

Evaluating Stacked Marginalised Denoising Autoencoders Within Domain Adaptation Methods

Boris Chidlovskii^(✉), Gabriela Csurka, and Stephane Clinchant

Xerox Research Centre Europe, 6 Chemin Maupertuis, Meylan, France
chidlovskii@xrce.xerox.com

Abstract. In this paper we address the problem of domain adaptation using multiple source domains. We extend the XRCE contribution to Clef’14 Domain Adaptation challenge [6] with the new methods and new datasets. We describe a new class of domain adaptation technique based on *stacked marginalized denoising autoencoders* (sMDA). It aims at extracting and denoising features common to both source and target domains in the unsupervised mode. Noise marginalization allows to obtain a closed form solution and to considerably reduce the training time. We build a classification system which compares sMDA combined with SVM or with Domain Specific Class Mean classifiers to the state-of-the-art in both unsupervised and semi-supervised settings. We report the evaluation results for a number of image and text datasets.

1 Introduction

Domain adaptation problem rises each time when we need to leverage labeled data in one or more related domains, hereafter referred to as *source* domains, to learn a classifier for unseen data in a *target* domain. Such a situation occurs in multiple real world applications with embedded machine learning components. Examples include named entity recognition across different text corpora, object recognition in images acquired in different conditions, and some others (see [18] for a survey on domain adaptation methods).

Domain adaptation has received a particular attention in computer vision applications [19, 22, 23] where domain shift is a consequence of taking images in different conditions (background scene, object location and pose, view angle changes) [24]. A large number of very different approaches have been proposed in the last few years to address the visual domain adaptation [3, 14–17, 21]. Due to this high interest, ImageCLEF 2014 organized the Domain Adaptation Challenge on multi-source domain adaptation for the image classification. XRCE team participated and won the challenge [6], by combining techniques of the instance reuse and metric learning.

In this paper, we extend our last year contribution in three ways. First, we lean on new methods for domain adaptation, in particular, ones based on *stacked marginalized denoising autoencoders* (sMDAs) [5, 26] developed in the

deep learning community. These methods aim at extracting features common to both source and target domains, by corrupting feature values and then marginalizing the noise out. Second, we extend the semi-supervised classification task to a more challenging unsupervised mode, where no target labeled instances are available. Finally, we include in the evaluation new, both image and text datasets.

The remainder of the paper is organized as follows. In Section 2 we introduce the problem of domain adaptation from multiple sources to a target domain. We recall the previous methods and describe in details sMDA. Section 3 describes datasets used in evaluation. Section 4 is a core part of the paper, it reports results of multiple comparative evaluations. Section 5 concludes the paper.

2 Domain Adaptation Problem and Methods

We define a domain \mathcal{D} as composed of a feature space $\mathcal{X} \subset R^d$ and a label space \mathcal{Y} . Any given task in domain \mathcal{D} (classification, regression, ranking, etc.) is defined by function $h : \mathcal{X} \rightarrow \mathcal{Y}$. In traditional machine learning, learning the task is to estimate a classifier function $\hat{h} : \mathcal{X} \rightarrow \mathcal{Y}$ from the sample data $D = \{(\mathbf{x}_1; y_1), \dots, (\mathbf{x}_n; y_n)\}$, $\mathbf{x}_i \in \mathcal{X}$; $y_i \in \mathcal{Y}$, that best approximates h , according to certain criteria.

In the domain adaptation setting, we assume working with $N + 1$ domains, including N *source* domains S_j and a *target* domain T . From the source domain $S_j, j = 1 \dots, N$, we can sample data with labels, $D_{S_j} = \{(\mathbf{x}_{j1}, y_{j1}), \dots, (\mathbf{x}_{jn_j}, y_{jn_j})\}$, $\mathbf{x}_{ji} \in \mathcal{X}$, $y_{ji} \in \mathcal{Y}$. From the target domain, we are able to sample data $D_T = \{\mathbf{x}_1, \dots, \mathbf{x}_{n_T}\}$, $\mathbf{x}_i \in \mathcal{X}$. In the unsupervised case, data is sampled without labels; in the semi-supervised setting, initial $r_T \ll n_T$ items in D_T have labels $\{y_1, \dots, y_{r_T}\}$. The domain adaptation goal is then to learn a classifier $h_T : \mathcal{X} \rightarrow \mathcal{Y}$ with the help of the labeled sets D_{S_j} and the (mostly) unlabeled set D_T , to accurately predict the labels of data from the target domain T .

In [6] we addressed the domain adaptation problem by techniques which either selectively reuse source domain instances for target domains, or transform both target and source domains in one common space. Here we extend our previous results with a new class of methods based on stacked marginalized denoising autoencoders (sMDAs) [5, 26], described in the following section.

