Article

Fuzzy Graph Learning Regularized Sparse Filtering for Visual Domain Adaptation

Lingtong Min ^{*} , Deyun Zhou, Xiaoyang Li, Qinyi Lv and Yuanjie Zhi

School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710129, China; dyzhou@nwpu.edu.cn (D.Z.); lixiaoyang@nwpu.edu.cn (X.L.); lvqinyi@nwpu.edu.cn (Q.L.); zyjiyuan@nwpu.edu.cn (Y.Z.)

* Correspondence: minlingtong@nwpu.edu.cn

Abstract: Distribution mismatch can be easily found in multi-sensor systems, which may be caused by different shoot angles, weather conditions and so on. Domain adaptation aims to build robust classifiers using the knowledge from a well-labeled source domain, while applied on a related but different target domain. Pseudo labeling is a prevalent technique for class-wise distribution alignment. Therefore, numerous efforts have been spent on alleviating the issue of mislabeling. In this paper, unlike existing selective hard labeling works, we propose a fuzzy labeling based graph learning framework for matching conditional distribution. Specifically, we construct the cross-domain affinity graph by considering the fuzzy label matrix of target samples. In order to solve the problem of representation shrinkage, the paradigm of sparse filtering is introduced. Finally, a unified optimization method based on gradient descent is proposed. Extensive experiments show that our method achieves comparable or superior performance when compared to state-of-the-art works.

Keywords: domain adaptation; fuzzy graph regularization; sparse filtering

Citation: Min, L.; Zhou, D.; Li, X.; Lv, Q.; Zhi, Y. Fuzzy Graph Learning Regularized Sparse Filtering for Visual Domain Adaptation. *Appl. Sci.* **2021**, *11*, 4503. <https://doi.org/10.3390/app11104503>

Academic Editor: Sungho Kim

Received: 2 April 2021

Accepted: 13 May 2021

Published: 14 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Visual recognition is a challenging subject and has attracted increasing attention of researchers over the world. Despite the success have achieved, the performance of leading models may degrade when applied on a cross-domain scenario, including but not limited to image classification [1], semantic segmentation [2] and person re-identification [3]. Such degradation stems from distribution mismatch, which is very common in real life. For images drawn from different sources, there are many factors leading to mismatches in distribution, including background, shooting angle, illumination and so on. Domain adaptation (DA) [4] aims at leveraging the rich knowledge (e.g., labels) in source domain to build classifiers and apply in a different but related target domain (with less or no labels), which has achieved dramatic success in computer vision [5]. Relevant researches can be divided into two main categories according to whether there are labels existing in target domain: Unsupervised domain adaptation (UDA), which means that there are no labels at all; Semi-supervised domain adaptation (SSDA), which supposes that part of labels in target domain are visible. In this paper, we focus on UDA problems. It is worth emphasizing that UDA follows the setting of transductive learning, data from both domains is available during training.

Mainstream UDA works focus on learning domain-invariant representations, they map original input into feature space with the goal of reducing domain discrepancy. Moment-based statistics are often employed to measure how the two domains (or distributions) like each other, such as Maximum Mean Discrepancy (MMD) [6] and CORrelation ALignmen (CORAL) [7]. As a practical example, Transfer Component Analysis (TCA) [8] attempted to minimize MMD by projecting features to a reproducing kernel Hilbert space (RKHS). CORAL matched the second-order statistics of two domains' distributions by means of linear transformation. Furthermore, the combination of moment-based statistics

and deep neural networks also shows excellent performance. Another kind of practical DA solution is pseudo labels-based method, which assigns pseudo labels to target samples so that class-wise adaptation is possible, also made a lot of advances [9]. The last one is adversarial training, both feature extractor and domain discriminator are constructed in the beginning, then they are updated asynchronously during training. Adversarial training can be applied as moment-based statistics, the principle difference is that a learning-based indicator is employed to measure domain discrepancy rather than handcrafted [10,11].

Though previous DA approaches have shown superior performance in some respects, there are two major drawbacks as follows: Firstly, moment-based and adversarial training-based DA methods are able to reduce distribution discrepancy between domains, but they can hardly guarantee that samples belonging to the same category between two domains are similar. For instance, studies indicated that the class weight bias would lead to significant performance reduction [12]. Secondly, for pseudo label-based DA methods, wrongly-chosen labels are inevitable, which will ruin further analysis. Some efforts have been paid on the selection of labels, which adopt an incremental framework. The underlying philosophy is that the classifier is weak in the beginning, so less labels should be picked, as the training process goes, the classifier obtains more stronger and more pseudo labels can be thought of high confidence. However, it has not fundamentally solved the problem of wrongly-chosen labels. In this paper, we attempt to handle this situation from another view: to reduce the negative effects of mislabeling. Specifically, we utilize soft-labeling to construct affinity graph, rather than hard-labeling. In order to keep the discrimination power of the learned representation, we regard the fuzzy graph objective as a regularization item and integrate it into the paradigm of sparse filtering [13]. Finally, a unified optimization framework based on gradient descent is proposed. The contributions of this paper are summarized as follows.

