Transfer Learning by Structural Analogy

Huayan Wang
Computer Science Department
Stanford University
353 Serra Street, Stanford, CA, U.S.A.

Qiang Yang
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Clearwater Bay, Kowloon, Hong Kong

Abstract

Transfer learning allows knowledge to be extracted from auxiliary domains and be used to enhance learning in a target domain. For transfer learning to be successful, it is critical to find the similarity between auxiliary and target domains, even when such mappings are not obvious. In this paper, we present a novel algorithm for finding the structural similarity between two domains, to enable transfer learning at a structured knowledge level. In particular, we address the problem of how to learn a non-trivial structural similarity mapping between two different domains when they are completely different on the representation level. This problem is challenging because we cannot directly compare features across domains. Our algorithm extracts the structural features within each domain and then maps the features into the Reproducing Kernel Hilbert Space (RKHS), such that the "structural dependencies" of features across domains can be estimated by kernel matrices of the features within each domain. By treating the analogues from both domains as equivalent, we can transfer knowledge to achieve a better understanding of the domains and improved performance for learning. We validate our approach on a large number of transfer learning scenarios constructed from a real world dataset.

Introduction and Motivation

Re-using knowledge across different learning tasks (domains) has long been addressed in the machine learning literature (Thrun 1998; Caruana 1997; Daumé III 2006; Dai 2008; Blitzer 2006). Existing research on this issue usually assume that the tasks are related on the *low* level, i.e. they share the same feature space or the same parametric family of models, such that knowledge transfer can be achieved by re-using weighted samples across tasks, finding a shared intermediate representation, or learning constraints (informative priors) on the model parameters.

However, examining knowledge transfer in human intelligence, we could find that human beings do not rely on such low-level relatedness to transfer knowledge across domains. Namely, we human beings are able to make analogy across different domains by resolving the *high* level (structural) similarities even when the learning tasks (domains) are seemingly irrelevant. For example, we can easily understand the analogy between debugging for computer viruses and diagnosing human diseases. Even though the computer viruses (harmful codes) themselves have nothing in common with bacteria or germs, and the computer systems is totally different from our bodies, we can still make the analogy base on the following *structural* similarities:

1. Computer viruses cause malfunction of computers. Diseases cause disfunction of the human body.
2. Computer viruses spread among computers through the networks. Infectious diseases spread among people through various interactions.
3. System updates help computers avoid certain viruses. Vaccines help human beings avoid certain diseases.

Understanding of these structural similarities helps us abstract away the details specific to the domains, and build a mapping between the abstractions (see Figure 1). The mapping builds on the high level structural relatedness of the two domains, instead of their low level "literal similarities". In other words, the attributes of the "computer" and the "human" themselves do not matter to the mapping, whereas their relationships to other entities in their own domains matter.

This is reminiscent of the *learning-by-analogy* paradigm in early endeavors in intelligent planing and problem solving. However, many previous operational systems in computational analogy, such as case-based reasoning, have used a simple similarity function between an old and new problem domain, whereby the features in the two domains are identical, albeit weighted. This similarity measure cannot handle some more intuitive cases of human problem solving, such as the above example, in which the similarity between the domains should be measured on the structural level. And such a "structural similarity" can only be determined if we can correctly identify *analogues* across completely different representation spaces.

On the other hand, in cognitive science, analogical learning indeed involves developing a set of mappings between features from different domains. Such a need is captured in structure mapping theory (Falkenhainer 1989; Gentner 1990) of analogical reasoning, which argued for deep re-

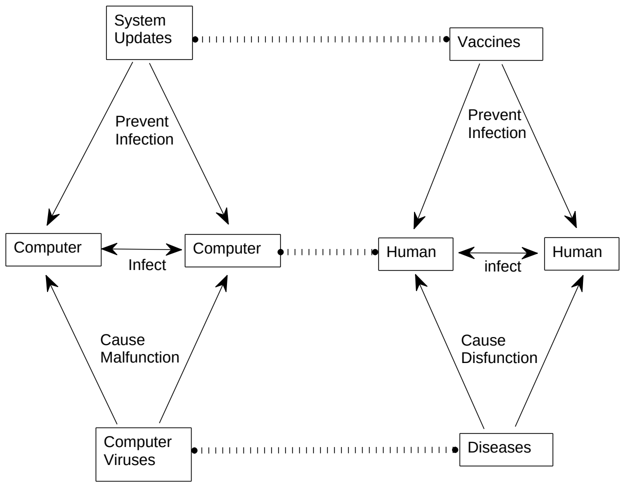

Figure 1: We can make the analogy between debugging for computer viruses and diagnosing human diseases based on structural similarities. The dash lines bridge analogues across domains.

lational similarity rather than superficial similarity. However, an operational computational theory has been lacking for how to come up with the mapping function. We try to fill this gap in this paper.

