Distance Based Source Domain Selection for Sentiment Classification

LEX ELIAS RAZOUX SCHULTZ, MARCO LOOG, PEYMAN MOHAJERIN ESFAHANI

Abstract. Automated sentiment classification (SC) on short text fragments has received increasing attention in recent years. Performing SC on unseen domains with few or no labeled samples can significantly affect the classification performance due to different expression of sentiment in source and target domain. In this study, we aim to mitigate this undesired impact by proposing a methodology based on a predictive measure, which allows us to select an optimal source domain from a set of candidates. The proposed measure is a linear combination of well-known distance functions between probability distributions supported on the source and target domains (e.g. Earth Mover's distance and Kullback-Leibler divergence). The performance of the proposed methodology is validated through an SC case study in which our numerical experiments suggest a significant improvement in the cross domain classification error in comparison with a random selected source domain for both a naive and adaptive learning setting. In the case of more heterogeneous datasets, the predictability feature of the proposed model can be utilized to further select a subset of candidate domains, where the corresponding classifier outperforms the one trained on all available source domains. This observation reinforces a hypothesis that our proposed model may also be deployed as a means to filter out redundant information during a training phase of SC.

1. Introduction

Automated SC is performed when a trained classifier is used to label documents with a sentiment label, based on the content of the document. In practice, a document represents a review, tweet or other small textual opinion expression and ought to be classified as either positive or negative (binary SC). Allied terminology for SC is semantic orientation or polarity, which indicates the direction a word deviates from the norm of its semantic group [17]. Often, the terms opinion mining and sentiment analysis are used to describe the computational treatment of opinion, sentiment and subjectivity in text [30]. Examples of current applications that benefit from SC are recommendation systems [39], stock market prediction systems [7] and political election predictors [40]. Also in the future, we could imagine SC put in practice to improve man-machine communication such as voice commands or even interaction between robots and humans.

One of the biggest challenges in many SC applications, is the discrepancy between the source domain where the classifier is trained on, and the target domain of interest. Typically, we have large amounts of labeled data from different source domains, but only possess few, unlabeled data from the target domain, both originated from different underlying distributions. In this setting, SC generally performs rather bad compared to inner domain training and testing. This so called domain transfer problem can be intuitively explained due to different fashions of sentiment expression in different domains [38]. A word may have different sentiment polarity with respect to the domains it is used in, or different words may be used to express similar sentiment among different domains [46].

Various research has been performed with the aim to decrease the loss in classification performance due to domain shift. This loss is referred to as the adaptation loss [18] [5]. One of the popular transfer learning techniques to maintain performance of SC when crossing domains, is Structural Correspondence Learning which uses pivot features as link between the source and target domain [6] [22]. Many other techniques are based on optimal transport which matches the conditional probability distribution of the training domain

Date: August 28, 2018.

The authors are with the Delft Center for Systems and Control, TU Delft, The Netherlands and the Patter Recognition Laboratory, TU Delft, The Netherlands.

and target domain by a transformation of the source domain data [10] [20]. The distance between source and target distribution is minimized [9]. Often distance is minimized in a lower dimensional embedded space, for example when performing Subspace Alignment [13] or Transfer Component Analysis [29].

In recent years, the use of neural models for adaptation learning has been proven successful. The ability of a neural model to be trained on both source and unlabeled target data has been leveraged to improve performance of the cross domain classification task. Various approaches are proposed to adapt the neural net towards superior performance of classification in a given target domain. In the field of SC, Domain-Adversarial Neural Networks (DANNs) have been effectively used in such a context [14]. DANNs improve performance by augmenting a feed-forward model with a few standard layers and a new gradient reversal layer. Due to this augmentation, the model promotes features that are discriminative on class label and indiscriminative regarding a shift in domain. Other research is based on the use of Stacked Denoising Auto-encoders (SDA's) [8] [15]. An SDA can be used to learn a higher lever feature extraction by using unlabeled data from the source and target domain. The feature extraction transforms the feature space, and subsequently, a classifier is trained on this transformed feature space. Other popular techniques with much resemblance to SDA's are the *word2vec* [28] and *doc2vec* [24] algorithms.

One may invoke other techniques in combination with the adaptation learning to improve the adaption loss, e.g., unsupervised source domain selection. Instead of focusing on the type of classifier or the underlying representation of the data, these techniques primarily revolve around preselecting an appropriate training dataset. The source domain selection method opts to select one or multiple optimal source domains from a set of candidate domains without supervision, i.e., it selects the source domains that have the best cross domain classification performance compared to the other candidates without the use of class labels of the target domain data. Source domain selection has been successfully deployed in other contexts, for example to improve brain-computer interface calibration [44]. In this case, supervised source domain selection based on distance between the class average vectors has been used. Alternatively, ranking convolutional neural networks trained on difference source domains for a target task shows success when using the mutual information between domains [1].

Source domain selection is a higher level of selection than instance selection as it chooses a set of instances from the same underlying domain rather than individual instances. It is based on the assumption that inner domain information is more informative and therefore yields an improved performance when used for training. That rises the question what constitutes a domain. It is argued that there is no common ground in what constitutes a domain, but in NLP it is typically used to refer tot some coherent data with respect to topic or genre [34]. However, there are numerous other factors that should be taken into consideration, e.g., the medium on which the domain data is exposed, the time spirit, and the purpose of the data.

In order to find an appropriate source domain, it is recently proposed to represent a domain as an average of its document representations in a pre-trained embedded space [36]. Additionally, this study performs same experiments when using denoising auto-encoder representations trained on all source data instead of a pre-trained embedding. They conclude that using a pre-trained embedded space for domain representation is not necessarily recommended. Their approach that relies on denoising auto-encoders is showed to be effective in selecting source domains, but only when large training sets are available.

More common approaches for source domain selection use domain similarity metrics. One of these metrics developed for assessing potential of transfer learning is the $\mathcal{A}$-distance [4]. In the field of source domain selection for SC, Blitzer et al. show a good correlation between the $\mathcal{A}$-distance and adaptation loss [5]. The adaptation loss is measured using the target domain's inner domain classification error which unfortunately requires labeled data of the target domain. The concept of $\mathcal{A}$-distance is also used in recent research for orthophoto classification, were it is approximated by the Maximum Mean Discrepancy [43]. For a more open search space for a source domain, techniques have been researched to find a suitable source domain in open online information sources such as Wikipedia [45]. However, this technique also uses labeled information of the target domain.

Another similarity metric used in source domain selection is the Kullback-Leibler divergence (KL) [23]. This metric may not be well defined in SC, and therefore the data requires smoothing. Alternatively, approximations to KL, such as the Jensen-Shannon divergence [26] or the skew divergence [25] can be used to determine distance between domains. These divergences indicators have proven their ability to select a suitable source domain and thus improving the classification performance of cross domain classification tasks [35].

In this article, we propose a new method, CMEK, for source domain selection that does not require any labeled information of the target domain. The CMEK method trains a source domain selection model by means of a set of given distance metrics. This approach differs from other state-of-the art approaches as it proposes a linear combination of distance metrics where weights are determined by training on the available source domain instead of a single metric for source domain selection. We emphasize that the proposed technique here can be utilized in combination with adaptation learning in a sequential manner, which potentially improves the performance of our domain transformation task.