- We propose a novel solution for UDA problems, which is able to learn both discriminative and domain-shared representations simultaneously. Extensive experiments on several real-world datasets demonstrate its superiority to existing works.
- Different from previous hard-labeling methods, we design a fuzzy graph regularization based on soft-labeling. Specifically, it attempts to describe cross-domain affinity by means of probability matrix.
- In order to deal with the problem of representation shrinkage, we combine the proposed fuzzy graph regularization with unsupervised sparse filtering, so that the two objective will be antagonistic and converge to a compromise.

The rest of this paper is organized as follows. Section 2 gives the background knowledge on existing DA works, especially the highly-correlated pseudo label methods. Then sparse filtering is introduced. In Section 3, we describe our method in detail. Sections 4 and 5 present the experiments and some empirical analysis. Finally, Section 6 concludes the paper and provides some ideas for future researches.

2. Related Works

In this section, we first give a formal definition of domain adaptation, then conclude existing pseudo label based DA methods. For the sake of better understanding, we introduce sparse filtering subsequently.

2.1. Domain Adaptation

Domain adaptation is a branch of transfer learning, which deals with the scenario where training and testing data have different distributions, some literatures call it dataset bias or covariate shift. Specifically, there are two important points in domain adaptation. Matching marginal distributions of source and target domains cannot guarantee the separability of target samples, so pseudo label based DA methods become more popular. Overall, it can be divided into two categories according to whether the method has selection strategies on pseudo labels.

2.1.1. Pseudo Label without Selection

During training, it assigns pseudo labels on the whole target domain without considering the confidence. Class-wise distribution alignment is therefore feasible by utilizing the pseudo labeled target samples together with labeled source samples. After iterative learning, the pseudo labels are expected to be progressively more accurate. Obviously, this expectation is not always held, especially when the initial prediction is weak. As an extension of TCA, Joint Distribution Adaptation (JDA) [14] aligned conditional and marginal distribution simultaneously using pseudo labels, then an iterative training scheme is employed to obtain more accurate labels. Inspired by this, a series of studies tried to modify classical feature extraction methods (e.g., linear discriminant analysis and locality preserving projections [15,16]) by means of pseudo labels.

2.1.2. Pseudo Label with Selection

To alleviate the negative effects caused by wrongly-chosen labels, it selects a subset of target samples with corresponding pseudo labels. Specifically, both labels and their confidences need to be considered and only the selected target samples can be combined with source samples for training. An easy-to-hard training scheme is widely used for DA problems, which adopts an increasing number of target samples. CAPLS is an classical method of selective pseudo label, which uses softmax to denote the classify confidence [17]. Further, SPL considers structure information for more reliable labels [18].

2.2. Sparse Filtering

Sparse filtering is an unsupervised feature extraction algorithm. Unlike auto-encoder and sparse coding, it does not seek to model the data explicitly, but attempts to obtain ideal features from the perspective of so-called good features. First of all, it gives three requirements on the question of What is good features.

- Population Sparsity. Each example should be represented by only a few active features. Ideally, a sample should be a vector which contains lots of zeros and few non-zero values.
- Lifetime Sparsity. Good features should be distinguishable, therefore, a feature is only allowed to be activated in few samples. On the contrary, if a feature is activated by all the samples, we cannot classify samples according to this feature, so it is not a good feature.
- High Dispersal. It requires each feature to have similar statistical properties across all samples, which seems to be useless. The authors indicate that this can prevent feature degradation, such as similar features across samples.

Ngiam et al. pointed out that we can obtain the features that meet all the requirements by optimizing population sparsity and high dispersal [13]. Suppose now we have n samples, each with m -dimensional features that can be written as $x = x_1, x_2, \dots, x_n \in R^m$. The learning paradigm of sparse filtering is as follows:

(1) Linear feature extraction and non-linear activation. Let $f_j^{(i)}$ represent the j th feature of the i th sample, $f_j^{(i)} = x^{(i)}w_j$. Then we can use some activation functions to make it more expressive, such as soft-absolute function.

$$f_j^{(i)} = \sqrt{\epsilon + (x^{(i)}w_j)^2} \approx |x^{(i)}w_j| \quad (1)$$

(2) Solving high dispersal. Each feature is divided by the l_2 -norm of the feature on all samples.

$$\tilde{f}_j = \frac{f_j}{\|f_j\|_2} \quad (2)$$

Remember that the requirement of high dispersal is that the statistical properties of each feature are similar. This step forces the sum of the squares of all features to be 1 roughly.

(3) Solving population sparsity. Each sample is divided by its own l_2 -norm. Then we can get the objective function.

$$\text{minimize } \sum_{i=1}^n \left\| \frac{\tilde{f}^{(i)}}{\|\tilde{f}^{(i)}\|_2} \right\|_1 \quad (3)$$

This objective function can be easily understood through an example. Imagine that we have different expressions of a sample, $(0.6, 0.8)^T$ and $(1, 0)^T$. We say $(1, 0)^T$ is a sparser representation since it has more zero, so it has a smaller loss, $\frac{1}{\sqrt{1^2+0^2}} + \frac{0}{\sqrt{1^2+0^2}} = 1 \leq \frac{0.6}{\sqrt{0.6^2+0.8^2}} + \frac{0.8}{\sqrt{0.6^2+0.8^2}} = 1.4$.

