

# A Data Driven Similarity Measure and Example Mapping Function for General, Unlabelled Data Sets

Damien Lejeune and Kurt Driessens<sup>1</sup>

**Abstract.** Deep networks such as autoencoders and deep belief nets are able to construct alternative, and often informative, representations of unlabeled data by searching for (hidden) structure and correlations between the features chosen to represent the data and combining them into new features that allow sparse representations of the data. These representations have been chosen to often increase the accuracy of further classification or regression accuracy when compared to the original, often human chosen representations. In this work, we attempt an investigation of the relation between such discovered representations found using related but differently represented sets of examples. To this end, we combine the cross-domain comparison capabilities of unsupervised manifold alignment with the unsupervised feature construction of deep belief nets, resulting in an example mapping function that allows re-encoding examples from any source to any target task. Using the t-Distributed Stochastic Neighbour Embedding technique to map translated and real examples to a lower dimensional space, we employ KL-divergence to define a dissimilarity measure between data sets enabling us to measure found representation similarities between domains.

## 1 Introduction

While raw data is abundant, the difficulty with using this data, and according to the authors' one of the biggest challenges in the current big data hype, is the lack of any structured way of representing the data, leading to many different, human chosen, but unmatching representations of similar or related, but most of the time not identical<sup>2</sup> data. Examples of this are numerous, ranging from data stemming from medical questionnaires, where almost never the same questions are asked, but the topics are often similar, over gene transcription data where old style microarray data and more recent RNAseq measurements exist over a pool of intersecting but not identical gene sets to control oriented data where samples of system behaviour of a number of control problems exist, but almost never match in the chosen representation.

The subfield of machine learning in which this problem is tackled is known under the names transfer learning, inductive transfer and domain adaptation. The idea behind transfer learning is to enhance learning performance on a task by employing, i.e., re-using data, experience, and/or solutions from different but related tasks that were solved earlier. Existing work on transfer learning and in domain adaptation for supervised tasks [18, 9, 11] mainly focuses on the shift of the probability distributions observed between different tasks and how to correct for those, but not on the issue of different

representations. Also in reinforcement learning, the usual drawback of the tabula rasa approach when confronting new tasks has lead to a flurry of research on transfer learning [28]. Historically, defining the relation between the old task and the new, e.g. feature mapping, goal mapping, translating the model or policy to match the representations was handled by a human expert [29].

More recent work proposes autonomous transfer methods that aim at deriving the inter task mapping from learning examples of the two tasks automatically [27, 8, 7, 5]. These work by studying, or having an algorithm analyse the internal structure of the examples in the data set and trying to exploit the similarities between data sets in such structure. For example, in reinforcement learning, the dynamics of the task to be solved can be observed from sample interactions with the environment, consisting of the state the agent was in, the action that was selected and the state the action lead to. Analysing these dynamics and mapping the samples from both tasks into a joint feature space can give an indication of how the two tasks are related, i.e. where and how they shared dynamics and control response. The same idea applies to standard supervised learning tasks such as classification or regression, where the structure arises from the fact that, in principle, examples do not uniformly cover the entire example space as defined by the human chosen representational format. Existing work that takes this approach has used sparse coding [8] and manifold alignment [32] to define the joint space.

The contribution of this work is twofold, as we introduce:

- (i) a **data driven difference measure** for comparing data sets
- (ii) an **automatically derived inter-task mapping** that can be used to compare any two data sets and thus learning tasks, whether they are supervised, unsupervised or reinforcement learning.

The approach takes the shape of a pipeline employing well studied and tested techniques. The pipeline relies on deep belief nets to generate expressive features for both data repositories and on unsupervised manifold alignment to find the best mapping between these features. Applying one deep belief net to generate hidden feature activations, a forward and backwards projection into the alignment space and the other deep belief net to reconstruct the visible node activations from the projected hidden node activations, it becomes possible to translate learning examples from a source task into examples for the target task. By comparing the embedding of the original and the translated examples in a low dimensional projection built using the t-Distributed Stochastic Neighbour Embedding, we define a difference measure based on the Kullback-Leibler divergence between the two example distributions. Experiments show that the pipeline is able to autonomously find meaningful analogies between data-sets that match human intuition.

The pipeline draws on the power of already developed techniques

<sup>1</sup> Maastricht University, email: kurt.driessens@maastrichtuniversity.nl

<sup>2</sup> With identical, we refer to the representation and domain of the data, not the examples measured.

and applies and combines them in a new problem domain. The combination of the techniques adds little to no complexity to these techniques except for the way in which they are combined. No additional parameters to set or tune are introduced and no expert knowledge on any of the involved tasks is required. The exact set-up of the pipeline will be further discussed in section 3.

The rest of the paper is structured as follows: we introduce the necessary concepts and discuss related work in section 2. Section 4 demonstrates the capabilities of the pipeline using data sets from a number of well known classification tasks and reinforcement learning benchmarks to show that the discovered relations and similarities have a striking resemblance to human intuition.

## 2 Preliminaries

In this section, we introduce the problem we are trying to address with this work, we present related work and introduce the different concepts we will use later on. Given the similarity possible application of our technique to transfer learning and the similarity of the settings, we will adopt the vocabulary of the transfer learning domain and talk about **source** and **target** data sets/tasks to make referencing the two different domain easier.

