Multinomial Adversarial Networks for Multi-Domain Text Classification

Xilun Chen
Department of Computer Science
Cornell Unversity
Ithaca, NY, 14853, USA
`xlchen@cs.cornell.edu`

Claire Cardie
Department of Computer Science
Cornell Unversity
Ithaca, NY, 14853, USA
`cardie@cs.cornell.edu`

Abstract

Many text classification tasks are known to be highly domain-dependent. Unfortunately, the availability of training data can vary drastically across domains. Worse still, for some domains there may not be any annotated data at all. In this work, we propose a *multinomial adversarial network*[1] (MAN) to tackle this real-world problem of multi-domain text classification (MDTC) in which labeled data may exist for multiple domains, but in insufficient amounts to train effective classifiers for one or more of the domains. We provide theoretical justifications for the MAN framework, proving that different instances of MANs are essentially minimizers of various f-divergence metrics (Ali and Silvey, 1966) among *multiple* probability distributions. MANs are thus a theoretically sound generalization of traditional adversarial networks that discriminate over *two* distributions. More specifically, for the MDTC task, MAN learns features that are invariant across multiple domains by resorting to its ability to reduce the divergence among the feature distributions of each domain. We present experimental results showing that MANs significantly outperform the prior art on the MDTC task. We also show that MANs achieve state-of-the-art performance for domains with no labeled data.

1 Introduction

Text classification is one of the most fundamental tasks in Natural Language Processing, and has found its way into a wide spectrum of NLP applications, ranging from email spam detection and social media analytics to sentiment analysis and data mining. Over the past couple of decades, supervised statistical learning methods have become the dominant approach for text classification (e.g. McCallum et al. (1998); Kim (2014); Iyyer et al. (2015)). Unfortunately, many text classification tasks are highly domain-dependent in that a text classifier trained using labeled data from one domain is likely to perform poorly on another. In the task of sentiment classification, for example, the phrase "runs fast" is usually associated with positive sentiment in the sports domain; not so when a user is reviewing the battery of an electronic device. In real applications, therefore, an adequate amount of training data from each domain of interest is typically required, and this is expensive to obtain.

Two major lines of work attempt to tackle this challenge: **domain adaptation** (Blitzer et al., 2007) and **multi-domain text classification** (MDTC) (Li and Zong, 2008). In domain adaptation, the assumption is that there is some domain with abundant training data (the source domain), and the goal is to utilize knowledge learned from the source domain to help perform classifications on another lower-resourced target domain.[2] The focus of this work, MDTC, instead simulates an arguably more realistic scenario, where labeled data may exist for multiple domains, but in insufficient amounts to train an effective classifier for one or more of the domains. Worse still, some domains may have *no* labeled data at all. The objective of MDTC is to leverage *all* the available resources in order to improve the system performance over all domains simultaneously.

One state-of-the-art system for MDTC, the CMSC system of Wu and Huang (2015), combines a classifier that is shared across all domains (for learning domain-invariant knowledge) with a set of classifiers, one per domain, each of which captures domain-specific text classification knowledge. This paradigm is sometimes known

---

[1] The source code of MAN is available at `https://github.com/ccsasuke/man`.

[2] See §6 for other variants of domain adaptation.

as the Shared-Private model (Bousmalis et al., 2016). CMSC, however, lacks an explicit mechanism to ensure that the shared classifier captures only domain-independent knowledge: the shared classifier may well also acquire some domain-specific features that are useful for a subset of the domains. We hypothesize that better performance can be obtained if this constraint were explicitly enforced.

In this paper, we thus propose Multinomial Adversarial Networks (henceforth, MANs) for the task of multi-domain text classification. In contrast to standard adversarial networks (Goodfellow et al., 2014), which serve as a tool for minimizing the divergence between *two* distributions (Nowozin et al., 2016), MANs represent a family of theoretically sound adversarial networks that, in contrast, leverage a *multinomial discriminator* to directly minimize the divergence among multiple probability distributions. And just as binomial adversarial networks have been applied to numerous tasks (e.g. image generation (Goodfellow et al., 2014), domain adaptation (Ganin et al., 2016), cross-lingual text classification (Chen et al., 2016)), we anticipate that MANs will make a versatile machine learning framework with applications beyond the MDTC task studied in this work.

We introduce the MAN architecture in §2 and prove in §3 that it directly minimizes the (generalized) f-divergence among multiple distributions so that they are indistinguishable upon successful training. Specifically for MDTC, MAN is used to overcome the aforementioned limitation in prior art where domain-specific features may sneak into the shared model. This is accomplished by relying on MAN's power of minimizing the divergence among the feature distributions of each domain. The high-level idea is that MAN will make the extracted feature distributions of each domain indistinguishable from one another, thus learning general features that are invariant across domains.

We then validate the effectiveness of MAN in experiments on two MDTC data sets. We find first that MAN significantly outperforms the state-of-the-art CMSC method (Wu and Huang, 2015) on the widely used multi-domain Amazon review dataset, and does so without relying on external resources such as sentiment lexica (§4.1). When applied to the second dataset, FDU-MTL (§4.3), we obtain similar results: MAN achieves substantially higher accuracy than the previous top-performing

method, ASP-MTL (Liu et al., 2017). ASP-MTL is the first empirical attempt to use a multinomial adversarial network for multi-task learning, but is more restricted and can be viewed as a special case of MAN. In addition, we provide the first theoretical guarantees for multinomial adversarial networks (§3). Finally, while many MDTC methods such as CMSC require labeled data for each domain, MANs can be applied in cases where no labeled data exists for a subset of domains. To evaluate MAN in this semi-supervised setting, we compare MAN to a method that can accommodate unlabeled data for (only) one domain (Zhao et al., 2017), and show that MAN achieves performance comparable to the state of the art (§4.2).

2 Model

In this paper, we strive to tackle the text classification problem in the real-world setting in which texts come from a variety of domains, each with a varying amount of labeled data. Specifically, assume we have a total of N domains, N_1 *labeled domains* (denoted as Δ_L) for which there is some labeled data, and N_2 *unlabeled domains* (Δ_U) for which no annotated training instances are available. Denote $\Delta = \Delta_L \cup \Delta_U$ as the collection of all domains, with $N = N_1 + N_2$. The goal of this work, and of MDTC in general, is to improve the overall classification performance across all N domains, measured in this paper as the average[3] classification accuracy across the N domains in Δ.

2.1 Model Architecture

As shown in Figure 1, the Multinomial Adversarial Network (MAN) adopts the Shared-Private paradigm of Bousmalis et al. (2016) and consists of four components: a *shared feature extractor* $\mathcal{F}_s$, a *domain feature extractor* $\mathcal{F}_{d_i}$ for each labeled domain $d_i \in \Delta_L$, a *text classifier* $\mathcal{C}$, and a *domain discriminator* $\mathcal{D}$. The main idea of MAN is to explicitly model the domain-invariant features that are beneficial to the main classification task across all domains (i.e. the *shared features*, extracted by $\mathcal{F}_s$), as well as the domain-specific features that mainly contribute to the classification in its own domain (the *domain features*, extracted by $\mathcal{F}_d$). Here, the adversarial domain discriminator $\mathcal{D}$ has a multinomial output that takes a shared feature

---

[3]In this work, we use macro-average over domains, but MAN can be readily adapted for micro-average or other (weighted) averaging schemes.