3. Methodology

In this section, we give the mathematical definition of DA problem, describe each component of the proposed method in detail and analyze the complexity.

3.1. Problem Definition and Notations

Given labeled source domain $\mathcal{D}_s = \{(x_s^1, y_s^1), (x_s^2, y_s^2), \dots, (x_s^{n_s}, y_s^{n_s})\}$ and unlabeled target domain $\mathcal{D}_t = \{(x_t^1), (x_t^2), \dots, (x_t^{n_t})\}$, the goal is to build classifier and make predictions on target data. Specifically, we study homogeneous DA, which means that the source and target data have the same dimensions, $x_s, x_t \in R^m$. The target labels y_t are only available for evaluating the algorithm. Table 1 introduces some necessary notations and descriptions used in this paper.

Domain: a domain, $\mathcal{D}$, consists of two parts, data $\mathcal{X}$ and the corresponding distribution $p(\mathcal{X})$, so $\mathcal{D} = \{\mathcal{X}, p(\mathcal{X})\}$. Generally, we have $\mathcal{D}_s$ and $\mathcal{D}_t$ to represent the labeled source and unlabeled target domain, respectively. When it refers to domain adaptation, we have $p(\mathcal{X}_s) \neq p(\mathcal{X}_t)$. It is worth emphasizing to obtain the explicit distribution $p(\mathcal{X})$ is often very hard, especially for high-dimensional data, so an alternative solution is utilizing some statistics, i.e., mean or variance, to estimate the distribution.

Task: a task, $\mathcal{T}$, is the specific application of domain adaptation, which contains two parts, label space $\mathcal{Y}$ and the mapping function $f(\cdot)$, so $\mathcal{T} = \mathcal{Y}, f(\cdot)$. Naturally, given data $\mathcal{X}$, we have $\mathcal{Y} = f(\mathcal{X})$. For standard domain adaptation problems, the tasks are the same, $\mathcal{T}_s = \mathcal{T}_t$.

Table 1. Notations and descriptions used in this paper.

Notations	Description
$\mathcal{D}_s / \mathcal{D}_t$	source/target domain
x_s / x_t	original source/target domain data
y_s / y_t	source/target domain label
n_s / n_t	number of source/target samples
m / k	original/transformed feature dimension
$\hat{x}_s / \hat{x}_t$	source/target features
$f(\cdot)$	mapping function, $\hat{X} = f(X)$
W	the transformation matrix to be solved
L_{FGL}	objective function for fuzzy graph learning
L_{SF}	objective function for sparse filtering
λ	the balance factors between two objectives

3.2. Fuzzy Graph Learning for Domain Alignment

How to reduce the domain discrepancy is a key problem for DA, moment-based and adversarial training-based strategies cannot guarantee the separability of target samples,

because they do not exploit the structural information existing across different classes. Therefore, we propose fuzzy graph learning for class-wise alignment. The underlying idea is: samples belonging to the same class (no matter which domain it comes from) should be close to each other. Since there are no labels for target samples, we must adopt pseudo labels as a substitution. Obviously, the quality of pseudo labels will greatly influence the learned classifier/representation, it may cause a catastrophic accumulation of errors if not handled properly.

For our method, we propose fuzzy pseudo labels which employ a probability matrix to update the classifier. In Figure 1, we give a graphical illustration to explain how soft label works. The class centroids are calculated by labeled source samples and the decision boundary can be obtained by means of the distance to each centroid. For an unreliable training sample which located in the intersection of decision boundary (also can be described as it has similar distance to each centroid), the probabilities of it belonging to any category is extremely close, e.g., 24%, 25%, 26%. In such situation, assigning the label with biggest probability is risky. Like the figure in the middle shows, the updated centroid would have a big influence to decision boundaries. When we adopt soft labels, the case is shown in the last figure, each centroid would move a distance to the training sample according to corresponding probability value. Intuitively, soft labels have significant less effect on decision boundaries than hard.

Figure 1. Unreliable training samples and its influence on decision boundary.

Specifically, suppose we have the learned representation $\hat{x}_s = f(x_s) \in R^{k \times n_s}$ and $\hat{x}_t = f(x_t) \in R^{k \times n_t}$ (we will introduce the transformation $f(\cdot)$ later), along with the source labels $y_s \in [1, 2, \dots, c]$. We use $x = [\hat{x}_s, \hat{x}_t] \in R^{k \times (n_s + n_t)}$ to denote the combination of two domains' features. The objective of fuzzy graph learning can be written as:

$$L_{FGL} = \frac{1}{2(n_s + n_t)} \sum_{i=1}^{n_s+n_t} \sum_{j=1}^{n_s+n_t} (x_i - x_j)^2 W_{ij} \quad (4)$$

where $W \in R^{(n_s+n_t) \times (n_s+n_t)}$ is the affinity graph where $W_{ij} \in [0, 1]$ denotes the similarity between x_i and x_j , and W can be defined as follows:

$$W = \begin{pmatrix} W_{ss} & W_{st} \\ W_{ts} & W_{tt} \end{pmatrix} \quad (5)$$

