Inductive Transfer

Ricardo Vilalta
Department of Computer Science, University of Houston, USA

Christophe Giraud-Carrier
Department of Computer Science, Brigham Young University, USA

Pavel Brazdil, Carlos Soares
LIAAD-INESC Porto L.A./Faculdade de Economia, University of Porto, Portugal

Synonyms

Transfer learning, transfer of knowledge across domains, multitask learning, domain adaptation

Abstract

We describe different scenarios where a learning mechanism is capable of acquiring experience on a source task, and subsequently exploit such experience on a target task. The core ideas behind this ability to transfer knowledge from one task to another have been studied in the machine learning literature under different titles and perspectives. Here we describe some of them under the names of inductive transfer, transfer learning, multitask learning, meta-searching, meta-generalization, and domain adaptation.

Definition

Inductive transfer refers to the ability of a learning mechanism to improve performance on the current or *target* task after having learned a different but related concept or skill on a previous *source* task. Transfer may additionally occur between two or more learning tasks that are being undertaken concurrently. The object being transferred may refer to instances, features, a particular form of search bias, an action policy, background knowledge, etc.

Motivation and Background

Learning is not the result of an isolated task that starts from scratch with every new problem. Instead, a learning algorithm should exhibit the ability to adapt through a mechanism dedicated to transfer knowledge gathered from previous experience. The problem of transfer of knowledge is central to the field of machine learning and is also known as *inductive transfer*. In this case, knowledge can be understood as a collection of patterns observed across tasks. One view of the nature of patterns across tasks is that of invariant transformations. For example, image recognition of a target object is simplified if the object is invariant under rotation, translation, scaling, etc. A learning system should be able to recognize a target object on an image even if previous images show the object in different sizes or from different angles. Hence, inductive transfer studies how to improve learning by detecting, extracting, and exploiting (meta)knowledge in the form of invariant transformations across tasks.

Similarly, in competitive games involving teams of robots (e.g., Robocup Soccer), transferring knowledge learned from one task to another task is crucial to acquire skills necessary to beat the opponent team. Specifically, imagine a situation where a team of robots has been taught to keep a soccer ball away from the opponent team. To achieve that goal, robots must learn to keep the ball, pass the ball to a close teammate, etc. always trying to remain at a safe distance from the opponents. Now let us assume that we wish to teach the same team of robots to be efficient at scoring against a team of defending robots. Knowledge gained during the first activity can be transferred to the second one. Specifically, a robot can prefer to perform an action learned in the past over actions proposed during the current task, because the past action has a significant higher merit value. For example, a robot under the second task may learn to recognize that it is preferable to shoot than to pass the ball because the goal is very close. This action can be learned from the first task by recognizing that the precision of a pass is contingent upon the proximity of the teammate.

Structure of the Learning System

The main idea behind a learning architecture using knowledge transfer is to produce a source model from which knowledge can be extracted and transferred to a target model. This allows for multiple scenarios (Brazdil, et. al. (2009) [], Pratt and Thrun (1997) []). For example, the target and source models can be trained at different times in such a way that the transfer takes place after the source model has been trained. In this case there is an explicit form of knowledge transfer, also called *representational transfer*. In contrast, we use the term *functional transfer* to denote the case where two or more models are trained simultaneously; in this case the models share (part of) their internal structure during learning (see Neural Networks below). Under representational transfer, we denote as *literal transfer* the case when the source model is left intact, and as *non-literal transfer* the case when the source model is modified before knowledge is transferred to the target model. In non-literal transfer some processing takes place on the source model before it is used to initialize the target model (see Fig. 1).

