes usually mean that they are related to each other to some extent. In the open checkout-free grocery scenario, we assume two nodes are linked if they are on the same trajectory. Practically, we adopt an attentive dot-product mechanism [28] for the classification of the trajectories.

As shown in Fig. 3, the recorded information of each pair of picked nodes are treated as inputs of the attention mechanism, including the position z_i and pose v_i of person relative to cameras and time stamps t_i. To predict whether Node a is on the trajectory of node b, a QKV dot-product is adopted to activate node b value matrix based on the calculated affinity map of two nodes. A

binary classification head is applied after the attention to get the probability of the prediction. The binary classification head consists of a Global Average Pooling (GAP) layer and the Multilayer Perceptron (MLP) layer. For each piece of recorded information in the node, we use the position vector, pose vector, and timestamp to constitute the input embedding. Specially, we concatenate the position vector and pose vector, then add a temporal embedding on it. The temporal embedding is learned from the time stamp, followed by [8]. Formally, the sequential input embedding of trajectory predictor is represented as:

$$[z_i; v_i] + TE(t_i), i \in [1, K_1 + K_2], \tag{5}$$

where K_1 and K_2 are the numbers of recorded information of two nodes. $TE(\cdot)$ represents the temporal embedding. We apply a trajectory predictor on each pair of nodes to get the final link relationship of the sub-graph.

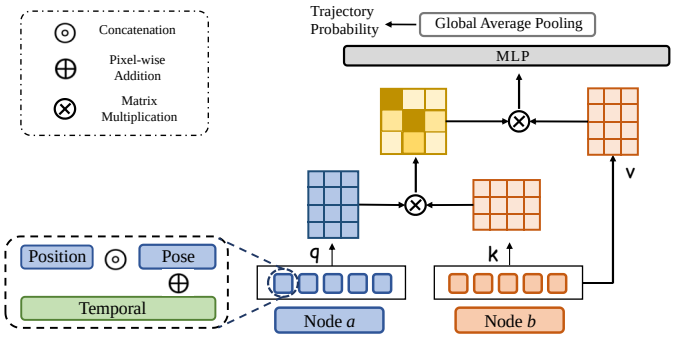

Fig. 3. Trajectory predictor which predicts whether two nodes are on the same trajectory.

Weight the link & vMF similarity of CF vectors After we **link** the nodes in the graph, we then **weight** these links by measuring how strong the linked pair is associated. Specially, we propose a similarity distance for the CF vectors based on vMF distribution and use this distance to generate the weights.

Note that each node in CSG is represented as a CF vector. Since the centroid and radius of the node can be easily computed from CF vector, previous work often represented the distance of two nodes by the discrepancy of two normal distributions. For example, a probabilistic model on the CF vector can be represented as a Gaussian model $\mathcal{N}(\mu, \sigma^2)$, where mean μ and variance σ can be easily through Equation (1). Then, we can **weight** the link of node a and node b through the distance of $\mathcal{N}(\mu_a, \sigma_a^2)$ and $\mathcal{N}(\mu_b, \sigma_b^2)$.

However, such a practice implicitly assumes the distance of the features can be fairly represented by Euclidean distance, which is invalid for the high-dimensional person features in our hand. The person features abstracted by the

neural networks have intrinsic angular distribution because of the softmax loss in the neural networks [12]. Thus the probability distributions on Euclidean space, like Gaussian distribution, are invalid on this d-sphere. In this paper, we use von Mises−Fisher (vMF) distribution instead to model these clusters [4]. The von Mises−Fisher distribution is an isotropic distribution over the d-dimensional unit hypersphere, which can fairly represent the distribution high-dimensional person feature. It has mean direction μ and concentration κ, and the probability density function of it for the d-dimensional unit vector x is given by:

$$f(\mu, \kappa) = C_d(\kappa) e^{\kappa \mu^T x}, \tag{6}$$

where $f(\cdot)$ represents the probability distribution, $\kappa \geq 0$, $\parallel \mu \parallel = 1$, and the normalization constant $C_p(\kappa)$, is equal to

$$C_d(\kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\kappa)}, \tag{7}$$

where I_v denotes the modified Bessel function of the first kind at order v. In vMF distribution, μ denotes the mean direction, and κ denotes the concentration. The greater the value of κ, the higher the concentration of the distribution around the mean direction μ. When it is needed, we can follow [26] to get the numerical solutions of κ and I_v.