In Section 2, we define the problem of source domain selection in general, that is, not in context of SC. Section 3 proposes an approach to predict the performance of a classifier trained on a source domain, and applies it on a target domain of interest of which the labels are unknown. This prediction is based on statistical distances between the distributions of the source and target data. To this end, we leverage several statistical measures to determine the distance. Section 4 addresses certain limitations to our proposed model, starting with fundamental limitations that occur in all classification tasks and ending with limitations of our approach within the task of SC. In Section 5 we present background knowledge on how to perform sentiment classification by using a bag-of-words model, as we do in our experiments. Section 6 describes the experimental setup and model settings for two different types of datasets; a homogeneous one where the datasets share similar contents and a heterogeneous one that contains datasets from very diverse topics and sources. In this article, our homogeneous set used is the DRANZIERA benchmark dataset [12]. In addition, we describe how we run experiments in an adaptation learning setting. Section 7 reports the numerical results in detail, and finally in Section 8 we conclude with our reflection on the results and final messages along with suggestions for future work.

2. Problem Definition

With a document represented by a random variable X and its sentiment label by random variable Y, each domain is uniquely characterized by its joint probability density function supported on $\mathbb{X} \times \mathbb{Y}$. Let us then define two underlying distributions to the evaluated source and target data on $\mathbb{X} \times \mathbb{Y}$, denoted by $\mathbb{P}$ and $\bar{\mathbb{P}}$ respectively. Their marginals on $\mathbb{X}$, are denoted as $\mathbb{P}_x$ and $\bar{\mathbb{P}}_x$ respectively. In machine learning, a realization of $\mathbb{P}$ is referred to as labeled data whereas realizations from $\mathbb{P}_x$ are referred to as unlabeled data.

We define an hypothesis function as a mapping from $\mathbb{X}$ to $\mathbb{Y}$, $h : \mathbb{X} \to \mathbb{Y}$. In the context of classification problems in the machine learning literature, the set $\mathbb{Y}$ is often discrete, often binary, and the hypothesis h is often referred to as a classifier as it is able to assign a class label in $\mathbb{Y}$ to each data point in $\mathbb{X}$. The true hypothesis function $h^\star_{\mathbb{P}}$ for a domain characterized by $\mathbb{P}$, is defined as

$$h^\star_{\mathbb{P}} := \arg\min_h \operatorname{Prob}\big(h(x) \neq y \mid (x, y) \sim \mathbb{P}\big). \tag{1}$$

We then define the inner domain classification error as the probability of misclassification in the same domain as the true hypothesis function is constructed in. And the cross domain classification error is defined as the probability of misclassification in the target domain of the true hypothesis function of the source domain

$$\begin{aligned}\xi(\mathbb{P}) &:= \mathrm{Prob}\big(h^\star_{\mathbb{P}}(x) \neq y \mid (x,y) \sim \mathbb{P}\big),\\ \xi(\mathbb{P},\bar{\mathbb{P}}) &:= \mathrm{Prob}\big(h^\star_{\mathbb{P}}(x) \neq y \mid (x,y) \sim \bar{\mathbb{P}}\big).\end{aligned} \tag{2}$$

Note that due to the stochastic nature of sentiment expression, $\xi(\mathbb{P})$ and $\xi(\mathbb{P},\bar{\mathbb{P}})$ are not likely to be zero, even if we would have infinite data to train on. In addition, due to the difference in source and target distribution, $\xi(\mathbb{P},\bar{\mathbb{P}})$ is likely to be higher than $\xi(\mathbb{P})$ [5]. The difference between the cross and inner domain classification error is referred to as the adaptation loss.

The goal in many classification tasks is to minimize the cross domain classification error (2) for a specific target domain while we have access to a finite set of labeled candidate source domains.

An appropriate choice of the source domain in this context is the main goal of this study, leading to the following question:

What source domain minimizes the cross domain classification error for a given target domain?

With the choice of source domain restricted to a domain characterized by a probability density function in the candidate set $\mathcal{P}$, our main goal can be formally described through the optimization program

$$\mathbb{P}^\star := \arg\min_{\mathbb{P}\in\mathcal{P}} \xi(\mathbb{P},\bar{\mathbb{P}}). \tag{3}$$

where $\xi(\mathbb{P},\bar{\mathbb{P}})$ is the cross domain classification error introduced in (2). In other words, when we posses a finite number of labelled datasets, each originated from a unique source domain, on which set should we train our classifier in order to minimize classification error on data from a specific target domain?

The challenge concerning the objective (3), is that for a target domain of interest we typically only have unlabeled data, therefore we cannot calculate the cross domain classification error. This means we only know the marginal distribution $\bar{\mathbb{P}}_x$ of our target domain instead of $\bar{\mathbb{P}}$, making it impossible to explicitly calculate $\xi(\mathbb{P},\bar{\mathbb{P}})$. In order to find the best source domain characterized by $\mathbb{P}^\star \in \mathcal{P}$ for our target domain with distribution $\bar{\mathbb{P}}$, we ought to predict $\xi(\mathbb{P},\bar{\mathbb{P}})$ for all $\mathbb{P} \in \mathcal{P}$. For this prediction we have full distributional information of the source domains, $\mathbb{P}$, and marginal information on the distribution of the target domain, $\bar{\mathbb{P}}_x$, available.

3. Method

In this section, we propose a model, called CMEK source domain selection, an acronym of the four measures we chose in our case study, to deal with the previously stated challenge. The model measures statistical distances between the marginal distributions of candidate source domains and the target domain and uses the inner domain classification error of the candidate source domains. The candidate source domain with the lowest distance is hypothesized to have the lowest cross domain classification error and is selected to train a classifier. This section describes the methodology, the distance measures we consider and how we evaluate the performance of the model.

We consider a set $\mathcal{D}$ containing a family of distance functions $d : \mathbb{P} \times \bar{\mathbb{P}}_x \to \mathbb{R}_+$. We now hypothesize that given an available set of source domain distributions $\mathcal{P}$ and a target domain distribution denoted by $\bar{\mathbb{P}}$ with the marginal $\bar{\mathbb{P}}_x$, there exists a $\hat{d} \in \mathcal{D}$ that can reliably predict the cross domain classification error $\xi(\mathbb{P},\bar{\mathbb{P}})$, that is,

$$\xi(\mathbb{P},\bar{\mathbb{P}}) \overset{\text{hyp}}{\approx} \hat{d}(\mathbb{P},\bar{\mathbb{P}}_x). \tag{4}$$

More specific, we hypothesize that the cross domain classification error can be predicted by looking at the difference in marginal distribution functions $\mathbb{P}_x$ and $\bar{\mathbb{P}}_x$, and $\xi(\mathbb{P})$. With this prediction, the best candidate source domain can be selected for training. We formalize this optimal choice of the measure by considering the optimization problem

$$\hat{d} := \arg\min_{d \in \mathcal{D}} \sum_{\mathbb{P} \in \bar{\mathcal{P}}} \big| \xi(\mathbb{P}, \bar{\mathbb{P}}) - d(\mathbb{P}, \mathbb{P}_x) \big|. \tag{5}$$

To construct the family $\mathcal{D}$ of candidate measures, we use a vector s with K known statistical distance measures as elements, $s_i : \mathbb{P}_x \times \bar{\mathbb{P}}_x \to \mathbb{R}_+$ for $i \in \{1, \cdots, K\}$. The family $\mathcal{D}$ consists of linear combinations of these measures s_i with corresponding weight coefficients $\beta_i \in \mathbb{R}_+$, with β_i being elements of the weight vector β. We augment the linear combination with a constant. Note that the distance measures are functions supported on the marginal distributions of source and target domain. Examples of these statistical distance metrics are Integral Probability Metrics and ϕ-divergence. When a source domain has poor inner domain classification performance, we expect it to be less suitable as source domain for cross domain classification. Therefore, we calculate the inner domain classification error for each candidate source domain and add this to our vector s. With our choice for $\mathcal{D}$, the optimization problem (5) can be reduced to

$$\begin{aligned} \hat{\beta} &:= \arg\min_{\beta \geq 0} \sum_{\mathbb{P} \in \mathcal{P}} \big| \xi(\mathbb{P}, \bar{\mathbb{P}}) - \beta s(\mathbb{P}, \bar{\mathbb{P}}_x) \big|, \\ \hat{d} &:= \hat{\beta} s(\mathbb{P}, \bar{\mathbb{P}}_x). \end{aligned} \tag{6}$$