$J_{\mathcal{D}}$ $\quad \lambda J_{\mathcal{F}_s}^{\mathcal{D}}$ $\qquad J_{\mathcal{C}}$ (if $d_i \in \Delta_L$)

Domain Label

Class Label

Domain Discriminator $\mathcal{D}$

Text Classifier $\mathcal{C}$

Shared Feature Extractor $\mathcal{F}_s$

Domain Feature Extractor $\mathcal{F}_{d_i}$

Mini-batch of documents from domain $d_i \in \Delta$

⇄ Forward and backward passes when updating the parameters of $\mathcal{F}_s$, $\mathcal{F}_d$ and C

⇄ Forward and backward passes when updating the parameters of $\mathcal{D}$

Algorithm 1 MAN Training

Require: labeled corpus $\mathbb{X}$; unlabeled corpus $\mathbb{U}$; Hyperpa-mameter $\lambda > 0$, $k \in \mathbb{N}$
1: **repeat**
2: $\quad \triangleright \mathcal{D}$ iterations
3: $\quad$ **for** $diter = 1$ to k **do**
4: $\qquad l_{\mathcal{D}} = 0$
5: $\qquad$ **for all** $d \in \Delta$ **do** $\qquad\qquad \triangleright$ For all N domains
6: $\qquad\quad$ Sample a mini-batch $\boldsymbol{x} \sim \mathbb{U}_d$
7: $\qquad\quad \boldsymbol{f}_s = \mathcal{F}_s(\boldsymbol{x})$ $\qquad\quad \triangleright$ Shared feature vector
8: $\qquad\quad l_{\mathcal{D}} \mathrel{+}= J_{\mathcal{D}}(\mathcal{D}(\boldsymbol{f}_s); d)$ $\qquad \triangleright$ Accumulate $\mathcal{D}$ loss
9: $\qquad$ Update $\mathcal{D}$ parameters using $\nabla l_{\mathcal{D}}$
10: $\quad \triangleright$ Main iteration
11: $\quad loss = 0$
12: $\quad$ **for all** $d \in \Delta_L$ **do** $\qquad \triangleright$ For all labeled domains
13: $\qquad$ Sample a mini-batch $(\boldsymbol{x}, \boldsymbol{y}) \sim \mathbb{X}_d$
14: $\qquad \boldsymbol{f}_s = \mathcal{F}_s(\boldsymbol{x})$
15: $\qquad \boldsymbol{f}_d = \mathcal{F}_d(\boldsymbol{x})$ $\qquad\quad \triangleright$ Domain feature vector
16: $\qquad loss \mathrel{+}= J_{\mathcal{C}}(\mathcal{C}(\boldsymbol{f}_s, \boldsymbol{f}_d); \boldsymbol{y})$ $\qquad \triangleright$ Compute $\mathcal{C}$ loss
17: $\quad$ **for all** $d \in \Delta$ **do** $\qquad\qquad \triangleright$ For all N domains
18: $\qquad$ Sample a mini-batch $\boldsymbol{x} \sim \mathbb{U}_d$
19: $\qquad \boldsymbol{f}_s = \mathcal{F}_s(\boldsymbol{x})$
20: $\qquad loss \mathrel{+}= \lambda \cdot J_{\mathcal{F}_s}^{\mathcal{D}}(\mathcal{D}(\boldsymbol{f}_s); d)$ $\quad \triangleright$ Domain loss of $\mathcal{F}_s$
21: $\quad$ Update $\mathcal{F}_s, \mathcal{F}_d, \mathcal{C}$ parameters using $\nabla loss$
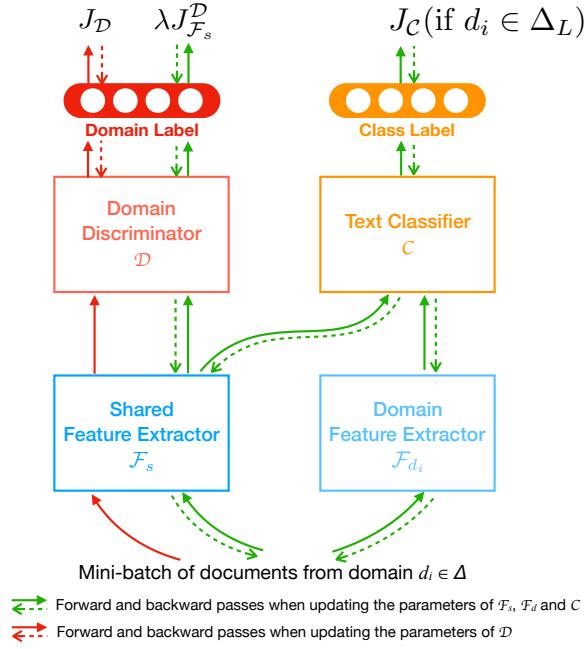22: **until** convergence

---

Figure 1: MAN for MDTC. The figure demonstrates the training on a mini-batch of data from one domain. One training iteration consists of one such mini-batch training from each domain. The parameters of $\mathcal{F}_s, \mathcal{F}_d, \mathcal{C}$ are updated together, and the training flows are illustrated by the green arrows. The parameters of $\mathcal{D}$ are updated separately, shown in red arrows. Solid lines indicate forward passes while dotted lines are backward passes. $J_{\mathcal{F}_s}^{\mathcal{D}}$ is the domain loss for $\mathcal{F}_s$, which is anticorrelated with $J_{\mathcal{D}}$ (e.g. $J_{\mathcal{F}_s}^{\mathcal{D}} = -J_{\mathcal{D}}$). (See §2,§3)

vector and predicts the likelihood of that sample coming from each domain. As seen in Figure 1, during the training of $\mathcal{F}_s$ (green arrows denote the training flow), $\mathcal{F}_s$ aims to confuse $\mathcal{D}$ by minimizing $J_{\mathcal{F}_s}^{\mathcal{D}}$, which is anticorrelated to $J_{\mathcal{D}}$ (detailed in §2.2), so that $\mathcal{D}$ cannot predict the domain of a sample given its shared features. The intuition is that if even a strong discriminator $\mathcal{D}$ cannot tell the domain of a sample from the extracted features, those features $\mathcal{F}_s$ learned are essentially domain invariant. By enforcing domain-invariant features to be learned by $\mathcal{F}_s$, when trained jointly via back-propagation, the set of domain feature extractors $\mathcal{F}_d$ will each learn domain-specific features beneficial within its own domain.

The architecture of each component is relatively flexible, and can be decided by the practitioners to suit their particular classification tasks. For instance, the feature extractors can adopt the form of Convolutional Neural Nets (CNN), Recurrent Neural Nets (RNN), or a Multi-Layer Perceptron (MLP), depending on the input data (see §4). The input of MAN will also be dependent on the feature

extractor choice. The output of a (shared/domain) feature extractor is a fixed-length vector, which is considered the (shared/domain) hidden features of some given input text. On the other hand, the outputs of $\mathcal{C}$ and $\mathcal{D}$ are label probabilities for class and domain prediction, respectively. For example, both $\mathcal{C}$ and $\mathcal{D}$ can be MLPs with a softmax layer on top. In §3, we provide alternative architectures for $\mathcal{D}$ and their mathematical implications. We now present a detailed description of the MAN training in §2.2 as well as the theoretical grounds in §3.

2.2 Training

Denote the annotated corpus in a labeled domain $d_i \in \Delta_L$ as $\mathbb{X}_i$; and $(x, y) \sim \mathbb{X}_i$ is a sample drawn from the labeled data in domain d_i, where x is the input and y is the task label. On the other hand, for any domain $d_{i'} \in \Delta$, denote the unlabeled corpus as $\mathbb{U}_{i'}$. Note for the choice of unlabeled data of a labeled domain, one can use a separate unlabeled corpus or simply use the labeled data (or use both).