Consider two vMF probability distributions of Node a and Node b are $f(\theta_a)$ and $f(\theta_b)$, and $f(\theta_{ab})$ denotes vMF distribution of their merged features. We **weight** the link l_{ab} by:

$$e^{-\frac{1}{2}(D_{JS}[f(\theta_a), f(\theta_{ab})] + D_{JS}[f(\theta_b), f(\theta_{ab})])}, \tag{8}$$

where D_{JS} denotes Jensen−Shannon divergence [13]:

$$D_{JS}[f(\theta_a), f(\theta_{ab})] = \frac{1}{2}(D_{KL}(f(\theta_a)) \| f(\theta_{ab}) + D_{KL}(f(\theta_{ab})) \| f(\theta_a)), \tag{9}$$

where D_{KL} denotes Kullback−Leibler divergence.

To get the analytic solution of the KL divergence of two vMF distribution is challenging. For the benefit of the computation and graph sparsification, we **weight** the links of similar feature distributions and set the others as zero. We consider the parameters of a distribution are close to one another, so that:

$$f(\theta_{ab}) = f(\theta_a) + \sum_j \Delta\theta^j \left. \frac{\partial f}{\partial \theta^j} \right|_{\theta_a}, \tag{10}$$

where θ^j represents a small change of θ in the j direction. Then Kullback−Leibler divergence $D_{\mathrm{KL}}[v(\theta_a) \| v(\theta_b)]$ has the second order Taylor Expansion in $\theta = \theta_0$ of the form

$$D_{\mathrm{KL}}[f(\theta_a) \| f(\theta_{ab})] = \frac{1}{2} \sum_{jk} \Delta\theta^j \Delta\theta^k g_{jk}(\theta_a) + \mathrm{O}(\Delta\theta^3), \tag{11}$$

in which

$$g_{jk}(\theta) = \int_X \frac{\partial \log f(\theta)}{\partial \theta_j} \frac{\partial \log f(\theta)}{\partial \theta_k} f(\theta)\, dx. \tag{12}$$

In our case, we have the parameter $\theta = (\kappa, \mu^T)^T$. Substituting Eqn. (12) for the given parameters, we can get

$$\begin{aligned} g_{\kappa,\kappa}(\kappa,\kappa) =& \tau_\kappa(\kappa,\mu),\ g_{\kappa,\mu}(\kappa,\mu) = \tau_{\kappa\mu}(\kappa)\mu^T, \\ g_{\mu,\kappa}(\mu,\kappa) =& \tau_{\kappa\mu}(\kappa)\mu,\ g_{\mu,\mu}(\mu,\mu) = \tau_\mu(\kappa)\mu\mu^T, \end{aligned} \tag{13}$$

in which

$$\begin{aligned} \tau_\kappa(\kappa,\mu) =& [e^{\kappa\mu^T x}(\kappa\mu^T - 1)]^2 + [\frac{d-2}{\kappa} - \frac{(I_{\frac{d}{2}-2}(\kappa) + I_{\frac{d}{2}}(\kappa))}{I_{\frac{d}{2}-1}(\kappa)}] \\ &+ x[e^{\kappa\mu^T x}(\kappa\mu^T - 1)] + [\frac{d}{2} - \frac{2(I_{\frac{d}{2}-2}(\kappa) + I_{\frac{d}{2}}(\kappa))}{\kappa I_{\frac{d}{2}-1}(\kappa)} - 1]^2, \end{aligned} \tag{14}$$