Note that program (6) constructs the optimal unsupervised predictor by using the full distribution of the target data, which is presumed to be unknown. In a practical setting with a finite number of elements in $\mathcal{P}$, we deal with this problem by training only on the domains in $\mathcal{P}$. In each run, one of the elements of $\mathcal{P}$ is extracted from the set and used as proxy $\bar{\mathbb{P}}$, at the end of the run, this element, denoted as $\tilde{\mathbb{P}}$ and its marginal $\tilde{\mathbb{P}}_x$, is placed back in $\mathcal{P}$. In every run, another element is extracted and placed back for training. This training program is described as

$$\hat{\beta} := \arg\min_{\beta \geq 0} \sum_{\mathbb{P} \in \mathcal{P}, \mathbb{P} \neq \tilde{\mathbb{P}}} \sum_{\tilde{\mathbb{P}} \in \mathcal{P}} \big| \xi(\mathbb{P}, \tilde{\mathbb{P}}) - \beta s(\mathbb{P}, \tilde{\mathbb{P}}_x) \big|.$$

We hypothesize that with a sufficient amount of source domain distributions in $\mathcal{P}$, the constructed predictor for the cross domain classification error, based on information from $\mathcal{P}$, is also reasonably accurate in predicting the cross domain classification error for a target domain not included in $\mathcal{P}$. This allows us to select the best source domain to train on for this unseen target domain.

Now, in view of our hypothesis (4), with the predictor $\hat{d}$ we are able to approximate $\xi(\mathbb{P}, \bar{\mathbb{P}})$ with only marginal information of the target domain $\bar{\mathbb{P}}_x$, which allows us to deal with the challenge through an approximation. This enables us to, given a target domain with distribution $\bar{\mathbb{P}}$, select the source domain, characterized by $\hat{\mathbb{P}}$ from a set of available source domains that is predicted to minimize the cross domain classification error $\xi(\mathbb{P}, \bar{\mathbb{P}})$ as formalized through the optimization program

$$\hat{\mathbb{P}} := \arg\min_{\mathbb{P} \in \mathcal{P}} \hat{d}(\mathbb{P}, \bar{\mathbb{P}}_x)$$

which is the goal of this study.

3.1. Measures

In the SC case study, we hypothesize that the way people express sentiment can be captured in how often certain words and word combinations are used. The underlying joint distributions $\mathbb{P}$ differ among different domains. To accurately classify sentiment for a target domain, we therefore ought to find a domain with a similar underlying distribution function. To this end, one may need to measure the similarity between different distributions. In this article we consider two classes of distance measures: Integral Probability Metrics (IPM's), and ϕ-divergences that map $\mathbb{X} \times \mathbb{X} \to \mathbb{R}$. Both measures compare the distributions of the features but in different fashions.

3.1.A. *Integral Probability Metrics*

The main characteristic component of IPM distance functions is the set $\mathcal{F}$ that contains a family of functions $f : \mathbb{X} \to \mathbb{R}$. Using $\mathcal{F}$, we define the distance between two distributions $\mathbb{P}_x$ and $\bar{\mathbb{P}}_x$ as

$$s_i(\mathbb{P}_x, \bar{\mathbb{P}}_x) := \sup_{f \in \mathcal{F}} \Big| \mathbb{E}^{\mathbb{P}_x}\big[f(X)\big] - \mathbb{E}^{\bar{\mathbb{P}}_x}\big[f(X)\big] \Big|.$$

Different choices of the family $\mathcal{F}$ give rise to different distance functions. A popular choice is to limit our search space to the unit ball in a predefined metric space. In this article, we consider two choices: (i) the unit ball in a Reproducing Kernel Hilbert Space which corresponds to the Maximum Mean Discrepancy (MMD) distance function [16], (ii) the unit ball of Lipschitz functions which corresponds to the Earth Mover's Distance (EMD) distance function [2].

More intuitively, the EMD metric measures the minimum transportation cost to transform one distribution to another. The transportation cost, roughly speaking, amounts to the effort measured in a so-called ground metric that is required to move a unit probability mass. The MMD metric finds a well behaved, smooth, function which is typically high on the points drawn from $\mathbb{P}_x$ and low on $\bar{\mathbb{P}}_x$ or vice verse.

3.1.B. *Phi-divergence*

For the second class of distance measures, we consider the ϕ-divergence between two distributions $\mathbb{P}$ and $\bar{\mathbb{P}}$ defined as

$$s_i(\mathbb{P}_x, \bar{\mathbb{P}}_x) := \begin{cases} \int\limits_{\mathbb{X}} \phi\left(\frac{d\mathbb{P}_x}{d\bar{\mathbb{P}}_x}\right) d\bar{\mathbb{P}_x} & \mathbb{P}_x \ll \bar{\mathbb{P}}_x, \\ +\infty, & \text{otherwise} \end{cases}$$

where ϕ is a convex function, and $\mathbb{P}_x \ll \bar{\mathbb{P}}_x$ denotes that $\mathbb{P}_x$ is absolutely continuous with respect to $\bar{\mathbb{P}}_x$ [37]. A widely used choice is $\phi(t) = t\log(t)$, resulting in the Kullback-Leibler Divergence (KLD). Another popular choice for $\phi(t)$ is $\phi(t) = (t-1)^2$ which is based on Pearson's χ^2 test statistic. This choice gives the χ^2-divergence (Chi2) distance measure. An interesting property of these two divergences is that they are asymmetric, $s_i(\mathbb{P}_x, \bar{\mathbb{P}}_x) \neq s_i(\bar{\mathbb{P}}_x, \mathbb{P}_x)$.

Since we hypothesize that the way people express sentiment can be captured in how often certain words and word combinations are used, regardless of the context, we are then left with a broad spectrum of possibilities to optimize over. We propose to use a linear combination of the two IPM approaches MMD and EMD, the two ϕ-divergences KLD and Chi2, a constant representing an offset error and the inner domain classification error of the source domain.

In most practical cases, we only have access to the true probability distribution through a finite number of observations. For that reason, in order to evaluate these discrete distributions, the integrals are replaced by summation. Selection based on the linear combination of these four metrics, the constant, and $\xi(\mathbb{P})$ is referred to as the CMEK selection model.

$$\begin{aligned} d(\mathbb{P}, \bar{\mathbb{P}}_x) = & \beta_1 Chi2(\mathbb{P}_x, \bar{\mathbb{P}}_x) + \beta_2 MMD(\mathbb{P}_x, \bar{\mathbb{P}}_x) \\ & + \beta_3 EMD(\mathbb{P}_x, \bar{\mathbb{P}}_x) + \beta_4 KLD(\mathbb{P}_x, \bar{\mathbb{P}}_x) \\ & + \beta_5 \xi(\mathbb{P}) + \beta_0 \end{aligned}$$

3.2. Performance evaluation

We have constructed a model that predicts cross domain classification error between a candidate source domain and a target domain with a linear combination of statistical distance measures based on the marginal distributions $\mathbb{P}_x$ and $\bar{\mathbb{P}}_x$, a constant and $\xi(\mathbb{P})$. Now we are interested in how well this model is able to identify the best candidate source domain, i.e. the one that gives the lowest cross domain classification error $\xi(\mathbb{P}, \bar{\mathbb{P}})$. For n_c acquired datasets, let us use one corpus as target domain, $\bar{\mathbb{P}}$, and the others as candidate source domains with the set annotation $\mathcal{P}'$. We can repeat the experiment n_c times by using each time a different target domain and average the results.

