THE VARIATIONAL FAIR AUTOENCODER

Christos Louizos[*], **Kevin Swersky**[×], **Yujia Li**[×], **Max Welling**[*†‡], **Richard Zemel**[×†]
[*] Machine Learning Group, University of Amsterdam
[×] Department of Computer Science, University of Toronto
[†] Canadian Institute for Advanced Research (CIFAR)
[‡] University of California, Irvine
C.Louizos@uva.nl, {kswersky, yujiali}@cs.toronto.edu
M.Welling@uva.nl, zemel@cs.toronto.edu

ABSTRACT

We investigate the problem of learning representations that are invariant to certain nuisance or sensitive factors of variation in the data while retaining as much of the remaining information as possible. Our model is based on a variational autoencoding architecture (Kingma & Welling, 2014; Rezende et al., 2014) with priors that encourage independence between sensitive and latent factors of variation. Any subsequent processing, such as classification, can then be performed on this purged latent representation. To remove any remaining dependencies we incorporate an additional penalty term based on the "Maximum Mean Discrepancy" (MMD) (Gretton et al., 2006) measure. We discuss how these architectures can be efficiently trained on data and show in experiments that this method is more effective than previous work in removing unwanted sources of variation while maintaining informative latent representations.

1 INTRODUCTION

In "Representation Learning" one tries to find representations of the data that are informative for a particular task while removing the factors of variation that are uninformative and are typically detrimental for the task under consideration. Uninformative dimensions are often called "noise" or "nuisance variables" while informative dimensions are usually called latent or hidden factors of variation. Many machine learning algorithms can be understood in this way: principal component analysis, nonlinear dimensional reduction and latent Dirichlet allocation are all models that extract informative factors (dimensions, causes, topics) of the data which can often be used to visualize the data. On the other hand, linear discriminant analysis and deep (convolutional) neural nets learn representations that are good for classification.

In this paper we consider the case where we wish to learn latent representations where (almost) all of the information about certain known factors of variation are purged from the representation while still retaining as much information about the data as possible. In other words, we want a latent representation $\mathbf{z}$ that is maximally informative about an observed random variable $\mathbf{y}$ (e.g., class label) while minimally informative about a *sensitive* or *nuisance* variable $\mathbf{s}$. By treating $\mathbf{s}$ as a sensitive variable, i.e. $\mathbf{s}$ is correlated with our objective, we are dealing with "fair representations", a problem previously considered by Zemel et al. (2013). If we instead treat $\mathbf{s}$ as a nuisance variable we are dealing with "domain adaptation", in other words by removing the domain $\mathbf{s}$ from our representations we will obtain *improved* performance.

In this paper we introduce a novel model based on deep variational autoencoders (VAE) (Kingma & Welling, 2014; Rezende et al., 2014). These models can naturally encourage separation between latent variables $\mathbf{z}$ and sensitive variables $\mathbf{s}$ by using factorized priors $p(\mathbf{s})p(\mathbf{z})$. However, some dependencies may still remain when mapping data-cases to their hidden representation using the variational posterior $q(\mathbf{z}|\mathbf{x}, \mathbf{s})$, which we stamp out using a "Maximum Mean Discrepancy" (Gretton et al., 2006) term that penalizes differences between all order moments of the marginal posterior distributions $q(\mathbf{z}|\mathbf{s} = k)$ and $q(\mathbf{z}|\mathbf{s} = k')$ (for a discrete RV $\mathbf{s}$). In experiments we show that this combined approach is highly successful in learning representations that are devoid of unwanted information while retaining as much information as possible from what remains.

2 Learning Invariant Representations

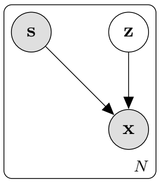


Figure 1: Unsupervised model

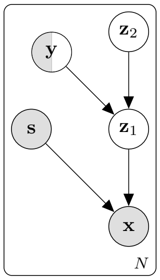


Figure 2: Semi-supervised model

2.1 Unsupervised model

Factoring out undesired variations from the data can be easily formulated as a general probabilistic model which admits two distinct (independent) "sources"; an observed variable $\mathbf{s}$, which denotes the variations that we want to remove, and a continuous latent variable $\mathbf{z}$ which models all the remaining information. This generative process can be formally defined as:

$$\mathbf{z} \sim p(\mathbf{z}); \qquad \mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z},\mathbf{s})$$

where $p_\theta(\mathbf{x}|\mathbf{z},\mathbf{s})$ is an appropriate probability distribution for the data we are modelling. With this formulation we explicitly encode a notion of 'invariance' in our model, since the latent representation is marginally independent of the factors of variation $\mathbf{s}$. Therefore the problem of finding an invariant representation for a data point $\mathbf{x}$ and variation $\mathbf{s}$ can be cast as performing inference on this graphical model and obtaining the posterior distribution of $\mathbf{z}$, $p(\mathbf{z}|\mathbf{x},\mathbf{s})$.

For our model we will employ a variational autoencoder architecture (Kingma & Welling, 2014; Rezende et al., 2014); namely we will parametrize the generative model (decoder) $p_\theta(\mathbf{x}|\mathbf{z},\mathbf{s})$ and the variational posterior (encoder) $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{s})$ as (deep) neural networks which accept as inputs $\mathbf{z},\mathbf{s}$ and $\mathbf{x},\mathbf{s}$ respectively and produce the parameters of each distribution after a series of non-linear transformations. Both the model (θ) and variational (ϕ) parameters will be jointly optimized with the SGVB (Kingma & Welling, 2014) algorithm according to a lower bound on the log-likelihood. This parametrization will allow us to capture most of the salient information of $\mathbf{x}$ in our embedding $\mathbf{z}$. Furthermore the distributed representation of a neural network would allow us to better resolve the dependencies between $\mathbf{x}$ and $\mathbf{s}$ thus yielding a better disentangling between the independent factors $\mathbf{z}$ and $\mathbf{s}$. By choosing a Gaussian posterior $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{s})$ and standard isotropic Gaussian prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{0},\mathbf{I})$ we can obtain the following lower bound:

