Journal of Machine Learning Research 1 (2015) 1-48 Submitted 4/15; Published 10/15

Feature Importance Regularization for Domain Adaptation

Wouter Kouw W.M.KOUW@TUDELFT.NL
Department of Intelligent Systems
Delft University of Technology
Mekelweg 6, 2628CD, the Netherlands

Jesse Krijthe J.H.KRIJTHE@TUDELFT.NL
Department of Intelligent Systems
Delft University of Technology
Mekelweg 6, 2628 CD Delft, the Netherlands
Department of Epidemiology
Leiden University Medical Center
Albinusdreef 2, 2333 ZA Leiden, the Netherlands

Marco Loog M.LOOG@TUDELFT.NL
Department of Intelligent Systems
Delft University of Technology
Mekelweg 6, 2628 CD Delft, the Netherlands
The Image Group
IT University of Copenhagen
Rued Langgaards Vej 7, 2300 Copenhagen S, Denmark

Laurens van der Maaten L.J.P.VANDERMAATEN@TUDELFT.NL
Department of Intelligent Systems
Delft University of Technology
Mekelweg 6, 2628 CD Delft, the Netherlands
Facebook Research

Editor: Leslie Pack Kaelbling

Abstract

Domain adaptation is a supervised learning problem where the training and test data come from different distributions. In this paper we study a classifier that assumes a particular dependence between two domains. Modeling that dependence with a conditional Bernoulli distribution leads to adapting to the relative importance of features. We show that if the model is correct, the adapted classifier is equivalent to the target classifier. Applying it to real-world datasets shows that it competes well with popular adaptation strategies.

Keywords: Domain adaptation, transfer learning, covariate shift, empirical risk minimization, data-dependent regularization, dropout.

1. Introduction

In supervised learning, classifiers learn from labeled samples and predict labels of new samples. It is implicitly assumed that these new samples come from the same distribution as the already observed ones. However, this assumption often does not hold. For example, a classifier that has learned to recognize one person's speech can not easily recognize another. Their sound patterns have different statistics (Leggetter and Woodland, 1995).

A common occurence of this problem is when we have a specific method of gathering data. Classifiers trained on these measurements do not generalize to other acquisition methods. For example, training a classifier to segment images from one MRI scanner does not imply that it will perform well on a different scanner (van Opbroek et al., 2013). The scanning protocols and mechanical specifications ensure that the intensity values are distributed differently over voxels. Another example is object recognition in computer vision (Saenko et al., 2010). If training images have been taken with one camera and testing images with another, the pixel intensity distributions may differ due to lens or exposure settings. In natural language processing it is hard to classify movie reviews after training on tweets due to the informal word usage (Peddinti and Chintalapoodi, 2011). As for bioinformatics, integrating genome sequence data from multiple acquisition centers can also be considered a domain adaptation problem, because cells are subject to different laboratory conditions (Borgwardt et al., 2006).

In each of these cases the bias in the sampling procedure leads to different probabilities over the same events. Transfer learning is the general term to any learning problem where the test and training data are assumed to originate from different distributions. If both sets are measured using the same features, the problem is referred to as domain adaptation. Each distribution over the same set of events is then a domain, where the labeled one is referred to as a source and the unlabeled one as a target.

The problem is closely related to a number of other problems. Whenever target labels are available during training, there are in fact two related supervised learning problems, which makes it a multi-task learning problem (Pan and Yang, 2010; Moreno-Torres et al., 2012). If an oracle is available that is capable of assigning labels to target samples, then it is an active learning problem (Cohn et al., 1996), and if the distribution of the target samples changes for small batches, then it should be treated as an online learning problem (Ma et al., 2009).

In some situations multiple domains are available for training (Mansour et al., 2009). More information allows for a wider range of strategies, but it also introduces more complexity as the influence of each domain has to be weighted. In this paper we restrict ourselves to pairwise combinations of training on one domain and testing on another.

Our method looks at how two domains are dependent on each other and incorporates the modeled dependency in our classification model.

1.1 Related Work

There is a theoretical result by Shimodaira (2000) that rewrites the expected classification error in the target domain as a function of the source domain and a weighting factor for each event: $R = \int_X \int_Y \frac{p_{Z,Y}(x,y)}{p_{X,Y}(x,y)} \left[-\log p_{Y|X}(y|x) \right] \mathrm{d}y \; \mathrm{d}x$. It reduces the likelihood of source samples that are very dissimilar from the target samples and increases the likelihood of samples that are similar. However, no information is available on the target joint distribution. If the assumption is made that the posteriors $p_{Y|Z}(y|x) = p_{Y|X}(y|x)$ are equal, the reweighting factor reduces to $\frac{p_Z(x)}{p_X(x)}$. Learning with this assumption is referred to as *covariate shift* and is very appealing as it converges to the target classifier in the limit.

In the original paper the data is assumed to be Gaussian and these probabilities are evaluated after maximum likelihood fitting of Gaussian distributions on each domain. However, the inversion of one of these covariance matrices is very expensive for high-dimensional datasets. Alternatively, one can also train a classifier to discriminate the two domains and use the posterior probabilities as the reweighting factor.

The estimation of the weights in general is very sensitive to how dissimilar the two domains are. If the probability masses of both domain distributions lie far apart, it leads to very small weights for a large number of samples and very large weights for a select few. This means the classifier will often find poor solutions. Ben-David et al. (2010) also shows that for particular types of dissimilarities, covariate shift is not a sufficient condition for successful learning. Our approach does not rely on reweighing instances and is not sensitive to estimation errors in the weights.

A different weight estimation method, proposed by Huang et al. (2007) and Gretton et al. (2009), relies on the maximum mean discrepancy between two distributions: $\text{MMD}(x, y) = \max_{h \in \mathcal{H}} \mid \mathbb{E}[h(x)] - \mathbb{E}[h(y)] \mid$. This discrepancy roughly states that two distributions are similar if the distance between the moments is small for any smooth continuous function applied to both sets of samples. If an appropriate function class is chosen, such as the reproducing kernel Hilbert space, it is possible to find weights that minimize the empirical MMD (Borgwardt et al., 2006). The approach is referred to as kernel mean matching and is less prone to downweighing a large number of samples. It is however computationally expensive, as it requires solving an optimization problem that is quadratic in the number of samples.

Blitzer et al. (2006) have proposed structural correspondence learning, a method that augments the data with a domain-invariant subspace. In order to create the subspace, one starts with identifying a set of features that are used frequently in both domains, referred to as the *pivot* features. The goal then is to learn a mapping from the data in both domains to these features through a binary classification problem for each pivot feature. The domain-invariant subspace then consists of the h principal components of the eigendecomposition of the mapping.

The succes of structural correspondence learning is heavily dependent on identifying features that are appropriate pivots. The authors indicate to look for features that are very frequent in both domains, but that does not imply that there are discrimative features in both domains that correlate with it. Our approach also relies on the frequency of feature use, but is not dependent on finding a correspondence with features that are discriminative in their own domain.

In this paper we are interested in what classifiers we obtain if we make a different assumption than covariate shift. More specifically, we assume that the two domains are dependent and that the labels are conditionally independent given the observations in one domain. This can be interpreted as a transformation of labeled source samples into labeled target samples. If we incorporate this assumption in our classification model, we obtain a classifier that can adapt to this transformation.

2. Feature Importance Regularization

Formally, domains are defined here to be random variables on the same sample space Ω and event set $\mathcal{F}$, but with different probability measures $\mathcal{Q}$ and $\mathcal{P}$. We refer to the random variables associated with these probability spaces as X on $(\Omega, \mathcal{F}, \mathcal{Q})$ and Z on $(\Omega, \mathcal{F}, \mathcal{P})$. Usually, we refer to a *source* domain if we have both data and labels $\{\mathbf{x_i}, y_i\}$ for $i = 1, ...N_X$ and a *target* domain if we're interested in predicting the labels $\{\mathbf{z_j}\}$ for $j = 1, ...N_Z$. Probability distributions will have subscripts that designate to which random variable they belong, for example $p_X(x)$. Note that we assume here that both domains lie on the same feature space, $\mathbf{x_i}, \mathbf{z_j} \in \mathbb{R}^M$. If this is not the case, then it is necessary to map both domains to a common representation. There has been extensive research in this area, but it is outside the scope of this paper (Ben-David et al., 2007; Shao et al., 2014).

