Occupation Clustering Using Deep Embedded Kmeans

Ngoc-Toan Le[1] and Thanh-Hieu Bui[1]*

[1]Department of Business Information Technology, University of Economics Ho Chi Minh City,
Ho Chi Minh City, Vietnam.
hieubt@ueh.edu.vn

Abstract. While applications of clustering in business commonly focus on business products or geographic concentrations, occupation clustering concentrates on the knowledge, skills, interests, and abilities of employees. Occupation clustering analysis provides more insights into the talent, knowledge, and skills of employees than the education level and qualifications of employees. To gain insights into occupation, this paper proposes an occupation clustering based on Deep Embedded KMeans clustering. Our occupation clustering works on three factors: interests, knowledge and skills of each occupation according to knowledge classification and measurement system. In the experimental result, we have twelve clusters of occupation with interests, knowledge, and skills. The result can help us to gain deeper insights into the skills, interests, and knowledge of employees.

Keywords: Occupation Clustering, Deep Embedded Clustering, Deep Embedded KMeans Clustering, KMeans.

1 Introduction

According to the definition of Occupational Information Network (O*NET), occupation clusters contain occupations in the same domain of jobs that require similar skills. While clustering applications in business commonly focuses on business products or geographic concentrations, occupation clustering focuses on the knowledge, skills, interests, and abilities of employees. Occupation clustering analysis provides more insights into the talent, knowledge, and skills of employees than the education level and qualifications of employees. Rapid changes in the job market have led to occupation requirements with a better mix of skills and knowledge. Occupation clustering can help human resources departments add necessary skills and knowledge to job requirements. In addition, corporations can provide internal training courses for employees to improve their knowledge and skills. Seeking a suitable career is extremely important for individuals, helping them develop their full potential, as well as facilitating the cultivation of necessary skills and professional knowledge. Occupation clustering based on interests makes it easier for parents, students, and educators to navigate career development. Employees can also proactively improve their skills and knowledge from occupation clustering. Occupation clustering provides insight into the talent for the economy which is a key

T. A. Trinh et al. (eds.), *Proceedings of the 2nd International Conference - Resilience by Technology and Design (RTD 2024)*, Advances in Intelligent Systems Research 186,
https://doi.org/10.2991/978-94-6463-583-6_2

tool in developing and implementing economic strategies. Besides, this analysis can help professionals build effective relationships between employers, employees, and educators. This paper proposes an occupation clustering using Deep Embedded KMeans clustering model. Our occupation clustering works on three factors interests, knowledge and skills of each occupation, according to knowledge classification and measurement system O*NET.

2 Literature Review

The study by Feser and associates (2009) expanded the concept of occupational structure by including the knowledge characteristics of individual occupations and developed knowledge based occupation clustering. Feser's study used Ward's aggregate hierarchical clustering method to construct occupation clusters. However, these studies limit occupations to high-tech and high-knowledge industries. The study by Nolan and associates (2011) divided 15 occupational clusters based on the knowledge level of each occupation according to the knowledge classification and measurement system O*NET. The study uses Ward's aggregate hierarchical clustering algorithm based on the study of Feser (2009). The study minimized cluster variation based on differences in measures of knowledge variables for each occupation. Outlier data points greatly influence the results of Ward clustering. Outlier data points have a very large distance from the standard data points and are distributed very differently from the standard data points. To overcome this defection, outlier occupations were removed from the clustering model. The study by Dai Debao and associates (2021) divided big data jobs into 10 different categories to explore the demand for big data jobs. The study applied natural language processing and K Means clustering model to process job names on the recruitment website Zhaopin.com. The study by Ternikov (2023) clustered IT jobs into 13 skill based job clusters. The job skill description will be applied natural language processing, then the hierarchical agglomerative clustering method. The recent studies by Dai Debao (2021) and Ternikov (2023) both apply natural language processing to process data about job information, then apply clustering models such as Kmeans or Hierarchical agglomerative clustering to cluster these jobs. However, current job requirements all label skill groups and knowledge groups, so it can be easier to apply clustering models for these classification variables without processing natural language. These studies are only applying for a specific occupation group, and not for a variety of occupation groups. If the study applied to many occupation groups, outlier observations must be removed from the dataset such as the study of Nolan (2011). So the occupation clustering results are not generalized. In addition, the previous clustering models were only based on one factor such as knowledge, skills, or occupation names. Clustering with many factors such as knowledge, skills, and interests will provide the general view of occupation groups. Besides the knowledge and skills applied in previous studies, this study proposes interests as an important factor in occupation clustering. Career development often focuses more on skills and knowledge and forgets about interests. Interests and hobbies can help employees identify the skills and strengths that they can develop. Clustering models applied in previous studies such as KMeans or Ward's aggregate hierarchical clustering have non-linearity in real-world data, making these models unable to perform well. Deep neural network models have stronger nonlinear representation ability to extract features, which will get better clustering performance. Based on that, this study proposes the Deep Embedded KMeans Clustering model to cluster occupation dataset.