We let the model select the predicted best source domain $\hat{\mathbb{P}} \in \mathcal{P}'$ and use this domain for training. To evaluate performance we define the relative cross domain classification error as

$$\xi_{relative}(\hat{\mathbb{P}}, \bar{\mathbb{P}}) = \xi(\hat{\mathbb{P}}, \bar{\mathbb{P}}) - \xi(\mathbb{P}^{\star}, \bar{\mathbb{P}}) \tag{7}$$

where $\mathbb{P}^{\star}$ is the domain in $\mathcal{P}'$ of which we know it has the lowest cross domain classification error, the true best domain. When the model selects the best source domain, the relative error will therefore be zero. We compare the distribution of this relative error when using the CMEK model with randomly selecting one source domain for training. Also, we look at the absolute cross domain classification error, the probability of selecting the best domain and the probability of selecting one of the five worst domains when using the CMEK model. We compare this with randomly selecting a domain and with an optimal selection model.

Next, we let our proposed CMEK model select the best n domains from $\mathcal{P}'$ and compare $\xi(\bigcup_{i=1}^{n} \hat{\mathbb{P}}_i, \bar{\mathbb{P}})$ with training on all domains in $\mathcal{P}'$ and with randomly selecting n domains. All results are tested on significance using a paired t-test over the results of 13 runs, we use a significance cutoff of $p = 0.05$. The paired t-test is strongly dependent on the assumption that the pairwise differences are normally distributed. For every comparison between results, we evaluated the normality assumption. We report only on significant results for which the normality assumption is confirmed.

To test if our methodology creates added value when using it in an adaptive learning setting, we let our model select the predicted best source domain and evaluate the cross domain classification error when using a DANN adaptive learning setting described in the recent study [14]. As a benchmark, the result is compared with random selecting a source domain in the same training and test setting.

4. Limitations

In solving the problem as defined in section 2 with our proposed method, we encounter limitations of various kinds. We first discuss a fundamental statistical limitation that is present in almost all practical machine learning problems. Then we discuss a limitation of computational nature and how to deal with this. In the last paragraph we elaborate on limitations that are more specific to the case study of SC: numerically representing text and the limited hypothesis space. We end with briefly explaining the limitations of our proposed CMEK model.

4.1. Statistical limitations

In the first place, and inherent to all classification problems, we are unable to retrieve the true distribution over the support set $\mathbb{X} \times \mathbb{Y}$ in practical settings. Instead, we need to infer this distribution through only a finite number of observations. In this case a natural approximation for the distribution is the empirical distribution supported on these observations, which are the available documents in our SC case study. We then encounter an inevitable approximation

$$\mathbb{P} \approx \frac{1}{n_s} \sum_{i}^{n_s} \delta_{\{x_i, y_i\}}$$

where n_s denotes the number of documents in the domain sample. Due to the approximation of the true distribution, the hypothesis function (1) is suboptimal, resulting in a higher classification error. To evaluate this error, the available data originated from one domain is split into a train and test set. The train set is used to construct $h^\star_{\mathbb{P}}$ as defined in (1). The test set is used to calculate the inner domain classification error. When using sufficient data in the test set, a large difference in apparent and true error indicates that the observations used for training are a poor support for approximating the true distribution since it means that the empirical distributions from the train and test set are not much alike. When this is the case, we are over fitting; the hypothesis function works for the data it is constructed on, but can not be properly generalized to perform well on new data from the same underlying distribution. To reduce over fitting, we ought to use a sufficient number of training objects and a low but not too low number of features.

4.2. **Computational limitations**

Another limitation is that the optimization in (1) uses the indicator function as loss function, which is non-convex. For computational reasons, we replace this loss function with a convex counterpart. For the true hypothesis function $h^\star_{\mathbb{P}}$ in (1), a common convex loss function $\ell(x,y)$ to minimize, is the quadratic error of the prediction, popular as the method of least squares

$$h^\star_{\mathbb{P}} = \arg\min_h \sum_{i=1}^{n_s} \ell(x_i, y_i) \qquad \ell(x,y) := \big(h(x) - y\big)^2, \tag{8}$$

and the cross domain classification error $\xi(\mathbb{P}, \bar{\mathbb{P}})$ is then efined by empirical distributions of the target and the hypothesis function from (8),

$$\xi(\mathbb{P}, \bar{\mathbb{P}}) := \frac{1}{n_s} \sum_{i}^{n_s} \mathbb{1}_{\left\{h^\star_{\mathbb{P}}(x_i) \neq y_i \ \mid \ (x,y) \sim \bar{\mathbb{P}}\right\}}. \tag{9}$$

4.3. **Further discussion on SC related limitations**

There are limitations concerned with performing SC with a nature that may also be relevant to other fields of application. Choosing a word representation model is a field of study by itself and should be considered carefully here. In the previous section we assumed a true hypothesis function $h^\star$ mapping each document represented in $\mathbb{X}$ to a label in $\mathbb{Y}$. For computational purpose, we need to numerically express documents in the $\mathbb{X}$ space. Let us call W the set of all words, $W = \{w_1, w_2, \dots w_{n_w}\}$ where n_w is the number of unique words. Then, documents in $\mathbb{X}$ are combinations of these words, indicating that $\mathbb{X}$ can be viewed as an element in the power set of W denoted by $\wp(W)$. Note that repetition in the subset is possible. For computational feasibility, we choose a subset $\mathcal{F} \subset \wp(W)$ to represent our documents. The subset $\mathcal{F}$ is called the feature set whose cardinality is denoted by $|\mathcal{F}|$. The features selection process depends on what representation model is at hand. The representation model $\mathfrak{F}$ maps $\mathbb{X}$ to $\mathbb{R}^{|\mathcal{F}|}$, i.e. $\mathfrak{F} : \mathbb{X} \to \mathbb{R}^{|\mathcal{F}|}$. The computational need for a representation model $\mathfrak{F}$ to select a subset of $\mathbb{X}$ for representation, limits the classification model since information has to be discarded, see Figure 1.

Furthermore, in the task of classification we need to define a search space $\mathcal{H}$ for our hypothesis function as it is computational impossible to search through all possible hypothesis functions as proposed in (1). SC tasks are characterized by the fact that the representation space is sparse, i.e. the ratio of number of training documents to $|\mathcal{F}|$ is low. In this situation, computational feasibility calls for a tractable subset of hypothesis functions $\mathcal{H}$ such as all linear functions. The true hypothesis function is not likely to be in our hypothesis space, $h^\star \notin \mathcal{H}$. We define the optimal hypothesis function constructed in the domain $\mathbb{P}$ as the hypothesis $\hat{h}$ within our limited hypothesis space $\mathcal{H}$, which is the collection of all linear mappings from $\mathbb{R}^{|\mathcal{F}|}$ to $\mathbb{R}$, that

$$\begin{tikzcd}
\mathbb{X} \arrow[r, "h^\star"] \arrow[d, "\mathfrak{F}"'] & \mathbb{Y} \\
\mathbb{R}^{|\mathcal{F}|} \arrow[ur, "\hat{h} \in \mathcal{H}"'] &
\end{tikzcd}$$

Figure 1. Representation and classification structure

minimizes the chosen loss function (8). The cross domain classification error from (9) is calculated with the optimal hypothesis function $\hat{h}_{\mathbb{P}}$.

Hereafter, we use $\mathbb{P}$ to annotate the empirical distribution instead of the true continuous distribution. Furthermore, the cross domain classification error defined as $\xi(\mathbb{P}, \bar{\mathbb{P}})$ in (2) is calculated with the optimal hypothesis function in $\mathcal{H}$ that minimizes the empirical loss in (8).