$$\begin{aligned}\sum_{n=1}^{N} \log p(\mathbf{x}_n|\mathbf{s}_n) &\geq \sum_{n=1}^{N} \mathbb{E}_{q_\phi(\mathbf{z}_n|\mathbf{x}_n,\mathbf{s}_n)}[\log p_\theta(\mathbf{x}_n|\mathbf{z}_n,\mathbf{s}_n)] - KL(q_\phi(\mathbf{z}_n|\mathbf{x}_n,\mathbf{s}_n)||p(\mathbf{z})) \qquad (1)\\ &= \mathcal{F}(\phi,\theta;\mathbf{x}_n,\mathbf{s}_n)\end{aligned}$$

with $q_\phi(\mathbf{z}_n|\mathbf{x}_n,\mathbf{s}_n) = \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}_n = f_\phi(\mathbf{x}_n,\mathbf{s}_n), \boldsymbol{\sigma}_n = e^{f_\phi(\mathbf{x}_n,\mathbf{s}_n)})$ and $p_\theta(\mathbf{x}_n|\mathbf{z}_n,\mathbf{s}_n) = f_\theta(\mathbf{z}_n,\mathbf{s}_n)$ with $f_\theta(\mathbf{z}_n,\mathbf{s}_n)$ being an appropriate probability distribution for the data we are modelling.

2.2 Semi-Supervised model

Factoring out variations in an unsupervised way can however be harmful in cases where we want to use this invariant representation for a subsequent prediction task. In particular if we have a situation where the nuisance variable $\mathbf{s}$ and the actual label $\mathbf{y}$ are correlated, then training an unsupervised model could yield *random* or *degenerate* representations with respect to $\mathbf{y}$. Therefore it is more appropriate to try to "inject" the information about the label during the feature extraction phase. This can be quite simply achieved by introducing a second "layer" of latent variables to our generative model where we try to correlate $\mathbf{z}$ with the prediction task. Assuming that the invariant features are now called $\mathbf{z}_1$ we enrich the generative story by similarly providing two distinct (independent) sources for $\mathbf{z}_1$; a discrete (in case of classification)variable $\mathbf{y}$ which denotes the label of the data point $\mathbf{x}$ and a continuous latent variable $\mathbf{z}_2$ which encodes the variation on $\mathbf{z}_1$ that is not explained by $\mathbf{y}$ ($\mathbf{x}$ dependent noise). The process now can be formally defined as:

$$\mathbf{y}, \mathbf{z}_2 \sim \text{Cat}(\mathbf{y})p(\mathbf{z}_2); \qquad \mathbf{z}_1 \sim p_\theta(\mathbf{z}_1|\mathbf{z}_2, \mathbf{y}); \qquad \mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z}_1, \mathbf{s})$$

Similarly to the unsupervised case we use a variational auto-encoder and jointly optimize the variational and model parameters. The lower bound now becomes:

$$\sum_{n=1}^{N} \log p(\mathbf{x}_n|\mathbf{s}_n) \geq \sum_{n=1}^{N} \mathbb{E}_{q_\phi(\mathbf{z}_{1n}, \mathbf{z}_{2n}, \mathbf{y}_n|\mathbf{x}_n, \mathbf{s}_n)}[\log p(\mathbf{z}_2) + \log p(\mathbf{y}_n) + \log p_\theta(\mathbf{z}_{1n}|\mathbf{z}_{2n}, \mathbf{y}_n) + \\ + \log p_\theta(\mathbf{x}_n|\mathbf{z}_{1n}, \mathbf{s}_n) - \log q_\phi(\mathbf{z}_{1n}, \mathbf{z}_{2n}, \mathbf{y}_n|\mathbf{x}_n, \mathbf{s}_n)] \tag{2}$$

where we assume that the posterior $q_\phi(\mathbf{z}_{1n}, \mathbf{z}_{2n}, \mathbf{y}_n|\mathbf{x}_n, \mathbf{s}_n)$ is factorized as $q_\phi(\mathbf{z}_{1n}, \mathbf{z}_{2n}, \mathbf{y}_n|\mathbf{x}_n, \mathbf{s}_n) = q_\phi(\mathbf{z}_{1n}|\mathbf{x}_n, \mathbf{s}_n)q_\phi(\mathbf{y}_n|\mathbf{z}_{1n})q_\phi(\mathbf{z}_{2n}|\mathbf{z}_{1n}, \mathbf{y}_n)$, and where:

$$\begin{aligned}
q_\phi(\mathbf{z}_{1n}|\mathbf{x}_n, \mathbf{s}_n) &= \mathcal{N}(\mathbf{z}_{1n}|\boldsymbol{\mu}_n = f_\phi(\mathbf{x}_n, \mathbf{s}_n), \boldsymbol{\sigma}_n = e^{f_\phi(\mathbf{x}_n, \mathbf{s}_n)}) \\
q_\phi(\mathbf{y}_n|\mathbf{z}_{1n}) &= \text{Cat}(\mathbf{y}_n|\boldsymbol{\pi}_n = \text{softmax}(f_\phi(\mathbf{z}_{1n}))) \\
q_\phi(\mathbf{z}_{2n}|\mathbf{z}_{1n}, \mathbf{y}_n) &= \mathcal{N}(\mathbf{z}_{2n}|\boldsymbol{\mu}_n = f_\phi(\mathbf{z}_{1n}, \mathbf{y}_n), \boldsymbol{\sigma}_n = e^{f_\phi(\mathbf{z}_{1n}, \mathbf{y}_n)}) \\
p_\theta(\mathbf{z}_{1n}|\mathbf{z}_{2n}, \mathbf{y}_n) &= \mathcal{N}(\mathbf{z}_{1n}|\boldsymbol{\mu}_n = f_\theta(\mathbf{z}_{2n}, \mathbf{y}_n), \boldsymbol{\sigma}_n = e^{f_\theta(\mathbf{z}_{2n}, \mathbf{y}_n)}) \\
p_\theta(\mathbf{x}_n|\mathbf{z}_{1n}, \mathbf{s}_n) &= f_\theta(\mathbf{z}_{1n}, \mathbf{s}_n)
\end{aligned}$$

with $f_\theta(\mathbf{z}_{1n}, \mathbf{s}_n)$ again being an appropriate probability distribution for the data we are modelling. The model proposed here can be seen as an extension to the ‘stacked M1+M2’ model originally proposed from Kingma et al. (2014), where we have additionally introduced the nuisance variable $\mathbf{s}$ during the feature extraction. Thus following Kingma et al. (2014) we can also handle the ‘semi-supervised’ case, i.e., missing labels. In situations where the label is observed the lower bound takes the following form (exploiting the fact that we can compute some Kullback-Leibler divergences explicitly in our case):