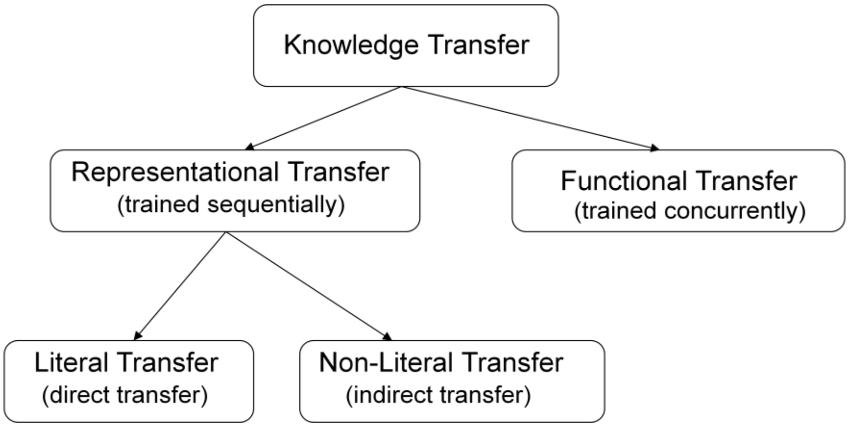

Fig.1. A taxonomy of inductive transfer.

Neural Networks. A learning paradigm amenable to test the feasibility of knowledge transfer is that of neural networks (Caruana (1993) []). A popular form of (functional) knowledge transfer is effected through multitask learning, where the output nodes in the multilayer network represent more than one task. In such a scenario, internal nodes are shared by different tasks dynamically during learning. As an illustration, consider the problem of learning to classify astronomical objects from images mapping the sky into multiple classes. One task may be in charge of classifying a star into several classes (e.g., main sequence, dwarf, red giant, neutron, pulsar, etc.). Another task can focus on galaxy classification (e.g., spiral, barred spiral, elliptical, irregular, etc.). Rather than separating the problem into different tasks where each task is in charge of identifying one type of luminous object, one can combine the tasks together into a single parallel multi-task problem where the hidden layer of a neural network shares patterns that are common to all classification tasks (see Fig. 2). The reasons explaining why learning often improves in accuracy and speed in this context is that training with many tasks in parallel on a single neural network induces information that accumulates in the training signals; if there exists properties common to several tasks, internal nodes can serve to represent common sub-concepts simultaneously.