In this paper, we present a framework of *transfer learning by structural analogy*, which builds on functional space embedding of distributions (Smola 2007). Specifically, we address transfer learning in a setting that the source domain and target domain are using *completely different representation spaces*. As we cannot directly compare features across domains, we extract the structural information of the features within each domain by mapping the features into the Reproducing Kernel Hilbert Space (RKHS), such that the "structural dependencies" of features across domains can be estimated by kernel matrices of the features within each domain (Smola 2007). Hence the learning process is formulated as simultaneously selecting and associating features from both domains to maximize the dependencies between the selected features and response variables (labels), as well as between the selected features from both domains. With the learned cross-domain mapping, a structural similarity between the two domains can be readily computed, which can be used in place of simple similarity measures in computational analogy systems such as case based reasoning. By treating the analogues from both domains as equivalent, we can transfer knowledge to achieve a better understanding of the domains, e.g. better accuracy in classification tasks.

Related Work

The idea of re-using knowledge across learning tasks (domains) has been addressed in the machine learning literature in different terminologies, such as learning to learn, multi-task learning, domain adaptation, and transfer learning (Thrun 1998; Caruana 1997; Daumé III 2006; Dai 2008; Blitzer 2006; Mahmud 2007). To the best of our knowledge, among these works (Dai 2008) and (Mahmud 2007) are the only ones that address transferring knowledge across different representations spaces. However, (Dai 2008) rely on co-occurrence observations that bridges the two feature spaces (such as a dictionary, which consists of co-occurrence observations of two languages), such that the cross-domain relations of the features can be estimated straightforwardly. In contrast, our work does not rely on the availability of such co-occurrence data. (Mahmud 2007) proposed theoretical foundations for transfer learning between arbitrary tasks based on Kolmogorov complexity. However they only showed how to implement their framework in the context of decision trees, whereas our framework of making structural analogy between the features can be applied together with many different learning algorithms.

Learning by analogy is one of the fundamental insights of artificial intelligence. Humans can draw on the past experience to solve current problems very well. In AI, there has been several early works on analogical reasoning, such as Dynamic Memory (Schank 1982). Using analogy in problem solving, (Carbonell 1981; Winston 1980) pointed out that analogical reasoning implies that the relationship between entities must be compared, not just the entity themselves, to allow effective recall of previous experiences. (Forbus 1998) has argued for high-level structural similarity as a basis of analogical reasoning. (Holyoak 1997) has developed a computational theory of analogical reasoning using this strategy, when abstraction rules given as input that allow the two instances to be mapped to a unified representation.

Analogical problem solving is the cornerstone for case-based reasoning (CBR), where many systems have been developed. For example, HYPO (Ashley 1991) retrieves similar past cases in a legal case base to argue in support of a claim or make counter-arguments. PRODIGY (Carbonell 1991) uses a collection of previous problem solving cases as a case base, and retrieves the most similar cases for adaptation.

However, most operational systems of analogical reasoning, such as CBR systems (Aamodt 1994; Watson 1997; Leake 1996; Kolodner 1993), have relied on the assumption the past instances and the new target problem be in the same representational space. Most applications of CBR fall in this case (Mark 1989; Cheetham 2007; Bayoudh 2007), where the sets of feature that describe the old cases and new problems are the same. For example, cases for car diagnosis are built on descriptions of automobile attributes such as battery and engine size, although the values are allowed to be different between a past case and the current problem.

Approach

Estimating Structural Dependencies by HSIC

We aim at resolving the structural analogy between two domains with completely different low-level representations. For the source domain we are provided with observations and response variables (labels):

$$\mathbb{S} = \{(x_1^{(s)}, y_1^{(s)}), (x_2^{(s)}, y_2^{(s)}), \cdots, (x_{N_s}^{(s)}, y_{N_s}^{(s)})\} \subset \mathcal{X}_s \times \mathcal{Y}_s, \tag{1}$$

where $\mathcal{X}_s$ is the source input domain and $\mathcal{Y}_s$ is the source output (label) domain. Similarity we have data for the target domain:

$$\mathbb{T} = \{(x_1^{(t)}, y_1^{(t)}), (x_2^{(t)}, y_2^{(t)}), \cdots, (x_{N_t}^{(t)}, y_{N_t}^{(t)})\} \subset \mathcal{X}_t \times \mathcal{Y}_t. \tag{2}$$

Note that $\mathcal{X}_t$, $\mathcal{Y}_t$ can be representation spaces that are *completely different* from $\mathcal{X}_s$, $\mathcal{Y}_s$.

For both the source and the target domain, we denote their feature domains as Φ_s and Φ_t. In practice, features are represented by their profiles[1] in the training set:

$$\{f_1^{(s)}, f_2^{(s)}, \cdots, f_S^{(s)}\} \subset \Phi_s, \tag{3}$$

$$\{f_1^{(t)}, f_2^{(t)}, \cdots, f_T^{(t)}\} \subset \Phi_t. \tag{4}$$

For vector representations, $(f_1^{(s)}, f_2^{(s)}, \cdots, f_S^{(s)})$ is simply the transpose of $(x_1^{(s)}, x_2^{(s)}, \cdots, x_N^{(s)})$. Nevertheless, our framework is applicable to more sophisticated representations (such as graphs *etc.*) as it is kernelized, which accesses data only through the kernel function.