$$\sum_{n=1}^{N} \mathcal{L}_s(\phi, \theta; \mathbf{x}_n, \mathbf{s}_n, \mathbf{y}_n) = \sum_{n=1}^{N_s} \mathbb{E}_{q_\phi(\mathbf{z}_{1n}|\mathbf{x}_n, \mathbf{s}_n)}[-KL(q_\phi(\mathbf{z}_{2n}|\mathbf{z}_{1n}, \mathbf{y}_n)||p(\mathbf{z}_2)) + \log p_\theta(\mathbf{x}_n|\mathbf{z}_{1n}, \mathbf{s}_n)] + \\ + \mathbb{E}_{q_\phi(\mathbf{z}_{1n}|\mathbf{x}_n, \mathbf{s}_n)q_\phi(\mathbf{z}_{2n}|\mathbf{z}_{1n}, \mathbf{y}_n)}[\log p_\theta(\mathbf{z}_1|\mathbf{z}_{2n}, \mathbf{y}_n) - \log q_\phi(\mathbf{z}_{1n}|\mathbf{x}_n\mathbf{s}_n)] \tag{3}$$

and in the case that it is not observed we use $q(\mathbf{y}_n|\mathbf{z}_{1n})$ to ‘impute’ our data:

$$\sum_{m=1}^{M} \mathcal{L}_u(\phi, \theta; \mathbf{x}_m, \mathbf{s}_m) = \sum_{m=1}^{M} \mathbb{E}_{q_\phi(\mathbf{z}_{1m}|\mathbf{x}_m, \mathbf{s}_m)}[-KL(q(\mathbf{y}_m|\mathbf{z}_{1m})||p(\mathbf{y}_m)) + \log p_\theta(\mathbf{x}_m|\mathbf{z}_{1m}, \mathbf{s}_m)] + \\ + \mathbb{E}_{q_\phi(\mathbf{z}_{1m}, \mathbf{y}_m|\mathbf{x}_m, \mathbf{s}_m)}[-KL(q_\phi(\mathbf{z}_{2m}|\mathbf{z}_{1m}, \mathbf{y}_m)||p(\mathbf{z}_2))] + \\ + \mathbb{E}_{q_\phi(\mathbf{z}_{1m}, \mathbf{y}_m, \mathbf{z}_{2m}|\mathbf{x}_m, \mathbf{s}_m)}[\log p_\theta(\mathbf{z}_{1m}|\mathbf{z}_{2m}, \mathbf{y}_m) - \log q_\phi(\mathbf{z}_{1m}|\mathbf{x}_m, \mathbf{s}_m)] \tag{4}$$

therefore the final objective function is:

$$\mathcal{F}_{\text{VAE}}(\phi, \theta; \mathbf{x}_n, \mathbf{x}_m, \mathbf{s}_n, \mathbf{s}_m, \mathbf{y}_n) = \sum_{n=1}^{N} \mathcal{L}_s(\phi, \theta; \mathbf{x}_n, \mathbf{s}_n, \mathbf{y}_n) + \sum_{m=1}^{M} \mathcal{L}_u(\phi, \theta; \mathbf{x}_m, \mathbf{s}_m) + \\ + \alpha \sum_{n=1}^{N} \mathbb{E}_{q(\mathbf{z}_{1n}|\mathbf{x}_n, \mathbf{s}_n)}[-\log q_\phi(\mathbf{y}_n|\mathbf{z}_{1n})] \tag{5}$$

where the last term is introduced so as to ensure that the predictive posterior $q_\phi(\mathbf{y}|\mathbf{z}_1)$ learns from both labeled and unlabeled data. This semi-supervised model will be called “VAE” in our experiments.

However, there is a subtle difference between the approach of Kingma et al. (2014) and our model. Instead of training separately each layer of stochastic variables we optimize the model jointly. The potential advantages of this approach are two fold: as we previously mentioned if the label $\mathbf{y}$ and the nuisance information $\mathbf{s}$ are correlated then training a (conditional) feature extractor separately poses the danger of creating a degenerate representation with respect to the label $\mathbf{y}$. Furthermore the label information will also better guide the feature extraction towards the more salient parts of the data, thus maintaining most of the (predictive) information.

2.3 Further invariance via Maximum Mean Discrepancy

Despite the fact that we have a model that encourages statistical independence between $\mathbf{s}$ and $\mathbf{z}_1$ a-priori we might still have some dependence in the (approximate) marginal posterior $q_\phi(\mathbf{z}_1|\mathbf{s})$. In particular, this can happen if the label $\mathbf{y}$ is correlated with the sensitive variable $\mathbf{s}$, which can allow information about $\mathbf{s}$ to "leak" into the posterior. Thus instead we could maximize a "penalized" lower bound where we impose some sort of regularization on the marginal $q_\phi(\mathbf{z}_1|\mathbf{s})$. In the following we will describe one way to achieve this regularization through the Maximum Mean Discrepancy (MMD) (Gretton et al., 2006) measure.

2.3.1 Maximum Mean Discrepancy

Consider the problem of determining whether two datasets $\{\mathbf{X}\} \sim P_0$ and $\{\mathbf{X}'\} \sim P_1$ are drawn from the same distribution, i.e., $P_0 = P_1$. A simple test is to consider the distance between empirical statistics $\psi(\cdot)$ of the two datasets:

$$\left\| \frac{1}{N_0} \sum_{i=1}^{N_0} \psi(\mathbf{x}_i) - \frac{1}{N_1} \sum_{i=1}^{N_1} \psi(\mathbf{x}'_i) \right\|^2 . \tag{6}$$

Expanding the square yields an estimator composed only of inner products on which the kernel trick can be applied. The resulting estimator is known as Maximum Mean Discrepancy (MMD) (Gretton et al., 2006):

$$\ell_{\text{MMD}}(\mathbf{X}, \mathbf{X}') = \frac{1}{N_0^2} \sum_{n=1}^{N_0} \sum_{m=1}^{N_0} k(\mathbf{x}_n, \mathbf{x}_m) + \frac{1}{N_1^2} \sum_{n=1}^{N_1} \sum_{m=1}^{N_1} k(\mathbf{x}'_n, \mathbf{x}'_m) - \frac{2}{N_0 N_1} \sum_{n=1}^{N_0} \sum_{m=1}^{N_1} k(\mathbf{x}_n, \mathbf{x}'_m). \tag{7}$$