2.1 Target risk function

The empirical risk minimization framework studies the expectation of a loss function, or *risk*, to evaluate a hypothetical classification function f. Here we shall express the risk in the target domain as a function of the source domain and a conditional distribution. Note that Z, X are the random variables for the target and the source domain, respectively.

$$R = \int_Z \int_Y \ell(f(z), y) \ p_{Y,Z}(y, z) \ \mathrm{d}y \ \mathrm{d}z$$

$$R = \int_Z \int_Y \int_X \ell(f(z), y) \ p_{Y,Z,X}(y, z, x) \ \mathrm{d}x \ \mathrm{d}y \ \mathrm{d}z$$

$$R = \int_Z \int_Y \int_X \ \ell(f(z), y) \ p_{Y|Z,X}(y|z, x) \ p_{Z|X}(z|x) \ p_X(x) \ \mathrm{d}x \ \mathrm{d}y \ \mathrm{d}z$$

If we assume conditional independence of the labels and target data given source data: $Y \perp\!\!\!\perp Z|X$, we can further simplify to:

$$R = \int_Z \int_Y \int_X \ \ell(f(z), y) \ p_{Y|X}(y|x) \ p_{Z|X}(z|x) \ p_X(x) \ \mathrm{d}x \ \mathrm{d}y \ \mathrm{d}z$$

$$R = \int_Z \mathbb{E}_{p_{Y,X}} \left[\ \ell(f(z), y) \ p_{Z|X}(z|x) \ \right] \ \mathrm{d}z \tag{1}$$

The expectation can be approximated with source data $\{(x_i, y_i)\}$ for $i = 1, .., N_X$:

$$R(x, y) \approx \int_Z \frac{1}{N_X} \ \sum_i^{N_X} \ell(f(z), y_i) p_{Z|X}(z|x_i) \ \mathrm{d}z \tag{2}$$

Note that the conditional distribution in (1) has changed from conditionial on any sample in the source domain $p_{Z|X}(z|x)$ to being conditional on a specific sample x_i; $p_{Z|X}(z|x_i)$ (2). Such a conditional can be thought of as a probability distribution centered on each sample, for example $\mathcal{N}(z|x_i, \sigma^2)$. Bishop (1995) has shown that for specifically a Gaussian distribution, the least-squares formulation can be rewritten as ridge regression: $\int_Z N_X^{-1} \ \sum_i^{N_X} (y_i -$

$\mathbf{w}^\top \mathbf{z})^2 \mathcal{N}(\mathbf{z}|\mathbf{x}_i, \sigma^2)\ \mathrm{d}z = N_X^{-1}\ \sum_i^{N_X} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \sigma^2 \|\mathbf{w}\|^2$. In his paper, he interprets this finding as *noise* in the measurements. Here we assume that this conditional distribution reflects the change in probability of an event in the target domain, given that event in the source domain. In this interpretation, we denote this conditional distribution as the *transfer* distribution.

Our main assumption is that the target data is conditionally independent of the labels given the source data ($Z \perp\!\!\!\perp Y|X$). Because we can calculate the distribution over the target data using $p_{Z|X}(z|x)$ and the distribution over the labels using $p_{Y|X}(y|x)$, observing the target data will not add information on the label distribution.

It can be interpreted as the fact that if we transform the source data, the labels transform along with them. It is different from covariate shift, because it implies that the posteriors are not equal $p_{Y|X}(y|x) \neq p_{Y|Z}(y|z)$. For problems that we will encounter where this assumption is valid, we can successfully adapt and outperform a naive classifier.

2.2 Classification

The empirical risk (2) is dependent on choosing a loss function ℓ. Some popular choices are the quadratic loss (used in least-squares regression), the exponential loss (used in boosting), the hinge loss (used in support vector machines) and the logistic loss (used in generalized linear models). The formulation in (2) has been studied before in the context of *dropout training* for the quadratic and logistic loss by Wager et al. (2013) and Rostamizadeh et al. (2011), and hinge loss by Chen et al. (2014). It was extended for the exponential loss in the context of *corruption* distributions by van der Maaten et al. (2013). Here we chose to employ the quadratic loss and the logistic loss for computational reasons. Throughout this paper, we will focus on the class of linear functions $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$.

2.2.1 Quadratic loss

The quadratic loss $l(f(\mathbf{x}), y) = (y - \mathbf{w}^\top \mathbf{x})^2$ is an interesting choice, because the expectation is analytically tractable and the gradient has a closed-form solution.

$$\begin{aligned} R(\mathbf{x}, \mathbf{y}) &= \int_Z \frac{1}{N_X} \sum_i^{N_X} (y_i - \mathbf{w}^\top \mathbf{z})^2\ p_{Z|X}(\mathbf{z}|\mathbf{x}_i)\ \mathrm{d}\mathbf{z} \\ &= \mathbf{y}^\top \mathbf{y} - 2\mathbf{w}^\top \mathbb{E}_{\tilde{p}}[\mathbf{z}]\mathbf{y} + \mathbf{w}^\top \mathbb{E}_{\tilde{p}}[\mathbf{z}]\mathbb{E}_{\tilde{p}}[\mathbf{z}]^\top \mathbf{w} + \mathbf{w}^\top \mathbb{V}_{\tilde{p}}[\mathbf{z}]\mathbf{w} \end{aligned} \tag{3}$$

where $\tilde{p}$ is shorthand for the expectation with respect to the transfer distribution $p_{Z|X}(\mathbf{z}|\mathbf{x}_i)$. Deriving the gradient for this loss function and setting it to 0:

$$\mathbf{w}^* = \Big(\mathbb{E}_{\tilde{p}}[\mathbf{z}]\mathbb{E}_{\tilde{p}}[\mathbf{z}]^\top + \mathbb{V}_{\tilde{p}}[\mathbf{z}]\Big)^{-1} \mathbb{E}_{\tilde{p}}[\mathbf{z}]\mathbf{y}^\top \tag{4}$$

2.2.2 Logistic loss

The logistic loss stems from the negative log-likelihood of generalized linear model with a logit link function: $l(f(\mathbf{x}), y) = -\log(p|\mathbf{x}) = -y\mathbf{w}^\top \mathbf{x} + \log(\sum_{y'} \exp(-y'\mathbf{w}^\top \mathbf{x}))$. Plugging in this loss into the risk function gives:

$$
\begin{aligned}
R(\mathbf{x},\mathbf{y}) &= \int_Z \frac{1}{N_X}\sum_{i}^{N_X} \big[- y_i \mathbf{w}^\top \mathbf{z} + \log \sum_{y'} \exp(-y'\mathbf{w}^\top\mathbf{z})\big] \; p_{Z|X}(\mathbf{z}|\mathbf{x}_i) \; \mathrm{d}\mathbf{z} \\
&= \frac{1}{N_X}\sum_{i}^{N_X} -y_i \mathbf{w}^\top \mathbb{E}_{\tilde{p}}[\mathbf{z}] + \mathbb{E}_{\tilde{p}}\big[\log\sum_{y'}\exp(y'\mathbf{w}^\top\mathbf{z})\big] \qquad (5)
\end{aligned}
$$