Let $\mathcal{H}_s$, $\mathcal{H}_t$, $\mathcal{G}_s$, $\mathcal{G}_t$, $\mathcal{F}_s$, and $\mathcal{F}_t$ be reproducing kernel Hilbert spaces (RKHS) on the domains $\mathcal{X}_s$, $\mathcal{X}_t$, $\mathcal{Y}_s$, $\mathcal{Y}_t$, Φ_s and Φ_t, with associated kernel functions m_s, m_t, l_s, l_t, k_s and k_t respectively. Then we are able to estimate dependencies across domains using the Hilbert-Schmidt Independence Criterion (HSIC) (Gretton 2005; 2007; Smola 2007), which is defined as the square of the Hilbert-Schmidt norm of the cross-covariance operator bridging the two RKHS.

Specifically, for the RKHS $\mathcal{F}_s$ and $\mathcal{F}_t$ on the feature domains Φ_s and Φ_t, in terms of the kernel functions k_s, k_t the HSIC can be expressed as

$$\begin{aligned} \mathcal{D}(\mathcal{F}_s,\, \mathcal{F}_t,\, \Pr_{st}) = & \,\mathbf{E}_{ss'tt'}[k_s(s,s')k_t(t,t')] \\ & + \mathbf{E}_{ss'}[k_s(s,s')]\mathbf{E}_{tt'}[k_t(t,t')] \\ & - 2\mathbf{E}_{st}[\mathbf{E}_{x'}[k_s(s,s')]\mathbf{E}_{y'}[k_t(t,t')]], \end{aligned} \tag{5}$$

where $\Pr_{st}$ is the joint distribution of source and target domain features over $\Phi_s \times \Phi_t$, and (s,t), (s',t') are distributed independently according to the joint distribution.

Given a sample

$$\mathbb{F} = \{(f_1^{(s)}, f_1^{(t)}), (f_2^{(s)}, f_2^{(t)}), \cdots, (f_W^{(s)}, f_W^{(t)})\} \tag{6}$$

of the joint distribution $\Pr_{st}$, HSIC can be estimated using the kernel matrices (Song 2007):

$$\begin{aligned} \mathcal{D}(\mathcal{F}_s,\, \mathcal{F}_t,\, \mathbb{F}) = & \frac{1}{W(W-3)}[\mathrm{tr}(\mathbf{K}^s\mathbf{K}^t) \\ & + \frac{\mathbf{1}^\top\mathbf{K}^s\mathbf{1}\mathbf{1}^\top\mathbf{K}^t\mathbf{1}}{(W-1)(W-2)} - \frac{2}{W-2}\mathbf{1}^\top\mathbf{K}^s\mathbf{K}^t\mathbf{1}], \end{aligned} \tag{7}$$

where $\mathbf{K}^s(i,j) = (1-\delta_{ij})k_s(f_i^{(s)}, f_j^{(s)})$ and $\mathbf{K}^t(i,j) = (1-\delta_{ij})k_t(f_i^{(t)}, f_j^{(t)})$ are the kernel matrices with diagonal entries set to zero.

[1] The "profile" of a feature is defined as its feature value on all instances of a dataset.

Similarly, we can estimate the dependencies across the domains $(\mathcal{X}_s, \mathcal{Y}_s)$ and $(\mathcal{X}_t, \mathcal{Y}_t)$ by the corresponding kernel matrices $\mathbf{M}^s$, $\mathbf{L}^s$, $\mathbf{M}^t$ and $\mathbf{L}^t$ computed by the samples $\mathbb{S}$, $\mathbb{T}$ (in (1) and (2)) from the joint distributions $\Pr_{xy}^{(s)}$ and $\Pr_{xy}^{(t)}$, where $\mathbf{M}^s(i,j) = (1-\delta_{ij})m_s(x_i^{(s)}, x_j^{(s)})$, $\mathbf{L}^s(i,j) = (1-\delta_{ij})l_s(y_i^{(s)}, y_j^{(s)})$, $\mathbf{M}^t(i,j) = (1-\delta_{ij})m_t(x_i^{(t)}, x_j^{(t)})$ and $\mathbf{L}^t(i,j) = (1-\delta_{ij})l_t(y_i^{(t)}, y_j^{(t)})$.

