entropy

MDPI

Article

Domain Adaptation with Data Uncertainty Measure Based on Evidence Theory

Ying Lv [1], Bofeng Zhang [2,3,*], Guobing Zou [1], Xiaodong Yue [1], Zhikang Xu [1] and Haiyan Li [3]

[1] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China; lvying2016@shu.edu.cn (Y.L.); gbzou@shu.edu.cn (G.Z.); yswantfly@shu.edu.cn (X.Y.); xuzhikangnba@shu.edu.cn (Z.X.)
[2] School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai 201209, China
[3] School of Computer Science and Technology, Kashi University, Kashi 844006, China; lihaiyan_2016@sjtu.edu.cn
* Correspondence: bfzhang@sspu.edu.cn

Abstract: Domain adaptation aims to learn a classifier for a target domain task by using related labeled data from the source domain. Because source domain data and target domain task may be mismatched, there is an uncertainty of source domain data with respect to the target domain task. Ignoring the uncertainty may lead to models with unreliable and suboptimal classification results for the target domain task. However, most previous works focus on reducing the gap in data distribution between the source and target domains. They do not consider the uncertainty of source domain data about the target domain task and cannot apply the uncertainty to learn an adaptive classifier. Aimed at this problem, we revisit the domain adaptation from source domain data uncertainty based on evidence theory and thereby devise an adaptive classifier with the uncertainty measure. Based on evidence theory, we first design an evidence net to estimate the uncertainty of source domain data about the target domain task. Second, we design a general loss function with the uncertainty measure for the adaptive classifier and extend the loss function to support vector machine. Finally, numerical experiments on simulation datasets and real-world applications are given to comprehensively demonstrate the effectiveness of the adaptive classifier with the uncertainty measure.

Keywords: domain adaptation; transfer learning; evidence theory; uncertainty measure

Citation: Lv, Y.; Zhang, B.; Zou, G.; Yue, X.; Xu, Z.; Li, H. Domain Adaptation with Data Uncertainty Measure Based on Evidence Theory. *Entropy* **2022**, *24*, 966. https://doi.org/10.3390/e24070966

Academic Editors: Andrea Prati, Luis Javier García Villalba and Vincent A. Cicirello

Received: 20 June 2022
Accepted: 8 July 2022
Published: 13 July 2022

1. Introduction

In the field of machine learning research, supervised learning methods have already witnessed the outstanding performance in many applications. The key point of supervised learning is to collect sufficient labeled datasets for model training, which also limits the usage of supervised learning in scenarios with a lack of training data. Furthermore, data annotating is usually a time-consuming, labor-expensive, or even unrealistic task. To settle this situation, domain adaption (DA) is a promising methodology that aims to learn an adaptive classifier for the target domain tasks by making use of labeled data from source domains [1–4]. It has been applied in various fields successfully, such as object recognition [5,6], text classification [7,8], medical field [9,10], machine translation [11] and so on.

However, due to the mismatch between the source domain data and the target domain task, there is an uncertainty in DA when source domain data transfers to tasks of the target domain. As shown in Figure 1, in the target domain classification task, each source domain datum may no longer fully belong to a class in the label space of the target domain. The possibility of it being in class 1* is 0.2, and the uncertainty is 0.8, or the possibility of it being in class 1* is 0.9, and the uncertainty is 0.1. Unfortunately, the uncertainty of source domain data with respect to the target domain task is given less attention in DA. Ignoring uncertainty may result in an issue that the classifier does not fully match the target domain task, which weakens the model's transfer performance.

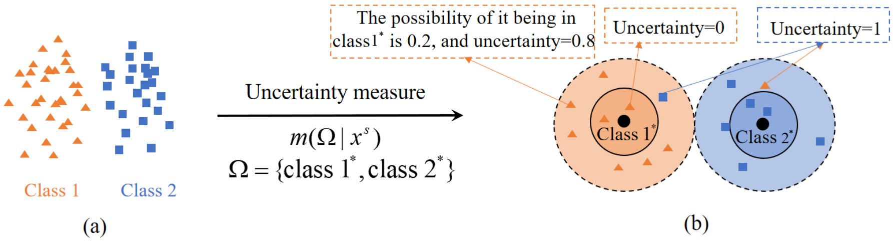

Class 1
Class 2
(a)
Uncertainty measure
$m(\Omega \mid x^s)$
$\Omega = \{\text{class } 1^*, \text{class } 2^*\}$
The possibility of it being in class1* is 0.2, and uncertainty=0.8
Uncertainty=0
Uncertainty=1
Class 1*
Class 2*
(b)

Figure 1. (**a**) Data distribution of source domain, (**b**) category distribution of source domain data in label space Ω of the target domain.

Most DA research works adopted metric learning to minimize the data differences between the source and target domain for getting an adaptive classifier. Some works map the source and target data instances into a common feature space by minimizing the gap between the data distributions of the source and target domain, such as transfer component analysis (TCA) [12], correlation alignment (CORAL) [13], and scatter component analysis (SCA) [14]. Some works construct a loss function with the data differences as the constraint to train an adaptive classifier, such as joint adaptation network (JAN) [15], manifold embedded distribution alignment (MEDA) [16], and multi-representation adaptation network (MRAN) [17]. However, existing methods (1) cannot measure the uncertainty of source domain data about the target domain task, and (2) cannot accomplish effective training of adaptive classifiers with a data uncertainty measure.

The uncertainty is important for evaluating the adaptation degree of the source domain data about target tasks. The study of uncertainty has been successfully applied in traditional machine learning, such as bayesian-based uncertainty [18], evidence theory-based uncertainty [19], information entropy-based uncertainty [20], and granular computing-based uncertainty [21]. In particular, the evidence theory has been widely combined with machine learning methods to improve their ability to handle the uncertainty data [22–26].

To solve these problems, in this paper, we revisit the domain adaptation from source domain data uncertainty based on evidence theory and thereby devise a reliable adaptive classifier with the uncertainty measure. Specifically, we first construct an evidence net based on evidence theory for measuring the uncertainty of source domain data about the target domain classification task. It can calculate the proportion of uncertainty for each source domain instance in the target domain classification task. Second, we design a general loss function with the uncertainty measure for the adaptive classifier and extend the loss function to support vector machine (SVM). The contributions of this paper are summarized as follows.

- Designing an evidence net based on evidence theory to measure the uncertainty of source domain data about a target domain classification task.
- Designing a general loss function with uncertainty measure for learning of the adaptive classifier.
- Extending the SVM by the general loss function with uncertainty measure for enhancing its transferred performance.

The remainder of the paper is organized as follows. We start by reviewing related works in Section 2. Section 3 describes the evidence net that is built based on evidence theory for estimating the uncertainty. Section 4 extends the general loss function to SVM. Section 5 presents the experimental results to validate the efficiency of the proposed method. The conclusion about our exploratory work is also given in the last section.

2. Related Work

In this section, we discuss previous works on domain adaptation that minimizes the data difference between the source and target domain. In addition, we introduce the evidence theory that is most related to our work.

2.1. Domain Adaptation with Metric Learning

We will briefly introduce the domain adaptation with metric learning. These methods leverage the metric methods to reduce the data difference between two domains.

