Deep Clustering for Domain Adaptation

Citation for published version:
Gao, B, Yang, Y, Gouk, H & Hospedales, TM 2020, Deep Clustering for Domain Adaptation. in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* Institute of Electrical and Electronics Engineers (IEEE), pp. 4247-4251, 2020 IEEE International Conference on Acoustics, Speech, and Signal Processing, Barcelona, Spain, 4/05/20. https://doi.org/10.1109/ICASSP40776.2020.9053622

Digital Object Identifier (DOI):
10.1109/ICASSP40776.2020.9053622

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

DEEP CLUSTERING FOR DOMAIN ADAPTATION

Boyan Gao[1], *Yongxin Yang*[1], *Henry Gouk*[1], *Timothy M. Hospedales*[1,2]

[1]School of Informatics, University of Edinburgh, United Kingdom
[2]Samsung AI Centre, Cambridge, United Kingdom

ABSTRACT

We address the heterogeneous domain adaptation task: adapting a classifier trained on data from one domain to operate on another domain that also has a different label space. We consider two settings that both exhibit label scarcity of some form—one where only unlabelled data is available, and another where a small volume of labelled data is available in addition to the unlabelled data. Our method is based on two specialisations of a recently proposed approach for deep clustering. It is shown that our approach noticeably outperforms other methods based on deep clustering in both the fully unsupervised and the semi-supervised settings.

Index Terms— Domain Adaptation, Deep Clustering, Unsupervised Learning, Semi-Supervised Learning

1. INTRODUCTION

Domain adaptation aims to alleviate the need for labelled examples in a given target domain using knowledge transferred from a related source domain, such as across image data of different camera types [1]. Domain adaptation (DA) is a very well studied topic with numerous competing methods [1]. Clustering-based DA approaches group unlabelled target domain data examples that are likely to belong to the same class. One area of DA where clustering methods are particularly useful is that of *heterogeneous* domain adaptation—where the target domain contains novel classes compared to the source domain. For DA problems of this type, clustering-based DA methods [2, 3] provide one of the only solutions.

These existing clustering-based approaches to heterogeneous domain adaptation suffer from a significant conceptual deficiency. Several of these techniques rely on a soft approximation to the k-means objective function. This type of clustering objective is known to lead to poorer separation between clusters in the shallow setting [4], and is likely to impose a suboptimal inductive bias on the deep network being trained to find a good representation for classification. In this paper, we demonstrate the application of our recently proposed Concrete k-Means (CKM) deep clustering framework [5] to heterogeneous domain adaptation, where it achieves excellent results in both the completely unsupervised and sparsely supervised target domain conditions.

2. BACKGROUND & RELATED WORK

The CKM method for deep clustering consists of two steps: (i) An autoencoder parameterised by $\vec{\phi}$ and $\vec{\varphi}$ for the encoder and decoder, respectively, is trained using the standard reconstruction error loss on the unlabelled data, $X = \{\vec{x}_i\}_{i=1}^N$,

$$\mathcal{L}^{AE}(X, \vec{\phi}, \vec{\varphi}) = \sum_{i=1}^{N} \|\vec{x}_i - g_{\vec{\varphi}}(f_{\vec{\phi}}(\vec{x}_i))\|_2^2. \qquad (1)$$

(ii) The autoencoder is fine-tuned with the concrete k-means loss function applied to the hidden representation,

$$\mathcal{L}^{CKM}(X, M, \vec{\phi}) = \sum_{i=1}^{N} ||f_{\vec{\phi}}(\vec{x}_i) - \vec{h}_i M||_2^2, , \qquad (2)$$

where M is a matrix of k centroids (each individually represented as $\vec{\mu}_i$), and $\vec{h}_i$ is a one-hot vector sampled from a Gumbel-Softmax distribution [6, 7] parameterised by the distance of $f(\vec{x}_i)$ to each of the centroids in M.