3 Methodology

3.1 Occupation Data Collection

Occupation dataset is collected from Occupational Information Network (O*NET). The dataset includes 774 occupations with a set of interests, knowledge, and skills. There are 6 groups of interests, 10 groups of knowledge, and 7 groups of skills. Our study applies Deep Embedded KMeans Clustering Model for 20,571 occupation sets with interests, knowledge, and skills. Three categorial variables including interests, knowledge, and skills are preprocessed by using onehotencoder from scikit-learn library. After applying onehotencoder, the dataset with 23 columns is decomposed by principal component analysis with 3 components.

3.2 Occupation Clustering Based on Deep Embedded KMeans

KMeans algorithm. The KMeans algorithm is a type of clustering algorithm published by J. B. MacQueen. This unsupervised algorithm is commonly used in data mining and pattern recognition. The target of minimizing the performance index, method error criteria, and bias is the basis of this algorithm. To find the optimization result, this algorithm tried to find K that allows the division to satisfy a certain criterion. First, choose several points to represent the initial cluster focus (the information will select the first K income samples to represent the initial cluster focus). Next, gather the remaining points to their focal points according to the minimum distance criterion, then take the initial classification, and if the classification is not reasonable, modify it (recalculate each cluster score criteria), iteratively until there is a reasonable classification. The KMeans algorithm includes the following steps:

Step 1: Calculate the Euclidean distance. Suppose X and Z are two samples of the sample vector, $X = (x_1, x_2, \dots, x_n)^T$, $Z = (z_1, z_2, \dots, z_n)^T$, and the distance between X and Z is specified by Equation (1):

$$D = \|X - Z\| = [\sum_{i=1}^{n}(x_i - z_i)^2]^{\frac{1}{2}} \tag{1}$$

The smaller D is, the more similar X and Z are (D is the distance of X and Z in n-dimensional space)

Step 2: Determine the cluster criteria function. The sample set is $\{X\} = \{X_1, X_2, \dots, X_N\}$, and classified into classes C, $S_1, X_2, \dots, X_N$. $M_j$và S_j are the average vectors. So we have Equation (2):

$$\mathrm{M_j} = \frac{1}{\mathrm{N_j}}\sum\nolimits_{\mathrm{X}\in \mathrm{S_j}} \mathrm{X},\ \ \mathrm{N_j} = \left|\mathrm{S_j}\right| \tag{2}$$

N_j and S_j is the number of samples. Then we define the cluster criterion function as in Equation (3):

$$\mathrm{J} = \sum\nolimits_{\mathrm{j}=1}^{\mathrm{C}} \sum\nolimits_{\mathrm{X}\in \mathrm{S_j}} \left\|\mathrm{X} - \mathrm{M_j}\right\|^2 \tag{3}$$

J represents the quadratic sum of the inaccuracies of all sample types and their average value. We can also call the sum of the distances of the samples and their average value. So, we should try our best to get the minimum value. The algorithm will repeat the above steps until an acceptable result is achieved.

Silhouette Coefficient. Silhouette is a method of interpretation and validation of consistency within data clusters. It was proposed by Peter Rousseeuw in 1987. The Silhouette Coefficient is calculated by using (a) and the mean nearest-cluster distance (b) for each sample. The Silhouette Coefficient *Sc* for a sample data is specified as:

$$\mathrm{Sc} = \frac{\mathrm{b} - \mathrm{a}}{\max(\mathrm{a,b})} \tag{4}$$

where *a* is the mean intra-cluster distance, *b* is the mean nearest-cluster distance for each sample, *b* is the distance between a sample and the nearest cluster that the sample is not a part of.