Fuzzy connections only exist in cross-domain cases, so we describe how to construct the affinity graph separately:

Construction of W^{ss} : For labeled source domain, we adopt the supervised graph, that is, the edges exist only between samples with the same label. Ideally, this can help obtain discriminative representations.

$$W_{ij}^{ss} = \begin{cases} 1, & \text{if } Y_i = Y_j \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

Construction of W^{tt} : For unlabeled target domain, we encourage target samples to be close to their neighborhood samples, thus a cut-off is necessary. In this paper, we exploit the top- k ($k = 10$ in this paper) nearest neighbors for target domain.

$$W_{ij}^{tt} = \begin{cases} 1, & \text{if } X_j \in \mathcal{N}_{10}(X_i) \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where $\mathcal{N}_{10}(X_i)$ means the aggregation of X_i 's top-10 nearest neighbors in target domain.

Construction of W^{st} and W^{ts} : To build the fuzzy graph to model the cross-domain similarity, we first compute the class centroids $\bar{x}$ of source samples, then the probability $p(y|\hat{x}_t^i)$ of target sample $\hat{x}_t^i$ belongs to class y can be computed based on the Euclidean distance to each centroid. Finally, W_{st} indicates the similarity from source to target samples, then for a certain source sample $\hat{x}_s^i$ and target sample $\hat{x}_t^j$, if they belongs to the same category, they should have a strong connection and vice versa. Since we utilize soft labels here, it can be interpreted as: if the target sample $\hat{x}_t^j$ has a big probability of possessing the same label of the source sample (noted that the source sample $\hat{x}_s^i$ and its label y_s^i are all available), they should have a strong connection. The pseudo code of cross-domain affinity graph construction is given in Algorithm 1, and we can obtain W_{ts} by $W_{ts} = W_{st}^T$ directly.

Algorithm 1: Cross-domain affinity graph (W_{st}) construction.

Input: $\hat{x}_s, \hat{x}_t, y_s, c$

```

1 for  $i = 1, 2, \dots, c$  do
2    $\bar{x}^i = \frac{\sum_{j=1}^{n_s} \hat{x}_s^j \delta(y_s^j, i)}{\sum_{j=1}^{n_s} \delta(y_s^j, i)}, \quad \delta(y_s^j, i) = \begin{cases} 1, & \text{if } y_s^j = i \\ 0, & \text{otherwise} \end{cases}$ 
3 end
4 Compute target probability labels.
5 for  $i = 1, 2, \dots, n_t$  do
6    $p(y|\hat{x}_t^i) = \frac{\exp(-||\hat{x}_t^i - \bar{x}^y||)}{\sum_{y=1}^c \exp(-||\hat{x}_t^i - \bar{x}^y||)}$ 
7 end
8 Construct fuzzy affinity graph  $W_{st}$ 
9 for  $i = 1, 2, \dots, n_s$  do
10  for  $j = 1, 2, \dots, n_t$  do
11     $(W_{st})_{ij} = p(y_s^i|\hat{x}_t^j)$ 
12  end
13 end
14 return  $Y, C_{test}$ 
```

3.3. Sparse Filtering for Discriminative Feature Learning

Fuzzy graph learning is able to reduce the domain discrepancy, but it is an ill-posed problem in nature. For instance, if we choose two constant function, $k_1(x) = a, k_2(x) = b$, where $a \neq b$, they both can minimize the objective L_{FGL} . For the intuitive example shown in Figure 1, we can see that unreliable samples lead to representation shrinkage. Consequently, the learned representation will lose the discriminability, hence an additional penalty is indispensable. In this paper, we adopt the objective of sparse filtering for two reasons: Firstly, it requires lifetime sparsity, which projects the feature onto a L2-ball, this can

effectively alleviate the feature contraction. Secondly, it allows us to exploit the potential discriminant structures in unlabeled target domain.

Corresponding to the optimization of sparse filtering, the transformation function $k(\cdot)$ of our method can be summarized as follows: Given original input $x_s, x_t \in R^{n_s, n_t \times m}$, the goal is to find optimal $W \in R^{m \times k}$, then obtain new representations $\hat{x}_s, \hat{x}_t \in R^{n_s, n_t \times k}$ for downstream applications.

1. Linear transformation and non-linear activation. We use the soft absolute function as activation function in this paper.

$$\begin{aligned}\hat{x}_{s1} &= |x_s W + \epsilon| \\ \hat{x}_{t1} &= |x_t W + \epsilon|\end{aligned}\quad (8)$$

where ϵ denotes a small number, such as $1e-5$.

2. L2 column normalization Observing the form of current $\hat{x}$, each row represents a sample, and each column represents a feature. This measure enforces the L2 norm of each feature to be 1 thus high dispersal can be achieved. Noted that we do the within-domain normalization since it is proven to be powerful for DA problems.

$$\begin{aligned}\hat{x}_{s2} &= \hat{x}_{s1} \circ M_{scol}, (M_{scol})_{ij} = \frac{1}{\sqrt{\sum_{i=1}^{n_s} (\hat{x}_{s1})_{ij}^2}} \\ \hat{x}_{t2} &= \hat{x}_{t1} \circ M_{tcol}, (M_{tcol})_{ij} = \frac{1}{\sqrt{\sum_{i=1}^{n_t} (\hat{x}_{t1})_{ij}^2}}\end{aligned}\quad (9)$$

which the symbol $\circ$ represents Hadamard product. Here we use Hadamard product to simplify gradient derivation.