### 2.1 Autonomous Transfer Learning

Transfer learning [30, 28] is concerned with the re-use of data, or experience in the case of reinforcement learning, or the learned models and policies from a previously learned source task, to improve learning performance in a new, sometimes more complex target task. **Shallow** transfer is concerned with the transfer of information when the learning examples from the source and the target task share their feature space. However, since these features are usually chosen by a human expert, or simply chosen by their availability for a learning task, they often differ between tasks, leading to what is known as **deep** transfer. This is the setting under which we operate in this work. When reusing previous experience from a source task  $S$  with feature representation  $\psi_S$  in a new target task  $T$  with different features  $\psi_T$ , the two representations need to be related. This relation often takes the form of an inter-task mapping  $\mathcal{T} : \psi_S \rightarrow \psi_T$  that relates the expert chosen features to each other.

When a set of source tasks  $(S_1, S_2, \dots, S_n)$  is available to transfer from, or if the task is to build an optimised learning path through a number of related learning tasks, a task similarity measure  $\mathcal{M}$  can be used to select which source task to transfer from, or which task to select next. Such a measure will map any combination of two learning tasks to  $\mathbb{R}$ , giving a score to each combination according to their similarity.

#### Related Work

Given the difficulties with its tabula rasa learning premise, transfer learning for reinforcement learning tasks has received quite a bit of attention in the past years [28]. Where historically the required inter-task mapping was defined by domain experts [29] more recent work has focussed on completely autonomous transfer. Bou Ammar et al. [8] construct a joint space for the source and task domain samples through sparse coding [21]. In this joint space, samples from the source and the target domain are paired using a Euclidean distance and used as learning examples in a supervised learning task that attempts to model the inter-task mapping. The problem with this approach is that in the commonly high dimensional joint space, the

Euclidean distance carries little information. In follow up work [7], Bou Ammar et al. use a three way restricted Boltzmann machine to learn the inter-task mapping. This setup relies on a complex training phase based on mini-batch learning and re-learning that randomly pairs samples from the two domains and relies on the reconstruction error of the Boltzmann machine to select which pairs match best. This makes training the machine quite involved, as many repetitions of randomly paired samples are necessary.

For supervised learning tasks, unsupervised manifold alignment has been used to relate the learning examples from the source and target tasks [32]. Very recently, Bou Ammar et al. [2] applied this manifold alignment technique to policy gradient reinforcement learning. Since this technique is so closely related to ours, in fact, the unsupervised manifold alignment is a part of our pipeline, we will compare the inter-task mapping generated by both techniques in our experimental section.

On top of the inter-task mapping, our approach also gives rise to a domain similarity measure. Of existing work on domain similarity measures, the most related to ours in the work by Bou Ammar et al. on RBDist [6] that bases its similarity measure on the reconstruction error of a single restricted Boltzmann machine and the follow up work on DRBDist [4] using a combination of Deep Belief Nets a bit like ours. The main difference is that DRBDist uses an identity mapping between the top layers of two DBNs. This not only places a restriction on the number of nodes of the top layer of the DBNs, i.e. the number of nodes must be equal, it also assumes a ordered pairwise similarity between the features learned by both networks. Additionally, since the similarity measure relies on the reconstruction error, it is very sensitive to the convergence results of the DBN learning phase. In fact, when trying to reproduce the results from this work, we were unable to reach the same results.

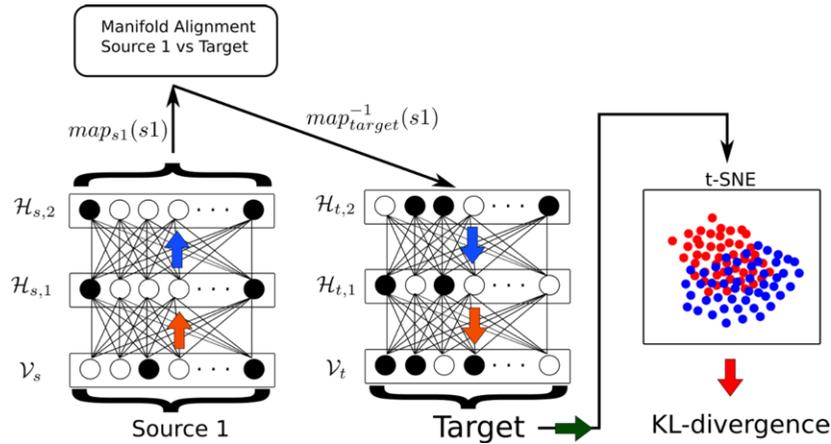
### 2.2 Deep Belief Networks

A restricted Boltzmann machine (RBM) is a neural network (NN) consisting of two layers (one visual and one hidden) of stochastic nodes that can be trained to represent a probability distribution over data points [19]. Each neuron in a layer is connected to all neurons in the other layer forming a fully connected bipartite graph. An RBM is trained using gradient descent where the gradient is not directly computed but estimated by the contrastive divergence algorithm [17]. A deep belief network (DBN) is a stack of RBMs trained in a layer-wise setup. Increasing the number of layers is done to improve the fit of the probability distribution and can be shown to extract, in an unsupervised way, higher level features from a dataset.

While a single hidden layer RBM is able to capture basic features, stacking multiple layers on top of each other allows learning higher level features. Such a stack of RBMs, which itself forms a NN, is called a Deep Belief Network (DBN). Each layer is trained in a greedy, layer wise fashion. This forces each additional hidden layer to learn a new feature representation to encode the layer below and makes this technique well suited for automated feature extraction. It has been successfully applied to domains like handwritten digit recognition [25], speech recognition [15] and Atari games [23], among other things [3].