Maximum mean discrepancy (MMD) [27] takes advantage of the kernel trick, which can measure the data difference between the source domain and target domain. MMD is widely used in domain adaptation. Some state-of-the-art methods are proposed based on MMD. Pan and Yang et al. [12] propose the transfer component analysis (TCA) model based on MMD. The TCA utilizes the MMD to reduce the gap between the source domain and target domain. Long et al. [28] put forward the joint distribution adaptation (JDA) algorithm that uses the MMD to adapt both the marginal distribution and conditional distribution in domain adaptation. Muhammad Ghifary et al. [29] propose a neural network model that embeds the MMD regularization to reduce the distribution mismatch. Long et al. [30] propose a novel framework that is called adaptation regularization-based transfer learning (ARTL). The ARTL optimizes the structural risk functional, joint distribution adaptation of both the marginal, and conditional distributions by embedding the MMD regularization. Yan et al. [31] propose a weighted domain adaptation network (WDAN) by both incorporating the weighted MMD into CNN and taking into account the empirical loss on target samples.

Kullback–Leibler (KL) divergence [32] can measure data distribution differences between the source domain and target domain. Dai et al. [33,34] use the KL divergence to measure the difference between the source domain and target domain and uses the difference in co-clustering to improve the performance of transferring. Zhuang et al. [35] propose a supervised representation learning method based on a deep auto-encoder for domain adaptation. In the embedding layer, the authors use the KL divergence to keep the two distributions of source and target domains similar.

Jensen–Shannon (JS) divergence is similar to KL divergence and measures the difference between the source domain and target domain. However, the JS divergence solves the asymmetry problem of KL divergence. Joshua Giles et al. [36] use JS divergence to compare calibration trails with an electroencephalogram dataset for selecting the target user in domain adaptation. Subhadeep Dey et al. [37] employ JS divergence in Information Bottleneck clustering to find clusters in domain adaptation.

The Wasserstein distance derives from the optimal transport problem. It can be used to measure distances between two probability distributions. Shen et al. [38] reduce the discrepancy between the source domain and target domain by gradient property of the Wasserstein distance for improving transfer performance. Lee et al. [39] use the Wasserstein discrepancy between classifiers to align distributions in domain adaptation.

In summary, the core idea of most methods is to minimize the distribution difference between the source and target domain. However, they ignore the uncertainty between the source domain data and the target domain task.

2.2. Learning with Evidence Theory

Evidence theory can be considered a generalized probability [19,40]. It can represent and measure data uncertainty using mass function [41]. The evidence theory uses Dempster's rule to finish uncertainty reasoning [42]. We will recall mass function and Dempster's rule from evidence theory.

2.2.1. Mass Function

Let $\Omega = \{z_1, z_2, \ldots, z_n\}$ be a finite domain (set) that includes all possible answers to the decision problem, and the elements of the set are mutually exclusive and exhaustive.

Ω is called the frame of discernment. In the classification problems, the element z_k can be regarded as the kth category, and Ω can be considered as the sample space or label space. We denote the power-set as 2^{Ω}, and the cardinality of the power-set is $2^{|\Omega|}$.

The mass function $m(\cdot)$ is the Basic Belief Assignment (BBA) that represents the support degree of evidence, and $m(\cdot)$ is a mapping from 2^{Ω} to the interval $[0, 1]$. It satisfies the condition as follows

$$\begin{cases} \sum_{A \in 2^{\Omega}} m(A) = 1 \\ \\ m(\emptyset) = 0 \end{cases} \tag{1}$$

where $m(A)$ measures the support degree for proposition A itself and $m(\emptyset)$ represents that the empty set has no support degree. If $m(A) > 0$, A is called a focal element. In classification problems, if $A = z_k$, $m(A)$ can be interpreted as a support degree (possible) that instance belongs to class z_k. If $A = \Omega$, $m(A)$ can be interpreted as the total ignorance degree for classification results. In this paper, $m(\Omega)$ can be used to reflect the instance uncertainty.

For example, we assume a classification problem that distinguishes colors. The frame of discernment is $\Omega = \{red, green, blue\}$. The power-set is $2^{\Omega} = \{\emptyset, \{red\}, \{green\}, \{bule\}$, $\{red, green\}, \{red, blue\}$, $\{green, blue\}, \Omega\}$ and $|\Omega| = 3$, $2^{|\Omega|} = 8$. $m(green|x; E)$ represents the possibility that x belong to green based on evidence E. $m(\Omega|x; E)$ represents that we can not determine which class the sample belongs to. It reflects the instance uncertainty.

2.2.2. Dempster's Rule

Dempster's rule reflects the combined effect of evidence. Let m_1 and m_2 be two mass functions induced by independent items of evidence. They can be combined using Dempster's rule to form a new mass function defined as

$$(m_1 \oplus m_2)(A) = \frac{1}{1-\kappa} \sum_{B \cap C = A} m_1(B) m_2(C), \tag{2}$$

for all $A \subseteq \Omega$, $A \neq \emptyset$ and $(m_1 \oplus m_2)(\emptyset) = 0$ ($\oplus$ is the combination operator of Dempster's rule). k is the degree of conflict between m_1 and m_2; it can be defined as

$$\kappa = \sum_{B \cap C = \emptyset} m_1(B) m_2(C). \tag{3}$$

3. Uncertainty Measure in Domain Adaptation Based on Evidence Theory

In domain adaptation, the key problem of the uncertainty measure is how to evaluate the uncertainty in the target domain classification task for each source domain data. We consider that the lower uncertainty of instance represents less information loss in domain adaptation. To achieve this, we construct an evidence net based on evidence theory. It consists of two key steps (1) obtaining a trusty evidence set, and (2) designing the evidence net based on evidence theory. We describe them separately below.

3.1. Obtaining the Trusty Evidence Set

Let us consider a simple scenario with a large number of instances labeled source-domain D^s and a small number of instances labeled target-domain D^t_l.

Given a source-domain instance x^s, its evidence set Φ^t consists of similar instances from the target domain and can be formulated as a neighborhood surrounding x^s.

$$\Phi^t = \{x^t_1, x^t_2, \cdots, x^t_n\}, \tag{4}$$

in which $x^t_1, x^t_2, \cdots, x^t_n$ are n target domain instances similar to the source domain instances x^s and $n > 10$. To ensure the validity of the evidence set, the discrepancy between a source-domain instance and the elements of its evidence set should be small. Motivated by this, we design the objective function of obtaining an evidence set for a source domain instance x^s as

$$\Phi^t = \arg\min_{\Phi} h(x^s, \Phi), \tag{5}$$

in which the function $h(\cdot)$ measures the discrepancy between the x^s of the source domain and the evidence set Φ^t in a reproducing kernel Hilbert Space (RKHS) $\mathcal{H}$, $h(\cdot)$ is formulated as

$$h\left(x^s, \Phi^t\right) = \left\| \phi(x^s) - \frac{1}{|\Phi^t|} \sum_{x^t \in \Phi^t} \phi(x^t) \right\|_{\mathcal{H}}^2, \tag{6}$$

where $\phi : \mathcal{X} \mapsto \mathcal{H}$ is the feature mapping, and $|\Phi^t|$ is the number of elements in the evidence set. In this paper, we utilize the radial basis function kernel to construct the kernel Hilbert space,

$$K\left(x^t, x^s\right) = \phi(x^t)^T \phi(x^s) = \exp\left(-\gamma \left\|x^t - x^s\right\|^2\right), \tag{7}$$

in which $\left\|x^t - x^s\right\|^2$ is the Euclidean distance between two points and γ is a scaling parameter. Substituting $K\left(x^t, x^s\right)$ into Equation (6), the function $h(\cdot)$ can be rewritten as

$$h\left(x^s, \Phi^t\right) = \left| \frac{1}{|\Phi^t|^2} \sum_{x_1^t, x_2^t \in \Phi^t} K\left(x_1^t, x_2^t\right) - \frac{2}{|\Phi^t|} \sum_{x^t \in \Phi^t} K\left(x^s, x^t\right) \right|. \tag{8}$$

Based on the above analysis, the objective function of Equation (5) to obtain the evidence set can be specified as

$$\Phi^t = \arg\min_{\Phi} \left| \frac{1}{|\Phi|^2} \sum_{x_1^t, x_2^t \in \Phi^t} K\left(x_1^t x_2^t\right) - \frac{2}{|\Phi|} \sum_{x^t \in \Phi} K\left(x^s, x^t\right) \right|. \tag{9}$$

The optimal evidence set Φ^t in Equation (9) can be solved by a greedy search on the labeled target domain.