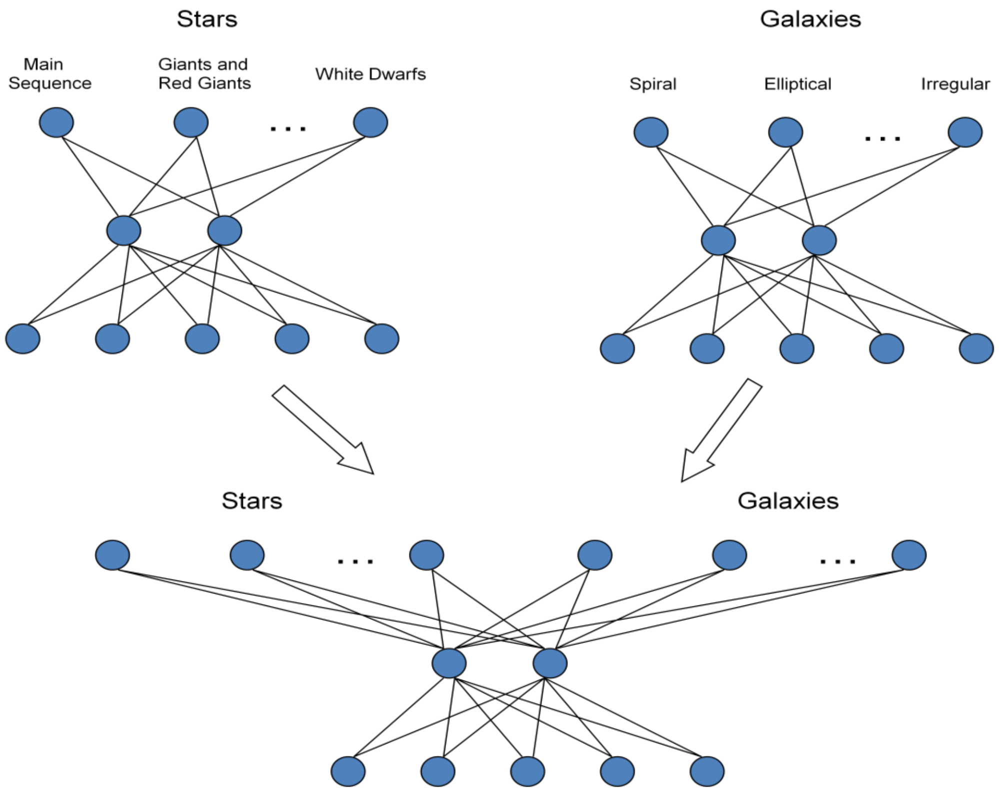

Fig.2. Example of multitask learning (functional transfer) applied to astronomical images.

Other Paradigms. Knowledge transfer can be performed using other learning and data-analysis paradigms --mainly in the form of representational transfer-- such as kernel methods, probabilistic methods, clustering, etc (Raina, et al. (2006) [], Evgeniou, et al. (2007) []). For example, inductive transfer can take place in learning methods that assume a probabilistic distribution of the data by guaranteeing a form of relatedness among the distributions adopted across tasks (Raina, et al. (2006) []). As an illustration, if learning to classify stars and galaxies both assume a mixture of normal densities to model the input-output or example-class distribution, one can force both distributions to have sets of parameters that are as similar as possible while preserving good generalization performance. In that case, shared knowledge can be interpreted as a set of assumptions about the data distribution for all tasks under analysis. The concept of knowledge-transfer is also related to the problem of introducing new intermediate concepts during rule induction. In the Inductive Logic Programming (ILP) setting this is referred to as *predicate invention* (Stahl (1995) []).

Meta-Searching for Problem Solvers. A different research direction in inductive transfer explores complex scenarios where the software architecture itself evolves with experience (Schmidhuber (1997) []). The main idea is to divide a program into different components that can be re-used during different stages of the learning process. As an illustration, one can work within the space of (self-delimiting binary) programs to propose an optimal ordered problem solver. The goal is to solve a sequence of problems, deriving one solution after the other, as optimally as possible. Ideally the system should be capable of exploiting previous solutions and of incorporating them into the solution to the current problem. This can be done by allocating computing time to the search for previous solutions that, if useful, become transformed into building blocks. We assume the current problem can be solved by copying or invoking previous pieces of code (i.e., building blocks or knowledge). In that case the mechanism will accept those solutions with substantial savings in computational time.

Domain Adaptation. A recent research direction in representational transfer seeks to adjust the model obtained in a source domain to account for differences exhibited in a new target domain. Unlike traditional studies in classification where both training and testing sets are assumed as realizations of the same joint input-output distribution, this *domain adaptation* approach either weakens or completely disregards such assumption (Ben-David, et al. (2007)[], Daumé, et al. (2006)[], Storkey (2009)[]). In addition, domain adaptation commonly assumes an abundance of labeled examples in the source domain, but little or no class labels in the target domain.

An example of these concepts lies in light curve classification from star samples obtained from different galaxies. A classification task set to differentiate different types of stars in a nearby source galaxy --where class labels are available-- will experience a change in distribution as it moves to a target galaxy lying farther away --where class labels are unavailable. A major reason for such change is that at greater distances, less luminous stars fall below the detection threshold and more luminous stars are preferentially detected. The corresponding dataset shift (Quinonero et al. (2009)[]) precludes the direct utilization of one single model across galaxies; it calls for a form of model adaptation to compensate for the change in the data distribution.