In Figure 1, the arrows illustrate the training flows of various components. Due to the adversarial nature of the domain discriminator $\mathcal{D}$, it is trained with a separate optimizer (red arrows), while the rest of the networks are updated with the main optimizer (green arrows). $\mathcal{C}$ is only trained on the annotated data from labeled domains, and it takes as input the concatenation of the shared and domain feature vectors. At test time, for data from

1228

unlabeled domains with no $\mathcal{F}_d$, the domain features are set to the **0** vector for $\mathcal{C}$'s input. On the contrary, $\mathcal{D}$ only takes the shared features as input, for both labeled and unlabeled domains. The MAN training procedure is described in Algorithm 1.

In Algorithm 1, $\mathcal{L}_\mathcal{C}$ and $\mathcal{L}_\mathcal{D}$ are the loss functions of the text classifier $\mathcal{C}$ and the domain discriminator $\mathcal{D}$, respectively. As mentioned in §2.1, $\mathcal{C}$ has a $softmax$ layer on top for classification. We hence adopt the canonical negative log-likelihood (NLL) loss:

$$\mathcal{L}_\mathcal{C}(\hat{y}, y) = -\log P(\hat{y} = y) \qquad (1)$$

where y is the true label and $\hat{y}$ is the $softmax$ predictions. For $\mathcal{D}$, we consider two variants of MAN. The first one is to use the NLL loss same as $\mathcal{C}$ which suits the classification task; while another option is to use the Least-Square (L2) loss that was shown to be able to alleviate the gradient vanishing problem when using the NLL loss in the adversarial setting (Mao et al., 2017):

$$\mathcal{L}_\mathcal{D}^{NLL}(\hat{d}, d) = -\log P(\hat{d} = d) \qquad (2)$$

$$\mathcal{L}_\mathcal{D}^{L2}(\hat{d}, d) = \sum_{i=1}^{N} (\hat{d}_i - \mathbb{1}_{\{d=i\}})^2 \qquad (3)$$

where d is the domain index of some sample and $\hat{d}$ is the prediction. Without loss of generality, we normalize $\hat{d}$ so that $\sum_{i=1}^{N} \hat{d}_i = 1$ and $\forall i : \hat{d}_i \geq 0$.

Therefore, the objectives of $\mathcal{C}$ and $\mathcal{D}$ that we are minimizing are:

$$J_\mathcal{C} = \sum_{i=1}^{N} \mathbb{E}_{(x,y) \sim \mathbb{X}_i} \left[\mathcal{L}_\mathcal{C}(\mathcal{C}(\mathcal{F}_s(x), \mathcal{F}_d(x)); y)\right] \quad (4)$$

$$J_\mathcal{D} = \sum_{i=1}^{N} \mathbb{E}_{x \sim \mathbb{U}_i} \left[\mathcal{L}_\mathcal{D}(\mathcal{D}(\mathcal{F}_s(x)); d)\right] \qquad (5)$$

For the feature extractors, the training of domain feature extractors is straightforward, as their sole objective is to help $\mathcal{C}$ perform better within their own domain. Hence, $J_{\mathcal{F}_d} = J_\mathcal{C}$ for any domain d. Finally, the shared feature extractor $\mathcal{F}_s$ has two objectives: to help $\mathcal{C}$ achieve higher accuracy, and to make the feature distribution invariant across all domains. It thus leads to the following bipartite loss:

$$J_{\mathcal{F}_s} = J_{\mathcal{F}_s}^\mathcal{C} + \lambda \cdot J_{\mathcal{F}_s}^\mathcal{D}$$

where λ is a hyperparameter balancing the two parts. $J_{\mathcal{F}_s}^\mathcal{D}$ is the domain loss of $\mathcal{F}_s$ anticorrelated

to $J_\mathcal{D}$:

$$(NLL) J_{\mathcal{F}_s}^\mathcal{D} = -J_\mathcal{D} \qquad (6)$$

$$(L2) J_{\mathcal{F}_s}^\mathcal{D} = \sum_{i=1}^{N} \mathbb{E}_{x \sim \mathbb{U}_i} \left[\sum_{j=1}^{N} (\mathcal{D}_j(\mathcal{F}_s(x)) - \frac{1}{N})^2\right] \qquad (7)$$

If $\mathcal{D}$ adopts the NLL loss (6), the domain loss is simply $-J_\mathcal{D}$. For the L2 loss (7), $J_{\mathcal{F}_s}^\mathcal{D}$ intuitively translates to pushing $\mathcal{D}$ to make random predictions. See §3 for theoretical justifications.

3 Theories of Multinomial Adversarial Networks

Binomial adversarial nets are known to have theoretical connections to the minimization of various f-divergences[4] between *two* distributions (Nowozin et al., 2016). However, for adversarial training among multiple distributions, no theoretical justifications have been provided to our best knowledge, despite that this idea has recently been explored empirically (Liu et al., 2017).

In this section, we present a theoretical analysis showing the validity of MAN. In particular, we show that MAN's objective is equivalent to minimizing the total f-divergence between each of the shared feature distributions of the N domains, and the centroid of the N distributions. The choice of loss function will determine which specific f-divergence is minimized. Furthermore, with adequate model capacity, MAN achieves its optimum for either loss function if and only if all N shared feature distributions are identical, hence learning an invariant feature space across all domains.

First, consider the distribution of the shared features **f** for instances in each domain $d_i \in \Delta$:

$$P_i(\mathbf{f}) \triangleq P(\mathbf{f} = \mathcal{F}_s(x) | x \in d_i) \qquad (8)$$

Combining (5) with the two loss functions (2), (3), the objective of $\mathcal{D}$ can be written as:

$$J_\mathcal{D}^{NLL} = -\sum_{i=1}^{N} \mathbb{E}_{\mathbf{f} \sim P_i} \left[\log \mathcal{D}_i(\mathbf{f})\right] \qquad (9)$$

$$J_\mathcal{D}^{L2} = \sum_{i=1}^{N} \mathbb{E}_{\mathbf{f} \sim P_i} \left[\sum_{j=1}^{N} (\mathcal{D}_j(\mathbf{f}) - \mathbb{1}_{\{i=j\}})^2\right] \qquad (10)$$

---

[4]An f-divergence (Ali and Silvey, 1966) is a function that measures the distance between two probability distributions, e.g. the KL or Jensen-Shannon divergence.

where $\mathcal{D}_i(\mathbf{f})$ is the i-th dimension of $\mathcal{D}$'s (normalized) output vector, which conceptually corresponds to the probability of $\mathcal{D}$ predicting that $\mathbf{f}$ is from domain d_i

We first derive the optimal $\mathcal{D}$ for any fixed $\mathcal{F}_s$.

Lemma 1. *For any fixed $\mathcal{F}_s$, with either NLL or L2 loss, the optimum domain discriminator $\mathcal{D}^*$ is:*

$$\mathcal{D}_i^*(\mathbf{f}) = \frac{P_i(\mathbf{f})}{\sum_{j=1}^N P_j(\mathbf{f})} \qquad (11)$$

The proof involves an application of the *Lagrangian Multiplier* to solve the minimum value of $J_\mathcal{D}$, and the details can be found in the Appendix. We then have the following main theorems for the domain loss for $\mathcal{F}_s$:

Theorem 1. *Let $\overline{P} = \frac{\sum_{i=1}^N P_i}{N}$. When $\mathcal{D}$ is trained to its optimality, if $\mathcal{D}$ adopts the NLL loss:*

$$
\begin{aligned}
J_{\mathcal{F}_s}^\mathcal{D} &= -\min_{\theta_\mathcal{D}} J_\mathcal{D} = -J_{\mathcal{D}^*} \\
&= -N \log N + N \cdot JSD(P_1, P_2, \ldots, P_N) \\
&= -N \log N + \sum_{i=1}^N KL(P_i \| \overline{P})
\end{aligned}
$$

where $JSD(\cdot)$ is the generalized Jensen-Shannon Divergence (Lin, 1991) among multiple distributions, defined as the average Kullback-Leibler divergence of each P_i to the centroid $\overline{P}$ (Aslam and Pavlu, 2007).