$$\tau_{\kappa\mu}(\kappa) = \frac{(d-2)I_{\frac{d}{2}-1}(\kappa) - \kappa(I_{\frac{d}{2}-2}(\kappa) - I_{\frac{d}{2}}(\kappa))}{2(I_{\frac{d}{2}-1}(\kappa))^2}, \tag{15}$$

$$\tau_\mu(\kappa) = \kappa^2. \tag{16}$$

Let us say $f(\theta_a)$ has parameters κ and μ, and $f(\theta_{ab})$ has parameters $\bar{\kappa}$ and $\bar{\mu}$. Then, according Eqn. (11), it has

$$\begin{aligned} D_{\mathrm{KL}}[f(\theta_a) \| f(\theta_{ab})] =& \frac{1}{2} \sum_{jk} \Delta\theta^j \Delta\theta^k f_{jk}(\theta_a) \\ =& \frac{1}{2}[\tau_\kappa(\bar{\kappa},\bar{\mu})(\bar{\kappa}-\kappa)^2 + 2\tau_{\kappa\mu}(\bar{\kappa})\bar{\mu}^T(\bar{\kappa}-\kappa)(\bar{\mu}-\mu) \\ &+ \tau_\mu(\bar{\kappa})(\bar{\mu}-\mu)\bar{\mu}\bar{\mu}^T(\bar{\mu}-\mu)]. \end{aligned} \tag{17}$$

After constructing CSG, we can get an adjacency matrix A. Non-zero values in the adjacency matrix indicate the existence of links between nodes. The values are normalized, and the sum of each row or column is equal to 1.

4.2 Graph convolution network

CSG is highly valuable to identify the nodes. To leverage this, we adopt a graph convolution network (GCN) to perform reasoning on CSG. Specifically, in order to adapt to the changing person features distribution caused by the open-world challenge, we use GCN to project the features into a linearly separable low dimensional space. With a simple Nearest Neighbor strategy adopted after then for the discrimination, the method can achieve high precision clustering results.

The input of the GCN is the original node feature matrix together with the adjacency matrix A. The output is the projected node feature matrix with the

same number of input features but with lower dimensions. Each graph convolution layer is inputted by the last node feature matrix together with the adjacency matrix and outputs the next node feature matrix. Formally, for the l^{th} layer, we have the following equation:

$$\mathbf{X}^{l+1} = \sigma([\mathbf{X}^l|\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{A}\mathit{\Lambda}^{-\frac{1}{2}}\mathbf{X}^l]\mathbf{W}_l), \tag{18}$$

where $\mathbf{X}^l \in \mathbb{R}^{n\times d_l}$ is the node feature matrix inputted to the l^{th} layer. n represents the number of nodes, and d^l represents the feature's dimension of the l^{th} layer. $\mathbf{\Lambda}$ is a diagonal matrix with $\mathbf{\Lambda}_{ii} = \sum_j \mathbf{A}_{ij}$. $\sigma(\cdot)$ is an nonlinear activation function. $\mathbf{W}_l \in \mathbb{R}^{2d_l\times d_{l+1}}$ is a layer-specific trainable weight matrix for the convolution layer. $\mathbf{X}^{l+1} \in \mathbb{R}^{n\times d_{l+1}}$ is the output node feature matrix of the layer.

GCN is supervised to project the nodes of the same person to be closer and others to be more distant. Toward that end, We adopt triplet loss [23] to our case. Denote the outputted features matrix as $\mathbf{O} \in \mathbb{R}^{n\times d_{out}}$, and each feature for one node as $o \in \mathbb{R}^{d_{out}}$. Given a crowded graph containing h people, and each person corresponding c node in the graph, we denote the feature of k^{th} node of i^{th} person as o_i^k, then our loss is given by:

$$\mathcal{L}_{GCN} = \sum_{i=1}^{h}\sum_{k=1}^{c_i}\max(log\sum_{r=1,r\neq k}^{c_i} e^{D(o_i^k,o_i^r)} + log\sum_{j=1,j\neq i}^{h}\sum_{s=1}^{c_j} e^{m-D(o_i^k,o_j^s)}, 0), \tag{19}$$

where m is the least margin that the projected feature is closer to the same class than the different classes. D is the distance measure, which is implemented by cosine similarity here.