### 2.3 Manifold Alignment

Finding feature correspondences in a source and a target task is important when assessing the similarity between them. Chang



**Figure 1.** Pipeline using DBNs, manifold alignment and t-SNE.

Wang [32] proposed a method to align two datasets of arbitrary dimensions by finding these correspondences in an unsupervised way. The alignment is performed in a new space into which the data can be mapped. The algorithm also provides inverse mappings which allow to reconstruct data points belonging to a source task into a target task’s original space. It then follows that the quality of this reconstruction will depend on the quality of the alignment and therefore will depend on the similarity between the two datasets.

The alignment and the corresponding transformation is achieved by first computing intra-similarities between the points in the same dataset. Then, the local geometries in both domains are represented by the distances of the  $k$ -nearest neighbours for each point. Intuitively, if the two sets have the same local geometries, up to affine transformations, the sets can be perfectly aligned.

Having this local information, the mappings for both domains can be found by minimising a cost function that forces points exhibiting similar geometries to be mapped closely together in the common space, and at the same time enforce the points which are close to others in their respective space to also be close in the common space.

## 2.4 t-Distributed Stochastic Neighbour Embedding

Measuring similarities between data points or data sets in high dimensional spaces is complex and not always meaningful. t-Distributed stochastic neighbour embedding or t-SNE [31] is a dimensionality reduction method that aims to maintain the same data distribution from the original high dimensional (HD) space in a low dimensional (LD) one. Since the technique originates in visualisation, LD space is usually two dimensional.

The mapping done in such a way that close points in the HD space should also be close into the LD space. In other words, t-SNE finds, in an unsupervised way, a mapping  $\mathbb{R}^n \rightarrow \mathbb{R}^2$  that conserves the local geometries of the original dataset. To do so, it computes a joint probability distribution for observing a point  $x_j$  around a point  $x_i$  using a multivariate Gaussian distribution centred at  $x_i$ . The algorithm also finds a proper variance for each distribution at  $x_i$  in order to model the local geometries by taking the density around that point into consideration. This probability is at the core of the method as the LD space should reflect the same probability distribution as the HD one and therefore model the same similarities. To achieve this, a gradient descent search is performed on the Kullback-Leibler divergence

with the aim at minimising the difference between the distribution in  $\mathbb{R}^n$  and the one in  $\mathbb{R}^2$ . The embedding in the low dimensional space comes at the price of not conserving the global geometries of the original dataset. Points far away in the HD space will be even farther away in the LD.

## 3 Data Driven Similarity Construction

We want our approach to match and find similarities between the two domains completely autonomously, so we define a set-up that computes a similarity measure between any two previously unseen (and possibly unlabelled) domains and constructs a mapping between the two domains that allows re-encoding samples from one domain into samples of the other. The similarity measure will indicate the degree to which the two domains match and will quantify the quality of the re-encoding.

### 3.1 Working Constraints

Having the inter-task mapping and the domain similarity measure constructed completely unsupervised leads to the following constraints:

- (i) No prior information about the domains must be required besides data samples.
- (ii) No requirements can be placed on the dimensionality of the different domains.

Our approach consists of a pipeline that makes use of three core methods to construct both the inter-task mapping and the similarity measure: (i) deep belief networks [16], (ii) manifold alignment [32] without correspondence and (iii) t-SNE [31].

### 3.2 The pipeline

Our pipelined approach, illustrated by Figure 1, goes as follows:

- (i) For each of the datasets, a separate DBN is used to extract high level features. Each dataset consists of (unlabelled) samples from one domain. This step is expected to result in a better representation of the data for each domain and to help increase the comparativeness by working on underlying characteristics instead of the

low level (human chosen) features of the raw data. No restrictions are placed on the number of nodes used in any layer or on the number of layers itself, so the DBN can be optimised to match the domain it is used for, independent of the rest of the pipeline and according to the experience of the user.

- (ii) The manifold alignment uses samples from two domains, re-encoded into the feature space extracted by the DBNs, and computes a mapping to a space where the samples from the source and the target domains can be compared. As mentioned in Section 2.3, this method is able to work with data originating from spaces with different dimensionalities, to get rid of affine transformations and to provide an inverse mapping that allows the data from one domain to be transformed into the other domain. The quality of this transformation is influenced by the quality of the alignment which in turn is dependent of the similarity between the two datasets. A good alignment will transform the source's data points into a set of data points that matches the original target distribution.
- (iii) In order to obtain a meaningful similarity measure, t-SNE reduces the original, usually high dimensional, space into a two dimensional one. Then, the Kullback-Leibler divergence (KL-divergence) is computed in t-SNE's space between the original target distribution and the distribution of the reconstructed source samples. This KL-divergence represents the measure between the source and target datasets.

The training of the DBNs is done for each domain to be measured, but since this step is independent of any other domains involved, it only has to be executed once for each domain. Subsequently, the manifold alignment is trained with the source's and target's last hidden layers activations for each pair of domains that we want to compare. The last phase is to train the t-SNE algorithm in order to reduce the dimensionality of the target's space. With the aim to use the KL-divergence over points laying on the plane generated by t-SNE, probability distributions for the target's original and reconstructed datasets have to be estimated. In the experiments reported below, we used kernel density estimators at  $150 \times 150$  equidistant points.