2.1 Stacked Marginalized Denoising Autoencoders

A denoising autoencoder (DA) is one-layer neural network trained to reconstruct input data from partial random corruption [25]. The denoisers can be stacked into multi-layered architectures (sDAs) where the weights are fine-tuned with back-propagation. Alternatively, the outputs of intermediate layers can be used as input features to other learning algorithms. These learned feature representations are known to improve classification accuracy in many cases. For example, Glorot et. al.[13] applied sDAs to domain adaptation and demonstrated that

these learned features, when used with a simple linear SVM classifier, yield record performance in benchmark sentiment analysis tasks.

The main downside of sDAs is a long training time, which often entails specialized computing supports such as GPUs, especially for large-scale tasks. To address this problem, a variation of sDA was proposed [5], in which the random corruption is marginalized out. This crucial step yields the optimal reconstruction weights computed in closed-form and eliminates the use of back-propagation for tuning. Features learned with this approach lead to classification accuracy comparable with sDAs [5, 26], with a remarkable reduction of the training time.

The basic building block is a one-layer linear denoising autoencoder. From a given set of inputs D , we sample inputs $\mathbf{x}_1, \dots, \mathbf{x}_m$. These inputs are corrupted by random feature removal, when each feature is set to 0 with probability p ; the corrupted version of \mathbf{x}_i is denoted as $\tilde{\mathbf{x}}_i$. Then, the corrupted inputs are reconstructed with a linear mapping $\mathbf{W} : R^d \rightarrow R^d$, that minimizes the squared reconstruction loss

$$\mathcal{L}(\mathbf{W}) = \frac{1}{m} \sum_{i=1}^m \|\mathbf{x}_i - \mathbf{W}\tilde{\mathbf{x}}_i\|^2. \quad (1)$$

The constant feature can be added to the input, $\mathbf{x}_i = [\mathbf{x}_i; 1]$, and an appropriate bias is incorporated within the mapping $\mathbf{W} = [\mathbf{W}; b]$. Note that the constant feature is never corrupted. Inputs design the matrix $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]$ and its corrupted version is denoted by $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_m]$. Then, the solution of (1) can be expressed as the closed-form solution for ordinary least squares

$$\mathbf{W} = \mathbf{P}\mathbf{Q}^{-1}, \quad \text{where} \quad \mathbf{Q} = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^T \quad \text{and} \quad \mathbf{P} = \mathbf{X}\tilde{\mathbf{X}}^T. \quad (2)$$

The solution to (2) depends on the sample inputs $\mathbf{x}_1, \dots, \mathbf{x}_m$ and which features are randomly corrupted. Ideally, it is preferable to consider all possible corruptions of all possible inputs when the denoising transformation \mathbf{W} is computed, i.e. letting $m \rightarrow \infty$. By the weak law of large numbers, the matrices \mathbf{P} and \mathbf{Q} converge to their expected values $\mathbb{E}[\mathbf{Q}], \mathbb{E}[\mathbf{P}]^{-1}$ as more copies of the corrupted data are created. In the limit, one can derive their expectations and express the corresponding mapping for \mathbf{W} in closed form as $\mathbf{W} = \mathbb{E}[\mathbf{P}] \cdot \mathbb{E}[\mathbf{Q}]$, where

$$\mathbb{E}[\mathbf{Q}]_{ij} = \begin{cases} \mathbf{S}_{ij}q_iq_j, & \text{if } i \neq j, \\ \mathbf{S}_{ij}q_i, & \text{if } i = j, \end{cases} \quad \text{and} \quad \mathbb{E}[\mathbf{P}]_{ij} = \mathbf{S}_{ij}q_j, \quad (3)$$

with $q = [1 - p, \dots, 1 - p, 1] \in R^{d+1}$ for the noise level p , and $\mathbf{S} = \mathbf{X}\mathbf{X}^T$ being the covariance matrix of the uncorrupted data \mathbf{X} . This closed-form denoising layer is denoted as *Marginalized Denoising Autoencoder* (MDA).

In the case of sDAs, the key component of their success consists in multiple stacked layers of denoising autoencoders, which create a *deep learning* architecture. Several MDA layers can also be stacked together by feeding the representations of the t -th denoising layer as the input to the $(t + 1)$ -th layer. Each transformation \mathbf{W}^t is learned to reconstruct the previous MDA output \mathbf{h}_t from

its corrupted equivalent. In order to extend our mapping beyond a linear transformation, a non-linear function between layers is applied. Each layer’s representation is obtained from its preceding layer through a non-linear transformation $\mathbf{h}_t = \tanh(\mathbf{W}^t \mathbf{h}_{t-1})$, with $\mathbf{h}_0 = \mathbf{x}$ denoting the input.