In the inference stage, we use a simple Nearest Neighbor (NN) clustering strategy on the projected features. The NN has two types, the NN-A (Assign a new *query* to the existing nodes) and NN-M (Merge existing nodes). We set different distance thresholds ξ_A and ξ_M for the NN-A and NN-M, respectively. And the distance measure is set to be as same as that in the training stage. NN-A will assign the new *query* to the nearest *person records* if their distance is lower than ξ_A. If this distance is alternatively higher than the threshold ξ_A, the *query* will be left as an outlier, which represents a new *person record*. The outliers will be processed as nodes when being contained in the other CSG.

NN-M merges two *person records* together if their distance is lower than ξ_M, which is used to make up the early stage division problem (predict one customer as multiple people). In this data-stream clustering process, with the continuous arriving of the queries, a node has the chance to be actives in many different sub-graphs. This process helps to gradually attach the global information, then correct the previous errors caused by over-conservative clustering strategy.

5 Experiments

5.1 Data and Evaluation metric

It should be noted that all the customers we collected information from assigned the informed agreement before going into the grocery. In addition, all data points

Table 1. Ablation study setup. 'vMF' and 'Cos' represent applying vMF similarity and Cosine similarity in the corresponding element, respectively. The 'NN-A' represents Nearest Neighbor clustering which Assigns a *query* to existing nodes. The 'NN-M' represents Nearest Neighbor clustering which Merges existing nodes

Model \ Element	Time-Space&Track	CSG	GCN	NN-A	NN-M
Baseline	-	-	-	Cos	-
TS	√	-	-	Cos	-
TS-M	√	-	-	Cos	Cos
TS-M-vMF	√	-	-	vMF	vMF
Cos-GCN	√	Cos	√	Cos	Cos
CSG-GCN	√	vMF	√	Cos	Cos

Table 2. Ablation study results. P : BCubed Precision, R : BCubed Recall, $F = \frac{2PR}{P+R}$, T : Time (Seconds per one thousand quires)

	DaiCOFG				IseCOFG			
	P	R	F	T	P	R	F	T
Baseline	84.46	73.28	78.47	46.61	86.62	84.21	85.40	26.78
TS	91.25	80.16	85.35	**8.27**	93.31	84.02	88.42	**7.32**
TS-M	91.22	84.39	87.67	9.13	96.02	89.30	92.54	7.95
TS-M-vMF	95.75	93.70	94.71	11.06	97.47	94.51	95.97	8.72
Cos-GCN	96.83	96.37	96.60	11.81	97.73	96.91	97.32	8.96
CSG-GCN	**98.77**	**98.24**	**98.50**	13.68	**99.07**	**98.75**	**98.91**	10.54

Table 3. Comparison with SOTA. P : BCubed Precision, R : BCubed Recall, $F = \frac{2PR}{P+R}$, T : Time (Seconds per one thousand quires, omitted if > 50 s/ptq)

	DaiCOFG				IseCOFG			
	P	R	F	T	P	R	F	T
Scalable-kmeans	84.49	79.36	81.84	16.68	86.63	84.36	85.48	16.15
ClusTree	91.81	85.73	88.67	14.00	93.41	90.64	91.87	14.00
ClusterGCN	94.55	90.75	92.61	-	94.12	91.46	92.77	-
AffinityGCN	97.05	96.82	96.93	-	98.47	97.83	98.14	-
GraphSaint	95.84	92.34	94.05	46.62	96.16	95.67	95.91	28.79
IPS-GCN	96.32	94.17	95.23	**11.49**	97.15	96.30	96.72	**9.69**
GLCN	96.86	94.96	95.90	12.37	97.82	96.51	97.16	10.37
CSG-GCN	**98.77**	**98.24**	**98.50**	13.68	**99.07**	**98.75**	**98.91**	10.54

used in the clustering stage are obfuscated high-dimensional features, which do not contain any customers' personal information.