## 4 Experiments

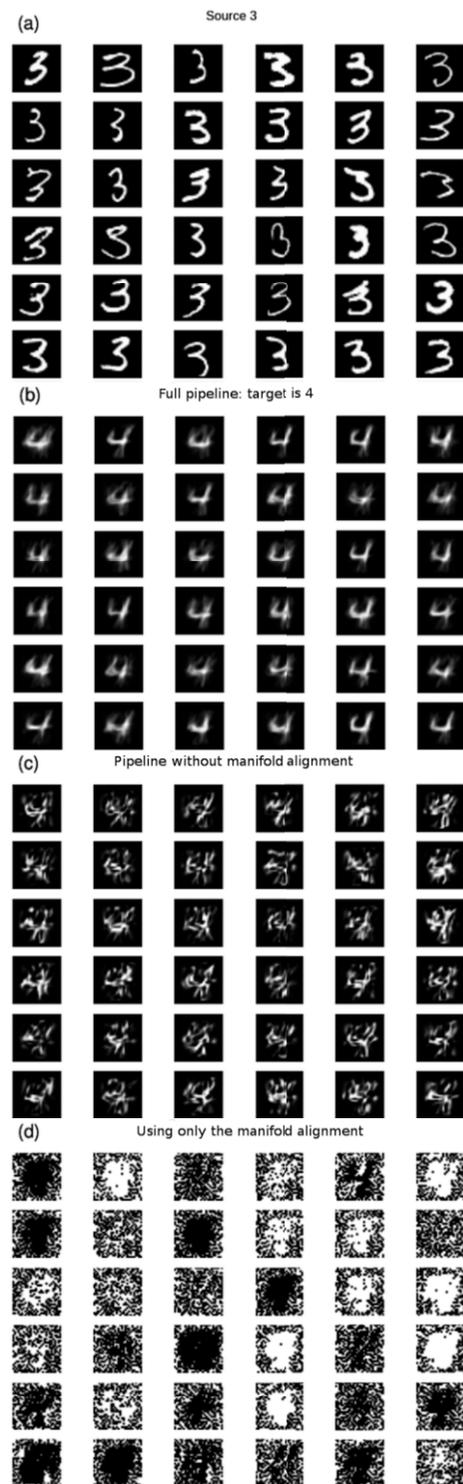
In this section we empirically evaluate the pipeline presented above. Since there is no ground truth when comparing different datasets with respect to the inter domain mapping and the computed similarity measure, we've selected domains that can be matched using human intuition, as well as domains from reinforcement learning where related work generated a base for comparison.

To compare with previous work and to illustrate the importance of each step in the pipeline we first use the MNIST dataset of written characters [20]. Since this dataset is composed of easy to interpret images, this data allows us to clearly show how the approach works and what the influence is of each step. To allow comparisons on this data set, we treat the samples of each digit as a separate domain, allowing us to visually show the results of the re-encoding to compare the similarity measure to human intuition.

To illustrate the use of the pipeline on data stemming from spaces with different dimensionality, we then apply our approach to accomplish transfer between the pen-digits dataset [1] and the MNIST dataset. This setup again allows us to show qualitative results of the re-construction. Additionally, we present some quantitative results of the influence our inter-task mapping method could have on classifier accuracy through example transfer.

We end with a set of experiments on reinforcement learning benchmarks which allows us to compare to previously published data

driven similarity measures [4].

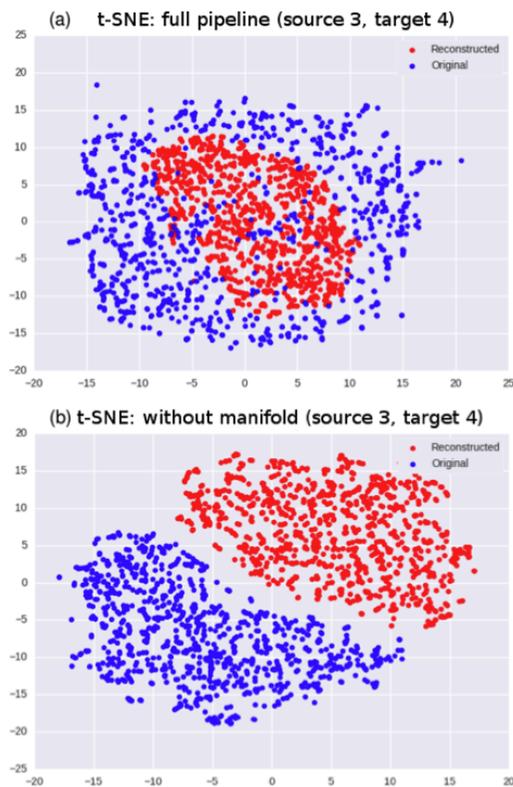


**Figure 2.** Experiment using 3 as source and 4 as target. (a) samples used as source (b) shows the reconstruction of with the full pipeline. (c) the reconstruction without manifold alignment. (d) the reconstruction without DBNs

## 4.1 MNIST

The MNIST [20] dataset holds images of centered handwritten digits, each composed of  $28 \times 28$  grayscale pixels. The goal of this experiment is to measure the similarity between different digits in MNIST. For example, it aims at testing if the digit 0 is closer to 8 than it is to 4. Multiple parameters had to be set up and have been kept fixed for all experiments in this section. Given that the pipeline is independent of the used DBN architecture, we decided to follow literature and composed the DBN of two hidden layers: the first one having 300 units and the second one having 100 units. Each NN was trained to model the distribution of one digit, using on average 6000 examples per digit. The manifold alignment used a parameter  $\mu = 1$  and was trained on 1000 sets of activations (i.e. samples) per dataset. Once the data is reconstructed by the target DBN, it is used to compute the t-SNE scatter plot. t-SNE initialised the LD space embedding using a PCA with 2 components and used a maximum of 1000 iterations for the optimisation.