Estimating dependencies by HSIC is a crucial component in our learning framework, which requires estimating dependencies for the three pairs of domains, namely the source input and output domain $(\mathcal{X}_s, \mathcal{Y}_s)$, the target input and output domain $(\mathcal{X}_t, \mathcal{Y}_t)$, and the source and target feature domain (Φ_s, Φ_t)

Transfer Learning by Structural Analogy

The joint distributions $\Pr_{xy}^{(s)}$ and $\Pr_{xy}^{(t)}$ are well characterized by the samples $\mathbb{S}$ and $\mathbb{T}$. So estimating HSIC for $(\mathcal{X}_s, \mathcal{Y}_s)$ and $(\mathcal{X}_t, \mathcal{Y}_t)$ can be carried out straightforwardly. However we have *no* direct sample from the joint distribution $\Pr_{st}$ because the samples in (3) and (4), *i.e.* the features from different domains, are not associated. Actually how to associate the features depends on the structures of each domain, and we therefore name the cross-domain dependency as "structural dependency", which can only be determined if we understand the structural analogy across the domains.

For a given association of the source and target domain features, as in (6), structural dependency between the domains can be estimated by (7). That means, by maximizing the estimated structural dependency, we find the "correct" association of the features from both domains, *i.e.* we make the *analogy* across domains.

Formally, given $W \le \min(S,T)$, let σ_s and σ_t be injectives from $\{1,\cdots,W\}$ to $\{1,\cdots,S\}$ and $\{1,\cdots,T\}$ respectively, we could describe the learning problem as selecting a *ordered* set of features

$$\begin{aligned} & \{f_{\sigma_s(1)}^{(s)}, f_{\sigma_s(2)}^{(s)}, \cdots, f_{\sigma_s(W)}^{(s)}\}, \text{ and} \\ & \quad \{f_{\sigma_t(1)}^{(t)}, f_{\sigma_t(2)}^{(t)}, \cdots, f_{\sigma_t(W)}^{(t)}\} \end{aligned} \tag{8}$$

from both the source and the target learning task, such that the objective function combining dependencies between $(\mathcal{X}_s, \mathcal{Y}_s)$, $(\mathcal{X}_t, \mathcal{Y}_t)$ and (Φ_s, Φ_t) is maximized:

$$\begin{aligned} (\hat{\sigma}_s, \hat{\sigma}_t) = & \arg\max_{\sigma_s, \sigma_t}[\mathcal{D}(\mathcal{F}_s,\, \mathcal{F}_t,\, \mathbb{F}) \\ & + \lambda_s\mathcal{D}(\mathcal{H}_s,\, \mathcal{G}_s,\, \mathbb{S}) + \lambda_t\mathcal{D}(\mathcal{H}_t,\, \mathcal{G}_t,\, \mathbb{T})] \end{aligned} \tag{9}$$

where $\mathbb{F} = \{(f_{\sigma_s(1)}^{(s)}, f_{\sigma_t(1)}^{(t)}), \cdots, (f_{\sigma_s(W)}^{(s)}, f_{\sigma_t(W)}^{(t)})\}$ is the pseudo-sample from the joint distribution $\Pr_{st}$ constructed by associating the selected features from both domains. All the three terms in (9) are estimated by the estimator (7) with kernel matrices $\mathbf{K}^s$, $\mathbf{K}^t$, $\mathbf{M}^s$, $\mathbf{L}^s$, $\mathbf{M}^t$ and $\mathbf{L}^t$ computed using the selected features in (8). λ_s and λ_t are free parameters controlling the relative influences the terms.

After determining σ_s and σ_t, each sample of the source domain can be "translated" into a sample for the target domain by treating the features $f_{\sigma_s(i)}^{(s)}$ and $f_{\sigma_t(i)}^{(t)}$ (analogues) as equivalent. Then standard supervised learning methods can be applied to the expanded training set of the target domain. Computing the structural similarity between the domains also becomes straightforward. One can directly measure the structural similarity by $\mathcal{D}(\mathcal{F}_s,\ \mathcal{F}_t,\ \mathbb{F})$.

It is noticeable that the above described learning paradigm bears some key features that can be viewed as prototype models of the components in human's learning by analogy:

1. The learner knows the key concepts in a familiar case (source domain).
2. The learner identifies key concepts in a new problem (target domain) by both analyzing the new problem itself and making the analogy from a previous familiar case base on their structural similarities.
3. The learner gains better understanding of the new problem thanks to the knowledge transferred from the previous familiar case.

Algorithm

We have presented the general framework of learning by structural analogy. However, finding the globally optimal solution to the optimization problem in (9) is not straightforward. In this paper, we present a simple algorithm to implement the framework by finding a local minimum of the objective.

Our algorithm first selects features from both domains by maximizing $\mathcal{D}(\mathcal{H}_s,\ \mathcal{G}_s,\ \mathbb{S})$ and $\mathcal{D}(\mathcal{H}_t,\ \mathcal{G}_t,\ \mathbb{T})$ respectively, without considering relations between the two domains. This is achieved by the forward selection method in (Song 2007).

Then we find the analogy by sorting the selected features for the source domain to maximize $\mathcal{D}(\mathcal{F}_s,\ \mathcal{F}_t,\ \mathbb{F})$. One advantage of this implementation is that we actually do not have to determine the weights λ_s and λ_t as the corresponding terms are maximized in separate procedures.