To evaluate our method in different scenes, we establish two datasets collected from a large grocery and a smaller grocery, respectively. The recurring identities across training and test set are removed to avoid bias. The dataset we collected from a large grocery is called DaiCOFG. DaiCOFG contains $362,300$ snapshots with $10,176$ identities for training, in which $125,378$ snapshots are labeled, and each identity contains at least one snapshot labeled. DaiCOFG contains $250,710$

labeled snapshots with 7,406 identities for testing. The snapshots are taken by 186 cameras deployed at the key spots of the grocery. The dataset we collected from the smaller grocery is called IseCOFG. IseCOFG contains 78,630 snapshots with 4,116 identities for training, in that 21,648 snapshots are labeled, each identity contains at least one snapshot labeled. IseCOFG contains 54,606 snapshots with 2,773 people for testing. The snapshots are taken by 76 cameras in the grocery.

To evaluate the performance of the proposed algorithm, we adopt the mainstream BCubed evaluation metrics [11,30]. Denote ground truth label and predicted label as a y and a y' respectively, the pairwise correctness is represented as:

$$Correct(i,j) = \begin{cases} 1 & y_i = y_j \ and \ y'_i = y'_j \\ 0 & otherwise \end{cases}, \tag{20}$$

If the i^{th} *query* and the j^{th} *query* belong to the same customer during clustering and labeling, we can get $Correct(i,j) = 1$. The BCubed Precision P and BCubed Recall R are respectively defined as:

$$P = \mathbb{E}_i[\mathbb{E}_{j:y'_i=y'_j}[Correct(i,j)]], \ R = \mathbb{E}_i[\mathbb{E}_{j:y_i=y_j}[Correct(i,j)]], \tag{21}$$

When taking both precision and recall into consideration, BCubed F-measure is defined as $F = \frac{2PR}{P+R}$.To evaluate the algorithms' speed, we record the seconds per one thousand quires (s/ptq) as the metric in the comparisons.

5.2 Experiment Setting

The variables in **link** operation are pre-trained on the training set of the selected dataset of the experiment. In the training stage, it is supervised by binary cross-entropy loss function with a mini-batch of 64 for 80 epochs using ADAM algorithm [18]. The learning rate is set to 0.01. We set the number of convolution layers in our GCN to 3. The number of units in the graph convolution network's hidden layer is set to 256, 128, and 64. The number of the nearby nodes is set as 256. We train GCN for a maximum of 120 epochs (training iterations) using an ADAM algorithm with a learning rate 0.01, and stop training if the validation loss does not decrease for 10 consecutive epochs, as suggested in work [19]. All the network weights θ are initialized using Glorot initialization [15]. The thresholds ξ_A and ξ_M are set as 0.91 and 0.88, respectively.

The experiments are run on the server cluster with 16 CPU: Intel Xeon Gold 5120, 256GB memory, and 8 GPU: NVIDIA Tesla P40.

5.3 Ablation Study

To show the advantages of the proposed components, we do comprehensive ablation studies. The setup and results of the ablation study are shown in Table 1 and Table 2.