3. L2 row normalization. Just like the previous step, this measure enforce the L2 norm of each sample to be 1 to project data onto L2-ball.

$$\begin{aligned}\hat{x}_{s3} &= \hat{x}_{s2} \circ M_{srow}, (M_{srow})_{ij} = \frac{1}{\sqrt{\sum_{j=1}^k (\hat{x}_{s2})_{ij}^2}} \\ \hat{x}_{t3} &= \hat{x}_{t2} \circ M_{trow}, (M_{trow})_{ij} = \frac{1}{\sqrt{\sum_{j=1}^k (\hat{x}_{t2})_{ij}^2}}\end{aligned}\quad (10)$$

For the sake of simplification, we integrate the mapping from x_s, x_t to $\hat{x}_{s3}, \hat{x}_{t3}$ as a single step $k(\cdot)$, so we can obtain the final representation by $\hat{x}_s = k(x_s) = \hat{x}_{s3}, \hat{x}_t = k(x_t) = \hat{x}_{t3}$. Correspondingly, the objective of sparse filtering can be represented as:

$$L_{SF} = \frac{1}{n_s} \|\hat{x}_s\|_1 + \frac{1}{n_t} \|\hat{x}_t\|_1 \quad (11)$$

The total objective of our method can be written as:

$$L = L_{SF} + \lambda L_{FGL} \quad (12)$$

where λ is hyper-parameter to balance two objectives.

3.4. Optimization

Intuitively, our method can be formally defined as a weighted sum of two objectives, so we analyze the two parts separately.

3.4.1. Gradient of $\frac{\partial L_{FGL}}{\partial \hat{x}}$

Thinking about the combination of two domains' features, $\hat{x} = [\hat{x}_s, \hat{x}_t]$ the objective can be reformulated as:

$$\begin{aligned} L_{FGL} &= \frac{1}{2(n_s + n_t)} \sum_{i=1}^{n_s+n_t} \sum_{j=1}^{n_s+n_t} (\hat{x}_i - \hat{x}_j)^2 W_{ij} \\ &= \frac{1}{2(n_s + n_t)} \left(\sum_{i=1}^{n_s+n_t} \hat{x}_i^T D_{ii} \hat{x}_i - \sum_{i=1}^{n_s+n_t} \sum_{j=1}^{n_s+n_t} \hat{x}_i^T W_{ij} \hat{x}_j \right) \\ &= \frac{1}{2(n_s + n_t)} \text{trace}(\hat{x}^T L \hat{x}) \end{aligned} \quad (13)$$

where $D_{ii} = \sum_{j=1}^{n_s+n_t} W_{ij}$ denotes the degree of $\hat{x}_i$, and $L = D - W$ is called Laplacian graph.

Naturally, we can compute the gradient by:

$$\frac{\partial L_{FGL}}{\partial \hat{x}} = \frac{1}{n_s + n_t} L \hat{x} = \left[\frac{\partial L_{FGL}}{\partial \hat{x}_s} \quad \frac{\partial L_{FGL}}{\partial \hat{x}_t} \right] \quad (14)$$

3.4.2. Gradient of L_{SF}

Noted that we have $\hat{x} \geq 0$ because of the activation. So the gradient of sparse filtering is simply a matrix full of 1.

$$\frac{\partial L_{SF}}{\partial \hat{x}} = \mathbf{1} = \left[\frac{\partial L_{SF}}{\partial \hat{x}_s} \quad \frac{\partial L_{SF}}{\partial \hat{x}_t} \right] \quad (15)$$

Here we use $\mathbf{1}$ to represent the 1 matrix with certain dimension, e.g., $\frac{\partial L_{SF}}{\partial \hat{x}_s} = \mathbf{1} \in R^{n_s \times k}$ and $\frac{\partial L_{SF}}{\partial \hat{x}_t} = \mathbf{1} \in R^{n_t \times k}$.

3.4.3. Gradient of $k(\cdot)$

To propagate the gradient back to the input layer, we need $\frac{\partial \hat{x}}{\partial W}$ to apply the chain rule. There are some differences for two domains since we adopt within-domain normalization. To obviate misunderstanding, we give the specific form for two domains.

$$\begin{aligned} \frac{\partial \hat{x}_s}{\partial W} &= x_s^T (M_{srow} \circ M_{scol} \circ \text{sign}(x_s W)) \\ \frac{\partial \hat{x}_t}{\partial W} &= x_t^T (M_{trow} \circ M_{tcol} \circ \text{sign}(x_t W)) \\ \frac{\partial \hat{x}}{\partial W} &= \left[\frac{\partial \hat{x}_s}{\partial W} \quad \frac{\partial \hat{x}_t}{\partial W} \right] \end{aligned} \quad (16)$$

3.4.4. Unified Optimization Based on Gradient Descent

Given above gradients, we are able to update W by gradient descent from random initialization. The training process is similar to train a neural network, some tricks that help converge can also be applied here, such as initializing with normal distribution and utilizing a decreasing step. The specific training scheme is shown in Algorithm 2.