Asymptotically, for a universal kernel such as the Gaussian kernel $k(x, x') = e^{-\gamma\|\mathbf{x}-\mathbf{x}'\|^2}$, $\ell_{\text{MMD}}(\mathbf{X}, \mathbf{X}')$ is 0 if and only if $P_0 = P_1$. Equivalently, minimizing MMD can be viewed as matching all of the moments of P_0 and P_1. Therefore, we can use it as an extra "regularizer" and force the model to try to match the moments between the marginal posterior distributions of our latent variables, i.e., $q_\phi(\mathbf{z}_1|s = 0)$ and $q_\phi(\mathbf{z}_1|s = 1)$ (in the case of binary nuisance information $\mathbf{s}$[1]). By adding the MMD penalty into the lower bound of our aforementioned VAE architecture we obtain our proposed model, the "Variational Fair Autoencoder" (VFAE):

$$\mathcal{F}_{\text{VFAE}}(\phi, \theta; \mathbf{x}_n, \mathbf{x}_m, \mathbf{s}_n, \mathbf{s}_m, \mathbf{y}_n) = \mathcal{F}_{\text{VAE}}(\phi, \theta; \mathbf{x}_n, \mathbf{x}_m, \mathbf{s}_n, \mathbf{s}_m, \mathbf{y}_n) - \beta \ell_{\text{MMD}}(\mathbf{Z}_{1\mathbf{s}=0}, \mathbf{Z}_{1\mathbf{s}=1}) \tag{8}$$

where:

$$\ell_{\text{MMD}}(\mathbf{Z}_{1\mathbf{s}=0}, \mathbf{Z}_{1\mathbf{s}=1}) = \| \mathbb{E}_{\tilde{p}(\mathbf{x}|\mathbf{s}=0)}[\mathbb{E}_{q(\mathbf{z}_1|\mathbf{x},\mathbf{s}=0)}[\psi(\mathbf{z}_1)]] - E_{\tilde{p}(\mathbf{x}|\mathbf{s}=1)}[\mathbb{E}_{q(\mathbf{z}_1|\mathbf{x},\mathbf{s}=1)}[\psi(\mathbf{z}_1)]]\|^2 \tag{9}$$

2.4 Fast MMD via Random Fourier Features

A naive implementation of MMD in minibatch stochastic gradient descent would require computing the $M \times M$ Gram matrix for each minibatch during training, where M is the minibatch size. Instead, we can use random kitchen sinks (Rahimi & Recht, 2009) to compute a feature expansion such that computing the estimator (6) approximates the full MMD (7). To compute this, we draw a random $K \times D$ matrix $\mathbf{W}$, where K is the dimensionality of $\mathbf{x}$, D is the number of random features and each entry of $\mathbf{W}$ is drawn from a standard isotropic Gaussian. The feature expansion is then given as:

$$\psi_{\mathbf{W}}(\mathbf{x}) = \sqrt{\frac{2}{D}} \cos\left(\sqrt{\frac{2}{\gamma}} \mathbf{x}\mathbf{W} + \mathbf{b} \right). \tag{10}$$

where $\mathbf{b}$ is a D-dimensional uniform random vector with entries in $[0, 2\pi]$. Zhao & Meng (2015) have successfully applied the idea of using random kitchen sinks to approximate MMD. This estimator is fairly accurate, and is typically much faster than the full MMD penalty. We use $D = 500$ in our experiments.

[1] In case that we have more than two states for the nuisance information $\mathbf{s}$, we minimize the MMD penalty between each marginal $q(\mathbf{z}|\mathbf{s} = k)$ and $q(\mathbf{z})$, i.e., $\sum_{k=1}^{K} \ell_{\text{MMD}}(\mathbf{Z}_{1\mathbf{s}=k}, \mathbf{Z}_1)$ for all possible states K of $\mathbf{s}$.

3 Experiments

We performed experiments on the three datasets that correspond to a "fair" classification scenario and were previously used by Zemel et al. (2013). In these datasets the "nuisance" or sensitive variable $\mathbf{s}$ is significantly correlated with the label $\mathbf{y}$ thus making the proper removal of $\mathbf{s}$ challenging. Furthermore, we also experimented with the Amazon reviews dataset to make a connection with the "domain-adaptation" literature. Finally, we also experimented with a more general task on the extended Yale B dataset; that of learning invariant representations.

3.1 Datasets

For the fairness task we experimented with three datasets that were previously used by Zemel et al. (2013). The German dataset is the smallest one with 1000 data points and the objective is to predict whether a person has a good or bad credit rating. The sensitive variable is the gender of the individual. The Adult income dataset contains $45,222$ entries and describes whether an account holder has over \$$50,000$ dollars in their account. The sensitive variable is age. Both of these are obtained from the UCI machine learning repository (Frank & Asuncion, 2010). The health dataset is derived from the Heritage Health Prize[2]. It is the largest of the three datasets with $147,473$ entries. The task is to predict whether a patient will spend any days in the hospital in the next year and the sensitive variable is the age of the individual. We use the same train/test/validation splits as Zemel et al. (2013) for our experiments. Finally we also binarized the data and used a multivariate Bernoulli distribution for $p_\theta(\mathbf{x}_n|\mathbf{z}_{1n}, \mathbf{s}_n) = \text{Bern}(\mathbf{x}_n|\boldsymbol{\pi}_n = \sigma(f_\theta(\mathbf{z}_{1n}, \mathbf{s}_n)))$, where $\sigma(\cdot)$ is the sigmoid function [3].

For the domain adaptation task we used the Amazon reviews dataset (with similar preprocessing) that was also employed by Chen et al. (2012) and Ganin et al. (2015). It is composed from text reviews about particular products, where each product belongs to one out of four different domains: "books", "dvd", "electronics" and "kitchen". As a result we performed twelve domain adaptation tasks. The labels $\mathbf{y}$ correspond to the sentiment of each review, i.e. either positive or negative. Since each feature vector $\mathbf{x}$ is composed from counts of unigrams and bigrams we used a Poisson distribution for $p_\theta(\mathbf{x}_n|\mathbf{z}_{1n}, \mathbf{s}_n) = \text{Poisson}(\mathbf{x}_n|\boldsymbol{\lambda}_n = e^{f_\theta(\mathbf{z}_{1n}, \mathbf{s}_n)})$. It is also worthwhile to mention that we can fully exploit the semi-supervised nature of our model in this dataset, and thus for training we only use the source domain labels and consider the labels of the target domain as "missing".