Beyond the stacking and noise level, the performance of sMDA may depend on the data normalization and pre-processing. In Section 4, we test different options and parameters of sMDA.

3 Datasets and Evaluation Framework

We tested sMDA on a large set of domain adaptation tasks, using three image and one text datasets.

ICDA. We denote by ICDA the dataset that was used in the ImageClef 2014 Domain Adaptation challenge. It consists of a set of SIFT BOV¹ features provided for 12 common classes of five different image collections: Caltech-256 (C), ImageNet (I), PascalVOC (I), Bing (B) and SUN (S). The first four collections are treated as *source* domains; for each of them 600 image features and the corresponding labels were provided. The SUN dataset served as the *target* domain, with 60 annotated and 600 non-annotated instances. The domain adaptation task is to provide predictions for the non-annotated target data. Neither the images nor their low-level features used to generate the BOV are available. The Challenge run in two phases where the participants were provided with a similar configuration but different features. We distinguish them by denoting the corresponding feature sets as ICDA1 or 3 (phase 1) and ICDA2 (phase 2). The ICDA1 and ICDA3 share the same feature sets but different in the evaluation setting; the former applies the cross validation on the full train and test set with 11 folds [8], while ICDA3 corresponds to results obtained with the provided train-test split at phase 1.

OC10. Office+Caltech10 is a dataset frequently used for testing domain adaptation techniques [1, 11, 14, 15]. In our experiments we use the SURF BOV² available from http://www.scf.usc.edu/~boqinggo/domain_adaptation/GFK_v1.zip. The dataset consists of four domains: Amazon (A), Caltech (C), dslr (D) and Webcam (W), with 10 common classes. Each domain is considered in its turn as a *target*, with the other domains considered as *sources*. First, we followed the experimental setting of [11, 14, 15], to build the training set with 8 images from each class (for D or W as source domains) or 20 images (for A or C) randomly selected, to which 3 target instances per class were added in the case of semi-supervised (SS) setting. All experiments were repeated 10 times and averaged. We denote this case by OC10s referring to the small source set. The case when all source data is used is denoting OC10a.

¹ Bag-of-visual (BOV) words [9] built using SIFT features [20] extracted on interest points.

² Bag-of-visual (BOV) words [9] built on SURF features [2] on interest points.

OFF31. Another popular dataset used to compare domain adaptation methods is the Office31 dataset [22] containing images of 31 product classes downloaded from amazon.com (Amazon) or taken in an office environment using a webcam or digital SLR camera (dslr), respectively. Note that the 3 corresponding domains in OffCal10 are subsets of this dataset. We consider the provided SURF BOV features available on <http://www.cs.uml.edu/~saenko/projects.html#data>, and the corresponding experimental framework, which is similar to the OC10 setting. We also consider the case where all available source data is used and denote it OFF31a, while the small set is denoted OFF31s.

AMT. The Amazon text dataset consists of products reviews in different domains. If a book review can be quite different from a kitchen item review, there are nevertheless some common features to assess whether the customers were satisfied with their purchase. Blitzer et al. [4] preprocessed a sub-part of this collection which has been used subsequently in several studies for domain adaptation. The task is to predict whether a customer review is positive or negative where a review with more than 3 stars is considered as positive and (strictly) less than 3 as negative. After preprocessing, documents are represented by a bag of unigrams and bigrams. For our experiments, we only considered the top 10,000 features according to document frequency and the four domains used in most studies: *kitchen* (K), *dvd* (D), *books* (B) and *electronics* (E). Furthermore, we varied the training set size as we considering first ‘all’ source data with roughly 5,000 document for each class, (denoted with AMTa), then considering a ‘medium’ size experiment (denoted by AMTm) with 2000 source documents for training and 2000 targets document for tests (this is the classical setting on most other domain adaptation studies). Finally, we also built random ‘small’ collections (denoted by AMTs) where 200 documents were selected randomly from each source (100 per class) and from the target as labeled set in the semi-supervised setting, and tested on the remained unlabeled target documents. This latter selection process was repeated 10 times and the results were averaged over the 10 runs.

4 Evaluation Results

We run two series of evaluations. In the first one, we test different aspects of the sMDA method, in particular, the number of stacking layers, the amount of noise, the data normalization and data pre-processing. In the second one, we compare these methods with the state-of-the art methods, both in unsupervised and semi-supervised settings.

Varying the Noise Level. First, we study the sensitivity of the sMDA methods to the noise by varying the probability p between 0.1 to 0.9, with all other parameters being fixed. In the experiments with the ICDA image dataset we used the linear SVM with 5 stacking layers on z-score normalized features concatenated to the original features (as in [5]). In the case of the AMT text set we used only a single layer on L2 normalized TFIDF+L2 features.