Then, sorting the selected features of the source domain to "make the analogy" is achieved by the algorithm proposed in (Quadrianto 2008). Specifically, we aim to find the optimal permutation π^* from the permutation group Π_W:

$$\pi^* = \arg\max_{\pi\in\Pi_W} \operatorname{tr} \bar{\mathbf{K}}^t \pi^\top \bar{\mathbf{K}}^s \pi \tag{10}$$

where $\bar{\mathbf{K}}^t = \mathrm{H}\mathbf{K}^t\mathrm{H}$, $\bar{\mathbf{K}}^s = \mathrm{H}\mathbf{K}^s\mathrm{H}$ and $\mathrm{H}_{ij} = \delta_{ij} - W^{-1}$. Note that a biased HSIC estimator $(W-1)^{-2}\operatorname{tr} \bar{\mathbf{K}}^t\pi^\top\bar{\mathbf{K}}^s\pi$ is used here instead of the unbiased estimator (7). Setting the diagonal elements in the kernel matrices to zero is recommended for sparse representations (such as bag-of-words for documents) which give rise to dominant signal on the diagonal. The optimization problem is solved iteratively by:

$$\pi_{i+1} = (1-\lambda)\pi_i + \lambda \arg\max_{\pi\in\Pi_W} \left[\operatorname{tr} \bar{\mathbf{K}}^t\pi^\top\bar{\mathbf{K}}^s\pi_i\right] \tag{11}$$

Since $\operatorname{tr} \bar{\mathbf{K}}^t\pi^\top\bar{\mathbf{K}}^s\pi_i = \operatorname{tr} \bar{\mathbf{K}}^s\pi_i\bar{\mathbf{K}}^t\pi^\top$, we end up solving a linear assignment problem (LAP) with the cost matrix $-\bar{\mathbf{K}}^s\pi_i\bar{\mathbf{K}}^t$. A very efficient solver of LAP can be found in (Cao 2008).

The whole procedure is formalized in Algorithm 1.

Algorithm 1 Transfer Learning by Structural Analogy

Input: $\mathbb{S}$ and $\mathbb{T}$.
Output: $\{f^{(s)}_{\sigma_s(1)}, f^{(s)}_{\sigma_s(2)}, \cdots, f^{(s)}_{\sigma_s(W)}\}$
and $\{f^{(t)}_{\sigma_t(1)}, f^{(t)}_{\sigma_t(2)}, \cdots, f^{(t)}_{\sigma_t(W)}\}$.
Feature selection in source domain using (Song 2007)
Feature selection in target domain using (Song 2007)
Compute $\bar{\mathbf{K}}^s$ and $\bar{\mathbf{K}}^t$ with all selected features together;
Initialize permutation matrix π_0;
for $i = 0$ **to** $\mathrm{MAX} - 1$ **do**
  Compute cost matrix $-\bar{\mathbf{K}}^s\pi_{i-1}\bar{\mathbf{K}}^t$;
  Solve the LAP with the cost matrix;
  Update permutation matrix as in (11);
  if converged **then**
    break;
  end if
end for

Experiments

Ohsumed Dataset

To validate our approach we conducted extensive experiments on the Ohsumed (Hersh 1994) text dataset[2]. The Ohsumed dataset consists of documents on medical issues covering 23 topics (classes) with ground truth labels on each document. The preprocessed corpus is bag-of-words data on a vocabulary of 30,689 unique words (features), and the number of documents per class ranges from 427 to 23,888.

We constructed a series of learning tasks from the dataset and attempted transfer learning between all pairs of tasks. Specifically, to construct one learning task, we selected a pair of classes (among the 23) with roughly the same number of documents (ratio between 0.9 and 1.1). One class is treated as positive examples and the other as negative. This resulted in 12 learning tasks. For each learning task, we selected 10 features (words) using (Song 2007). As these features were selected independently for each learning task, there was almost no shared features between different learning tasks, which makes it impossible to transferring knowledge using traditional transfer learning methods.

The 12 learning tasks give rise to $\frac{12\times 11}{2} = 66$ "task pairs" and we attempted transfer learning in all of them. For each task pair, we compare our method and two baselines described below:

1. **Transfer by Analogy (ours)**: We train a SVM classifier on the source learning task; find the structural analogy between the features; apply the classifier to the target learning task using the feature correspondences indicated by the analogy. Performance is measured by the classification accuracy in the target learning task.
2. **Independent Sorting**: We sort the source (and target) learning task features according to their "relevance" in the learning task itself. The relevance is measured by HSIC between the feature and the label. And the correspondences between the features are established by their ranks

[2]The dataset is downloaded from P.V. Gehler's page http://www.kyb.mpg.de/bs/people/pgehler/rap/index.html

Table 1: Experimental results on 66 transfer learning pairs. **Columns: ANAL.**: transfer by analogy (ours). **SORT**: feature correspondence from independent sorting. **RAND**: average performance over 20 random permutations. **PAIR**: The transfer learning scenarios. For example, in $1|8 \rightarrow 2|15$, the source learning task is to distinguish between class 1 and class 8, the target learning task is to distinguish between class 2 and class 15, according to the original class ID in the Ohsumed dataset.