Algorithm 2: Optimization of the proposed method.

```

1 Random initialization.
2  $W = \mathcal{N}(0, 1) \in R^{m \times k}$ 
3 Gradient descent.
4 while not converge do
5    $\Delta = \left( \frac{\partial L_{SF}}{\partial \hat{x}} + \lambda \frac{\partial L_{FGL}}{\partial \hat{x}} \right) \frac{\partial \hat{x}}{\partial W}$ 
6    $W = W - \alpha \Delta$ 
7 end
8 return  $W$ 
```

4. Experiments

In this section, we introduce two widely-used DA datasets, then the proposed method is compared with state-of-the-art works. In addition, we provide parameter sensitivity analysis to show the robustness of the proposed method.

4.1. Datasets

ImageCLEF <https://www.imageclef.org/2014/adaptation> (accessed on 14 May 2021). ImageCLEF is an online DA competition, which contains three domains: Caltech256 (C) [19], ImageNet (I) [20] and PASCAL (P) [21]. There are twelve kinds of objects, and fifty images of each object. Some pictures can be found in Figure 2. In our experiments, we adopt the ResNet50 (input dimension $m = 2048$) features of images, and six tasks, e.g., $C \rightarrow I, C \rightarrow P$, are constructed.

Figure 2. Objects in different domain.

Office-Caltech10 [22]: Office-Caltech10 contains ten object categories drawn from 4 image domains: Amazon (A), Webcam (W), DSLR (D), and Caltech256 (C). There are 8–151 samples per category per domain, and 2533 images in total. In our experiments, we adopt the Decaf6 (input dimension $m = 4096$) features of images as input for testing algorithms, twelve cross-domain tasks, e.g., $C \rightarrow A, C \rightarrow W, C \rightarrow D$, are constructed.

4.2. Experimental Setting

Since there are no deep architectures in the proposed method, we compare our approach to several state-of-the-art shallow methods to evaluate its effectiveness.

- Nearest Neighbor(NN) [23]. NN is served as a baseline model to check whether the learned representations really work for DA problems.
- Joint Distribution Alignment(JDA) [14]. JDA [Long et al. ICCV2013] adopts pseudo labels to align the conditional distributions of two domains.
- Correlation Alignment(CORAL) [7]. CORAL [Sun et al. AAAI2016] obtains transferable representations by aligning the second-order statistics of distributions.
- Confidence-Aware Pseudo Label Selection(CAPLS) [17]. CAPLS [Wang et al. IJCNN2019] uses a selective pseudo labeling procedure to obtain more reliable labels.
- Modified A-distance Sparse Filtering(MASF) [24]. MASF [Han et al. Pattern Recognit.2020] employs an L2 constraint combining sparse filtering to learn both domain-shared and discriminative representations.
- Selective Pseudo Labeling(SPL) [18]. SPL [Wang et al. AAAI2020] is also a selective pseudo labeling strategy based on structured prediction.

Follow the experimental setting of JDA and MASF, we set the subspace dimension $k = 100$. For JDA, we set the regularization coefficient $\lambda = 1$ and the number of iterations $T = 10$. For CAPLS, we set the number of iteration $T = 10$. For MASF, we set the regularization coefficient $\alpha = 1e - 3$. For SPL, we set the number of iterations $T = 11$. It is worth emphasizing that the input features are extracted by deep networks without fine-tuning and not any pre-processing strategy is applied in the experiments.

4.3. Results

In this section, we report the accuracy of the proposed method (abbreviated as ‘FGLSF’ for Fuzzy Graph Learning regularized Sparse Filtering) and other state-of-the-art works, the results are shown on Table 2. From experimental results, we have the following observations:

- FGLSF vs. NN. According to the results, FGLSF is significantly better than NN. NN cannot handle the domain discrepancy, thus results in unsatisfying performance. On the other hand, it indicates that our method is able to learn transferable representations.
- FGLSF vs. CORAL, JDA. FGLSF is superior to CORAL and JDA. This two methods are classical distribution matching works, but they have limited considerations on the discrimination of learned representations.
- FGLSF vs. MASF. MASF is another framework based on sparse filtering, which adopts a modified $\mathcal{A}$ distance for domain alignment. Compared to our method, it cannot deal with the problem of conditional distribution matching.
- FGLSF vs. CAPLS, SPL. Objectively speaking, our method FGLSF has comparable performance when compared state-of-the-art selective hard labeling method, only a 0.3% improvement is gained. It shows that the proposed fuzzy graph regularization is also valid for alleviating the negative effects caused by wrong labeling.

Another interesting finding is that the proposed method works well in ImageCLEF, achieves best performance on all subtasks, but when it refers to Office-Caltech10, it seems that SPL works better. For these two datasets, the biggest difference is the class weights, ImageCLEF is an absolutely class-balanced dataset, which means that each object has the same number of pictures, but Office-Caltech10 has class weight bias, which makes it a more complicated problem. When we infer the pseudo labels, the object who has more examples is expected to be dominant, and vice versa. For the proposed method, we only use source labels for obtaining pseudo labels, so it may easily influenced by the class weight bias. SPL considers both source labels and manifold structures, thus achieves better performance on Office-Caltech10.