4.4. **CMEK selection model limitations**

Our model assumes that a good source domain, with low cross domain classification error, can be identified by the distributions $\mathbb{P}$ and $\bar{\mathbb{P}}_x$. However, where in reality sentiment is determined by joint distribution of the features, the measure we use only measure distance on the individual distributions of the features. This means we try to find similarity of the joint distributions from marginal distributions of individual features. The same holds for classification that is based on marginal data of individual features whereas sentiment in real life is determined by combinations of features. Since we use marginal data for calculating the cross domain classification error as well as the prediction of this error, this approach seems justified, but it creates a model separated from the true rules of expressing sentiment.

In addition, we do not know in what way we should compare distributions, only that when distributions are the same, we expect no adaptation loss. It might very well be that the linear combination of our chosen measures does not include the true distance function that distinguishes domains from each other.

5. Case Study: Sentiment Classification

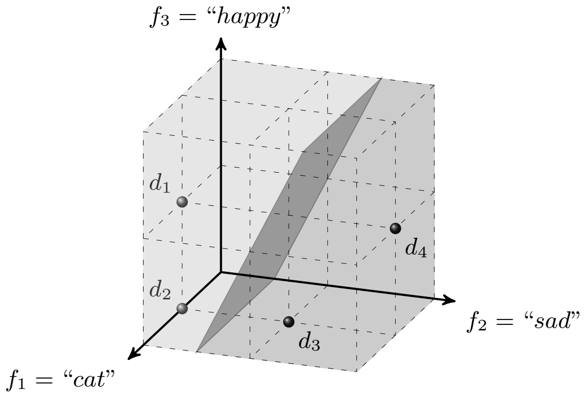
Let us give some brief background information on how to perform SC, mainly to clarify design choices made in the experimental set up. The pipeline for SC can be segmented into three parts: data preprocessing, document representation and classification. The next paragraphs describe the pipeline. For a lower dimensional visualization, see Figure 2.

5.1. **Preprocessing**

In the preprocessing part one tries to remove noise from the text that does not hold sentiment information in order to reduce complexity, i.e. dimensionality. Common techniques to do so are stop word removal, lower casing words, spelling correction and removing punctuation. Other techniques, including part of speech tagging, stemming and lemmatization, attempt to find implicit sentiment information by predicting syntactics of words or combinations of words.

5.2. **Document representation**

The second part of the pipeline for sentiment classification deals with representing documents in a mathematical interpretable fashion. The most upfront approach is to use words as features, and word counts as feature values without looking at word order, a bag-of-words approach. With n_F words or combinations of words as features, $n_F = |\mathcal{F}|$, we can build an n_F-dimensional feature space in which each word is represented



d_1 = "The **cat** makes me **happy**."- POS
d_2 = "The **cat** is beautiful."- POS
d_3 = "The **cat** makes me **sad**."- NEG
d_4 = "I was **happy**, the **sad** **cat** made me **sad**."- NEG

Figure 2. Classified feature space

by an n_F-dimensional vector. The number of occurrences of a feature in the document determines the value in that dimension. Each document is then represented as the sum of all the vectors of its features. When using n_F features, a corpus of n_d documents can be represented as a matrix $X \in \mathbb{N}^{n_d \times n_F}$ which is referred to as the feature value matrix. X represents a projection of the empirical distribution $\mathbb{P}$. In addition, one could choose to use combinations of N words as features, called N-grams. Often, features are weighted to assign less value to more common words such as stop words. A widely used weighting scheme borrowed from information retrieval systems is the TF-IDF weighting scheme. To reduce dimensionality, feature selection algorithms can be used, selecting the most class label informative features by using for example χ^2, mutual information or lexicon based selection [19]. Another popular approach for document representation uses word embeddings [24] [28], a lower dimensional projection of a high dimensional feature space. To construct a good projection, we need a lot of data, preferably from the domain that is under evaluation.

5.3. Classification

When we have a mathematical representation for the source and target documents, we try to find a hypothesis function that separates the feature space in subspaces belonging to the different class labels. All documents that are represented in certain subspace, are predicted to have the accompanying label of that specific subspace. The hypothesis function, or classifier, is constructed by minimizing a loss function, see (8). Popular classifiers for sentiment classification are Support Vector Machines, Naive Bayes and Logistic Regression (LR).

6. Experimental Setup

In this section we briefly elaborate on different datasets we used for our experiments as well as the design choices we consider concerning the SC pipeline and the proposed distance measures.

6.1. Datasets

To evaluate the methodology, we choose two sets of data that are different in their nature of topic variety, source medium, number of documents, document length, and class balance. Our primary goal to conduct two experiments with datasets at different homogeneity levels is to investigate the performance and robustness of our proposed method with respect to the diversity of the training datasets.

6.1.A. *Homogeneous dataset*

The homogeneous dataset consisting of 20 corpora from the DRANZIERA benchmark dataset [12]. We use 5000 positive and 5000 negative documents from each corpus. The documents range from 64 to 123 words with an average of 90. Each corpus consists of Amazon reviews on products from a different category. This set is referred to as the "homogeneous" set as the nature of all corpora are similar, but only the subtopics of the corpora differ.

6.1.B. *Heterogeneous dataset*

Our second dataset contains 13 corpora. Each corpus consists of a different number of documents ranging from 502 to 30602 documents with an average of 6326 documents. The average length of each document among the corpora varies between 10 words and 20 words with an average of 15 words. We deploy a corpus of online reviews on movies, venues and consumer products [21], short opinions about movies [41], and nine sets of tweets with various topics [11]. The acquired datasets do not have any major class imbalances. This set is referred to as the "heterogeneous" set as the nature of each corpus differs in number of documents, source and topic. In a direct comparison with the first datasets, we note that the homogeneous set is richer and more balanced in terms of data points, and the topics of the corpora lay closer to each other in semantic meaning.

6.1.C. *Adaptive learning dataset*

The third set acquired is a well known dataset containing Amazon reviews on four different categories of products [5]. We use 2000 reviews from each category. Motivated by the original paper [14] on DANN, This dataset is used in order to evaluate the performance of our proposed method in an adaptive learning setting.

6.2. **Design choices**

Considering the SC pipeline design choices, in our experiments we only remove punctuation and lower case all words as prepossessing. For classification in our experiments, we use a bag-of-words approach with unigrams and bigrams and use all features that occur more than 4 times in the training corpus and at maximum in 40% of the training documents. We weight the features according to the TF-IDF weighting scheme. We deliberately do not use word embeddings since the lower dimensional projection needs too much textual domain data to construct. Using data from other domains will distort results as the representation becomes dependent on other domains than the source and target domain that are under performance evaluation. We use LR for its good performance. We use binary classification, $y \in \{0, 1\}$, giving the following loss function for our classifier

$$\ell(x, y)_{LR} = \sum_{i=1}^{n_d} \log\left(1 + e^{-(2y_i - 1)x_i\beta + \beta_0}\right) + \alpha\frac{1}{2}\|\beta\|^2$$

where w and w_0 represent the optimizers, α represents the regularization parameter and the hypothesis function is defined as

$$h(x) = \begin{cases} 1 \text{ if } \beta x + \beta_0 > 0 \\ 0 \text{ else.} \end{cases}$$

For α, we use the default classifier settings of our used toolbox [31].

In the adaptive learning setting, we use exactly the same model settings as in [14]. The model deploys DANN for classification both with an original representation of 5000 features as with a marginalized Stacked Denoising Auto-encoders (mSDE) representation [8], an advanced representation that improves performance in SC. For both representations, we evaluate the performance of our methodology.