PAIR	ANAL.	SORT	RAND	PAIR	ANAL.	SORT	RAND	PAIR	ANAL.	SORT	RAND
$1 \mid 8 \rightarrow 1 \mid 12$	60.3%	**81.4%**	64.1%	$1 \mid 8 \rightarrow 2 \mid 15$	**75.0%**	72.2%	60.1%	$1 \mid 8 \rightarrow 2 \mid 16$	54.0%	**74.7%**	68.8%
$1 \mid 8 \rightarrow 4 \mid 14$	**68.2%**	56.2%	62.1%	$1 \mid 8 \rightarrow 5 \mid 13$	**66.5%**	51.1%	58.3%	$1 \mid 8 \rightarrow 5 \mid 17$	**62.5%**	50.3%	55.5%
$1 \mid 8 \rightarrow 7 \mid 22$	85.2%	**85.4%**	65.5%	$1 \mid 8 \rightarrow 8 \mid 12$	65.8%	**69.5%**	60.5%	$1 \mid 8 \rightarrow 11 \mid 16$	**72.7%**	56.7%	56.9%
$1 \mid 8 \rightarrow 13 \mid 17$	54.3%	54.3%	**58.7%**	$1 \mid 8 \rightarrow 20 \mid 21$	**71.7%**	50.1%	58.9%	$1 \mid 12 \rightarrow 2 \mid 15$	**74.6%**	69.1%	67.2%
$1 \mid 12 \rightarrow 2 \mid 16$	52.9%	63.5%	**69.5%**	$1 \mid 12 \rightarrow 4 \mid 14$	**79.1%**	62.7%	58.3%	$1 \mid 12 \rightarrow 5 \mid 13$	**58.3%**	50.3%	57.5%
$1 \mid 12 \rightarrow 5 \mid 17$	53.0%	**64.6%**	54.8%	$1 \mid 12 \rightarrow 7 \mid 22$	54.0%	**71.8%**	61.0%	$1 \mid 12 \rightarrow 8 \mid 12$	55.4%	55.0%	**58.6%**
$1 \mid 12 \rightarrow 11 \mid 16$	59.3%	50.9%	**60.7%**	$1 \mid 12 \rightarrow 13 \mid 17$	**61.4%**	58.1%	57.6%	$1 \mid 12 \rightarrow 20 \mid 21$	**73.8%**	60.1%	66.0%
$2 \mid 15 \rightarrow 2 \mid 16$	77.3%	**81.1%**	66.5%	$2 \mid 15 \rightarrow 4 \mid 14$	**62.0%**	59.0%	60.7%	$2 \mid 15 \rightarrow 5 \mid 13$	**64.4%**	58.1%	59.8%
$2 \mid 15 \rightarrow 5 \mid 17$	**54.8%**	51.2%	54.7%	$2 \mid 15 \rightarrow 7 \mid 22$	**70.9%**	59.0%	58.2%	$2 \mid 15 \rightarrow 8 \mid 12$	53.7%	**62.4%**	59.2%
$2 \mid 15 \rightarrow 11 \mid 16$	68.1%	**73.6%**	62.2%	$2 \mid 15 \rightarrow 13 \mid 17$	50.4%	**67.1%**	58.6%	$2 \mid 15 \rightarrow 20 \mid 21$	**75.2%**	65.7%	64.4%
$2 \mid 16 \rightarrow 4 \mid 14$	57.3%	**67.4%**	55.1%	$2 \mid 16 \rightarrow 5 \mid 13$	70.4%	**71.0%**	58.3%	$2 \mid 16 \rightarrow 5 \mid 17$	50.2%	51.4%	**54.0%**
$2 \mid 16 \rightarrow 7 \mid 22$	**69.1%**	64.6%	65.6%	$2 \mid 16 \rightarrow 8 \mid 12$	**80.8%**	55.2%	61.6%	$2 \mid 16 \rightarrow 11 \mid 16$	53.7%	53.4%	**54.3%**
$2 \mid 16 \rightarrow 13 \mid 17$	**61.5%**	60.0%	57.0%	$2 \mid 16 \rightarrow 20 \mid 21$	62.3%	**68.7%**	60.2%	$4 \mid 14 \rightarrow 5 \mid 13$	**71.2%**	50.5%	58.2%
$4 \mid 14 \rightarrow 5 \mid 17$	**59.5%**	52.6%	54.0%	$4 \mid 14 \rightarrow 7 \mid 22$	**85.7%**	57.0%	63.0%	$4 \mid 14 \rightarrow 8 \mid 12$	**83.4%**	53.5%	62.2%
$4 \mid 14 \rightarrow 11 \mid 16$	**71.7%**	52.6%	56.4%	$4 \mid 14 \rightarrow 13 \mid 17$	**76.8%**	52.6%	56.4%	$4 \mid 14 \rightarrow 20 \mid 21$	**76.0%**	64.3%	58.4%
$5 \mid 13 \rightarrow 5 \mid 17$	**64.6%**	63.2%	57.1%	$5 \mid 13 \rightarrow 7 \mid 22$	**86.2%**	57.2%	64.0%	$5 \mid 13 \rightarrow 8 \mid 12$	**76.9%**	50.6%	57.3%
$5 \mid 13 \rightarrow 11 \mid 16$	**70.6%**	64.2%	56.2%	$5 \mid 13 \rightarrow 13 \mid 17$	**68.8%**	52.2%	57.0%	$5 \mid 13 \rightarrow 20 \mid 21$	**76.5%**	50.1%	59.8%
$5 \mid 17 \rightarrow 7 \mid 22$	**69.9%**	54.3%	64.9%	$5 \mid 17 \rightarrow 8 \mid 12$	61.3%	**64.8%**	58.6%	$5 \mid 17 \rightarrow 11 \mid 16$	**61.8%**	60.0%	57.8%
$5 \mid 17 \rightarrow 13 \mid 17$	**61.5%**	59.6%	57.2%	$5 \mid 17 \rightarrow 20 \mid 21$	**69.4%**	62.1%	61.2%	$7 \mid 22 \rightarrow 8 \mid 12$	67.5%	**71.5%**	58.3%
$7 \mid 22 \rightarrow 11 \mid 16$	**58.0%**	54.8%	56.7%	$7 \mid 22 \rightarrow 13 \mid 17$	**67.9%**	56.5%	58.3%	$7 \mid 22 \rightarrow 20 \mid 21$	**73.6%**	52.4%	60.2%
$8 \mid 12 \rightarrow 11 \mid 16$	**71.0%**	50.4%	58.5%	$8 \mid 12 \rightarrow 13 \mid 17$	**62.5%**	56.5%	59.0%	$8 \mid 12 \rightarrow 20 \mid 21$	**74.1%**	56.1%	59.2%
$11 \mid 16 \rightarrow 13 \mid 17$	**78.8%**	58.1%	61.4%	$11 \mid 16 \rightarrow 20 \mid 21$	58.8%	**68.8%**	59.8%	$13 \mid 17 \rightarrow 20 \mid 21$	52.8%	52.1%	**61.5%**