Table 2. Performance (accuracy %) on Office-Caltech10 (No. 1–12) and ImageCLEF (No. 13–18).

No.	Task	NN	JDA	CORAL	CAPLS	MASF	SPL	JAN	DAN	FGLSF
1	C → A	85.69	89.77	92.00	90.90	90.81	92.80	91.90	92.00	93.32
2	C → W	66.10	83.73	80.00	88.83	87.46	85.08	85.40	90.60	84.75
3	C → D	74.52	86.62	84.70	90.08	89.81	91.72	88.80	89.30	87.90
4	A → C	70.35	82.28	83.20	80.66	87.36	81.39	85.00	84.10	85.31
5	A → W	57.29	78.64	74.60	80.69	81.02	84.07	86.10	91.80	88.14
6	A → D	64.97	80.25	84.10	89.45	86.62	90.45	89.00	91.70	87.26
7	W → C	60.37	83.53	75.50	86.62	85.04	74.00	78.00	81.20	74.27
8	W → A	62.53	90.19	81.20	91.38	91.34	91.96	84.90	92.10	89.25
9	W → D	98.73	100.00	100.00	100.00	99.36	100.00	100.00	100.00	97.45
10	D → C	52.09	85.13	76.80	88.05	85.75	88.51	81.10	80.30	83.35
11	D → A	62.73	91.44	85.50	92.32	90.40	93.32	89.50	90.00	91.75
12	D → W	89.15	98.98	99.30	98.66	98.98	100.00	98.20	98.50	95.59
13	C → I	85.16	92.00	83.00	91.00	89.83	90.83	89.50	89.50	94.00
14	C → P	69.16	75.50	71.50	77.33	72.83	78.17	74.20	75.80	79.70
15	I → C	91.16	92.33	88.66	94.17	93.17	94.33	94.70	94.20	97.00
16	I → P	73.16	77.00	73.66	75.80	76.83	77.50	76.80	78.20	80.71
17	P → C	81.33	83.83	72.50	90.67	85.33	91.33	91.70	89.20	95.50
18	P → I	74.50	79.16	72.33	85.00	80.83	85.83	88.00	87.50	93.00
19	AVG	73.28	86.13	82.14	88.42	87.38	88.40	87.38	88.66	88.79

4.4. Parameter Sensitivity Analysis

For our method, there is one hyper-parameter to be adjusted, λ indicates the weight of fuzzy graph learning. We run FGLSF varying with values of parameter λ which are sampled from $\{0, 0.001, 0.01, 0.1, 0.2\}$ on three tasks, e.g., $C \rightarrow W$, $W \rightarrow C$ and $P \rightarrow C$. If λ is too small, the model would fail to align distributions of two domains. And as it getting larger, the problem of representation shrinkage is emerging. The experimental results shown on Figure 3 are consistent with the analysis, the accuracy first rises and then falls as λ gets bigger.

Figure 3. Performance on three subtasks with different λ . X-axis: tasks. Y-axis: accuracy.

5. Discussion

To prove the effectiveness of the proposed method, we conduct two sets of comparative experiments: 1. How the accuracy changes during training? 2. How about using selective hard labeling?

5.1. Does Iterative Learning Help Improve the Model?

To answer the question “Does iterative learning help improve the model”, we report the performance of 18 tasks during training, whose results are shown on Figure 4. Specifically, we divide the full training process to 10 parts, and record current accuracy. Obviously, the model becomes more precise as training goes. On the other hand, we can also see some fluctuations, which suggests that our approach cannot completely eliminate the effects of mislabeling.

Figure 4. Performance during training. X-axis: training process. Y-axis: accuracy.

5.2. Fuzzy Labeling versus Selective Hard Labeling

Figure 5 shows the performance of two labeling strategies. Intuitively, the proposed fuzzy labeling and state-of-the-art selective hard labeling have similar performance, e.g., selective hard labeling has better performance on tasks {7, 8} while fuzzy labeling is superior on tasks {5, 15}. This finding confirms that the proposed fuzzy labeling is able to alleviate the issue of wrong-labels.

Figure 5. Performance of two labeling strategies. X-axis: tasks. Y-axis: accuracy.

6. Conclusions

In this paper, we propose a novel DA solution based on fuzzy graph learning, which aims to learn both discriminative and domain-shared representations simultaneously. When compared to data augmentation methods, it considers how to learn an supervised model under dataset bias, aims at minimizing the performance degradation brought by domain discrepancy. The main contribution is that we propose another view for the problem of misselected pseudo labels, from selecting labels to minimizing the negative effects of mislabeling. When compared to existing methods, the proposed method adopts a fuzzy labeling framework, which uses label matrix rather than a single selected label for target samples. To avoid trivial solutions, we combine sparse filtering and fuzzy graph learning to solve the problem of representation shrinkage. The experimental results verify that our method outperforms many SOTA shallow methods; the increases of mean accuracy are 0.3–6.6%. In the future, we plan to study how to adjust the hyper-parameter λ according to task similarity.