Theorem 2. *If $\mathcal{D}$ uses the L2 loss:*

$$
\begin{aligned}
J_{\mathcal{F}_s}^\mathcal{D} &= \sum_{i=1}^N \mathbb{E}_{\mathbf{f} \sim P_i} \left[\sum_{j=1}^N (\mathcal{D}_j^*(\mathbf{f}) - \frac{1}{N})^2 \right] \\
&= \frac{1}{N} \sum_{i=1}^N \chi^2_{Neyman}(P_i \| \overline{P})
\end{aligned}
$$

where $\chi^2_{Neyman}(\cdot \| \cdot)$ is the Neyman χ^2 divergence (Nielsen and Nock, 2014). The proof of both theorems can be found in the Appendix.

Consequently, by the non-negativity and joint convexity of the f-divergence (Csiszar and Korner, 1982), we have:

Corollary 1. *The optimum of $J_{\mathcal{F}_s}^\mathcal{D}$ is $-N \log N$ when using NLL loss, and 0 for the L2 loss. The optimum value above is achieved if and only if $P_1 = P_2 = \cdots = P_N = \overline{P}$ for either loss.*

Therefore, the loss of $\mathcal{F}_s$ can be interpreted as simultaneously minimizing the classification loss

	Book	DVD	Elec.	Kit.	Avg.
Domain-Specific Models Only					
LS	77.80	77.88	81.63	84.33	80.41
SVM	78.56	78.66	83.03	84.74	81.25
LR	79.73	80.14	84.54	86.10	82.63
MLP	81.70	81.65	85.45	85.95	83.69
Shared Model Only					
LS	78.40	79.76	84.67	85.73	82.14
SVM	79.16	80.97	85.15	86.06	82.83
LR	80.05	81.88	85.19	86.56	83.42
MLP	82.40	82.15	85.90	88.20	84.66
MAN-L2-MLP	82.05	83.45	86.45	88.85	85.20
MAN-NLL-MLP	81.85	83.10	85.75	89.10	84.95
Shared-Private Models					
RMTL[1]	81.33	82.18	85.49	87.02	84.01
MTLGraph[2]	79.66	81.84	83.69	87.06	83.06
CMSC-LS[3]	82.10	82.40	86.12	87.56	84.55
CMSC-SVM[3]	82.26	83.48	86.76	88.20	85.18
CMSC-LR[3]	81.81	83.73	86.67	88.23	85.11
SP-MLP	82.00	**84.05**	86.85	87.30	85.05
MAN-L2-SP-MLP	82.46 (±0.25)	83.98 (±0.17)	**87.22*** (±0.04)	88.53 (±0.19)	85.55* (±0.07)
MAN-NLL-SP-MLP	**82.98*** (±0.28)	84.03 (±0.16)	87.06 (±0.23)	**88.57*** (±0.15)	**85.66*** (±0.14)

[1] Evgeniou and Pontil (2004)
[2] Zhou et al. (2011)
[3] Wu and Huang (2015)

Table 1: MDTC results on the Amazon dataset. Models in bold are ours while the performance of the rest is taken from Wu and Huang (2015). Numbers in parentheses indicate standard errors, calculated based on 5 runs. Bold numbers indicate the highest performance in each domain, and $*$ shows statistical significance ($p < 0.05$) over CMSC under a one-sample T-Test.

$J_\mathcal{C}$ as well as the divergence among feature distributions of all domains. It can thus learn a shared feature mapping that is invariant across domains upon successful training while being beneficial to the main classification task.

4 Experiments

4.1 Multi-Domain Text Classification

In this experiment, we compare MAN to state-of-the-art MDTC systems on the multi-domain Amazon review dataset (Blitzer et al., 2007), which is one of the most widely used MDTC datasets. Note that this dataset was already preprocessed into a bag of features (unigrams and bigrams), losing all word order information. This prohibits the use of CNNs or RNNs as feature extractors, limiting the potential performance of the system. Nonetheless, we adopt the same dataset for fair comparison and employ a MLP as our feature extractor. In particular, we take the 5000 most frequent features and represent each review as a 5000d feature vector, where feature values are raw counts of the fea-

tures. Our MLP feature extractor would then have an input size of 5000 in order to process the reviews.

The Amazon dataset contains 2000 samples for each of the four domains: *book, DVD, electronics*, and *kitchen*, with binary labels (positive, negative). Following Wu and Huang (2015), we conduct 5-way cross validation. Three out of the five folds are treated as the training set, one serves as the validation set, while the remaining is the test set. The 5-fold average test accuracy is reported.

Table 1 shows the main results. Three types of models are shown: *Domain-Specific Models Only*, where only in-domain models are trained[5]; *Shared Model Only*, where a single model is trained with all data; and *Shared-Private Models*, a combination of the previous two. Within each category, various architectures are examined, such as Least Square (LS), SVM, and Logistic Regression (LR). As explained before, we use MLP as our feature extractors for all our models (bold ones). Among our models, the ones with the MAN prefix use adversarial training, and MAN-L2 and MAN-NLL indicate MAN with the L2 loss and the NLL loss, respectively.

From Table 1, we can see that by adopting modern deep neural networks, our methods achieve superior performance within the first two model categories even without adversarial training. This is corroborated by the fact that our SP-MLP model performs comparably to CMSC, while the latter relies on external resources such as sentiment lexica. Moreover, when our multinomial adversarial nets are introduced, further improvement is observed. With both loss functions, MAN outperforms all Shared-Private baseline systems on each domain, and achieves statistically significantly higher overall performance. For our MAN-SP models, we provide the mean accuracy as well as the standard errors over five runs, to illustrate the performance variance and conduct significance tests. It can be seen that MAN's performance is relatively stable, and consistently outperforms CMSC.

4.2 Experiments for Unlabeled Domains

As CMSC requires labeled data for each domain, their experiments were naturally designed this way. In reality, however, many domains may not

---

Target Domain	Book	DVD	Elec.	Kit.	Avg.
MLP	76.55	75.88	84.60	85.45	80.46
mSDA[1]	76.98	78.61	81.98	84.26	80.46
DANN[2]	77.89	78.86	84.91	86.39	82.01
MDAN (H-MAX)[3]	78.45	77.97	84.83	85.80	81.76
MDAN (S-MAX)[3]	**78.63**	80.65	**85.34**	86.26	**82.72**
MAN-L2-SP-MLP	78.45	81.57	83.37	85.57	82.24
MAN-NLL-SP-MLP	77.78	**82.74**	83.75	**86.41**	82.67

[1] Chen et al. (2012)
[2] Ganin et al. (2016)
[3] Zhao et al. (2017)

Table 2: Results on unlabeled domains. Models in bold are our models while the rest is taken from Zhao et al. (2017). Highest domain performance is shown in bold.

have any annotated corpora available. It is therefore also important to look at the performance in these unlabeled domains for a MDTC system. Fortunately, as depicted before, MAN's adversarial training only utilizes unlabeled data from each domain to learn the domain-invariant features, and can thus be used on unlabeled domains as well. During testing, only the shared feature vector is fed into $\mathcal{C}$, while the domain feature vector is set to **0**.