in their own learning tasks. Note that this baseline method uses even more information (the labels) than our "transfer by analogy" method.

3. **Random Permutation**: We randomly permutate the features and apply the classifier trained in the source task to the target task using the random correspondence. Performance is average over 20 random permutations.

Target task performances obtained in all the 66 task pairs are shown in Table 1, and some statistics are shown in Table 2. Note that when applying the classifier trained in source task to the target task, the identity of positive/negative class in the target task is not identifiable as we do not use any target domain label. And because all learning tasks have roughly balanced positive and negative examples, getting an accuracy of 20% is equivalent to 80% as they both indicate a strong correlation between the classifier output and the label. Therefore the reported target domain accuracy is always larger than or equal to 50% (with the worst case 50% indicating that the classifier output and the label are independent). From the results we can conclude that, it is statistically significant that the structural analogy approach successfully transfers knowledge between learning tasks with completely different representations (features). And the learner benefits most from the source learning task if the appropriate analogy is made.

Conclusion

In this paper we addressed the problem of transfer learning by structural analogy between two domains with completely different low-level representations. By making use of statistical tools, we tried to bridge transfer learning and the old paradigm of learning by analogy, and extend them to more general settings. The current work and our future research aim at automatically making structural analogies and determine the structural similarities with as few prior knowledge and background restrictions as possible.

Table 2: Some statistics over the 66 transfer learning pairs. Rows follow the same explanation as in columns of Table 1. **Columns: Best Count**: number of transfer learning pairs (out of 66) for which the current method outperforms the other two. **Ave. Accuracy**: average target task accuracy among all 66 transfer learning pairs. **Margin**: The margin by which the current method beats the other two methods averaged among all pairs counted in "Best Count" (because in practice we only care about the performance when "positive transfer" occurs).

METHODS	BEST COUNT	AVE. ACCURACY	MARGIN
ANALOGY	**43** / 66	**66.6%**	**9.7%**
SORT	16 / 66	60.2%	6.4%
RAND	7 / 66	59.8%	3.9%

Acknowledgement

We thank the support of Hong Kong RGC/NSFC project N_HKUST 624/09 and RGC project 621010.

References

Agnar Aamodt and Enric Plaza, Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, *Artificial Intelligence Communications* 7 : 1, 39-52. 1994.

D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, *Machine Learning*, 6, 37-66, 1991.