Several other approaches to deep clustering follow the same high level approach—leveraging a clustering-based objective to impose additional inductive bias on the representation learned by an autoencoder. Xie et al. [8] show how to jointly optimise an autoencoder and a k-means model to get a "k-means friendly" latent space. The hard assignment in the k-means objective prevent them from optimising the true loss function, so their DEC method makes use of an approximation based on soft assignment of instances to clusters. However, this surrogate objective means that the solution to their model is not necessarily a minimum of the k-means objective. In contrast, DCN [9] resolves the issue by alternating optimisation. Each minibatch of training data is first used to update the deep representation while keeping the centroids held constant, and then used to update the centroids while holding the representation constant. However, alternating optimisation may be slow and ineffective compared to an end-to-end solution. More importantly it hampers integration

This work was supported by EPSRC grants EP/S000631/1 and EP/R026173/1; and the MOD University Defence Research Collaboration (UDRC) in Signal Processing.

of clustering as a module in a larger end-to-end deep learning system. In contrast to these methods, we show how one can jointly train a deep representation and cluster centroids with the standard k-means objective using backpropagation and conventional deep learning optimisers.

Domain adaptation is the process of learning a model on some source distribution in such a way that it will still perform well when applied to a different, but related, target distribution [1]. In the case of unsupervised domain adaptation (UDA), one has labelled data from the source domain and unlabelled data from the target domain, while supervised domain adaptation (SDA) assumes sparse labelled data and plentiful unlabelled data in the target domain. Only a few recent papers addressed the unsupervised [3, 2] and (semi)-supervised [10] heterogeneous DA problem. Conventional deep UDA methods like domain adversarial neural networks [11] are unsuited to this setting since naively making datasets indistinguishable is counterproductive if their categories are disjoint. An intuitive strategy for heterogeneous UDA is based on deep clustering. Specifically, learning the representation jointly across source and target domains provides knowledge transfer; while modelling separate source and target cluster centres supports disjoint categories. We show that the excellent deep clustering provided by our CKM method can underpin an effective heterogeneous DA algorithm that outperforms the few existing methods for this task.

Most domain adaptation studies assume shared label-space between source and target domain, or *homogeneous* domain adaptation. However, a particularly challenging variant of this problem is the setting where the source and target domains have differing or disjoint label-spaces, i.e., *heterogeneous* domain adaptation. It is the heterogeneous variant of the problem that we consider in this paper.

3. HETEROGENEOUS DOMAIN ADAPTATION

This section describes how the CKM framework [5] can be modified to perform heterogeneous domain adaptation. Two problem variants are considered: unsupervised domain adaptation, where one only has unlabelled data from the target domain; and semi-supervised domain adaptation, where one has access to the unlabelled data and a small set of labelled data.

3.1. Unsupervised Adaptation

In unsupervised domain adaptation, we assume a labelled source domain and unlabelled target domain. We denote by $X_s = \{\vec{x}_i^s\}_{i=1}^n$ and $Y_s = \{\vec{y}_i^s\}_{i=1}^n$, $\vec{y}^s \in \mathcal{Y}^s$ the features and labels from the source domain training set. For the target domain training, we have only features, $X_t^{tr} = \{\vec{x}_i^t\}_{i=1}^m$. After training, the resulting model is evaluated on a set of target domain data with labels, $X_t^{te} = \{\vec{x}_i^t\}_{i=1}^q$ and $Y_t^{te} = \{\vec{y}_i^t\}_{i=1}^q$, $\vec{y}^t \in \mathcal{Y}^t$. The particularly challenging aspect of the heterogeneous UDA task is that the source and target labels spaces are disjoint: $\mathcal{Y}^s \cap \mathcal{Y}^t = \emptyset$.