In order to validate MAN's effectiveness, we compare to state-of-the-art *multi-source domain adaptation* (MS-DA) methods (see §6). Compared to standard domain adaptation methods with one source and one target domain, MS-DA allows the adaptation from multiple source domains to a single target domain. Analogically, MDTC can be viewed as *multi-source multi-target* domain adaptation, which is superior when multiple target domains exist. With multiple target domains, MS-DA will need to treat each one as an independent task, which is more expensive and cannot utilize the unlabeled data in other target domains.

In this work, we compare MAN with one recent MS-DA method, MDAN (Zhao et al., 2017). Their experiments only have one target domain to suit their approach, and we follow this setting for fair comparison. However, it is worth noting that MAN is designed for the MDTC setting, and can deal with multiple target domains at the same time, which can potentially improve the performance by taking advantage of more unlabeled data from multiple target domains during adversarial training. We adopt the same setting as Zhao et al. (2017), which is based on the same multi-domain Amazon review dataset. Each of the four domains in the dataset is treated as the target domain in four separate experiments, while the re-

	books	elec.	dvd	kitchen	apparel	camera	health	music	toys	video	baby	magaz.	softw.	sports	IMDb	MR	Avg.
Domain-Specific Models Only																	
BiLSTM	81.0	78.5	80.5	81.2	86.0	86.0	78.7	77.2	84.7	83.7	83.5	91.5	85.7	84.0	85.0	74.7	82.6
CNN	85.3	87.8	76.3	84.5	86.3	89.0	87.5	81.5	87.0	82.3	82.5	86.8	87.5	85.3	83.3	75.5	84.3
Shared Model Only																	
FS-MTL	82.5	85.7	83.5	86.0	84.5	86.5	88.0	81.2	84.5	83.7	88.0	92.5	86.2	85.5	82.5	74.7	84.7
MAN-L2-CNN	**88.3**	88.3	87.8	88.5	85.3	90.5	**90.8**	85.3	89.5	89.0	89.5	91.3	88.3	89.5	**88.5**	73.8	87.7
MAN-NLL-CNN	88.0	87.8	87.3	88.5	86.3	90.8	89.8	84.8	89.3	89.3	87.8	91.8	90.0	**90.3**	87.3	73.5	87.6
Shared-Private Models																	
ASP-MTL	84.0	86.8	85.5	86.2	87.0	89.2	88.2	82.5	88.0	84.5	88.2	92.2	87.2	85.7	85.5	**76.7**	86.1
MAN-L2-SP-CNN	87.6* (0.2)	87.4 (1.0)	88.1* (0.4)	89.8* (0.4)	**87.6** (0.7)	**91.4*** (0.4)	89.8* (0.3)	**85.9*** (0.1)	90.0* (0.1)	89.5* (0.2)	90.0 (0.6)	92.5 (0.5)	90.4* (0.4)	89.0* (0.4)	86.6 (0.5)	76.1 (0.5)	88.2* (0.1)
MAN-NLL-SP-CNN	86.8* (0.4)	**88.8** (0.6)	**88.6*** (0.4)	**89.9*** (0.4)	**87.6** (0.4)	90.7 (0.4)	89.4 (0.3)	85.5* (0.1)	**90.4*** (0.2)	**89.6*** (0.3)	**90.2** (0.6)	**92.9** (0.4)	**90.9*** (0.7)	89.0* (0.2)	**87.0*** (0.1)	**76.7** (0.8)	**88.4*** (0.1)

Table 3: Results on the FDU-MTL dataset. Bolded models are ours, while the rest is from Liu et al. (2017). Highest performance is each domain is highlighted. For our full MAN models, standard errors are shown in parenthese and statistical significance ($p < 0.01$) over ASP-MTL is indicated by *.

maining three are used as source domains.

In Table 2, the target domain is shown on top, and the test set accuracy is reported for various systems. It shows that MAN outperforms several baseline systems, such as a MLP trained on the source-domains, as well as single-source domain adaptation methods such as mSDA (Chen et al., 2012) and DANN (Ganin et al., 2016), where the training data in the multiple source domains are combined and viewed as a single domain. Finally, when compared to MDAN, MAN and MDAN each achieves higher accuracy on two out of the four target domains, and the average accuracy of MAN is similar to MDAN. Therefore, MAN achieves competitive performance for the domains without annotated corpus. Nevertheless, unlike MS-DA methods, MAN can handle multiple target domains at one time.

4.3 Experiments on the MTL Dataset

To make fair comparisons, the previous experiments follow the standard settings in the literature, where the widely adopted Amazon review dataset is used. However, this dataset has a few limitations. First, it has only four domains. In addition, the reviews are already tokenized and converted to a bag of features consisting of unigrams and bigrams. Raw review texts are hence not available in this dataset, making it impossible to use certain modern neural architectures such as CNNs and RNNs. To provide more insights on how well MAN works with other feature extractor architectures, we provide a third set of experiments on the FDU-MTL dataset (Liu et al., 2017). This dataset is created as a multi-task learning dataset with 16 *tasks*, where each task is essentially a different domain of reviews. It has 14 Amazon domains: books, elec-

tronics, DVD, kitchen, apparel, camera, health, music, toys, video, baby, magazine, software, and sports, in addition to two movie review domains from the IMDb and the MR datasets. Each domain has a development set of 200 samples, and a test set of 400 samples. The amount of training and unlabeled data vary across domains but are roughly 1400 and 2000, respectively.

We compare MAN with ASP-MTL (Liu et al., 2017) on this FDU-MTL dataset. ASP-MTL also adopts adversarial training for learning a shared feature space, and can be viewed as a special case of MAN that adopts the NLL loss (MAN-NLL) and chooses LSTM as their feature extractor. In contrast, we found a CNN-based feature extractor (Kim, 2014) achieves much better accuracy while being ~ 10 times faster. Indeed, as shown in Table 3, with or without adversarial training, our CNN models outperform LSTM ones by a large margin. When used in our MAN framework, we attain the state-of-the-art performance on every domain with a 88.4% overall accuracy, surpassing ASP-MTL by a significant margin of 2.3%.

We hypothesize the reason a LSTM performs much worse than a CNN is its lack of an attention mechanism. In ASP-MTL, only the last hidden unit is taken as the extracted features. While LSTMs are effective for representing the context for each token, it might not be powerful enough for directly encoding the entire document (Bahdanau et al., 2015). Therefore, various attention mechanisms have been introduced on top of the vanilla LSTM to select words (and contexts) most relevant for making the predictions. In our preliminary experiments, we find that a Bi-directional LSTM with the dot-product attention (Luong et al., 2015) yields better performance

than the vanilla LSTM in ASP-MTL. However, it still does not outperform CNN and is much slower. As a result, we conclude that, for text classification tasks, CNN is both effective and efficient in extracting local and higher-level features for making a single categorization.

Finally, we observe that MAN-NLL achieves slightly higher overall performance compared to MAN-L2, providing evidence for the claim in a recent study (Lucic et al., 2017) that the original GAN loss (NLL) may not be inherently inferior to the L2 loss. Moreover, the two variants excel in different domains, suggesting the possibility of further performance gain when using ensemble.

5 Implementation Details

For all three of our experiments, we use $\lambda = 0.05$ and $k = 5$ (See Algorithm 1). For both optimizers, Adam (Kingma and Ba, 2015) is used with learning rate 0.0001. The size of the shared feature vector is set to 128 while that of the domain feature vector is 64. Dropout of $p = 0.4$ is used in all components. $\mathcal{C}$ and $\mathcal{D}$ each has one hidden layer of the same size as their input ($128 + 64$ for $\mathcal{C}$ and 128 for $\mathcal{D}$). ReLU is used as the activation function. Batch normalization (Ioffe and Szegedy, 2015) is used in both $\mathcal{C}$ and $\mathcal{D}$ but not $\mathcal{F}$. We use a batch size of 8.