3.2. Constructing Evidence Net Based on Evidence Theory

In the evidence theory, suppose that $m(\cdot|x;\Phi)$ is the mass function, Ω is the label space, and Φ is the evidence set, the mass function $m(\Omega|x;\Phi)$ can represent the uncertainty of x about the classification task. In domain adaptation, Ω comes from the label space of the target domain. In a built-up evidence set Φ^t, from the target domain D^t, for instance, x^s, from source domain D^s, $m(\Omega|x^s;\Phi^t)$ can represent the uncertainty of the source domain instance x^s about the target domain classification task.

In this section, motivated by evidential k-Nearest Neighbor [22] and neural network, we construct an evidence net based on Dempster's rule to calculate $m(\Omega|x^s;\Phi^t)$. The details of the evidence net are described as follows.

According to Section 3.1, the evidence set Φ^t has been generated from the labeled target domain D_l^t. Given k classes, we decompose the evidence set Φ^t into different classes,

$$\Phi^t = \{\Phi_1^t, \Phi_2^t, \ldots, \Phi_k^t\}, \tag{10}$$

where $\Phi_k^t = \{x_{k1}^t, \ldots x_{kl}^t\}$ is the evidence subset in which all the target domain instances have the class label z_k, and x_{kl}^t is the lth element in the evidence subset.

According to the decomposition of the evidence set Φ^t and Dempster's rule, the evidence net can be represented in the connectionist formalism as a network with an input layer, three evidence layers L_1, L_2, and L_3, and an output layer.

As shown in Figure 2, the input layer is an instance of source domain x^s, and the output layer is $m(z_k|x^s;\Phi^t)$ and $m(\Omega|x^s;\Phi^t)$. Each evidence layer $L_i (i = 1, 2, 3)$ corresponds to one step of the procedure described as follows.

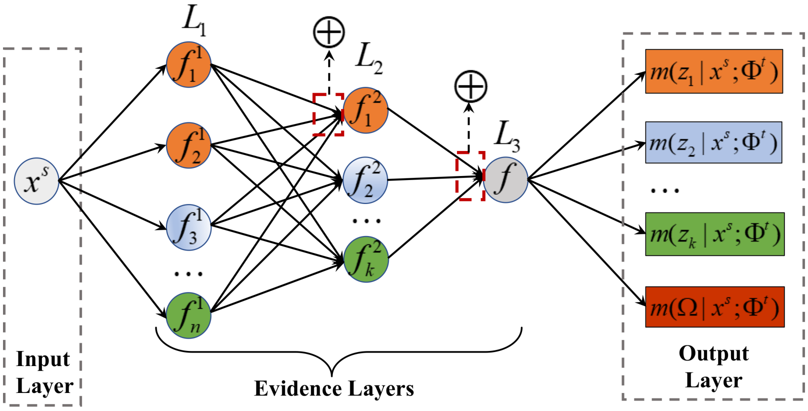


Figure 2. Evidence net architecture.

(1) Layer L_1 contains n nodes, and we denote the node of layer L_1 as $f_i^1(\cdot \mid x^s; x_i^t)$. The input of the node is an instance x^s from source domain D^s. At the fine-grained evidence level, given an element x_i^t in an evidence subset, we compute $f_i^1(\cdot \mid x^s; x_i^t)$ as

$$f_i^1(\cdot \mid x^s; x_i^t) = \begin{cases} m(z_k \mid x^s; x_i^t) \\ m(\Omega \mid x^s; x_i^t). \end{cases} \tag{11}$$

in which

$$\begin{aligned} m(z_k | x^s; x_i^t) &= \exp(-d(x^s, x_i^t)), \\ m(\Omega | x^s; x_i^t) &= 1 - \exp(-d(x^s, x_i^t)), \end{aligned} \tag{12}$$

where $d(\cdot)$ is defined as follows

$$d(x^s, x_i^t) = \mathrm{K}(x^s, x^s) - 2\mathrm{K}(x^s, x_i^t) + \mathrm{K}(x^t, x_i^t), \tag{13}$$

in which $\mathrm{K}(\cdot)$ is the radial basis function kernel.

(2) Layer L_2 contains k nodes, and we denote the node as $f_k^2(\cdot \mid x^s; \Phi_k^t)$. Using Dempster's rule to combine $f_i^1(\cdot \mid x^s; x_i^t)$ under single evidence $x^t \in \Phi_k^t$, we can obtain $f_k^2(\cdot \mid x^s; \Phi_k^t)$ under the evidence subset Φ_k.

$$f_k^2(\cdot \mid x^s; \Phi_k^t) = \bigoplus_{x^t \subseteq \Phi_k^t} f^1(\cdot \mid x^s; x^t) = \begin{cases} m(z_k \mid x^s; \Phi_k^t) \\ m(\Omega \mid x^s; \Phi_k^t), \end{cases} \tag{14}$$

in which

$$\begin{aligned} m(\Omega | x^s; \Phi_k^t) &= \bigoplus_{x^t \in \Phi_k^t} m(\Omega | x^s; x^t) = \prod_{x^t \in \Phi_k^t} m(\Omega | x^s; x^t), \\ m(z_k | x^s; \Phi_k^t) &= \bigoplus_{x^t \in \Phi_k^t} m(z_k | x^s; x^t) = 1 - \prod_{x^t \in \Phi_k^t} m(\Omega | x^s; x^t). \end{aligned} \tag{15}$$

where the orthogonal sum $\oplus$ represents the combination operator of Dempster's rule.

(3) In layer L_3, we denote the node as $f(\cdot \mid x^s; \Phi^t)$. $f(\cdot \mid x^s; \Phi^t)$ can be calculated under the entire evidence set Φ^t through accumulating $f_j^2\left(\cdot \mid x^s; \Phi_j^t\right)$ under evidence subsets.

$$f(\cdot \mid x^s; \Phi^t) = \bigoplus_{\Phi_k \subseteq \Phi^t} f^2(\cdot \mid x^s; \Phi_k^t) = \begin{cases} m(z_k \mid x^s; \Phi^t) \\ m(\Omega \mid x^s; \Phi^t), \end{cases} \tag{16}$$

in which

$$\begin{aligned}
&\mathrm{m}\big(z_k|x^s;\Phi^t\big) = \bigoplus_{\Phi_k^t\subseteq\Phi^t} m\big(z_k|x^s;\Phi_k^t\big) = \frac{1}{\kappa}m\big(z_k|x^s;\Phi_k^t\big)\prod_{j\neq k} m\Big(\Omega|x^s;\Phi_j^t\Big),\\
&\mathrm{m}\big(\Omega|x^s;\Phi_k^t\big) = \bigoplus_{\Phi_k^t\subseteq\Phi} m\big(\Omega|x^s;\Phi_k^t\big) = \frac{1}{\kappa}\prod_{k=1}^{n} m\big(\Omega|x^s;\Phi_k^t\big),\\
&\sum_{k\in\Omega} m\big(z_k \mid x^s;\Phi^t\big) + m\big(\Omega \mid x^s;\Phi^t\big) = 1,
\end{aligned} \tag{17}$$

where κ is a normalizing factor.