Figure 2b shows the results when using the full pipeline, taking samples of the digit 3 as source and digit 4 as the target. The reconstructed digits are a bit more blurry than the originals, but the samples do cover both the open version of the digit 4 as well as the closed top version. The t-SNE scatter plot in Figure 3a shows that the reconstructed samples are centred inside the distribution of the original samples. This means that less variance will be observed from the reconstructed samples. The dissimilarity measure for this experiment was 11.61.



**Figure 3.** Examples of t-SNE for the experiment using 3 as source and 4 as target. (a) t-SNE associated to Figure 2b of the experiment with the full pipeline. (b) t-SNE associated to Figure 2c when the manifold alignment has been removed from the pipeline

Table 1 shows all dissimilarity values computed by our measure. For example 8 is easier to reconstruct from a 3 than a 4 which, from a human perspective, is explained by the loops and shape of 3 being similar to 8. Because of the lack of a ground truth, the table also includes the summarised results of an online survey we held to estimate human intuition about the similarity of written digits. In case two digits received approximately the same number of votes, both are listed. The survey was filled in by approximately 100 participants. One can observe that our approach agrees with the survey results for a number of cases, but surely not all.

### 4.1.1 Removing the DBNs

As discussed in the related work section, recent work by Bou Ammar et al. [2] constructs an inter-task mapping using only the manifold alignment part of our setup. In Figure 2d we depict the results obtained by attempting to transform a 3 digit into a 4 using the manifold alignment on the raw MNIST data. Comparing the results with those of Figure 2b shows the amount of noise generated by this approach.

### 4.1.2 Removing the manifold alignment

Figure 2c shows the experiment again, but removing the manifold alignment step, a setup briefly mentioned in [4]. Since both DBNs have the same architecture, the activations of the source’s last hidden layer can be mapped with an identity function to those of the target. It has to be noted that there is no theoretical foundation for the use of an identity mapping. The weights of the DBN are initialised randomly in order to break symmetry [14, p. 173] which leads to a random allocation of the high level features to the nodes, even when the DBN is trained multiple times on the same data. This experiment empirically demonstrates the necessity of using a method invariant under affine transformations. The results given by t-SNE in Figure 3b show that two distant distributions have been found. The KL-divergence for these distributions is 57.17, much higher than the value from the experiment using the full pipeline.

The results presented above demonstrate the necessity and advantages of each step in the pipeline. Each stage is required to solve a part of the problem described.

## 4.2 Pendigits

The pen-digits dataset [1] represents handwritten digits captured using a graphic tablet. Each instance is composed of 8 pairs of  $x, y$  coordinates taken along the path of the digit as depicted in Figure 4. The following experiment compares this sequential digit representation to the images of MNIST. While the domain of the two datasets is hand-written digits, the representation of the data is very different: first of all, the pen-digit samples have only 16 dimensions and represents coordinates over time, while MNIST uses 784 pixels intensities. While in this case, there seems to be a ground truth for the similarity measure to discover, without any background information about the two data-sets, they are very difficult, if not impossible for humans to match.

The pen-digit DBNs use two hidden layers with sizes of 60 units for the middle and 80 units for the last layer. The MNIST DBNs as well as the manifold alignment and t-SNE were configured as before.

Figure 4 shows the original pen-digit data for the digit 8 mapped on a two dimensional field, the reconstructed MNIST like images

Source\Target	0	1	2	3	4	5	6	7	8	9
0	$\ll 1$	49.67	12.38	3.47	10.15	12.70	9.03	15.91	6.75	11.18
1	14.19	$\ll 1$	10.96	10.27	10.99	12.50	10.03	7.16	7.74	13.20
2	13.98	46.41	$\ll 1$	12.36	11.78	8.68	14.35	16.64	4.94	9.83
3	11.24	39.71	10.90	$\ll 1$	11.61	8.48	13.14	16.91	3.58	8.08
4	13.58	32.64	11.14	7.80	$\ll 1$	11.40	12.71	10.12	4.40	9.04
5	13.39	56.18	9.50	5.45	10.80	$\ll 1$	15.84	14.43	5.42	15.16
6	11.78	41.37	10.27	6.21	8.22	11.73	$\ll 1$	15.66	5.23	13.61
7	14.38	47.07	8.61	7.19	6.07	12.23	12.22	$\ll 1$	9.69	8.34
8	14.14	44.05	9.30	6.95	12.56	10.44	16.73	17.46	$\ll 1$	15.98
9	12.99	41.39	8.37	5.07	9.30	13.56	13.81	9.36	6.29	$\ll 1$
2nd most similar	3	4	9	0	7	3	0	1	3	3
3rd most similar	6	3	7	9	6	2	1	9	4	4
human choices	6	7	3	8;2	9	6	0;5	1	3	4

**Table 1.** Similarity measure values obtained when comparing different MNIST digits. The bottom row represents the most similar digit according to a humans obtained with an online survey.

generated and the t-SNE plot of the resulting probability distributions. Again, it can be observed that although the reconstructions are a bit more blurry than the originals, a variety of recognisable 8 digits is produced.

We show the full dissimilarity matrix for all digits in Table 2. Surprisingly, most of the time, the pen-digits and their corresponding MNIST digit turn out to be the most similar by a huge margin when compared to others. For digit 3, there is a clear mismatch. For digits 1 and 2, the difference between the first and second closest digit is so small that it should be considered ambiguous. Given that t-SNE involves stochasticity in the mapping, two runs of the algorithm could give slightly different results [31] pointing to a different most similar digit. Nonetheless, even then, the matching digits remain among the topmost similar.