Algorithm 1: Concrete k-means for Heterogeneous Unsupervised Domain Adaptation

```
Input: $X_s, Y_s, X_t, \alpha, \eta, \lambda$
Onput: $f_{\vec{\phi}}, g_{\vec{\varphi}}, q_{\vec{\theta}}, M$
Init: $\vec{\phi}, \vec{\varphi}$,
while not converged do
    $(\vec{\phi}, \vec{\varphi}) \leftarrow (\vec{\phi}, \vec{\varphi}) - \alpha \nabla_{(\vec{\phi}, \vec{\varphi})} \mathcal{L}^{AE}(\{X_s, X_t\})$;
end
Init: $\vec{\theta}, M$ with k-means $++$
while not converged do
    $\vec{\phi} \leftarrow \vec{\phi} - \eta \nabla_{\phi}(\mathcal{L}^{AE}(\{X_s, X_t\}) + \lambda_1 \mathcal{L}^{CKM}(X_s))$;
    $\vec{\varphi} \leftarrow \vec{\varphi} - \eta \nabla_{\vec{\varphi}} \mathcal{L}^{AE}(\{X_s, X_t\})$;
    $M \leftarrow M - \eta \nabla_M \lambda_1 \mathcal{L}^{CKM}(X_t)$;
    $\vec{\theta} \leftarrow \vec{\theta} - \eta \nabla_{\vec{\theta}} \lambda_2 \mathcal{L}^{CE}(X_s, Y_s)$;
end
```

The CKM method, briefly outlined in Section 2, can be modified to perform heterogeneous UDA by taking advantage of the labelled data in the source domain. This is done by constructing a classifier, $q_{\vec{\theta}} : \mathcal{Z} \to \mathcal{Y}^s$, that takes embeddings generated by the autoencoder and classifies them into one of the source categories. This classifier can be jointly trained with the encoder using the cross entropy loss function,

$$\mathcal{L}^{CE}(X, Y, \vec{\phi}, \vec{\theta}) = -\sum_{i=1}^{|X|} \sum_{j=1}^{|\mathcal{Y}^s|} y_{i,j} \log(q_{\vec{\theta}}(f_{\vec{\phi}}(\vec{x}_i))), \qquad (3)$$

to encourage the features learned by the autoencoder to be more discriminative. This results in a final optimisation objective of

$$\min_{M, \vec{\theta}, \vec{\phi}, \vec{\varphi}} \quad \mathcal{L}^{AE}(X_s \cup X_t, \vec{\phi}, \vec{\varphi}) + \lambda_1 \mathcal{L}^{CKM}(X_t, M, \vec{\phi}) \\ + \lambda_2 \mathcal{L}^{CE}(X_s, Y_s, \vec{\phi}, \vec{\theta}) \qquad (4)$$

which can be optimised using Algorithm 1. The intuition here is that the data ($\mathcal{L}^{AE}$) and label information ($\mathcal{L}^{CE}$) in the source domain improves the representation defined by $\vec{\phi}$, which in turn benefits unsupervised grouping in the target domain.

To evaluate heterogeneous UDA at testing time, we compute the match between the learned cluster identities and the true target sample labels.

3.2. Semi-Supervised Adaptation

Section 3.1 discusses how one can use the CKM framework to address the task fo heterogeneous UDA, where no labelled data is available for the target domain. We next show how to solve the related problem of heterogeneous semi-supervised domain adaptation, where one additionally has a small labelled dataset of training examples from the target domain [10]. The features and labels in the auxiliary target domain training set are denoted by X_t^l and Y_t^l, respectively. To take advantage of this information, we add a classifier $r_M : \mathcal{Z} \rightarrow \mathcal{Y}^t$, trained on the labelled target domain data to the model in Section 3.1. The probability of an instance belonging to class j is computed by

$$r_M^{(j)}(\vec{z}) = \frac{\exp\{-\|\vec{\mu}_j - \vec{z}\|_2^2\}}{\sum_{c=1}^{k} \exp\{-\|\vec{\mu}_c - \vec{z}\|_2^2\}}. \tag{5}$$