For our first two experiments on the Amazon review dataset, the MLP feature extractor is used. As described in §4.1, it has an input size of 5000. Two hidden layers are used, with size 1000 and 500, respectively.

For the CNN feature extractor used in the FDU-MTL experiment, a single convolution layer is used. The kernel sizes are 3, 4, and 5, and the number of kernels are 200. The convolution layers take as input the 100d word embeddings of each word in the input sequence. We use *word2vec* word embeddings (Mikolov et al., 2013) trained on a bunch of unlabeled raw Amazon reviews (Blitzer et al., 2007). After convolution, the outputs go through a ReLU layer before fed into a max pooling layer. The pooled output is then fed into a single fully connected layer to be converted into a feature vector of size either 128 or 64. More details of using CNN for text classification can be found in the original paper (Kim, 2014). MAN is implemented using PyTorch (Paszke et al., 2017).

6 Related Work

Multi-Domain Text Classification The MDTC task was first examined by Li and Zong (2008), who proposed to fuse the training data from multiple domains either at the feature level or the classifier level. The prior art of MDTC (Wu and Huang, 2015) decomposes the text classifier into a general one and a set of domain-specific ones. However, the general classifier is learned by parameter sharing and domain-specific knowledge may sneak into it. They also require external resources to help improve accuracy and compute domain similarities.

Domain Adaptation Domain Adaptation attempts to transfer the knowledge from a source domain to a target one, and the traditional form is the *single-source, single-target* (**SS,ST**) adaptation (Blitzer et al., 2006). Another variant is the **SS,MT** adaptation (Yang and Eisenstein, 2015), which tries to simultaneously transfer the knowledge to multiple target domains from a single source. However, it cannot fully take advantage the training data if it comes from multiple source domains. **MS,ST** adaptation (Mansour et al., 2009; Zhao et al., 2017) can deal with multiple source domains but only transfers to a single target domain. Therefore, when multiple target domains exist, they need to treat them as independent problems, which is more expensive and cannot utilize the additional unlabeled data in these domains. Finally, MDTC can be viewed as **MS,MT** adaptation, which is arguably more general and realistic.

Adversarial Networks The idea of adversarial networks was proposed by Goodfellow et al. (2014) for image generation, and has been applied to various NLP tasks as well (Chen et al., 2016; Yu et al., 2017). Ganin et al. (2016) first used it for the **SS,ST** domain adaptation followed by many others. Bousmalis et al. (2016) utilized adversarial training in a shared-private model for domain adaptation to learn domain-invariant features, but still focused on the **SS,ST** setting. Finally, the idea of using adversarial nets to discriminate over multiple distributions was empirically explored by a very recent work (Liu et al., 2017) under the multi-task learning setting, and can be considered as a special case of our MAN framework with the NLL domain loss. We propose MAN as a more general framework with alternative architectures for the adversarial component, and for the first time

provide theoretical justifications the multinomial adversarial nets. Moreover, Liu et al. (2017) used a LSTM without attention as their feature extractor, which we found to perform sub-optimal in the experiments. We instead chose Convolutional Neural Nets as our feature extractor that achieves higher accuracy while running an order of magnitude faster (see §4.3).

7 Conclusion

In this work, we propose a family of Multinomial Adversarial Networks (MANs) that generalize the traditional binomial adversarial nets in the sense that MAN can simultaneously minimize the difference among multiple probability distributions instead of just two. We provide theoretical justifications for two instances of MAN, MAN-NLL and MAN-L2, showing they are minimizers of two different f-divergence metrics among multiple distributions, respectively. This indicates MAN can be used to make multiple distributions indistinguishable from one another. It can hence be applied to a variety of tasks, similar to the versatile binomial adversarial nets, which have been used in many areas for making *two* distributions alike.

In this paper, we design a MAN model for the MDTC task, following the shared-private paradigm that has a shared feature extractor to learn domain-invariant features and domain feature extractors to learn domain-specific ones. MAN is used to enforce the shared feature extractor to learn only domain-invariant knowledge, by resorting to MAN's power of making indistinguishable the shared feature distributions of samples from each domain. We conduct extensive experiments, demonstrating our MAN model outperforms the prior art systems in MDTC, and achieves state-of-the-art performance on domains without labeled data when compared to multi-source domain adaptation methods.

Acknowledgments

References

S. M. Ali and S. D. Silvey. 1966. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Methodological)* 28(1):131–142. http://www.jstor.org/stable/2984279.

Javed A. Aslam and Virgil Pavlu. 2007. Query hardness estimation using jensen-shannon divergence among multiple scoring functions. In *Advances in Information Retrieval*. Springer Berlin Heidelberg, Berlin, Heidelberg, pages 198–209. https://doi.org/10.1007/978-3-540-71496-5_20.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations (ICLR 2015)*. http://arxiv.org/abs/1409.0473.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics, Prague, Czech Republic, pages 440–447. http://www.aclweb.org/anthology/P07-1056.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Sydney, Australia, pages 120–128. http://www.aclweb.org/anthology/W/W06/W06-1615.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. In *Advances in Neural Information Processing Systems 29*, Curran Associates, Inc., pages 343–351. http://papers.nips.cc/paper/6254-domain-separation-networks.pdf.

Minmin Chen, Zhixiang Xu, Kilian Q. Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, Omnipress, USA, ICML'12, pages 1627–1634. http://dl.acm.org/citation.cfm?id=3042573.3042781.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2016. Adversarial deep averaging networks for cross-lingual sentiment classification. *Computing Research Repository* arXiv:1606.01614. https://arxiv.org/abs/1606.01614.

Imre Csiszar and Janos Korner. 1982. *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Academic Press, Inc., Orlando, FL, USA. https://dl.acm.org/citation.cfm?id=601016.

Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi–task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, KDD '04, pages 109–117. https://doi.org/10.1145/1014052.1014067.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research* 17:1–35. http://dl.acm.org/citation.cfm?id=2946645.2946704.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, Curran Associates, Inc., pages 2672–2680. http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*. pages 448–456. http://jmlr.org/proceedings/papers/v37/ioffe15.pdf.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, pages 1681–1691. https://doi.org/10.3115/v1/P15-1162.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pages 1746–1751. https://doi.org/10.3115/v1/D14-1181.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations 2015*. https://arxiv.org/abs/1412.6980.

Shoushan Li and Chengqing Zong. 2008. Multi-domain sentiment classification. In *Proceedings of ACL-08: HLT, Short Papers*. Association for Computational Linguistics, Columbus, Ohio, pages 257–260. http://www.aclweb.org/anthology/P/P08/P08-2065.

Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory* 37(1):145–151. https://doi.org/10.1109/18.61115.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 1–10. https://doi.org/10.18653/v1/P17-1001.

Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. 2017. Are GANs Created Equal? A Large-Scale Study. *Computing Research Repository* arXiv:1711.10337. http://arxiv.org/abs/1711.10337.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1412–1421. https://doi.org/10.18653/v1/D15-1166.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2009. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems 21*, Curran Associates, Inc., pages 1041–1048. http://papers.nips.cc/paper/3550-domain-adaptation-with-multiple-sources.pdf.

Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least squares generative adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*. pages 2813–2821. https://doi.org/10.1109/ICCV.2017.304.