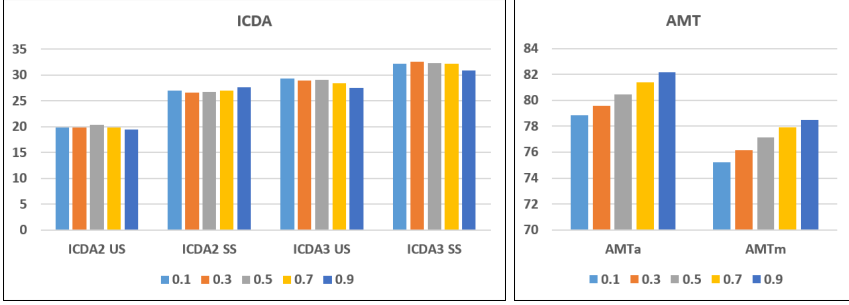


Fig. 1. ICDA2 and ICDA3 (left) and AMT (right) average accuracy for different noise levels.

Figure 1.left) shows the results for ICDA sets averaged over 15 configurations: (C) \rightarrow S, (I) \rightarrow S, \dots , (C,I,P,B) \rightarrow S, both in unsupervised (US) and semi-supervised (SS) modes. It is easy to see that for the image data, the sMDA methods seem to be fairly robust to the noise level, the globally most convenient value is close to 0.5. On contrary, Figure 1.right) shows results on AMT set where increasing the noise level increases the accuracy. In the followings, we systematically use $p=0.5$ for the image sets, and $p=0.9$ for the text set.

Feature Normalization for Images. Like in any deep learning architecture, we pay a particular attention to the data preprocessing when using sMDAs, as these methods appear to be highly sensitive to the spread and variance of feature values. We mainly focus on the features themselves, therefore instead of combining the results with SVM, in this section we use them with domain specific class mean classifier³ (DSCM)[6,8], because the DSCM does not require any meta parameter tuning and is extremely fast. Furthermore, we consider only a single stacking layer, and we concatenate the MDA output (denoted by L1) with the original (NO) or previously normalized features; we then train the DSCM in this concatenated space.

We experimented with two feature normalizations on the image datasets. The first denoted as **P05** is the power normalization ($x_{ij} = x_{ij}^{0.5}$), previously used in [6,8]. The second is the z-score function $\mathcal{Z}(\mathbf{X})$. For the input data $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$, we set $x_{ij} = (x_{ij} - \mathbb{E}(\mathbf{X}_j)) / \text{std}(\mathbf{X}_j)$, where $\mathbb{E}(\mathbf{X}_j)$ and $\text{std}(\mathbf{X}_j)$ are the mean and standard deviation of feature j . It can be applied in three following ways:

- **ZA**: *jointly* on all sources S_j and the target data, $\mathcal{Z}([S_1, \dots, S_N, T])$;
- **ZS**: *independently* on each source and target data, $\mathcal{Z}(S_1), \dots, \mathcal{Z}(S_N), \mathcal{Z}(T)$;

³ The domain specific class mean classifier assign a test data to a class based on a weighted softmax distance to domain-specific class means: $p(c|\mathbf{x}_i) = \frac{1}{Z_i} \sum_{d=1}^D w_d \exp(-\frac{1}{2} \|\mathbf{x}_i - \boldsymbol{\mu}_d^c\|_2^2)$, where $\boldsymbol{\mu}_d^c$ is the average of the class c in domain d , w_d re domain specific weights and Z_i is a normalizer. We used $w_s = 1$ for all sources and in the semi-supervised setting $w_t = 2$ for the target.

Table 1. Different normalization and pre-processing strategies for image datasets. Strategy-best cases are underlined, overall best ones are shown in red.