Because of the nonlinearity, we can no longer calculate the expectation analytically. But as in Wager et al. (2013), we approximate the expectation of the log-partition function using a Taylor expansion around the inner product of the weights and the data. Note that this generalizes to K classes if we take an $M \times K$ weight matrix $\mathbf{W}$. The loss is then approximately:

$$
\begin{aligned}
R(\mathbf{x},\mathbf{y}) \approx & \frac{1}{N_X}\sum_{i}^{N_X} -y_i \mathbf{w}^\top \mathbb{E}_{\tilde{p}}[\mathbf{z}] + \log\sum_{y'}\exp(y'\mathbf{w}^\top\mathbf{x}_i) + \\
& \frac{\sum_{y'} y' \exp(y'\mathbf{w}^\top\mathbf{x}_i)}{\sum_{y'}\exp(y'\mathbf{w}^\top\mathbf{x}_i)}\mathbf{w}^\top(\mathbb{E}_{\tilde{p}}[\mathbf{z}] - \mathbf{x}_i) + \\
& 2\sigma\big(-2\mathbf{w}^\top\mathbf{x}_\mathbf{i}\big)\sigma\big(2\mathbf{w}^\top\mathbf{x}_i\big)\mathbf{w}^\top\Big(\mathbb{V}_{\tilde{p}}[\mathbf{z}] + (\mathbb{E}_{\tilde{p}}[\mathbf{z}] - \mathbf{x}_i)(\mathbb{E}_{\tilde{p}}[\mathbf{z}] - \mathbf{x}_i)^\top\Big)\mathbf{w} \qquad (6)
\end{aligned}
$$

with the gradient as:

$$
\begin{aligned}
\frac{\partial R}{\partial \mathbf{w}} = & \frac{1}{N_X}\sum_{i}^{N_X} -y_i \mathbb{E}_{\tilde{p}}\big[\mathbf{z}\big] + \frac{\sum_{y'} y' \exp(y'\mathbf{w}^\top\mathbf{x}_i)}{\sum_{y'}\exp(y'\mathbf{w}^\top\mathbf{x}_i)}\mathbf{x}_i + \\
& \Bigg[\Big(1 - \Big[\frac{\sum_{y'} y' \exp(y'\mathbf{w}^\top\mathbf{x}_i)}{\sum_{y'}\exp(y'\mathbf{w}^\top\mathbf{x}_i)}\Big]^2\Big)\mathbf{x}_i\mathbf{w}^\top + \frac{\sum_{y'} y' \exp(y'\mathbf{w}^\top\mathbf{x}_i)}{\sum_{y'}\exp(y'\mathbf{w}^\top\mathbf{x}_i)}\Bigg]\big(\mathbb{E}_{\tilde{p}}\big[\mathbf{z}\big] - \mathbf{x}_i\big) + \\
& 4\sigma\big(-2\mathbf{w}^\top\mathbf{x}_\mathbf{i}\big)\sigma\big(2\mathbf{w}^\top\mathbf{x}_i\big)\Big[\Big(\big(\sigma(-2\mathbf{w}^\top\mathbf{x}_i) - \sigma(2\mathbf{w}^\top\mathbf{x}_i)\big)\mathbf{x}_i\mathbf{w}^\top + 1\Big) \\
& \Big(\mathbb{V}_{\tilde{p}}[\mathbf{z}] + (\mathbb{E}_{\tilde{p}}[\mathbf{z}] - \mathbf{x}_i)(\mathbb{E}_{\tilde{p}}[\mathbf{z}] - \mathbf{x}_i)^\top\Big)\mathbf{w}\Big] \qquad (7)
\end{aligned}
$$

Here we don't obtain a closed-form solution, but the gradient is convex and one can use an iterative optimization algorithm to find $\mathbf{w}^*$.

2.3 Transfer distribution

The new formulation of the target risk (1) states that an adapted classifier trained on source samples can be equivalent to a classifier trained on the target samples, if the underlying transfer distribution is found.

Our strategy is based on proposing a source, target and transfer distribution from an appropriate family of distributions and fitting them to the data. First, the source and target data, $\hat{p}_Z(z)$ and $\hat{p}_X(x)$,are modeled using maximum likelihood fitting. Then, the joint distribution of the source and the transfer model is taken where we marginalize over the source domain, $\int_X p_{Z|X}(z|x,\theta)\; \hat{p}_X(x)\; \mathrm{d}x$. Finally, the divergence between the marginalized joint and the target model is minimized with respect to the transfer parameters θ.

The choice of a family of distributions to model the transfer will determine if the classifier can appriopriately adapt. Here we focus on how often features are used in sparse nonnegative data, with the assumption that features that are used less frequently in the target domain are also less important there. For this dependence we study Bernoulli distributions but for continuous data it would be wiser to study a different choice of distributions.

There are a number of settings where differences in the relative importance of a feature occur. In natural language processing, if a feature is absent in a target sample, it means that the writer has not used that word in the document. Similarly, any visual bag-of-words representation for image processing relies on histograms, which can also have bins that are used less frequently in the target domain.

The classifier we obtain from choosing this transfer distribution will regularize features that are relatively less important and will be referred to as a *feature importance regularized* classifier (fir).

2.3.1 Bernoulli transfer

Feature presence is described with a Bernoulli distribution, $x = 1$ for presence and $x = 0$ for absence, and the rate parameter tells us how probable it is for the feature to be present in a sample. If the data consists of integers, a dummy variable is taken, where $x = 0$ for integer $= 0$ and $x = 1$ for integer ≥ 1. Features are assumed to be independent and we will be studying separate univariate distributions on each feature:

$$p_X(x_m | r_{Xm}) = r_{Xm}^{x_m} \; (1 - r_{Xm})^{(1-x_m)} \tag{8}$$

$$p_Z(z_m | r_{Zm}) = r_{Zm}^{z_m} \; (1 - r_{Zm})^{(1-z_m)} \tag{9}$$

where r_{Xm} is the rate parameter of feature m in the source data and r_{Zm} of the target data.

The relative presence of a feature is encoded with a conditional Bernoulli distribution. The reasoning is as follows: for features that are roughly equally present in both domains (i.e. the events $z = 1 \mid x = 1$ and $z = 0 \mid x = 0$), the solution after training on the source domain should be unaltered. There is an additional assumption here that these features do not change from being positively correlated with a class to being negatively correlated with that class, and vice versa.

We will not model if a feature is more present in the target domain (i.e. the event $z = 1 \mid x = 0$). Since the target labels are not known and the feature does not seem to be discriminative in the source domain, we have no knowledge of whether it is positively or negatively correlated with a class in the target domain. Therefore, the classifier should not assign a large weight.

Features that are relatively less frequently used in the target domain (i.e. the event $z = 0 \mid x = 1$) should not be assigned large weights, since we have prior knowledge that the feature will not discriminate between the classes. A classifier that has this information will most likely find a different solution which relies only on the remaining features. The conditional distribution we obtain by assigning probabilities to these events is also known as *dropout* and has been used before as a data-dependent regularizer during training (Wager et al., 2013; Rostamizadeh et al., 2011).

Here we alter this formulation further by replacing the event $z = 1 \mid x = 1$ by a new outcome that is dependent on the probability of dropout θ. This has the effect that it ensures that the expectation of the transfer distribution is the same as the data point on which it is centered: $\mathbb{E}_{\tilde{p}}[z] = x_i$. Otherwise, the transfer distribution introduces biases during training.