Crucially, this classifier is parameterised in terms of the same centroids M used for clustering. This constrains each cluster to have a one-to-one association with classes and provides a clean form of semi-supervised learning, as the cluster centroids now receive a supervisory signal from both target labels and the unsupervised clustering loss. To initialise the centroids, the embeddings of all the instances in the small labelled target domain dataset are computed. The mean embedding associated with each class is then used as the starting value for the corresponding cluster centroid,

$$\vec{\mu}_j = \frac{k}{|X_t^l|} \sum_{i=1}^{|X_t^l|} f_{\vec{\phi}}(\vec{x}_i)\mathbb{I}(y_{i,j} = 1), \tag{6}$$

where $\mathbb{I}(\cdot)$ is the indicator function that evaluates to one when its parameter is true, and zero otherwise. We define the k-shot loss as the cross entropy loss given predictions r_M,

$$\mathcal{L}^{k\text{-shot}}(X, Y, \vec{\phi}, M) = -\sum_{i=1}^{|X|} \sum_{j=1}^{|\mathcal{Y}^t|} y_{i,j} \log(r_M(f_{\vec{\phi}}(\vec{x}_i))). \tag{7}$$

The objective function for this semi-supervised variant of heterogeneous domain adaptation with CKM is

$$\begin{aligned} \min_{M,\vec{\theta},\vec{\phi},\vec{\varphi}} \quad & \mathcal{L}^{AE}(X_s \cup X_t, \vec{\phi}, \vec{\varphi}) + \lambda_1 \mathcal{L}^{CKM}(X_t, M, \vec{\phi}) \\ & + \lambda_2 \mathcal{L}^{CE}(X_s, Y_s, \vec{\phi}, \vec{\theta}) + \lambda_3 \mathcal{L}^{k\text{-shot}}(X_t^l, Y_t^l, \vec{\phi}, M), \end{aligned} \tag{8}$$

and can be minimised through the use of Algorithm 2.

In summary we find both a feature extractor and a set of centroids where the unlabelled data groups nicely around the centroids, and also the labelled data can be predicted by interpreting the centroids as a set of RBF classifier means. The intuition here is that knowledge is transferred from the source domain through its participation in supervised (cross-entropy) and unsupervised (autoencoder) representation learning. Meanwhile, both the labelled and unlabelled target data are both effectively utilised through learning the set of embedding prototypes M that both group the unlabelled data and classify the labelled data. This deep semi-supervised learning in the target domain can be considered as having a similar intuition to clustering-based [12] and entropy-minimisation [13] strategies widely used in shallow SSL.

Algorithm 2: Concrete k-means for Heterogeneous Semi-Supervised Domain Adaptation

```
Input: $X_s, Y_s, X_t, X_t^l, Y_t^l, \alpha, \eta$
Ontput: $f_{\vec{\phi}}, g_{\vec{\varphi}}, q_{\vec{\theta}}, M$
Init: $\vec{\phi}, \vec{\varphi}$
while not converged do
    $(\vec{\phi}, \vec{\varphi}) \leftarrow (\vec{\phi}, \vec{\varphi}) - \alpha \nabla_{(\vec{\phi}, \vec{\varphi})} \mathcal{L}^{AE}(\{X_s, X_t\})$;
end
Init: $\vec{\theta}, M$ with $\{X_t^l, Y_t^l\}$
while not converged do
    $\vec{\phi} \leftarrow \vec{\phi} - \eta \nabla_{\vec{\phi}} (\mathcal{L}^{AE}(\{X_s, X_t\}) + \lambda_1 \mathcal{L}^{CKM}(X_t))$;
    $\vec{\varphi} \leftarrow \vec{\varphi} - \eta \nabla_{\vec{\varphi}} \mathcal{L}^{AE}(\{X_s, X_t\})$;
    $M \leftarrow$
      $M - \eta \nabla_M (\lambda_1 \mathcal{L}^{CKM}(\vec{X}_t) + \lambda_3 \mathcal{L}^{k\text{-shot}}(X_t^l, Y_t^l))$;
    $\vec{\theta} \leftarrow \vec{\theta} - \eta \nabla_{\vec{\theta}} \lambda_2 \mathcal{L}^{CE}(X_s, Y_s)$;
end
```

4. EXPERIMENTS

In this section, we experimentally evaluate the SSDA and UDA methods described in Section 3.