Dataset		Semi supervised					Unsupervised				
	S_i	NO	P05	ZA	ZS	ZS _{ZT}	NO	P05	ZA	ZS	ZS _{ZT}
ICDA2	S_0	19.82	24.83	<u>25.91</u>	25.74	25.69	11.83	13.4	13.81	15.67	15.69
	S_1	17.92	25.08	<u>25.27</u>	24.51	24.46	13.33	13.44	13.74	15.5	15.58
	S_2	14.84	17.58	<u>25.89</u>	25.51	25.29	10.51	10.82	13.74	15.41	<u>15.6</u>
	S_3	17.92	25.08	25.94	25.91	25.77	13.33	13.44	13.69	15.57	<u>15.63</u>
ICDA3	S_0	22.49	32.16	35.2	33.99	32.98	18.09	23.83	24.96	26.79	26.7
	S_1	21.73	32.21	<u>35.01</u>	34.27	34.27	17.3	23.72	25.36	26.84	26.84
	S_2	22.96	26.63	<u>34.68</u>	34.84	33.81	18.83	20.96	24.74	<u>26.29</u>	26.16
	S_3	21.73	32.21	<u>34.82</u>	33.96	34	17.3	23.72	24.78	<u>26.71</u>	26.53
OC10s	S_0	52.05	<u>56.42</u>	55.35	56.15	56.15	43.63	45.94	48.52	<u>49.84</u>	49.4
	S_1	54.97	57.16	56.71	57.27	57.3	46.15	46.82	49.59	49.92	<u>50.07</u>
	S_2	54.48	55.12	56.26	56.74	<u>56.76</u>	45.88	46.43	49.17	<u>49.91</u>	49.81
	S_3	54.97	<u>57.16</u>	56.76	57.09	57.11	46.15	46.82	49.73	49.93	50.11
OFF31s	S_0	33.33	43.03	42.64	<u>45.33</u>	44.77	15.06	20.71	20.74	<u>25.75</u>	23.76
	S_1	34.47	41.11	43.48	<u>45.52</u>	45.03	14.78	20.24	21.55	<u>26.44</u>	24.66
	S_2	34.14	30.94	44.95	47.08	46.75	14.85	16.22	21.54	26.7	25.1
	S_3	34.47	41.11	43.48	<u>45.88</u>	45.42	14.78	20.24	21.04	<u>26.19</u>	24.33

- **ZS_{ZT}**: *separately* on the source combination and the target data, $\mathcal{Z}([S_1, \dots, S_N]), \mathcal{Z}(T)$.

In addition we compare the normalization effects to the *no normalization* (NO) case.

The feature normalization can be further coupled with the following pre-processing options, applied after normalization but before using the sMDA:

- S_0 , *baseline*: features are used directly to learn a classifier, without any MDA layer;
- S_1 : features are used as such by the MDA;
- S_2 : features are binarized; this can help MDA to capture the feature co-occurrences;
- S_3 : all negative feature values are set to zero.

Note that these pre-processing options are applied on the input of the MDA, but not on the original (normalized) features that are concatenated with the MDA output.

We test all image normalization and pre-processing combinations on all image datasets, both in unsupervised and semi-supervised modes. In Table 1) we report average results over all possible DA tasks (target and source combinations). For example, for Office 31, the USL scores are averaged over 9 possible tasks: $D \rightarrow A$, $W \rightarrow A$, $(D, W) \rightarrow A$, $A \rightarrow D$, $W \rightarrow D$, $(A, W) \rightarrow D$, $A \rightarrow W$, $D \rightarrow W$ and $(A, D) \rightarrow W$.

We analyze Table 1 and draw the following conclusions:

- Feature normalization is an important factor for the MDA+DSCM classification. With no normalization (S_i, NO) the results are always low. Z-score normalization performs better than with P05. Among normalization strategies,

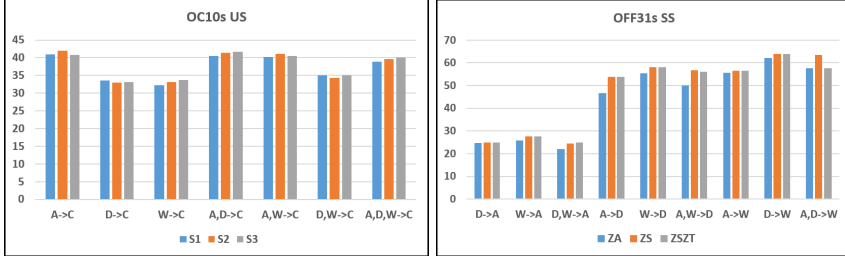


Fig. 2. Further examples comparing different feature correlation strategies results.

independent source normalization (**ZS**) is either the best or close to the best. Note that **ZS** and **ZSZT** differ when we have more than one sources.

- Combining normalized features with the output of the MDA (O+L1) does not seem to always help in the case of ICDA dataset, but we have a consistent gain in the case of OC10 and OFF31. The tree feature preprocessing strategies seem to give relatively similar results except for OC10s, where binarizing the z-scored feature vectors performs better.
- Amongst the three preprocessing strategies we do not have a clear winner, but strategy S_2 seems to be a good compromise in most cases.

When we analyze the results for every individual domain adaptation task for each dataset (see also Figure 2) and tracked the correlation between the best normalization and pre-processing strategies again we find the strategy S_2 with **ZS** normalization as a good choice in most cases.