### 4.3 A Transfer Learning Scenario: Mapping Pendigit to MNIST

While not the primary aim of our data similarity measure, the inter-task mapping generated by our approach gives rise to a simple transfer learning scenario, as examples from one domain can be transformed into additional learning examples in another domain. To make this approach useful however, one would need to have both sufficient data to learn a deep belief network modelling the target data while at the same time, too few examples of the target set to be able to learn a good classification algorithm. Although these two seem to contradict each other, unsupervised learning of the DBN, followed by a fine tuning stage using a limited amount of data might be possible.

We tested the performance of such a transfer scenario using our automatically generated inter-task mapping. We performed a simple classification experiment that aims at identifying the digit 8 in a full set of digits, i.e. classifying 8's versus all other digits. For this we compared using a standard MNIST dataset with a 10% share of the dataset for each digit, with datasets that included an extra share of re-constructed 8's. We believe that this is a good analogy to how this technique could be used in practice, by generating a number of extra learning examples through the re-construction of a number of available source task examples. All experiments were performed using 10-fold cross validation, using the SMO support vector machine implementation in WEKA [13]. The results are displayed in Table 3.

They show a contribution made by the re-encoded learning examples. Unfortunately, we were unable to measure a correlation between the similarity measure and the amount of improvement in this setting. We assume the re-encodings of the examples are too close to measure a difference, enhanced by the fact that the re-encodings often appear at the center of the originals data distribution.

We would also like to emphasise that the experiment described in Table 3 is closely related to the problem described in domain adaptation [18, 9, 11]. Techniques used in domain adaptation attempt at using source data related to target data used to train a classifier in order to improve its accuracy [11]. The pipeline presented in this work could and should be investigated further in conjunction with the problems addressed by domain adaptation.

### 4.4 Comparing Markov Decision Processes

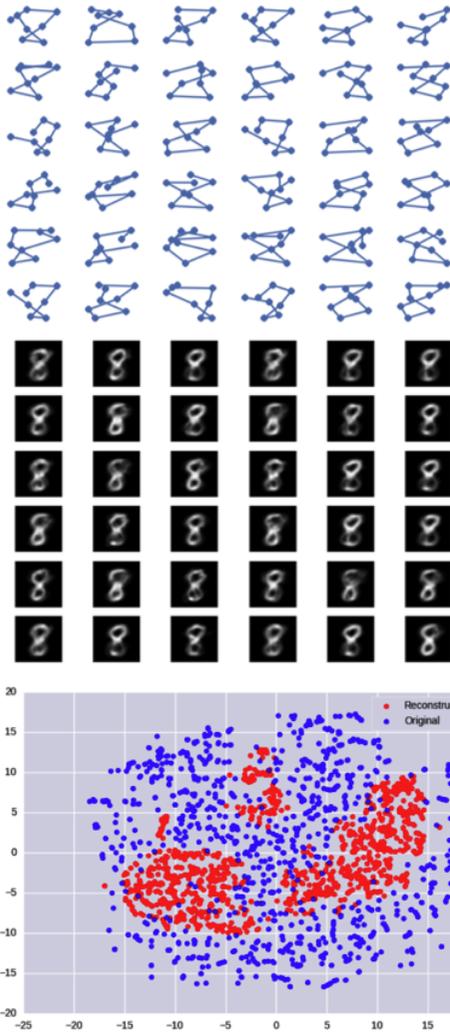
Next, we test our approach on three standard reinforcement learning benchmarks that have been previously compared [7]: (1) inverted pendulum, where the goal is to swing up and balance a pole with an underpowered motor, (2) cart pole, where the goal is to balance a pole hinged to a cart by pulling the cart back and forth and (3) mountain car, where the goal is to drive an under-powered car up a hill by building up momentum.

The datasets for these tasks were generated by uniformly sampling the environment for a state  $s$ , picking the action  $a$  that maximises a Q-function learned by SARSA and adding the state  $s'$  that the chosen action led to to build  $\langle s, a, s' \rangle$  triplets that sample the domain's transition function. For each task 5000 samples were generated. Computing the similarities between the different domains resulted in Table 4 and match the ranking found in [4].

The table highlights the similarities between IP $\rightarrow$ MC and CP $\rightarrow$ MC and the fact that the similarity measure is not symmetric. MC does not seem to present a close similarity to IP nor CP. We believe this is caused by the relative simplicity of the learned policy of MC. The t-SNE space for the CP $\rightarrow$ MC transfer case is shown in Figure 5. It can be observed that the samples from MC are divided into, what we believe are, separate pathways. The samples re-constructed from CP do not follow these pathways, but represent a decent spread over the example space. In the reverse transfer case (MC $\rightarrow$ CP), this is not the case and all samples re-encoded from MC are grouped together instead.