As shown in Table 2, the raw NN-A strategy based only on the features' cosine similarity (Baseline) takes a long time and gets a lower recall. Applying

time-space constraint and track information (TS) helps to narrow the search range since we only compare the nodes under time-space constraints and linked through track information, which speeds up the algorithm a lot with 6.88% and 3.02% F1 score improvement on DaiCOFG and IseCOFG respectively. Applying NN-M strategy (TS-M) actually turns the method to a k-means liked algorithm. It can be seen that, NN-M strategy helps to significantly improve the recall with a slight degradation of the precision. Further applying vMF-based divergence in NN-A and NN-M (TS-M-vMF) significantly improves both the precision and the recall. Also, the processing time increased by the extra consumption. Applying GCN (Cos-GCN) helps to improve the recall, due to the low-dimensional features can be better discriminated in the early stage. Further replacing cosine similarity by vMF-based divergence to construct the graph (Cos-GCN→CSG-GCN) turns it to the proposed algorithm. It can be seen that vMF weight strategy improves 1.90% and 1.59% F1 score on DaiCOFG and IseCOFG, respectively.

From the comparison of TS-M/Cos-GCN with TS-M-vMF/CSG-GCN, it can be seen that modeling the person features by vMF distribution significantly outperforms cosine similarity based pointwise comparison. Comparing TS-M with Cos-GCN, the combination of proposed CSG and GCN shows better precision and recall even without vMF distribution modeling. This shows the effectiveness of the proposed CSG when facing the dynamic and unseen person flow. In general, from Table 2, we can see that the proposed algorithm CSG-GCN gets both high performance and processes in the tolerable time on the dataset.

5.4 Comparison with alternative methods

In this part, to verify the effectiveness of the proposed method, we compare the proposed method with a wide range of alternative clustering methods, including non-learning-based and learning-based methods. Since those methods are not designed for our scene, and most of the learning-based methods are oriented to closed data sets, we have to adapt some methods for the comparison.

Most previous methods toward data-stream clustering are not learning-based. We compare our method with scalable k-means [6] and ClusTree [20], which are two commonly used methods in the data-stream clustering tasks. Scalable k-means employs different mechanisms to identify objects that need to be retained in memory. It stores data objects in a buffer in the main memory. By utilizing CF vectors, it discards objects that were previously statistically summarized into the buffer. When the block is full, an extended version of k-means is executed over the stored data. ClusTree algorithm [20] proposed to use a weighted CF vector, which is kept into a hierarchical tree (R-tree family). ClusTree provides strategies for dealing with time constraints for anytime clustering, that is, the possibility of interrupting the process of inserting new objects in the tree at any moment.

Recently, GCN has been proved an efficient method for clustering tasks [37]. However, few works applied GCN to the data-stream clustering. We adapted ClusterGCN [9], AffinityGCN [38], GraphSaint [39], IPS (Instance Pivot Subgraph)-GCN [30] and GLCN [17] for the comparison. The results are shown in Table.

3. [9, 30, 38, 39] cluster the features by applying GCN on the constructed sub-graph. Their methods are applied to construct the subgraph for each coming *query* and then trained by our own strategy. [17] proposed to learn a graph for GCN clustering based on Euclidean distance, which is called Graph Learning Convolutional Network (GLCN). To adapt it to our data-stream scenario, we apply it on CSG linked nodes to cluster each sub-graph locally.

Since the algorithm takes more time when the grocery becomes larger, we record the algorithms' speed when the grocery is full. ClusTree is a time-adaptable method that fits our scenario well. However, when we simulate the fast stream in the open checkout-free groceries (e.g., 14s/ptq), ClusTree gets lower Precision and recall.

In addition, through Table 3, we can see that learning-based methods are much more efficient than traditional methods. They generally achieve better performance by projecting the features to the low-dimensional linear subspace. [38] achieves competitive overall performance due to the global-aware clustering, but the time consumption of the strategy is intolerable in this scene. IPS based GCN [17, 30] are more efficient comparing with the others. To take our *query* as pivot for IPS construction, the sub-graph represents the local correlation of nodes just like the proposed method, and gains a balanced time and performance improvement. However, they still measure the node distance by cosine similarity, which ignores the distribution nature of the nodes.