The Silhouette Coefficient has a value from -1 to 1. The best Silhouette Coefficient is 1 and the worst is -1. Values near 0 indicate overlapping clusters. Negative values generally indicate that a sample has been assigned to the wrong cluster, as a different cluster is more similar.

Deep Embedded Clustering. We consider a set of n points $\{x_i \in X\}_{i=1}^{n}$ with k clusters, each cluster is represented by the centroid μ_j with j = 1, . . ., k. Instead of direct clustering in the data space X, the proposed method transforms the data using a nonlinear mapping f_θ: X → Z, where θ are the learnable parameters and Z is the latent feature space. The dimensionality of Z is usually much smaller than that of X to avoid combinatorial explosion in storage and computation in the dimensionality of the representation (Bellman, 1961). To parameterize f_θ, deep neural network is an obvious choice due to their properties of approximating the theoretical function (Hornik, 1991) and their demonstrated feature learning capabilities (Bengio et al., 2013). Deep Embedded Clustering (DEC) model clusters data by simultaneously learning a set of k cluster centers $\{\mu_j \in Z\}_{j=1}^{k}$ in the feature space Z and the parameters θ of the deep neural network, mapping data points to Z. DEC has two stages: stage one is parameter initialization with a deep autoencoder (Vincent et al., 2010) and stage two is parameter optimization (i.e. is clustering), where the method iterates between computing the auxiliary target distribution and minimizing the Kullback–Leibler (KL) divergence with respect to it. The method begins by describing stage 2 parameter optimization/clustering, with initial estimates of θ và $\{\mu_j\}_{j=1}^{k}$.

Clustering with the Kullback–Leibler (KL) divergence. To initially estimate the non-linear mapping f_θ and the initial cluster centroid $\{\mu_j\}_{j=1}^{k}$, the proposed method improves clustering using an unsupervised algorithm that alternates between two steps. In the first step, the method calculates a soft label assignment between the embedded points and the cluster centroids. In the second step, the method updates the deep mapping f_θ and refines the cluster centroids by learning from existing high-confidence assignments using the auxiliary target distribution. This process is repeated until a convergence criterion is met. According to van der Maaten and Hinton (2008), the method using the t-distribution as a kernel to measure the similarity between the embedding point z_i and the centroid μ_j is specified as in Equation (5):

$$q_{ij} = \frac{(1+\|z_i-\mu_j\|^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'}(1+\|z_i-\mu_{j'}\|^2/\alpha)^{-\frac{\alpha+1}{2}}} \tag{5}$$

where $z_i = f_\theta(x_i) \in Z$ corresponds to $x_i \in X$ after embedded, α is the degrees of freedom of the student distribution and q_{ij} can be understood as the probability of assigning sample i to cluster j (i.e. soft label assignment). Because cross-validation of α on the validation set is not possible in unsupervised settings and learning it is not necessary (van der Maaten, 2009), $\alpha = 1$ should be set for all experiments. The method iteratively fine-tunes the clusters by learning from their high-fidelity assignments with the help of an auxiliary target distribution. Specifically, the model is trained by fitting the soft labeling task to the target distribution. Finally, we define our target as the KL divergence loss between the soft label assignments q_i and the auxiliary distribution p_i as in Equation (6) as follows:

$$L = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{6}$$

The selection of the P distribution target is critical to the performance of DEC. A naive approach would set each p_i to a delta distribution (to the nearest centroid) for data points above the confidence threshold and ignore the rest. However, because q_i are soft labeling tasks, it is more natural and flexible to use softer probabilistic targets. Specifically, the target distribution will have the following properties: enhanced prediction (i.e., improved cluster purity), greater emphasis on data points assigned with high confidence, and normalization loss distribution of each center point to prevent large clusters from distorting the hidden feature space. In the experiments, p_i is calculated by raising q_i to the second power and then normalizing by frequency per cluster by Equation (7):

$$p_{ij} = \frac{q_{ij}^2/f_j}{\sum_{j'} q_{ij'}^2/f_{j'}} \tag{7}$$

where $f_j = \sum_i q_{ij}$ is the frequency of the soft clustering.