$$\kappa = \sum_{k=1}^{n} m\big(z_k|x^s;\Phi_k^t\big)\prod_{j\neq k} m\Big(\Omega|x^s;\Phi_j^t\Big) + \prod_{k=1}^{n} m\big(\Omega|x^s;\Phi_k^t\big). \tag{18}$$

$m(\Omega|x^s;\Phi^t)$ represents the proportion of uncertainty in the target domain classification task for the source domain instance x^s. $m(z_k|x^s;\Phi^t)$ represents the possibility that source domain instance x^s belongs to class z_k of the target domain. In this paper, we use $m(\Omega|x^s;\Phi^t)$ to measure the uncertainty of source domain data about the target domain task. Algorithm 1 summarizes the evidence net-based uncertainty measure of source domain data in domain adaptation.

Algorithm 1 The uncertainty measure based on evidence net for source domain data

```
Input: source domain $D^s$, labeled target domain $D_l^t$.
Output: source domain $D^s$ with uncertainty $m(\Omega|x^s;\Phi^t)$.
for all $x^s \in D^s$ do
    Generate an evidence set $\Phi^t$ for $x^s$ according to Equation (9).
    Estimate uncertainty $m(\Omega|x^s;\Phi^t)$ of $x^s$ based on the evidence net $f(\cdot|x^s;\Phi^t)$.
end for
return $D^s$ with $m(\Omega|x^s;\Phi^t)$.
```

4. Learning Algorithm of Adaptive Classifier with Uncertainty Measure

Section 3 has successfully solved the uncertainty measure of source domain data for target domain tasks. In domain adaptation, another key issue is how to use the uncertainty to learn an adaptive classifier. To solve this problem, we propose a general loss function with an uncertainty measure.

The learning algorithm with uncertainty measure can be transformed into a problem of regularized risk minimization with uncertainty $R[m(\Omega|x^s;\Phi^t), L(x^s,z,w)]$. Thus, the general loss function of the learning algorithm with uncertainty can be written as

$$R[m(\Omega|x^s;\Phi^t), L(x^s,z,w)] = \frac{1}{N}\sum_{i=1}^{N}(1 - m(\Omega|x_i^s;\Phi^t))L(x_i^s,z_i,w) + \lambda||w||, \tag{19}$$

where instance x^s comes from source domain D^s, $L(\cdot)$ is loss function, and w is the parameter of the model. In order to verify its effectiveness, we extend the loss function with an uncertainty measure to support vector machine (SVM).

4.1. Support Vector Machine with Uncertainty Measure (SVMU)

Based on the general loss function, we propose an improved support vector machine with an uncertainty measure (SVMU), which integrates the uncertainty of the source domain instance about the target domain task to SVM. The SVM uses only one penalty factor to control the balance between margin maximization and misclassification. However, in domain adaptation, due to domain differences, the classification hyperplane controlled by only one penalty factor cannot effectively distinguish classes of the target domain. The

SVMU can change the penalty factor by the uncertainty measure. It makes the instances of the source domain that are beneficial to the target domain classification task become the new support vectors and diminishes the importance of some instances that have negative effects. Thus, SVMU is more flexible and superior in domain adaptation than SVM. The details of SVMU are described as follows.

SVM maps the input points into a high-dimensional feature space and finds a separating hyperplane that maximizes the margin between two classes in this space. According to the general loss function, Equation (19), the optimization problem for SVMU is then regarded as the solution to

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(1 - m(\Omega|x_i^s;\Phi^t))\xi_i, \tag{20}$$

subject to

$$\begin{aligned} &z_i(w \cdot \phi(x_i^s) + b) \geq 1 - \xi_i \quad i = 1, 2, \cdots, N, \\ &\xi_i \geq 0 \quad i = 1, 2, \cdots, N, \end{aligned} \tag{21}$$

where parameter ξ_i is the slack variable. $C > 0$ is the penalty factor, which controls the trade-off between the slack variable penalty and the margin. $\phi(\cdot)$ denotes a fixed feature-space transformation. b is the bias parameter.

To solve this optimization problem, we construct the Lagrangian function

$$\begin{aligned} L(w, b, \xi, \sigma, \lambda) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(1 - m(\Omega|x_i^s;\Phi^t))\xi_i \\ - \sum_{i=1}^{N}\sigma_i(z_i(w \cdot \phi(x_i^s) + b) - 1 + \xi_i) - \sum_{i=1}^{N}\lambda_i\xi_i, \end{aligned} \tag{22}$$

To find the saddle point of $L(w, b, \xi, \sigma, \lambda)$, the parameters satisfy the following conditions

$$\begin{aligned} \frac{\partial L(w, b, \xi, \sigma, \lambda)}{\partial \xi_i} &= (1 - m(\Omega|x_i^s;\Phi^t)) * C - \sigma_i - \lambda_i = 0, \\ \frac{\partial L(w, b, \xi, \sigma, \lambda)}{\partial w} &= w - \sum_{i=1}^{N}\sigma_i z_i \phi(x_i^s) = 0, \\ \frac{\partial L(w, b, \xi, \sigma, \lambda)}{\partial b} &= -\sum_{i=1}^{N}\sigma_i z_i = 0. \end{aligned} \tag{23}$$

By applying these conditions to the Lagrangian function (22), problem (20) can be transformed into

$$\min_{w,b,\xi} L(w, b, \xi, \sigma, \lambda) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\sigma_i\sigma_j z_i z_j \mathrm{K}\left(x_i^s, x_j^s\right) + \sum_{i=1}^{N}\sigma_i, \tag{24}$$

subject to

$$\sum_{i=1}^{N}\sigma_i z_i = 0, \quad 0 \leq \sigma_i \leq (1 - m(\Omega|x_i^s;\Phi^t)) * C, \tag{25}$$

where $K(\cdot)$ is a kernel function

$$K(x_i^s, x_j^s) = \phi(x_i^s)^T \cdot \phi(x_j^s), \tag{26}$$

and the KKT conditions are defined as

$$\begin{aligned} &\sigma_i^*(z_i(w^* \cdot \phi(x_i^s) + b^*) - 1 + \xi_i^*) = 0, \\ &((1 - m(\Omega|x_i^s;\Phi^t)) * C - \sigma_i^*)\xi_i^* = 0, \end{aligned} \tag{27}$$

The optimal solution of (24) can be denoted as $\sigma^* = (\sigma_1^*, \sigma_2^*, \cdot, \sigma_N^*)$, where x_i^s corresponding to $\sigma_i^* > 0$ is a support vector. The support vector x_i^s falls exactly on the margin boundary if $0 < \sigma_i^* < (1 - m(\Omega|x_i^s;\Phi^t)) * C$. If $\sigma_i^* = (1 - m(\Omega|x_i^s;\Phi^t)) * C$, $0 < \xi_i < 1$, then the classification is correct, and x_i^s is between the boundary and the hyperplane. If $\alpha_i^* = (1 - m(\Omega|x_i^s;\Phi^t)) * C$ and $\xi_i = 1$, then x_i^s is on the classification hyperplane; if $\alpha_i^* = (1 - m(\Omega|x_i^s;\Phi^t)) * C$ and $\xi_i > 1$, then x_i^s is on the misclassified side of the classification hyperplane.