Feature Normalization for Text. Feature normalization for text is more known as term weighting. It often differs from the normalization of image features, due to a higher sparsity of textual representation. Here we experiment with the AMTa (all) and AMTm (small) cases using the following six strategies:

- raw term frequency, without normalization (**TF**) and with L2 normalization (**TF+L2**),
- term frequency binarization (**BIN**),
- TF-IDF with L2 normalization (**TFIDF+L2**),
- Q-Learning term weighting function [7] without (**QLN**) and with L2 normalization (**QLN+L2**).

Previous experiments showed that a stronger noise level is needed for the text reconstruction, so we set the noise level p to 0.9. We use linear SVM and a single layer MDA without feature concatenation (L1). We only test the unsupervised domain adaptation case, where the SVM classifiers are cross-validated on the source data, and then evaluated on the target documents. Table 2 shows the results averaged over all domain adaptation tasks. While the gain and the accuracy varies a lot from one configuration to another, MDA always helps, independently of the initial normalization.

Different Number of Stacking Layers. In the previous experiments, we used DSCM for the sake of simplicity and speed. However, DSCM did not benefit from

Table 2. Different term weighting strategies for text.

Dataset	S_i	TF	TF + L2	BIN	TFIDF + L2	QLN	QLN + L2
AMTa	S_0	79.75	79.62	80	80.4	<u>81.24</u>	80.72
	S_1	84.34	84	84.04	83.88	84.16	83.97
AMTm	S_0	76.38	75.72	76.26	77.02	<u>78</u>	77.69
	S_1	79.98	79.13	79.87	79.3	79.69	79.76

using multiple layers. For DSCM, neither increasing the number of dimensions nor using feature redundancy is necessarily helpful. This is why, in this section, we turned to SVM classifiers as they can cope with both.

To analyze the stacking effect, when two or more layers are used we use the linear multi-class SVM from the LIBSVM package⁴ with the fixed cost $c = 0.1$ in all experiments. In Table 3 we report only results for the normalization and preprocessing strategy (\mathbf{ZS}, S_2) as one performing well with DSCM; for other strategies we observe a similar behavior.

We tested configurations including 1 to 5 layers and the feature concatenation options including:

- (\mathbf{Li}) uses the last layer as features in the SVM,
- $(\mathbf{O+Li})$ concatenates the original features with last layer output $(\mathbf{O+Li})$,
- $(\mathbf{O+L1} \rightarrow \mathbf{Li})$ concatenates the original features with all the layers up to \mathbf{Li} .

As in the case of DSCM, Table 3 shows results averaged over all different domain adaptation tasks and configurations. From the results we can conclude that in general (except for ICDA2 US) best results are obtained when we concatenate the output of all the layers $(\mathbf{O+L1} \rightarrow \mathbf{Li})$. However it is rare that we need to stack more than 3 layers to get significantly better results. In Figure 3 we show some configuration results for $\mathbf{O+L1} \rightarrow \mathbf{Li}$. While the best stacking option varies from configuration to configuration, considering 3 layers seems a good compromise in general.

On the AMT text set, we limited the stacking to 3 layers due to the high feature dimensionality. In these experiments, we tested both the semi-supervised and the unsupervised settings for the small collection (200 document per domain) with the $(\mathbf{TFIDF+L2})$ normalization and a noise level of 0.9. In the case of semi supervised settings, we added randomly 100 documents per class (satisfied and unsatisfied) from the target. We show the average results over all possible target sets and all possible source configurations in Table 4 where we varied the number of stacking layers. From the table we can see that adding extra stacks helps but the gain is relatively small except for the unsupervised case where using a more than a single stacking layers really helps.

4.1 Comparing sMDAs to Other Domain Adaptation Approaches

In this section we compare our domain adaptation results to the ones published recently in [10, 12] using the same experimental settings (see Section 3).

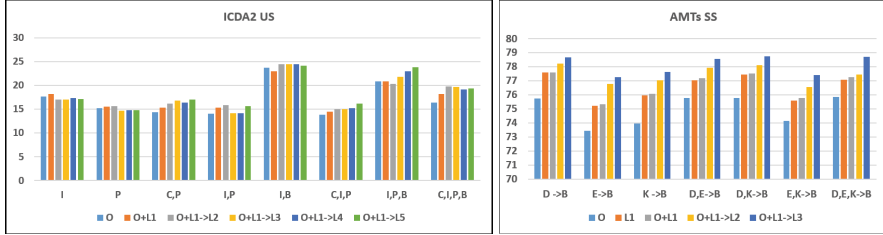
⁴ <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

Table 3. Different normalization and pre-processing strategies for image datasets.