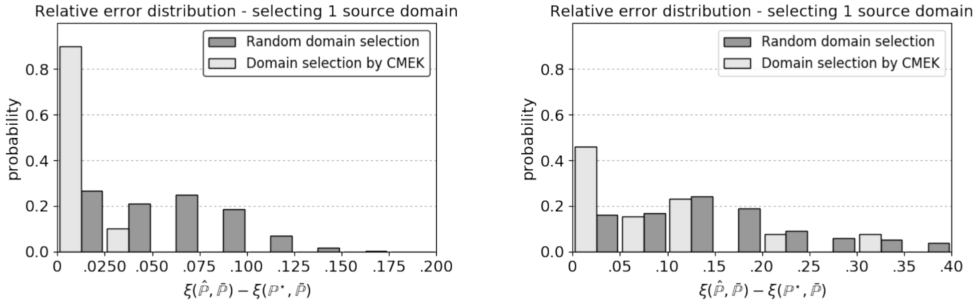
For all distance measures, we only measure the distance over the $n = 1000$ most occurring features for computational convenience. We do not compare the distribution of all features for two reasons. In the first place, the distance calculations will take long when using all features. And second, and more important, distributions of rarely occurring features are less reliable. For the Chi2 divergence, we smoothen the distributions by adding a constant $\lambda = 0.05$ over the entire distribution $p(X)$ to prevent the distance between two discrete distributions to become infinite when we encounter a probability of zero. For the KLD we add a constant $\lambda = 10^{-5}$ to the entire distribution for same reasons. For the MMD we choose a Reproducing Kernel Hilbert Space that corresponds to the Gaussian Radial Basis Function kernel $K(x, y) = e^{\frac{||x-y||^2}{2\sigma^2}}$ where σ is our optimizer. We use corpora of the same number of objects, i.e. documents, as input by using a subset of the largest corpus of the two corpora used. For computational convenience we use a maximum of 5000 documents. For the EMD we use the set of all continuous Lipschitz functions with respect to the 1-norm. Note that the distribution is highly dependent on the length of a document. Therefore, we match document lengths among the two domains before calculating the EMD. For the optimization for the two IPMs we use available code for discrete distributions [42] [27] [32] [33].

For the inner domain classification error that is used as predictor in the CMEK model, we use the same logistic regression as for the cross domain classification and use 10-fold cross validation to assess the inner domain classification error.

7. Results

In this section we will objectively present the results of our experiments according to our performance evaluation described in subsection 3.2.

Figure 3 shows the empirical distribution of the relative error as defined in (7) when selecting one source domain. The relative error distributions when randomly selecting a source domain and when using the CMEK source domain selection model are shown.



(A) Homogeneous dataset

(B) Heterogeneous dataset

FIGURE 3. Relative error of CMEK and random source domain selection when selecting a single training domain

For the homogeneous and heterogeneous dataset, the CMEK model uses the optimized weights

$$\begin{aligned} \hat{\beta}_{homogeneous} &= [0.13,\ 0.52,\ 1.02,\ 0.00,\ 0.00,\ 0.11] \\ \hat{\beta}_{heterogeneous} &= [1.13,\ 0.13,\ 0.07,\ 0.00,\ 0.45,\ 0.27] \end{aligned} \tag{10}$$

for respectively, Chi2, MMD, EMD, KLD, the source domain inner domain classification error, and the constant 1.

Table 1 shows the probability of selecting the true best domain, the probability of selecting one of the five worst domains in terms of the cross domain classification error, and the average absolute cross domain

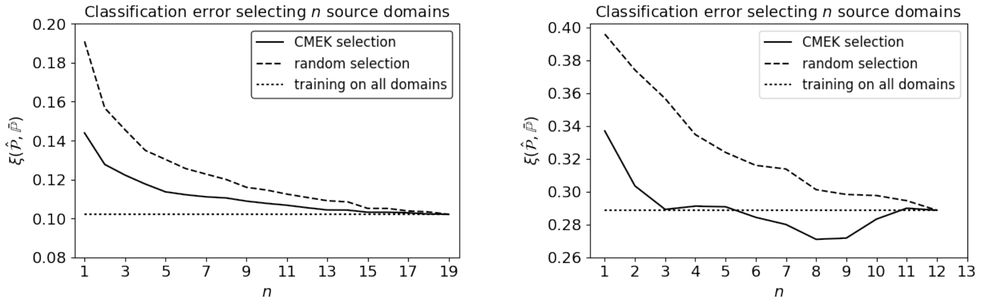
Table 1. Performance of CMEK, random and optimal selection.

Selection method	**Probability Best possible domain selected**	**Probability One of 5 worst domains selected**	**Average absolute $\xi(\hat{\mathbb{P}},\bar{\mathbb{P}})$**
Optimal	1	0	.139
CMEK	.600	0	.144
Random	.053	.263	.191

(a) Homogeneous dataset

Selection method	**Probability Best possible domain selected**	**Probability One of 5 worst domains selected**	**Average absolute $\xi(\hat{\mathbb{P}},\bar{\mathbb{P}})$**
Optimal	1	0	.252
CMEK	.385	.154	.337
Random	.083	.417	.403

(b) Heterogeneous dataset



(a) Homogeneous dataset

(b) Heterogeneous dataset

Figure 4. Performance of the CMEK model and random selection for different number of selected source domains in comparison with training on all source domains

classification error, averaged over the all target domains. We list the results for using the CMEK model for domain selection, when the source domain is selected at random and an optimal model that would always select the best source domain. The optimal cross domain classification error can be seen as lower bound of error. For the homogeneous dataset, the CMEK model uses the optimized weights

Figure 4 shows the absolute cross domain classification error when we let our CMEK selection model select n source domains. Note that the CMEK model is constructed by optimizing the predicted cross domain classification error when training the classifier on single source domains, not training on multiple. The results are compared with randomly selecting n domains and training on all candidate source domains.

Table 2 reports the probability of selecting the best domain in terms of the cross domain classification error, and the respective average absolute cross domain classification error; the result is empirically computed based on the four available target domains provided by [5]. We list the results corresponding to the CMEK model as a means to select the source domain, when the source domain is selected at random and an optimal model that would always select the best source domain.

Table 2. Performance of CMEK, random, and optimal selection in adaptive learning setting [14].

Selection method	Probability Best possible domain selected	Average absolute $\xi(\hat{\mathbb{P}}, \bar{\mathbb{P}})$
Optimal	1	.199
CMEK	1	.199
Random	.33	.237

(a) Original representation

Selection method	Probability Best possible domain selected	Average absolute $\xi(\hat{\mathbb{P}}, \bar{\mathbb{P}})$
Optimal	1	.152
CMEK	1	.152
Random	.33	.187

(b) mSDA representation

8. Conclusion

To select a suitable source domain in terms of cross domain classification error, we hypothesized that this error can be predicted by a function of the source domain distribution $\mathbb{P}$ and marginal target domain distribution $\bar{\mathbb{P}}_x$. We proposed the CMEK model to select the best source domain(s) from a set of candidates. For our case study, the CMEK model uses a linear combination of the Chi2, MMD, EMD, KLD, the inner domain classification error and the constant 1, with weight vector β as predictor for the cross domain classification error. To evaluate performance, we consider an example including $N-1$ distinct domains forming $N(N-1)$ different source-target domain pairs. The proposed metric parameter β is optimized to minimize the absolute error of the prediction. The optimized predictor was used to select one or multiple best source domains among those $N-1$ domains in order to train a classifier for the Nth, unseen target domain. This classifier is tested by calculating the cross domain classification error of the selected source domain(s) and the target domain. The process is repeated N times, each time using a different unseen target domain to test on. We benchmark the performance of our solution with randomly selecting a source domain as well as training on all candidate source domains. We perform this method over a homogeneous dataset ($N = 20$) and for a heterogeneous dataset ($N = 13$).