For the general task of learning invariant representations we used the Extended Yale B dataset, which was also employed in a similar fashion by Li et al. (2014). It is composed from face images of 38 people under different lighting conditions (directions of the light source). Similarly to Li et al. (2014), we created 5 states for the nuisance variable $\mathbf{s}$: light source in upper right, lower right, lower left, upper left and the front. The labels $\mathbf{y}$ correspond to the identity of the person. Following Li et al. (2014), we used the same training, test set and no validation set. For the $p(\mathbf{x}_n|\mathbf{z}_{1n}, \mathbf{s}_n)$ distribution we used a Gaussian with means constrained in the 0-1 range (since we have intensity images) by a sigmoid, i.e. $p_\theta(\mathbf{x}_n|\mathbf{z}_{1n}, \mathbf{s}_n) = \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_n = \sigma(f_\theta(\mathbf{z}_{1n}, \mathbf{s}_n)), \boldsymbol{\sigma}_n = e^{f_\theta(\mathbf{z}_{1n}, \mathbf{s}_n)})$.

3.2 Experimental Setup

For the Adult dataset both encoders, for $\mathbf{z}_1$ and $\mathbf{z}_2$, and both decoders, for $\mathbf{z}_1$ and $\mathbf{x}$, had one hidden layer of 100 units. For the Health dataset we had one hidden layer of 300 units for the $\mathbf{z}_1$ encoder and $\mathbf{x}$ decoder and one hidden layer of 150 units for the $\mathbf{z}_2$ encoder and $\mathbf{z}_1$ decoder. For the much smaller German dataset we used 60 hidden units for both encoders and decoders. Finally, for the Amazon reviews and Extended Yale B datasets we had one hidden layer with 500, 400 units respectively for the $\mathbf{z}_1$ encoder, $\mathbf{x}$ decoder, and 300, 100 units respectively for the $\mathbf{z}_2$ encoder and $\mathbf{z}_1$ decoder. On all of the datasets we used 50 latent dimensions for $\mathbf{z}_1$ and $\mathbf{z}_2$, except for the small German dataset, where we used 30 latent dimensions for both variables. For the predictive posterior $q_\phi(\mathbf{y}|\mathbf{z}_1)$ we used a simple Logistic regression classifier. Optimization of the objective function was done with Adam (Kingma & Ba, 2015) using the default values for the hyperparameters, minibatches of 100 data points and temporal averaging. The MMD penalty was simply multiplied by the minibatch size so as to keep the scale of the penalty similar to the lower bound. Furthermore, the extra strength

[2] www.heritagehealthprize.com

[3] $\sigma(t) = \frac{1}{1+e^{-t}}$

of the MMD, β, was tuned according to a validation set. The scaling of the supervised cost was low ($\alpha = 1$) for the Adult, Health and German datasets due to the correlation of $\mathbf{s}$ with $\mathbf{y}$. On the Amazon reviews and Extended Yale B datasets however the scaling of the supervised cost was higher: $\alpha = 100 \cdot \frac{N_{\text{batch\_source}} + N_{\text{batch\_target}}}{N_{\text{batch\_source}}}$ for the Amazon reviews dataset (empirically determined after observing the classification loss on the first few iterations on the first source-target pair) and $\alpha = 200$ for the Extended Yale B dataset. Similarly, the scaling of the MMD penalty was $\beta = 100 \cdot N_{\text{batch}}$ for the Amazon reviews dataset and $\beta = 200 \cdot N_{\text{batch}}$ for the Extended Yale B.

Our evaluation is geared towards two fronts; removing information about $\mathbf{s}$ and classification accuracy for $\mathbf{y}$. To measure the information about $\mathbf{s}$ in our new representation we simply train a classifier to predict $\mathbf{s}$ from $\mathbf{z}_1$. We utilize both Logistic Regression (LR) which is a simple linear classifier, and Random Forest (RF) which is a powerful non-linear classifier. Since on the datasets that we experimented with the nuisance variable $\mathbf{s}$ is binary we can easily find the random chance accuracy for $\mathbf{s}$ and measure the discriminatory information of $\mathbf{s}$ in $\mathbf{z}_1$. Furthermore, we also used the discrimination metric from Zemel et al. (2013) as well a more "informed" version of the discrimination metric that instead of the predictions, takes into account the probabilities of the correct class. They are provided in the appendix A. Finally, for the classification performance on $\mathbf{y}$ we used the predictive posterior $q_\phi(\mathbf{y}|\mathbf{z}_1)$ for the VAE/VFAE and a simple Logistic Regression for the original representations $\mathbf{x}$. It should be noted that for the VFAE and VAE models we use a sample from $q_\phi(\mathbf{z}_1|\mathbf{x}, \mathbf{s})$ to make predictions, instead of using the mean. We found that the extra noise helps with invariance.

We implemented the Learning Fair Representations (Zemel et al., 2013) method (LFR) as a baseline using $K = 50$ dimensions for the latent space. To measure the accuracy on $\mathbf{y}$ in the results below we similarly used the LFR model predictions.