Domain adaptation has gained much attention recently, mainly due to the pervasive character of problems where distributions change over time. It assumes that the learning task remains constant, but the marginal and class posterior distributions between source and target domain may differ (as opposed to traditional transfer learning where tasks can in addition exhibit different input representations, i.e., different input spaces). Domain adaptation has been attacked from different angles: by searching for a single representation that unifies both source and target domains (Glorot, et al. (2011)[]); by proving error bounds as a function of empirical error and the distance between source and target distributions (Ben-David et al. (2010)[]); within a co-training framework where target vectors are incorporated into the source training set based on confidence (Chen, et al. (2011)[]); by re-weighting source instances (Mansour, et al. (2009)[]); by using regularization terms to learn models that perform well on both source and target domains (Daumé, et al. (2010)[]); and several others.

Theoretical Work

Several studies have provided a theoretical analysis of the case where a learner uses experience from previous tasks to learn a new task. This process is often referred to as meta-learning or meta-generalization. The aim is to understand the conditions under which a learning algorithm can provide good generalizations when embedded in an environment made of related tasks. Although the idea of knowledge transfer is normally made implicit in the analysis, it is clear that the meta-learner extracts and exploits knowledge on every task to perform well on future tasks. Theoretical studies fall within a Bayesian model, and within a Probably Approximately Correct (PAC) model. The idea is to find not only the right hypothesis in a hypothesis space (base learning), but in addition to find the right hypothesis space in a family of hypothesis spaces (meta-learning).

We briefly review the main ideas behind these studies (Baxter (2000) [1]). We begin by assuming that the learner is embedded in a set of related tasks that share certain commonalities. Going back to the problem where a learner is designed for recognition of astronomical objects, the idea is to classify objects (e.g., stars, galaxies, nebulae, and planets) extracted from images mapping certain region of the sky. One way to transfer learning experience from one astronomical center to another is by sharing a meta-learner that carries a bias towards recognition of astronomical objects. In traditional learning, we assume a probability distribution **p** that indicates which examples are more likely to be seen in such a task. Now we assume that there is a more general distribution **P** over the space of all possible distributions. In essence, the meta-distribution **P** indicates which tasks are more likely to be found within the sequence of tasks faced by the meta-learner (distribution **p** indicates which examples are more likely to be seen in one task). In our example, the meta-distribution **P** peaks over tasks corresponding to classification of astronomical objects. Given a family of hypothesis spaces **{H}**, the goal of the meta-learner is to find a hypothesis space **H*** that minimizes a functional risk corresponding to the expected loss of the best possible hypothesis in each hypothesis space. In practice, since we ignore the form of **P,** we need to draw samples $\mathbf{T_1, T_2, \ldots, T_n}$ to infer how tasks are distributed in our environment. To summarize, in the transfer learning scenario our input is made of samples $\mathbf{T = \{T_i\}}$, where each sample $\mathbf{T_i}$ is composed of examples. The goal of the meta-learner is to output a hypothesis space with a learning bias that generates accurate models for a new task.

Future Directions

The research community faces several challenges on how to efficiently transfer knowledge across tasks. One challenge involves devising learning architectures with an explicit representation of knowledge about models and algorithms, i.e., meta-knowledge. Most systems that integrate knowledge transfer mechanisms make an implicit assumption about the type of knowledge being transferred. This is indeed possible when strong assumptions are made on the relationship between the source and target tasks. For example, most approaches to domain adaptation work under strong assumptions about the similarity between the source and target tasks, imposing similar class posterior distributions, marginal distributions, or both. Ideally we would like to track the evolution of the source task to the target task to be able to justify any assumptions about their differences.

From a global perspective, it seems clear that proper treatment of the inductive transfer problem requires more than just statistical or mathematical techniques. Inductive transfer can be embedded in a complex artificial intelligence system that incorporates important components such as knowledge representation, search, planning, reasoning, etc. Without the incorporation of artificial intelligence components, we are forced to work with a large hypothesis space and a set of stringent assumptions about the nature of the discrepancy between the source and target tasks.

See Also: Meta-learning, Concept Drift

Recommended Reading

1. Baxter, J. (2000). A Model of Inductive Learning Bias. *Journal of Artificial Intelligence Research*, **12**, pp. 149-198.