As shown in Table 3, the proposed method surpasses the other clustering methods by a large margin and achieves state-of-the-art performance on the datasets. It also outperforms CSG-linked GLCN by a 2.60% and 1.75% F1 score on DaiCOFG and IseCOFG, respectively, with comparable time consumption, indicating the vMF-based *weight* strategy works even better than the learning-based strategy. In practice, the proposed method is thus the most applicable algorithm for the complicated open check-out free grocery scenario.

6 Conclusion and future work

This paper proposed a real-time Person Clustering method, namely CSG-GCN, for Open checkout-free groceries under data streams and the unknown number of persons. The proposed method fully utilizes the human time-space information and makes a variance-considered comparison on the spherical summarized data to improve the method's speed and accuracy. And the experimental results show the effectiveness and advantages of the method. Future research will focus on applying the technology to more real-world open checkout-free groceries.

References

1. Ackermann, M.R., Märtens, M., Raupach, C., Swierkot, K., Lammersen, C., Sohler, C.: Streamkm++ a clustering algorithm for data streams. Journal of Experimental Algorithmics (JEA) **17**, 2–1 (2012)
2. Aggarwal, C.C.: Data streams: models and algorithms, vol. 31. Springer Science & Business Media (2007)
3. Almasawa, M.O., Elrefaei, L.A., Moria, K.: A survey on deep learning-based person re-identification systems. IEEE Access **7**, 175228–175247 (2019)
4. Banerjee, A., Dhillon, I.S., Ghosh, J., Sra, S., Ridgeway, G.: Clustering on the unit hypersphere using von mises-fisher distributions. Journal of Machine Learning Research **6**(9) (2005)
5. Bendale, A., Boult, T.: Towards open world recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1893–1902 (2015)
6. Bradley, P.S., Fayyad, U.M., Reina, C.A., Bradley, F.R., Bradley, P., Fayyad, U., Reina, C.: Scaling clustering algorithms to large databases, microsoft research report (1998)
7. Braun, M., Krebs, S., Flohr, F., Gavrila, D.M.: Eurocity persons: A novel benchmark for person detection in traffic scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(8), 1844–1861 (2019). https://doi.org/10.1109/TPAMI.2019.2897684
8. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)
9. Chiang, W.L., Liu, X., Si, S., Li, Y., Bengio, S., Hsieh, C.J.: Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 257–266 (2019)
10. Domingos, P., Hulten, G.: A general method for scaling up machine learning algorithms and its application to clustering. In: In proceedings of the eighteenth international conference on machine learning. Citeseer (2001)
11. Enrique, Amigó, Julio, Gonzalo, Javier, ArtilesFelisa, Verdejo: A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information Retrieval (2009)
12. Fan, X., Jiang, W., Luo, H., Fei, M.: Spherereid: Deep hypersphere manifold embedding for person re-identification. Journal of Visual Communication and Image Representation **60**, 51–58 (2019)
13. Fuglede, B., Topsoe, F.: Jensen-shannon divergence and hilbert space embedding. In: International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings. p. 31. IEEE (2004)
14. Gama, J.: Knowledge discovery from data streams. CRC Press (2010)
15. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp. 249–256. JMLR Workshop and Conference Proceedings (2010)
16. Ji, W., Yu, S., Wu, J., Ma, K., Bian, C., Bi, Q., Li, J., Liu, H., Cheng, L., Zheng, Y.: Learning calibrated medical image segmentation via multi-rater agreement modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12341–12351 (2021)