4.1. Heterogeneous Unsupervised Domain Adaptation

We consider SVHN (0–4)→ MNIST (5–9) [10] as a domain adaptation benchmark with disjoint source and target labels. The source domain is Street View House Numbers (SVHN), which is a real-world digit dataset collected from Google street view. It contains 73,257 colored digits for training, and 26,032 for testing. To set up the source domain, only the images with digits 0 to 4 are selected. The target domain dataset is built from the MNIST dataset using only digits 5 to 9 are selected. Target image labels are only used for evaluation at testing. In heterogeneous UDA, we pre-train a convolutional neural network auto-encoder. To enable fair comparison with other state-of-the-art and pave the way for experiments in the Heterogeneous Semi-supervised UDA setting in Section 4.2, our encoder's architecture is same as [10]. The (pre)training of the auto-encoder varies across the baselines as detailed next.

We compare our **CKM-UDA** to the following baselines and competitors:

1. **Target Clustering.** Cluster the target domain data only. The AE is pre-trained only on the target domain, and then the CKM and AE objectives are fine-tuned together on target domain.

Method	Accuracy
Target Clustering	82.1±2.1%
Src + Targ Clustering	86.7±1.8%
DEC-UDA [8]	91.6±3.8%
DCN-UDA [9]	90.1±1.0%
CKM-UDA	**96.12±1.7%**

Table 1. Heterogeneous unsupervised domain adaptation accuracy (± one standard deviation) on SVHN $(0-4) \rightarrow$ MNIST $(5-9)$.

2. **Src+Targ Clustering.** The AE model is pre-trained on both source and target domain data. Then the CKM clustering and AE objectives are fine-tuned on the target domain.
3. **Deep Embedded Clustering (DEC)** [8]**:** We instantiate the UDA algorithm defined in Section 3.1 and Algorithm 1 by plugging in DEC instead of CKM.
4. **Deep Clustering Network (DCN)** [9]**:** We instantiate the same UDA algorithm using DCN rather than CKM as the base clustering algorithm.

The target domain accuracy of each method is shown in Table 1. From the results, we make the following observations: (1) Our CKM-based DA algorithm performs best overall, demonstrating the efficacy of our end-to-end deep clustering approach compared to the DEC and DCN alternatives. This is attributed to the benefit of CKM's unique ability to optimise hard clustering jointly and end-to-end with the rest of the DA model unlike DEC and DCN. (2) The margin between CKM-UDA and Target Clustering shows the benefit of using source data, as opposed to using target data alone. (3) The margin between CKM-UDA and Src+Targ clustering shows the benefit of using source labels, as opposed to solely source images in domain adaptation.

4.2. Heterogeneous Semi-Supervised Domain Adaptation

We finally evaluate our model on the heterogeneous semi-supervised domain adaptation problem introduced by [10]. This experiment evaluates (SVHN 0-4) → (MNIST 5-9) transfer where a few ($K = 2 \ldots 5$) labelled examples of the target domain classes available during training. The architecture of our model in this section is the same as in the previous Section 4.1, except we add the target domain classifier as described in Section 3.2. For competitors we consider state of the art alternatives **CFSM** [2] and Label-Efficient-Transfer (**LET**) [10]. We use the same data partitioning and deep architecture as [10, 2] for fair comparison.

Method	k = 2	k = 3	k = 4	k = 5
LET [10]	91.7±0.5	93.6±0.6	94.2±0.6	95.0±0.4
CFSM [2]	93.5±0.5	94.8±0.5	95.5±0.3	96.7±0.2
CKM-SSDA	**98.0±0.1**	**98.2±0.1**	**98.3±0.1**	**98.4±0.1**

Table 2. Heterogeneous Semi-supervised DA on SVHN (0-4) → MNIST (5-9). Accuracy (%, ± one standard deviation) for $k = 2 \ldots 5$ labels per class in the target domain.