Pen\MNIST	0	1	2	3	4	5	6	7	8	9
0	<b>22.42</b>	68.09	87.47	71.72	81.99	63.04	70.19	83.23	72.56	77.77
1	57.56	<b>45.06</b>	59.04	62.30	57.06	64.08	55.95	60.24	51.51	62.28
2	68.76	55.05	<b>45.15</b>	63.49	60.09	51.46	60.44	51.13	63.08	58.88
3	66.01	54.63	81.02	51.42	63.86	32.06	81.85	61.56	64.90	65.14
4	56.88	61.22	55.08	58.84	<b>17.40</b>	48.34	44.61	50.52	64.18	29.66
5	43.86	46.51	49.61	<b>31.60</b>	46.95	<b>18.01</b>	49.10	44.59	31.57	37.20
6	70.12	60.91	65.05	73.87	69.00	71.35	<b>24.17</b>	63.41	73.54	62.02
7	61.41	56.61	53.34	50.77	47.28	57.02	63.48	<b>14.02</b>	58.72	46.40
8	61.43	<b>43.34</b>	<b>44.75</b>	42.24	45.75	24.45	58.16	46.80	<b>8.31</b>	48.55
9	73.64	66.28	65.04	62.55	48.07	53.82	61.94	44.28	56.28	<b>12.71</b>
Most similar	0	8,1*	8,2*	5	4	5	6	7	8	9

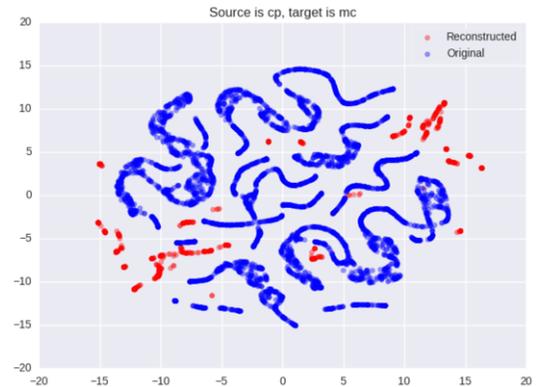
**Table 2.** Table of dissimilarities between the pen-digits as source and the MNIST as target. Asterisks highlight ambiguities in the similarity between digits.



**Figure 4.** Pen-digit experiment. The top image shows samples from the pen-digit dataset. Only the dots are taken into consideration by the DBNs, the lines represent the sequential nature of the measurements. The middle image shows the corresponding MNIST reconstruction. The bottom image is the t-SNE plot.

Source	Dissimilarity Value	ROC area for class 8
none	n/a	0.848
pen digit 5	31.57	0.92
pen digit 6	73.54	0.92
pen digit 8	8.31	0.92

**Table 3.** Table of dissimilarities between the pen-digits as source and the MNIST as target. Asterisks highlight ambiguities in the similarity between digits.



**Figure 5.** t-SNE space for transferring CP samples to MC.

## 5 Conclusions and Future Work

We presented a fully autonomous and data driven technique for computing (i) an inter-task mapping and (ii) dissimilarity measure for unrestricted data sets. While domain specific data driven similarity measures exists, for example in the image analysis field [26], it is our belief that we are the first to present a technique that is able to trans-

Source\Target	IP	CP	MC
IP	<b>0.76</b>	20.84	4.23
CP	18.35	<b>5.19</b>	7.05
MC	20.12	21.70	<b>0.002</b>

**Table 4.** The dissimilarity values between all reinforcement learning tasks.

late learning examples between highly dissimilar domains with such high correlation to human intuition. We showed in the experimental section that the presented technique can be applied to supervised, unsupervised and reinforcement learning tasks. We showed a number of experiments that both quantitatively and qualitatively illustrate of the power of the presented technique as well as the necessity and contribution of each part of the involved pipeline.

In future work we would like to approach even more challenging tasks such as measuring the similarity of phonemes in speech and test the robustness by using, for example, affNIST (i.e. MNIST with affine transformations). We would also like expand the technique to e.g. bigger images that require convolutional networks. This leads to additional complexity caused by the challenge in reversing convolution and pooling layers. However, related work exists for reversing these networks and using them to reconstruct data [22, 10] that could be useful to extend the pipeline presented.

We also plan to research transfer learning scenarios in which this technique could be used as a basis. While there are straightforward applications of the dissimilarity measure, such as source domain selection when multiple sources are available, the initial test using the inter-tasks mapping to do example transfer does not yet seem to lead to a useful approach. The examples transformed through a successful mapping function seem to converge to the center of the target data set, not necessarily leading to additional information about the classes boundaries. To make the approach successful, a way will have to be devised to generate a more diverse set of transferred examples.

## Acknowledgments

We would like to thanks Chang Wang for providing the source code of the manifold alignment and allowing us to reproduce the experiments in [32]. In addition, we would like to mention the projects Pylearn2 [12] and Scikit-learn [24] which have been used in the experiments. Finally, we would like to thank Haitham Bou Ammar for the interesting discussions.