The Bernoulli transfer for feature m is described by:

$$p_{Z|X}(z_m|x_m;\theta_m) = \begin{cases} 0 & \text{if } z_m = 1 \mid x_m = 0 \\ \theta & \text{if } z_m = 0 \mid x_m = 1 \\ 1-\theta & \text{if } z_m = \frac{x_m}{1-\theta_m} \mid x_m = 1 \\ 1 & \text{if } z_m = 0 \mid x_m = 0 \end{cases} \tag{10}$$

Equations (3) and (6) need the moments of the transfer distribution, which are:

$$\mathbb{E}_{\tilde{p}}\big[z_m\big] = x_m \tag{11}$$

$$\mathbb{V}_{\tilde{p}}\big[z_m\big] = \frac{\theta_m}{1-\theta_m} x_m^2 \tag{12}$$

2.3.2 Fitting the transfer distribution

The marginalized joint distribution for every feature is described by:

$$\begin{aligned} \hat{p}_Z(z_m|r_{Xm},\theta_m) &= \int_X p_{Z|X}(z_m|x_m,\theta_m)\ \hat{p}_X(x_m|r_{Xm})\ \mathrm{d}x_m \\ &= ((1-\theta_m)r_{Xm})^{z_m}(1-(1-\theta_m)r_{Xm})^{1-z_m} \end{aligned} \tag{13}$$

Finding the optimal transfer parameter can be done by minimizing the divergence between the target and the marginalized joint:

$$\theta_m^* = \underset{\theta\in\Theta}{\arg\min}\ D_{KL}\Big(\ \hat{p}_Z(z_m|r_{Zm}) \parallel \hat{p}_Z(z_m|r_{Xm},\theta_m)\ \Big)$$

Note that θ_m is also a rate parameter and is therefore bounded in $[0,1]$. We can set the gradient with respect to θ_m to 0, which yields a constrained closed-form solution.

$$\theta_m^* = \max\left\{0, 1 - \frac{r_{Zm}}{r_{Xm}}\right\} \tag{14}$$

3. Experiments

In these experiments the behaviour of the fir classifier is studied under a controlled setting, followed by an evaluation on five real-world datasets reflecting common domain adaptation settings.

3.1 Artifical datasets

In order to create an artificial domain adaptation problem, we generate a dataset by sampling from Bernoulli distributions that reflect class-conditional likelihoods in the source domain. The target domain can then be constructed by transforming the source data using a known transfer distribution. This also gives us the true transfer parameters.

3.1.1 Convergence

In order to empirically test if the adapted classifier converges to the target classifier, we perform an experiment where it has to estimate the true transfer parameters and find the same solution as a classifier trained on the labeled target dataset.

The source data is transformed using a Bernoulli transfer distribution, where we set the values of both features to 0 for 30% of the samples ($\theta = [0.3\ 0.3]$), regardless of the label of those samples. Next we train a naive classifier on the source data (w_S), a naive classifier on the target data (w_T) and a FIR classifier (fir) on the source data. Figure 1 shows that the solutions found by our FIR classifier and the naive target classifier are equivalent, for both the quadratic loss (left) and the logistic loss (right).

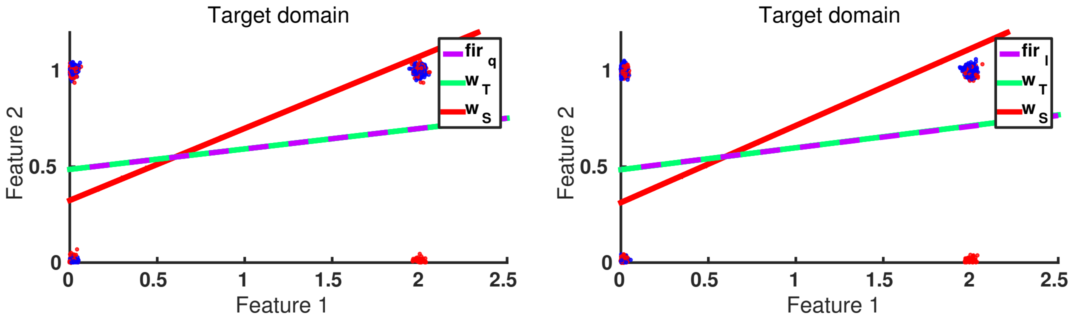

Figure 1: The data is generated by Bernoulli class-conditional likelihoods and transformed using a Bernoulli transfer. Red and blue dots show different classes that have been jittered for visualization purposes. The lines are the decision boundaries found by the naive source classifier (w_S), the naive target classifier (w_T) and the adapted FIR classifier (fir). Left shows the solutions of classifier with a quadratic loss and right with a logistic loss. Note that the adapted classifier finds the same boundary as the target classifier.

Interestingly, if we generate datasets from different distributions, such as Poissons, and estimate the transfer parameters through dummy variables, the classifier also converges to the naive target classifier (see figure 2). Successful adaptation does not depend on the data, but on our ability to estimate the appropriate transfer distribution.

3.1.2 Learning curves

Learning curves can give us a measure of how fast the adapted classifier converges to the target classifier. Figure (3) shows the mean classification error (lines) on an independent test set and standard deviation (shaded) of the source, target and adapted classifier. As expected the source classifier outperforms the target and adapted classifiers on the source domain (dotted lines), while the other two outperform the source classifier on the target domain (solid lines). To calculate the standard error of the mean, we took 30 repeats for every sample size.

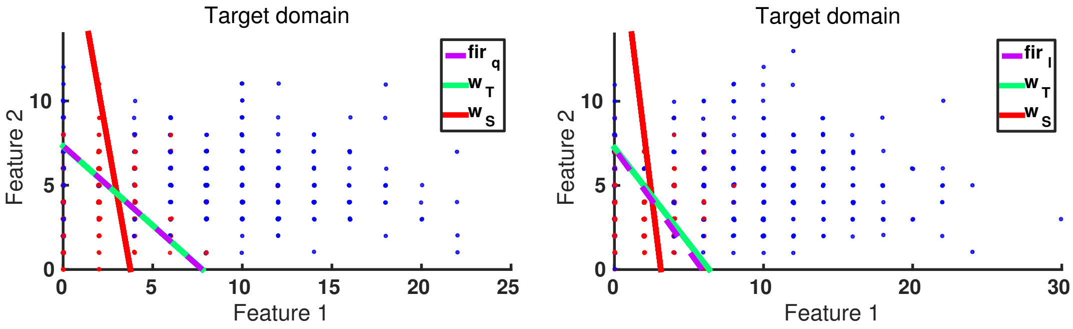

Figure 2: The plot shows a similar classification problem as figure 1, but here the data was generated by Poisson distributions. Note that the regardless of the data, if there is a Bernoulli transfer our FIR classifier can adapt to the target domain.

For a low-dimensional problem with no other domain dependencies than rate decreases, a few samples are enough to converge. Most likely, the number of samples needed will increase with the number of dimensions.

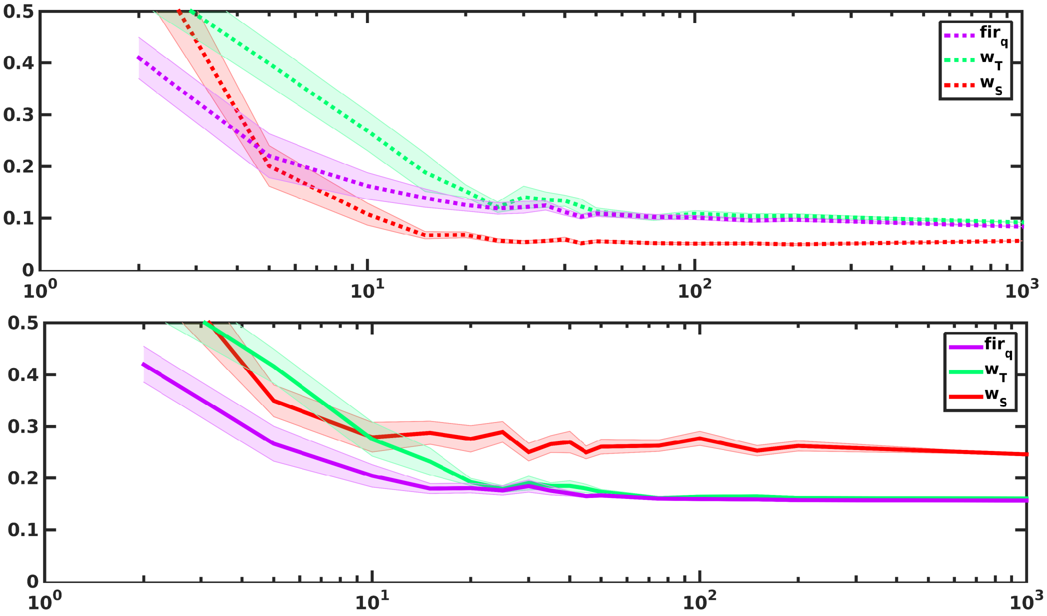

Figure 3: Learning curves of the source w_S, target w_T and adapted classifier fir_{qd}. The top figure shows the error on an independent validation set generated from the same distributions as the source domain. Bottom figure shows the error on a set generated from the target domain.