3.3 Results

3.3.1 Fair classification

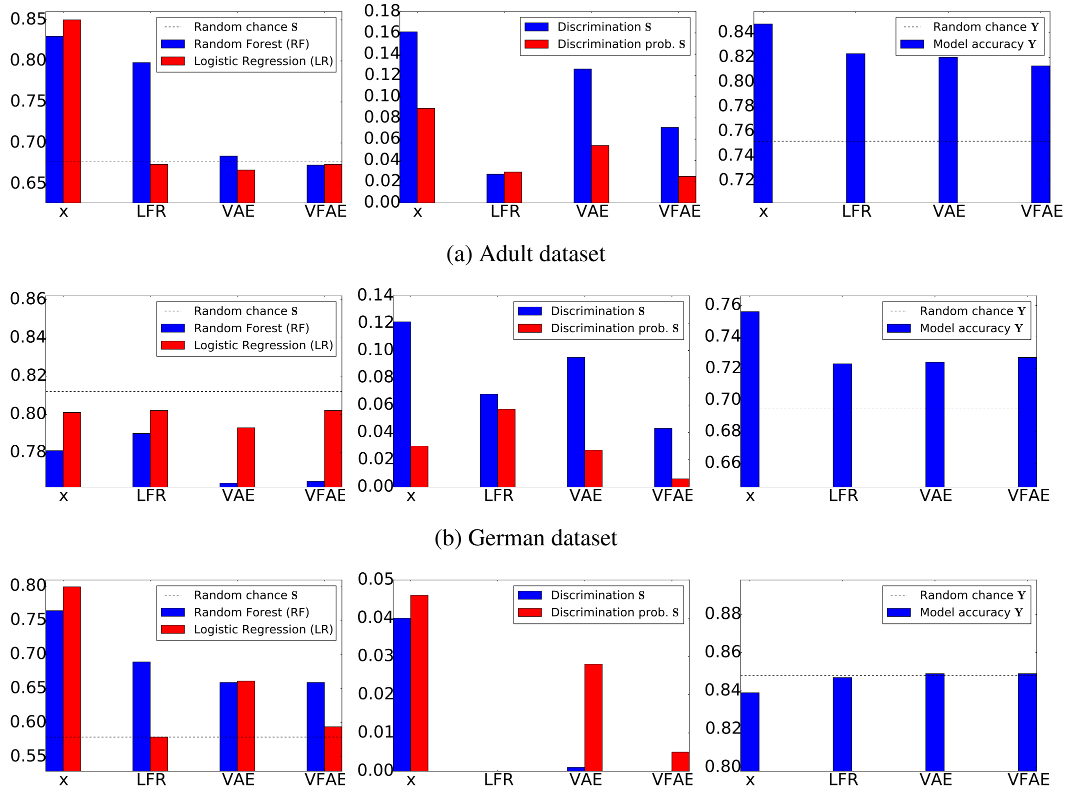
The results for all three datasets can be seen in Figure 3. Since we are dealing with the "fair" classification scenario here, low accuracy and discrimination against $\mathbf{s}$ is more important than the accuracy on $\mathbf{y}$ (as long as we do not produce degenerate representations).

On the Adult dataset, the highest accuracy on the label $\mathbf{y}$ and the lowest discrimination against $\mathbf{s}$ is obtained by our LFR baseline. Despite the fact that LFR appears to give the best tradeoff between accuracy and discrimination, it appears to retain information about $\mathbf{s}$ in its representation, which is discovered from the random forest classifier. In that sense, the VFAE method appears to do the best job in actually removing the sensitive information and maintaining most of the predictive information. Furthermore, the introduction of the MMD penalty in the VFAE model seems to provide a significant benefit with respect to our discrimination metrics, as both were reduced considerably compared to the regular VAE.

On the German dataset, all methods appear to be invariant with respect to the sensitive information $\mathbf{s}$. However this is not the case for the discrimination metric, since LFR does appear to retain information compared to the VAE and VFAE. The MMD penalty in VFAE did seem improve the discrimination scores over the original VAE, while the accuracy on the labels $\mathbf{y}$ remained similar.

As for the Health dataset; this dataset is extremely imbalanced, with only 15% of the patients being admitted to a hospital. Therefore, each of the classifiers seems to predict the majority class as the label $\mathbf{y}$ for every point. For the invariance against $\mathbf{s}$ however, the results were more interesting. On the one hand, the VAE model on this dataset did maintain some sensitive information, which could be identified both linearly and non-linearly. On the other hand, VFAE and the LFR methods were able to retain less information in their latent representation, since only Random Forest was able to achieve higher than random chance accuracy. This further justifies our choice for including the MMD penalty in the lower bound of the VAE. .

In order to further assess the nature of our new representations, we visualized two dimensional Barnes-Hut SNE (van der Maaten, 2013) embeddings of the $\mathbf{z}_1$ representations, obtained from the model trained on the Adult dataset, in Figure 4. As we can see, the nuisance/sensitive variables $\mathbf{s}$ can be identified both on the original representation $\mathbf{x}$ and on a latent representation $\mathbf{z}_1$ that does not have the MMD penalty and the independence properties between $\mathbf{z}_1$ and $\mathbf{s}$ in the prior. By



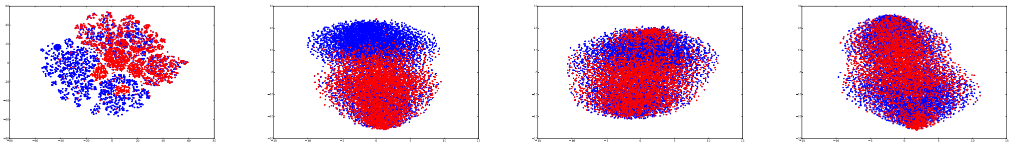
(a) Adult dataset

(b) German dataset

(c) Health dataset

Figure 3: Fair classification results. Columns correspond to each evaluation scenario (in order): Random/RF/LR accuracy on $\mathbf{s}$, Discrimination/Discrimination prob. against $\mathbf{s}$ and Random/Model accuracy on $\mathbf{y}$. Note that the objective of a “fair” encoding is to have low accuracy on S (where LR is a linear classifier and RF is nonlinear), low discrimination against S and high accuracy on Y.

introducing these independence properties as well as the MMD penalty the nuisance variable groups become practically indistinguishable.



(a)

(b)

(c)

(d)

Figure 4: t-SNE (van der Maaten, 2013) visualizations from the Adult dataset on: (a): original $\mathbf{x}$, (b): latent $\mathbf{z}_1$ without $\mathbf{s}$ and MMD, (c): latent $\mathbf{z}_1$ with $\mathbf{s}$ and without MMD, (d): latent $\mathbf{z}_1$ with $\mathbf{s}$ and MMD. Blue colour corresponds to males whereas red colour corresponds to females.

3.3.2 Domain adaptation

As for the domain adaptation scenario and the Amazon reviews dataset, the results of our VFAE model can be seen in Table 1. Our model was successful in factoring out the domain information, since the accuracy, measured both linearly (LR) and non-linearly (RF), was towards random chance (which for this dataset is 0.5). We should also mention that, on this dataset at least, completely removing information about the domain does not guarantee a better performance on $\mathbf{y}$. The same effect was also observed by Ganin et al. (2015) and Chen et al. (2012). As far as the accuracy on $\mathbf{y}$

is concerned, we compared against a recent neural network based state of the art method for domain adaptation, Domain Adversarial Neural Network (DANN) (Ganin et al., 2015). As we can observe in table 1, our accuracy on the labels $\mathbf{y}$ is higher on 9 out of the 12 domain adaptation tasks whereas on the remaining 3 it is quite similar to the DANN architecture.