In the traditional SVM, the only penalty factor C controls the balance between margin maximization and misclassification. A larger C allows the SVM to have fewer misclassification and a narrower margin. Conversely, a smaller C makes the SVM ignore more training points and obtains a larger margin. Due to the existing uncertainty of the source domain data about the target domain task, with only one penalty factor, it is difficult to control the balance between margin maximization and misclassification in the target domain task. This may result in negative transfer when using SVM as the classifier.

Based on the above analysis, applying uncertainty to SVM, it can be found that the single penalty factor C becomes $(1 - m(\Omega|x_i^s;\Phi^t)) * C$, whose number of penalty factors increases from one to the number of source domain instances. Each support vector corresponds to a penalty factor $(1 - m(\Omega|x_i^s;\Phi^t)) * C$ with an uncertainty measure instead of corresponding to a single constant value C. Thus, the selection of support vectors does not rely on a single penalty factor but is determined by the uncertainty $m(\Omega|x_i^s;\Phi^t)$ of each source domain instance with respect to the target domain task. As shown in Figure 3, changing the penalty factor C by uncertainty $m(\Omega|x_i^s;\Phi^t)$ can make the instances of the source domain that are beneficial to the target domain classification task become the new support vectors and diminish the importance of some instances that have negative effects. The classification hyperplane that is generated by these new support vectors is suited to discriminate the target data. Thus, integrating the uncertainty to SVM can adjust the classification hyperplane to suit the target domain task.

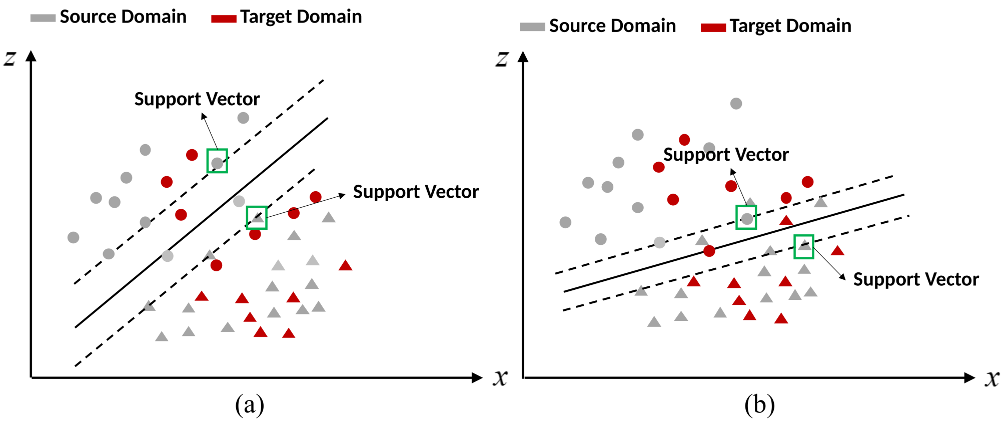


Figure 3. Schematic diagram: (**a**) Classification hyperplane is generated by SVM. (**b**) Classification hyperplane is generated by SVM with uncertainty.

5. Experiments

In the experiments, we evaluate the adaptive classifier with an uncertainty measure on various kinds of data, including texts and images. The descriptions of the datasets are listed below.

Amazon product reviews dataset [43] is the benchmark text corpora widely used for domain adaptation evaluation. The reviews are about four product domains: books (denoted as B), dvds (denoted as D), electronics (denoted as E), and kitchen appliances (denoted as K). Each review is assigned to a sentiment label, -1 (negative review) or +1 (positive review), based on the rating score given by the review author. In each domain, there are 1000 positive reviews and 1000 negative reviews. In this dataset, we construct 12 cross-domain sentiment classification tasks: $B \to D$, $B \to E$, $B \to K$, $D \to B$, $D \to E$,

$D \to K$, $E \to B$, $E \to D$, $E \to K$, $K \to B$, $K \to D$, $K \to E$, where the word before an arrow corresponds to the source domain and the word after an arrow corresponds with the target domain. In each cross-domain classification task, we extract the features of the texts by using the word2vec tool.

Office+Caltech dataset [44] is commonly used for the task of visual object recognition in domain adaptation. It includes four domains: Amazon (denoted as A, images downloaded from online merchants), Webcam (denoted as W, low-resolution images from a web camera), DSLR (denoted as D, high-resolution images from a digital SLR camera), and Caltech-256 (denoted as C). The dataset includes 10 classes: backpack, touring bike, calculator, head Caltech, phones, computer keyboard, laptop-101, computer monitor, computer mouse, coffee mug, and video projector. There are 8 to 151 samples per category per domain and 2533 images in total. In this dataset, we construct 12 cross-domain multi-classification tasks: $A \to C$, $A \to D$, $A \to W$, $C \to A$, $C \to D$, $C \to W$, $D \to A$, $D \to C$, $D \to W$, $W \to A$, $W \to C$, and $W \to D$.

In the experiment, for each domain adaptation classification task, we use the classification accuracy of the target domain as the evaluation criterion. Suppose D^t is the target domain dataset,

$$\text{Accuracy} \ = \frac{|\{x : x \in D^t \wedge v(x) = y\}|}{|\{x : x \in \mathcal{X}\}|}, \tag{28}$$

where y is the ground truth label of x, and $v(x)$ is the label predicted by the classifier.

5.1. Comparative Studies

To evaluate the transfer performance of SVM with the uncertainty measure (SVMU), we compared it with 9 domain adaptation methods on Amazon product reviews dataset and Office+Caltech datasets, respectively. The methods of comparison include transfer component analysis (TCA) [12], correlation alignment (CORAL) [13], geodesic flow kernel (GFK) [44], joint distribution adaptation (JDA) [28], kernel mean matching (KMM) [45], metric transfer learning (MTLF) [46], scatter component analysis (SCA) [14], practically easy transfer learning (EasyTL) [47], and Wasserstein distance-guided representation learning (WDGAL) [38].

(1) Testing on Amazon product reviews dataset

In this testing, we evaluate SVM with an uncertainty measure (SVMU) on the Amazon product reviews dataset. The classification accuracies of the comparative study are listed in Table 1.

Table 1. Cross-domain sentiment classification accuracies of Amazon product reviews generated by SVMU and baseline methods.

Task	SVMU	TCA	CORAL	GFK	JDA	KMM	MTLF	SCA	EasyTL	WDGAL
$B \to D$	**84.61**	77.76	70.76	75.76	77.26	83.76	68.59	81.56	79.80	83.05
$B \to E$	**81.01**	75.54	66.21	72.00	75.93	79.02	69.63	78.08	79.70	80.09
$B \to K$	81.92	78.74	70.00	73.50	78.09	75.90	72.74	79.09	80.90	**85.45**
$D \to B$	82.11	76.05	73.05	71.85	77.65	80.50	70.70	**82.35**	79.90	80.72
$D \to E$	**82.84**	76.38	68.70	68.96	76.03	68.51	71.90	78.82	80.80	82.26
$D \to K$	82.64	79.34	71.96	75.70	78.29	76.45	74.18	80.39	82.00	**85.23**
$E \to B$	**79.44**	73.35	69.90	72.60	72.65	73.70	69.20	77.00	75.00	77.22
$E \to D$	**82.79**	73.66	65.71	71.11	72.16	77.86	70.73	77.26	75.30	78.28
$E \to K$	86.40	79.74	72.35	76.20	80.14	80.39	71.36	84.63	84.90	**88.16**
$K \to B$	**81.11**	73.05	67.45	73.75	75.05	74.25	66.04	78.90	76.50	77.16
$K \to D$	**82.12**	77.26	68.61	74.21	77.56	75.96	70.31	77.46	76.30	78.89
$K \to E$	**86.61**	78.74	75.68	76.58	80.32	85.00	68.58	85.65	82.50	86.29
Average	**82.80**	76.63	70.03	73.52	76.76	77.61	70.33	80.10	79.47	81.90