Training can be considered a form of self-training (Nigam and Ghani, 2000). As in self-training, the model takes an initial classifier and an unlabeled dataset, then labels the dataset with the classifier to train on confident predictions. its own height. Indeed, in experiments, the DEC model improved the initial estimate in each iteration by learning from high-confidence predictions, which in turn helped improve low-confidence predictions. The model optimized the cluster center $\{\mu_j\}$ and deep neural network parameters θ using stochastic gradient descent with momentum. The slope of L for the feature space embedding of each data point z_i and each cluster centroid μ_j is calculated in Equation (8) and (9) as follows:

$$\frac{\partial L}{\partial z_i} = \frac{\alpha+1}{\alpha} \sum_j \left(1 + \frac{\|z_i - \mu_j\|^2}{\alpha}\right)^{-1} (p_{ij} - q_{ij})(z_i - \mu_j) \tag{8}$$

$$\frac{\partial L}{\partial \mu_j} = -\frac{\alpha+1}{\alpha} \sum_i \left(1 + \frac{\|z_i - \mu_j\|^2}{\alpha}\right)^{-1} (p_{ij} - q_{ij})(z_i - \mu_j) \tag{9}$$

The gradients ∂L/∂zi are then passed down to the deep neural network and used in standard backpropagation to calculate the deep neural network parameter gradient ∂L/∂θ. For the purpose of exploring clustering tasks, the method stops when less than tol% of cluster distribution change points between two consecutive iterations.

Parameter Initialization. The DEC Model is initialized with a Sparse Autoencoder (SAE) because recent research has shown that they consistently produce semantically meaningful and well-separated representations across datasets in the real world (Vincent et al., 2010; Hinton and Salakhutdinov, 2006; Le, 2013). Therefore, the unsupervised representation learned by SAE facilitates the learning of clustering representations with DEC. The SAE network layer is initialized layer by layer, with each layer being a denoising autoencoder trained to reconstruct the output of the previous layer after random corruption (Vincent et al., 2010). The denoising autoencoder is a two-layer neural network defined as in Equation (10), (11), (12), and (13):

$$\tilde{x} \sim \mathrm{Dropout}(x) \tag{10}$$

$$h = g_1(W_1\tilde{x} + b_1) \tag{11}$$

$$\tilde{h} \sim \mathrm{Dropout}(h) \tag{12}$$

$$y = g_2(W_2\tilde{h} + b_2) \tag{13}$$

where the Dropout function is a mapping random variable that randomly sets a portion of its input size to 0, g_1 and g_2 are activation functions for the encoding and decoding layer respectively, and $\theta = \{W_1, b_1, W_2, b_2\}$ are model parameters. Training is done by minimizing the least squares loss function $\|x - y\|_2^2$.

After training a layer, its output h is used as input to train the next layer. Rectified Linear Units (ReLUs) (Nair and Hinton, 2010) are used in all encoder/decoder pairs, except g_2 of the first pair (it needs rebuilding input data can have positive and negative values, such as mean zero) and g1 of the last pair, so the final embedded data retains full information (Vincent et al., 2010). After greedy layer training, all the encoder layers are concatenated, followed by all the decoder layers, reversing the layerwise training order, to form a deep learning autoencoder, and then then fine-tune it to minimize loss. The result is a bottleneck multi-layer deep learning autoencoder with an encoding layer in the middle. Then, we remove the decoder layers and use the encoder layers as the initial mapping between the data space and the feature space. To initialize the cluster centers, we pass the data through the initialized DNN to get the embedded data points and then perform standard Kmeans clustering in the feature space Z to obtain k initial centroids $\{\mu_j\}_{j=1}^{k}$.

Deep Embedded KMeans Clustering. The DEC model learns from feature representations and cluster assignments using neural networks for deep learning. To achieve this, the DEC model is combined with the KMeans algorithm for data clustering. The model combines encoded input data and output data consisting of encoded output data with soft labels applied and decoded output data. Then, the DEC model is trained with an iterative process that refines the clusters by learning from high-confidence assignments with the help of an auxiliary target distribution function. Specifically, the DEC model is trained by fitting the soft distribution to the target distribution. Finally, the objective function or loss function is defined as the Kullback-Leibler divergence loss between the soft label assignments and the auxiliary distribution. After training, the weights of the classes will be saved and uploaded for use when applying the model to cluster data.