From figure 3 we see that the CMEK model is well able to identify source domains with low cross domain classification error. In 90% of the runs on the homogeneous datasets, the selected source domain was within 2.5 percent points error of the optimal choice whereas random selection only selected 26% of the times a source domain in this category. For the heterogeneous set we see a similar increase in the performance of the CMEK model when looking at the probability a domain is selected within 5 percent points error of the optimal choice, 45%, and 16%, respectively.

From Table 1, we see that compared to random domain selection, the CMEK realizes a significant improvement in average cross domain classification error for both the homogeneous set and the heterogeneous set ($p_{hom} = 3.6 \times 10^{-9}, p_{het} = 0.0071$). The CMEK model has a significant larger probability of selecting the best domain compared to random selection ($p_{hom} = 3.8 \times 10^{-11}, p_{het} = 0.0029$). If we look at the probability of not selecting one of the five worst domains for training, the CMEK model performs significantly better than the random selection model ($p_{hom} = 2.1 \times 10^{-9}, p_{het} = 0.045$).

From (10) we see that the CMEK model gives different weights to the distance measures for the two used datasets. This suggests that some of the considered measures are relatively better in predicting the cross domain classification error in homogeneous datasets, and the others are better when using heterogeneous datasets. The CMEK model optimizes the weights accordingly.

Although the proposed CMEK selection model significantly improves performance compared to random selecting one source domain, we would still be better off training on all the available source domains. However, if we let our model select multiple source domains to train on, we are able to get better performance than training on all source domains for some n, see Figure 4b. For the heterogeneous dataset, the CMEK model seems to perform significantly better than training on all domains when it selects 9 or 10 domains ($p_{het} = 0.043$ and $p_{het} = 0.043$). Note, for all mentioned significant results, the normality assumption of the t-test is confirmed, however in this specific evaluation, the pairwise differences seem not normally distributed for $n = 10$. What the optimal number of domains to train on is, may be very dependent on what candidate source domains are available. When we have candidate source domains that are somehow similar to the target domain, the optimal n will be higher, or even equal to the total number of sets, as in the heterogeneous dataset, see Figure 4a. For a diverse set of candidate source domains the optimal n will be lower.

Looking at the results when combining the CMEK model with adaptation learning, Table 2, we observe that the CMEK model enabled us to select the best source domain out of three candidates for each of the four target domains. This gives us a significant improvement with regard to a random selection model ($p = 0.012$).

In the light of general performance, we would recommend the CMEK model to select a source domain for an SC task for an unlabeled target domain. The CMEK model shows significantly good performance and stable behavior in selecting multiple source domains and it has solid performance in selecting the single best domain for both our homogeneous and heterogeneous datasets. We notice that the CMEK model also adds value in the evaluated adaptive learning setting.

In retrospect to our hypothesis (4), we can conclude that it is to some extend possible to approximate the cross domain classification error as we were able to use this approximation for successful source domain selection. However, there is much room for improvement.

8.1. Further discussion

Reflecting on our conclusion, we would like to place some remarks. We showed that the CMEK model works superior in selecting one source domain compared to random selection. However, the results show that, even for candidate source domains with a wide spread in topic and source medium, it is quite beneficial to train on all the candidate source domains. One of the reasons not to train on all data could be that it is too computational expensive. Another reason might be that the candidate source domains are too diverse. To establish if this is the case, we would need a measure that informs us about the diversity of the candidate source domains.

When we know we have candidate source domains that are similar to each other in terms of expressing sentiment, it might very well be that our CMEK model is not able to improve performance compared to training on all data. We illustrated this by using the homogeneous dataset, see Figure 4a. Therefore, if we have candidate source domains that are somewhat similar to each other, we may choose to train on all domains.

Another disadvantage of the CMEK model, is that it uses some distance measures that are quite expensive to calculate. This can be a problem when we have many or large candidate source domains. This challenge might be addressed by using less features and more informative features to calculate distance over and we could optimize used code on efficiency. We might investigate what happens to the performance of the CMEK model in case we leave out the EMD and KLD measure, as they are given fewest importance, i.e. lowest weight β_i.

We constructed, tested, and evaluated the model on three different acquired datasets. We would like to remind that sentiment expression is characterized by a severe notion of stochasticity. It is hard to tell how much performance of the CMEK model will deviate when other sets are chosen to construct the model. For more available domains to construct the model, we may assume that the model will be more refined and better performing due to the wider support for the β weight vector.

The dataset used for the adaptive learning setting is quite popular in the field. However, it only contains data from four domains. When using one of the domains as the target, only three domains remain in the candidate source domain set. The chance of coincidentally selecting the optimal domain is not negligible. Although we presented that the results are statistically significant, for future work, we would prefer to use a dataset that contains data originated from more domains.

Although we described a method that can be easily implemented in other fields of machine learning that encounter a domain shift such as computer vision, fraud detection or spam detection, we only showed our model works reasonably well in the domain of SC.

8.2. **Future Work**

As the CMEK model is to some extension able to select a source domain with lower than average cross domain classification error, we concluded that the used measures are to some extend able to predict the cross domain classification error in order to select a suitable source domain. However, if our model would fully do justice to our hypothesis (4), we would improve even further. We could say that proof of concept is given: it is possible to roughly predict the cross domain classification error based on the distribution of the source domain distribution and marginal distribution of the target domain. However, we still see much room for improvement. As the lower bound for the error when selecting one domain in the heterogeneous dataset is .252, we are only half way. For the homogeneous dataset, it turns out that we are closer to achieve an optimal model.

If we would like to improve results further, first steps would be to look closer into the features over which we measure distance. Should we use the 1000 most common, or is distance better measured with more or less features? This could be dependent on corpus size as well. There might be feature selection methods to select the features that are most informative in terms of distance, instead of simply using the N most occurring features. We, could for example, look at domain specific emotion lexicons to measure distance [3].

To improve, we might add some predictors to the linear combination, such as the document length distribution or the length of a corpus. Brief statements are on average more similar with each other than brief statements compared with long expressions. Also, if we have more objects in a source domain, we might benefit more when training on that large source domain than training on a corpus with only a few documents. However, with more elements in β, we might need more data to prevent over fitting.

We can also extend our model to give a weight factor on every source domain: source domains that are predicted not suitable get a low weight, domains with low distance to the target are given more weight. This approach leans towards importance sampling.

We constructed a predictor for the cross domain classification error of one source domain, and used the model to select multiple source domains. If this is what we are interested in, it would make sense to calculate distances between a set of multiple domains and the target domain and make a selection model optimized on predicting the cross domain classification error when training on multiple domains. In that case, synergies of datasets will be taken into account. This approach will, however, be very computational expensive.

Concerning the results when deploying the CMEK model in an adaptive learning setting, we evaluated the performance when selecting a source domain based on the original source data distribution. It would be interesting to see what we could achieve when we let the CMEK model select based on adapted source data distribution. In that scenario, the source data is first transformed with adaptive techniques, and then the respective distance is measured between the transformed source data and the target data in order to select the predicted best domain in an adaptive learning setting.

At last, we notice that when the available datasets are rather heterogeneous, the CMEK model is typically able to outperform the classifiers trained on all datasets by selecting an appropriate subset of the training domains. To realize this, it is essential to determine what the optimal number of domains to train on is. We could approach this question by defining an absolute threshold in distance between the source and target domain, or by clustering the candidate source domains and selecting a cluster of candidates to train on. Combining the proposed methodology with a model that selects the optimal number of domains to train on might bring a very useful solution to many machine learning problems.