## REFERENCES

- [1] F. Alimoglu and E. Alpaydin, ‘Combining multiple representations and classifiers for pen-based handwritten digit recognition’, in *Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on*, volume 2, pp. 637–640 vol.2, (Aug 1997).
- [2] Haitham Bou Ammar, Eric Eaton, Paul Ruvolo, and Matthew E. Taylor, ‘Unsupervised Cross-Domain Transfer in Policy Gradient Reinforcement Learning via Manifold Alignment’, in *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, (January 2015). 27
- [3] Yoshua Bengio, Aaron Courville, and Pascal Vincent, ‘Representation learning: A review and new perspectives’, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **35**(8), 1798–1828, (2013).
- [4] Haitham Bou-Ammar, *Automated Transfer in Reinforcement Learning*, Ph.D. dissertation, Maastricht University, 2013.
- [5] Haitham Bou-Ammar, Eric Eaton, Paul Ruvolo, and Matthew E. Taylor, ‘Unsupervised cross-domain transfer in policy gradient reinforcement learning via manifold alignment’, in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, January 25-30, 2015, Austin, Texas, USA., pp. 2504–2510, (2015).
- [6] Haitham Bou-Ammar, Eric Eaton, Matthew E. Taylor, Decebal Mocanu, Kurt Driessens, Gerhard Weiss, and Karl Tuyls, ‘An automated measure of MDP similarity for transfer in reinforcement learning’, in *Proceedings of the AAAI’14 Workshop on Machine Learning for Interactive Systems*, (July 2014).
- [7] Haitham Bou-Ammar, Decebal Constantin Mocanu, Matthew E. Taylor, Kurt Driessens, Karl Tuyls, and Gerhard Weiss, ‘Automatically mapped transfer between reinforcement learning tasks via three-way restricted boltzmann machines’, in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part II*, pp. 449–464, (2013).
- [8] Haitham Bou-Ammar, Karl Tuyls, Matthew E. Taylor, Kurt Driessens, and Gerhard Weiss, ‘Reinforcement learning transfer via sparse coding’, in *International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, Valencia, Spain, June 4-8, 2012 (3 Volumes)*, pp. 383–390, (2012).
- [9] Barbara Caputo and Novi Patricia, ‘Overview of the imageclef 2014 domain adaptation task’, in *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014.*, pp. 341–347, (2014).
- [10] A. Dosovitskiy, T.J. Springenberg, and T. Brox, ‘Learning to generate chairs with convolutional neural networks.’, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1538–1546, (2015).
- [11] Boqing Gong, Kristen Grauman, and Fei Sha, ‘Learning kernels for unsupervised domain adaptation with applications to visual object recognition’, *International Journal of Computer Vision*, **109**(1-2), 3–27, (2014).
- [12] Ian J. Goodfellow, David Warde-Farley, Pascal Lamblin, Vincent Dumoulin, Mehdi Mirza, Razvan Pascanu, James Bergstra, Frédéric Bastien, and Yoshua Bengio, ‘Pylearn2: a machine learning research library’, arXiv preprint arXiv:1308.4214, (2013).
- [13] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten, ‘The weka data mining software: An update’, *SIGKDD Explor. Newsl.*, **11**(1), 10–18, (November 2009).
- [14] Mohamad H. Hassoun, *Fundamentals of Artificial Neural Networks*, MIT Press, Cambridge, MA, USA, 1st edn., 1995.
- [15] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury, ‘Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups’, *Signal Processing Magazine, IEEE*, **29**(6), 82–97, (November 2012).
- [16] Geoffrey Hinton, Simon Osindero, and Yee-Whye Teh, ‘A fast learning algorithm for deep belief nets’, *Neural computation*, **18**(7), 1527–1554, (2006).
- [17] Geoffrey E. Hinton, ‘Training products of experts by minimizing contrastive divergence’, *Neural Comput.*, **14**(8), 1771–1800, (August 2002).
- [18] Judy Hoffman, Erik Rodner, Jeff Donahue, Brian Kulis, and Kate Saenko, ‘Asymmetric and category invariant feature transformations for domain adaptation’, *International Journal of Computer Vision*, **109**(1-2), 28–41, (2014).
- [19] Yann Lecun, Sumit Chopra, Raia Hadsell, Ranzato marc’ aurelio, and fu-Jie Huang, ‘A Tutorial on Energy-Based Learning’, in *Predicting Structured Data*, eds., G. Bakir, T. Hofman, B. schölkopf, A. Smola, and B. Taskar. MIT Press, (2006).
- [20] Yann Lecun and Corinna Cortes, ‘The MNIST database of handwritten digits’.
- [21] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng, ‘Efficient sparse coding algorithms’, in *In NIPS*, pp. 801–808. NIPS, (2007).
- [22] A. Mahendran and A. Vedaldi, ‘Understanding deep image representations by inverting them.’, in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 5188–5196, (June 2015).
- [23] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller, ‘Playing atari with deep reinforcement learning’, *CoRR, abs/1312.5602*, (2013).
- [24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, ‘Scikit-learn: Machine learning in Python’, *Journal of Machine Learning Research*, **12**, 2825–2830, (2011).
- [25] Jürgen Schmidhuber, ‘Multi-column deep neural networks for image classification’, in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), CVPR ’12*, pp. 3642–3649, Washington, DC, USA, (2012). IEEE Computer Society.
- [26] Abhinav Shrivastava, Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros, ‘Data-driven visual similarity for cross-domain image matching’, in *Proceedings of the 2011 SIGGRAPH Asia Conference, SA ’11*, pp. 154:1–154:10, New York, NY, USA, (2011). ACM.
- [27] Matthew E. Taylor, Gregory Kuhlmann, and Peter Stone, ‘Autonomous transfer for reinforcement learning’, in *Proceedings of the Seventh*

- International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 283–290, (May 2008).
- [28] Matthew E. Taylor and Peter Stone, ‘Transfer learning for reinforcement learning domains: A survey’, J. Mach. Learn. Res., **10**, 1633–1685, (December 2009).
- [29] Matthew E. Taylor, Shimon Whiteson, and Peter Stone, ‘Transfer via inter-task mappings in policy search reinforcement learning’, in Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 156–163, (May 2007).
- [30] L. Torrey and J. Shavlik, ‘Transfer learning’, Handbook of Research on Machine Learning Applications, IGI Global, **3**, 17–35, (2009).
- [31] Laurens Van der Maaten and Geoffrey Hinton, ‘Visualizing data using t-sne’, Journal of Machine Learning Research, **9**(2579-2605), 85, (2008).
- [32] Chang Wang and Sridhar Mahadevan, ‘Manifold alignment without correspondence’, in IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009, pp. 1273–1278, (2009).