Table 1: Results on the Amazon reviews dataset. The DANN column is taken directly from Ganin et al. (2015) (the column that uses the original representation as input).

Source - Target	S		Y	
	RF	LR	VFAE	DANN
books - dvd	0.535	0.564	**0.799**	0.784
books - electronics	0.541	0.562	**0.792**	0.733
books - kitchen	0.537	0.583	**0.816**	0.779
dvd - books	0.537	0.563	**0.755**	0.723
dvd - electronics	0.538	0.566	**0.786**	0.754
dvd - kitchen	0.543	0.589	**0.822**	0.783
electronics - books	0.562	0.590	**0.727**	0.713
electronics - dvd	0.556	0.586	**0.765**	0.738
electronics - kitchen	0.536	0.570	0.850	**0.854**
kitchen - books	0.560	0.593	**0.720**	0.709
kitchen - dvd	0.561	0.599	0.733	**0.740**
kitchen - electronics	0.533	0.565	0.838	**0.843**

3.4 Learning Invariant Representations

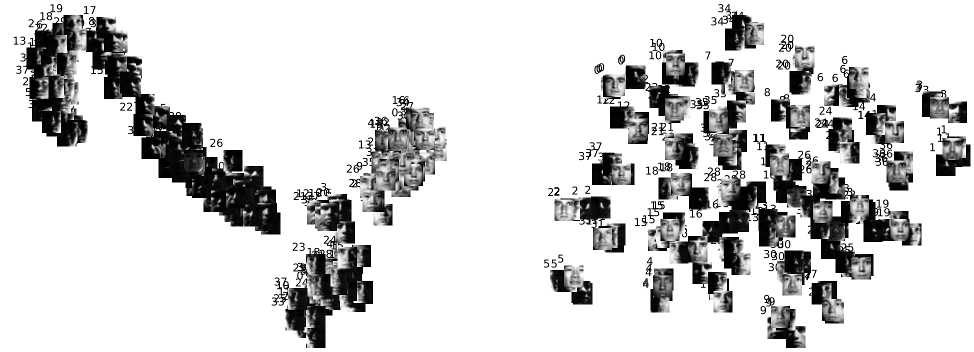
Regarding the more general task of learning invariant representations; our results on the Extended Yale B dataset also demonstrate our model's ability to learn such representations. As expected, on the original representation $\mathbf{x}$ the lighting conditions, $\mathbf{s}$, are well identifiable with almost perfect accuracy from both RF and LR. This can also be seen in the two dimensional embeddings of the original space $\mathbf{x}$ in Figure 5a: the images are mostly clustered according to the lighting conditions. As soon as we utilize our VFAE model we simultaneously decrease the accuracy on $\mathbf{s}$, from 96% to about 50%, and increase our accuracy on $\mathbf{y}$, from 78% to about 85%. This effect can also be seen in Figure 5b: the images are now mostly clustered according to the person ID (the label $\mathbf{y}$). It is clear that in this scenario the information about $\mathbf{s}$ is purely "nuisance" with respect to the labels $\mathbf{y}$. Therefore, by using our VFAE model we are able to obtain improved generalization and classification performance by effectively removing $\mathbf{s}$ from our representations.

Table 2: Results on the Extended Yale B dataset. We also included the best result from Li et al. (2014) under the NN + MMD row.

Method	S		Y
	RF	LR	
Original $\mathbf{x}$	0.952	0.961	0.78
NN + MMD	-	-	0.82
VFAE	0.435	0.565	**0.846**

4 Related Work

Most related to our "fair" representations view is the work from Zemel et al. (2013). They proposed a neural network based semi-supervised clustering model for learning fair representations. The idea is to learn a localised representation that maps each datapoint to a cluster in such a way that each cluster gets assigned roughly equal proportions of data from each group in s. Although their approach was successfully applied on several datasets, the restriction to clustering means that it cannot leverage the representational power of a distributed representation. Furthermore, this penalty does not account for higher order moments in the latent distribution. For example, if $p(z_k = 1|x_i, s = 0)$ always



(a)

(b)

Figure 5: t-SNE (van der Maaten, 2013) visualizations of the Extended Yale B training set. (a): original $\mathbf{x}$, (b): latent $\mathbf{z}_1$ from VFAE. Each example is plotted with the person ID and the image. Zoom in to see details.

returns 1 or 0, while $p(z_k = 1|x_i, s = 1)$ returns values between values 0 and 1, then the penalty could still be satisfied, but information could still leak through. We addressed both of these issues in this paper.

Domain adaptation can also be cast as learning representations that are "invariant" with respect to a discrete variable $\mathbf{s}$, the domain. Most similar to our work are neural network approaches which try to match the feature distributions between the domains. This was performed in an unsupervised way with mSDA (Chen et al., 2012) by training denoising autoencoders jointly on all domains, thus implicitly obtaining a representation general enough to explain both the domain and the data. This is in contrast to our approach where we instead try to learn representations that explicitly remove domain information during the learning process. For the latter we find more similarities with "domain-regularized" supervised approaches that simultaneously try to predict the label for a data point and remove domain specific information. This is done with either MMD (Long & Wang, 2015; Tzeng et al., 2014) or adversarial (Ganin et al., 2015) penalties at the hidden layers of the network. In our model however the main "domain-regularizer" stems from the independence properties of the prior over the domain and latent representations. We also employ MMD on our model but from a different perspective since we consider a slightly more difficult case where the domain $\mathbf{s}$ and label $\mathbf{y}$ are correlated; we need to ensure that we remain as "invariant" as possible since $q_\phi(\mathbf{y}|\mathbf{z}_1)$ might 'leak' information about $\mathbf{s}$.

5 Conclusion

We introduce the Variational Fair Autoencoder (VFAE), an extension of the semi-supervised variational autoencoder in order to learn representations that are explicitly invariant with respect to some known aspect of a dataset while retaining as much remaining information as possible. We further use a Maximum Mean Discrepancy regularizer in order to further promote invariance in the posterior distribution over latent variables. We apply this model to tasks involving developing fair classifiers that are invariant to sensitive demographic information and show that it produces a better tradeoff with respect to accuracy and invariance. As a second application, we consider the task of domain adaptation, where the goal is to improve classification by training a classifier that is invariant to the domain. We find that our model is competitive with recently proposed adversarial approaches. Finally, we also consider the more general task of learning invariant representations. We can observe that our model provides a clear improvement against a neural network that incorporates a Maximum Mean Discrepancy penalty.