Evaluation Metric. The unsupervised clustering accuracy (ACC) is the standard metric to evaluate the clustering methods. The ACC values are from 0 to 1. The higher the values, the better the clustering results. ACC measures the proportion of samples whose cluster assignments can be correctly mapped to the ground-truth labels. ACC is defined as follows:

$$\mathrm{ACC} = \max_{\mathrm{m}} \frac{\sum_{\mathrm{i}=1}^{\mathrm{n}} 1\{\mathrm{g_i} = \mathrm{m}(\mathrm{c_i})\}}{\mathrm{n}} \tag{14}$$

where g_i is the ground-truth label of the *i*-th data point, $\mathrm{c_i}$ is the cluster assignment of the *i*-th data point, *m* ranges over all possible one-to-one mappings between ground-truth labels and cluster assignments. The mapping is based on the Hungarian algorithm.

4 Results and Discussions

To train the KMeans algorithm, we need to estimate the number of clusters. From the results of the Silhouette Coefficient (see Fig 1), it shows that the optimal number of clusters for the data is 18 clusters, with Silhouette Score = 0.7785. Based on the optimal number of clusters, the dataset is trained by using the KMeans algorithm to cluster the data.

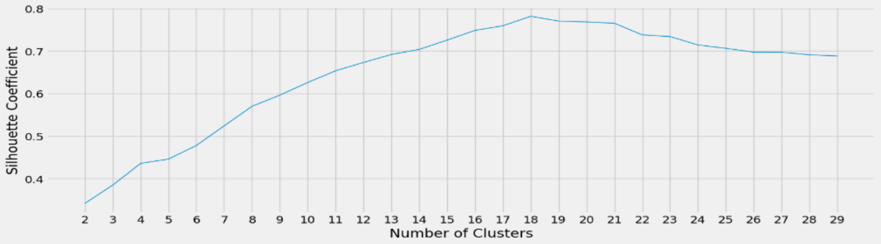

Fig 1. The Silhouette Coefficient.

Deep Embedded KMeans clustering method reduces the number of clusters to 12 clusters. The clustering results show that Deep Embedded KMeans clustering method has the higher accuracy score (0.280784) than KMeans clustering method (0.231692). Therefore, Deep Embedded KMeans Clustering Model is chosen for occupation clustering. Based on the result of Deep Embedded KMeans Clustering Model, we have 12 clusters of occupation with interests, knowledge, and skills in Table 1.

Table 1. Occupation Clusters.

Cluster	No. Occupation	Interests	Knowledge	Kills
1	48	Artistic	Engineering and Technology	Content
2	1639	Conventional	All of knowledge groups	Content
3	730	Conventional	Science, Law, and Public Safety	Technical and Resource Management Skills
4	349	Realistic	Arts and Humanities	Content
5	348	Realistic	Arts and Humanities	Process
6	1366	Realistic	Science, Law, and Public Safety	Content
7	1453	Realistic	Science, Law, and Public Safety	Process
8	2452	Realistic	All of knowledge groups	Technical, System and Soft Skills
9	3167	Investigative, Enterprising, Social and Artistic	All of knowledge groups	Content
10	449	Investigative, Enterprising, Social and Artistic	Business and Engineering	Technical, System and Resource Management Skills
11	5939	Investigative, Enterprising, Social, Artistic and Conventional	All of knowledge groups	Process and Social Skills
12	2632	Investigative, Enterprising, Social, Artistic and Conventional	All of knowledge groups	Technical, System and Soft Skills

According to the definition of Occupational Information Network (
occupation clusters contain occupations in the same domain of jobs tha
similar skills. While clustering applications in business commonly fo
business products or geographic concentrations, occupation clustering focus
knowledge, skills, interests, and abilities of employees. Occupation clustering
provides more insights into the talent, knowledge, and skills of employees
education level and qualifications of employees. Rapid changes in the jo
have led to occupation requirements with a better mix of skills and kn
Occupation clustering can help human resources departments add necessary s
knowledge to job requirements. In addition, corporations can provide interna
courses for employees to improve their knowledge and skills. Seeking a
career is extremely important for individuals, helping them develop t
potential, as well as facilitating the cultivation of necessary skills and pro
knowledge. Occupation clustering based on interests makes it easier for
students, and educators to navigate career development. Employees
proactively improve their skills and knowledge from occupation c
Occupation clustering provides insight into the talent for the economy which