2. Ben-David, S. and Blitzer J. and Crammer K. and Pereira, F. (2007). Analysis of Representations for Domain Adaptation, *Advances in Neural and Information Processing Systems* (*NIPS)*, pp. 137-144.

3. Ben-David, S. and Blitzer, J. and Crammer, K. and Kulesza, A. and Pereira, F. and Wortman, J. (2010). A Theory of Learning from Different Domains, *Machine Learning, Special Issue on Learning from Multiple Sources*, 79 pp. 151--175.

4. Brazdil, P. and Giraud-Carrier, C. and Soares, C. and Vilalta, R. (2009). *Metalearning: Applications to Data Mining*. Springer.

5. Caruana, R. (1993). Multitask Learning: A Knowledge-based Source of Inductive Bias. In *Proceedings of the 10th International Conference on Machine Learning (ICML)*, pp. 41-48.

6. Chen, M. and Weinberger, K. Q. and Blitzer, J. (2011). Co-Training for Domain Adaptation, *Advances in Neural Information Processing Systems* (*NIPS).*

7. Daumé, H. and Marcu, D. (2006). Domain Adaptation for Statistical Classifiers. *Journal of Machine Learning Research*: 26, pp. 102-126.

8. Daumé, H. and Kumar A. and Saha, A. (2010). Co-regularization Based Semi-supervised Domain Adaptation, *Advances in Neural Information Processing Systems (NIPS).*

9. Glorot, X. and Bordes, A. and Bengio, Y. (2011). Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach, *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pp. 513-520.

10. Mansour, Y. and Mohri, M. and Rostamizadeh, A. (2009). Domain Adaptation with Multiple Sources, *Advances in Neural Information Processing Systems* (*NIPS)*, pp. 1041--1048.

11. Mihalkova, L. and Huynh, T. and Mooney, R.J. (2007). Mapping and Revising Markov Logic Networks for Transfer Learning. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pp. 608-614.

12. Oblinger, D. and Reid, M. and Brodie, M. and de Salvo Braz, R. (2002). Cross-training and its Application to Skill-Mining. *IBM Systems Journal* **41**, No. 3, pp. 449-460.

13. Pratt, L. and Thrun, S. (1997). Second Special Issue on Inductive Transfer. *Machine Learning,* **28**.

14. Quinonero-Candela, J. and Sugiyama, M. and Schwaighofer, A. and Lawrence, N. D. *Dataset Shift in Machine Learning*, MIT Press, 2009.

15. Raina, R. and Ng, A. Y. and Koller, D. (2006). Constructing Informative Priors using Transfer Learning. In *Proceedings of the 23rd International Conference on Machine Learning* (*ICML)*, pp. 713-720.

16. Reid, M. (2004). Improving Rule Evaluation Using Multitask Learning. In *Proceedings of the 14th International Conference on ILP*, pp. 252-269.

17. Schmidhuber, J. (1997). Shifting Inductive Bias with Success-Story Algorithm, Adaptive Levin Search, and Incremental Self-Improvement. *Machine Learning*, **28**, pp. 105-130.

18. Storkey, A. (2009). When Training and Test Sets are Different, in: Quinonero-Candela, J. and Sugiyama, M. and Schwaighofer, A. and Lawrence, N. D. (Eds.), *Dataset Shift in Machine Learning*, MIT Press, pp. 3-28.

19. Dai, W. and Yang, Q. and Xue, G. and Yu, Y. (2007). Boosting for Transfer Learning. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pp. 193–200.

20. Evgeniou T. and Micchelli, C. A. and Pontil, M. (2005). Learning Multiple Tasks with Kernel Methods. *Journal of Machine Learning Research*, **6**, pp. 615-637.

21. Stahl. I. (1995). Predicate Invention in Inductive Logic Programming. In L. De Raedt (ed.) *Advances in Inductive Logic Programming*, pp. 34-47. IOS Press.