Dataset		Semi supervised					Unsupervised				
	Layer	L1	L2	L3	L4	L5	L1	L2	L3	L4	L5
ICDA2	Li	25.83	24.70	22.88	21.89	22.24	20.46	19.39	18.78	16.31	17.08
	O+Li	26.14	25.73	25.00	25.15	25.45	18.36	18.10	17.99	18.37	<u>18.65</u>
	O+L1→Li	26.14	25.96	25.86	25.86	25.96	18.36	18.50	18.39	18.39	<u>18.84</u>
ICDA3	Li	30.25	28.76	27.00	25.79	22.95	26.96	25.47	24.27	22.82	20.60
	O+Li	31.09	<u>31.20</u>	30.84	30.70	30.30	28.10	<u>28.11</u>	27.99	27.80	27.59
	O+L1→Li	31.09	31.75	31.75	31.78	31.45	28.10	28.36	28.53	28.33	28.02
OC10s	Li	54.58	55.11	<u>55.42</u>	53.88	51.92	52.13	<u>52.60</u>	52.23	49.59	47.52
	O+Li	52.97	53.41	53.72	<u>53.75</u>	53.44	50.73	51.44	<u>51.94</u>	51.71	51.26
	O+L1→Li	52.97	54.27	54.82	55.32	55.51	50.73	51.99	52.69	52.99	53.03
OFF31s	Li	<u>43.78</u>	43.3	42.29	40.75	39.31	<u>22.89</u>	22.15	18.31	15.23	13.20
	O+Li	42.61	43.82	<u>43.86</u>	<u>43.86</u>	43.16	<u>26.97</u>	26.91	26.46	26.07	25.50
	O+L1→Li	42.61	44.52	45.32	45.17	45.23	26.97	27.31	26.73	25.91	25.31

Table 4. Different number of stacking layers for the text dataset.

Dataset		Semi supervised			Unsupervised		
	Layer	L1	L2	L3	L1	L2	L3
AMTs	Li	79.97	80.08	<u>80.63</u>	74.89	76.23	<u>76.93</u>
	O+Li	80.14	80.16	<u>80.64</u>	74.98	76.3	<u>76.94</u>
	O+L1→Li	80.14	80.55	80.81	74.98	76.54	77

**Fig. 3.** Different stacking evaluations for ICDA2 and AMTs cases.

In the case of ICDA datasets, Table 5 compares our results to the Self-adaptive Metric Learning for Domain Adaptation (SaMLDa) as it also exploits the unlabeled target instances to iteratively adjust the metric learned for the DSCM [8]. From these results, we can see that using DSCM with independent (**ZS**) feature normalization performs the best on both ICDA datasets. This is an interesting finding, as the DSCM method is very fast and requires no parameter tuning. In addition, as Table 1 shows this method performs extremely well also in the case of the unsupervised learning.

Table 5. Classification accuracy on ICDA1.

	SVM P05	SVM ZS	SVM+sMDA + ZS,S2,(O+L1→L3)	DSCM P05	DSCM ZS	DSCM+MDA ZS,S2(O+L1)	SaMLDa [8] P05
ICDA1 SS	30.31	35	32	31.37	35.21	34.97	33.67
ICDA2 SS	25.92	24.61	25.65	26.13	27.37	26.95	27

Table 6. Results on OC10. We show our best results in underlined and the overall best results in red.

	SVM P05	SVM ZS	SVM+sMDA + ZS,S2,(O+L1→L3)	DSCM P05	DSCM ZS	DSCM+MDA ZS,S2(O+L1)	[11] SA
OC10a (US)	42.69	41.54	44.22	43.13	45.49	<u>45.84</u>	45.9
OC10a (SS)	53.83	51.03	53.68	55.83	53.7	54.37	53.67
OC10s US	44.8	45.66	<u>47.99</u>	41.2	43.07	43.56	51.4
OC10s SS	51.7	49.65	52.62	54.72	54.13	54.81	-
A→W (US)	14.8	16.87	17.86	17.95	22.36	20.8	15.3
A→W (SS)	47.59	40.91	44.07	54.13	53.99	56.55	45
D→W (US)	49.97	48.18	55.13	42.88	42.59	48.29	50.1
D→W (SS)	68.73	63.7	67.22	56.98	58.97	63.96	63.8
W→D (US)	39.65	45.83	47.19	37.53	48.89	<u>51.85</u>	56.9
W→D (SS)	58.42	64.44	<u>66.05</u>	52.12	56.79	58.02	69.9

Table 6 compares our results to the results of [11, 12], for the OC10a and OC10s cases⁵, and the 3 available source target configuration available in the literature from OFF31 datasets always using the same experimental protocol (described in Section 3). From the table we can conclude the following. Again DSCM with z-normalization and even without sMDA performs extremely well in the case of semi-supervised setting in spite of its simplicity, but SVM performs better when we do not have any labeled target sample. sMDA in general helps to increase the accuracy in average with 2-3% both in the case of DSCM and SVM.