17. Jiang, B., Zhang, Z., Lin, D., Tang, J., Luo, B.: Semi-supervised learning with graph learning-convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
19. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
20. Kranen, P., Assent, I., Baldauf, C., Seidl, T.: The clustree: indexing micro-clusters for anytime stream mining. Knowledge and information systems **29**(2), 249–272 (2011)
21. Mohammadi, S.S., Wang, Y., Bue, A.D.: Pointview-gcn: 3d shape classification with multi-view point clouds. In: 2021 IEEE International Conference on Image Processing (ICIP). pp. 3103–3107 (2021). https://doi.org/10.1109/ICIP42928.2021.9506426
22. Ryu, S., Kwon, Y., Kim, W.Y.: A bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification. Chemical science **10**(36), 8438–8446 (2019)
23. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823 (2015)
24. Shen, J., Liu, Y., Dong, X., Lu, X., Khan, F.S., Hoi, S.C.: Distilled siamese networks for visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
25. Silva, J.A., Faria, E.R., Barros, R.C., Hruschka, E.R., Carvalho, A.C.d., Gama, J.: Data stream clustering: A survey. ACM Computing Surveys (CSUR) **46**(1), 1–31 (2013)
26. Sra, S.: A short note on parameter approximation for von mises-fisher distributions: and a fast implementation of i s (x). Computational Statistics **27**(1), 177–190 (2012)
27. Tang, H., Zhao, G., Bu, X., Qian, X.: Dynamic evolution of multi-graph based collaborative filtering for recommendation systems. Knowledge-Based Systems **228**, 107251 (2021)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
29. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
30. Wang, Z., Zheng, L., Li, Y., Wang, S.: Linkage based face clustering via graph convolution network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1117–1125 (2019)
31. Wu, J., Di, X.: Integrating neural networks into the blind deblurring framework to compete with the end-to-end learning-based methods. IEEE Transactions on Image Processing (2020)
32. Wu, J., Fang, H., Shang, F., Wang, Z., Yang, D., Zhou, W., Yang, Y., Xu, Y.: Learning self-calibrated optic disc and cup segmentation from multi-rater annotations. arXiv preprint arXiv:2206.05092 (2022)
33. Wu, J., Fang, H., Shang, F., Yang, D., Wang, Z., Gao, J., Yang, Y., Xu, Y.: Seatrans: Learning segmentation-assisted diagnosis model via transforme. arXiv preprint arXiv:2206.05763 (2022)
34. Wu, J., Fang, H., Wu, B., Yang, D., Yang, Y., Xu, Y.: Opinions vary? diagnosis first! arXiv preprint arXiv:2202.06505 (2022)

35. Wu, J., Fu, R.: Universal, transferable and targeted adversarial attacks. arXiv preprint arXiv:2109.07217 (2019)
36. Wu, J., Yu, S., Chen, W., Ma, K., Fu, R., Liu, H., Di, X., Zheng, Y.: Leveraging undiagnosed data for glaucoma classification with teacher-student learning. In: International Conference on Medical Image Computing and Computer-Assisted Intervention(MICCAI). pp. 731–740. Springer (2020)
37. Yang, L., Zhan, X., Chen, D., Yan, J., Loy, C.C., Lin, D.: Learning to cluster faces on an affinity graph. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2298–2306 (2019)
38. Yang, L., Zhan, X., Chen, D., Yan, J., Loy, C.C., Lin, D.: Learning to cluster faces on an affinity graph. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2298–2306 (2019)
39. Zeng, H., Zhou, H., Srivastava, A., Kannan, R., Prasanna, V.: Graphsaint: Graph sampling based inductive learning method. arXiv preprint arXiv:1907.04931 (2019)
40. Zhang, J., Shi, X., Xie, J., Ma, H., King, I., Yeung, D.Y.: Gaan: Gated attention networks for learning on large and spatiotemporal graphs. arXiv preprint arXiv:1803.07294 (2018)
41. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: an efficient data clustering method for very large databases. ACM sigmod record **25**(2), 103–114 (1996)
42. Zhang, Y., Di, X., Zhang, B., Ji, R., Wang, C.: Better than reference in low-light image enhancement: Conditional re-enhancement network. IEEE Transactions on Image Processing **31**, 759–772 (2022). https://doi.org/10.1109/TIP.2021.3135473