Andrew McCallum, Kamal Nigam, et al. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*. Madison, WI, USA, volume 752, pages 41–48. http://www.aaai.org/Papers/Workshops/1998/WS-98-05/WS98-05-007.pdf.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations 2013 Workshop*. https://arxiv.org/abs/1301.3781.

Frank Nielsen and Richard Nock. 2014. On the chi square and higher-order chi distances for approximating f-divergences. *IEEE Signal Processing Letters* 21(1):10–13. https://doi.org/10.1109/LSP.2013.2288355.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. 2016. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., pages 271–279. http://papers.nips.cc/paper/6066-f-gan-training-generative-neural-samplers-using-variational-divergence-minimization.pdf.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. *NIPS 2017 Autodiff Workshop* https://openreview.net/pdf?id=BJJsrmfCZ.

Fangzhao Wu and Yongfeng Huang. 2015. Collaborative multi-domain sentiment classification. In *2015 IEEE International Conference on Data Mining*. pages 459–468. https://doi.org/10.1109/ICDM.2015.68.

Yi Yang and Jacob Eisenstein. 2015. Unsupervised multi-domain adaptation with feature embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 672–682. https://doi.org/10.3115/v1/N15-1069.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI Conference on Artificial Intelligence*. https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14344.

Han Zhao, Shanghang Zhang, Guanhang Wu, João P. Costeira, José M. F. Moura, and Geoffrey J. Gordon. 2017. Multiple source domain adaptation with adversarial training of neural networks. *Computing Research Repository* arXiv:1705.09684. http://arxiv.org/abs/1705.09684.

J. Zhou, J. Chen, and J. Ye. 2011. *MALSAR: Multi-tAsk Learning via StructurAl Regularization*. Arizona State University. http://www.public.asu.edu/~jye02/Software/MALSAR.

Appendix A Proofs

A.1 Proofs for MAN-NLL

Assume we have N domains, consider the distribution of the shared features $\mathcal{F}_s$ for instances in each domain d_i:

$$P_i(\mathbf{f}) \triangleq P(\mathbf{f} = \mathcal{F}_s(x)|x \in d_i)$$

The objective that $\mathcal{D}$ attempts to minimize is:

$$J_{\mathcal{D}} = -\sum_{i=1}^{N} \mathop{\mathbb{E}}_{\mathbf{f} \sim P_i} [\log \mathcal{D}_i(\mathbf{f})] \tag{12}$$

where $\mathcal{D}_i(\mathbf{f})$ is the i-th dimension of $\mathcal{D}$'s output vector, which conceptually corresponds to the softmax probability of $\mathcal{D}$ predicting that $\mathbf{f}$ is from domain d_i. We therefore have property that for any $\mathbf{f}$:

$$\sum_{i=1}^{N} \mathcal{D}_i(\mathbf{f}) = 1 \tag{13}$$

Lemma 1. *For any fixed $\mathcal{F}_s$, the optimum domain discriminator $\mathcal{D}^*$ is:*

$$\mathcal{D}_i^*(\mathbf{f}) = \frac{P_i(\mathbf{f})}{\sum_{j=1}^{N} P_j(\mathbf{f})} \tag{14}$$

Proof. For a fixed $\mathcal{F}_s$, the optimum

$$\mathcal{D}^* = \arg\min_{\mathcal{D}} J_{\mathcal{D}} = \arg\min_{\mathcal{D}} -\sum_{i=1}^{N} \mathop{\mathbb{E}}_{\mathbf{f} \sim P_i} [\log \mathcal{D}_i(\mathbf{f})]$$

$$= \arg\max_{\mathcal{D}} \sum_{i=1}^{N} \int_{\mathbf{f}} P_i(\mathbf{f}) \log \mathcal{D}_i(\mathbf{f}) d\mathbf{f}$$

$$= \arg\max_{\mathcal{D}} \int_{\mathbf{f}} \sum_{i=1}^{N} P_i(\mathbf{f}) \log \mathcal{D}_i(\mathbf{f}) d\mathbf{f}$$

We employ the Lagrangian Multiplier to derive $\arg\max_{\mathcal{D}} \sum_{i=1}^{N} P_i(\mathbf{f}) \log \mathcal{D}_i(\mathbf{f})$ under the constraint of (13). Let

$$L(\mathcal{D}_1, \dots, \mathcal{D}_N, \lambda) = \sum_{i=1}^{N} P_i \log \mathcal{D}_i - \lambda(\sum_{i=1}^{N} \mathcal{D}_i - 1)$$

Let $\nabla L = 0$:

$$\begin{cases} \nabla_{\mathcal{D}_i} \sum_{j=1}^{N} P_j \log \mathcal{D}_j = \lambda \nabla_{\mathcal{D}_i} (\sum_{j=1}^{N} \mathcal{D}_j - 1) & (\forall i) \\ \sum_{i=1}^{N} \mathcal{D}_i - 1 = 0 \end{cases}$$

Solving the two equations, we have:

$$\mathcal{D}_i^*(\mathbf{f}) = \frac{P_i(\mathbf{f})}{\sum_{j=1}^{N} P_j(\mathbf{f})}$$

$\square$

On the other hand, the loss function of the shared feature extractor $\mathcal{F}_s$ consists of two additive components, the loss from the text classifier $\mathcal{C}$, and the loss from the domain discriminator $\mathcal{D}$:

$$J_{\mathcal{F}_s} = J_{\mathcal{F}_s}^{\mathcal{C}} + \lambda J_{\mathcal{F}_s}^{\mathcal{D}} \triangleq J_{\mathcal{C}} - \lambda J_{\mathcal{D}} \tag{15}$$

We have the following theorem for the domain loss for $\mathcal{F}_s$:

Theorem 1. *When $\mathcal{D}$ is trained to its optimality:*

$$J^D_{\mathcal{F}_s} = -J_{\mathcal{D}^*} = -N \log N + N \cdot JSD(P_1, P_2, \ldots, P_N) \tag{16}$$

where $JSD(\cdot)$ is the generalized Jensen-Shannon Divergence (Lin, 1991) among multiple distributions.

Proof. Let $\overline{P} = \frac{\sum_{i=1}^N P_i}{N}$.

There are two equivalent definitions of the generalized Jensen-Shannon divergence: the original definition based on Shannon entropy (Lin, 1991), and a reshaped one expressed as the average Kullback-Leibler divergence of each P_i to the centroid $\overline{P}$ (Aslam and Pavlu, 2007). We adopt the latter one here:

$$JSD(P_1, P_2, \ldots, P_N) \triangleq \frac{1}{N} \sum_{i=1}^N KL(P_i \| \overline{P}) = \frac{1}{N} \sum_{i=1}^N \mathbb{E}_{\mathbf{f} \sim P_i} \left[\log \frac{P_i(\mathbf{f})}{\overline{P}(\mathbf{f})} \right] \tag{17}$$

Now substituting $\mathcal{D}^*$ into $J^{\mathcal{D}}_{\mathcal{F}_s}$:

$$
\begin{aligned}
J^{\mathcal{D}}_{\mathcal{F}_s} = -J_{\mathcal{D}^*} &= \sum_{i=1}^N \mathbb{E}_{\mathbf{f} \sim P_i} \left[\log \mathcal{D}_i^*(\mathbf{f}) \right] \\
&= \sum_{i=1}^N \mathbb{E}_{\mathbf{f} \sim P_i} \left[\log \frac{P_i(\mathbf{f})}{\sum_{j=1}^N P_j(\mathbf{f})} \right] \\
&= -N \log N + \sum_{i=1}^N \mathbb{E}_{\mathbf{f} \sim P_i} \left[\log \frac{P_i(\mathbf{f})}{\sum_{j=1}^N P_j(\mathbf{f})} + \log N \right] \\
&= -N \log N + \sum_{i=1}^N \mathbb{E}_{\mathbf{f} \sim P_i} \left[\log \frac{P_i(\mathbf{f})}{\frac{\sum_{j=1}^N P_j(\mathbf{f})}{N}} \right] \\
&= -N \log N + \sum_{i=1}^N \mathbb{E}_{\mathbf{f} \sim P_i} \left[\log \frac{P_i(\mathbf{f})}{\overline{P}} \right] \\
&= -N \log N + \sum_{i=1}^N KL(P_i \| \overline{P}) \\
&= -N \log N + N \cdot JSD(P_1, P_2, \ldots, P_N)
\end{aligned}
$$