As shown in Table 1, the average classification accuracy of SVMU on the 12 tasks is 82.80%. The performance improvement is 6.17%, 12.77%, 9.28%, 6.04%, 5.19%, 12.47%, 2.70%, 3.33%, and 0.9% compared to baseline method. The average classification accuracy of TCA, CORAL, GFK, JDA, and KMM are 76.63%, 70.03%, 73.52%, 76.76%, and 77.61% on Amazon product reviews. These methods aim to minimize the different between the source and target domains, while ignoring the uncertainty of the instances in the source domain with respect to the task. Although they can find a representation space with the greatest commonality between the source and target domains, they cannot determine whether the source domain instance is suitable for the target domain task. This limits the performance of these methods. The performance improvement of our method is 6.17%, 12.77%, 9.28%, 6.04%, and 5.19% compared to them. In results of text classification, since these results are obtained from a larger number of datasets, it can convincingly verify that SVMU is reliable and effective for classifying cross-domain text accurately.

(2) Testing on Office+Caltech datasets

In this testing, we evaluate SVM with an uncertainty measure (SVMU) on Office+Caltech datasets. In each cross-domain classification task, we extract the features of images by speeded up robust features (SURF). The classification accuracies of the comparative study are listed in Table 2.

Table 2. Cross-domain classification accuracies on Office+Caltech image datasets (SURF features) generated by SVMU and baseline methods.

Task	SVMU	TCA	CORAL	GFK	JDA	KMM	MTLF	SCA	EasyTL
$A \rightarrow C$	**51.55**	47.76	45.37	40.25	49.36	45.41	45.37	48.29	43.01
$A \rightarrow D$	44.31	41.12	43.75	43.31	42.49	41.40	41.38	44.21	**45.85**
$A \rightarrow W$	**47.28**	44.63	44.78	43.98	45.97	42.85	42.59	43.90	40.68
$C \rightarrow A$	**63.29**	58.20	53.59	51.20	54.78	50.10	54.17	53.74	50.10
$C \rightarrow D$	44.00	41.40	46.22	42.85	43.22	43.58	40.69	39.49	**48.41**
$C \rightarrow W$	**46.44**	42.64	43.73	40.68	41.69	43.81	46.10	43.56	42.49
$D \rightarrow A$	**63.29**	52.15	58.81	52.05	53.09	58.60	59.92	57.72	61.94
$D \rightarrow C$	**51.33**	49.70	48.01	48.28	45.52	47.81	45.73	50.32	51.17
$D \rightarrow W$	**46.44**	46.10	44.40	45.59	43.49	44.45	43.50	42.81	44.49
$W \rightarrow A$	**63.29**	58.06	56.20	59.75	56.78	52.15	51.07	60.48	60.18
$W \rightarrow C$	**51.22**	45.30	42.08	48.72	49.17	49.81	49.38	50.63	49.65
$W \rightarrow D$	**47.77**	43.26	44.08	40.89	46.17	45.62	44.76	46.36	47.07
Average	**51.68**	47.52	47.58	46.46	47.64	47.13	47.05	48.45	48.75

It is obvious that SVMU achieves better performance than the methods of comparison on Office+Caltech datasets. Specifically, the average classification accuracy of SVMU on 12 cross-domain classification tasks is 50.60%, which gains significant performance improvements of 4.16%, 4.1%, 5.22%, 4.04%, 4.55%, 4.63%, 3.23%, and 2.93% compared to the baseline methods. The experimental results reveal that the improved SVM with the uncertainty measure is reliable and effective in cross-domain image classification tasks.

5.2. *Effectiveness Verification of Uncertainty Measure*

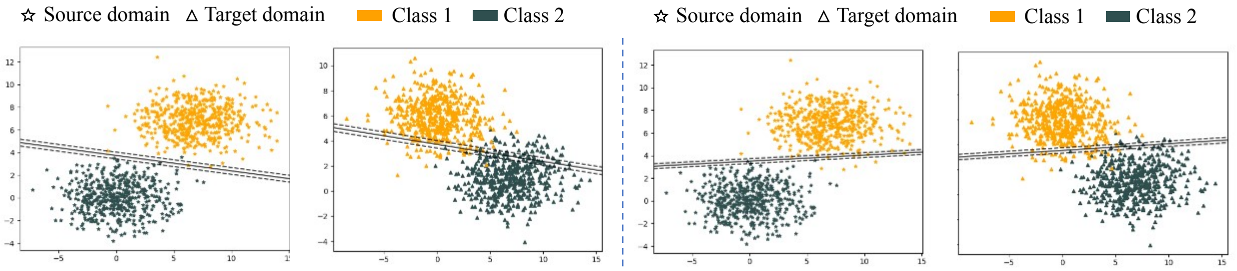
In this experiment, we verify the effectiveness of the uncertainty measure from three views: (1) Testing on synthetic data, visualizing the classification hyperplane of an adaptive classifier with and without an uncertainty measure. (2) Testing on real-world datasets, comparing the performance of SVM with and without an uncertainty measure. (3) Case study, explaining the role of uncertainty measure.

5.2.1. Testing on Synthetic Data

In order to demonstrate the effectiveness of the adaptive classifier with an uncertainty measure in domain adaptation, we visualize the classification hyperplane of an adaptive classifier on a synthetic dataset. The synthetic dataset is generated from a Gaussian distribution $x \sim \mathcal{N}(\mu, \sigma)$, where μ and σ are the mean and standard deviation, respectively. We apply different μ and σ to generate the data from the source domain and target domain.

In the dataset, the source domain and target domain consist of two-dimensional data points under two classes, and each class has 500 data points. The source domain is marked by a pentagram, and the target domain is marked by a triangle. Class 1 is marked in orange, and Class 2 is marked in dark slate-gray.

In Figure 4, (a) and (b) show the classification hyperplanes that are generated based on the source domain by SVM and SVMU, respectively. Due to the difference in data distribution between the source and target domains, the classification hyperplane generated by SVM cannot accurately distinguish the categories of the target domain and cannot satisfy the domain adaptation task. In contrast, the classification hyperplane generated by SVMU can accurately classify the target domain categories, and the classification results are shown in (a) and (b). The experimental results are consistent with the conclusions about SVMU in Section 4.1. Therefore, the uncertainty measure is effective and can improve the transfer performance of the adaptive classifier.



(a)

(b)

Figure 4. Results on synthetic data: (**a**) Classification hyperplane is generated by SVM. (**b**) Classification hyperplane is generated by SVM with uncertainty.

5.2.2. Testing on Real-World Datasets

To further explain the effectiveness of the adaptive classifier with uncertainty, we compare the SVM with and without uncertainty on the Amazon product reviews dataset.