Acknowledgment

The authors are grateful to Dr. Mauro Dragoni for providing us the DRANZIERA dataset, and to the company Crunchr for support throughout the project. The third author gratefully acknowledges funding from the Swiss National Science Foundation under grant P2EZP2-165264.

References

[1] M. J. Afridi, A. Ross, and E. M. Shapiro, *On automated source selection for transfer learning in convolutional neural networks*, Pattern Recognition, 73 (2018), pp. 65–75.

[2] M. Arjovsky, S. Chintala, and L. Bottou, *Wasserstein gan*, arXiv preprint arXiv:1701.07875, (2017).

[3] A. Bandhakavi, N. Wiratunga, D. Padmanabhan, and S. Massie, *Lexicon based feature extraction for emotion text classification*, Pattern recognition letters, 93 (2017), pp. 133–142.

[4] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, *Analysis of representations for domain adaptation*, in Advances in neural information processing systems, 2007, pp. 137–144.

[5] J. Blitzer, M. Dredze, F. Pereira, et al., *Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification*, in ACL, vol. 7, 2007, pp. 440–447.

[6] J. Blitzer, R. McDonald, and F. Pereira, *Domain adaptation with structural correspondence learning*, in Proceedings of the 2006 conference on empirical methods in natural language processing, Association for Computational Linguistics, 2006, pp. 120–128.

[7] J. Bollen, H. Mao, and X. Zeng, *Twitter mood predicts the stock market*, Journal of computational science, 2 (2011), pp. 1–8.

[8] M. Chen, Z. Xu, K. Weinberger, and F. Sha, *Marginalized denoising autoencoders for domain adaptation*, arXiv preprint arXiv:1206.4683, (2012).

[9] S. Cohen and L. Guibasm, *The earth mover's distance under transformation sets*, in Computer vision, 1999. The proceedings of the seventh IEEE international conference on, vol. 2, IEEE, 1999, pp. 1076–1083.

[10] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, *Optimal transport for domain adaptation*, arXiv preprint arXiv:1507.00504, (2015).

[11] CrowdFlower, *Data for everyone*, 2017.

[12] M. Dragoni, A. G. Tettamanzi, and C. D. C. Pereira, *Dranziera: an evaluation protocol for multi-domain opinion mining*, in Tenth International Conference on Language Resources and Evaluation (LREC 2016), European Language Resources Association (ELRA), 2016, pp. 267–272.

[13] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, *Unsupervised visual domain adaptation using subspace alignment*, in Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE, 2013, pp. 2960–2967.

[14] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, *Domain-adversarial training of neural networks*, The Journal of Machine Learning Research, 17 (2016), pp. 2096–2030.

[15] X. Glorot, A. Bordes, and Y. Bengio, *Domain adaptation for large-scale sentiment classification: A deep learning approach*, in Proceedings of the 28th international conference on machine learning (ICML-11), 2011, pp. 513–520.

[16] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, *A kernel two-sample test*, Journal of Machine Learning Research, 13 (2012), pp. 723–773.

[17] V. Hatzivassiloglou and K. R. McKeown, *Predicting the semantic orientation of adjectives*, in Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 1997, pp. 174–181.

[18] Y. He, C. Lin, and H. Alani, *Automatically extracting polarity-bearing topics for cross-domain sentiment classification*, in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, 2011, pp. 123–131.

[19] F. H. Khan, U. Qamar, and S. Bashir, *Swims: Semi-supervised subjective feature weighting and intelligent model selection for sentiment analysis*, Knowledge-Based Systems, 100 (2016), pp. 97–111.

[20] S. Kolouri, A. B. Tosun, J. A. Ozolek, and G. K. Rohde, *A continuous linear optimal transport approach for pattern analysis in image datasets*, Pattern recognition, 51 (2016), pp. 453–462.

[21] D. Kotzias, M. Denil, N. De Freitas, and P. Smyth, *From group to individual labels using deep features*, in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 597–606.

[22] W. M. Kouw, L. J. Van Der Maaten, J. H. Krijthe, and M. Loog, *Feature-level domain adaptation*, Journal of Machine Learning Research, 17 (2016), pp. 1–32.

[23] S. Kullback and R. A. Leibler, *On information and sufficiency*, The annals of mathematical statistics, 22 (1951), pp. 79–86.

[24] Q. Le and T. Mikolov, *Distributed representations of sentences and documents*, in Proceedings of the 31st International Conference on Machine Learning (ICML-14), 2014, pp. 1188–1196.

[25] L. Lee, *On the effectiveness of the skew divergence for statistical language analysis.*, in AISTATS, Citeseer, 2001, pp. 65–72.

[26] J. Lin, *Divergence measures based on the shannon entropy*, IEEE Transactions on Information theory, 37 (1991), pp. 145–151.

[27] W. Mayner, *pyemd*, 2017.

[28] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781, (2013).

[29] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, *Domain adaptation via transfer component analysis*, IEEE Transactions on Neural Networks, 22 (2011), pp. 199–210.

[30] B. Pang, L. Lee, et al., *Opinion mining and sentiment analysis*, Foundations and Trends® in Information Retrieval, 2 (2008), pp. 1–135.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, *Scikit-learn: Machine learning in Python*, Journal of Machine Learning Research, 12 (2011), pp. 2825–2830.

[32] O. Pele and M. Werman, *A linear time histogram metric for improved sift matching*, in Computer Vision–ECCV 2008, Springer, October 2008, pp. 495–508.

[33] ———, *Fast and robust earth mover's distances*, in 2009 IEEE 12th International Conference on Computer Vision, IEEE, September 2009, pp. 460–467.

[34] B. Plank, *What to do about non-standard (or non-canonical) language in nlp*, arXiv preprint arXiv:1608.07836, (2016).

[35] B. Plank and G. Van Noord, *Effective measures of domain similarity for parsing*, in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, 2011, pp. 1566–1576.

[36] S. Ruder, P. Ghaffari, and J. G. Breslin, *Data selection strategies for multi-domain sentiment analysis*, arXiv preprint arXiv:1702.02426, (2017).

[37] B. K. Sriperumbudur, K. Fukumizu, A. Gretton, B. Schölkopf, and G. R. Lanckriet, *On integral probability metrics,\phi-divergences and binary classification*, arXiv preprint arXiv:0901.2698, (2009).

[38] S. Tan, X. Cheng, Y. Wang, and H. Xu, *Adapting naive bayes to domain adaptation for sentiment analysis*, Advances in Information Retrieval, (2009), pp. 337–349.

[39] L. Terveen, W. Hill, et al., *Phoaks: a system for sharing recommendations*, Communication of the ACM, 40 (1997), pp. 59–63.

[40] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, *Predicting elections with twitter: What 140 characters reveal about political sentiment.*, Icwsm, 10 (2010), pp. 178–185.

[41] University of Michigan, *UMICH SI650 - Sentiment Classification*, 2011.

[42] V. van Asch, *mmd.py*, 2012.

[43] K. Vogt, A. Paul, J. Ostermann, F. Rottensteiner, and C. Heipke, *Boosted unsupervised multi-source selection for domain adaptation*, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 4 (2017), p. 229.

[44] D. Wu, V. J. Lawhern, and B. J. Lance, *Reducing offline bci calibration effort using weighted adaptation regularization with source domain selection*, in Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on, IEEE, 2015, pp. 3209–3216.

[45] E. W. Xiang, S. J. Pan, W. Pan, J. Su, and Q. Yang, *Source-selection-free transfer learning*, in IJCAI proceedings-international joint conference on artificial intelligence, vol. 22, 2011, p. 2355.

[46] Y. Zhang, X. Hu, P. Li, L. Li, and X. Wu, *Cross-domain sentiment classification-feature divergence, polarity divergence or both?*, Pattern recognition letters, 65 (2015), pp. 44–50.