References

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

Ben-David, Shai, Blitzer, John, Crammer, Koby, Kulesza, Alex, Pereira, Fernando, and Vaughan, Jennifer Wortman. A theory of learning from different domains. *Machine learning*, 79(1-2): 151–175, 2010.

Chen, Minmin, Xu, Zhixiang, Weinberger, Kilian, and Sha, Fei. Marginalized denoising autoencoders for domain adaptation. *International Conference on Machine Learning (ICML)*, 2012.

Frank, A. and Asuncion, A. UCI machine learning repository, 2010. URL `http://archive.ics.uci.edu/ml`.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-Adversarial Training of Neural Networks. *ArXiv e-prints*, May 2015.

Gretton, Arthur, Borgwardt, Karsten M, Rasch, Malte, Schölkopf, Bernhard, and Smola, Alex J. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pp. 513–520, 2006.

Kifer, Daniel, Ben-David, Shai, and Gehrke, Johannes. Detecting change in data streams. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pp. 180–191. VLDB Endowment, 2004.

Kingma, Diederik P and Ba, Jimmy. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

Kingma, Diederik P and Welling, Max. Auto-encoding variational bayes. *International Conference on Learning Representations (ICLR)*, 2014.

Kingma, Diederik P, Mohamed, Shakir, Rezende, Danilo Jimenez, and Welling, Max. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pp. 3581–3589, 2014.

Li, Yujia, Swersky, Kevin, and Zemel, Richard. Learning unbiased features. *arXiv preprint arXiv:1412.5244*, 2014.

Long, Mingsheng and Wang, Jianmin. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

Rahimi, Ali and Recht, Benjamin. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in neural information processing systems*, pp. 1313–1320, 2009.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. *International Conference on Machine Learning (ICML)*, 2014.

Tzeng, Eric, Hoffman, Judy, Zhang, Ning, Saenko, Kate, and Darrell, Trevor. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014. URL `http://arxiv.org/abs/1412.3474`.

van der Maaten, L. Barnes-Hut-SNE. *ArXiv e-prints*, January 2013.

Zemel, Rich, Wu, Yu, Swersky, Kevin, Pitassi, Toni, and Dwork, Cynthia. Learning fair representations. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 325–333, 2013.

Zhao, Ji and Meng, Deyu. Fastmmd: Ensemble of circular discrepancy for efficient two-sample test. *Neural computation*, 2015.

A Discrimination metrics

The Discrimination metric (Zemel et al., 2013) and the Discrimination metric that takes into account the probabilities of the correct class are mathematically formalized as:

$$\begin{aligned} \text{Discrimination} &= \left| \frac{\sum_{n=1}^{N} \mathbb{I}[y_n^{s=0}]}{N_{s=0}} - \frac{\sum_{n=1}^{N} \mathbb{I}[y_n^{s=1}]}{N_{s=1}} \right| \\ \text{Discrimination prob.} &= \left| \frac{\sum_{n=1}^{N} p(y_n^{s=0})}{N_{s=0}} - \frac{\sum_{n=1}^{N} p(y_n^{s=1})}{N_{s=1}} \right| \end{aligned}$$

where $\mathbb{I}[y_n^{s=0}] = 1$ for the predictions y_n that were done on the datapoints with nuisance variable $s = 0$, $N_{s=0}$ denotes the total amount of datapoints that had nuisance variable $s = 0$ and $p(y_n^{s=0})$ denotes the probability of the prediction y_n for the datapoints with $s = 0$. For the predictions and their respective probabilities we used a Logistic Regression classifier.

B Proxy A-Distance (PAD) for Amazon Reviews dataset

Similarly to Ganin et al. (2015), we also calculated the Proxy A-distance (PAD) (Ben-David et al., 2007; 2010) scores for the raw data $\mathbf{x}$ and for the $\mathbf{z}_1$ representations of VFAE. Briefly, Proxy A-distance is an approximation to the $\mathcal{H}$-divergence measure of domain distinguishability proposed in Kifer et al. (2004) and Ben-David et al. (2007; 2010). To compute it we first need to train a learning algorithm on the task of discriminating examples from the source and target domain. Afterwards we can use the test error ϵ of that algorithm in the following formula:

$$\text{PAD}(\epsilon) = 2(1 - 2\epsilon)$$

It is clear that low PAD scores correspond to low discrimination of the source and target domain examples from the classifier. To obtain ϵ for our model we used Logistic Regression. The resulting plot can be seen in Figure 6, where we have also added the plot from DANN (Ganin et al., 2015), where they used a linear Support Vector Machine for the classifier, as a reference. It can be seen that our VFAE model can factor out the information about $\mathbf{s}$ better, since the PAD scores on our new representation are, overall, lower than the ones obtained from the DANN architecture.

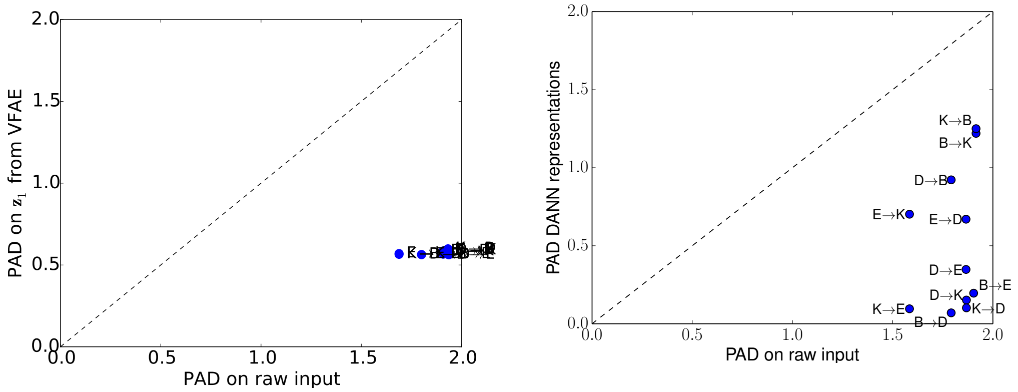


Figure 6: Proxy A-distances (PAD) for the Amazon reviews dataset: left from our VFAE model, right from the DANN model (taken from Ganin et al. (2015))