3.1.3 Robustness to parameter estimation errors

In this experiment, we study how fast classification error deteriorates as we increase the estimation error in the transfer parameters. Figure 4 shows an experiment where we added a value of 0, 0.1, 0.2 and 0.3 to the estimate the FIR classifier made. Note that the classifier is quite robust to small errors but deteriorates for large errors.

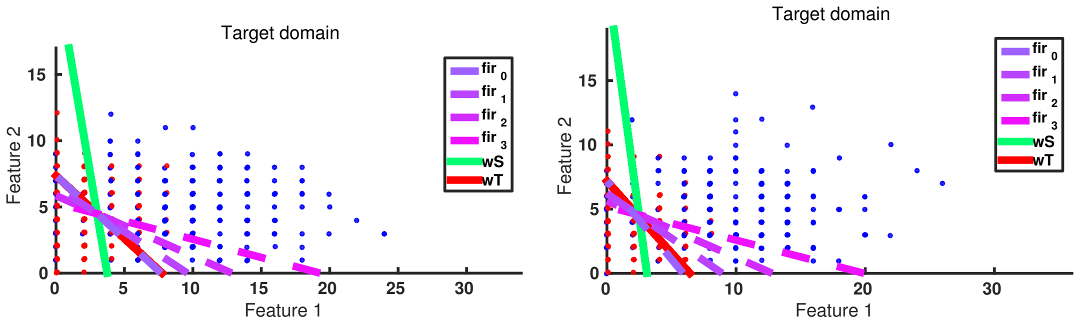

Figure 4: Experiment to determine how sensitive the model is to errors in estimating the parameters of the transfer distribution. FIR 0-3 shows the solution found by the adapted classifier for increments of 0, 0.1 0.2 and 0.3 to the transfer parameter estimate of feature 1. Left shows the quadratic loss and right shows the logistic loss.

3.2 Natural datasets

We have selected a number of domain adaptation problems consisting of nonnegative integer data, either in a (visual) bag-of-words representation or pixel intensities (between 0 and 255). They are typical scenarios where we believe a Bernoulli transfer occurs. Three are natural language based and two are based on images.

We compare our FIR classifier with 3 competing methods; IRW, KMM and SCL. To faciliate a fair evaluation of the adaption method, we train all methods using the same loss function. This is to ensure that performance measures do not depend on the classifier's flexibility or complexity. In order to compare the performance of an adapted classifier, we train and test classifiers on the same domain as well. The error rates for within-domain combinations can be considered the target classifiers and would represent the best performance the adapted classifier should be able to obtain.

For structural correspondence learning (SCL), as pivots we took the 30 features that were most frequent in both domains. Their values are predicted based on all features of both domains using a modified Huber loss, as in (Ando and Zhang, 2005). The 25 first principal components of the predictors form the subspace with which both domains are augmented. For our standard instance reweighting method (IRW) weights are estimated by computing the posterior domain probability using a logistic classifer. For kernel mean matching (KMM), a radial basis function kernel was used.

3.2.1 Crossvalidation

The goal of the classifier is to generalize well to new samples. In order to obtain a measure of generalization, we perform 3 repeats of 3-fold crossvalidation. This is a standard procedure for training and testing on the same domain. But for transfer learning, crossvalidation becomes trickier. If we split the training dataset in folds, we obtain a classifier that generalizes better to unseen source samples. However, that is not what we are interested in. The goal is to generalize better to unseen target samples. Therefore, we split the target data in folds where we hold out 1 fold for validation and use the remaining folds for estimating the transfer parameters.

3.2.2 Amazon

The Amazon sentiment analysis dataset is a popular domain adaptation set and consists of written product reviews, first used in Blitzer et al. (2007). The data is a 30 000 dimensional bag-of-words representation of 27677 reviews with the labels derived from the binarized 5-star rating. Each review is based on a product from one of four categories: books, dvds, electronics and kitchen appliances. Since people tend to use different words to describe liking a book versus liking a gadget, the product categories are the domains.

Figure 5 shows the normalized frequencies of some example words. It can be seen that some words are only used in one domain and these are the ones that we want to regularize heavily during training.

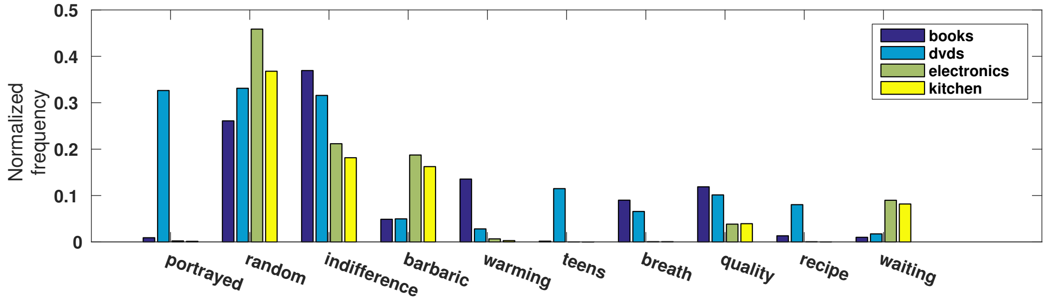

Figure 5: Normalized frequencies across domains of some example words of the amazon dataset. Note that some words can be quite common in one domain, but virtually absent in another.

In figure 6 we report the mean misclassification errors of each repeat of the crossvalidation procedure on all pairwise combinations of domains. The figure is set up in a grid where the row label indicates the source domain and the column label the target domain. The results indicate that none of the classifiers consistently outperforms the others. There are a few combinations where the adapted classifiers perform worse than the naive one, indicating that there are complex dependencies between these domains that are not accurately captured.

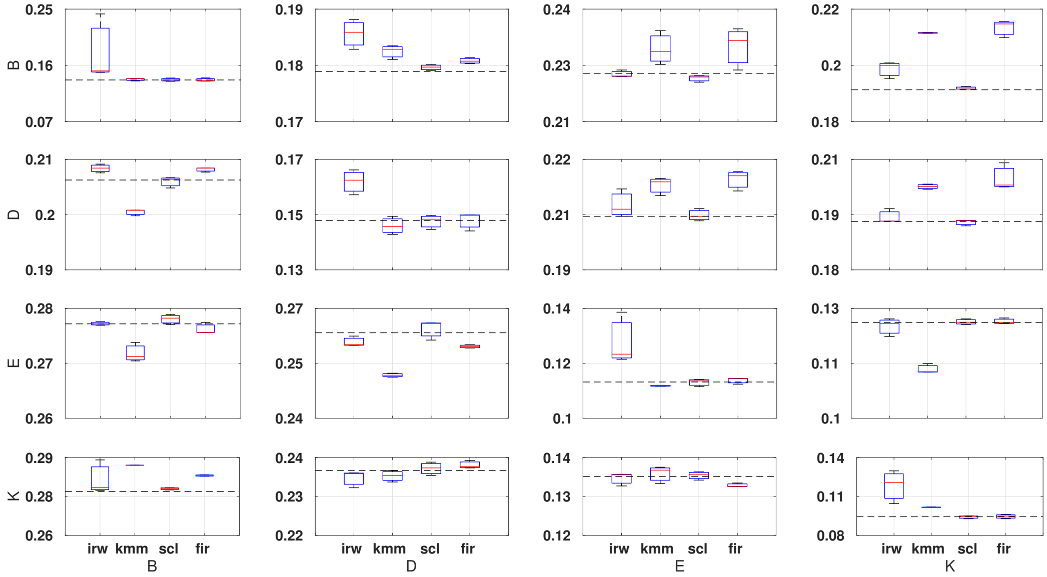

Figure 6: Box plots of the mean classification error of the repeats of the crossvalidation procedure. The dotted black line is the mean classification error of a naive logistic regressor. Grid shows pairwise combinations of traning on one source domain (rows) and testing on a target domain (columns). B = 'books', D = 'dvd', E = 'electronics' and K = 'kitchen'.