Figure 5 shows the result of SVM with and without uncertainty on the Amazon product reviews dataset; it is obvious that in all the cross-domain text tasks, SVMU achieves better performance than SVM. SVMU improves the transfer accuracy over SVM on the 12 subtasks by 11.71%, 6.62%, 8.28%, 8.06%, 9.46%, 12.1%, 5.24%, 9.89%, 14.82%, 6.56%, 11.27%, and 12.48%. Comparing the average classification accuracy, SVMU improves the average classification accuracy by 9.71% over SVM. The above results show that it is effective at enhancing the transfer performance of the adaptive classifier by introducing the uncertainty between the source domain data and target domain task.

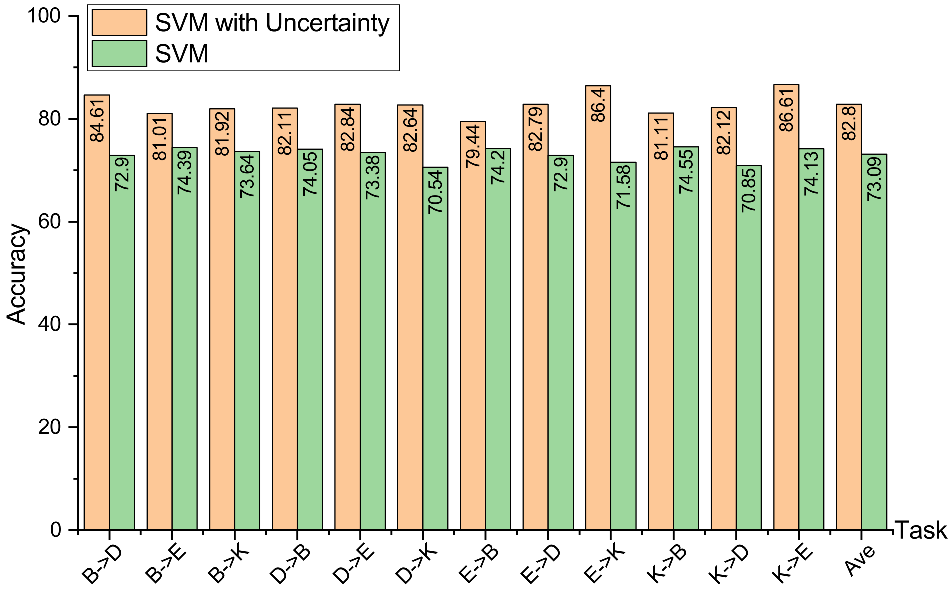


Figure 5. Cross-domain sentiment classification accuracies on Amazon product reviews generated by SVM with and without uncertainty.

5.2.3. Case Study

Based on the above sub-experiments, it can be verified that the uncertainty measure is able to enhance adaptive classifier transfer performance. To explain the role of the uncertainty measure on the transfer process for an adaptive classifier, we use the Caltech-256 image data (complex background) as the source domain and the Amazon image data (no background) as the target domain. When the Caltech-256 dataset transfers to the Amazon dataset, the uncertainty values of some instances in the backpack and bicycle categories in the Caltech-256 dataset are shown.

As shown in Figure 6, for images (a1) to (a6) in the Caltech-256 dataset, it can be found that (a1) and (a2) are cartoon images of a backpack, and (a5) and (a6) are bicycles with obscure features. These instances are not significantly helpful for the target domain classification task. On the contrary, in (a3) and (a4), the features of the backpack and bicycle are obvious and beneficial for the target domain classification task.

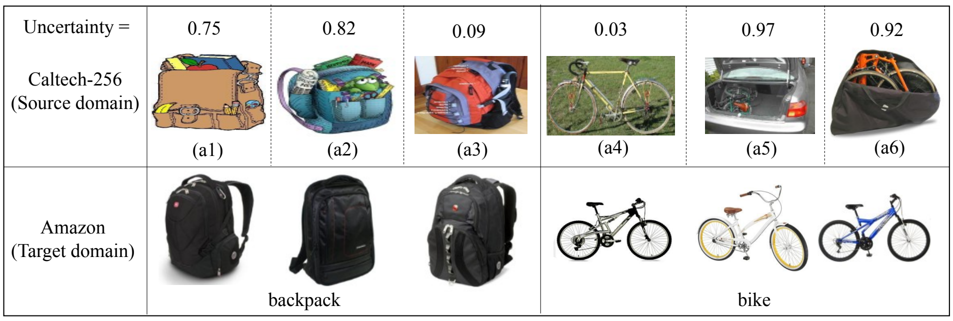


Figure 6. The uncertainty of the category ’backpack’ and ’bike’ in source domain *C* about the target domain *A* classification task.

We use the evidence net to calculate the uncertainty between (a1)–(a6) and the target domain task; as shown in Figure 6, we can find that the uncertainties of (a1), (a2), (a5), and (a6) calculated by the evidence network are high; 0.75, 0.82, 0.97, and 0.92, respectively. The uncertainties of (a3) and (a4) are low, at 0.09 and 0.03, respectively. When the Caltech-256 dataset transfers to the Amazon dataset, the images (a1)–(a6) no longer fully belong to the category of backpack and bicycle. (a1), (a2), and (a3) belong to the backpack category with the possibilities 0.25, 0.18, and 0.91, and 0.75, 0.82, and 0.09 are the uncertainties. (a4), (a5), and (a6) belong to the bicycle category with the possibilities 0.97, 0.03, and 0.08, and 0.03, 0.97, and 0.92 are the uncertainties. Based on the above results, it can be found that our proposed uncertainty measure is consistent with people's cognition. Therefore, the uncertainty can accurately measure the adaptability of instances with respect to the target domain task.

6. Conclusions

In this article, based on evidence theory, we revisited the domain adaptation from source domain data uncertainty and thereby devised a reliable adaptive classifier with the uncertainty measure. Specifically, for solving the uncertainty measure between the source domain data and target domain tasks, we designed an evidence net based on evidence theory. To solve the problem of model learning with a data uncertainty measure, we proposed a general loss function with an uncertainty measure for an adaptive classifier and extended the loss function to support vector machine. Experiments on the text dataset and image dataset validate that the proposed uncertainty measure is effective at improving the transfer performance of an adaptive classifier. In the future, we plan to extend the classifier with the uncertainty measure to handle the domain adaptation with multiple source domains and the domain adaptation on open sets.

Author Contributions: Conceptualization, Y.L. and B.Z.; Data curation, Z.X. and H.L.; Methodology, Y.L. and G.Z.; Writing—original draft, Y.L.; Writing-review and editing, B.Z. and X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key R&D Program of China grant number 2017YFC09 07505.

Data Availability Statement: Not applicable.