The results in Table 2 show that our semi-supervised **CKM-SSDA** method outperforms these state of the art alternatives across the range of few-shot data evaluated. Compared to the competing algorithms CFSM and LET, we attribute the good quantitative performance of CKM to our effective clustering algorithm and better inductive bias for domain adaptation. LET uses unlabelled target domain data primarily for domain alignment by making the domains indistinguishable by discriminator. Domain alignment is not an ideal objective given disjoint labels. Moreover, we make better use of the labelled and unlabelled target data by finding both an embedding and a set of centroids where the unlabelled data groups nicely around the centroids, and also the labelled data can be predicted by interpreting the centroids as the RBF classifier means. In contrast, CFSM does not try to align the domains, but it makes a factorisation assumption that each example should be well explained by a set of low-entropy factors. However, this is not ideal for grouping the data into discrete categories compared to our clustering assumption. In contrast our CKM-SSDA both effectively transfers from source to target while also making good use of both labelled and unlabelled target data in an end-to-end deep learning setting.

5. CONCLUSION

This paper shows how the concrete k-means deep clustering framework can be modified to perform heterogeneous domain adaptation. Two variants of the domain adaptation problem are considered: one where no labels are available for target domain samples, and another where only a few target domain examples are labelled. Our experimental results show that the proposed method is competitive with state-of-the-art approaches for solving deep clustering problems, and surpasses state-of-the-art for heterogeneous domain adaptation problems. Our stochastic hard assignment method is able to estimate gradients of the nondifferentiable k-means loss function with respect to cluster centres. This, in turn, enables end-to-end training of a neural network feature extractor and a set of cluster centroids in this latent space. We attribute the success of our domain adaptation approaches to this ability to jointly train hard-assignment clustering models and neural networks.

6. REFERENCES

[1] Gabriela Csurka, *Domain Adaptation in Computer Vision Applications*, Springer, 2017.

[2] Xiaobin Chang, Yongxin Yang, Tao Xiang, and Timothy M Hospedales, "Disjoint label space transfer learning with common factorised space," in *AAAI Conference on Artificial Intelligence*, 2019.

[3] Hehe Fan, Liang Zheng, Chenggang Yan, and Yi Yang, "Unsupervised person re-identification: Clustering and fine-tuning," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 14, no. 4, 2018.

[4] Michael Kearns, Yishay Mansour, and Andrew Y. Ng, "An information-theoretic analysis of hard and soft assignment methods for clustering," in *Uncertainty in Artificial Intelligence*, 1997.

[5] Boyan Gao, Yongxin Yan, Henry Gouk, and Timothy M. Hospedales, "Deep clustering with concrete k-means," *arXiv preprint arXiv:1910.08031*, 2019.

[6] Chris J Maddison, Andriy Mnih, and Yee Whye Teh, "The concrete distribution: A continuous relaxation of discrete random variables," in *International Conference on Learning Representations*, 2017.

[7] Eric Jang, Shixiang Gu, and Ben Poole, "Categorical reparameterization with gumbel-softmax," in *International Conference on Learning Representations*, 2017.

[8] Junyuan Xie, Ross Girshick, and Ali Farhadi, "Unsupervised deep embedding for clustering analysis," in *International Conference on Machine Learning*, 2016.

[9] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," in *International Conference on Machine Learning*, 2017.

[10] Zelun Luo, Yuliang Zou, Judy Hoffman, and Li F Fei-Fei, "Label efficient learning of transferable representations acrosss domains and tasks," in *Advances in Neural Information Processing Systems 31*, 2017.

[11] Yaroslav Ganin and Victor Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning*, 2015.

[12] Xiaojin Zhu and Andrew B. Goldberg, "Introduction to semi-supervised learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 3, no. 1, pp. 1–130, 2009.

[13] Yves Grandvalet and Yoshua Bengio, "Semi-supervised learning by entropy minimization.," in *Advances in Neural Information Processing Systems 18*, 2004.