Author Contributions: Conceptualization, L.M. and D.Z.; methodology, L.M. and D.Z.; software, L.M. and X.L.; validation, L.M. and Q.L.; formal analysis, Q.L.; investigation, L.M.; resources, Y.Z.; data curation, Y.Z.; writing—original draft preparation, L.M.; supervision, D.Z.; project administration, D.Z.; funding acquisition, D.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 62076204).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Saenko, K.; Kulis, B.; Fritz, M.; Darrell, T. Adapting visual category models to new domains. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 213–226.
2. Patel, V.M.; Gopalan, R.; Li, R.; Chellappa, R. Visual Domain Adaptation: A survey of recent advances. *IEEE Signal Process. Mag.* **2015**, *32*, 53–69. [\[CrossRef\]](#)
3. Li, X.; Grandvalet, Y.; Davoine, F.; Cheng, J.; Cui, Y.; Zhang, H.; Belongie, S.; Tsai, Y.; Yang, M. Transfer learning in computer vision tasks: Remember where you come from. *Image Vis. Comput.* **2020**, *93*, 103853. [\[CrossRef\]](#)
4. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [\[CrossRef\]](#)
5. Wang, M.; Deng, W. Deep visual domain adaptation: A survey. *Neurocomputing* **2018**, *312*, 135–153. [\[CrossRef\]](#)
6. Smola, A.J.; Gretton, A.; Song, L.; Scholkopf, B. A Hilbert Space Embedding for Distributions. In *Algorithmic Learning Theory*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 13–31.
7. Sun, B.; Feng, J.; Saenko, K. Return of frustratingly easy domain adaptation. In Proceedings of the National Conference on Artificial Intelligence, Phoenix, AZ, USA, 2–9 February 2016; pp. 2058–2065.
8. Pan, S.J.; Tsang, I.W.; Kwok, J.T.; Yang, Q. Domain Adaptation via Transfer Component Analysis. *IEEE Trans. Neural Netw. Learn. Syst.* **2011**, *22*, 199–210. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Zhang, W.; Ouyang, W.; Li, W.; Xu, D. Collaborative and Adversarial Network for Unsupervised Domain Adaptation. In Proceedings of the Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3801–3809.
10. Tzeng, E.; Hoffman, J.; Saenko, K.; Darrell, T. Adversarial Discriminative Domain Adaptation. In Proceedings of the Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2962–2971.
11. Ganin, Y.; Lempitsky, V. Unsupervised Domain Adaptation by Backpropagation. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 1180–1189.
12. Yan, H.; Ding, Y.; Li, P.; Wang, Q.; Xu, Y.; Zuo, W. Mind the Class Weight Bias: Weighted Maximum Mean Discrepancy for Unsupervised Domain Adaptation. In Proceedings of the Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 945–954.
13. Ngiam, J.; Chen, Z.; Bhaskar, S.A.; Koh, P.W.; Ng, A.Y. Sparse Filtering. In Proceedings of the Neural Information Processing Systems, Granada, Spain, 12–15 December 2011; pp. 1125–1133.
14. Long, M.; Wang, J.; Ding, G.; Sun, J.; Yu, P.S. Transfer Feature Learning with Joint Distribution Adaptation. In Proceedings of the International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 2200–2207.
15. He, X.; Niyogi, P. Locality Preserving Projections. In Proceedings of the Neural Information Processing Systems, Vancouver, BC, Canada, 8–13 December 2003; pp. 153–160.
16. Sanodiya, R.K.; Mathew, J. A novel unsupervised Globality-Locality Preserving Projections in transfer learning. *Image Vis. Comput.* **2019**, *90*, 103802. [\[CrossRef\]](#)

17. Wang, Q.; Bu, P.; Breckon, T.P. Unifying Unsupervised Domain Adaptation and Zero-Shot Visual Recognition. In Proceedings of the International Joint Conference on Neural Network, Budapest, Hungary, 14–19 July 2019; pp. 1–8.
18. Wang, Q.; Breckon, T.P. Unsupervised Domain Adaptation via Structured Prediction Based Selective Pseudo-Labeling. In Proceedings of the National Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 1–10.
19. CaltechAUTHORS. *Caltech-256 Object Category Dataset*; Technical Report; Pasadena, CA, USA, 2007. Available online: http://www.vision.caltech.edu/Image_Datasets/Caltech256 (accessed on 14 May 2021).
20. Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; Feifei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
21. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. Available online: <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html> (accessed on 14 May 2021).
22. Gong, B.; Shi, Y.; Sha, F.; Grauman, K. Geodesic flow kernel for unsupervised domain adaptation. In Proceedings of the Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 2066–2073.
23. Cover, T.M.; Hart, P.E. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [[CrossRef](#)]
24. Han, C.; Lei, Y.; Xie, Y.; Zhou, D.; Gong, M. Visual Domain Adaptation Based on Modified A Distance and Sparse Filtering. *Pattern Recognit.* **2020**, *104*, 107254. [[CrossRef](#)]