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

References

1. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [CrossRef]
2. Lu, J.; Behbood, V.; Hao, P.; Zuo, H.; Xue, S.; Zhang, G. Transfer learning using computational intelligence: A survey. *Knowl.-Based Syst.* **2015**, *80*, 14–23. [CrossRef]
3. Zhang, L. Transfer adaptation learning: A decade survey. *arXiv* **2019**, arXiv:1903.04687.
4. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A comprehensive survey on transfer learning. *Proc. IEEE* **2020**, *109*, 43–76. [CrossRef]
5. Chen, Y.; Li, W.; Sakaridis, C.; Dai, D.; Van Gool, L. Domain adaptive faster r-cnn for object detection in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3339–3348.
6. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 379–387.
7. Ye, H.; Tan, Q.; He, R.; Li, J.; Ng, H.T.; Bing, L. Feature adaptation of pre-trained language models across languages and domains for text classification. *arXiv* **2020**, arXiv:2009.11538.
8. Guo, H.; Pasunuru, R.; Bansal, M. Multi-Source Domain Adaptation for Text Classification via DistanceNet-Bandits. In Proceedings of the AAAI, New York, NY, USA, 7–12 February 2020; pp. 7830–7838.
9. Apostolopoulos, I.D.; Mpesiana, T.A. COVID-19: Automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. *Phys. Eng. Sci. Med.* **2020**, *43*, 635–640. [CrossRef]
10. Raghu, M.; Zhang, C.; Kleinberg, J.; Bengio, S. Transfusion: Understanding transfer learning for medical imaging. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 3347–3357.
11. Zhao, H.; Hu, J.; Risteski, A. On learning language-invariant representations for universal machine translation. *arXiv* **2020**, arXiv:2008.04510.
12. Pan, S.J.; Tsang, I.W.; Kwok, J.T.; Yang, Q. Domain adaptation via transfer component analysis. *IEEE Trans. Neural Netw.* **2010**, *22*, 199–210. [CrossRef]
13. Sun, B.; Feng, J.; Saenko, K. Return of frustratingly easy domain adaptation. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
14. Ghifary, M.; Balduzzi, D.; Kleijn, W.B.; Zhang, M. Scatter Component Analysis: A Unified Framework for Domain Adaptation and Domain Generalization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1414–1430. [CrossRef]
15. Long, M.; Zhu, H.; Wang, J.; Jordan, M.I. Deep transfer learning with joint adaptation networks. In Proceedings of the International Conference on Machine Learning. PMLR, Sydney, Australia, 6–11 August 2017; pp. 2208–2217.

16. Wang, J.; Feng, W.; Chen, Y.; Yu, H.; Huang, M.; Yu, P.S. Visual domain adaptation with manifold embedded distribution alignment. In Proceedings of the 26th ACM international conference on Multimedia, Seoul, Korea, 22–26 October 2018; pp. 402–410.
17. Zhu, Y.; Zhuang, F.; Wang, J.; Chen, J.; Shi, Z.; Wu, W.; He, Q. Multi-representation adaptation network for cross-domain image classification. *Neural Netw.* **2019**, *119*, 214–221. [CrossRef]
18. Bielza, C.; Larranaga, P. Discrete Bayesian network classifiers: A survey. *ACM Comput. Surv. (CSUR)* **2014**, *47*, 1–43. [CrossRef]
19. Shafer, G. A mathematical theory of evidence turns 40. *Int. J. Approx. Reason.* **2016**, *79*, 7–25. [CrossRef]
20. Principe, J.C.; Xu, D.; Fisher, J.; Haykin, S. Information theoretic learning. *Unsupervised Adapt. Filter.* **2000**, *1*, 265–319.
21. Zadeh, L.A.; Klir, G.J.; Yuan, B. *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers*; World Scientific: Singapore, 1996; Volume 6.
22. Denoeux, T. A k-nearest neighbor classification rule based on Dempster-Shafer theory. In *Classic Works of the Dempster-Shafer Theory of Belief Functions*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 737–760.
23. Su, Z.; Hu, Q.; Denaeux, T. A distributed rough evidential K-NN classifier: Integrating feature reduction and classification. *IEEE Trans. Fuzzy Syst.* **2020**, *29*, 2322–2335. [CrossRef]
24. Quost, B.; Denœux, T.; Li, S. Parametric classification with soft labels using the evidential EM algorithm: Linear discriminant analysis versus logistic regression. *Adv. Data Anal. Classif.* **2017**, *11*, 659–690. [CrossRef]
25. Denoeux, T. Logistic regression, neural networks and Dempster–Shafer theory: A new perspective. *Knowl.-Based Syst.* **2019**, *176*, 54–67. [CrossRef]
26. Denoeux, T.; Sriboonchitta, S.; Kanjanatarakul, O. Evidential clustering of large dissimilarity data. *Knowl.-Based Syst.* **2016**, *106*, 179–195. [CrossRef]
27. Borgwardt, K.M.; Gretton, A.; Rasch, M.J.; Kriegel, H.P.; Schölkopf, B.; Smola, A.J. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics* **2006**, *22*, e49–e57. [CrossRef]
28. Long, M.; Wang, J.; Ding, G.; Sun, J.; Yu, P.S. Transfer feature learning with joint distribution adaptation. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013 ; pp. 2200–2207.
29. Ghifary, M.; Kleijn, W.B.; Zhang, M. Domain adaptive neural networks for object recognition. In Proceedings of the Pacific Rim International Conference on Artificial Intelligence, Gold Coast, QLD, Australia, 1–5 December 2014; pp. 898–904.
30. Long, M.; Wang, J.; Ding, G.; Pan, S.J.; Philip, S.Y. Adaptation regularization: A general framework for transfer learning. *IEEE Trans. Knowl. Data Eng.* **2013**, *26*, 1076–1089. [CrossRef]
31. 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 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2272–2281.
32. Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [CrossRef]
33. Dai, W.; Xue, G.R.; Yang, Q.; Yu, Y. Co-clustering based classification for out-of-domain documents. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, VA, USA, 2007; pp. 210–219.
34. Dai, W.; Yang, Q.; Xue, G.R.; Yu, Y. Self-taught clustering. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 200–207.
35. Zhuang, F.; Cheng, X.; Luo, P.; Pan, S.J.; He, Q. Supervised representation learning: Transfer learning with deep autoencoders. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
36. Giles, J.; Ang, K.K.; Mihaylova, L.S.; Arvaneh, M. A Subject-to-subject Transfer Learning Framework Based on Jensen-shannon Divergence for Improving Brain-computer Interface. In Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 3087–3091.
37. Dey, S.; Madikeri, S.; Motlicek, P. Information theoretic clustering for unsupervised domain-adaptation. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2016; pp. 5580–5584.
38. Shen, J.; Qu, Y.; Zhang, W.; Yu, Y. Wasserstein Distance Guided Representation Learning for Domain Adaptation. In Proceedings of the AAAI, New Orleans, LA, USA, 2–7 February 2018.
39. Lee, C.Y.; Batra, T.; Baig, M.H.; Ulbricht, D. Sliced wasserstein discrepancy for unsupervised domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 10285–10295.
40. Dempster, A.P. Upper and lower probabilities generated by a random closed interval. *Ann. Math. Stat.* **1968**, *39*, 957–966. [CrossRef]
41. Walley, P. Belief function representations of statistical evidence. *Ann. Stat.* **1987**, *15*, 1439–1465. [CrossRef]
42. Denœux, T. Reasoning with imprecise belief structures. *Int. J. Approx. Reason.* **1999**, *20*, 79–111. [CrossRef]
43. Blitzer, J.; Dredze, M.; Pereira, F. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, 23–30 June 2007; pp. 440–447.
44. Gong, B.; Shi, Y.; Sha, F.; Grauman, K. Geodesic flow kernel for unsupervised domain adaptation. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 2066–2073.
45. Huang, J.; Gretton, A.; Borgwardt, K.M.; Schölkopf, B.; Smola, A.J. Correcting Sample Selection Bias by Unlabeled Data. *Adv. Neural Inf. Process. Syst.* **2006**, *19*, 601–608.

46. Xu, Y.; Pan, S.J.; Xiong, H.; Wu, Q.; Luo, R.; Min, H.; Song, H. A Unified Framework for Metric Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 1158–1171. [CrossRef]
47. Wang, J.; Chen, Y.; Yu, H.; Huang, M.; Yang, Q. Easy Transfer Learning By Exploiting Intra-Domain Structures. In Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME), Shanghai, China, 8–12 July 2019; pp. 1210–1215.