3.2.3 IMDB

The IMDB movie database is provided by Maas et al. (2011) and contains written reviews of movies labeled with a 1-10 star rating. We binarized the labels as > 5 postive and ≤ 5 as negative, leaving us with nicely balanced classes. For computational reasons we only kept features that had more than 100 counts in total, leaving us with 4180 features for 21954 samples. Since sentiment is described differently for different types of movies, the domains are the genres here. They are quite imbalanced, with 17 008 'action' movies, 1249 'family' movies and 3697 'war' movies.

The normalized frequencies of this dataset show few differences across domains (see figure 7). That would explain why for this dataset, there are also small differences between the error rates of the within-domain combinations versus the between-domain combinations. Figure 8 indicates that there is again no consistently better classifier. For some combinations a form of instance reweighting seems to be the method of choice, while for other combinations it leads to worse performance. fir and scl do not improve often but also do not perform worse than the naive classifier.

3.2.4 Spam

We can also view spam detection in different media as a domain adaptation problem. Here we have collected two datasets from the UCI machine learning repository, one containing the enron e-mail spam database and one containing the sms-spam dataset. Both datasets were

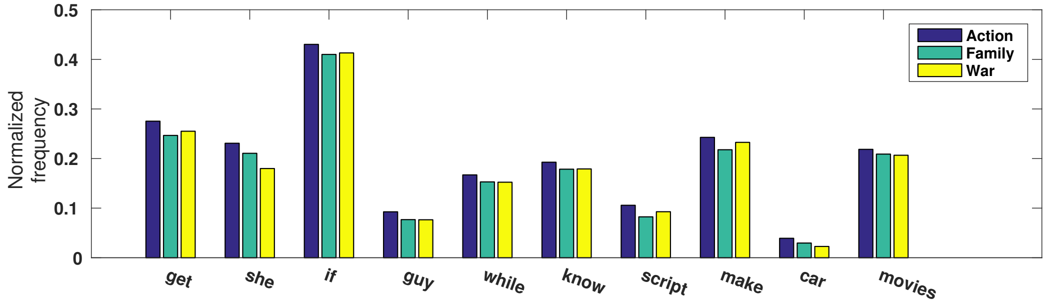

Figure 7: Normalized frequencies across domains of some example words of the imdb dataset. Note that the frequencies are more similar across domains than for the amazon dataset.

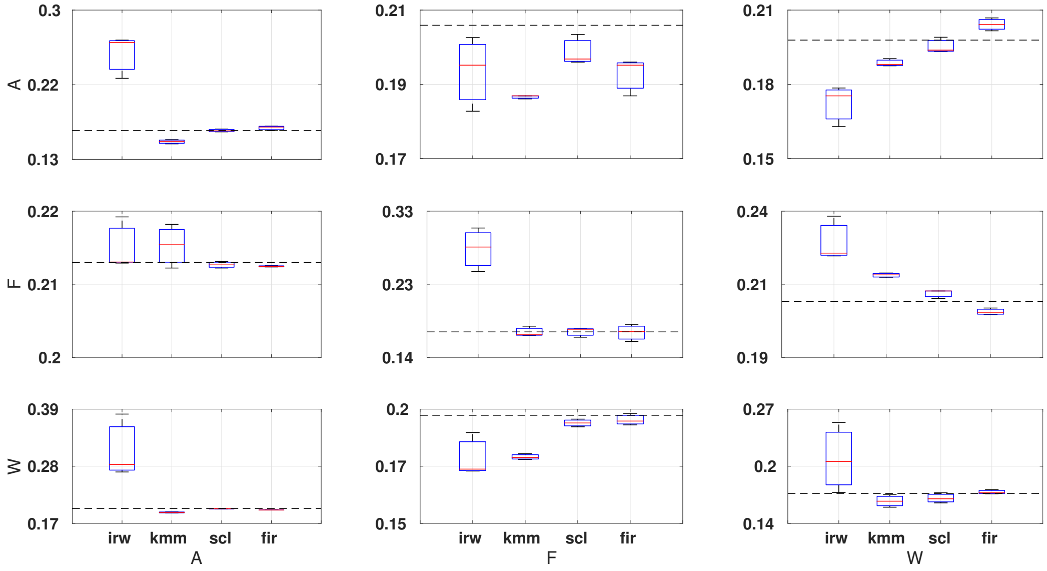

Figure 8: Box plots of the mean classification error of the repeats of the crossvalidation procedure. The dotted black line is the mean classification error of a naive logistic regressor. Grid shows pairwise combinations of traning on one source domain (rows) and testing on a target domain (columns). A= 'action', F = 'family' and W = 'war'.

parsed into a bag-of-words representation but we kept only the 4272 words that occurred in both domains as the new common feature space. Classes are balanced but the domains are very imbalanced; 5000 out of 40 000 documents are text messages.

Figure 9 is a good example of the differences between the two domains. Text messages use shortened words and do not have very common phrases, while email messages are more formal and tend to use the same kinds of words. It tells us that these domains are extremely different. If we study figure 10 we can see that the error is low within domains. Because we

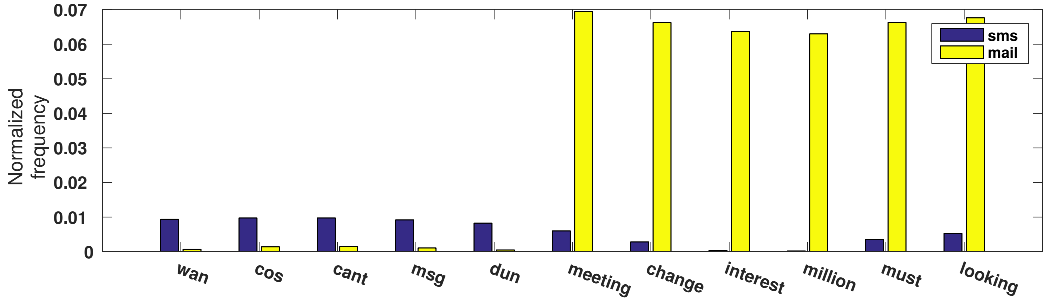

Figure 9: Normalized frequencies across domains of some example words of the spam dataset. Note that the frequencies and word types are very different for the domains.

are using linear classifiers, we can conclude that the domains are almost linearly separable. If we study the between-domain errors, we see that it rapidly falls back to chance. This is very interesting finding; apparently the domains are so dissimilar that few to none of the discriminative features in the source domain are discriminative in the target domain.

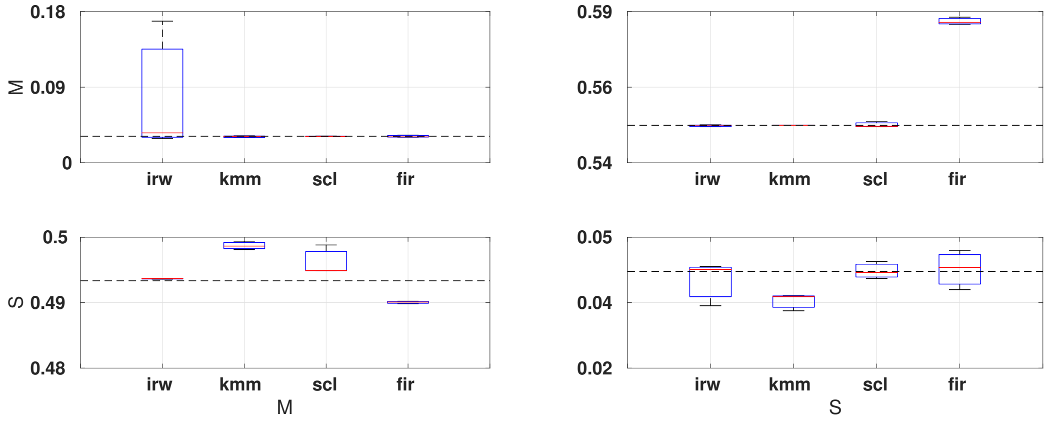

Figure 10: Box plots of the mean classification error of every repeat of the crossvalidation procedure. The dotted black line is the mean classification error of a naive logistic regressor. Grid shows pairwise combinations of traning on one source domain (rows) and testing on a target domain (columns). S = sms' and M = 'mail'.