$\square$

Consequently, by the non-negativity of JSD (Lin, 1991), we have the following corollary:

Corollary 1. *The optimum of $J^{\mathcal{D}}_{\mathcal{F}_s}$ is $-N \log N$, and is achieved if and only if $P_1 = P_2 = \cdots = P_N = \overline{P}$.*

A.2 Proofs for MAN-L2

The proof is similar for MAN with the L2 loss. The loss function used by $\mathcal{D}$ is, for a sample from domain d_i with shared feature vector $\mathbf{f}$:

$$\mathcal{L}_{\mathcal{D}}(\mathcal{D}(\mathbf{f}), i) = \sum_{j=1}^N (\mathcal{D}_j(\mathbf{f}) - \mathbb{1}_{\{i=j\}})^2 \tag{18}$$

So the objective that $\mathcal{D}$ minimizes is:

$$J_{\mathcal{D}} = \sum_{i=1}^N \mathbb{E}_{\mathbf{f} \sim P_i} \left[\sum_{j=1}^N (\mathcal{D}_j(\mathbf{f}) - \mathbb{1}_{\{i=j\}})^2 \right] \tag{19}$$

For simplicity, we further constrain $\mathcal{D}$'s outputs to be on a simplex:

$$\sum_{i=1}^{N} \mathcal{D}_i(\mathbf{f}) = 1 \quad (\forall \mathbf{f}) \tag{20}$$

Lemma 2. *For any fixed $\mathcal{F}_s$, the optimum domain discriminator $\mathcal{D}^*$ is:*

$$\mathcal{D}_i^*(\mathbf{f}) = \frac{P_i(\mathbf{f})}{\sum_{j=1}^{N} P_j(\mathbf{f})} \tag{21}$$

Proof. For a fixed $\mathcal{F}_s$, the optimum

$$\begin{aligned}
\mathcal{D}^* = \arg\min_{\mathcal{D}} J_{\mathcal{D}} &= \arg\min_{\mathcal{D}} \sum_{i=1}^{N} \mathop{\mathbb{E}}_{\mathbf{f} \sim P_i} \left[\mathcal{L}_{\mathcal{D}}(\mathcal{D}(\mathbf{f}), i) \right] \\
&= \arg\min_{\mathcal{D}} \sum_{i=1}^{N} \int_{\mathbf{f}} P_i(\mathbf{f}) \mathcal{L}_{\mathcal{D}}(\mathcal{D}(\mathbf{f}), i) d\mathbf{f} \\
&= \arg\min_{\mathcal{D}} \int_{\mathbf{f}} \sum_{i=1}^{N} P_i(\mathbf{f}) \sum_{j=1}^{N} (\mathcal{D}_j(\mathbf{f}) - \mathbb{1}_{\{i=j\}})^2 d\mathbf{f}
\end{aligned}$$

Similar to MAN-NLL, we employ the Lagrangian Multiplier to derive $\arg\max_{\mathcal{D}} \sum_{i=1}^{N} P_i(\mathbf{f}) \sum_{j=1}^{N} (\mathcal{D}_j(\mathbf{f}) - \mathbb{1}_{\{i=j\}})^2$ under the constraint of (20). Let $\nabla L = 0$:

$$\begin{cases} 2((\sum_{j=1}^{N} P_j)\mathcal{D}_i - P_i) = \lambda \quad (\forall i) \\ \sum_{i=1}^{N} \mathcal{D}_i - 1 = 0 \end{cases}$$

Solving the two equations, we have $\lambda = 0$ and:

$$\mathcal{D}_i^*(\mathbf{f}) = \frac{P_i(\mathbf{f})}{\sum_{j=1}^{N} P_j(\mathbf{f})}$$

$\square$

For the domain loss of $\mathcal{F}_s$:

Theorem 2. *Let $\overline{P} = \frac{\sum_{i=1}^{N} P_i}{N}$. When $\mathcal{D}$ is trained to its optimality:*

$$\begin{aligned}
J_{\mathcal{F}_s}^{\mathcal{D}} &= \sum_{i=1}^{N} \mathop{\mathbb{E}}_{\mathbf{f} \sim P_i} \left[\sum_{j=1}^{N} (D_j(\mathbf{f}) - \frac{1}{N})^2 \right] \\
&= \frac{1}{N} \sum_{i=1}^{N} \chi_{Neyman}^2 (P_i \| \overline{P})
\end{aligned} \tag{22}$$

where $\chi_{Neyman}^2(\cdot \| \cdot)$ is the Neyman χ^2 divergence (Nielsen and Nock, 2014).

1239

Proof. Substituting $\mathcal{D}^*$ into $\mathcal{L}^{\mathcal{D}}_{\mathcal{F}_s}$:

$$
\begin{aligned}
J^{\mathcal{D}}_{\mathcal{F}_s} &= \sum_{i=1}^{N} \mathop{\mathbb{E}}_{\mathbf{f} \sim P_i} \left[\sum_{j=1}^{N} (D_j^*(\mathbf{f}) - \frac{1}{N})^2 \right] \\
&= \sum_{i=1}^{N} \int_{\mathbf{f}} P_i \sum_{j=1}^{N} (\frac{P_j}{N\overline{P}} - \frac{1}{N})^2 d\mathbf{f} \\
&= \int_{\mathbf{f}} \sum_{i=1}^{N} \sum_{j=1}^{N} P_i (\frac{P_j}{N\overline{P}} - \frac{1}{N})^2 d\mathbf{f} \\
&= \frac{1}{N^2} \sum_{j=1}^{N} \int_{\mathbf{f}} \sum_{i=1}^{N} P_i (\frac{P_j}{\overline{P}} - 1)^2 d\mathbf{f} \\
&= \frac{1}{N^2} \sum_{j=1}^{N} \int_{\mathbf{f}} N\overline{P} (\frac{P_j}{\overline{P}} - 1)^2 d\mathbf{f} \\
&= \frac{1}{N} \sum_{j=1}^{N} \int_{\mathbf{f}} \frac{(P_j - \overline{P})^2}{\overline{P}} d\mathbf{f} \\
&= \frac{1}{N} \sum_{i=1}^{N} \chi^2_{Neyman}(P_i \| \overline{P})
\end{aligned}
$$

$\square$

Finally, by the joint convexity of f-divergence, we have the following corollary:

Corollary 2.

$$
\begin{aligned}
\mathcal{L}^{\mathcal{D}}_{\mathcal{F}_s} &= \frac{1}{N} \sum_{i=1}^{N} \chi^2_{Neyman}(P_i \| \overline{P}) \\
&\geq \chi^2_{Neyman}(\frac{1}{N} \sum_{i=1}^{N} P_i \| \frac{1}{N} \sum_{i=1}^{N} \overline{P}) \\
&= \chi^2_{Neyman}(\overline{P} \| \overline{P}) = 0
\end{aligned}
$$

and the equality is attained if and only if $P_1 = P_2 = \cdots = P_N = \overline{P}$.