K.D. Ashley, Reasoning with Cases and Hypotheticals in Hypo, In *International Journal of Man?Machine Studies*, Vol. 34, pp. 753-796. Academic Press. New York. 1991

S. Bayoudh, L. Miclet, and A. Delhay, Learning by Analogy: a Classification Rule for Binary and Nominal Data. In *IJCAI*, 2007.

J. Blitzer, R. McDonald, F. Pereira, Domain Adaptation with Structural Correspondence Learning, In *EMNLP* 2006, pages 120-128,

Yi Cao, Munkres' Assignment Algorithm, Modified for Rectangular Matrices. http://csclab.murraystate.edu/bob.pilgrim/445/munkres.html

Jaime G. Carbonell, A Computational Model of Analogical Problem Solving, In *IJCAI*, 1981.

Jaime G. Carbonell, Oren Etzioni, Yolanda Gil, Robert Joseph, Craig A. Knoblock, Steven Minton, Manuela M. Veloso, PRODIGY: An Integrated Architecture for Planning and Learning, *SIGART Bulletin* 2(4): 51-55, 1991.

R. Caruana, Multitask learning, *Machine Learning*, 28(1):41-75, 1997.

W. Cheetham, K. Goebel, Appliance Call Center: A Successful Mixed-Initiative Case Study, *Artificial Intelligence Magazine*, Volume 28, No. 2, (2007). pp 89-100.

Wenyuan Dai, Yuqiang Chen, Gui-Rong Xue, Qiang Yang, and Yong Yu, Translated Learning: Transfer Learning across Different Feature Spaces. In *NIPS* 2008.

H. Daumé III, D. Marcu, Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26, 101C126. 2006.

Brian Falkenhainer, Kenneth D. Forbus, and Dedre Gentner The Structure-Mapping Engine: Algorithm and Examples *Artificial Intelligence*, 41, page 1-63, 1989.

K.D. Forbus, D. Gentner, A.B. Markman, and R.W. Ferguson, Analogy just looks like high-level perception: Why a domain-general approach to analogical mapping is right. *Journal of Experimental and Theoretical Artificial Intelligence*, 10(2), 231-257, 1998.

D. Gentner, The mechanisms of analogical reasoning. In J.W. Shavlik, T.G. Dietterich (eds), *Readings in Machine Learning*. Morgan Kaufmann, 1990.

Arthur Gretton, Olivier Bousquet, Alex Smola and Bernhard Schölkopf, Measuring Statistical Dependence with Hilbert-Schmidt Norms In *Algorithmic Learning Theory, 16th International Conference*. 2005.

Arthur Gretton, Kenji Fukumizu, Choon Hui Teo, Le Song, Bernhard Schölkopf and Alex Smola A Kernel Statistical Test of Independence In *NIPS* 2007.

William Hersh, Chris Buckley, TJ Leone, David Hickam, OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research In *SIGIR* 1994.

K.J. Holyoak, and P. Thagard, The Analogical Mind. In *Case-Based Reasoning: An Overview* Ramon Lopez de Montaras and Enric Plaza, 1997.

Janet Kolodner, Reconstructive Memory: A Computer Model, *Cognitive Science* 7, (1983): 4.

Janet Kolodner, *Case-Based Reasoning*. San Mateo: Morgan Kaufmann, 1993.

D.B. Leake, *Case-Based Reasoning: Experiences, Lessons and Future Directions* MIT Press, 1996.

M. M. Hassan Mahmud and Sylvian R. Ray Transfer Learning using Kolmogorov Complexity: Basic Theory and Empirical Evaluations In *NIPS* 2007.

Bill Mark, Case-Based Reasoning for Autoclave Management, In *Proceedings of the Case-Based Reasoning Workshop* 1989.

D.W. Patterson, N. Rooney, and M. Galushka, Efficient similarity determination and case construction techniques for case-based reasoning, In S. Craw and A. Preece (eds): *EC-CBR*, 2002, *LNAI 2416*, pp. 292-305.

N. Quadrianto, L. Song, A.J. Smola, Kernelized Sorting. In *NIPS* 2008.

B. Schölkopf, A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2001

A.J. Smola, A. Gretton, Le Song, and B. Schölkopf, A hilbert space embedding for distributions. In E. Takimoto, editor, *Algorithmic Learning Theory* , *Lecture Notes on Computer Science*. Springer, 2007.

Le Song, A.J. Smola, A. Gretton, K. Borgwardt, and J. Bedo, Supervised Feature Selection via Dependence Estimation. In *ICML* 2007.

Roger Schank, *Dynamic Memory: A Theory of Learning in Computers and People* New York: Cambridge University Press, 1982.

Sebastian Thrun and Lorien Pratt (eds), *Learning to Learn*, Kluwer Academic Publishers, 1998.

Ian Watson, *Applying Case-Based Reasoning: Techniques for Enterprise Systems.* Elsevier, 1997.

P.H. Winston, Learning and Reasoning by Analogy, *Communications of the ACM*, 23, pp. 689-703, 1980.