3.2.5 Digits

For this image dataset, we gathered 3 different versions of recorded handwritten digits from the UCI machine learning repository: MNIST, SEMEION and USPS. All images are downsampled to 16 by 16 pixels, examples are show in figure 11. Again the classes are fairly balanced, but the domains are not; 60 000 images of MNIST, 1500 from SEMEION and 9000 from USPS.

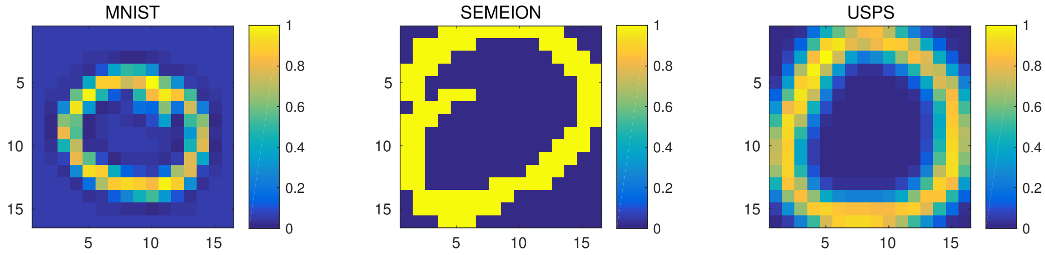

Figure 11: Examples of handwritten digits in each domain

Figure 12 tells us that some combinations of data lead to poor adaptation while others can have a tremendous improvement. If we inspect some of the examples from each domain (see figure 11), we can see for instance that USPS uses the outer features more than MNIST. Since we are only able to adapt to features that are used less frequently, going from the denser domain to the sparser domain will lead to improvements but not the other way around. For this dataset, the FIR classifier seems to be consistently outperforming the other methods. This could be due to the fact that the other methods were specifically built for text data while our method is more general.

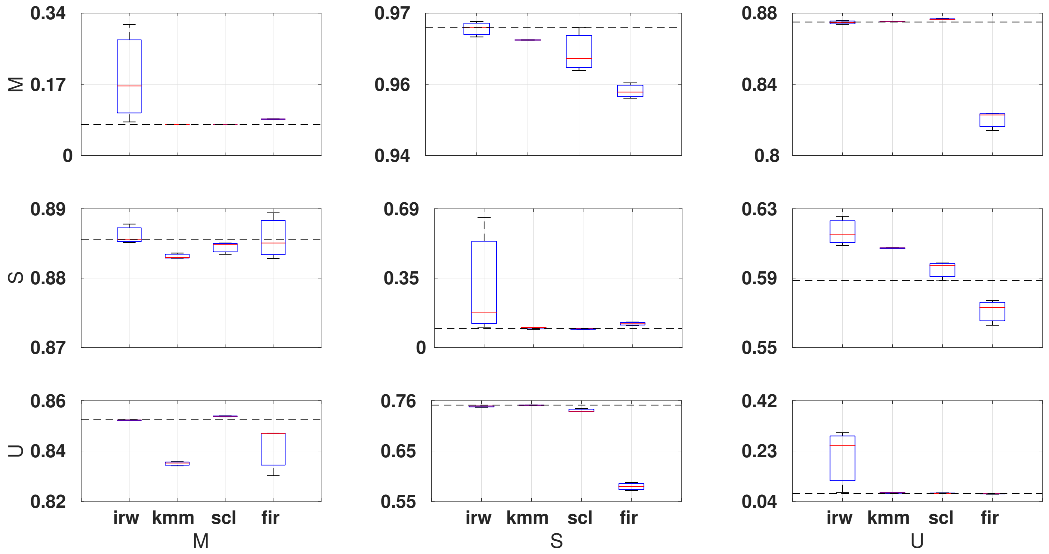

Figure 12: Box plots of the mean classification error of every repeat of the crossvalidation procedure. The dotted black line is the mean classification error of a naive logistic regressor. Grid shows pairwise combinations of traning on one source domain (rows) and testing on a target domain (columns). M='mnist', S='semeion' and U='usps'.

Here we can visualize some of the parameters and go through an example step-by-step. Figure 13 (left) shows the weights of a classifier trained on MNIST discriminating 0 from the other digits. The examples in figure 11 suggest that in the USPS set the inner features are used less than in MNIST which is confirmed by the transfer parameters (figure 13 middle). In order to adapt, the classifier should be informed of this difference and find a solution that only relies on the outer features (figure 13 right). This should increase its' performance on the USPS set, perhaps at the cost of its' performance on MNIST (figure 12).

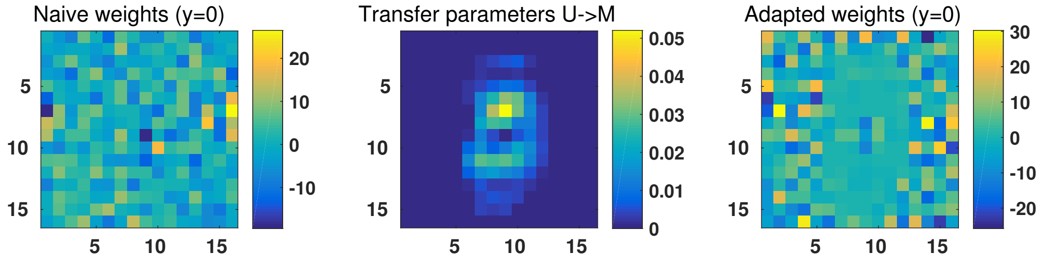

Figure 13: Visualization of the transfer parameters and the classification weights for training on USPS and testing on MNIST.

3.2.6 Office

This dataset comes from Saenko et al. (2010) and is made from images of objects gathered by three different methods: one from images uploaded to Amazon, one taken with a digital SLR camera and on with a webcam. For every image, a SURF descriptor (Bay et al., 2006) is used that descibes interest points in every image. It returns a 800 bin non-normalized histogram and reduces an image to a visual bag-of-words representation. One of the difficulties in this dataset is that there are 31 classes with roughly 120 samples for each class.

Figure 14 once again shows that the domains are very dissimilar, as the general performance of between-domain combinations is much worse than for within-domain combinations. kmm and fir seem to perform the best for this dataset.

3.3 Limitations

Here we look at some limitations of a fir classifier using Bernoulli transfer. Note that these are specific to the transfer distribution.

3.3.1 Target rate increases

One thing to consider are rate increases in the target domain. The adapted classifier is not sensitive to this transformation and will revert back to the naive source classifier. Figure 15 shows that the classifier is behaving as predicted. Methods that rely on learning a domain-invariant representation will not fail in this scenario.