We can gain some insights from these occupation clusters. For example, occupations in cluster number one will be suitable for the person with artistic talent. They love knowledge about Engineering and Technology and want to gain content skills. Occupations in cluster number five will be suitable for the realistic person who interest knowledge about Arts and Humanities and want to gain process skills. Occupations in cluster number ten will be suitable for people with Investigative, Enterprising, Social and Artistic Interests who love knowledge of Business and Engineering and want to gain Technical, System and Resource Management Skills.

5 Conclusion

This paper presented an occupation clustering based on Deep Embedded KMeans clustering. Our occupation clustering works on three factors interests, knowledge and skills of each occupation. The proposed approach with many factors such as knowledge, skills, and interests can provide the general view of occupation groups. Career development often focuses more on skills and knowledge and forgets about interests. Interests and hobbies can help employees identify the skills and strengths that they can develop. The result of our proposed method shows 12 clusters of occupation with interests, knowledge, and skills. Based on 12 occupation clusters, parents, students, and educators can navigate career development. The proposed occupation clustering provides more insights into the talent for the economy which is a key tool in developing and implementing economic strategies. Besides, this analysis can help professionals build effective relationships between employers, employees, and educators. In the future work, our study will supplement Deep Embedded KMeans Clustering with more factors such as abilities, work activities, work context, work styles and work values.

References

1. Bellman, R. E.: Dynamic Programming. Princeton University Press (1961)
2. Bengio, Y., Courville, A., Vincent, P.: Unsupervised feature learning and deep learning. In: Proceedings of the 27th International Conference on Machine Learning (ICML), pp. 1–8 (2013)
3. Dai, D., Ma, Y., Zhao, M.: Analysis of big data job requirements based on K-means text clustering in China. PLOS ONE 16(8), 1–14 (2021)
4. Feser, E., Renski, H., Koo, J.: Regional cluster analysis with interindustry benchmarks. In: Targeting Regional Economic Development, pp. 213–238. Routledge (2009)
5. Guo, W., Lin, K., Ye, W.: Deep embedded K-means clustering. In: 2021 International Conference on Data Mining Workshops (ICDMW) (2021)
6. Hinton, G. E., & Salakhutdinov, R. R.: Reducing the dimensionality of data with neural networks. Science 313(5786), 504–507 (2006)

7. Hornik, K.: Approximation capabilities of multilayer feedforward networks. Mathematics of Operations Research 16(2), 252–266 (1991)
8. Le, Q. V.: Building high-level features using large scale unsupervised learning. In: Proceedings of the 30th International Conference on Machine Learning (ICML), pp. 256–264 (2013)
9. MacQueen, J. B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, pp. 281–297. University of California Press (1967)
10. Nair, V., & Hinton, G. E.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML), pp. 807–814 (2010)
11. Nigam, K., & Ghani, R.: Analyzing the effect of co-training on text classification. In: Proceedings of the 17th International Conference on Machine Learning (ICML), pp. 334–342 (2000)
12. Nolan, C., Morrison, E., Kumar, I., Galloway, H., Cordes, S.: Linking industry and occupation clusters in regional economic development. Economic Development Quarterly 25(1), 26–35 (2011)
13. Rousseeuw, P. J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20, 53–65 (1987)
14. Rousseeuw, P. J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20, 53–65 (1987)
15. Ternikov, A. A.: Artificial intelligence and the demand for skills in Russia. Voprosy Ekonomiki 11, 65–80 (2023)
16. van der Maaten, L. J. P., & Hinton, G. E.: Visualizing data using t-SNE. Journal of Machine Learning Research 9, 2579–2605 (2008)
17. van der Maaten, L. J. P.: Learning a parametric embedding by preserving local structure. In: Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 384–391 (2009)
18. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning (ICML), pp. 1096–1103 (2010)