5 Conclusion

In this paper we address the problem of domain adaptation using multiple source domains. In particular we intensively evaluated the deep learning technique of the *stacked marginalized denoising autoencoders* (sMDA). A detailed analysis of evaluations of sMDA parameters and comparison to other state of art methods allow us to make the following conclusions:

- sMDA gives a consistent classification improvement in different domain adaptation scenarios;

⁵ We average the results for only the 12 “one source versus one target” for OC10 and only 9 “one source versus one target” cases, as in [11, 12].

- It is complementary to any other components, like learning from multiple sources, available target labels instances, image or text classification, etc.
- Due to the noise marginalization in the closed form, sMDA is a fast and low-cost alternative to the energy-expensive deep learning solutions [13];
- Optimal values of two main parameters, the stacking size and the noise level, can be detected by cross validation, but the default setting $p = 0.5$ and $m = 3$ works well in most cases;
- Data normalization plays an important role; independent or joint domain data normalization are top preferences;
- Due to unsupervised feature extraction, sMDA yields a larger gain over the baselines in unsupervised learning, when no target label information is available.

References

1. Baktashmotlagh, M., Harandi, M.T., Lovell, B.C., Salzmann, M.: Unsupervised domain adaptation by domain invariant projection. In: ICCV (2013)
2. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
3. Beijbom, O.: Domain adaptations for computer vision applications (2012). arXiv (1211.4860)
4. Blitzer, J., Foster, D., Kakade, S.: Domain adaptation with coupled subspaces. In: ICAIS (2011)
5. Chen, M., Xu, Z., Weinberger, K.Q., Sha, F.: Marginalized denoising autoencoders for domain adaptation (2012). arXiv (1206.4683)
6. Chidlovskii, B., Csurka, G., Gangwar, S.: Assembling heterogeneous domain adaptation methods for image classification. In: Working Notes for CLEF 2014 (2014)
7. Clinchant, S.: Concavity in IR models. In: CIKM (2012)
8. Csurka, G., Chidlovskii, B., Perronnin, F.: Domain adaptation with a domain specific class means classifier. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) ECCV 2014 Workshops. LNCS, vol. 8927, pp. 32–46. Springer, Heidelberg (2015)
9. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: SLCV, ECCV Workshop (2004)
10. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. In: ICML (1999)
11. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: ICCV (2013)
12. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Subspace alignment for domain adaptation (2014). arXiv (1409.5241)
13. Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: a deep learning approach. In: ICML (2011)
14. Gong, B., Grauman, K., Sha, F.: Reshaping visual datasets for domain adaptation. In: NIPS (2013)
15. Gopalan, R., Li, R., Chellappa, R.: Domain adaptation for object recognition: an unsupervised approach. In: ICCV (2011)

16. Hoffman, J., Kulis, B., Darrell, T., Saenko, K.: Discovering latent domains for multisource domain adaptation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 702–715. Springer, Heidelberg (2012)
17. Jhuo, I.-H., Liu, D., Lee, D.T., Chang, S.-F.: Robust visual domain adaptation with low-rank reconstruction. In: CVPR, pp. 2168–2175 (2012)
18. Jiang, J.: A literature survey on domain adaptation of statistical classifiers (2008). <https://scholar.google.com.sg/citations?user=hVTK2YwAAAAJ>
19. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: domain adaptation using asymmetric kernel transforms. In: CVPR (2011)
20. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV (1999)
21. Patel, V.M., Gopalan, R., Li, R., Chellappa, R.: Visual domain adaptation: An overview of recent advances. *IEEE Transactions on Geoscience and Remote Sensing* **52**(2) (2007)
22. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 213–226. Springer, Heidelberg (2010)
23. Tommasi, T., Caputo, B.: Frustratingly easy NBN domain adaptation. In: ICCV (2013)
24. Torralba, A., Efros, A.: Unbiased look at dataset bias. In: CVPR (2011)
25. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A.: Extracting and composing robust features with denoising autoencoders. In: ICML (2008)
26. Xu, Z., Chen, M., Weinberger, K.Q., Sha, F.: From sBoW to dCoT marginalized encoders for text representation. In: CIKM (2012)

Experimental IR Meets Multilinguality, Multimodality, and
Interaction

6th International Conference of the CLEF Association,
CLEF'15, Toulouse, France, September 8-11, 2015,
Proceedings

Mothe, J.; Savoy, J.; Kamps, J.; Pinel-Sauvagnat, K.;
Jones, G.; Sanjuan, E.; Cappellato, L.; Wolf, I. (Eds.)
2015, XXIII, 567 p. 110 illus. in color., Softcover
ISBN: 978-3-319-24026-8