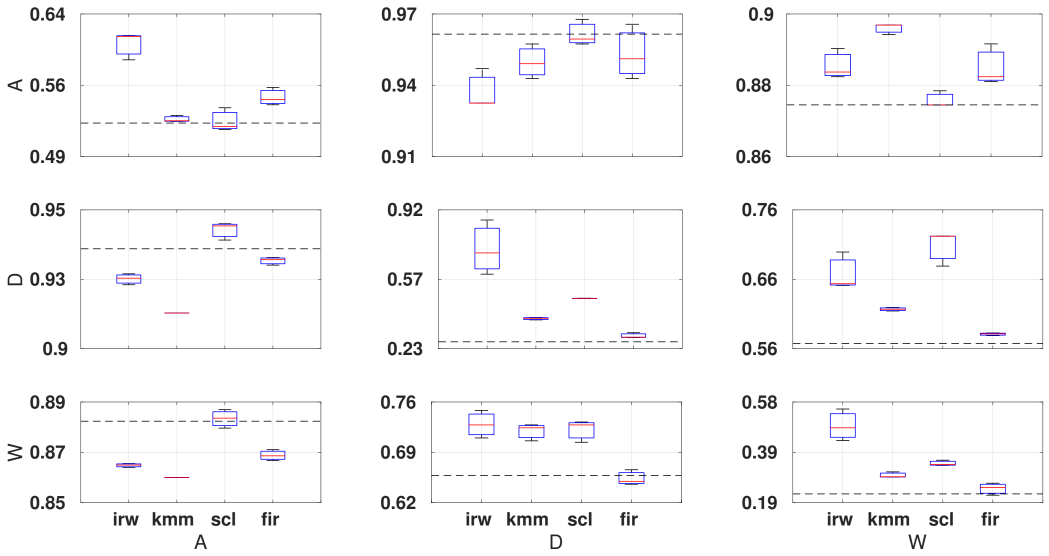

Figure 14: Box plots of the mean classification error of every repeat of the crossvalidation procedure. The dotted black line is the mean classification error of a naive logistic regressor. Grid shows pairwise combinations of traning on one source domain (rows) and testing on a target domain (columns). A='amazon', D='dslr' and W='webcam'.

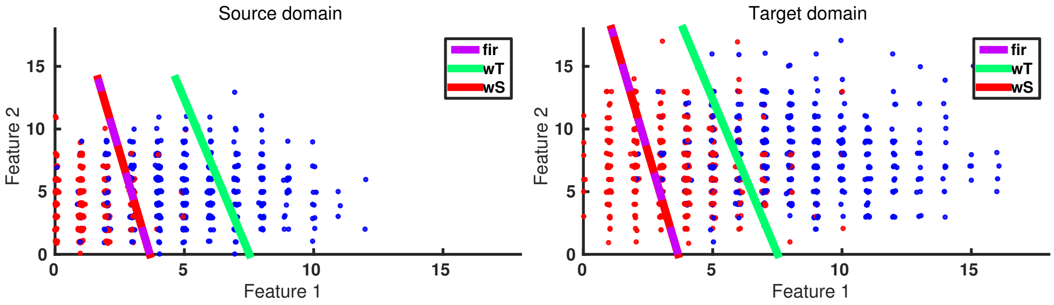

Figure 15: Dropout and blankout transfer distributions can not capture increases in the frequency of word use.

3.3.2 Rate-invariant transformations

For this experiment, we are interested in how robust the method is to transformations that only minimally affect the frequency of feature use. Figure 16 shows a newly generated target domain with the same rate parameters as the source domain, but with different class distributions. Note that since the estimated transfer parameters are nearly 0, the adapted classifier returns the same solution as the source classifier.

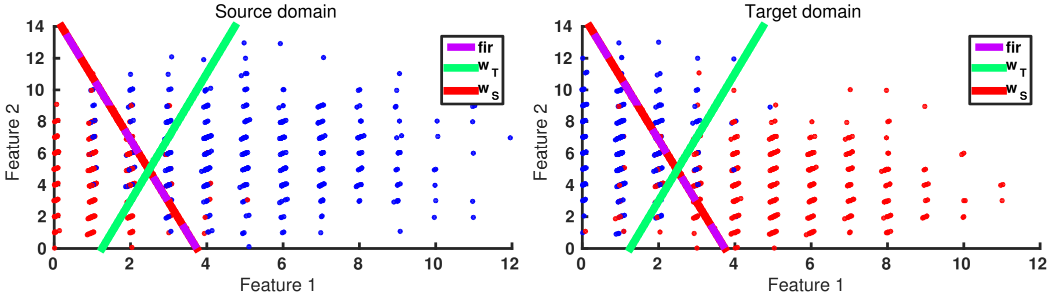

Figure 16: The model is not sensitive to transformations that do not change how much a feature is used.

Note that some of the other models, such as instance reweighting, are also not robust against this type of transformation. Furthermore the more features we capture, the lower the probability of having the same frequency over all features.

4. Discussion

Overall, the fir classifier competes well with the other methods; it has the lowest error rate for most of the combinations shown here. Furthermore, its' worst case performance is still the naive source classifier and it seems to perform equally well on image data as on text data, something that for instance scl seems to struggle with.

For both loss functions chosen here, the adapted classifer results in the original loss on the source samples plus a data-dependent regularization term. It is interesting to consider regularization as a method of incorporating prior knowledge. Of course, this is already done: if it is known that the solution is not extremely complex, ℓ_2 regularization is used, while if the solution is known to be sparse, $\ell_!$ regularization is used. However, since we are looking at very specific test sets it makes sense to construct a regularization term based on the domain differences.

Note that for independent domains, the transfer distribution does not exist. Successfully adapting a classifier is arguably not possible for these problems. We do not know in advance if two domains are independent, but one sufficient condition would be if if both domains lived in disjoint subspaces of the feature space. The spam dataset we presented here is an example where it turned out that the two domains, mail and sms, are most likely independent.

Our model is conservative in the sense that if it cannot capture the dependency between the domains, it reverts to the naive classifier. We have seen that there is always the danger of performing worse than a naive classifier for most natural datasets and therefore conservatism should be a valued property for this type of problem.

5. Conclusion

In this work we propose an approach to domain adaptation that can adapt to relatively less important features between domains. We show that the resulting classifier can converge to the target classifier and competes well with popular other dominant adaptation strategies.

Future work will focus on an exploration of which transfer distributions are possible, likely and useful. We also aim to extend the current method to continuous data.

Acknowledgments

This work was supported by the Netherlands Organization for Scientific Research (NWO; grant 612.001.301).

References

Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *The Journal of Machine Learning Research*, 6:1817–1853, 2005.

Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer vision–ECCV 2006*, pages 404–417. Springer, 2006.

Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007.

Shai Ben-David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *International Conference on Artificial Intelligence and Statistics*, pages 129–136, 2010.

Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128. Association for Computational Linguistics, 2006.

John Blitzer, Mark Dredze, Fernando Pereira, et al. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447, 2007.

Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

Ning Chen, Jun Zhu, Jianfei Chen, and Bo Zhang. Dropout training for support vector machines. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 1996.

Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

Jiayuan Huang, Alexander J Smola, Arthur Gretton, Karsten M Borgwardt, and Bernhard Schölkopf. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19:601, 2007.

Christopher J Leggetter and Philip C Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech & Language*, 9(2):171–185, 1995.

Justin Ma, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker. Identifying suspicious urls: an application of large-scale online learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 681–688. ACM, 2009.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Advances in neural information processing systems*, pages 1041–1048, 2009.

Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45 (1):521–530, 2012.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

Viswa Mani Kiran Peddinti and Prakriti Chintalapoodi. Domain adaptation in sentiment analysis of twitter. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

Afshin Rostamizadeh, Alekh Agarwal, and Peter L Bartlett. Learning with missing features. In *UAI*, pages 635–642. Citeseer, 2011.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Computer Vision–ECCV 2010*, pages 213–226. Springer, 2010.

Ming Shao, Dmitry Kit, and Yun Fu. Generalized transfer subspace learning through low-rank constraint. *International Journal of Computer Vision*, 109(1-2):74–93, 2014.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Laurens van der Maaten, Minmin Chen, Stephen Tyree, and Kilian Weinberger. Learning with marginalized corrupted features. In *Proceedings of the International Conference on Machine Learning*, 2013.

Annegreet van Opbroek, M Arfan Ikram, Meike W Vernooij, and Marleen de Bruijne. A transfer-learning approach to image segmentation across scanners by maximizing distribution similarity. In *Machine Learning in Medical Imaging*, pages 49–56. Springer, 2013.

Stefan Wager, Sida Wang, and Percy Liang. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems*, pages 351–359, 2013.